KMITL Sci. Tech. J. Vol. 5 No. 3 Jul.-Dec. 2005

HIGH – PERFORMANCE ADAPTIVE INTELLIGENT DIRECT TORQUE CONTROL SCHEMES FOR INDUCTION MOTOR DRIVES

M. Vasudevan and R. Arumugam

Department of Electrical and Electronics Engineering, Anna University,
Chennai, Tamilnadu, India

ABSTRACT

This study presents a detailed comparison between viable adaptive intelligent torque control strategies of induction motor, emphasizing advantages and disadvantages. The scope of this study is to choose an adaptive intelligent controller for induction motor drive proposed for high performance applications. Induction motors are characterized by complex, highly non-linear and time varying dynamics and inaccessibility of some states and output for measurements and hence can be considered as a challenging engineering problem. The advent of torque and flux control techniques have partially solved induction motor control problems, because they are sensitive to drive parameter variations and performance may deteriorate if conventional controllers are used. Intelligent controllers are considered as potential candidates for such an application. In this paper, the performance of the various sensorless intelligent Direct Torque Control (DTC) techniques of Induction motor such as neural network, fuzzy and genetic algorithm based torque controllers are evaluated. Adaptive intelligent techniques are applied to achieve high performance decoupled flux and torque control. The theoretical principle, numerical simulation procedures and the results of these methods are discussed.

KEYWORDS: Direct Torque Control, Induction Motor, Intelligent Control, Fuzzy, Neural Networks and Genetic Algorithm.

1. INTRODUCTION

Direct Torque Control (DTC) of pulse - width - modulated inverter fed induction motor drive is receiving wide attention in the recent years [1-5]. Figure 1 shows the basic configuration for the direct torque controlled induction motor drive. The scheme uses stator flux vector and torque estimators on a PWM – inverter- fed drive. The stator flux amplitude ψ_s^* and torque T_e^* are the command signal and which are compared with the estimated ψ_s and T_e values, respectively, giving instantaneous flux error E_ψ and torque error E_{Te} as shown in Figure 1.

*Corresponding author.

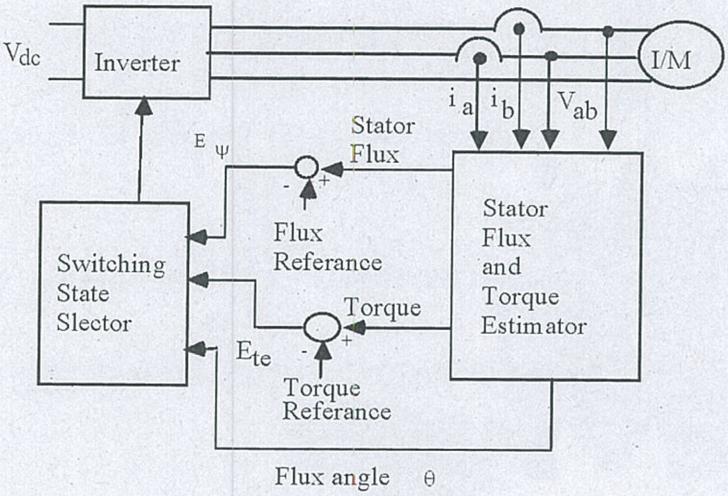


Figure 1 Basic configuration of DTC scheme

In the conventional scheme, the flux error E_ψ and torque error E_{Te} signals are delivered to two hysteresis comparators. The corresponding digitalized output variables and the stator flux position sector create a digital word, which selects the appropriate voltage vector from the switching table. Selection of voltage vector is also depending upon the sector in which the stator flux positioned. Thus, the selection table generates pulses S_a, S_b, S_c to control the power switches in the inverter. Figure 2 shows the pulses S_a, S_b, S_c generated when the position of stator flux is in sector 1.

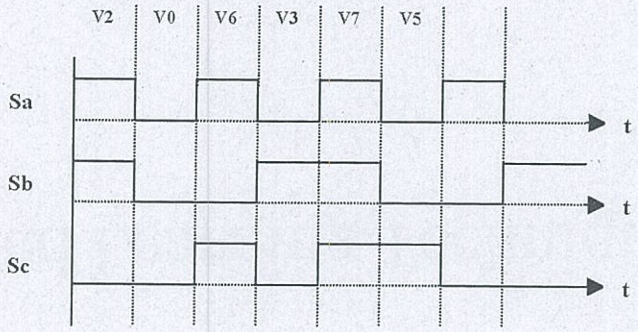


Figure 2 Generation of Pulses for PWM inverter when flux vector lies on sector 1

The expression for the developed torque of an induction motor is given by (1).

$$T = \frac{N_p M \psi_s \psi_r \sin \delta}{\sigma L_s L_r} \tag{1}$$

where:

$\sigma = 1 - M^2 / (L_s L_r)$

ψ_s = Stator flux

Under normal operating conditions, the amplitude of the working flux is kept constant at the maximum value. Hence the developed torque is proportional to the sine of the torque angle 'δ' between stator and rotor fluxes, and can be controlled by suitably changing the angle 'δ'. Since the time constant of rotor current is large compared to stator, the stator flux is accelerated or decelerated with respect to the rotor flux to change the torque angle. Stator flux is a computational quantity, which is obtained using the stator-measured current 'I_s' and voltage 'V_s'.

$$\psi_s = \int_0^t (V_s - I_s R_s)\,dt \tag{2}$$

In general, Conventional DTC scheme has the following disadvantages:

i) Variable switching frequency
ii) Violence of polarity consistency rules
iii) Current and torque distortions caused by the sector changes
iv) Starting and low - speed operation problems
v) High sampling frequency needed for digital implementation of hysteresis comparators.

