AC motors are used worldwide in many residential, commercial, industrial and utility applications. AC motors are found in a variety of applications from those that require a single motor to applications requiring several motors. In terms of sheer numbers, the AC induction motor is the most widely used type of electric motor in the modern world. AC motors are primarily used as a source of constant-speed mechanical power but are increasingly being used in variable speed control applications. They are popular because they can provide rotary power with high efficiency, low maintenance, and exceptional reliability - all at relatively low cost. These desirable qualities are the result of two factors: AC motors are usually connected directly to power lines - DC motors require the added expense of a rectifier circuit and most AC motors do not need brushes as DC motors do. Also, the AC motor tends to be more reliable and last longer because there are fewer parts to go wrong and there is no “brush dust” to contaminate the bearings or windings [1-4].

In fact, it can be generalized that prior to the late 1970s, any industrial application that required a motor to have a constant speed would be handled by an AC motor, and any application that required the load to be driven at variable speed would automatically be handled by a DC motor. This statement was true because the speed of a DC motor was easier to change than an AC motor. Since the advent of solid-state components and microprocessor controls, this condition is no longer true. In fact, today a solid-state AC variable-frequency motor drive can vary the speed of an AC motor as easily as that of DC motors and the AC motor requires less maintenance since it does not have brushes.

Few researchers have reported the possibility of applying fuzzy logic controllers for the speed control of AC motor [5-16]. But, it is rare to find that anybody has hardly designed and developed a complete hardware and software for the speed control of AC motor using fuzzy and integrated fuzzy logic controllers. Hence, in the present investigation an attempt is made to indigenously design and implement of fuzzy and integrated fuzzy logic controllers for speed control of AC motor. This chapter explains in detail the hardware and software aspects of design and development of computer based fuzzy and integrated fuzzy logic AC motor speed control system.
6.1 PRINCIPLE

The principle and block diagram of the computer based AC motor speed control system is illustrated in Fig. 6.1. It consists of AC motor, an optical encoder, frequency to voltage (F/V) converter, A/D converter, personal computer (PC), PIT 8254, phase control network, and actuator. The optical encoder senses the speed of the motor and converts it into a train of TTL compatible pulses. Frequency of these pulses is directly proportional to the speed of the motor. This frequency is converted into proportional voltage by F/V converter. Computer acquires voltage through A/D converter available on analog interface card (AIC). This voltage in digital form is converted back to corresponding frequency by the equation \( f = a_1 \cdot v + a_0 \), where \( f \) is the frequency of the signal generated from optical encoder, \( v \) is the measured voltage of F/V converter, \( a_1 \) = slope of frequency v/s voltage graph, & \( a_0 \) = intercept on y-axis. Further this frequency is converted into speed in RPM by the equation:

\[
\text{Speed} = (\text{Frequency} \times 60 \text{ seconds}) \times (1/p) \text{ RPM} \\
= (\text{Frequency} \times 5) \text{ RPM}
\]

where, \( p \) = number of pulses for one revolution. For the optical encoder used, 12 pulses are generated for one complete revolution. The measured motor speed is compared with the set value to obtain error. And this error along with change-in-error is applied to the PID, FLC, and IFLC programs. The controller produces the control action according to the error. The computer then applies this control action, in the form of digital data, to the motor through 8254 programmed as pulse width modulator, available on DIOT, phase control network, and actuator. The ON time of PWM wave varies with digital data. If digital data is more, ON time will be more and vice-versa. Hence, the power applied to motor through actuator will vary with PWM wave. This procedure is repeated till the motor reaches the desire speed. Thus the motor speed is controlled at the desired value. The details of individual blocks of the block diagram are discussed in the following sections.

6.2 HARDWARE FEATURES

The following sections describe the hardware features of the computer based fuzzy and integrated fuzzy logic AC motor speed control system. Fig. 6.5 shows the complete schematic diagram of computer based fuzzy and integrated fuzzy logic AC motor speed control system, which consists of the following elements.
Fig. 6.1. Block diagram of the computer based AC motor speed control system

- Speed sensor
- AC MOTOR
- MT1
- F/V CONVERTER
- OPTO-DIAC
- GATE
- A/D CONVERTER
- PWM CONTROL NETWORK
- ZERO CROSS DETECTOR
- 230V/50Hz
- OUTq
- CLK0

Fig. 6.1. Block diagram of the computer based AC motor speed control system
6.2.1 AC Motor and Speed Sensing Unit

