A New Fuzzy Logic Controller Based IPM Synchronous Motor Drive

M. A. Abido[1], *Member IEEE*, M. Nasir Uddin[2], *Member IEEE*, and M. A. Rahman[3], *Fellow, IEEE*

1. Electrical Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, 31261, Saudi Arabia.
E-mail: mabido@kfupm.edu.sa

2. Dept. of Electrical Engineering
Lakehead University
Thunder Bay, Ontario P7B 5E1, Canada
E-mail: mnuddin@mail.lakeheadu.ca

3. Faculty of Engg. & Appl. Science
Memorial University of Newfoundland
St. John's, Newfoundland A1B 3X5, Canada
E-mail: rahman@engr.mun.ca

***Abstract*- This paper presents a novel fuzzy logic controller (FLC) scheme for speed control of an interior permanent magnet synchronous motor (IPMSM) drive. The proposed FLC is designed to have less computational burden, which makes it suitable for online implementation. The FLC parameters are optimized by genetic algorithm. The complete vector control scheme incorporating the FLC is successfully implemented in real-time using a digital signal processor board DS 1102 for a laboratory 1 hp interior permanent magnet (IPM) motor. The efficacy of the proposed FLC based IPMSM drive is verified by simulation as well as experimental results at different dynamic operating conditions such as sudden load change, parameter variations, step change of command speed, etc. The proposed fuzzy logic controller is found to be a robust controller for application in IPMSM drive.**

***Keywords*-** Interior Permanent Magnet Motor, Vector Control, Real-Time Implementation, Genetic Algorithm, Fuzzy Logic, Imbedded System and Digital Signal processor.

I. INTRODUCTION

Recent developments in power semiconductor technology, digital electronics, magnetic materials and control theory have enabled modern ac motor drives to face challenging high efficiency and high performance requirements in the industrial sector. Among ac drives, the permanent magnet synchronous motor has been becoming popular owing to its high torque to current ratio, large power to weight ratio, high efficiency, high power factor and robustness [1]. These features are due to the incorporation of high energy rare-earth alloys such as Neodymium-Iron-Boron in its construction. Especially, the interior permanent magnet synchronous motor (IPMSM), which has magnets buried in the rotor core exhibits certain good properties, such as, mechanically robust rotor construction, a rotor physically non-saliency and small effective air gap. These features allow the IPMSM drive to be operated in high speed with quiet operation. However, the rotors of these machines have a complex geometry to ensure optimal use of the expensive permanent magnet material while maintaining a high magnetic field in the air-gap. This makes the control of IPMSM difficult at different dynamic operating conditions. Usually high performance motor drive systems used in robotics, rolling mills, machine tools etc. require fast and accurate speed response, quick recovery of speed from any disturbance and uncertainties. The vector control technique is used so that the IPMSM can achieve the highly desirable dynamic performance capabilities of the separately excited dc machine, while retaining the general advantages of ac over dc motors [2].

The conventional proportional-integral (PI), proportional-integral-derivative (PID) and various adaptive controllers have some limitations, as their design depend on exact machine model and accurate parameters. The difficulties of obtaining the exact d-q axis reactance parameters of the IPMSM leads to cumbersome design approach. Furthermore, the conventional fixed gain PI and PID controllers are very sensitive to disturbances [3,4]. On the other hand, the designs of intelligent controllers do not need the exact mathematical model of the system. Therefore, an intelligent controller demands special attention for speed control of high performance IPMSM drive systems.

As an intelligent controller, a simple and new type of fuzzy logic controller (FLC) is developed in the present work. In FLC, the system control parameters are adjusted by a fuzzy rule based system, which is a logical model of the human behavior for process control. In recent years, authors and other researchers [3-9] have utilized fuzzy algorithms for motor control applications. However, the real-time application of FLC has been facing some disadvantages due to its high computational burden [6,10]. That is why so far most of the reported works for motor drives are based on the simulation results only [7-9] or limited experimental results [3-6] with poor performance due to low sampling frequency, which is a major limitation for real-time digital implementation.