Introducing adaptive controllers instead of conventional hysteresis controllers can eliminate all the above difficulties. In this study, viable intelligent controllers in DTC scheme are discussed to improve the performance in low speed operations and to minimize the torque ripples. Intelligent controls using expert systems, fuzzy logic, neural networks and genetic algorithms have been recently recognized as important tools to enhance the performance of the power electronic systems. The combination of intelligent control with adaptive and robust control appears today the most promising research accomplishment in the drive control area and in the meantime, as the best approach for the optimal exploitation of intelligent control prerogatives and practical realization of adaptive and robust ac motor drives. In this study, detailed investigations on viable intelligent torque control schemes are carried out by simulation and the results are compared.

2. NEURAL NETWORK CONTROLLERS FOR DTC SCHEME

A neural network is a machine like human brain with properties of learning capability and generalization. They require a lot of training to understand the model of the plant. The basic property of this network is that it can able to approximate complicated nonlinear functions [6-7]. In direct torque control scheme, neural network is used as a sector selector. The direct torque neuro controller is shown in Figure 3. In this control strategy, torque and flux errors are multiplied by the constant value 'c' and which are given as inputs along with the flux position information to the neural network controller. Output of the controller is compared with the previous switching states of inverter. Artificial Neural Network (ANN) offers inherent advantages over other conventional DTC schemes for induction motors, namely:

i) Reduction of the complexity of the controller;

ii) Reduction of the effects of motor parameter variations, particularly in the stator-flux estimation;

iii) Improvement of controller time response, i.e., the ANN controller uses only parallel processing of sums, products by constant gains, and a set of well known non-linear functions so that no time- consuming sequential integrations routines are required;

iv) Improvement of drive robustness – ANN's are fault tolerant and can extract useful information from noisy signals

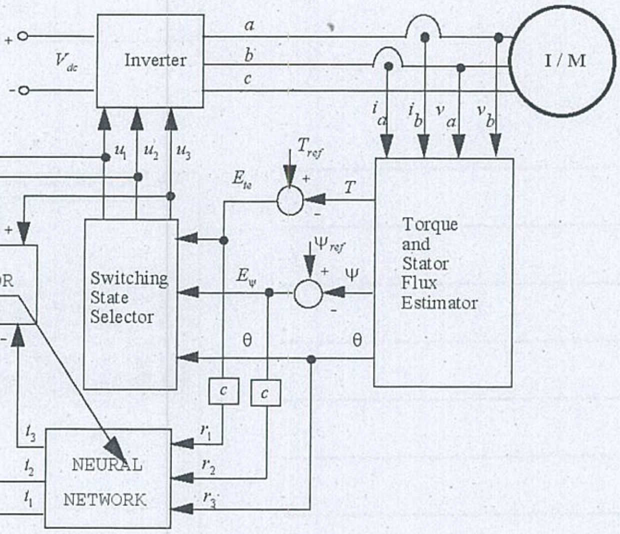


Figure 3 Schematic of DTC using Neural-Network controller

3. PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS

Feed forward artificial neural networks are universal approximators of nonlinear functions [8]. As such, the ANN's use a dense interconnection of neurons that correspond to computing nodes. Each node performs the multiplication of its input signals by constant weights, sums up the results, and maps the sum to a nonlinear function; the result is then transferred to its output. The structure of neuron is shown in Figure 4 and the mathematical model of a neuron is given by

$$y = \varphi\left(\sum_{i=1}^{N} \omega_i x_i - b\right) \tag{3}$$

where, $x_i = (x_1, x_2, ..., x_N)$ are inputs from the previous layer neurons.
$\omega_i = (\omega_1, \omega_2, ..., \omega_N)$ are the corresponding weights, and 'b' is the bias of the neuron.

For a logarithmic sigmoidal activation function, the output is given by

$$y = \frac{1}{1 + e^{\left[\sum_{i=1}^{N} \omega_i x_i - b\right]}} \tag{4}$$

A feed forward neural network is organized in layers: an input layer, one or more hidden layers, and an output layer. No computation is performed in the input layer and the signals are directly supplied to the first hidden layer through input layer. Hidden and output neurons generally have a sigmoidal activation function. The knowledge in an ANN is acquired through a learning algorithm, which performs the adaptation of weights of the network iteratively until the error between the target vectors and output of network falls below a certain error goal. The most popular learning algorithm for multi-layer networks is the back propagation algorithm, which consists of a forward and backward action. In the first, the signals are propagated through the network layer by layer. An output vector is thus generated and subtracted from the desired output vector. The resultant error vector is propagated backward in the network and serves to adjust the weights in order to minimize the output error. The back propagation training algorithm and its variants are implemented by many general – purpose software packages such as the neural-network toolbox from MATLAB [9-10] and the neural-network development systems described by Bose [11]. The time required to train an ANN depends on the size of the training data set and training algorithm. An improved version of back propagation algorithm with adaptive learning rate is proposed and which permits a reduction of the number of iterations. Figure 5 shows the proposed neural network for DTC scheme in which, input, output and hidden layers are shown. The error signals and stator flux angle are given to input layer. Switching state information is taken from the output layer.