A FHP (universal) AC single-phase motor from TULLU make is used for the present study. The specifications of the AC motor are given in the following Table 6.1.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kilowatts</td>
<td>0.050 KW</td>
</tr>
<tr>
<td>Horsepower</td>
<td>1/15 HP</td>
</tr>
<tr>
<td>Weight (approx.)</td>
<td>1.60 Kg</td>
</tr>
<tr>
<td>Rating</td>
<td>Continuous</td>
</tr>
<tr>
<td>Maximum Speed</td>
<td>13000 RPM, at no load</td>
</tr>
<tr>
<td></td>
<td>4000 RPM, at full load</td>
</tr>
<tr>
<td>Current</td>
<td>0.75 A</td>
</tr>
<tr>
<td>Voltage</td>
<td>230 VAC, 50Hz</td>
</tr>
</tbody>
</table>

The details of the speed-sensing unit are already discussed in chapter 3.2.1. The same mechanism is employed here to sense the speed of the AC motor. An aluminum slotted disk is attached to the AC motor shaft that will produce 12 pulses for one revolution. The photograph of the motor is shown in Fig. 6.2 (a).

6.2.2 Frequency to Voltage (F/V) Converter

The detailed discussion of F/V converter is also mentioned in the chapter 3.2.2. The same technique is used for converting frequency of the pulses from speed sensor to proportional voltage as part of measurement and control of AC motor speed control system. Hence the section 3.2.2 of chapter 3 can be referred for more information. The photograph of the signal conditioning board is shown in Fig. 6.2 (b).

6.2.3 Analog to Digital (A/D) Converter

The details of 16-bit analog interface card designed and fabricated by for the present study are already discussed in chapter 3.2.3. The same card is employed here for the AC motor speed measurement and control. The A/D converter available on AIC acquires the analog voltage from the speed sensor, F/V converter and will provide digital data for the computer and analog control actions to the real process. The section 3.2.3 of chapter 3 can be referred for more details of the A/D converter.
Fig. 6.2 Photographs of the (a) AC motor unit and (b) signal conditioner along with the motor
6.2.4 DIOT Card

The details of digital input/output and timer card designed and fabricated for the present study are already discussed in chapter 3.2.4. The same card is employed here for the AC motor speed measurement and control. DIOT card is used to interface AIC to computer. The 8254 -programmable interval timer/counter, available on DIOT, is employed for generating PWM wave.

6.2.5 Personal Computer

The particulars of the personal computer used for the present work are discussed already in chapter 3.2.5.

6.2.6 PWM Control Network

The PWM control network consists of clock generator, inverter, and a zero crossing detector. This network is interfaced with the 8254 to produce the pulse width modulated signal. Here 8254 is initialized in mode-0 (mono-shot mode) and counter-0 is chosen for the operation. A clock of frequency 2MHz drives clock-0 of 8254. The clock generation is achieved using quartz crystal oscillator. The zero crossing detector circuit is employed to provide triggering signal to the gate-0 of 8254. Zero crossing detector produces a positive spike (+5V amplitude) for every zero crossing of the AC mains. Thus it produces a series of positive spikes of frequency 100Hz. (double to line frequency). The out-0 of 8254 is inverted and given to the actuator (triac BTA06) through opto-coupler MOC3010. The detailed schematic of PWM control network is presented in Fig. 6.5.

The PWM signal is obtained as follows; initially out-0 is high and when gate-0 is triggered, out-0 goes low and remains there till the count becomes zero. Once the count becomes zero, out-0 goes high and remains there until there is a trigger pulse at gate-0 of 8254. For next time it is triggered the count is automatically reloads and repeats the process. The gate is triggered at rate of 10 ms; hence for a given clock of frequency 2MHz., the maximum count that can be decremented with in 10ms is 20,000 for which 100% ON time is obtained. The ON time of out-0 (inverted signal) of 8254 is proportional to the count loaded
in to the counter, if count is more, more the ON time and vice versa. Thus it produces the PWM signal.

6.2.7 Opto coupler

An opto coupler is used to isolate computer from high power AC motor drive circuitry. MOC3010, from Motorola make, is an integrated circuit opto coupler having triac driver circuit. It contains an opto diac at the output stage for driving the triac.

6.2.8 Final Control Element (Actuator)

A triac is used as the final power control element. Triac is the most commonly used device for power control in AC circuits [17]. The final control element is nothing but an actuator, which controls the power or energy, supplied to the system to bring the physical parameter to the desired level. In the present study a triac (BTA06) is used as final control element. A triac can conduct in both directions (it conducts during both the half cycles of AC mains) and is normally used in AC phase control. It can be considered as two SCRs connected in anti-parallel with a common gate connection. Since a triac is a bi-directional device, its terminals cannot be designated as anode and cathode and hence designated them as MT1 and MT2. If terminal MT2 is positive with respect to terminal MT1, applying a positive gate signal between gate G and terminal MT1 can turn ON the triac [18]. It is not necessary to have both polarities of gate signals and a triac can be turned ON with either a positive or negative gate signal. The final power control circuit details are shown in Fig. 6.5. The proportional control of power to the load "phase angle firing control technique" is employed in the present experiment. The PWM control signal (100 Hz.) is applied to the gate of triac, controls the firing angle, which in turn controls power applied to the motor.