Thus, the objective of this paper is to develop and implement a new FLC for IPMSM drive in real-time, which overcomes the high computational burden and can handle nonlinear nature of IPMSM. The traditional FLC is complicated with too many membership functions and a large number of rules, which is the main cause of computational burden [3-9]. In this paper, the developed FLC is very simple, effective and more suitable for real-time implementation, as it has only one membership function for each of the two inputs and the output variable does not have any membership function. Therefore, a sampling frequency of 10 kHz is obtained in this work. Moreover, the parameters of the FLC are optimized by genetic algorithm (GA), which is a more scientific way than the conventional trial and error procedures [11]. The proposed controller is successfully implemented in real-time using DSP board DS-1102 for a prototype 1 hp IPMSM drive. The performance of the proposed drive is investigated both in simulation and experiment at different operating conditions in order to verify the robustness of the proposed controller.

0-7803-7817-2/03/$17.00 ©2003 IEEE

II. IPMSM DRIVE MODEL

The mathematical model of an IPMSM drive can be described by the following equations in a synchronously rotating rotor d-q reference frame as [6]:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R + pL_d & -P\omega_r L_q \\ P\omega_r L_d & R + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ P\omega_r \psi_f \end{bmatrix} \quad (1)$$

$$T_e = T_L + J_m p\omega_r + B_m \omega_r \quad (2)$$

$$T_e = \frac{3P}{2}\left(\psi_f i_q + (L_d - L_q) i_d i_q\right) \quad (3)$$

where, v_d, v_q = d- and q-axis stator voltages;

i_d, i_q = d- and q-axis stator currents;

R = stator per phase resistance;

L_d, L_q = d- and q-axis stator inductances;

T_e, T_L = electromagnetic and load torques;

J_m = moment of inertia of the motor and load;

B_m = friction coefficient of the motor;

P = number of poles of the motor;

ω_r = rotor speed in angular frequency;

p = differential operator (=d/dt);

ψ_f = rotor magnetic flux linking the stator.

It is well known that a synchronous motor is unable to self-start when supplied with a constant frequency source. The starting torque in the IPMSM drive used in this research is provided by a rotor squirrel cage winding. The starting process of the IPMSM drive can be considered as a superposition of two operating modes, namely, unsymmetric asynchronous motor mode and magnet-excited asynchronous generator mode. Therefore, the effect of shorted rotor windings has to be considered, if one wants to examine the process of run-up to the synchronization.

III. CONTROL PRINCIPLE

It is evident from equations (1-3) that the speed control can be achieved by controlling the d-q-axis components of the stator current. For extended over speed and constant power modes of operation, various control techniques have been used earlier [12-13]. For the sake of testing the proposed new FLC, the speed control over the normal mode of operation can be achieved by controlling the q-axis component i_q of the supply current as long as the d-axis current i_d is maintained at zero. Then, the resultant IPMSM model can be represented as,

$$pi_q = \frac{1}{L_q}\left(v_q - Ri_q - P\omega_r \psi_f\right) \quad (4)$$

$$v_d = -\omega_r L_q i_q \quad (5)$$

$$T_e = T_L + J_m p\omega_r + B_m \omega_r \quad (6)$$

$$T_e = \frac{3P}{2}\left(\psi_f i_q\right) \quad (7)$$

The machine parameters are given in Table-I. In the following subsections the FLC and GA used for tuning of the FLC parameters are explained.

TABLE-I: MACHINE PARAMETERS

Motor rated power	3-phase, 1 hp
Rated voltage	208 V
Rated current	3 A
Rated frequency	60 Hz
Pole pair number (P)	2
d-axis inductance, L_d	42.44 mH
q-axis inductance, L_q	79.57 mH
Stator resistance, R	1.93 Ω
Motor inertia, J_m	0.003 kgm^2
Friction coefficient, B_m	0.001 Nm/rad/sec
Magnetic flux constant, ψ_f	0.311 volts/rad/sec