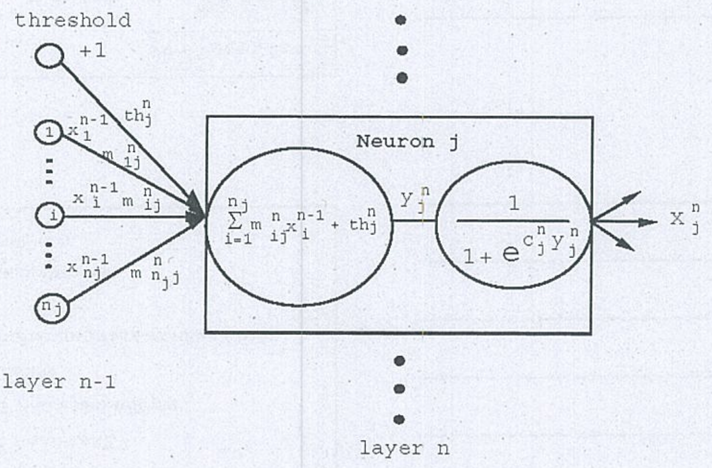


Figure 4 Structure of Neuron

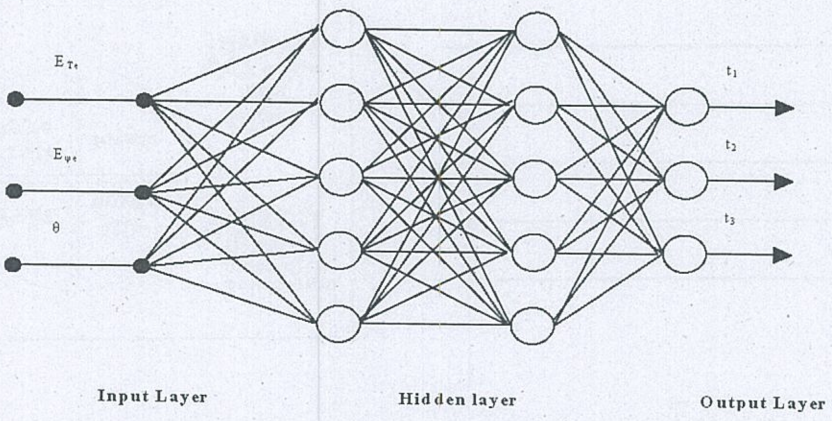


Figure 5 Structure of Neural network proposed for DTC scheme

4. DTC USING GENETIC ALGORITHM

Genetic algorithm is an advanced technique for direct torque control of induction motor drive. Genetic algorithms (GA's) are optimization procedures, which are well suitable to train the neural networks [11]. Generally, GA has following components:

v) A genetic representation for potential solution encoded as strings or chromosomes.
vi) A way to create an initial population of potential solutions
vii) An evaluation function for rating solutions in terms of their fitness
viii) Genetic operator that alter the composition of children
ix) Values for various parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators, etc.).

For executing genetic algorithm to train the neural networks, detailed procedures were followed. Figure 7 shows the flowchart to execute a genetic algorithm. It gives an algorithm to select best chromosome from the total population of chromosomes. To select best chromosome, parent selection is prominent. Steps for parent selection are as summarized as follows:

i) Selection of parents for reproduction is stochastic.
ii) Selection of parents with higher fitness value.
iii) Roulette wheel technique for parent selection. A roulette wheel shown in Figure 7 has slots, which are sized according to the fitness of each chromosome.
iv) Selection process is to spin the roulette wheel.

In Figure 6, f_1, f_2, f_3, f_4, f_5 are fitness of chromosomes 1, 2, 3, 4 and 5, respectively. Pop represents the total population size; that is, if total number of chromosomes is 50, population size is also 50. Therefore,

$$f_{pop} = f_{50} = \text{Fitness of } 50^{\text{th}} \text{ chromosome} \tag{5}$$