6.3 EXPERIMENTAL IMPLEMENTATION

The experimental implementation of PID, fuzzy and integrated fuzzy logic speed controllers for AC motor is accomplished in a computer using 'C' language. The following illustrates the design methodology of PID, fuzzy (9-member bell) and integrated fuzzy logic (bell and triangular) controllers for the proposed system.
The AC motor equipped with speed sensing mechanism is interfaced to the personal computer through EISA compatible DIOT card. The computer acquires the voltage proportional to the speed of the motor through A/D converter available on AIC and substitutes in a well-calibrated equation to evaluate the actual speed of the motor. The frequency versus voltage (acquired by the computer) plot is fitted to a 4th degree polynomial equation using MATLAB [19], and the equation is given by

\[
\text{Frequency} = pl \cdot v^4 + p2 \cdot v^3 + p3 \cdot v^2 + p4 \cdot v + p5 \quad \ldots \quad (6.1)
\]

where, 
- \(pl = 0.38413\)
- \(p2 = 6.588\)
- \(p3 = 35.278\)
- \(p4 = 809.8\)
- \(p5 = 5.3028\) are the coefficients and \(v\) is the voltage acquired by the computer. From the above equation (6.1), the speed in RPM is calculated as follows,

\[
\text{Speed} = (\text{Frequency} \cdot 1/P) \cdot 60 \text{ RPM}
\]

where, \(P\) represents the number of slots (12) on the disk. After evaluation of the speed, the computer determines the error (reference speed - measured speed) and change in error (present error - previous error), and applies to PID, fuzzy and integrated fuzzy control algorithms. The design of PID, fuzzy and integrated fuzzy logic speed controllers is discussed next.

a. Design of PID speed controller

The basics of the PID controller are already discussed in chapter 1.4. The improved PID controller is employed in the present application and its difference equation representing the velocity algorithm is given as,

\[
V_n = V_{n-1} + K_p (e_n - e_{n-1}) + K_i (e_n + e_{n-1})/2T + K_d/6T [(e_n - 2e_{n-1} - 6e_{n-2} + 2e_{n-3} + 2e_{n-4})] \ldots \quad (6.2)
\]

where, \(K_p, K_i\), and \(K_d\) are proportional, integral and derivative constants respectively

- \(V_{n-1}\) is the previous control action
- \(V_n\) is the present control action
- \(e_n\), \(e_{n-1}\) are the present and previous errors respectively
- \(e_{n-2}, e_{n-3},\) and \(e_{n-4}\) are previous to previous errors

In the present application, the best-tuned \(K_p, K_i,\) and \(K_d\) values are found to be equal to 10.0, 0.006, and 0.05 respectively and cycle time \(T\) is equal to 1.
b. Design of fuzzy logic speed controller

The fundamentals of fuzzy logic controller are already discussed in chapter 1.5; only the implementation specifications (design considerations) of 9-member bell and triangular shaped fuzzy logic speed controllers for AC motor control application are discussed here.

The two inputs of the fuzzy logic controller, the error ‘e(k)’ and change in error ‘ce(k)’ defined as,

\[ e(k) = \frac{(\text{reference speed } r(k) - \text{measured speed } u(k))}{7500.0} \]  
\[ ce(k) = \text{present error } e(k) - \text{previous error } e(k-1) \]

and the output of fuzzy logic controller is defined as \( cu(k) [20-22] \). The two inputs and output of fuzzy logic controller are mapped to both triangular and bell-shaped fuzzy membership functions on the given universe of discourses for error, change in error and output as shown in Fig.6.3 and Fig. 6.4.

![Fig. 6.3 Nine member bell-shaped membership functions for error, change in error and output](image-url)

![Fig. 6.4 Nine member triangular membership functions for error, change in error and output](image-url)
The error, change in error and output variables are scaled using appropriate scaling factors and are defined on the same universe of discourse [23]. The above-mentioned are well-tuned values for the respective controller. The scaled input and output data are then represented using nine linguistic variables such as NL-Negative Large, NM-Negative Medium, NS-Negative Small, NZ-Negative Zero, ZE-Zero Error, PZ-Positive Zero, PS-Positive Small, PM-Positive Medium, and PL-Positive Large and these can be viewed as labels of fuzzy sets. The input and output variables are mapped between -7500 and +7500 and the same are normalized to -1.0 and +1.0.

