Tuning of PI Speed Controller in DTC of Induction Motor Based on Genetic Algorithms and Fuzzy Logic Schemes

Shady M. Gadoue, D. Giaouris and J. W. Finch
University of Newcastle upon Tyne
Power Electronics, Drives and Machines Group,
School of Electrical, Electronic and Computer Engineering
Merz Court
NE1 7RU Newcastle upon Tyne, UK
{Shady.Gadoue, Damian.Giaouris, J.W.Finch}@ncl.ac.uk

***Abstract*- PID controllers are very common in industrial systems applications. The tuning of these controllers is governed by system nonlinearities and continuous parameter variations. In this paper, a complete and rigorous comparison is made between two tuning algorithms. The PI controller was used in a speed control loop in a Direct Torque Control (DTC) scheme applied on an induction motor. The first method applies off-line genetic algorithm (GA) strategies and the other one makes use of an online Fuzzy Logic (FL) tuning scheme. DTC is then tested with the two schemes for two cases, with normal operating conditions and with a sudden change in load torque applied to the motor. Results obtained show that the fuzzy logic on-line tuning technique can provide better speed control performance when system parameter variations occur. On the other hand for nominal operation the genetic algorithms scheme is preferred.**

Index Terms – Speed control, induction motor, direct torque control, Fuzzy logic, Genetic algorithms.

I. INTRODUCTION

Recently there has been a fast growth in industrial applications of the DTC technique. This is due to its quick torque response, simplicity and robustness against rotor parameters variation. Compared with a vector control scheme, DTC provides similar dynamic performance with simpler controller architecture [6]. The basic block diagram representation of the direct torque control of three-phase induction motors with a speed control loop is shown in Fig.1.

The most common choice for the speed controller shown in Fig. 1 is the so called PID compensator since they have a simple structure

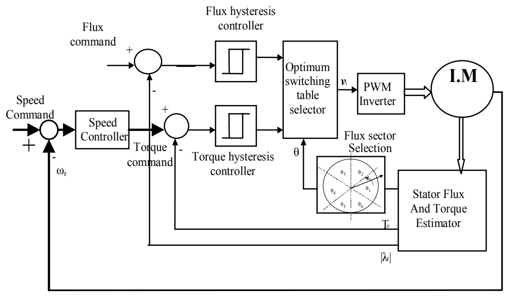


Fig.1 Block diagram of DTC with speed control loop

and they can offer a satisfactory performance over a wide range of operation.

The main problem of that simple controller is the correct choice of the PID gains and the fact that by using fixed gains, the controller may not provide the required control performance, when there are variations in the plant parameters and operating conditions. Therefore, a tuning process must be performed to insure that the controller can deal with the variations in the plant.

To tune the PI controller (usually in drives applications the derivative part of the controller is not used) a lot of strategies have been proposed. The most famous, which is frequently used in industrial applications, is the Ziegler-Nichols method which does not require a system model and control parameters are designed from the plant step response. Tuning using this method is characterised by a good disturbance rejection but on the other hand, the step response has a large percentage overshoot in addition to a high control signal that is required for the adequate performance of the system. Another technique uses frequency response methods to design and tune PI controller gains based on specified phase and gain margins as well as crossover frequency. Furthermore, root locus and pole assignment design techniques are also proposed in addition to transient response specifications. All these methods are considered as model based strategies and then the efficiency of the tuning law depends on the accuracy of the proposed model as well as the assumed conditions with respect to actual operating conditions.

Artificial Intelligence (AI) techniques such as neural networks, fuzzy logic and genetic algorithms are gaining increased interest nowadays.

A lot of techniques have been proposed to tune the gains of PI controller based on AI techniques: Self tuning FL and neural network techniques are some of these methods proposed for the online adaptive tuning of PI controller. In such application, the controller gains are tuned with the variation of system conditions. Moreover, GA is proposed for both online and offline tuning procedure [4] even though this may require high processing power. The advantage of tuning with GA is the ability of choosing controller gains which optimize drive performance based on multi objective criterion without tripping in a local minima solution. Furthermore, combinations of AI techniques are also introduced such as Neuro-Fuzzy and Genetic-Fuzzy techniques [8]. The advantage of these techniques is that they are model free strategies because they use the human experience for the generation of the tuning law.