Total fitness is given by, F= Sum of the fitness of the population

$$F = \sum_{j=1}^{pop} eval_j \tag{6}$$

Probability function for each chromosome is

$$p_i = eval_i / F, i = 1,2,3, \cdots\cdots, pop \tag{7}$$

Accumulative probability function for each chromosome is

$$q_i = \sum_{j=1}^{i} p_i, i = 1,2,3, \cdots\cdots, pop \tag{8}$$

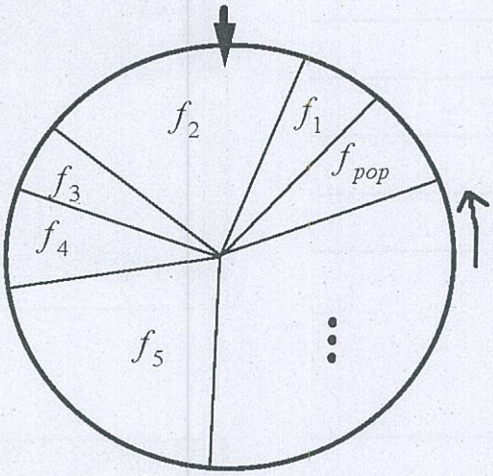


Figure 6 Roulette Wheel

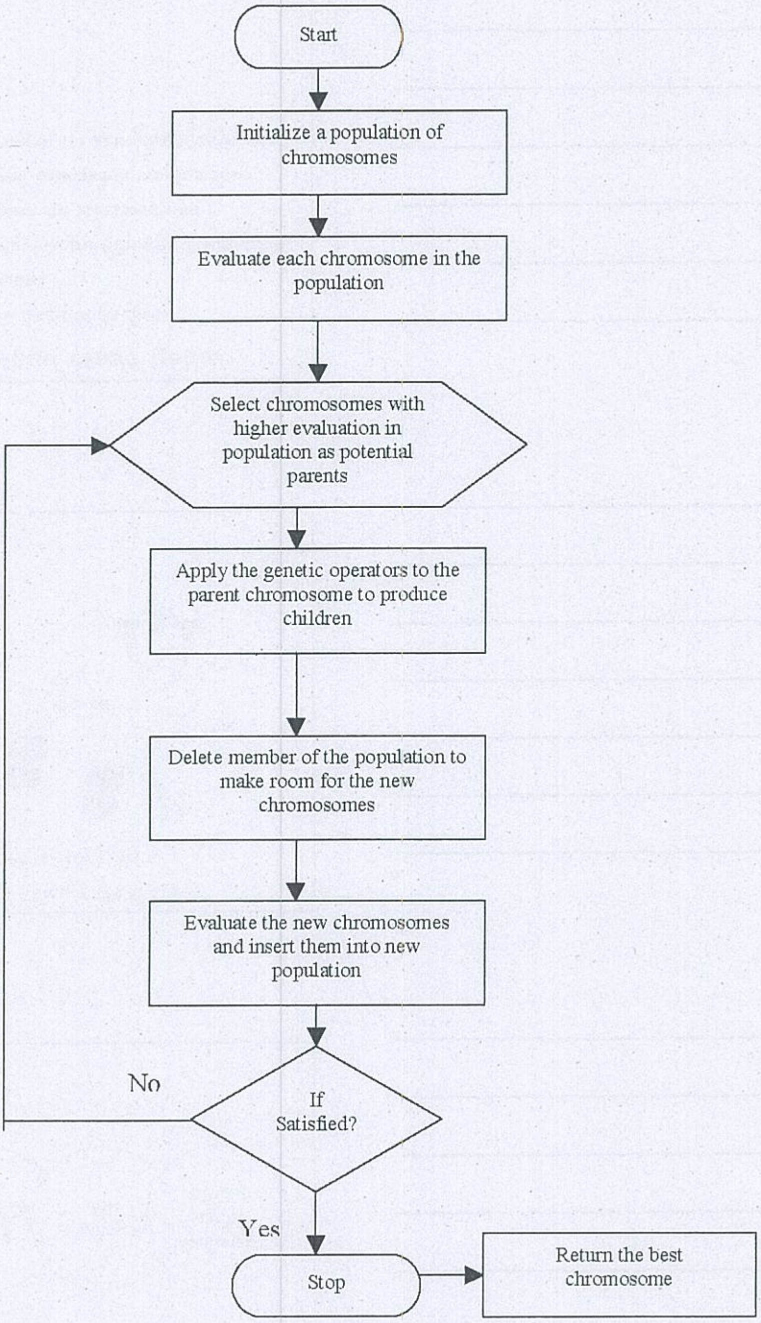

Start
Initialize a population of chromosomes
Evaluate each chromosome in the population
Select chromosomes with higher evaluation in population as potential parents
Apply the genetic operators to the parent chromosome to produce children
Delete member of the population to make room for the new chromosomes
Evaluate the new chromosomes and insert them into new population
If Satisfied?
No
Yes
Stop
Return the best chromosome

Figure 7 Flowchart for execution of a Genetic Algorithm

5. NEURAL NETWORKS TRAINED BY GENETIC ALGORITHMS

In neural networks, genetic algorithms are used to determine the weights and threshold values. Figure 8 shows the structure of neural networks trained using GA. The respective error vectors between the state selector of conventional DTC and the neural networks outputs are e_1, e_2, e_3. To achieve minimum value of performance index, the groups of threshold values and weights have to be determined.

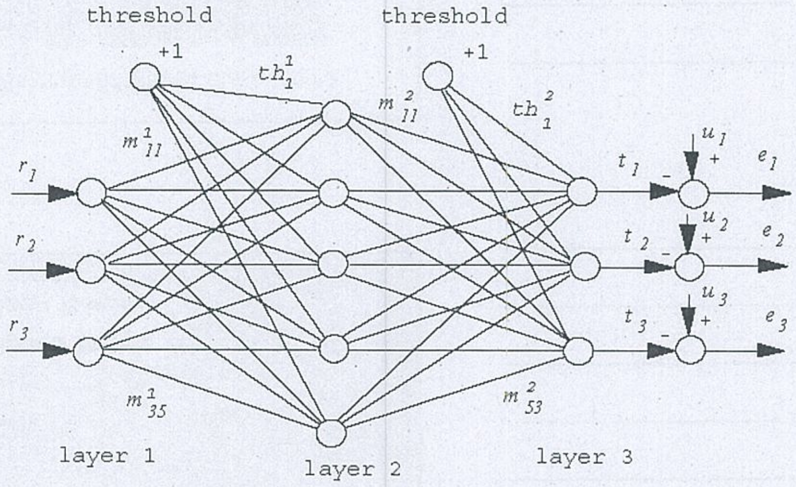


Figure 8 Structure of neural networks trained using GA

Performance index *E (W)* can be given by:

$$E(W) = \frac{1}{2}\sum_{j=1}^{N} e^T(j)\Lambda e(j) \qquad (9)$$

where: $e^T = [e_1 e_2 e_3]^T$ = error vectors

Λ = Symmetric positive definite matrix

N = Sample size

6. FUZZY LOGIC DIRECT TORQUE CONTROL OF INDUCTION MOTOR

In DTC induction motor drive, there are torque and flux ripples because none of the inverter states is able to generate the exact voltage value required to make zero both the torque electromagnetic error and the stator flux error [12-13]. The suggested technique is based on applying switching state to the inverter and the selected active state just enough time to achieve the torque and flux references values. A null state is selected for the remaining switching period, which won't almost change both the torque and the flux. Therefore, the switching state has to be determined based on the values of torque error, flux error and stator flux angle. Exact value of stator flux angle (θ) determines where stator flux lies [6].

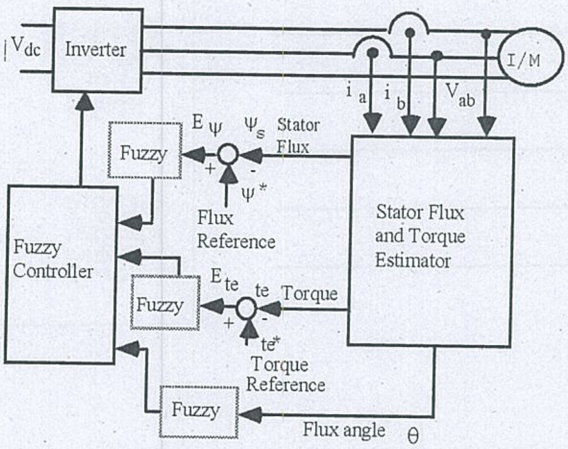
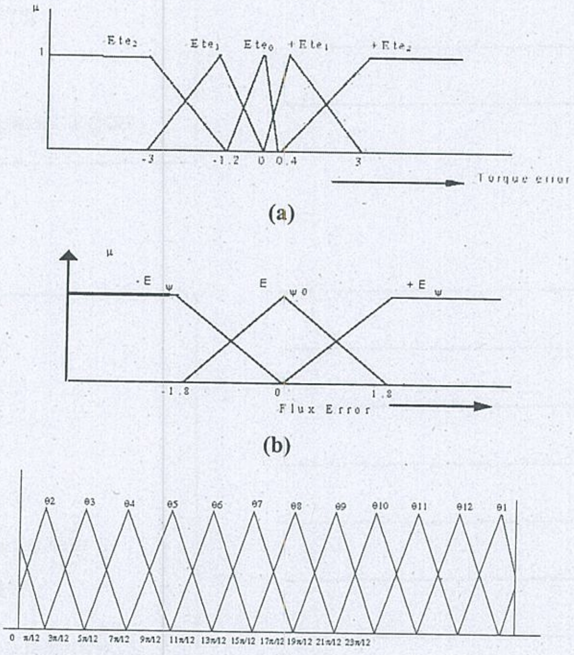