For each value of e(k), ce(k) and cu(k), the degree of membership is evaluated for all the membership functions defined in Fig. 6.3 and Fig. 6.4. The mathematical representation of bell and triangular membership functions are already presented as an example in chapter 3.3. The inference process of the FLC relates the input variables e(k) and ce(k) to fuzzy output control action cu(k) in terms of membership functions (via a set of linguistic rules). The control statements are represented as a set of IF...THEN rules. Table 6.2 shows rule base editor containing 81 linguistic control rules is used in the present application. As an example, the following is the possible control rule of the fuzzy speed controller,

\[
\text{IF } e(k) \text{ is the PS and } ce(k) \text{ is PZ THEN } cu(k) \text{ is PS}
\]

Table 6.2 Rule base with 81 control rules

<table>
<thead>
<tr>
<th>change in error 'ce'</th>
<th>NL</th>
<th>NM</th>
<th>NS</th>
<th>NZ</th>
<th>ZE</th>
<th>PZ</th>
<th>PS</th>
<th>PM</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>NL</td>
<td>NL</td>
<td>NL</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
</tr>
<tr>
<td>NM</td>
<td>NL</td>
<td>NL</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>NZ</td>
<td>ZE</td>
<td>PZ</td>
</tr>
<tr>
<td>NS</td>
<td>NL</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>NZ</td>
<td>NZ</td>
<td>ZE</td>
<td>PZ</td>
<td>PS</td>
</tr>
<tr>
<td>NZ</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>NZ</td>
<td>ZE</td>
<td>ZE</td>
<td>PZ</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>ZE</td>
<td>NM</td>
<td>NS</td>
<td>NZ</td>
<td>ZE</td>
<td>ZE</td>
<td>PZ</td>
<td>PS</td>
<td>PM</td>
<td>PL</td>
</tr>
<tr>
<td>PZ</td>
<td>NM</td>
<td>NZ</td>
<td>ZE</td>
<td>ZE</td>
<td>PZ</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PL</td>
</tr>
<tr>
<td>PS</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PZ</td>
<td>PZ</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PL</td>
</tr>
<tr>
<td>PM</td>
<td>NZ</td>
<td>ZE</td>
<td>PZ</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PL</td>
<td>PL</td>
</tr>
<tr>
<td>PL</td>
<td>ZE</td>
<td>PZ</td>
<td>PZ</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PL</td>
<td>PL</td>
<td>PL</td>
</tr>
</tbody>
</table>

Mamdani’s fuzzy inference strategy is used in the present application and is represented as,
where, $\alpha_i$ is the measure of the contribution of the $i^{th}$ rule to the fuzzy control action and $\mu_{ui}$ is the grade of membership of $i^{th}$ possible control element. The weighing factor $\alpha_i$ is usually expressed as [24],

$$\alpha_i = \bigcup_{i=1}^{9} \mu_{ei} \cap \mu_{cei} \quad \text{for triangular shaped member function (min. and max. operation)}$$

$$\alpha_i = \mu_{ei} \cap \mu_{cei} \quad \text{for bell-shaped membership function (only min. operation)}$$

where, $\mu_{ei}$ and $\mu_{cei}$ are the grade of memberships of $i^{th}$ element in the error and change in error fuzzy sets. In the present application the output of fuzzy inference engine (a fuzzy set with nine elements) is defuzzified using the center of gravity method and the expression is given as [25,26],

$$c_u = \frac{\sum_{i=1}^{9} \mu_{cu}(w_i)w_i}{\sum_{i=1}^{9} \mu_{cu}(w_i)}$$

where, $w_i$ is the support member value for the $i^{th}$ element, and $\mu_{cu}(w_i)$ is the value of grade of membership function for $i^{th}$ element.

Velocity fuzzy control algorithm has been employed in the present study and hence the control action applied to the process is given as,

$$u = cu(k) + cu(k-1)$$

where, $u$ is the final control value, $cu(k)$ is present fuzzy computed control action and $cu(k-1)$ is the previous control action.

c. Design of integrated fuzzy logic speed controller

The integrated fuzzy logic controller (IFLC) is composed of PID controller and fuzzy logic controller [27,28]. The fundamentals of IFLC already discussed in chapter 1.6. The design specifications of IFLC for AC motor speed control application are given as, the PID parameters $K_p = 10.8$, $K_i = 0.005$, $K_d = 0.059$, and $T = 1.0$ for bell-shaped IFLC, and $K_p = 9.9$, $K_i = 0.003$, $K_d = 0.07$, and $T = 1.0$ for triangular IFLC and the FLC specifications are given as same as discussed in the previous section.

The control action of above controllers’ as an output from the computer is a digital value, which is loaded (count) in to counter-0 of 8254, which is programmed to produce pulse width modulated signal. The ON time of PWM signal is proportional to the count.
The PWM signal is applied to the actuator (triac) through optocoupler (opto-diac). The actuator decides the amount of power added to or removed from the motor in order to maintain desired speed. The complete circuit schematic of AC motor speed control system is presented in Fig. 6.5. The photographs of the complete experimental setup and working model of the computer based fuzzy and integrated fuzzy logic AC motor speed control system is shown in Fig. 6.6.