This paper provides a comparison between two strategies used for tuning the PI speed controller in the direct torque controlled induction motor. The first one is based on Genetic Algorithms (GA), while the second one is based on Fuzzy Logic (FL). In the first method, GA searches for the optimal gains of the PI speed controller based on a fitness function and the search procedure is such that a performance index is minimized. In the other method, FL is used for tuning the PI controller online based on a group of IF-THEN control rules which mimic the human logic and are generally derived from experts' knowledge.

II. DTC Strategy

In principle, DTC is a direct hysteresis stator flux and electromagnetic torque control scheme, which triggers one of the eight available discrete voltage vectors generated by a Voltage Source Inverter (VSI) to keep the stator flux and torque within the limits of two predefined bands. The correct application of this principle allows a decoupled control of flux and torque [6].

The primary space voltage vector of the PWM inverter u_s can be expressed in terms of three switching functions s_1, s_2, s_3 and the DC link voltage (V) as:

$$u_s(s_1, s_2, s_3) = u_{sd} + j u_{sq}$$

$$= \frac{2}{3} V \left[s_1 + s_2 \exp\left(\frac{j2\pi}{3}\right) + s_3 \exp\left(\frac{j4\pi}{3}\right) \right] \tag{1}$$

Where u_{sd} and u_{sq} are the d-axis and q-axis stator voltage components.

In direct torque control schemes the magnitude of the stator flux linkage vector is controlled which can further be decomposed to its orthogonal components expressed at a stationary reference frame as:

$$\psi_{sd} = \int (u_{sd} - R_s i_{sd}) dt \tag{2}$$

$$\psi_{sq} = \int (u_{sq} - R_s i_{sq}) dt \tag{3}$$

where $\boldsymbol{\psi}_{sd}$ and $\boldsymbol{\psi}_{sq}$ are the d-axis and q-axis stator flux linkage components.

The stator flux position (θ_s) is determined by dividing the d-q plane into six 60° regions and three sign detectors are used to determine the sector over which that vector passes.

The basic DTC strategy is that the status of the errors of stator flux magnitude | $\boldsymbol{\psi}_s$ | and electromechanical torque (T_{em}) can be detected and digitalized by two- and three-level hysteresis comparators. The optimum switching table is then used to calculate the status of three switches s_1, s_2, s_3 that will determine the location of the voltage space vector $\boldsymbol{u}_s$ which depends on the stator flux angle (θ_s).

If the drive contains a speed control loop, then the reference speed input is compared with the actual motor speed and the speed error is fed to a speed controller. The output of the speed controller is the reference electromagnetic toque.

In an induction motor, the mechanical balance equation can be written as:

$$T_{em} - T_L = \frac{J}{P} \dot{\omega}_r + \frac{B}{P} \omega_r \tag{4}$$

And the electromagnetic torque is given by:

$$T_{em} = \frac{3}{2} P \left| \psi_s \right| \left| \mathbf{i}_s \right| \sin \alpha \tag{5}$$

Where T_L is the load torque, J is the combined motor and load inertia, B is the friction coefficient, P is the number of motor pole pairs, ω_r is the rotor electrical speed, $\boldsymbol{\psi}_s$ is the stator flux linkage space vector, $\mathbf{i}_s$ is the stator current space vector both expressed in the stationary reference frame and α is the angle between the stator flux linkage and stator current space vector.

III. PI Controller Tuning, Using GA

GA is a stochastic global search optimisation technique based on the mechanisms of natural selection. Recently, GA has been recognised as an effective technique to solve optimisation problems and compared with other optimisation techniques; GA is superior in avoiding local minima which is a common aspect in nonlinear systems.

GA starts with an initial population containing a number of chromosomes where each one represents a solution of the problem which performance is evaluated by a fitness function. Generally, GA consists of three main stages: Selection, Crossover and Mutation. The application of these three basic operations allows the creation of new individuals which may be better than their parents. This algorithm is repeated for many generations and finally stops when reaching individuals that are the optimum solution to the problem [3, 7]. The GA architecture is shown in Fig.2.