A. Fuzzy Logic Control Scheme

The block diagram of the fuzzy logic based vector control of IPMSM drive is shown in Fig.1. In this figure i_q^* is the control output, which can be defined as,

$$i_q^*(k) = i_q^*(k-1) + U(k) \quad (8)$$

At time t, $u(t)$ is given by,

$$u(t) = U(k) \quad ; kT_s < t < (k+1)T_s \quad (9)$$

The value of $U(k)$ is determined at each sampling time based on fuzzy logic through the following steps:

Step 1: The speed deviation, $\Delta\omega(k)$, is measured at every sampling time, and the acceleration of the machine, $A(k)$, is calculated as follows,

$$A(k) = [\,\Delta\omega(k) - \Delta\omega(k-1)\,] / T_s \quad (9)$$

Step 2: Compute the scaled acceleration, $A_s(k)$, using

$$A_s(k) = A(k) * F_a \quad (10)$$

Step 3: The motor operating condition as shown in Fig.2 is given by the point $C(k)$ where

$$C(k) = (\,\Delta\omega(k)\,, A_s(k)\,) \quad (11)$$

Step 4: Calculate $R(k)$ and $\theta(k)$ using

$$R(k) = |\,C(k)\,| \quad (12)$$

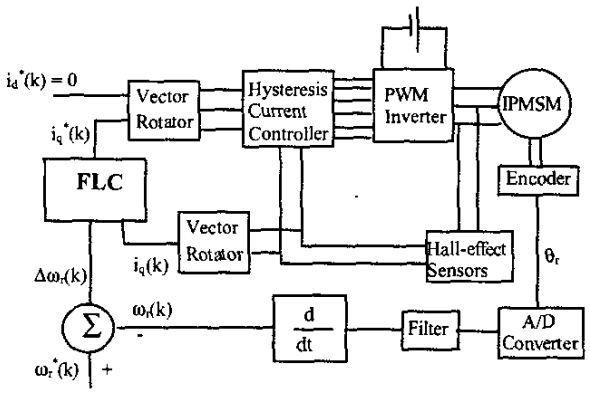


Fig.1. Block diagram of the proposed FLC based IPMSM drive.

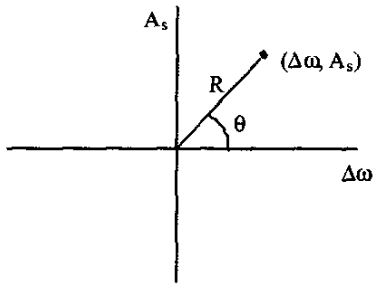
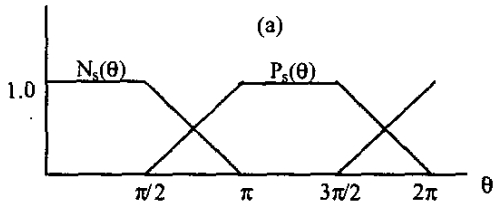


Fig.2. The motor operating point.



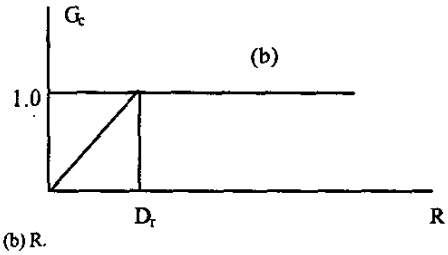


Fig.3. Membership functions for:, (a) θ, and (b) R.

$$\text{and, } \theta(k) = \tan^{-1}\left(A_s(k) / \Delta\omega(k)\right) \quad (13)$$

Step 5: Compute the values of membership functions $N_s(\theta)$ and $P_s(\theta)$,defined as shown in Fig.3(a),

$$N_s(\theta) = \begin{cases} 1 & 0 \le \theta \le \pi/2 \\ 2(\pi-\theta)/\pi & \pi/2 < \theta \le \pi \\ 0 & \pi < \theta \le 3\pi/2 \\ 2(\theta-3\pi/2)/\pi & 3\pi/2 < \theta \le 2\pi \end{cases} \quad (14)$$

and;

$$P_s(\theta) = \begin{cases} 0 & 0 \le \theta \le \pi/2 \\ 2(\theta-\pi/2)/\pi & \pi/2 < \theta \le \pi \\ 1 & \pi < \theta \le 3\pi/2 \\ 2(2\pi-\theta)/\pi & 3\pi/2 < \theta \le 2\pi \end{cases} \quad (15)$$

