Robust Speed Control of an Indirect Field-Oriented Induction Machine Drive Using Fuzzy Logic Control

Y. Miloud* , A. Draou**

(*) : University Center of Saida, BP 138, En – Nasr, Saida 20000 , Algeria.
Tel: 213 – 7 –51 95 56 Fax (048) 52 42 62 E-Mail : miloudyahiadz@yahoo.fr
(**) : Institute of Electrotechnics University of Sciences and technology of Oran. BP 1505 El M'naouar, Oran,

AlgeriaTel:213-41-422377 Fax:213-41-432639 Email:adraou@yahoo.com

Abstract- A simple FC controller and its application to the speed control of an induction motor drive compared to a traditionally (PI) controller is presented in this paper. The (PI) controller has trouble meeting with parameter variations and load disturbances. The proposed fuzzy controller with a nine linguistic rules in the output in the rule base is applied to solve this problem. Computer simulations are provided to demonstrate the robustness of the proposed fuzzy controller in presence of load disturbances and parameter variations.

1. INTRODUCTION

With the field orientation control (FOC) method, induction machine drives are becoming a major candidate in high-performance motion control applications, where servo quality operation is required. Fast transient response is made possible by decoupled torque and flux control. The most widely used control method is perhaps the proportional integral control (PI) . It is easy to design and implement, but it has difficulty in dealing with parameter variations, and load disturbances [1].

Recent literature has paid much attention to the potential of fuzzy control in machine drive applications .

Generally speaking , the fuzzy controller has the features of : (a) rather than using mathematical derivations, its control algorithms are built up based on intuition and experience about the plant to be controlled ; (b) it possesses some extent of adaptive capability [2].

This paper presents a relatively simple FLC that is robust in terms of disturbance rejection,tracking performance and parameter variations [6]-[7] without the need for complex adaptive control techniques. Thi is achieved by carefully designing the rule base with a diagonal row of zeros (i.e., outputs are ''0''), that separate positive output from negative output and a nine linguistic sets in the output of the rule base.

II. THE INDUCTION MOTOR DRIVE

The block diagram of an indirect field-oriented induction motor drive is drawn in Fig. 1. It mainly consists of a squirrel-cage induction motor , a triangulo-sinusoidal voltage controlled pulse width modulated (PWM) inverter, a slip angular speed estimator, an inverse park, and an outer speed feedback control loop. The induction motor is three-phase, Y-connected, four-pole, 1.5 Kw. 220/380V, and 50Hz. The torque command T^* is generated from the speed error between the command and the measured rotor speed through the torque controller .

The equations describing the motor operation in decoupling mode are given by :

$$i_{qs} = \frac{2}{3}\frac{T^* L_r}{P L_m \phi_r} \qquad (1)$$

$$i_{ds} = \frac{\phi_r (sT_r + 1)}{L_m} \qquad (2)$$

$$\omega_{sl} = \frac{L_m}{\phi_r T_r} i_{qs} \qquad (3)$$

and $\omega_e = \omega_r + \omega_{sl}$ (4)

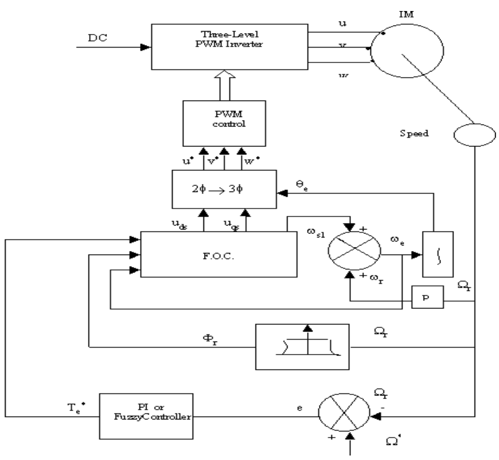


Fig.1. Indirect field orientation control block diagram

III. DESIGN OF A FUZZY LOGIC CONTROLLER

The fuzzy logic is utilised to design controllers for plants with complex dynamics that often cannot be precisely known. In a motor control system, the function of a fuzzy logic controller is to convert linguistic control rules into control strategy based on heuristic information or expert knowledge. The fuzzy logic control approach is very useful for induction motor speed drives since no exact mathematical model of the induction motor or the closed-loop system is required [3]-[4].

