Design and Real Time Implementation of Integrated Fuzzy Logic Controller for a High Speed PMDC Motor

S.S. Patil and P. Bhaskar*

Department of Instrumentation Technology, Gulbarga University P. G. Centre, Raichur - 584 133, KA, INDIA
*Corresponding author Email: p_bhaskar68@yahoo.com

Abstract

This paper presents the design and implementation of an integrated fuzzy logic controller (IFLC) for a high-speed permanent magnet DC (PMDC) motor speed control system. The proposed strategy is intended to improve the performance of the conventional controller by use of IFLC. The system is studied for step input with and without load (magnetic) conditions and for various standard input test commands such as square, triangular, sinusoidal, and ramp. The experimental implementation demonstrates that IFLC is particularly effective in speed control of PMDC motor under the above-mentioned conditions. The graphical results of the proposed controller are presented and are compared with conventional controllers.

Keywords: Integrated fuzzy logic controller, high speed control, PMDC motor, analog interface card.

Introduction

The PMDC motor is one of the most widely used prime movers in industry today. PMDC motors used in many applications such as steel rolling mills, electric tracking systems, textile mills -including weaving and spinning, robotic manipulators, defense etc., require precise speed controllers to perform these tasks. The major problems in applying the conventional control algorithms in a speed controller are the effects of non-linearity in a DC motor. The nonlinear characteristics of a DC motor such as saturation and friction could degrade the performance of conventional controllers [1]. Many advanced model-based control methods such as variable structure control and model reference adaptive control have been developed to reduce these effects.
However, the performance of these methods depends on the accuracy of system models and parameters. Generally, an accurate nonlinear model of an actual DC motor is difficult to find, and parameter values obtained from system identification may be only approximated values.

Emerging intelligent techniques have been developed and extensively used to improve or to replace conventional control techniques, because these techniques do not require a precise model. One of the intelligent techniques, fuzzy logic developed by Lotfi A. Zadeh [2], is applied for controller design in many applications. A fuzzy logic controller (FLC) was proved analytically to be equivalent to a nonlinear PI controller, when a nonlinear defuzzification method is used [3].

There are many reports [4-12] on implementation of FLC and IFLC on PC, DSP, and microcontroller for DC motor speed control. These platforms have their own advantages and disadvantages. In the proposed work we have implemented integrated fuzzy logic controller, which is a combination of both fuzzy logic and PID controllers (PIDC) [13], on a PC with a new high-resolution, high-speed 16-bit analog interface card (AIC) designed for parallel port of PC. This approach is novice and makes the hardware portable. The performance of the proposed controller is studied at high speed of 5000 rpm and for various linear and nonlinear input test signals. The linear input commands include triangular and sinusoidal, the non-linear input includes step and square, and semi-linear input i.e., ramp.

In this paper, the real time implementation of IFLC for high-speed PMDC motor is discussed. Heuristic knowledge is applied to define fuzzy membership functions and rules. The membership functions and rules are modified after initially barrowing the knowledge from PID controller developed from simple linear model. The process of fuzzification is done based on min-max (MOM) method. The defuzzification is done using centre of gravity (COG) method.

**Integrated Fuzzy Logic Controller**

The basic IFLC configuration is shown in Fig. 1, where FLC is used in a supplementary role to enhance the conventional PIDC. When the control conditions change, FLCs are easy to realize and the system behavior can be easily redesigned by modifying the fuzzy logic rules. One does not have to redesign the existing control system hardware in order to acquire satisfactory response during the change of load conditions and appearance of disturbances [14].

