Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 7 (24): 4001-4006, 2007
ISSN 1812-5654

GA-Based Optimization of PI Speed Controller Coefficients for ANN-Modelled Vector Controlled Induction Motor

Seydi Vakkas Ustun
Department of Electric Education, Afyon Kocatepe University, Afyonkarahisar, Turkey

Abstract: This study proposes a new approach to indirect vector controlled induction motor drives. Induction motor drives for variable speed have many common industrial applications. An application of Artificial Neural Network (ANN) and genetic algorithms on vector control are carried out using space vector pulse width modulation in this study. Proportional plus derivative (PI) controller is used to control speed of induction motor. In this study, optimization of PI coefficients in vector control is carried out by ANN-Genetic. These controllers are applied to drive system with 0.55 kW induction motor. A Digital Signal Processor Controller (dsPIC30F6010) was used to carry out control applications. It is suitable to control induction motor as a soft starter and speed adjustment in compressors, blowers, fans, pumps and many other applications.

Key words: ANN, genetic, indirect vector control, DSP

INTRODUCTION

Vector control is a control strategy to decouple flux and torque from an induction motor in order to emulate a DC motor. The great advantage is that it can be controlled as easy as a DC motor and induction one with all of its advantages such as high efficiency, robustness, no maintenance and low cost. Of course there are many industrial applications for an induction motor with variable speed for which this control had become so important (Betin and Depernet, 1997).

The motor control issues are traditionally handled by fixed gain Proportional Integral (PI) and Proportional Integral Derivative (PID) controllers. However, the fixed gain controllers are very sensitive to parameter variations, load disturbances, etc., So, the controller parameters have to be continually adapted. The problem can be solved by several adaptive control techniques such as model reference adaptive control (Sugimoto and Tamai, 1987), sliding mode control, variable structure control and self tuning PI controllers, etc. The design of all of the above controllers depends on the exact system mathematical model. However, it is often difficult to develop an accurate system mathematical model due to unknown load variation, unknown and unavoidable parameter variations due to saturation, temperature variations and system disturbances (Uddin *et al*., 2002).

In high performance applications, it is useful automatically extract the complex relations that represent the drive behavior. The use of learning through example algorithms can be a powerful toll for automatic modeling variable speed drives (Maia *et al*., 1994). They can automatically extract a functional relationship representative of the drive behavior. These methods present some advantages over the classical ones since they do not rely on the precise knowledge of mathematical models and parameters. On the other hand, electromechanical systems usually present internal nonlinearities and parameter deviation, which are difficult to model (Cardoso *et al*., 1998).

PI controller is unquestionably the most commonly used control algorithm the process control industry. The main reason is its relatively simple structure, which can be easily understood and implemented in practice and that many sophisticated control strategies, such as model predictive control, are based on it. In spite of its wide spread use there exists no generally accepted design method for the controller (Wang and Shao, 2000).

Most industrial processes exhibit nonlinear dynamics and this places additional complexity on the modeling procedure used. In practice, many nonlinear processes are approximated by reduced order models, possibly linear, which are clearly related to the underlying process characteristics. However, these models may only be valid within certain specific operating ranges. When operating conditions change, a different model may be required to be used or the model parameters may need to be adapted.

System model is necessary for tuning controller coefficients in an appropriate manner (e.g., percent overshoot, settling time). Because of neglecting some parameters, the mathematical model cannot represent the physical system exactly in most applications. That's why, controller coefficients cannot be tuned appropriately.

The tuning of electric drive controller is a complex problem due to the many non-linearities of the machines, power converter and controller. Therefore, the whole system model can be obtained by using the ANN structure. Optimization process of PI speed coefficients in the vector control is determined by using genetic algorithms. The ANN architecture is explained the next section.

Neural network: Artificial Neural Networks (ANNs) are successfully used in a lot of areas such as control, early detection of electrical machine faults and digital signal processing in everyday technology. The memory of a neural network lies in the weights and biases. The neural networks can be classified, in terms of how the weights and biases are obtained, into three categories.