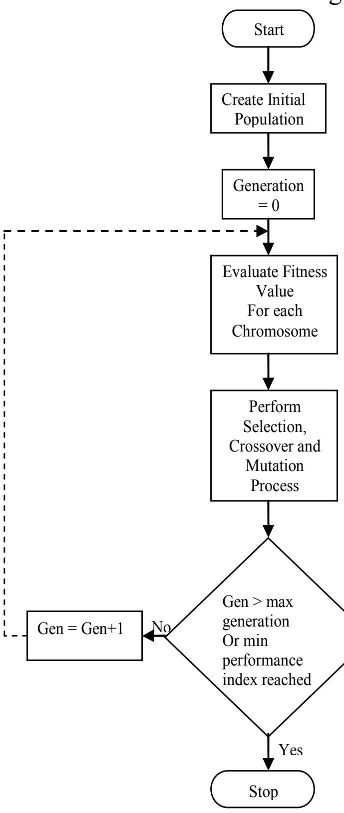


Fig.2 Genetic Algorithm Architecture

A lot of techniques have been proposed to increase the convergence speed of GA such as Parallel Genetic Algorithm based on the Allied Strategy (PGAAS) which is suitable for online applications [4], multi population techniques and others.

In the case of using a GA method to tune the PI gains in the previously mentioned speed control loop, the fitness function used to evaluate the individuals of each generation can be chosen to be the integral time of absolute error (ITAE):

$$ITAE = \int |e(t)|\, t\, dt \qquad (6)$$

During the search process, the GA looks for the optimal setting of the PI speed controller gains which minimizes the fitness function (ITAE). This function is considered as the evolution criteria for the GA. The choice of this fitness function has the advantage of avoiding cancellation of positive and negative errors. Each chromosome represents a solution of the problem and hence it consists of two genes: the first one is the Kp value and the other one is the Ki value: Chromosome vector = [Kp Ki]. It must be noted here that the range of each gain must be specified. The genetic algorithm parameters chosen for the tuning purpose are shown in Table I. Here the GA starts by generating an initial population containing 8 chromosomes then the fitness value of each chromosome in the initial population is calculated.

TABLE I
Genetic Algorithm Parameters

Number of generations	20
No of chromosome in each generation	8
No of variables in each chromosome	2
Chromosome length	40 bit
Selection method	Stochastic Universal Selection (SUS)
Crossover method	Double point
Crossover probability	0.7
Mutation rate	0.05

The three main parts of GA: Selection, Crossover and Mutation take place and then a new generation is produced. This procedure continues for some generations and then a convergence to the optimal solution represented by a given chromosome is reached.

IV. Fuzzy Self Tuning PI Controller

Fuzzy Logic Control (FLC) has been found to be excellent in dealing with systems that are imprecise, non-linear, or time-varying.

FLC is relatively easy to implement, since it does not need any mathematical model of the controlled system. This is achieved by converting the linguistic control strategy of human experience and knowledge into an automatic control strategy.

FLC has become very popular in the field of industrial control applications.

On-line tuning of controllers becomes of interest since it is very difficult for off-line tuning algorithms to deal with the continuous variations in the induction motor parameters and the nonlinearities present in inverter, motor and/or controller. The stator and rotor resistances of induction motor may change with the temperature variation up to 50%, motor magnetizing inductance varies with the magnetic operating point and becomes nonlinear near the saturation level.

Furthermore, the load torque and inertia may change due to mechanical disturbances. Nonlinearity also arises in the drive system due to voltage and current limits of the power converter and the load torque nature.

When FL is used for the on-line tuning of the PI speed controller of induction motor DTC drive, it receives the scaled values of the speed error and the change in speed error. Its output is the updating in PI controller gains (ΔKp and ΔKi) based on a set of rules to maintain excellent control performance even in the presence of parameter variation and drive nonlinearity [1].

The most important parameters of Fuzzy Logic are the scaling factors. The input scaling factors affect the FLC sensitivity while the output scaling factors affect the system stability [7]. The block diagram of the control system is shown in Fig.3.