![Figure 1: The block diagram of integrated fuzzy logic controller.](image-url)
Design of PIDC
The block diagram of PIDC is shown in Fig. 2. The PID controller algorithm is implemented with the following well-known PID difference equation given by:

\[ V_n = K_p (e_n - e_{n-1}) + K_i e_n T + K_d/T [(e_n - 2e_{n-1} + e_{n-2})] \]  

(1)

where, \( V_n \) is the control action, 
\( e_n, e_{n-1}, \) and \( e_{n-2} \) are the present, previous, and previous to previous errors respectively,
and \( K_p, K_i, \) and \( K_d \) are proportional, integral, and derivative constants respectively and \( T \) is the cycle time.
In the present application, the best-tuned \( K_p, K_i, \) and \( K_d \) values are found to be equal to 150.0, 352.0, and 11.0 respectively and \( T \) is equal to 10ms.

Figure 2: The block diagram of PID controller.

Design of FLC
Fuzzy logic uses membership functions to define the degree to which crisp physical values belong to terms in a linguistic variable set [15-18]. The block diagram of a basic FLC is shown in Fig. 3. The FL controller consists of mainly three basic components namely the fuzzification interface, fuzzy inference engine (rule base) and defuzzification stage. The fuzzification stage converts real number input values into fuzzy values, the fuzzy inference engine processes the input data and computes the control outputs using IF and THEN rules. These outputs, which are fuzzy values, are converted into real numbers in the defuzzification stage.

Figure 3: Block diagram of fuzzy logic controller.
The two-input-one-output FLC is designed for the present application. The inputs to the FLC are error \( e(k) = (\text{set-point speed} - \text{measured speed})/\text{set-point speed} \), and change-in-error \( ce(k) = (\text{present error} - \text{previous error}) \) [19]. These two inputs are defined on a universe of discourse with the nine membership functions (NL, NM, NS, NZ, ZE, PZ, PS, PM, and PL). The output of the FLC is ‘CU’ is given as input to the PMDC motor. The inputs and controlled output of the FLC are described by,

\[
E = e(k) = r(k) - y(k) \quad (2)
\]

\[
CE = ce(k) = e(k) - e(k-1) \quad (3)
\]

\[
CU = cu(k) \quad (4)
\]

The triangular membership function is used to fuzzify the error and change-in-error. The error and change-in-errors are mapped between -1.0 and +1.0 on the universe of discourse. The membership boundaries for error, change-in-error and control output are shown in Fig. 4. The fuzzy inference engine is the heart of the FLC comprises both the knowledge base and decision-making logic. The knowledge base consists of a data base with necessary linguistic variables (rule set) and decision-making logic used to decide what control action to be taken. The inference process of the FLC relates the fuzzy state variables \( e(k) \) and \( ce(k) \) to the fuzzy controlled action \( cu(k) \) with the help of linguistic rules. The decision-making logic uses IF and THEN rules to pick up appropriate control action for the process. As an example, the following is a possible control rule for a FLC

IF \( e(k) \) is ‘PM’ and \( ce(k) \) is ‘NS’, THEN \( cu(k) \) is ‘PS’

A control rule can be regarded as an implication, \( E_i, CE_i, \rightarrow U_i \)

The implementation of the inference mechanism in the present study is using Mamdani’s minimum operation \( R_c \), which is given as

\[
R_c : U_i=1, n \alpha_i \cap \mu_i \quad (5)
\]

Where \( \alpha_i \), the weighing factor, is the measure of the contribution of the \( i^{th} \) rule to the fuzzy control action and is expressed as

\[
\alpha_i = \mu_{Ci} \cap \mu_{CEi} \quad (6)
\]

The output of the fuzzy inference engine is a fuzzy set on the output universe of discourse, so this needs to convert into non-fuzzy (crisp value).

The centre of gravity (COG) method is used for defuzzification. The defuzzified output for the process is calculated from the equation

\[
CU = \frac{\sum_{i=1, n} \mu_U (w_i) \cdot w_i}{\sum_{i=1, n} \mu_U (w_i)} \quad (7)
\]

Where, \( n \) is the number of elements in control output fuzzy set.
\( w_i \) is the support member value for the \( i^{th} \) element
\( \mu_U (w_i) \) is the value of grade of membership function for \( i^{th} \) element
Figure 4. Triangular membership functions of error, change-in error, and control output.
Fuzzification stage
The fuzzification stage converts a crisp number (e(k) and ce(k)) into the fuzzy values within a universe of discourse U. The U is quantified and normalized to [-1, +1] by scaling factors Ge and Gce. We use the triangle-shaped membership function with seven terms as shown in Fig. 4. Unlike the traditional digital logic, the fuzzy logic extends the ranges in degree of truth from 0 to 100 percent.