Multi-Layer Perceptrons (MLPs) are the simplest and therefore most commonly used neural network architectures. The back propagation algorithm is the most commonly adopted MLP training algorithm. This type of neural network is known as a supervised network, because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data, so that the model can then be used to produce the output when the desired output is unknown. The ANN model structure of the system is shown in Fig. 1, where f, K_p and K_i are fitness function, PI coefficients, respectively. The ANN parameters for the model system are shown in Table 1.

There was no criterion to select cell number at every layer of the ANN structure; layer number and cell number were determined with experiment. In the same way, the learning and momentum coefficients were determined by experiences at previous studies.

Vector control: The main objective of the vector control of induction motors is, as in DC machines, to independently control the torque and the flux; this is done by using a d-q rotating reference frame synchronously with the rotor flux space vector (Lorenz and Lawson, 1990) as shown in Fig. 2, the d axis is aligned with the rotor flux space vector. Under this condition,

$$\psi^*_{rq} = 0 \text{ and } \psi^*_{rd} = \psi^*_r$$

For the ideal state decoupling the torque equation become analogous to the dc machine as follows:

$$T_e = \frac{3}{2}\frac{P.L_m\psi_r}{L_r}$$

Figure 2 shows the implemented diagram of an induction motor indirect field-oriented control (Ross and Theys, 2005)

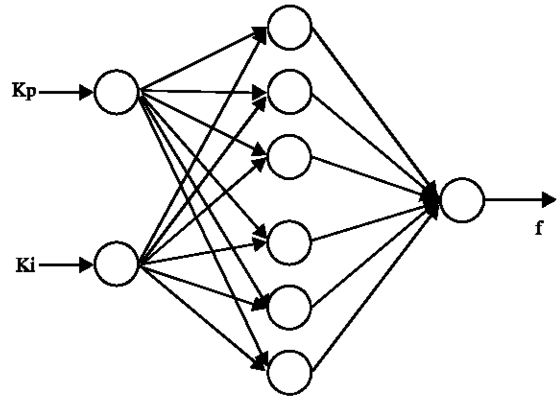

Fig. 1: The ANN model structure of the system

Table 1: The ANN parameters for the model system

Parameters	Value
Number of neurons for input layer	2
Number of neurons of the output layer	1
Layer number	1
First layer cell number	6
Second layer cell number	-
First layer activation function	Sigmoid
Second layer activation function	Sigmoid
Maximum iteration number	30000
Error limit	0.0001
Training coefficient	0.7
Momentum coefficient	0.9

In an asynchronous squirrel cage induction motor the mechanical speed of the rotor is slightly less than the rotating flux field. The difference in angular speed is called slip and is represented as a fraction of the rotating flux speed. Park and Inverse Transforms require an input angle θ. The variable θ represents the angular position of the rotor flux vector. The correct angular position of the rotor flux vector must be estimated based on known values and motor parameters. This estimation uses a motor equivalent circuit model. The slip required to operate the motor is accounted for in the flux estimator equations and is included in the calculated angle. The flux estimator calculates a new flux position based on stator currents, the rotor velocity and the rotor electrical time constant. In this study, this implementation of the flux estimation is based on the motor current model and in particular these three equations (Ross and Theys, 2005).

Magnetizing current;

$$I_{mr} = I_{mr} + \frac{T}{T_r}(I_d - I_{mr}) \tag{1}$$

Flux speed;