Figure 9 Schematic of fuzzy logic DTC

The schematic of fuzzy logic direct torque control scheme for induction motor drive is shown in Figure 9. The fuzzy output of torque, flux errors and stator flux angle are given as input variables to fuzzy controller and output variable obtained from the fuzzy controller is switching state of the inverter. Switching state of the inverter is a crisp value. The input variables membership functions are shown in Figure 10.



(a)

(b)

(c)

Figure 10 Membership distributions for input variables
(a) Torque Error (b) Flux Error and (c) Stator Flux angle

7. FUZZY RULES FOR DIRECT TORQUE CONTROL SCHEME

To improve the performance of classical DTC scheme, Fuzzy rules have been developed. In the Table 1, '1' represents the upper limb switches and '0' represents the lower limb switches of the inverter. Switching states of the inverter varies from V_0 to V_7. From this table it is concluded that, V_0=V_7 and which are null states. That is, V_0 and V_7 are zero vectors. The fuzzy system comprises 12 groups of rules and each of which contains 15 rules. Each group represents the respective stator flux angle θ. For example, rules are shown in Table 2 for stator flux angle θ_1, θ_2 and θ_3. For every combination of inputs and outputs, one rule can be applied. Totally, there are twelve-stator flux angles from θ_1 to θ_{12} and 180 rules are formed. With the help of them, corresponding switching state of the inverter is selected.

Table 1 Switching States of Voltage Vectors

States	**u_1**	**u_2**	**u_3**
V_0	0	0	0
V_1	1	0	0
V_2	1	1	0
V_3	0	1	0
V_4	0	1	1
V_5	0	0	1
V_6	1	0	1
V_7	1	1	1

Table 2 Fuzzy Rules Developed for Direct Torque Control Technique

	θ_1			θ_2			θ_3		
E_ψ Ete	P	Z	N	P	Z	N	P	Z	N
PL	V_1	V_2	V_2	V_2	V_2	V_3	V_2	V_3	V_3
PS	V_2	V_2	V_3	V_2	V_3	V_3	V_3	V_3	V_4
ZE	V_0	V_0	V_0	V_0	V_0	V_0	V_0	V_0	V_0
NL	V_6	V_0	V_4	V_6	V_0	V_5	V_1	V_0	V_5
NS	V_6	V_5	V_5	V_6	V_6	V_5	V_1	V_6	V_6

From the rules, fuzzy inference equations are given as

$$\alpha_i = \min(\mu A_i(E_\psi), \mu B_i(E_{te}), \mu C_i(\theta)) \tag{10}$$

$$\mu Ni'(n) = \min(\alpha i, \mu N_i(n)) \tag{11}$$

$$\mu N(n) = \max \sum_{i=1}^{180} \mu Ni'(n) \tag{12}$$

8. SIMULATION PROCEDURES

A 1-kW induction motor was used as a case study. The parameters of the machine were determined experimentally and are given in the Appendix. For the simulation of the viable torque control schemes, Voltage source inverter (VSI) was employed. The simulations were carried out using MATLAB / SIMULINK technical package described in [8].

8.1 DIRECT TORQUE NEURAL NETWORK CONTROLLER

The neural network is trained using the MATLAB neural-networks toolbox. This network consists of a three layer neural –network with three input nodes connected to five log sigmoid neurons and three pure output nodes connected to five log sigmoid neurons (3-5-5-3) shown in Figure 5. The training strategy consists the parallel recursive error prediction was chosen as a learning technique for simulation purposes to update the weights of the neural network. The algorithm was chosen because of its learning speed, robustness and high learning capability. This algorithm is so powerful when complicated and nonlinear functions are to be learned by the neural network [9]. The neural network structure mentioned previously was simulated using this algorithm and using the hyperbolic tangent function

$$S(x) = \tanh\left(\frac{1}{2}cx\right) = \frac{1-e^{-cx}}{1+e^{-cx}} \tag{13}$$

as the nonlinearity in the transfer functions of the hidden and output layers. The parameter *'c'* was fixed to one for all the cases. Small values of *'c'* are found to give larger weights and vice versa.

Simulation results were determined using an electromagnetic torque and stator flux commands of 2.5 Nm and 0.85 Weber respectively. The switching frequency of the inverter used by the simulations was 10 kHz while the frequency of the neural network was 100 Hz. The neural network frequency was chosen to give the plant enough time to stabilize its output. The data used to train the neural network have been determined by direct simulation of DTC using a sampling frequency of 100 Hz.

8.2. DTC USING GENETIC ALGORITHM

Neural network trained with genetic algorithm is implemented in such a way that the total number of thresholds and weights of the neural network be packed in n - dimensional vector 'w' as given in equation (14).

$$w = \left\lfloor th_1{}^2 m_{11}{}^2 \cdots m_{51}{}^2 \cdots th_5{}^1 m_{15}{}^1 \cdots m_{35}{}^1 \right\rfloor \tag{14}$$

where, *th* = threshold vector
m= weight vector
and n=38;

To represent the values of weights *w*, binary encoding or floating point encoding is used as a chromosome. Genetic operators used for binary representation are one point crossover, two-point crossover and bit mutation and for floating point representation are two point arithmetical crossover, uniform mutation, non-uniform mutation and non- uniform arithmetical mutation. Table 3 shows the parameters used for simulation:

Table 3 Parameters used for Genetic Algorithm based DTC

Parameters used	Binary representation	Floating point representation
Number of chromosomes	30	100
Crossover probability	0.8	0.9
Mutation probability	0.005	0.008

In binary encoding algorithm, Lower number of chromosomes was used than floating point encoding algorithm. The performance of the system is affected if number of chromosomes reduced. To improve the performance and to overcome this drawback, the best member of each generation must be copied into the succeeding generation. Crossover probability can be chosen from 0.5 to 0.9. Convergence rate becomes slower with the higher crossover probability values. Convergence rate should be in high bias level. Mutation rate taken for simulation as shown in Table II will make the convergence faster. In floating point-encoding algorithm, non-uniform mutation and non-uniform arithmetic mutation operators were introduced to prevent premature convergence. Fine tuning capabilities of genetic algorithm were achieved by using these operators and performance of the algorithm was also improved.

