Abstract - In this paper, we discuss the speed control of a 3-phase induction motor using PI (Proportional Integral) control method and fuzzy logic control method. The induction motor was designed using indirect vector control. We simulated our induction motor model using MATLAB/SIMULINK. We obtained the speed control results using PI controller. Next simulated our induction motor model using fuzzy logic controller. The result obtained was compared with that of PI.

Keywords - Fuzzy logic, Induction motor, scalar control, simulink, speed control, Proportional Integral, Fuzzy logic controller, indirect vector control.

I. INTRODUCTION

Induction motor is the most common electrical machine used in modern industries. It has gained such popularity due to its various advantages. The various advantages are high efficiency, low cost, good self-starting [1], simplicity in design, the absence of the collector brooms system, and a small inertia. Even though it has a wide range of advantages it has quite a few disadvantages. Induction motor has disadvantages, such as complex, multivariable and nonlinear of mathematical model of induction motor, and the induction motor is not inherently capable of providing variable speed operation.

The interest in sensorless drives of induction motor (IM) has grown significantly over the past few years due to some of their advantages, like mechanical robustness, less maintenance, and basic(easy) construction. These applications include paper and textile mills, pumps and fans, electric and hybrid vehicles, subway and locomotive propulsions, machine tools and robotics, home appliances, air conditioners and heat pumps, rolling mills, wind generation systems. Therefore, Induction motors have been used more in the industrial variable speed drive system with the development of the vector control technology.

Therefore the electrical drives good dynamic performance is mandatory so as to respond the changes in command speed and torques. The most commonly used controller for the speed control of Induction motor is Proportional plus Integral (PI) controller.

However, the PI controller has some demerits such as: sensitivity to controller gains, the high starting overshoot and sluggish response due to sudden disturbance. Therefore to overcome these problems, replacement of PI controller by an intelligent controller based on Fuzzy set theory [5] is proposed in this work. The fuzzy logic controller has certain advantages compared to the classical controllers such as low cost, simplicity of control, and the possibility to design without knowing the exact mathematical model.

The vector control methods are normally classified into two types, direct vector control and indirect vector control methods. In this paper the Indirect Vector Control method is used. Because of it’s, fast dynamic response and speed response without cogging or torque pulsations at low speed, smooth speed reversal under any torque conditions.

A new control of a drive system is formulated and it is often convenient to study the system performance by using Matlab/Simulink package, when compared to other packages like EMTP and PSPICE. The simulation using Matlab has an ease in modeling, the transients of electrical machines and drives and to include controls in the simulation [3]. Then the superior control performance of the proposed controllers is demonstrated at Simulink platform using the Fuzzy logic tool box for different operating conditions.

II. CONTROLLING SCHEMES FOR SPEED CONTROL OF THREE PHASE INDUCTION MOTOR

A. Scalar Control

Scalar control is due to magnitude variation of the control variable only, and it disregards the coupling effect in machine. The voltage of machine can be controlled to control the flux, and frequency or slip. It can be used to control the torque. However, torque and flux are also function of voltage and frequency respectively. A scalar controlled drive gives an inferior performance and is easy to implement. But the inherent coupling effect (both torque and flux are function of voltage or current and frequency) gives sluggish response and system is easily prone to instability because of higher order (fifth order) system effect.
If torque is increased by incrementing the slip then the flux tends to decrease [2]. It has been noted that the flux variation is also sluggish. Decreases in flux then compensated by the sluggish flux control loop feeding an additional voltage.

**B. Vector Control or Field Oriented Control (FOC):**

In DC machine the field flux is perpendicular to the armature flux. These two fluxes produce no net interaction on one another, because they are orthogonal. Adjusting the field current can be used for controlling the DC machine flux, and thus torque can also be controlled independently of flux by adjusting the armature current. An AC machine is not simple because of the interactions between the stator and the rotor fields. Due to this the orientations are not held at 90 degrees but it varies with the operating conditions. To obtain DC machine-like performance in holding a fixed and orthogonal orientation between the field and armature in an AC machine can be obtained by orienting the stator current with respect to the rotor flux so as to attain independently controlled flux and torque. This control scheme is called flux-oriented control or vector control. Vector control is applicable to both synchronous and induction motors.