A FLC has a fixed set of control rules, usually derived from expert's knowledge. The membership function (MF's) of the associated input and output linguistic variables are generally predefined on a common universe of discourse. For the successful design of FLC's proper selection of input and output scaling factors (SF's) and/or tuning of the other controller parameters are crucial jobs, which in many cases are done through trial and error to achieve the best possible control performance [1],[5].

The block diagram showing the implementation of the FLC is illustrated in Fig. 2. It includes four major blocks: knowledge base, fuzzification, inference mechanism, and defuzzification. The knowledge base is composed of a data and a rule base. The data base, consisting of input and output membership functions, provides information for the appropriate fuzzification operations, the inference mechanism and defuzzification. The rule base is made of a set of linguistic rules relating the fuzzy input variables to the desired fuzzy control actions. The actual inputs to the fuzzy system are, e_N and de_N , which are a scaled version of the speed error and the change in speed error as defined by (5) and (6) .

The gains G_e and G_{de} , can be varied to tune the fuzzy controller for a desired performance.

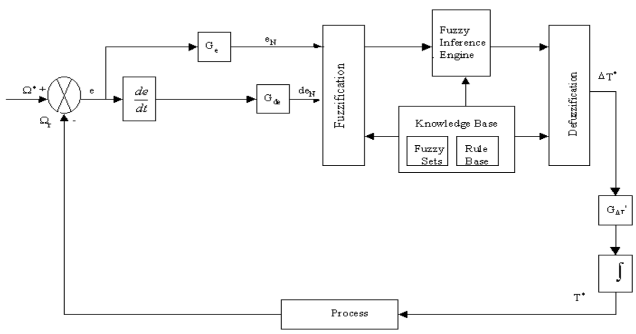


Fig.2. Fuzzy Controller block diagram

The input variables are normalised to an 'universe of discourse' with scaling factors. Using these normalised quantities, the fuzzy logic controller inputs can be described by membership factors for every linguistic code. This operation which is called ''Fuzzification'', requires the definition of linguistic sets and their membership functions. We have chosen seven linguistic sets (NB, NM, NS, ZE, PS, PM, PB) for the error, the change of error and nine linguistic sets for the output.

(a)

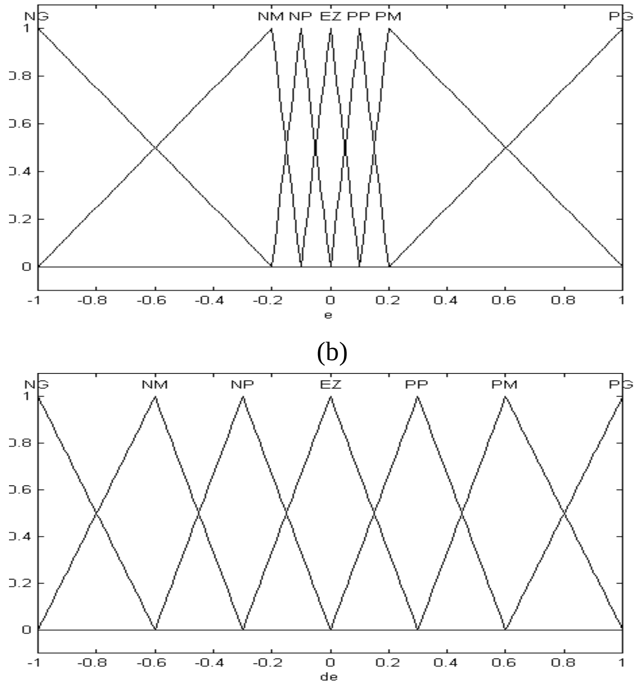
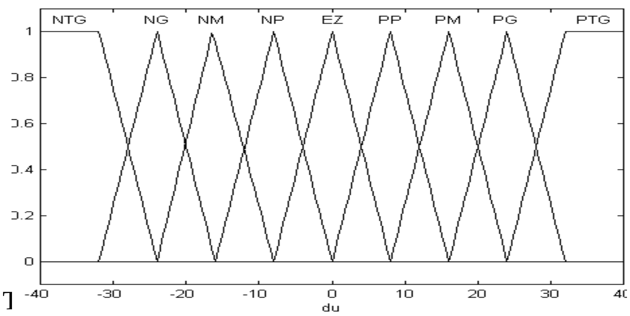