8.3. DIRECT TORQUE FUZZY LOGIC CONTROLLER

Direct torque control of induction motor using fuzzy logic was also simulated using the MATLAB / SIMULINK package. Membership functions were chosen and simulations were carried out. Only for three flux angle positions, rules were given in Table 2. Similarly, rules could be formed for another nine flux angle positions and totally for twelve positions, rules were written and membership functions were formed. Simulations include all the possible rules and total number of rules found is 180.

9. RESULTS AND DISCUSSION

Results obtained for viable torque and flux control techniques such as conventional DTC, DTC using neural networks, DTC using neural networks trained with genetic algorithm and DTC using fuzzy logic have been discussed. As described earlier, 1 kW induction motor was used for simulation and results were obtained. Switching frequency of the inverter taken for simulation was 10 KHz. There fore, the sampling time taken for simulation was 0.1 ms. Torque and flux reference values taken were 2.5 Nm and 0.5 wb when torque and flux hysteresis values are 0.5 Nm and 0.02 wb respectively. An index error has been used to quantify the error in both the stator flux and torque responses. This index is the integral of the square error (IE2), which is computed by means of the square error instead of just the error. Errors obtained in control schemes have been compared with each other. The error comparison is shown in Table 4.

Table 4 Errors obtained in various control strategies

Index Error (EI)		Classical DTC		DTC_NN		DTC_NN_GA		DTC_ Fuzzy	
$T=a*T_n$	$\omega=b*\omega_n$	Flux	Torque	Flux	Torque	Flux	Torque	Flux	Torque
a = 100%	b = 10%	2.53 e-3	0.189	2.2 e-3	0.165	1.97 e-3	0.156	2.74e-3	0.169
a = 50%	b = 50%	2.57 e-3	0.068	0.53 e-3	0.025	0.68 e-3	0.023	0.88 e-3	0.033
a = 10%	b = 10%	7.46 e-3	0.0367	1.58 e-3	0.0014	5.68 e-3	0.0015	0.14 e-3	0.00135
a = 100%	b = 100%	2.46 e-3	0.297	2.1 e-3	0.263	2.33 e-3	0.31	2.55 e-3	0.251

From the Table 4,
T = Actual torque;
T_n = Nominal torque = 5 Nm;
ω = Actual motor speed;
ω_n = Nominal motor speed = 1420 r. p. m.;
e-3 = $X10^{-3}$;

From the Table 4, it is realized that the index errors for flux and torque have been calculated for the different values of torque and speed in terms of their respective nominal values. Figure 11 shows the actual torque developed in induction motor using conventional DTC. Referring to the Figure, torque rises from 0 to 2.5 Nm in 10 ms and then oscillates around the reference value in a narrow band. Figure 12 shows the torque developed using artificial neural network. Figures 13 and 14 show the actual torque developed using DTC by neural network trained with genetic algorithm in which Figure 13 represents binary coding and Figure 14 represents the floating point coding representations. Stator phase currents of induction motor using genetic algorithm is shown in Figure 15. Figure 16 exhibits the step function of the developed torque in induction motor using neural network trained with genetic algorithm in binary coding representation. Figure 17 shows the locus of the stator flux and it is noticed that flux follows a circular shape. The components of stator fluxes in stationary reference frame are sinusoidal and 90° phase displacement to each other. Figures 18 and 19 are the torque developed by fuzzy controller and which have been compared with the conventional DTC technique.