**C. Proportional – Integral (PI) Control**

In this project complete mathematical model of induction motor is described and simulated in Matalab. The performance of vector control drive with proportional plus integral (PI) controller is analyzed. Control law used for this strategy is given by

\[ T = K_p e + K_i \int e \, dt \]

Its output is the updating in PI controller gains (Kp and Ki) based on a set of rules to maintain excellent control performance even in the presence of drive nonlinearity and parameter variation. The use of PI controllers for speed control of induction machine drives is characterized by an overshoot during tracking mode and a poor load disturbance rejection [2]. This is mainly due to the fact that the complexity of the system does not allow the gains of the PI controller to exceed a certain low value and at the starting mode the high value of the error is amplified across the PI controller causing annoyance high variations in the command torque. Whenever the gains of the controller exceed a certain value, the variations in the command torque become too high and will destabilize the system. In order to overcome this problem we propose the use of a limiter ahead of the PI controller [2]. This limiter causes the speed error to be maintained within the saturation limits provoking, when appropriately chosen, smooth variations in the command torque even when the PI controller gains are very high [2].

The motor reaches the reference speed rapidly and without overshoot, when step commands are tracked with almost zero steady state error and no overshoot. When load disturbances are rapidly rejected and variations of some of the motor parameters are fairly well dealt with.

**D. Fuzzy Logic Control**

Due to continuously developing automation systems and more demanding small control performance requirements, conventional control methods are not always adequate. In another word, practical control problems are usually imprecise. The input output relations of the system may be uncertain and they can be changed by unknown external disturbances. Thus a more efficient control scheme called fuzzy logic was implemented. Fuzzy control is based on fuzzy logic [4]. Which provides an efficient method to handle in exact information as basis reasoning. With the help of fuzzy logic it is possible to convert knowledge, which is expressed in an uncertain form, to an exact algorithm. Fuzzy logic embodies human-like thinking into a control system. A fuzzy controller employs a mode of approximate reasoning resembling the decision making route of humans (i.e. The process people use to infer conclusions from what they know) [2]. Fuzzy control has been primarily applied to the control of processes through fuzzy linguistic descriptions stipulated by membership functions.

**III. INDIRECT VECTOR CONTROL INDUCTION MOTOR DRIVE - MATHEMATICAL MODEL:**

**A. Hysteresis Current Regulator:**

The current regulator, which consists of three hysteresis controllers, is built with Simulink blocks [2]. The motor actual currents are cater by the measurement output of the Asynchronous Machine block. The actual motor currents and reference current are compared in hysteresis type relay.

![Fig 1: Hysteresis Current Regulator block diagram.](image)
B. Flux Calculation Block

Fig 2: Flux Calculation block diagram

\[ L_r = L_{ir} + L_m = 0.8 + 34.7 = 35.5 \text{ mH} \]
\[ L_m = 34.7 \text{ mH} \]
\[ T_r = \frac{L_r}{R_r} = 0.1557 \text{ sec} \]
\[ R_r = 0.228 \Omega \]
\[ \Phi_{ir} = \frac{L_m \cdot I_d}{1 + T_r \cdot s} \]

C. Universal Bridge (IGBT) Inverter:

The Universal Bridge block implements a universal three-phase power converter that consists of up to six power switches connected in a bridge configuration. The type of power switch and converter configuration is selectable from the dialog box. Power Electronic device and Port configuration options are selected as IGBT/Diode and ABC as output terminals respectively. The DC link input voltage is represented by a 780 V DC voltage source [2].

D. Theta Calculation Block:

The rotor flux position (\( \theta_e \)) is calculated.

Fig 4: \( \theta_e \) Calculation block diagram

\[ T_{eta} = \text{Electrical angle} = \int (w_r + w_m) \]
\[ w_r = \text{Rotor frequency in rad/s} = \frac{L_m \cdot I_q}{(T_r \cdot \Phi_{ir})} \]
\[ w_m = \text{Rotor mechanical speed in rad/s} \]
\[ L_m = 34.7 \text{ mH} \]
\[ L_r = L_{ir} + L_m = 0.8 + 34.7 = 35.5 \text{ mH} \]
\[ R_r = 0.228 \text{ ohms} \]
\[ T_r = \frac{L_r}{R_r} = 0.1557 \text{ s} \]

E. \( d-q \) to abc Transformation Blocks

Fig 5: \( d-q \) to abc transformation block diagram

F. abc to \( d-q \) Transformation Blocks

Fig 6: abc to \( d-q \) transformation block diagram
IV. ALGORITHM OF INDIRECT VECTOR CONTROL

1. The induction motor is fed by a variable frequency, variable voltage PWM inverter [1], which operates in current control mode [1]. The motor speed \( \omega \) is examined with the reference speed \( \omega^* \) and the error is produced which is fed to the speed controller. The output of speed controller is electromagnetic torque \( T_e^* \).