Fig. 3. Input membership functions. (a) e and (b) de.



]

Fig. 4. Output membership functions

(5)

$$de_N = G_{de}\, de \qquad (6)$$

1-Fuzzification, Inference and defuzzification

We have used symmetric triangular shapes for the change of error and output (except the two MF's at the extreme ends) which are trapezoidal and an asymmetric triangular shapes for the error. The input membership functions are defined in the interval [-1, 1] whereas the output membership functions is defined in the interval [-40, 40], Fig.3. and Fig. 4. The values of the actual inputs e and de are mapped onto [-1, 1] by the input SF's G_e and G_{de} , respectively.

The inference engine, based on the input fuzzy sets, uses the appropriate IF-THEN rules in the knowledge base to make decisions, where the Max operation is used for the premises and the Min operation is used for the implication.

The implied fuzzy set is transformed to a crisp output by the centre of gravity defuzzification technique as given by the formula (7) , z_i is the numerical output at the ith number of rules and $\mu(z_i)$ corresponds to the value of fuzzy membership function at the ith number of rules as shown in Fig. 4. The summation is from one to n, where n is the number of rules that apply for the given fuzzy inputs. The output of the fuzzy controller is integrated to give the torque command to the block of FOC (8).

$$Z_0 = \frac{\sum_{i=1}^{n} z_i . \mu(z_i)}{\sum_{i=1}^{n} \mu(z_i)} \tag{7}$$

$$T^* = T^* + G_{\Delta T}{}^* \Delta T^* \tag{8}$$

2 -The Fuzzy Rule Base

The fuzzy controller's strongest asset is the knowledge base. By carefully designing the knowledge base, the expert's experience is incorporated into the fuzzy controller.

TABLE. I.
Fuzzy controller rule base

de / e	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NVB	NVB	NVB	NB	NM	NS	ZE
NM	NVB	NVB	NB	NM	NS	ZE	PS
NS	NVB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PVB
PM	NS	ZE	PS	PM	PB	PVB	PVB
PB	ZE	PS	PM	PB	PVB	PVB	PVB

This experience is synthesised by the choice of the input-output membership functions and the rule base. In general uniformly distributed triangular membership functions are used in order to simplify the digital implementation.

This paper uses uniformly distributed triangular membership functions for both change of error and output membership functions whereas the error is a non-uniformly distributed triangular membership functions. The range for the input and output membership functions are as shown in Figs. 3 and 4, respectively. The complete control rules used in our system are shown in table. I. They are developed based on expert knowledge. The linguistic labels contained in the table are :

NVB : Negative Very Big
NB : Negative Big
NM : Negative Medium
NS : Negative Small
ZE : Zero
PS : Positive Small
PM : Positive Medium
PB : Positive Big
PVB : Positive Very Big

Most FC's have a diagonal row of zeros (i.e. , outputs are ''0''), that separate positive output from negative output as does our Fuzzy controller rule base. However, the new from this rule base compared to a typical FC rule base is the number of linguistic labels which are nine instead of seven. NVB and PVB plus the other seven typical linguistic labels. The advantage of this new rule base controller is the good performance in terms of settling time and the fast recovery in presence of load disturbances as will be seen later in the simulation results. This proves the robustness of the proposed system. For example, the rule :

IF ''e'' is ''PB'' and ''de'' is ''PS'' THEN ΔT^* is ''PVB''

This changes the torque command just enough to drive the error to zero faster than a ''PB'' output with a typical FC. Therefore, the extremes of the FC's rule base near the negative error and negative change of error and positive error and positive change of error reduce the error more effectively by incrementing the torque command, thus improving the steady state performance.