6.4 SOFTWARE FEATURES

The software for computer based AC motor speed control system is discussed in this section. The complete software of the system is developed in C language. Three controller programs such as PID, fuzzy and integrated fuzzy logic controllers are discussed through the necessary flowcharts.

Beginning of the program displays a text menu, which will enable the user to monitor the current value of the parameter (speed) being measured and controlled, and the controller parameters; to enter the new set point; to enter filename to store measured data; and tune the controller parameters etc. The software begins with initialization of respective controller's parameters such as $K_p$, $K_i$, $K_d$ in case of PID controller, membership boundaries and linguistic control values in case of FLC. The system hardware (DIOT and AIC) is initialized next. The program switches off power to the actuator initially and displays the initial speed of the motor, and controller parameters. It then prompts the user to enter the set point or desired speed command. Program calls A/D converter subroutine to measure the voltage and calculates actual speed in RPM. Accepts the desired command from user, finds the error, change-in-error and apply them to respective control algorithm selected by the user to generate control action. It scales the controller output to 16-bit data and sends to counter-0 of 8254. Before sending the control output to 8254 it checks the control output for whether it lies beyond the 16-bit counter range. If it is beyond the range it sends the same control action that it has sent in previous loop else, it sends the current control action. Then the program scans the keyboard for the user to press the key to change the set point or other function that is assigned to that particular key. If no key is pressed it does the normal operation and stores the measured data into an array. Finally it updates the controller parameters and jumps to continuous loop. It repeats the same procedure mentioned above.
Fig. 6.5 Circuit schematic of the computer based AC motor speed control system.
Fig. 6.6 Photographs of the experimental setup and working model of AC motor speed control system (a) top view and (b) front view
6.4.1 Flowchart of the PID Program

The complete flowchart of computer based PID AC motor speed control system is shown in Fig.6.7 and 6.8. The flowchart will provide sufficient information about the program implementation. The general details of program functioning are already explained in above section.

6.4.2 Flowchart of the FLC Program

The complete flowchart of computer based fuzzy logic AC motor speed control system is shown in Fig.6.9 and 6.10. The flowchart will provide adequate information about the program implementation. The general details of program functioning are already explained in above section.

6.4.3 Flowchart of the IFLC Program

The complete flowchart of computer based integrated fuzzy logic AC motor speed control system is shown in Fig.6.11. The flowchart will give sufficient information about the program implementation. The general details of the program functioning are already explained in above section.

6.5 RESULTS AND DISCUSSION

The computer based PID, fuzzy, and integrated fuzzy logic speed controllers have been designed and fabricated for AC motor control system. The AC motor is controlled and tested by the proposed controllers such as improved PID, 9-member bell-shaped FLC, bell-shaped IFLC and triangular-shaped IFLC. The efficiency of these controllers for speed control of AC motor is evaluated by applying several tests over a wide range of operating conditions. The performance indexes (in terms settling time, steady state error, overshoot, and undershoot) of the proposed controllers are studied for the following conditions,

- In absence of load (no load)
- In presence of load (on load) and
- In presence of noise (external disturbance)
Fig. 6.7 Flowchart of the computer based PID AC Motor speed control system

Start

Declaracion of variables and functions and initialization of $V_{n-1}, e_{n-1}, e_{n-2}, e_{n-3}$, and $e_{n}=0.0, K_p=70.5, K_i=0.88, K_d=1.01$, and $t=1.0$

Initialization of system hardware (8255-I pA, pB & pCh as i/p and pCl as o/p) (8254 in mode-1 and counter-0 operation)

Stop the AC motor & display the main menu on monitor (Send 0000 to counter-0 of 8254) (Display initial speed, initial set point speed, and PID controller parameters etc.)

Prompt the user to enter set-point speed

Read the current speed through AIC (Call A/D conversion program) Frequency=$(0.38413)(pow(x,2.4))-(6.588(pow(x,2.3))+(35.278(pow(x,2.2)))+(809.8*x)+5.3028)$ Speed in RPM = (Frequency*60)/12

Compute the error ($e_n$= set-point speed - measured speed)

PID Computation ($V_n$)

If $V_n$>20000, send 20000. Else if $V_n<0$ send 0 to counter-0 of 8254

Apply control action to the actuator through 8254 (Call 8254 PWM generation program) $h_{byte}=V_n/256$ and $l_{byte}=V_n\%256$

Display the menu on the monitor (Current speed, set point speed, file name, PID parameters, etc.)