$$f_s = (p_{pr}.n) + \left(\frac{1}{T_r w_b}\frac{I_q}{I_{mr}}\right) \tag{2}$$

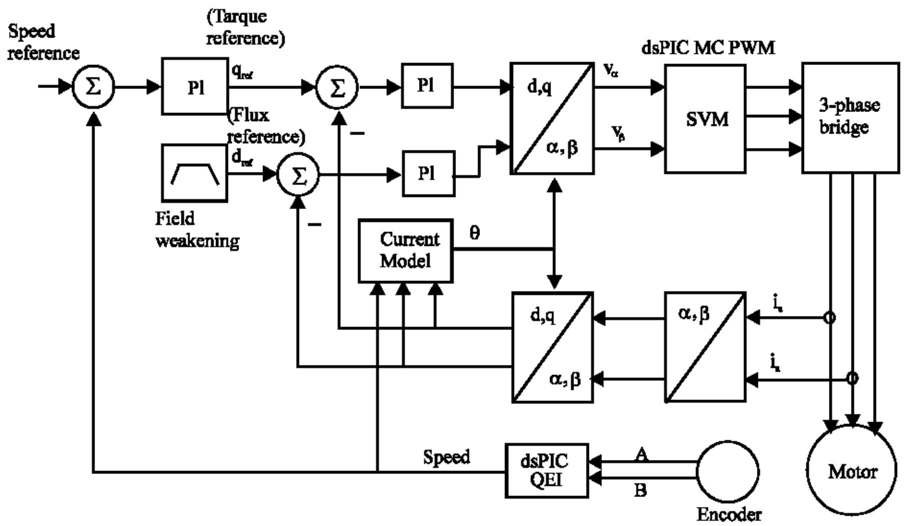

Fig. 2: The overall vector-controlled induction motor drive system

Flux angle;

$$\theta = \theta + w_b.f_s.T \tag{3}$$

Where:

I_{mr} = Magnetizing current (as calculated from measured values)

f_s = Flux speed (as calculated from measured values)

T = Sample (loop) time (parameter in program)

n = Rotor speed (measured with the shaft encoder)

($T_r = L_r/R_r$) = Rotor time constant (must be obtained from the motor manufacturer)

θ = Rotor flux position (output variable from this module)

ω_b = Electrical nominal flux speed (from motor name plate)

P_{pr} = Number of pole pairs (from motor name plate)

During steady state conditions, the I_d current component is responsible for generating the rotor flux. For transient changes, there is a low-pass filtered relationship between the measured I_d current component and the rotor flux. The magnetizing current, I_{mr}, is the component of I_d that is responsible for producing the rotor flux. Under steady-state conditions, I_d is equal to I_{mr}. Eq. 1 relates I_d and I_{mr}. This equation is dependent upon accurate knowledge of the rotor electrical time constant. Essentially, Eq. 1 corrects the flux producing component of I_d during transient changes.

The computed I_{mr} value is then used to compute the slip frequency, as shown in Eq. 2. The slip frequency is a function of the rotor electrical time constant, I_q, I_{mr} and the current rotor velocity. Equation 3 is the final equation of the flux estimator. It calculates the new flux angle based on the slip frequency calculated in Eq. 2 and the previously calculated flux angle. If the slip frequency and stator currents have been related by Eq. 1 and 2, then motor flux and torque have been specified. Furthermore, these two equations ensure that the stator currents are properly oriented to the rotor flux. If proper orientation of the stator currents and rotor flux is maintained, then flux and torque can be controlled independently. The I_d current component controls rotor flux and the I_q current component controls motor torque. This is the key principle of indirect vector control (Ross and Theys, 2005).

Experimental setup: The experimental setup consisted of a motor and generator that was connected to it by a connecting element. The motor used was a 0.55 kW, 1.34A, 50 Hz, Cosφ = 0.84, three phase squirrel-cage induction motor. The processor used in this work was a 7.38 MHZ dsPIC30F6010 Digital Signal Processor Controller (DSP Controller). The processor communicated with the PC via USB port. The block diagram of this application circuit is shown in Fig. 3.

Required values for PWM output of the DSP controller are calculated by using the vector control method. PWM time base is 100 microsecond for this application. The control loop is carried out once during each 40 PWM time base. Dead time is formed by the controller. The value of dead time determined by a register