2. The quadrature-axis stator current reference \( i_{qs}^* \) is calculated from electromagnetic torque reference \( T_e^* \) as

\[
i_{qs}^* = \left( \frac{2}{3} \right) \left( \frac{L_m}{L_n} \right) \left( T_e^*/\phi_i \right)\]

Where \( \phi_i = |\psi_r|_{est} \) is the estimated value of rotor flux linkage given by,

\[
\phi_i = \frac{L_m i_{ds}}{1 + \tau_s s}
\]

Where, \( \tau_r = L_r/R_r \) is the rotor time constant.

3. The direct-axis stator current reference \( i_{ds}^* \) is obtained from reference rotor flux input \( |\psi_r|^* \).

4. The rotor flux position \( \theta_e \) required for coordinates transformation is obtained from the rotor speed \( \omega_r \) and slip frequency \( \omega_s \). \( \theta_e \) is calculated as

\[
\theta_e = \int \omega_r dt = \int (\omega_r + \omega_s) dt = \theta_r + \theta_s
\]

5. The slip frequency is calculated from the stator current reference \( i_{qs}^* \) and the motor parameters. is given by from equation [1],

\[
\omega_s = \frac{L_m R_i}{\phi_i L_n} i_{qs}^*
\]

6. The \( i_{qs}^* \) and \( i_{ds}^* \) current references are converted into phase current references \( i_a^*, i_b^*, i_c^* \) using inverse park transform (two phase to three phase conversion) & fed to the current controller [1]. The controller processes the measured and reference currents to produce the inverter gating signals.

The main role of the speed controller is to keep the motor speed equal to the speed reference input in steady state and to provide a good dynamic response during transients & the closed loop current control operates the voltage source inverter in current control mode which ensures that the winding current follow same pattern as that reference currents generated from control logic.

V. DESIGN OF FUZZY LOGIC CONTROLLER

The fuzzy logic controller consists of four blocks, Fuzzification, inference mechanism, knowledge base and Defuzzification.

a) Fuzzifications: In this stage the crisp variables of input \( e(t) \) and \( ce(t) \) are converted into fuzzy variables. The fuzzification maps the error and change in error to linguistic labels of fuzzy sets. Membership function is associated to each label with triangular shape which consists of two inputs and one output. The proposed controller uses following linguistic labels NL, NM, ZE, PM, PL. Each of the inputs and output contain membership function with all these five linguistics.

b) Knowledge base and inference stage: Knowledge base involves defining the rules represented as IF-THEN rules statements governing the relationship between input and output variables in terms of membership function. In this stage the input variables \( e(t) \) and \( ce(t) \) are processed by the Inference mechanism that executes rules represented in rule table1 shown below. Mamdani fuzzy logic algorithm is used for the rule base computation.
c) **Defuzzification:** This stage introduces different methods that can be used to produce fuzzy set value for the output fuzzy variable. Defuzzification interface performs the following functions.

1. A scale mapping, which converts the range of values of output variables into corresponding universe of discourse.
2. Defuzzification, which yields a non-fuzzy control action from an inferred fuzzy control action.

Each universe of discourse is divided into seven overlapping fuzzy sets: NL (Negative Large), NM (Negative Medium), ZE (Zero), PM (positive Medium), and PL (Positive Large) \(^3\). Each fuzzy variable is a member of the subsets with a degree of membership \(\mu\) varying between 0 (non-member) and 1 (full-member).

![Fig 8: “speed_error” membership function](image)

![Fig 9: “change_in_error” membership function](image)

![Fig 10: “control” membership function](image)

<table>
<thead>
<tr>
<th>(e)</th>
<th>(\Delta e)</th>
<th>NL</th>
<th>NM</th>
<th>ZE</th>
<th>PM</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>NLO</td>
<td>NMO</td>
<td>PMO</td>
<td>PLO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>NLO</td>
<td>NMO</td>
<td>ZE</td>
<td>PMO</td>
<td>PLO</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>NLO</td>
<td>NLO</td>
<td>NMO</td>
<td>PLO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Rule Statement:**