Scan keyboard for the following functions If any key is pressed & found as ('s' to enter new set-point, 't' to tune the PID, 'f' to enter file name to write data, and 'q' to quit from the program)

Sample and store the time and speed into an array

Update input parameters to PID expression $V_{n-1}=V_n$ $e_{n-4}=e_{n-3}$ $e_{n-3}=e_{n-2}$ $e_{n-2}=e_{n-1}$ $e_{n-1}=e_n$

YES

NO

Fig. 6.7 Flowchart of the computer based PID AC Motor speed control system
Fig. 6.8 (a) A/D conversion (b) PID computation and (c) 8254 PWM generation routines
Fig. 6.9 Flowchart of the computer based fuzzy logic AC motor speed control system

1. Start

2. Declaration of variables and functions and initialization of fuzzy controller variables and membership boundaries

3. Initialization of the system hardware (DIOT and AIC)

4. Stop the AC motor & display the main menu on monitor
   (Send 0x0000 to counter-0 of 8254)
   (Display initial speed, initial set point speed, fuzzy controller parameters etc.)

5. Prompt the user to enter the set-point speed

6. Measure the current speed through Analog Interface Card (AIC)
   (Call A/D Conversion routine)
   Frequency = (0.38413)*pow(v,2.4) - (6.588* pow(v,2.3)) + (5.278* pow(v,2.2)) + (0.809*v) + (5.3026)
   Speed in RPM = (Frequency* 60)/12

7. Compute the error ‘E’ and change in error ‘CE’
   \[ E = \text{(set-point - measured value)/set-point} \]
   \[ CE = \text{present error - previous error} \]

8. Fuzzy computation

9. Scale the fuzzy computed control action (CU) to 16-bit count for c0
   (Control action ‘CU’ = cu_1 + cu)
   (Count = cu*2.0)

10. Is
    0<count<20000

11. YES

12. Send the same previous control action

13. NO

14. Send scaled fuzzy control action (count) to the actuator through 8254
   (Call 8254 PWM routine)
   h_byte = count/256 & l_byte = count%256

15. Display menu on the monitor
   (Current speed, new set point speed, fuzzy parameters etc.)

16. Scan keyboard for a key press
   (To enter change set-point speed, fuzzy controller tuning, write sampled data to a file, quit the program, etc.)

17. If no key is pressed, do the normal operation

18. Sample the speed and time into an array

19. Update fuzzy variables (en1 and cu_1)
Fig. 6.10 Flowchart for fuzzy computation

1. **Fuzzification**
   - Process the error 'E' and change in error 'CE' and define on the universe of discourse -1.0 to +1.0

2. **Redefine the E and CE on the universe of discourse -4.0 to +4.0**

3. **Compute fuzzy sets for error and change in error using bell-shaped membership function**

4. **Compute fuzzy set for picked (predicted) control action from the rule-base**

5. **Fuzzy inference engine rule evaluations**
   - Perform minimum operation on error and change in error fuzzy sets $(a_i = c_i \cap \tilde{c}_i)$

6. **Perform minimum and maximum operations on 'a' and picked control action fuzzy sets**
   - $\mu_{cu} = \bigcup_{i=1}^{n} \mu_{c_i \cap a_i}$

7. **Defuzzification**
   - Convert fuzzy control action into a crisp control action (defuzzification) using centre of gravity method
   - $CU = \sum_{j=1}^{n} (\mu_{cu_j} \cdot c_j) / \sum_{j=1}^{n} \mu_{cu_j}$

$CU$
Fig. 6.11 Flowchart of the computer based integrated fuzzy logic AC motor speed control system

Start

Declaration of variables, and functions and initialization of fuzzy controller variables, and members

Initialization of system hardware (DIOT and AIC)

Stop the AC motor & read the current speed through AIC and display it on the monitor
Speed in RPM = [(0.38413)*(pow(v2,4)) - (6.588*(pow(v2,3)))+(35.278*(pow(v2,2))]
+[(809.8*v2)+(5.3028)]*5
Prompt the user to enter set point speed

Measure the current speed, compute the error ‘E’ and change in error ‘CE’

Fuzzy computation

Compute set-point for PID controller
(set-point = fuzzy output + actual set point)

PID computation

Scale the PID computed value and send to c0 of 8254 to produce PWM signal to the actuator

Display the menu on the monitor
(current speed, new set-point speed, integrated fuzzy, & PID parameters etc.)
And provide options to tune IFLC, change the set-point, quit program, etc.