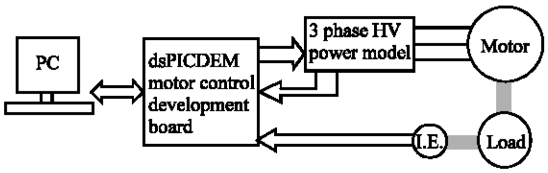

Fig. 3: The block diagram of the application circuit

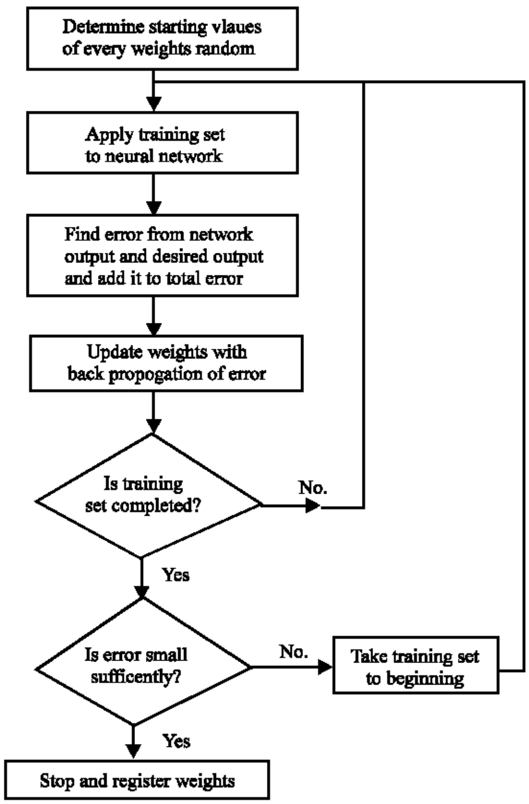

Fig. 4: The flow chart for the training process

is taken 4 microseconds. The DSP controller program for the control process was written in dsPIC30F6010 assembly language and C30 language. Controlling and compiling process were performed by a compiler program.

Modelling of the induction motor using ANN: The ANN model used is a multi-layer perceptron model, in which there is more than one layer between input and output. The backpropogation of the error algorithm used as the training algorithm is used for training of generalized delta rule. The training process of this ANN model is shown Fig. 4.

Thirty sets of input-output data taken from the application circuit are given in Table 1. The coefficients of the ANN are trained using data in Table 2. Change in the error in training process is shown in Fig. 5.

Table 2: Data used for the ANN

Data set	Kp	Ki	f = 1/(1+Mo(rpm) +2*Ts(ms))
1	6.500	1.100	0.477
2	6.500	0.975	0.488
3	6.500	0.725	0.348
4	6.500	0.225	0.683
5	6.062	1.100	0.488
6	6.062	0.975	0.454
....	....	....	....
....	....	....	....
25	3.437	1.100	0.391
26	3.437	0.850	0.349
27	3.437	0.600	0.402
28	3.437	0.350	0.429
29	3.437	0.108	0.511
30	4.577	0.225	0.370

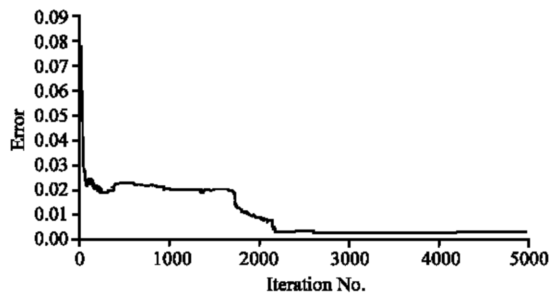

Fig. 5: The error values according to iteration number

As shown in Fig. 5, the error values reduce acceptable values when iteration number is 3000. Therefore, the training process was finished at 3000 iterations. Then, the best Kp and Ki pairs are obtained by using genetic algorithm program.

Optimization of PI coefficients using GA: GAs are based on an analogy to the genetic code in our own DNA (deoxyribonucleic acid) structure, where its coded chromosome is composed of many genes (Goldberg, 1989). GA approach involves a population of individuals represented by strings of characters or digits. Each string is, however, coded with a search point in the hyper search-space. From the evolutionary theory, only the most suited individuals in the population are likely to survive and generate off-spring that passes their genetic material to the next generation.

The GA used in this paper known as the simple genetic algorithm. In the algorithm, the three-operator GA with only minor deviations from the original is used (Dimeo and Lee, 1995).

Different crossover and mutation rates are used for processing of optimization of genetic algorithms. Ten of the fitness values obtained, listed from the largest fitness value to the smallest and the fitness values of the members of the first generation are shown in Table 3. The flow chart of the GA is shown in Fig. 6 (Ustun and Demirtas, 2005).