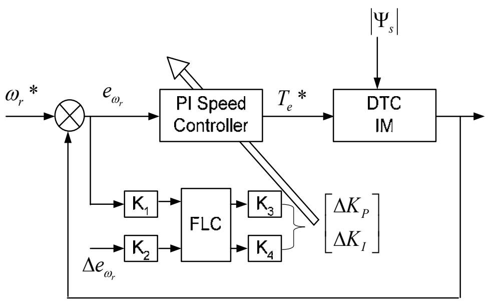


Fig.3 Fuzzy self tuning PI speed controller

A lot of researches have proposed tuning methods for the FL scaling factors: Han et al. [2] proposed a neural network with variable learning rate. The artificial neural network is used to represent the output part of the FLC, and then a back propagation algorithm is used to adjust the weights of the neural network. In this case, the weights represent the output scaling factors of FLC. Mokrani et al. [5] proposed a fuzzy adaptation scheme for the purpose of tuning the scaling factors of FLC.

Each input of the FLC has 5 triangular membership functions with equal width and overlap. The first output (ΔKp) has 3 triangular membership functions, while as the second output (ΔKi) has 5 membership functions. The inference rules base has 25 rules. Membership functions can be tuned by trial error techniques or using any other tuning strategies such as GA. The Fuzzy inference rules used for on-line tuning of PI controller gains are shown in table II and table III [1].

The flow chart of the Fuzzy self tuning PI speed controller is shown in Fig.4.

TABLE II
FUZZY INFERENCE RULES FOR UPDATING THE PROPORTIONAL GAIN **(ΔK_p)**

e_ω / Δe_ω	**NB**	**NS**	**ZE**	**PS**	**PB**
NB		PB	PB	PB	
NS		PB	PS	ZE	
ZE		PB	ZE	PB	
PS		ZE	PS	PB	
PB		PB	PB	PB	

TABLE III
FUZZY INFERENCE RULES FOR UPDATING THE INTEGRAL GAIN **(ΔK_I)**

e_ω / Δe_ω	**NB**	**NS**	**ZE**	**PS**	**PB**
NB	ZE	NS	NB	NS	ZE
NS	PS	ZE	NS	ZE	PS
ZE	PB	PS	ZE	PS	PB
PS	PS	ZE	NS	ZE	PS
PB	ZE	NS	NB	NS	ZE

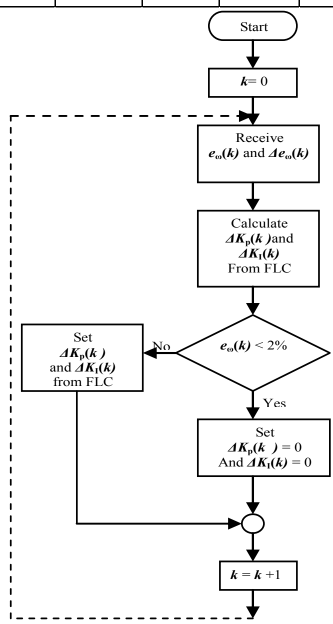


Fig.4 Flow chart of Fuzzy self tuning PI speed controller

V. Simulation Results And Discussion

To compare the two tuning strategies, DTC of a 10 hp squirrel cage induction motor with speed control loop shown in Fig.1 is simulated using Matlab-Simulink software. GA is first applied to the PI speed controller tuning. During the tuning process, the induction motor is loaded with 25% of rated load torque. A speed reference of 50 rad/s in addition to the rated stator flux linkage magnitude is the input commands to the drive.

The speed response of the motor is observed and the ITAE is recorded for each chromosome in each generation. After 20 generations, the optimal solution proposed by GA under this loading condition is achieved with the chromosome consisting of Kp = 151.6 and Ki = 3.98.

When the drive system is simulated using the proposed gains, which are set to be constant during the simulation period, with the motor operating with a load torque of 25 % for 2 s, the ITAE is equal to 0.1269.

The Fuzzy self-tuning PI speed controller, with the parameters listed in Table IV, is simulated with the same conditions and the ITAE obtained is 0.1319. This discrepancy was expected since the GA scheme found the "optimum" gains for the specific conditions while the FLC used the "imperfect" human knowledge and experience.