Fuzzy computation

E

CE

Fuzzification
(compute fuzzy sets for ‘E’ and ‘CE’ using bell/triangular shaped membership function)

Fuzzy inference engine rule evaluations
Process the error and change in error and pick the appropriate control action
(compute fuzzy set for control value)
Do min. operation on ‘e’ and ‘ce’
\[ w_k = c_1 \cap c_2 \]
Do min-max operation on ‘u’ picked and control action
\[ cu = \bigcup_{i=1}^{n} \mu_{ci} \cap \alpha_i \]

Defuzzification
(compute crisp control action by COG method)
\[ CU = \Sigma_{i=1}^{n} (\mu_{ci} \cdot c_i) / \mu_{cu} \]

CU

PID computation

Inputs to PID expression
\( V_{n-1}, e_n, e_{n-1}, e_{n-2}, e_{n-3}, \) and \( e_{n-4} \)

Compute PID control action
\[ \frac{V_n}{2T} + K_p (e_n - e_{n-1}) + K_i (e_n + e_{n-1}) \]
\[ 1/2T + K_d (V_n / 2T) + V_{n-1} / 2T + 2V_{n-2} / 2T + 2V_{n-3} / 2T + 2V_{n-4} / 2T \]

Update inputs to PID expression
\[ V_{n-1} = V_n \]
\[ e_{n-4} = e_{n-3} \]
\[ e_{n-3} = e_{n-2} \]
\[ e_{n-2} = e_{n-1} \]
\[ e_{n-1} = e_n \]

U
a. In absence of load

The various studies including step input, set point variation, and step variation responses of PID controller are shown in Fig. 6.12. The above-mentioned studies for bell-shaped nine member FLC are shown in Fig. 6.13. And similarly, the studies of bell and triangular shaped IFLCs are presented in Fig. 6.14 and Fig. 6.15 respectively. The step response of the AC motor was investigated by applying a step input corresponding to a speed reference of 7500 RPM. Fig. 6.12(a), Fig. 6.13(a), Fig. 6.14(a), and Fig. 6.15(a) show the starting of the motor from standstill to the reference speed at no load. The plot of comparison of step responses of PID, FLC, and IFLC for desired speed of 7500 RPM is shown in Fig. 6.16. From the graph it is observed that the IFLC has the best transient and steady state response.

The performance indexes of PID, fuzzy, and integrated fuzzy controllers are mentioned in Table 6.3. It summarizes the results obtained from the tests performed. The experimental results obviously shows that IFLC exhibits a very fast response and no overshoot, no under shoot, with negligible steady state error. Also, IFLC has very small settling time (4.3336 sec) when compare to other controllers.

b. In presence of load

The performance of above proposed controllers is also studied in the presence of load. The motor load is generated by applying a magnetic brake on the rotational disk attached to the motor shaft. The magnetic brake works by means of an aluminum disk which when rotated between the poles of magnet, eddy currents form on the disk producing the effect of a frictional load. When the motor is running at a rated speed of 7500 RPM, the load is applied. The step input responses of PID, bell shaped fuzzy, bell shaped IFLC and triangular IFLC under load are graphically represented in Fig. 6.17. The experimental results summarized in Table 6.4 reveal maximum undershoot/overshoot when load is applied/removed. As it can be seen in the Table 6.4 the bell shaped IFLC has the best performance in presence of load. It exhibits a very fast recover time when load is applied (4.000 sec for an undershoot of 248 RPM) and when load is removed (2.2324 sec for an overshoot of 295 RPM).

c. In presence of noise

The system performance for a random noise of varying amplitudes is studied. The term noise is used to designate unwanted signals that tend to disturb the system and over which we have incomplete control. The sources of noise may be external or internal to the system. It is essential to study the performance of proposed controllers for speed control of AC motor in
the presence of random noise. Hence, an external random noise of varying magnitudes (±100 mV) has been deliberately introduced in the system for studying the effect of noise on controllers. A C language program is written for generating the random noise with varied amplitude. The responses of proposed controllers, when an external random noise is applied, are shown in Fig. 6.18. The graphs reveal that the effect of noise is near the set point of 7500 RPM. Though small overshoots and undershoots are observed at the set point, but these variations are negligible for IFLC and it has least steady state error (3.91 for bell and 3.31 for triangular) under noise of ±100 mV as compared to PID and FLC.

The present results are compared with the work of El-Saady et al. proposed a novel fuzzy logic controller for high performance induction motor drive systems [29]. Both digital simulation and laboratory test results for speed control of the induction motor drive are presented. Also a comparison between fuzzy logic controller and the traditional PI controller are presented. They studied the system performance both in presence and absence of load. The results validate the robustness and the effectiveness of the proposed fuzzy logic controller for high performance of induction motor drive. They have not presented exact numerical values to compare with the present values. In the present application also, the FLC exhibits better control (transient and steady state responses) over improved, best-tuned PID controller. The numerical results also presented for the detailed discussion.