Decision logic stage
Basically, the decision logic stage is similar to a rule base, consisting of the fuzzy control rules, which decide how the FLC works. This stage is the core of the FLC and is constructed from expert knowledge and experience. With specific reference to the characteristics of the PMDC motor, we construct the decision-making logic based on Min-Max operations.

Defuzzification stage
Defuzzification is a mapping from a space of fuzzy control actions defined over an output universe of discourse into a space of non-fuzzy (crisp) control actions.

Experimental PMDC motor control system
The block diagram of experimental PMDC motor control system is illustrated in Fig. 5. We have used PMDC motor unit from LUNAR Motors Pvt. Ltd., India. The motor details are presented in Table I. The system consists of PMDC motor, speed sensor, analog interface card, personal computer, and driver. An indigenous 16-bit analog interface card (AIC), designed by the authors [20], is used as the interface for implementation of digital controller. The card is designed for multi purposes i.e., it can be used both for measurement and control. The card mainly consists of a high-speed four channel 16-bit serial A/D converter –AD974 from Analog Devices [21], and a 16-bit serial D/A converter –MAX542 from Maxim [22]. The D/A converter is operated in bipolar mode in order to drive the motor in both clockwise and anti-clockwise directions. The control output from PC is a digital data, which is converted into equivalent analog voltage by D/A converter. The D/A converter is provided with onboard four channels using analog multiplexer –CD4051 to control multiple parameters. The card is interfaced to PC through parallel port. The AIC is shown in Fig. 6. The photograph of AIC is shown in Fig. 7. In the present study a PC -Intel Pentium-IV processor with 1.7GHz clock frequency, 128MB RAM, 40GB HDD, one parallel port, and two serial ports is employed for controlling the speed of a PMDC motor. The PC receives input signals and transmits output control signals through the AIC. The proposed real-time control algorithms such as PID, FL, and IFL controllers and other control schemes are implemented on this platform.

The hardware configuration of the experimental system is shown in Fig. 8. The driver circuit is built with the power transistors to provide enough current to drive the motor. The control signal from PC does not directly drive the motor. A pre-amplifier increases the strength of output analog signal to the motor voltage level. It is followed by a driver circuit consisting of an op-amp (LM356), and Darlington pair (CL100 and
Design and Real Time Implementation of Integrated Fuzzy

2N3055) in closed loop. The amplifier included in the closed loop will provide the compensation for voltage drop across the Darlington pair.

The load is applied in the form of magnetic brake. The magnetic brake works by means of an aluminum disc, which is mounted on the motor shaft. The aluminum disc serves two purposes, one as a speed sensor and other as a speed brake. When disc is rotated between the poles of a magnet, eddy currents form on the disc producing the effect of frictional load, which in turn retards speed of the motor.

The optical encoder is used as a speed sensor. It converts the speed of the motor into corresponding frequency with the help of the disc attached to the shaft. The optical encoder used in this application produces 12 TTL compatible pulses for one revolution. The frequency of these pulses is further converted into proportional voltage by a F/V converter, which is constructed using LM2907. Hence, the output voltage of F/V converter is directly proportional to the speed of the DC motor. This proportional voltage is acquired by PC through AIC. The picture of the complete system is shown in Fig. 9.

![Figure 5: Block diagram of IFLC based PMDC motor speed control system.](image)

![Figure 6: Block diagram of analog interface card.](image)
Figure 7: Photographs of analog interface card.

Figure 8: Circuit schematic of PC based PMDC motor speed control system.
Figure 9: Photographs of experimental setup.