When the error and change of error are of opposite linguistic sets i.e. the output of the command torque in the diagonal is zero, the fuzzy controller will reach the command speed and will be holding at this speed.

IV. SIMULATION RESULTS

In this section, the computer simulation results for a 1.5 Kw cage rotor induction machine, using the fuzzy controller described in section III is compared to a conventional controller PI.The machine parameters are given in table II and the simulation used a voltage PWM scheme.

Fig.5. and Fig.6. show the disturbance rejection of each controller when the machine is fully loaded and operated at 1420 rpm and a load disturbance torque (2-Nm) is suddenly applied, first, at 2.5 s and then at 4.5 s. The fuzzy controller rejects the load disturbance very quickly with no overshoot and with a negligible steady state error. Whereas the PI controller takes much longer to return to speed command and presents an overshoot at the starting. Fig .7. demonstrates clearly the comparison of both controllers in presence of load disturbances. The Fuzzy controller returns the speed to the command speed within 0.05 s with a maximum drop of 5 rpm. The PI controller takes about 1.7 s to return the speed to 1420 rpm with a maximum drop of 47 rpm..The PI controller's disturbance rejection performance can be improved by readjusting the

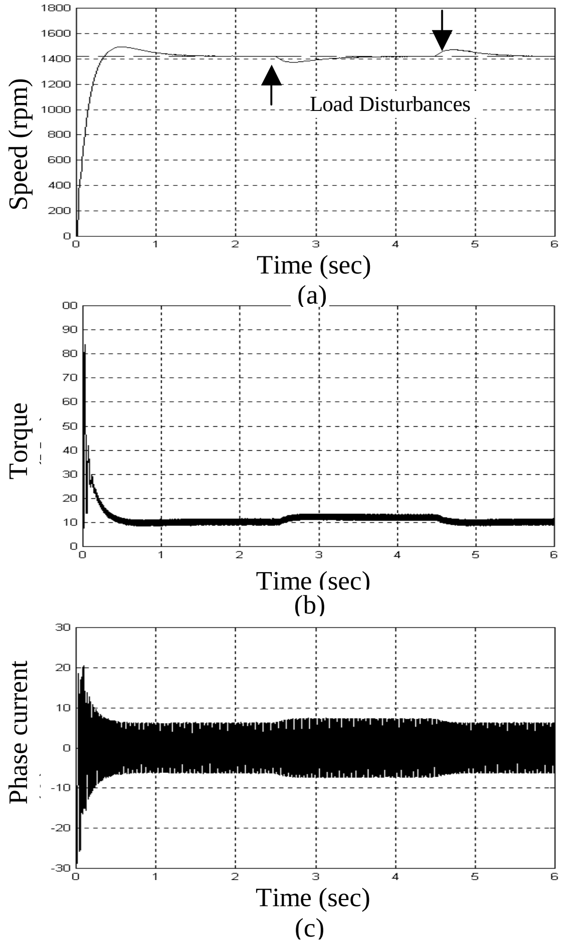


Fig. 5. PI controller : Load Torque Disturbance (± 2Nm)
(a) Speed, (b) Torque, (c) Phase Current