Step 6: Determine the value of the gain function $G_c(k)$ defined as shown in Fig. 3(b),

$$G_c(k) = \begin{cases} R(k) / D_r, \ \forall R(k) \le D_r \\ 1.0 \ \forall R(k) > D_r \end{cases} \quad (16)$$

Step 7: Compute the stabilizing signal $U(k)$ using

$$U(k) = G_c(k)\left[P_s(\theta) - N_s(\theta)\right] U_{max} \quad (17)$$

Step 8: Increase k by 1 and return to step 1.

The parameters U_{max}, F_a, and D_r are tuned by genetic algorithm [13,14]. For the optimal settings of these parameters, following quadratic performance index J is considered:

$$J = \sum_{k=1}^{L} [kT_s \Delta\omega(k)]^2 \quad (18)$$

In the above equation (18), the speed deviation $\Delta\omega(k)$ is weighted by the respective time kT_s. The index J is selected because it reflects small settling time, small steady state error, and small overshoots. The tuning parameters are adjusted so as to minimize the index$J$.

B. Genetic Algorithms

Genetic algorithms are exploratory search and optimization procedures that were devised on the principles of natural evolution and population genetics [14]. Unlike other optimization techniques, GA work with a population of individuals represented by bit strings and modify the population with random search and competition. Typically, the GA starts with little or no knowledge of the correct solution depending entirely on responses from interacting environment and their evolution operators to arrive at optimal or near optimal solutions. In general, GA includes operations such as reproduction, crossover, and mutation. Reproduction is a process in which a new generation of population is formed by selecting the fittest individuals in the current population. Crossover is the most dominant operator in GA. It is responsible for producing new offsprings by selecting two strings and exchanging portions of their structures. The new offsprings may replace the weaker individuals in the population. Mutation is a local operator, which is applied with a very low probability. Its function is to alter the value of a random position in a string.

B.1. Real-Coded Genetic Algorithm (RCGA)

Due to difficulties of binary representation when dealing with continuous search space with large dimension, the proposed approach has been implemented using real-coded genetic algorithm (RCGA). A decision variable x_i is represented by a real number within its lower limit a_i and upper limit b_i, i.e. $x_i \in [a_i, b_i]$.

The RCGA crossover and mutation operators are described as follows:

Crossover: A blend crossover operator has been employed in this study. This operator starts by choosing randomly a number from the interval $[x_i - \alpha(y_i - x_i), y_i + \alpha(y_i - x_i)]$, where x_i and y_i are the i^{th} parameter values of the parent solutions, and $x_i < y_i$. To ensure the balance between exploitation and exploration of the search space, $\alpha = 0.5$ is selected.

Mutation: The non-uniform mutation operator has been employed in this study. The new value x_i' of the parameter x_i after mutation at generation t is given as

$$x_i' = \begin{cases} x_i + \Delta(t, b_i - x_i) & \text{if } \tau = 0 \\ x_i - \Delta(t, x_i - a_i) & \text{if } \tau = 1 \end{cases} \quad (19)$$

and;

$$\Delta(t, y) = y\left(1 - r^{\left(1 - \frac{t}{g_{max}}\right)^{\beta}}\right) \quad (20)$$

where τ is a binary random number, r is a random number $r \in [0,1]$, g_{max} is the maximum number of generations, and β is a positive constant chosen arbitrarily. In this study, $\beta = 5$ was selected. This operator gives a value $x_i' \in [a_i, b_i]$ such that the probability of returning a value close to x_i increases as the algorithm advances. This makes uniform search in the initial stages and very locally at the later stages.