A PI controller with the transfer function

$$G_c(s) = K_p + \frac{K_i}{s}$$

is employed to control the process. The optimum values for the Kp and Ki pairs were obtained using a computer program written in C++ language for the GA. This process executes with three different operators at bit level. Thirty of the Kp and Ki pairs were determined at random. Kp and Ki pairs consisted of 15 bits and 15 bits, respectively. These Kp and Ki pairs were entered to ANFIS model as input. The fitness values were obtained from the ANFIS outputs. These values were then used as the fitness function.

Table 3: Fitness values of the members for ANN method and GA parameters in the first generation

Parameters	Values
Population size	30.000
Crossover operator	0.600
Mutation size	0.100
Fitness of member 1	2.049
Fitness of member 2	2.049
Fitness of member 3	2.049
Fitness of member 4	1.944
Fitness of member 5	1.944
Fitness of member 6	1.944
Fitness of member 7	1.944
Fitness of member 8	1.944
Fitness of member 9	1.793
Fitness of member 10	1.793

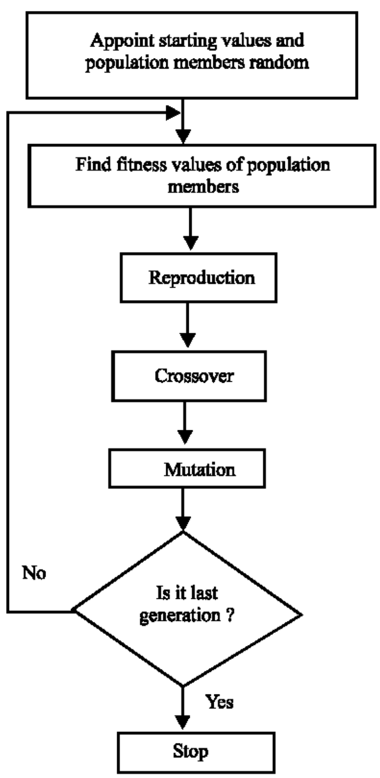

Fig. 6: The flow chart of the GA

The one-point crossover method was used on the crossover operator. Mutual parameters of two random members on the crossover were divided into two parts and their positions were changed. A random bit of a random number on the mutation process was changed 0 to 1 and 1 to 0. For the optimization process, mutation rate is increased when converge occurs in 5-10 generation. Therefore, early converge is prevented and in addition, members that have high fitness values were obtained.

The range of Kp and Ki values chosen lay between (3-6.5) and (0.1-1.1), respectively. The fitness function is defined as:

$$f = \frac{1}{M_o + 2 * T_s + 1}$$

RESULTS AND DISCUSSION

A model-based control structures are suggested that include the ANN dynamics model of the system, in this work. The ANN is systematically constructed from the input-output data.

The controller is applied to the system when the motor speed is about 1000 rpm. That is, the speed is increased from 1000 to 2000 rpm by using the proposed controllers. The system is worked to 1000 rpm as open-loop control.

The ANN model follows the system output, with a small error that arises from differences between experimental conditions and the model of the non-linear system. It shows that the ANN model created for the system models the system successfully. System outputs are demonstrated for different Kp and Ki pairs in Fig. 7 and 8. Those Kp and Ki pairs are random selected.

The optimum PI coefficients were found by using ANN-Genetic method. The optimal Kp and Ki pairs for ANN-Genetic method are found as Kp = 6.47 and Ki = 0.1. The speed of the rotor for these values is shown in Fig. 9.