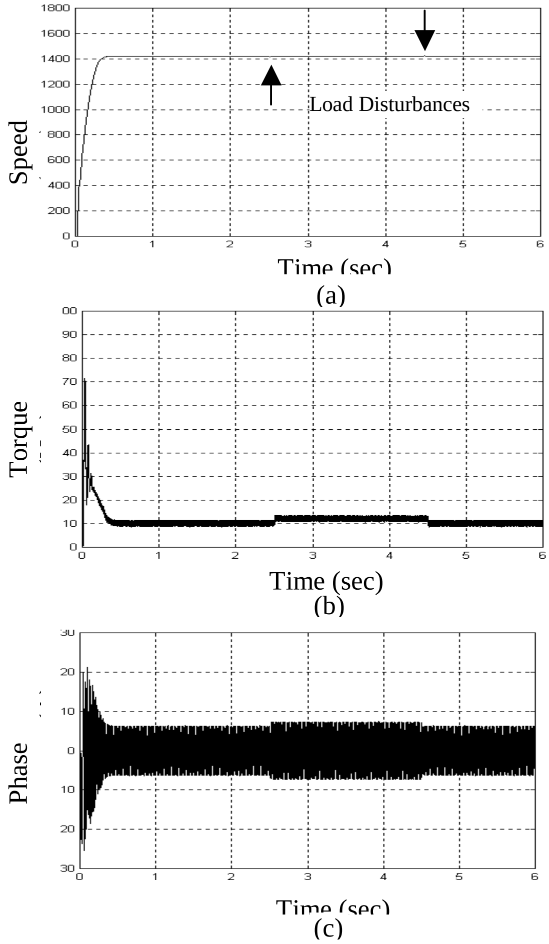


Fig. 6. FLC controller : Load Torque Disturbance (± 2Nm)
(a) Speed, (b) Torque, (c) Phase Current

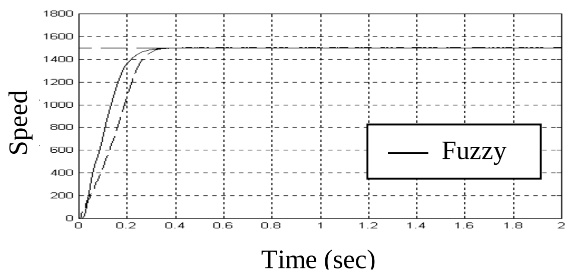


Fig. 8. Speed Drive Response : Inertia (J = J_0)

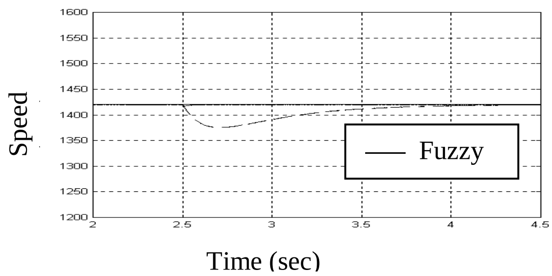


Fig. 7 .Fuzzy vs PI : 2 Nm Load Torque Disturbance

gains at the expense of speed tracking performance. For example, larger integral gains can be used to reduce the errors, but will cause serious speed overshoots .

Fig .8. shows that the system using Fuzy logic and PI control under no load has good performance in terms of settling time (0.3 s).

Fig. 9 and Fig. 10. show the speed tracking performance under no load, for both PI and Fuzzy controllers respectively. The PI controller tracks the command speed with a delay time of 0.1 s but the FLC controller tracks the command speed with no steady-state error as expected but with a small overshoot at the corners.