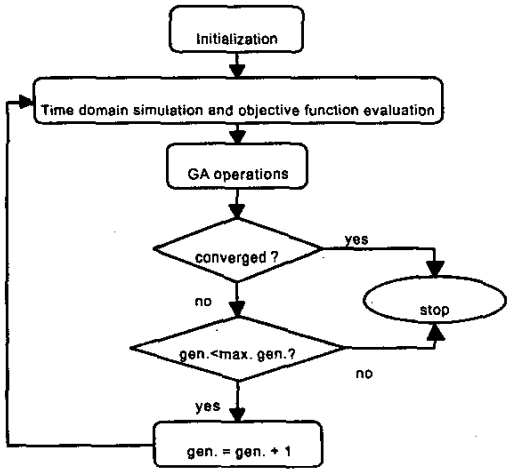


Fig.4. Computational flow chart for GA.

Applying GA to the problem of FLC design involves repetitively performing the following two basic steps.

Step 1: The objective function value must be calculated for each of the strings in the current population.

Step 2: GA operations are applied to produce the next generation of the strings.

These steps are repeated until the population has converged. The computational flow of the problem can be shown in Fig. 4. The optimal parameters are chosen as: $U_{max} = 3$, $D_r = 10$ and $F_a = 7$.

IV. REAL-TIME IMPLEMENTATION

The proposed FLC for IPMSM is experimentally implemented using digital signal processor (DSP) board DS1102 through both hardware and software [15]. The DSP board is installed in a PC with uninterrupted communication capabilities through dual-port memory. The hardware schematic for real-time implementation of the proposed FLC based IPMSM drive is shown in Fig.4. The DS1102 board is based on a Texas Instrument (TI) TMS320C31, 32-bit floating point digital signal processor. The DSP has been supplemented by a set of on-board peripherals used in digital control systems, such as A/D, D/A converters and incremental encoder interfaces. The DS 1102 is also equipped with a TI TMS320P14, 16-bit micro controller DSP that acts as a slave processor and is sued for some special purposes. In this work, slave processor is used for digital I/O configuration. The actual motor currents are measured by the Hall-effect sensors, which have good frequency response and fed to the DSP board through A/D converter. As the motor neutral is isolated, only two phase currents are fed back and the other phase current is calculated from them. The rotor position is measured by an optical incremental encoder, which is mounted at the rotor shaft end. Then it fed to the DSP board through encoder interface. The encoder generates 4096 pulses per revolution. By using a 4-fold pulse multiplication the number of pulses is increased to 4x4096 in order to get better resolution. A 24-bit position counter is used to count the encoder pulses and is read by a calling function in the software.

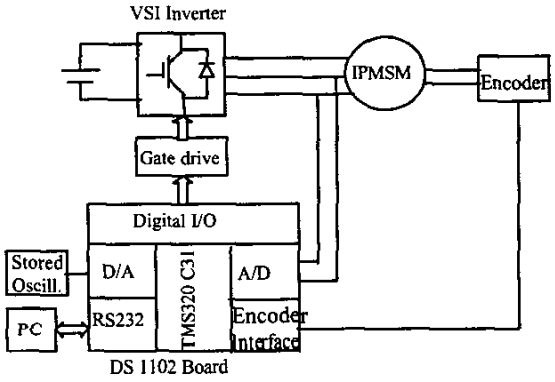


Fig.5. Hardware schematic for real-time implementation.

The motor speed is calculated from the rotor position by backward difference interpolation. A digital moving average filter is used to remove the noise from the speed signal. The calculated actual motor speed is used to calculate the torque component of the current i_q^* using the FLC algorithm. The command a-b-c phase currents are generated from i_q^* using inverse Park's transformation [3]. In order to implement the vector control algorithm, the hysteresis controller is used as current controllers. The hysteresis current controller compares the command currents with the corresponding actual motor currents and generates the logic signals, which act as firing pulses for the inverter switches. Thus, these six PWM logic signals are the output of the DSP board and fed to the base drive circuit of the inverter power module. The D/A channels are used to capture the necessary output signals in digital storage oscilloscope.