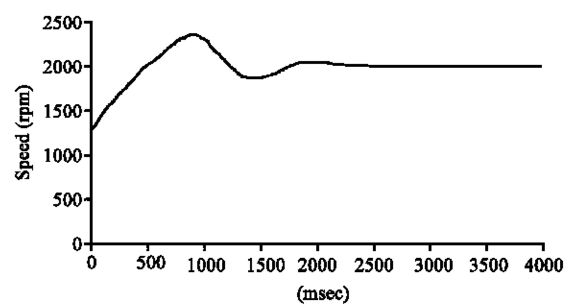

Fig. 7: Speed of rotor for Kp = 3.44 and Ki = 1.1

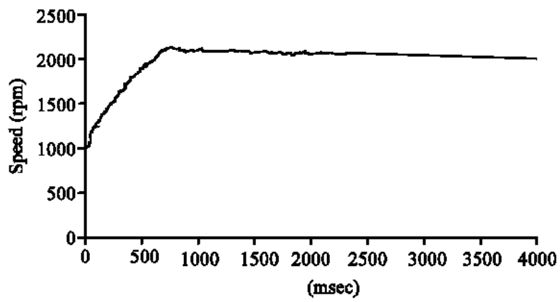

Fig. 8: Speed of rotor for Kp = 6.06 and Ki = 0.47

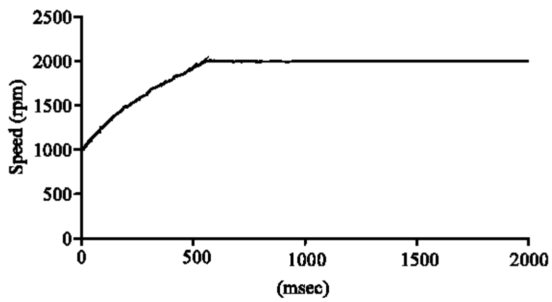

Fig. 9: The speed response for optimal Kp and Ki (Kp = 5.069-Ki = 0.132)

The settling time is shorter and the maximum overshoot is minimized for these values. The Results of the ANN-Genetic method show that this method is a good control system.

CONCLUSIONS

This study describes the ANN-Genetic method. Actual system (motor and controller) could be modelled using ANN structure. It was also determined that the maximum overshoot and settling time are very small if the system is controlled by control parameters obtained from the optimization process which uses GA.

The results presented show that the ANN is able to produce accurate dynamic models of process response directly from I/O data. GA is suitable for optimization of controller coefficients by the performance criteria considered. This process can be also applied for nonlinear systems controlled by PD and PID controller, or a number of applications such a machine tool, robotics and servo drives.

ACKNOWLEDGMENT

The project is supported by TUBITAK (The Scientific and Technological Research Council of Turkey). The authors would like to acknowledge the financial support provided by TUBITAK.

REFERENCES

Betin, F. and D. Depernet, 1997. Control of induction machine drive using fuzzy logic approach: Robustness study. IEEE, 21 (3): 361-366.

Cardoso, F.D.S., J.F. Martins and V.F. Pires, 1998. A comparative study of a PI, neural network and fuzzy genetic approach controllers for an AC-drive. IEEE AMC COIMBRA, pp: 375-380.

Dimeo, R. and K.Y. Lee, 1995. The use of a genetic algorithm in power plant control system design. IEEE Proceeding of the 34th Conference on Decision and Control, pp: 737-742.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-wesley, Reading, MA.

Lorenz, R.D. and D.B. Lawson, 1990. A simplified approach to continuous on-line tuning of field oriented induction machine drives. IEEE Trans. Ind. Applic., 26 (3): 444-449.

Maia, J.H., P.J. Branco and J.A. Dente, 1994. Automatic modeling of electrical drives. Proceedings of Modern Electrical Drives in NATO Advanced Study Institute, pp: 73-78.

Ross, D. and J. Theys, 2005. Using the dsPIC30F for Vector Control of an ACIM, Microchip Technology Inc. Application Notes.

Sugimoto, H. and S. Tamai, 1987. Secondary resistance identification of an induction motor applied model reference adaptive system and its characteristics. IEEE Trans. Ind. Applic., 23 (2): 296-303.

Uddin, M.N., T.S. Radwan and M.A. Rahman, 2002. Performances of novel fuzzy logic based indirect vector control for induction motor drive. IEEE Trans. Ind. Applic., 38 (5): 1219-1225.

Ustun, S.V. and M. Demirtas, 2005. Optimal tuning of PI speed controller coefficients for electric drives using neural network and genetic algorithms. Electrical Eng., 87 (2): 77-82.

Wang, Y. and H. Shao, 2000. Optimal tuning for PI controller. Automatica, 36: 147-152.