TABLE IV
FUZZY LOGIC PARAMETERS

Input scaling factors K_1, K_2	1.1, 0.1
Output scaling factors K_3, K_4	0.2, 1.1
Defuzzification method	Centre of gravity
Kp initial	10
Ki initial	12

To investigate the performance of the tuning strategies under system parameter variation, the induction motor is subjected to a 100 % load torque change (from 25% to 50% of the motor rated torque) at t = 1 s.

As shown in Figs.5 and 6, the speed response using GA tuning has a speed overshoot of 0.6 % while that from FL auto tuning has no overshoot and the drive system behaves like a critical damped system. At t=1s when the load is increased, the speed response using fixed gains (obtained by GA) drops to 49.94 rad/s with a speed regulation of 0.12 % with a total ITAE (after 2 s) of 0.2132, while as that obtained by FL self tuning drops rapidly to 49.6 rad/s before the speed builds up again to track the reference value (50 rad/s) with a total ITAE of 0.1641. The ITAE obtained by each tuning technique is shown in Fig.7. The results are summarised in table V.

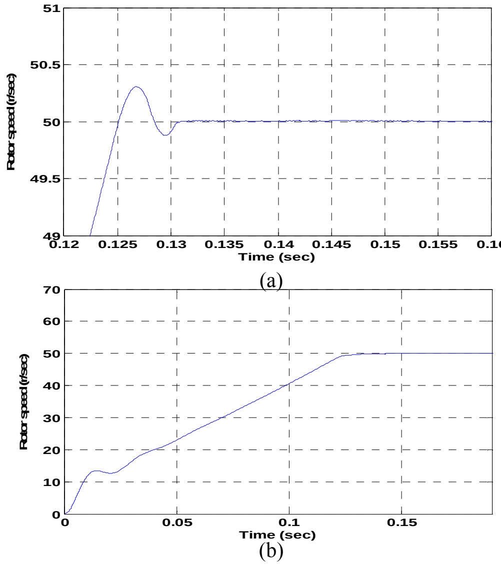


Fig.5 Transient response (a) GA tuning (b) FL tuning

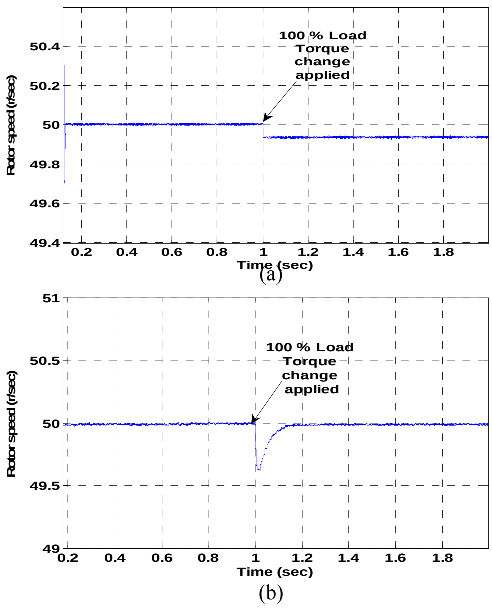


Fig.6 Speed response at time of application of 100% load torque increase at t=1s (a) GA tuning (b) FL tuning

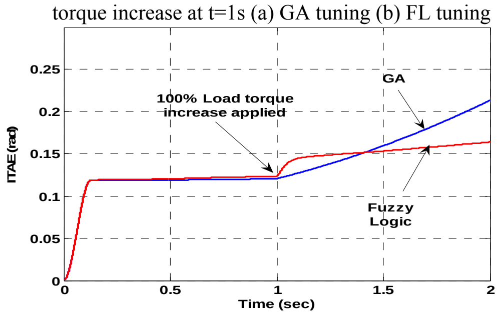
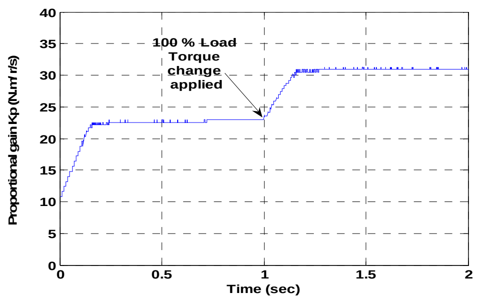