Table I: Specifications of PMDC Motor.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>35.7mm outer diameter, &amp; 57.0mm length with 2.3mm shaft diameter</td>
</tr>
<tr>
<td>Normal voltage</td>
<td>12 VDC</td>
</tr>
<tr>
<td>Power</td>
<td>46.7 Watts</td>
</tr>
<tr>
<td>No load speed (Max)</td>
<td>10500 rpm</td>
</tr>
<tr>
<td>No load current (Max)</td>
<td>0.270 Amp</td>
</tr>
<tr>
<td>Commutation</td>
<td>Carbon Brush</td>
</tr>
<tr>
<td>Housing material</td>
<td>Steel</td>
</tr>
<tr>
<td>Weight</td>
<td>205 gms</td>
</tr>
<tr>
<td>Torque</td>
<td>83.0 gm-cm</td>
</tr>
<tr>
<td>Part Number</td>
<td>No: CR-505-BS-2835</td>
</tr>
</tbody>
</table>

Experimental results and discussions
The experimental results of PMDC motor are discussed here for a desired speed (of 5000 RPM), for load (magnetic) conditions, and for various standard input test commands such as step, square, triangular, sinusoidal, and ramp.
For desired speed
Fig. 10 shows the step response of PMDC motor for PID, FL, and IFL controllers for a desired speed of 5000 rpm. The rise time of IFLC is observed to be 0.8 sec and is better than the FLC, and PIDC.

![Figure 10: Comparison of PIDC, FLC, and IFLC for step input.](image)

Under load condition
Fig. 10 also shows the response of PMDC motor for PID, FL, and IFL controllers for step input with load variations. The experimental results depict that during the load conditions the speed corresponding to undershoot and overshoot are less than 2% in case of PID and 5-6% in case of FLC and IFLC and the settling times are less than 2 sec for PIDC and less than 2.6 sec for IFLC and FLC. Therefore the PIDC is the best controller for load variations. Where as the responses of FL and IFL controllers are almost similar for load variations.

For various standard input test commands Square
A square input command is applied to all the controllers. Fig. 11 shows the response of PMDC motor for PID, FL, and IFL controllers for square input command. The rise time of IFLC is observed to be better than the FLC, and PIDC. It is clear from the plot that IFLC’s trail is close to the input command than the other two.

Triangular and Sinusoidal
Triangular and sinusoidal input commands are applied to all the controllers. Fig. 12 shows the response of PMDC motor for PID, FL, and IFL controllers for triangular and sinusoidal input commands respectively. In both the cases it is found that PIDC and IFLC’s trails are closer to the input command than the FLC.

Ramp
The performance of the controllers is also studied for a semi-linear input command such as ramp. The responses of all three controllers are presented in Fig. 13. Here, PID and IFLC follow the input command in the linear incrementing region of the
input. But during the trailing region of the waveform FLC and IFLC follow the input closely than the PIDC.

![Response for Square Input](image)

**Figure 11.** Comparison of PIDC, FLC, and IFLC for square input.

![Response for Triangular Input](image)

![Response for Sine Input](image)

**Figure 12:** Responses of PIDC, FLC, and IFLC for (a) triangular input, (b) sine input.
Conclusions
By studying the response of PID, FL, and IFL controllers for various inputs, it is concluded that for non-linear inputs (step and square) the FL and IFL controllers perform better than the conventional PID controller. Where as for linear inputs (triangular and sine) PID and IFL controllers are better than FLC. The above statements can be confirmed by observing the response of ramp input (raising part of the wave linear and trailing part is non-linear). PIDC response is best for linear portion of the wave where as it has poor response for non-linear portion of the wave (changing from max to zero). But for both the cases IFL responds quicker than the other controller as it contains both the controllers.

References
Design and Real Time Implementation of Integrated Fuzzy


[21] www.analogdevices.com

[22] www.maximsemiconductors.com