1. If (speed\_error is NL) and (change\_in\_error is N) then (control is NLO)
2. If (speed\_error is NM) and (change\_in\_error is N) then (control is NMO)
3. If (speed\_error is ZE) and (change\_in\_error is N) then (control is PMO)
4. If (speed\_error is PM) and (change\_in\_error is N) then (control is PLO)
5. If (speed\_error is PL) and (change\_in\_error is N) then (control is PLO)
6. If (speed\_error is NL) and (change\_in\_error is Z) then (control is NLO)
7. If (speed\_error is NM) and (change\_in\_error is Z) then (control is NMO)
8. If (speed\_error is ZE) and (change\_in\_error is Z) then (control is ZE)
9. If (speed\_error is PM) and (change\_in\_error is Z) then (control is PMO)
10. If (speed\_error is PL) and (change\_in\_error is Z) then (control is PLO)
11. If (speed\_error is NL) and (change\_in\_error is P) then (control is NLO)
12. If (speed\_error is NM) and (change\_in\_error is P) then (control is NLO)
13. If (speed\_error is ZE) and (change\_in\_error is P) then (control is NMO)
14. If (speed\_error is PM) and (change\_in\_error is P) then (control is PLO)
15. If (speed_error is PL) and (change_in_error is P) then (control is PLO)

VI. MATLAB SIMULATION

A. MATLAB Simulation of Indirect Vector Control
Induction Motor based on PI controller:

The simulation of IVCIM using PI controller was done as diagram shown in the below figure. The IM was simulated at four various conditions. First at reference speed of 120 and no load. Second at speed varied from 120 to 160 after a time \( t_1(s) \) and no load. Third by varying the load after a time \( t_2(s) \) and keeping the speed at 120. The final condition being variation of load at a time \( t_3(s) \) and variation of speed from 120 to 160 after a time \( t_4(s) \). The feedback is given to summer, output of summer error given to PI controller which change the in output for better result.

B. MATLAB Simulation of Indirect Vector Control
Induction Motor based on Fuzzy controller:

The simulation of IVCIM using Fuzzy controller was done as diagram shown in the below figure. The IM was simulated at four various conditions. First at reference speed of 120 and no load. Second at speed varied from 120 to 160 after a time \( t_1(s) \) and no load. Third by varying the load after a time \( t_2(s) \) and keeping the speed at 120. The final condition being variation of load at a time \( t_3(s) \) and variation of speed from 120 to 160 after a time \( t_4(s) \). Fuzzy logic controller block is used which have two input Speed_Error and rate of change in error and one output.

VII. EXPERIMENTAL RESULT

A. Performance of Indirect Vector Control IM Using P-I Control:

The performance characteristic of a 50 hp, 460 V, 60 Hz IM, operating at various conditions with a PI speed controller was done and the following were the results:

**Case 1: At No Load and Reference Speed of 120 RAD/S**

Speed at first reaches 140 rad/sec and then settles down at reference speed 120 rad/sec. The speed attains steady state at time \( t = 2.2s \)
Case 2: AT NO LOAD AND SPEED VARYING FROM 120 RAD/S TO 160 RAD/S

Speed at first reaches 140 rad/sec and then settles down at reference speed 120 rad/sec.

The speed attains steady state at time t = 2.1s

Speed is then varied to 160 rad/s after a time t = 7s.

After variation the speed reaches 170 rad/s and then reaches the reference speed 160 rad/s

After changing the reference speed to 160 rad/s the steady state speed is obtained at t = 1.4s

Case 3: LOAD APPLIED AFTER A TIME T(S) AND REFERENCE SPEED OF 120 RAD/S

Speed at first reaches 140 rad/sec and then settles down at reference speed 120 rad/sec.

The speed attains steady state at time t = 2.1s

Then the load is applied after a time t = 14s.

Speed after variation dips to 105 rad/s and then settles to reference speed 120 rad/s

After applying the load the steady state speed is obtained at t = 1.2s

Case 4: LOAD APPLIED AFTER A TIME T(S) AND SPEED VARYING FROM 120 RAD/S TO 160 RAD/S

Speed at first reaches 140 rad/sec and then settles down at reference speed 120 rad/sec.

The speed attains steady state at time t = 2.1s

Speed is then varied to 160 rad/s after a time t = 7s.

After variation the speed reaches 170 rad/s and then reaches the reference speed 160 rad/s

After changing the reference speed to 160 rad/s the steady state speed is obtained at t = 1.4s

Then the load is applied after a time t = 14s.