controller performs poorly taking about 2.5 s to restore the speed with a drop of 137 rpm, whereas the FLC controller is still performing nicely with a maximum drop of 9.5 rpm and a restoring time of 0.1 s. Next the rotor's resistance is doubled at 2 s Fig. 11. while the induction motor is still loaded (10 N.m). Fig. 12 and Fig 13 show the tracking performance when the rotor resistance is doubled at 1 s for the PI and FLC, respectively. The PI controller performs poorly when the system becomes detuned. The FLC controller still tracks the speed command and follows the trapezoidal profile but with an overshoot at the corners greater than under normal condition. The last simulation which was carried out for both controllers is shown in Fig 14 . The I. M was started with no load and with doubling moment of inertia ($2*J_0$).We notice from the graph that the speed settling time for both controllers is higher than when driving the induction machine with a rated rotor inertia (J_0) and the settling time of Fuzzy controller is better than the PI controller. started with no load and with doubling moment of inertia ($2*J_0$).We notice from the graph that the speed settling time for both controllers is higher than when driving the induction machine with a rated rotor inertia (J_0) and the settling time of Fuzzy controller is better than the PI controller.

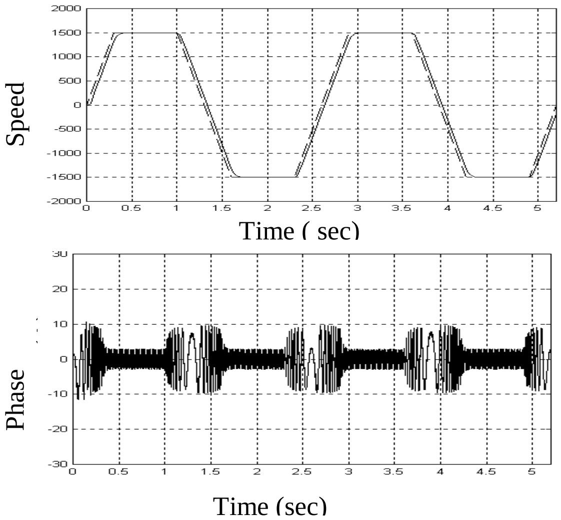


Fig. 9. PI Tracking : Speed (top), Phase current

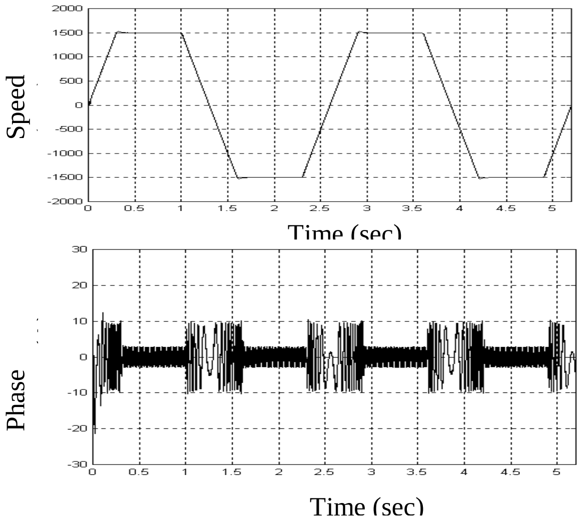


Fig.10. Fuzzy Tracking : Speed (top), Phase current

Command Speed Actual Speed

Command Speed Actual Speed

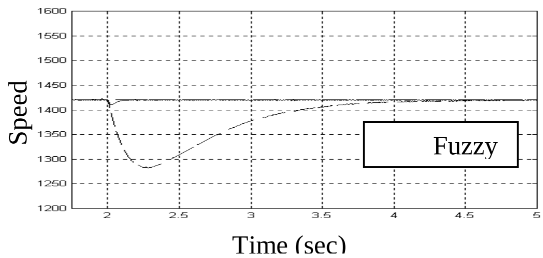


Fig . 11 Fuzzy vs PI : Parameter Variations (2*R_r)

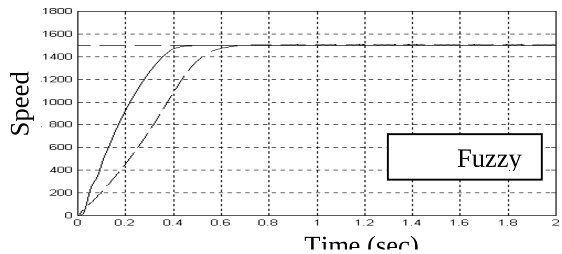


Fig. 14. Speed Drive Response : Inertia (J = 2*J_0)