Fig.7 ITAE for GA and FL tuning

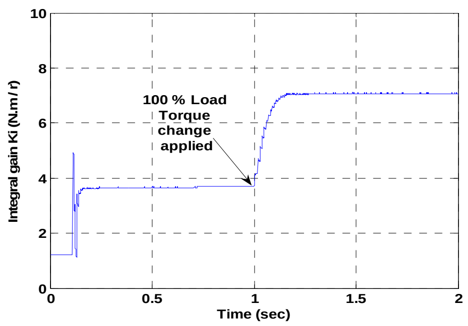
TABLE V
SUMMARY OF RESULTS

	GA $K_p = 151.6$, $K_i = 3.98$	**FL** Variable Gains
ITAE (t=0to t=1s)	0.1207	0.125
ITAE (t=1to t=2s)	0.0925	0.039
Total ITAE	0.2132	0.1641
Transient response	0.6 % speed overshoot	No overshoot
Response to load torque application	Speed drop with 0.12 % speed regulation	Initial drop then quickly regain the speed

The PI controller gain variations for the self tuning Fuzzy Logic scheme are shown in Fig.8 (a) and Fig.8 (b). The gains increase gradually until the command speed is reached. After that, the gains are kept constant. When the load torque increase occurs, they rise again to compensate for the speed decrease and to return the speed back to the command value. That is, to ensure complete speed tracking.



(a)



(b)

Fig.8 Fuzzy Self tuning Performance (a) Proportional gain (b) Integral gain

VI. CONCLUSION

A comparison between two artificial intelligence based techniques used for the tuning of the PI speed controller in DTC of induction motor has been presented. Under normal operating conditions, an off-line GA tuning scheme shows better performance than FL auto tuning since it contains the optimal controller gains.

When a change in system parameters takes place, the results show that self tuning FL is superior in terms of speed reference tracking than fixed gains controller obtained from GA tuning. This is due to the adaptive behaviour of FL when used with an online tuning strategy.

REFERENCES

[1] L. Mokrani and R. Abdessemed, "A Fuzzy self-tuning PI controller for speed control of induction motor drive", *Proceedings of 2003 IEEE Conference on Control Applications,* IEEE Part vol.2, 2003, pp.785-90 vol.2. 'Istanbul, Turkey'.

[2] Woo-Yong Han, Sang-Min Kim, Sung-Joong Kim and Chang-Goo Lee, "Sensorless vector control of induction motor using improved self-tuning Fuzzy PID controller", *SICE 2003 Annual Conference*. Soc. of Instrum. and Control Eng. Part Vol.3, 2003, pp.3112-17 Vol.3. Tokyo, Japan.

[3] The MATLAB Genetic Algorithm Toolbox v1.2 User's Guide, Department of Automatic Control and Systems Engineering of The University of Sheffield, UK. *http://www.shef.ac.uk/acse/research/ecrg/gat.html*

[4] Lin Feng, Zheng Hongtao and Yang Qiwen, "Sensorless vector control of induction motors based on online GA tuning PI controllers", *Fifth International Conference on Power Electronics and Drive Systems.* IEEE Vol.1, 2003, pp.222-5 Vol.1. 'Singapore'.

[5] L. Mokrani and K. Kouzi, "Influence of fuzzy adapted scaling factors on the performance of a Fuzzy logic controller based on an indirect vector control for induction motor drive", *Journal of Electrical Engineering*, Vol.55, No. 7-8, 2004, pp.188-194.

[6] A. L. Mohamadein, R. Hamdy and Shady M. Gadoue, "A Comparison Between Two Direct Torque Control Strategies For Flux And Torque Ripple Reduction For Induction Motors Drives", *Proceedings of the Ninth International Middle East Power Systems Conference* (MEPCON'2003), Shebeen Al-Koum, Egypt, December 16-18,2003.

[7] P.Vas, "Artificial-Intelligence-Based Electrical Machines and Drives-Application of Fuzzy, Neural, Fuzzy-Neural and Genetic Algorithm Based Techniques", Oxford University Press, 1999.

[8] Jing-Chung Shen, "Fuzzy Neural Networks for Tuning PID Controller for Plants with Underdamped Responses", IEEE Transactions on Fuzzy Systems, Vol.9, No.2, April 2001.