The complete IPMSM drive is implemented through software by developing a program in high level ANSI 'C' programming language. The program is compiled by the TI 'C' compiler and then the program is downloaded to the DSP controller board. The sampling frequency for experimental implementation of the proposed IPMSM drive system is 10 kHz.

V. RESULTS AND DISCUSSION

In order to establish the effectiveness of the proposed FLC scheme, the performance of the IPMSM drive based on the proposed control scheme is investigated both in simulation and experiment at different operating conditions. Sample results are presented below. The complete drive has been simulated using Matlab/Simulink [16].

The simulated motor speed and current responses are shown in Figs. 6(a)-(c) to see the starting performance as well as the response with a load disturbance of the drive. The drive system is started at a constant load of 1 Nm with the speed reference set at 1800 rpm (188.5 rad/sec). It can be seen from Fig. 6(a) that the actual speed converges to the reference value within 0.1 seconds without any overshoot and undershoot and with zero steady-state error. At t=0.3 seconds, a load torque of 2 Nm is applied to the motor shaft in a stepwise manner. The actual speed does not change during the disturbance while the stator current swiftly reaches to its new value corresponding to the load applied. This shows the capability of new controller to start from standstill condition to the rated speed as well as to reject the disturbance. Another simulated speed and the corresponding q-axis command current i_q^* responses for a sudden change in command speed are shown in Fig. 7(a) & 7(b). It is evident from Fig. 7 that the proposed FLC based drive is also capable of handling the disturbance in speed command.

The experimental starting performance including speed, torque and stator current i_a are shown in Figs. 8(a)-(c), respectively. It is shown that the proposed drive is also capable of following the command speed very quickly with zero steady-state error and without any overshoot or undershoot in a real-time situation.

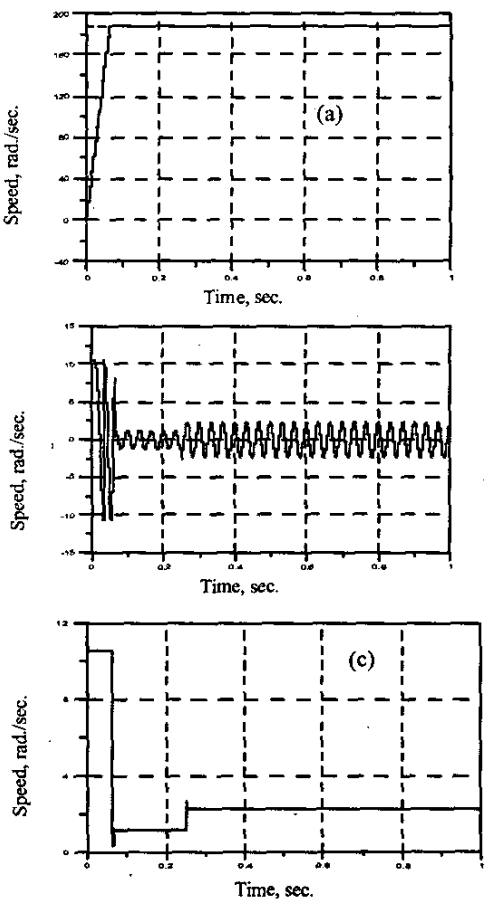


Fig.6. Simulated starting responses of the drive: (a) speed, (b) current, i_a and (c) current, i_q.

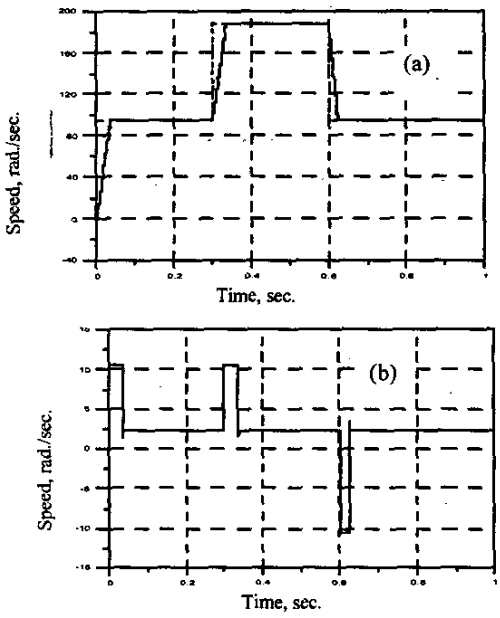


Fig.7. Simulated responses of the drive for a sudden increase in speed: (a) speed, (b) current, i_q.