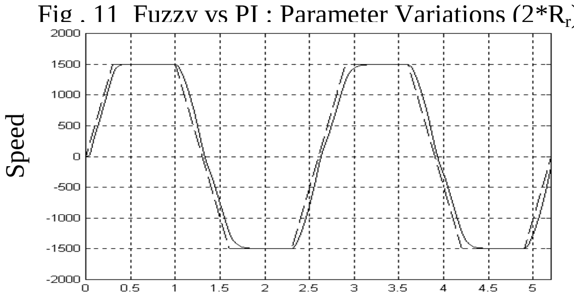


Fig . 12 PI Detuned Tracking

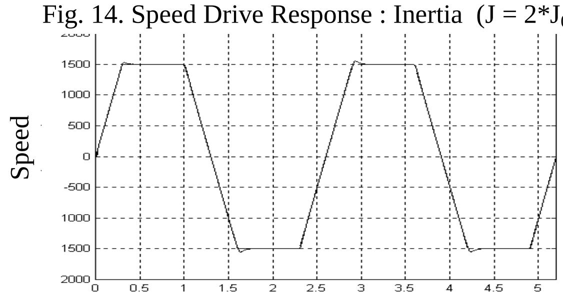


Fig . 13 FLC Detuned Tracking

V. CONCLUSIONS

A comparison between a FLC controller and a PI controller for indirect field-oriented induction motor drive has been presented in this paper. The proposed FLC controller consisting of nine linguistic sets in the output of the rule base and a non-uniformly distributed triangular membership functions for the error gave very satisfactory results in terms of load disturbances rejection and parameter variations , but its implementation is complicated .

According to different simulations carried out, the following comparisons between FLC and PI controllers are made:

- The FLC is more robust than the PI controller when a sudden load disturbance is applied.
- The performance of the FLC when parameter variations are doubled was still good and far better than the PI controller's performance when the same parameters are doubled.

REFERENCES

[1] Brian Hebert, Longya Xu and Yifan Tang, ''Fuzzy Logic Enhanced Speed Control of an Indirect Field-Oriented Induction Machine Drive''. IEEE Trans. Power Electronics. Vol. 12. NO 5. September 1997, pp 772-778.

[2] C.M. Liaw and S. Y. Cheng, '' Fuzzy Two-Degrees-of-Freedom Speed Controller for Motor Drives. IEEE Trans. Industrial Electronics. Vol. 7, February 1999 pp 209-217.

[3] Rejani K. Mudi and Nikhil R. Pal, " A Robust Self-Tuning Scheme for PI-and PD-Type Fuzzy Controllers". IEEE Trans. Fuzzy Systems, Vol. 7, February 1999, pp 2-14.

[4] Mao-Fu Lai, Michio Nakano and Guan-chyun Hsieh. " Application of Fuzzy Logic in the Phase-Locked Loop Speed Control of Induction Motor Drive". IEEE Trans. Industrial Electronics. Vol. 43 NO 6, December 1999, pp 630-638.

[5] Gilberto C. D. Sousa, Bimal K. Bose and John G. Cleland. " Fuzzy Logic Based On-Line Efficiency Optimisation Vector-Controlled Induction Motor &²Drive". IEEE Trans. Industrial Electronics. Vol 42. NO 2. April 1995, pp 192-198.

[6] Scott Wade, Mathew W. Dunnigan and Bary W. Williams. " Modelling and Simulation of Induction Machine Vector Control with Rotor Resistance Identification". IEEE Trans. Power Electronics, Vol. 12, NO. 3, May 1997, pp 495-505.

[7] Minh Ta-Cao Hoang Le-Huy. " Rotor resistance Estimation Using Fuzzy Logic for High Performance Induction Motor Drives" . IEEE 1998, pp 303-308.