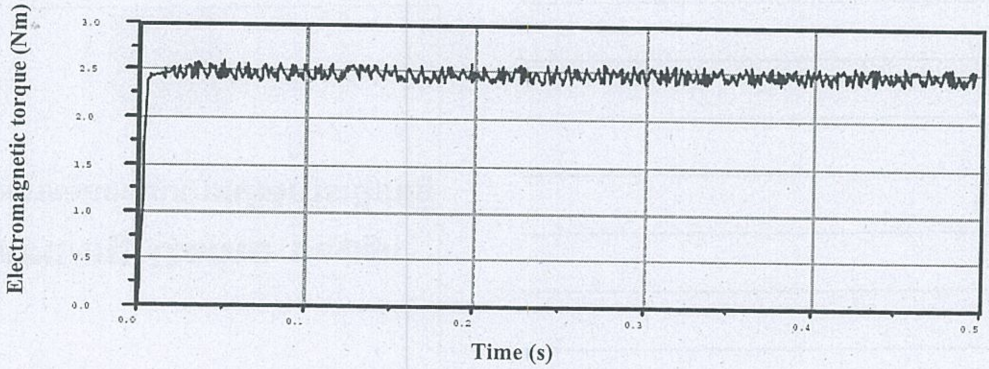


Figure 11 Torque developed in conventional DTC

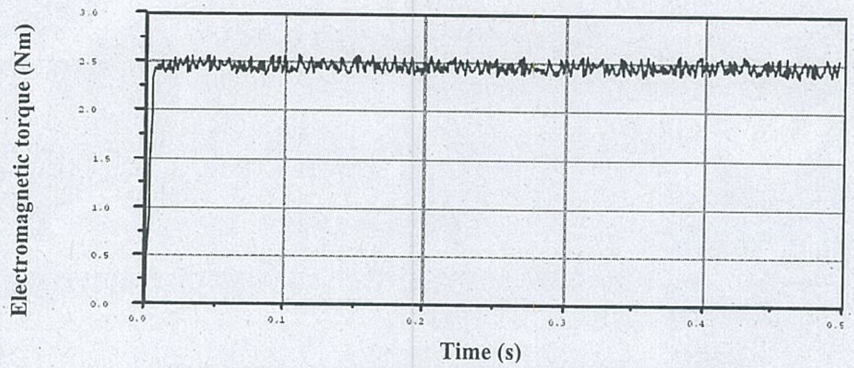


Figure 12 Torque developed in DTC using neural network

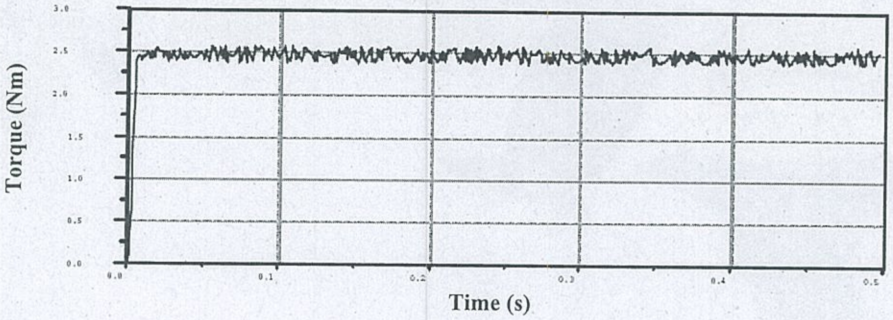


Figure 13 Torque developed in DTC using neural network trained with genetic algorithm (Binary coding representation)

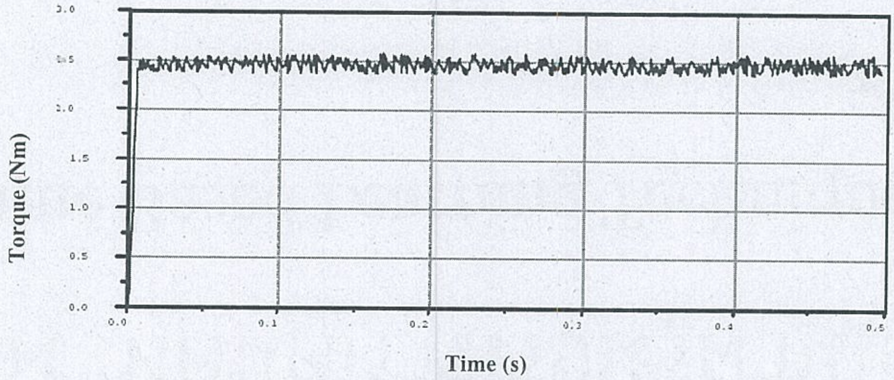


Figure 14 Torque developed in DTC using neural network trained with genetic algorithm (Floating-point coding representation)

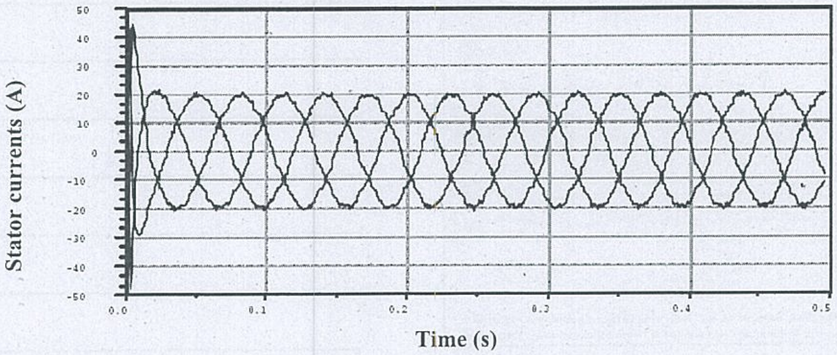


Figure 15 Stator phase currents

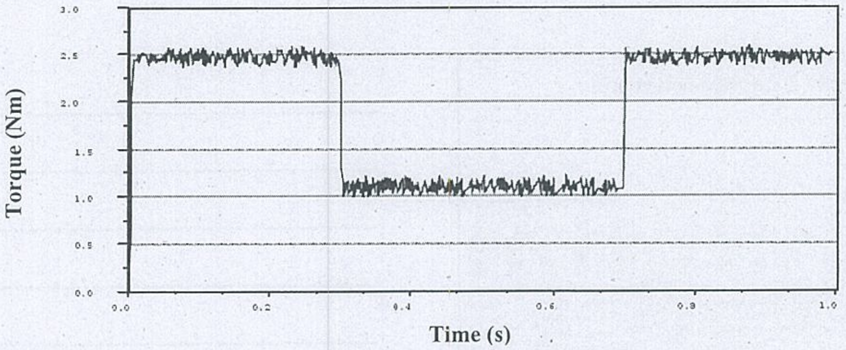


Figure 16 Torque step response using genetic algorithm (Binary coding representation)