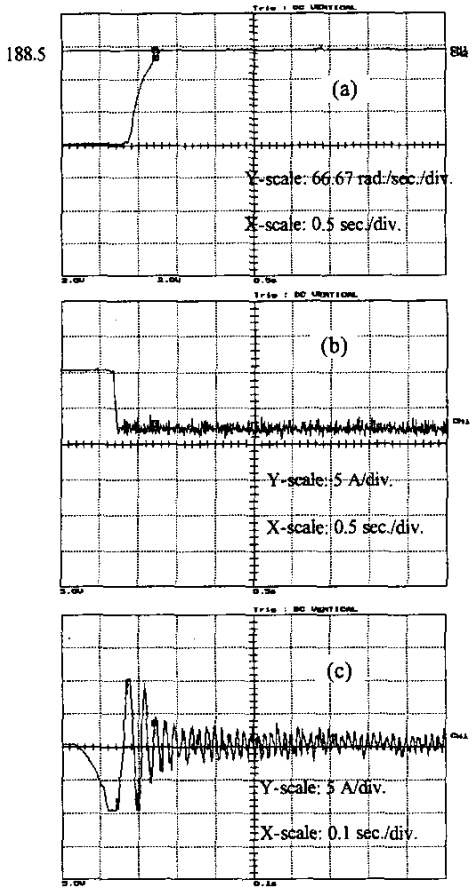


Fig.8. Experimental starting responses of the drive: (a) speed, (b) torque, and (c) phase current, i_a.

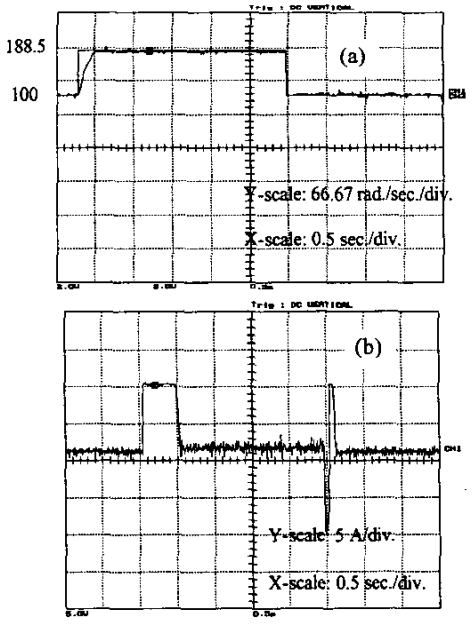


Fig.9. Experimental responses of the drive for a sudden change in speed: (a) speed, (b) torque.

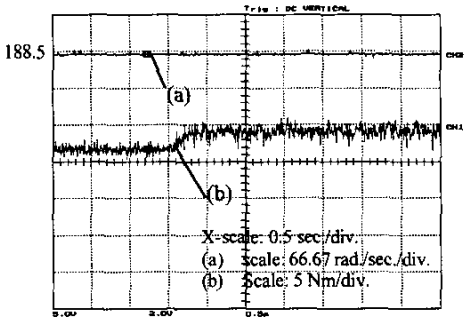


Fig.10. Experimental responses of the drive for a step increase in load: (a) speed, (b) torque.

Another set of experimental speed and the corresponding torque responses are shown in Fig. 9(a) & 9(b), respectively, for a step change in command speed. It is evident that the proposed drive can adapt itself with speed disturbance. Fig.10 shows the speed and the corresponding torque responses for a step increase in load torque from the dynamometer. It is shown in this figure that the drive is insensitive with load disturbance. Thus, the proposed GFLC based IPMSM drive is found robust at different dynamic operating conditions.