Speed after variation dips to 150 rad/s and then settles to reference speed 160 rad/s

After applying the load the steady state speed is obtained at t = 1.1s
B. Performance of Indirect Vector Control IM Using FUZZY Control:

The performance characteristic of a 50 hp, 460 V, 60 Hz IM, operating at various conditions with a FUZZY speed controller was done and the following where the results:

Case 1: AT NO LOAD AND REFERENCE SPEED OF 120 RAD/S
Speed at first reaches directly 120 rad/sec and then settles down at reference speed 120 rad/sec thereafter.
The speed attains steady state at time t = 0.8s

Case 2: AT NO LOAD AND SPEED VARYING FROM 120 RAD/S TO 160 RAD/S
Speed at first reaches directly 120 rad/sec and then settles down at reference speed 120 rad/sec.
The speed attains steady state at time t = 0.8s
Speed is then varied to 160 rad/sec after a time t = 7s.
After variation the speed reaches 160 rad/sec and subsequently settles at reference speed 160 rad/sec
After changing the reference speed to 160 rad/sec the steady state speed is obtained at t = 0.6s

Case 3: LOAD APPLIED AFTER A TIME T(S) AND REFERENCE SPEED OF 120 RAD/S
Speed at first reaches 120 rad/sec and then settles down at reference speed 120 rad/sec.
The speed attains steady state at time t = 1.0s
Then the load is applied after a time t = 14s.
Speed after variation dips to 100 rad/sec and then settles to reference speed 120 rad/sec
After applying the load the steady state speed is obtained at t = 0.4s
Graph 1: Represents Voltage vs Time
Graph 2: Represents Current vs Time
Graph 3: Represents Speed vs Time
Graph 4: Represents Electromagnetic Torque vs Time

Case 4: LOAD APPLIED AFTER A TIME T(S) AND SPEED VARYING FROM 120 RAD/S TO 160 RAD/S

Speed at first reaches 120 rad/sec and then settles down at reference speed 120 rad/sec.
The speed attains steady state at time t = 0.8s
Speed is then varied to 160 rad/s after a time t = 7s.
After variation the speed reaches 160 rad/s and then reaches the reference speed 160 rad/s
After changing the reference speed to 160 rad/s the steady state speed is obtained at t = 0.6s
Then the load is applied after a time t = 14s.
Speed after variation dips to 140 rad/s and then settles to reference speed 160 rad/s
After applying the load the steady state speed is obtained at t = 0.8s

<table>
<thead>
<tr>
<th>CONDITION</th>
<th>STEADY STATE FOR PI</th>
<th>STEADY STATE FOR FUZZY</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT NO LOAD &amp; REFERENCE SPEED</td>
<td>1.2</td>
<td>0.8</td>
</tr>
<tr>
<td>LOAD APPLIED AFTER 14s &amp; AT REFERENCE SPEED</td>
<td>2.1, 1.4</td>
<td>0.8, 0.6</td>
</tr>
<tr>
<td>AT LOAD &amp; SPEED INCREASED FROM 120 TO 160 AFTER 7s</td>
<td>2.1, 1.2</td>
<td>1, 0.4</td>
</tr>
<tr>
<td>LOAD APPLIED AFTER 14s &amp; SPEED INCREASED FROM 120 TO 160 AFTER 7s</td>
<td>2.1, 1.4, 1.1</td>
<td>0.8, 0.5, 0.8</td>
</tr>
</tbody>
</table>

VIII. CONCLUSION AND FUTURE SCOPE

The project was successfully modeled and designed using MATLAB/Simulink and its was simulated at various conditions. The two controllers, namely PI and FUZZY were studied and was compared with each other for speed control of indirect vector control of induction motor drive. At different conditions the controller were simulated and their result and data from the motor drive such as motor current, motor torque and speed was obtained at no load and 200 N-m. The load performance of Fuzzy logic controller was better than that of PI controller. Based on simulation results verification, the following conclusions are made:

- The Fuzzy logic controller is more robust than the PI and when load disturbances occurred.
- The Fuzzy logic controller performance when certain motor parameters (i.e. current and motor torque) were increased by a factor was still quite good and far better than the PI performance when the same parameters.
- The fuzzy logic controller base makes the superior to PI control techniques.

The project can be improved even further by making a hybrid model of the controller. The Fuzzy Logic controller and PI controller combined together will form the hybrid controller. By this the speed can be controlled even faster and the desired state will reach quickly.
REFERENCES