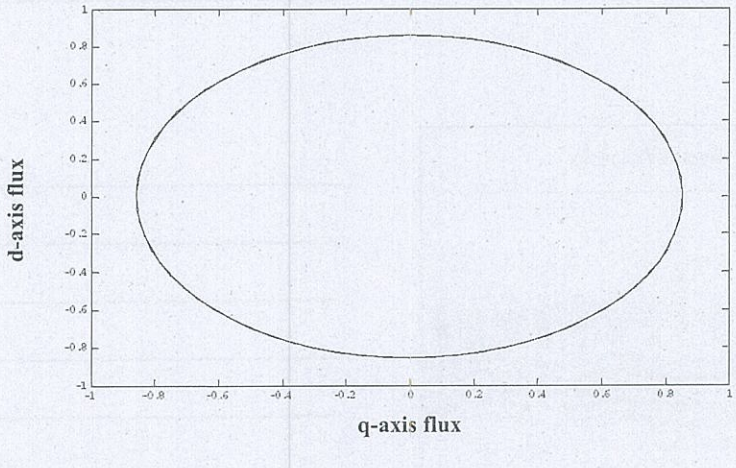


Figure 17 Locus of the stator fluxes in the stationary reference frame

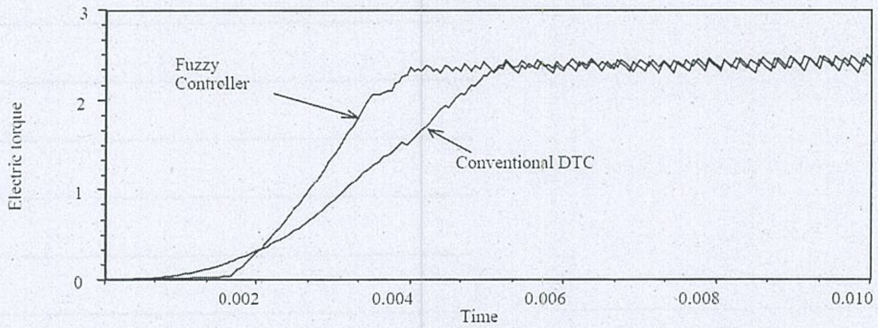


Figure 18 Torque developed in conventional DTC and DTC using fuzzy logic

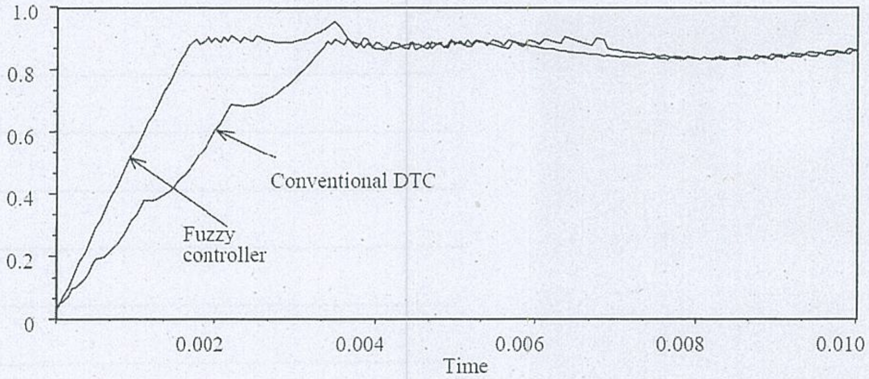


Figure 19 Initial stator flux rise

10. CONCLUSION

Three different intelligent torque control schemes, direct torque neuro controller, direct torque neuro controller trained with genetic algorithm and direct torque fuzzy controller have been evaluated for induction motor control and have been compared with the conventional direct torque control technique. Since the conventional DTC presents some disadvantages such as difficulties in torque and flux control at very low speed, high current and torque ripple, variable switching frequency behavior, high noise level at low speed and lack of direct current control, an adaptive torque controller must be proposed for high performance applications. In this study, three adaptive intelligent torque controllers have been proposed and results were compared. Each and every scheme has individual advantages and limitations.

APPENDIX

Rating	1kW
P	4
Rs	7.23 Ω
Rr	8.38 Ω
L_m	0.7014 H
L_s=L_r	0.0391 H
ω_{nom}	1420 r.p.m.
T_{nom}	6.7 Nm
J	0.006 kg.m^2

REFERENCES

[1] Vas, P. **1998** *Sensorless Vector and Direct Torque Control.* Oxford, U.K., Oxford Univ. Press.

[2] Buja, G., Casadei, D. and Serra, G. **2002** DTC–Based Strategies for Induction Motor Drives, *IEEE-IAS Annual Meeting*, 1506-1516.

[3] Habetler, T.G., Profumo, F., Pastorelli, M. and Tolbert, L.M. **1992** Direct Torque Control of Induction Machines Using Space Vector Modulation, *IEEE Transactions on Industry Applications*, *28*, 1045-1053.

[4] Griva, G. and Habetler, T.G. **1995** Performance Evaluation of a Direct Torque Controlled Drive in the Continuous PWM-Square Wave Transition Region, *IEEE Transactions on Power Electronics*, *10*, 464- 471.

[5] Habetler, T.G., Profumo, F. and Pastorelli, M. **1992** Direct Torque Control of Induction Machines Over a Wide Speed Range, *IEEE-IAS Annual Meeting,* Conf. Rec., 600-606

[6] Krose, B.J.A. and van der Smagt P.P. **1991** *An Introduction to Neural Networks*. The University of Amsterdam, The Netherlands.

[7] Grawbowski, P.Z., Kazmierkowski, M.P., Bose, B.K. and Blaabjerg, F. **2000** A Simple Direct-Torque Neuro Fuzzy Control of PWM- Inverter- Fed Induction Motor Drive,. *IEEE transactions on Industrial Electronics, 47*, 863-870.

[8] Narendra, K.S. and Parthasarathy, K. **1990** Identification and Control of Dynamical Systems Using Neural Networks, *IEEE Transactions on Neural Networks*, *1*, 4-27.

[9] "*The MATLAB compilers user's guide*" in Math works handbook Math works **1994**.

[10] William, J. **2001** *Introduction to MATLAB 6 for Engineers*. Palm III, McGraw–Hill International Edition.

[11] Bose, B.K. **1994** Expert System, Fuzzy Logic, and Neural Network Applications in Power Electronics and Motion Control, *Proceedings IEEE*, *82*, 1303-1323.

[12] Lascu, C., Boldea, I. and Blaabjerg, F. **2000** A Modified Direct Torque Control for Induction Motor Sensor less Drive, *IEEE Transactions on Industry Applications*, *36*(1), 122-130.

[13] Lee, J.-H., Kim, C.-G. and Youn, M.-J. **2002** A Dead-Beat Type Digital Controller for the Direct Torque Control of an Induction Motor, *IEEE Transactions on Power Electronics*, *17*(5), 739-746.