VI. CONCLUSIONS

A new FLC has been developed and successfully implemented in real-time for IPMSM drive. The effectiveness of the proposed control technique has been established both in simulation and experiment at different operating conditions. There is a close agreement between simulation and experimental results. The unique contribution of this paper is that the developed FLC is very simple, as it has only one membership function for each of the two inputs, and the output variable does not have any membership function. This results in a low real-time computational burden and hence, found very effective in terms of speed tracking with disturbance rejection.

REFERENCES

[1] G. R. Slemon, "Electric Machines and drives", Addison-Wesley Publication Company, 1992, pp. 503-511.

[2] F. Blaschke, *The Principle of Filed Orientation as Applied to The New Transvector Closed-Loop Control System for Rotating-Field Machines*, Siemens Review, Vol. 34, No.3, pp. 217- 220, May 1972.

[3] M. N. Uddin, T. S. Radwan and M. A. Rahman, "Performances of Fuzzy Logic Based Indirect Vector Control for Induction Motor Drive", IEEE Trans. on Industry Applications, Vol. 38, No. 5, Sept./Oct. 2002, pp. 1219-1225.

[4] Z. Ibrahim and E. Levi, "A Comparative Analysis of Fuzzy Logic and PI Speed Control in High Performance AC Drives Using Experimental Approach", IEEE Trans. on Industry Applications, Vol. 38, No. 5, Sept./Oct. 2002, pp. 1210-1218.

[5] A. Rubaai, D. Ricketts, D. Kankam, "Experimental Verification of a Hybrid Fuzzy Control Strategy for a High Performance Brushless DC Drive", IEEE Trans. on Industry Applications, vol. 37, No. 32, pp.503-512. March /April 2001.

[6] M. N. Uddin and M. A. Rahman, "Fuzzy Logic Based Speed Control of an IPM Synchronous Motor Drive", Journal of Advanced Computational Intelligence, vol. 4, No. 3, Dec. 2000, pp. 212-219.

[7] S. Bolognani and M. Zigliotto, "Fuzzy Logic Control of a Switched Reluctance Motor Drive", *IEEE Trans. on Industry Appl.*, vol. 32, No. 5, Sept./Oct. 1996, pp. 1063-1068.
[8] B. Singh, V. K. Sharma and S. S. Murthy, "Performance Analysis of Adaptive Fuzzy Logic Controller for Switched Reluctance Motor Drive System", *IEEE/IAS Annual Meeting Conference Record*, 1998, pp. 571-579.
[9] E. Cerruto, A. Consoli, A. Raciti and A. Testa, "Fuzzy Adaptive Vector Control of Induction Motor Drives", *IEEE Trans. on Power Elect.*, vol. 12, No. 6, Nov. 1997, pp. 1028-1039.
[10] S. Bolognani and M. Zigliotto, "Hardware and Software Effective Configurations for Multi-Input Fuzzy Logic Controllers", *IEEE Trans. on Fuzzy Systems*, vol.6, no.1, Feb. 1998, pp.173-179.
[11] M. A. Abido and Y. L. Abdel-Magid, "Tuning of a fuzzy logic power system stabilizer using genetic algorithms," *Fourth IEEE International Conference on Evolutionary Computation ICEC'97*, Indianapolis, USA, April 13-16, 1997, pp. 595-599.
[12] T. M. Jahns, "Flux-Weakening Regime Operation of an Interior Permanent- Magnet Synchronous Motor Drive," *IEEE Transactions on Industry Applications*, vol. IA-23, July/August 1987,pp. 681-689.
[13] M. N. Uddin, T. S. Radwan and M. A. Rahman, "Performance of Interior Permanent Magnet Synchronous Motor Over Wide Speed Range", IEEE Trans. on Energy Conversion, vol. 17, no. 1, March 2002, pp. 79-84.
[14] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, 1989.
[15] *dSPACE*, "Digital Signal Processing and Control Engineering, Manual Guide, GmbH, Paderborn, Germany, 1996.
[16] Mat lab, Simulink User Guide, The Math Works Inc., 1997.