The present results also compared with the results of Mokrani L. et al. designed and illustrated a fuzzy self-tuning PI controller for speed control of induction motor drive [30]. Their simulation results show that the proposed fuzzy self tuning PI controller is better than the fixed gains one in terms of robustness and speed rise time, even under great variations of operating conditions and load disturbance. They have not presented the exact numerical results of the system to extend comparison with the present results. The present results show that the FLC structure is robust and flexible. The experimental study has been carried out under no-load, load and noise conditions. The numerical results show the FLC exhibited better control in all the conditions. There are no reports found on the application of integrated fuzzy logic controllers for AC motor speed control. From the present study it is concluded that integrated fuzzy logic structure is effective, robust and more flexible for AC motor speed control application.
Fig. 6.12 PID based AC motor speed control system responses under no-load for (a) step input (b) set-point variation and (c) step variation
Fig. 6.13 FLC (9-member bell) based AC motor speed control system responses under no-load for (a) step input (b) set-point variation and (c) step variation
Fig. 6.14 IFLC (bell) based AC motor speed control system responses under no-load for (a) step input (b) step variation and (c) set-point variation
AC MOTOR SPEED CONTROL USING INTEGRATED FLC (triangular) response for step input (7500 RPM)

DC MOTOR SPEED CONTROL USING INTEGRATED FLC (triangular) response for step variation

DC MOTOR SPEED CONTROL USING INTEGRATED FLC (triangular) response for set-point variation

Fig. 6.15 IFLC (triangular) based AC motor speed control system responses under no-load for (a) step input (b) step variation and (c) set-point variation
Fig. 6.16 Comparison of step (7500 RPM) responses of PID, FLC (bell), and IFLC (bell) for AC motor speed control under no-load.
Table 6.3 Experimental results of the computer based AC motor speed control system in absence of load

<table>
<thead>
<tr>
<th>Controller</th>
<th>Sampling Interval (Seconds)</th>
<th>Maximum</th>
<th>Settling time (Seconds)</th>
<th>Steady-state Error (RPM)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>0.01500</td>
<td>5.15</td>
<td>3.58</td>
<td>6.6306</td>
<td>1.83</td>
</tr>
<tr>
<td>9-member FLC (Bell function)</td>
<td>0.02018</td>
<td>26.23</td>
<td>9.12</td>
<td>4.9448</td>
<td>1.80</td>
</tr>
<tr>
<td>IFLC (Bell function)</td>
<td>0.02209</td>
<td>3.96</td>
<td>2.81</td>
<td>4.3336</td>
<td>1.52</td>
</tr>
<tr>
<td>IFLC (Triangular function)</td>
<td>0.02381</td>
<td>4.05</td>
<td>3.15</td>
<td>4.4587</td>
<td>1.56</td>
</tr>
</tbody>
</table>

AC MOTOR SPEED CONTROL SYSTEM RESULTS IN ABSENCE OF LOAD
For a step size of 7500 RPM (0-7500RPM)

Table 6.4 Experimental results of the computer based AC motor speed control system in presence of load

<table>
<thead>
<tr>
<th>Controller</th>
<th>Load applied</th>
<th>Maximum</th>
<th>Recover time (Seconds)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-member FLC (Bell function)</td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IFLC (Bell function)</td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IFLC (Triangular function)</td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AC MOTOR SPEED CONTROL SYSTEM RESULTS IN PRESENCE OF LOAD
Running at 7500 RPM
Fig. 6.17 Step input responses of (a) PID (b) FLC (bell) (c) bell IFLC and (d) triangular IFLC for AC motor speed control system under load.
Fig. 6.18 Step input responses of (a) PID (b) FLC (bell) (c) bell IFLC and (d) triangular IFLC for AC motor speed control system under noise ± 100mV
Table 6.5 Experimental results of the computer based AC motor speed control system in presence of noise

<table>
<thead>
<tr>
<th>Controller</th>
<th>Noise level (in mV)</th>
<th>Sampling Interval (Seconds)</th>
<th>Maximum Overshoot (RPM)</th>
<th>Maximum Undershoot (RPM)</th>
<th>Settling time (Seconds)</th>
<th>Steady State Error (RPM)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>±100</td>
<td>0.01612</td>
<td>20.14</td>
<td>9.3</td>
<td>6.9981</td>
<td>4.71</td>
<td></td>
</tr>
<tr>
<td>9-member FLC (Bell function)</td>
<td>±100</td>
<td>0.02495</td>
<td>44.77</td>
<td>8.13</td>
<td>5.0652</td>
<td>4.73</td>
<td></td>
</tr>
<tr>
<td>IFLC (Bell function)</td>
<td>±100</td>
<td>0.02874</td>
<td>12.42</td>
<td>15.3</td>
<td>4.3464</td>
<td>3.31</td>
<td>Best</td>
</tr>
<tr>
<td>IFLC (Triangular function)</td>
<td>±100</td>
<td>0.02998</td>
<td>11.91</td>
<td>15.2</td>
<td>4.5839</td>
<td>3.91</td>
<td>Better</td>
</tr>
</tbody>
</table>
REFERENCES


