Expert Systems with Applications 36 (2009) 5533–5540

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

DSP-based sliding mode speed control of induction motor using neuro-genetic structure

Metin Demirtas *

Electrical & Electronics Engineering Department, Balikesir University, 10100 Balikesir, Turkey

ARTICLE INFO

Keywords:
Sliding surface
Boundary layer
Genetic
Neural
AC motor

ABSTRACT

In this paper, a novel approach is presented for optimization of sliding mode controller parameters. The main purpose is to optimize sliding surface slope and thickness of the boundary layer.

The tuning of the electrical drive controller is a complex problem due to the many non-linearities of the machines, power converter and controller. Therefore, it is difficult to develop mathematical models of the system accurately because of unknown and unavoidable parameter variations due to saturation temperature variations and system disturbance. To solve that problem artificial neural network (ANN) is used. That is, the whole system is modeled by using ANN. Then, sliding surface slope and thickness of the boundary layer is optimized using genetic algorithms. The proposed method is applied to an induction motor. Experimental results verify that the proposed control approach is very good for complex and non-linear systems.

1. Introduction

With the well-known merits of reliability, simple construction and low weight, AC induction motors (IMs) have been gradually utilized in place of DC motors with drawbacks of spark, corrosion and necessity of maintenance (Leonhard, 1996). Moreover, because of the advances in power electronics and microprocessors, IM drives used in variable speed and position control have become more attractive in industrial processes such as robot manipulators, factory automations and transportation applications. However, IMs have more complexities in control characteristics than DC motors due to their coupled and nonlinear time-varying dynamics. In the past years, many techniques for the control of IMs have been investigated (Benchaib, Rachid, & Audrezet, 1999; Ho & Sen, 1988; Jansen, Lorenz, & Novotny, 1994; Leonhard, 1996). Among them, the field-oriented control is the most popular one. With the technique of field orientation, the rotor speed is asymptotically decoupled from rotor flux, and the speed is linearly related to torque current. Thus, the IM possesses the same behavior of a separately excited DC motor (Leonhard, 1996). In general, the field-oriented control performance is sensitive to the deviation of motor parameters, particularly the rotor time-constant (Moreira & Lipo, 1993; Schauder, 1992). To deal with this problem, there are many flux measurement and estimation mechanisms in the published literature (Jeon, Oh, & Choi, 2002; Moreira & Lipo, 1993; Schauder, 1992).

Indirect field-oriented techniques are now widely used for the control of IM in high-performance applications. With this control strategy, the decoupled control of IM is guaranteed, and can be controlled and provide the same performance as achieved from a separately excited DC machine. However, the control performances of the resulted linear system are still influenced by the uncertainties, which usually are composed of unpredictable parameter variations, external load disturbances. Therefore, in order to solve some of the problems of field-oriented control, the motor drive must be techniques that are appropriate to discontinuous operation of the switching devices and allow the robustness of the algorithm, with regard to changing parameters and external disturbances. This common drawback can be overcome by using variable structure control (VSC) (Wai, Lin, & Lin, 2004). The essential property of VSC is that the discontinuous feedback control switches on one or more manifolds in the state space. Ideally, the switching of control occurs at infinitely high frequency to eliminate deviations from sliding manifolds. In practice, the frequency is not infinitely high due to the finite switching time and with effects of unmodeled dynamics, cause undesired chattering of the control (Sarwer, Rafiq, Data, Ghosh, & Komada, 2005).

The sliding mode control (SMC) can offer good properties, such as insensitivity to parameter variations, external disturbance rejection, and fast dynamics response. However, in SMC, the high frequency chattering phenomenon that results from the discontinuous control action is a severe problem when the state of the system is close to the sliding surface (Wai et al., 2004).

SMC is one of the effective nonlinear robust control approaches since it provides system dynamics with an invariant property to

* Tel.: +90 266 6121194; fax: +90 266 6121257.
E-mail address: mdtas@balikesir.edu.tr

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.06.086

uncertainties once the system dynamics are controlled in the sliding mode (Slotine & Li, 1991). In recent years, there are many SMC systems for AC servo drives in the opening articles (Lin, Chiu, & Shyu, 1998; Park & Lee, 1998; Shiau & Lin, 2001; Utkin, 1993). However, the insensitivity of the controlled system to uncertainties exists only in the sliding mode, but not during the reaching phase (Wai, 2000). That is, the system robustness can not be maintained in the whole control process. To overcome this problem, one effective way is to speed up the period of reaching phase via a larger control gain. However, it will result in excessive chattering control efforts and may excite high-frequency unstable dynamics. Though the concept of a boundary layer can be introduced to improve this phenomenon, the control accuracy will be reduced and the stability within the boundary layer can not be guaranteed. To keep robustness in the whole sliding-mode control system, several researchers have focused on eliminating the effect of the reaching phase (Bartolini, Ferrara, Usai, & Utkin, 2001; Slotine & Li, 1991).

The most common approach to alleviate the effect of chattering is the so-called "boundary layer" control (Chang, Twu, & Chang, 1992; Zhang & Barton, 1991), in which the sign function is replaces by some smooth approximation when the state trajectory lies within a suitable boundary layer of the switching surface. Unfortunately as pointed out in Young, Utkin, and Ozguner (1999), this method solves the problem only verifies that the proposed control scheme has the advantage of partially, because, within the boundary layer, the system no longer behaves as a variable structure system (Sarwer et al., 2005). If the boundary layer chosen is very small then under some operating conditions chattering may re-occur. Alternatively, if the boundary layer chosen is large then chattering is completely eliminated. However, a large boundary layer results in slow system response and therefore degrades the dynamic performance of the system (Zhang & Barton, 1991).

It is a basic fact that the system performance is sensitive to the sliding surface slope C for SMC application. For instance, if large values of C are considered then the system will give a fast response in SMC application due to the large values of the control signal but the system may become unstable. Conversely, if small values of C are chosen the system will be more stable but the performance of the system may degrade since the system response will become slower due to small values of the control signal (Eksin et al., 2002).

Thus, determination of an optimum C value and thickness of the boundary layer for a system is an important problem. These problems may be solved by determining the optimum sliding surface slope and optimum boundary layer width.

System model is necessary for tuning controller coefficients in an appropriate manner (e.g. percent overshoot, settling time). But, in most applications the mathematical model cannot represent the physical system exactly because of neglecting some parameters. Therefore, the controller coefficients cannot be tuned appropriately. In recent years, the use of artificial neural networks (ANNs) for identification and control of nonlinear dynamic systems in power electronics and ac drives have been proposed (Burton, Harley, Diana, & Rodgerson, 1998; Mondal, Pinto, & Bose, 2002), as they are capable of approximating wide range of nonlinear functions to a high degree of accuracy (Karanayil, Rahman, & Grantham, 2007). The 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.

Genetic Algorithm (GA) is used for optimization of sliding surface slope and boundary layer width. GAs is 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; NG et al., 1995). 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 & Lee, 1995).

In this way, the performance of the overall system using the proposed method is improved with respect to the classical SMC.

This paper is organized as follows. In Section 2, Vector controlled IM is described. Section 3 details ANN and GA. In Section 4, sliding mode control of IM is reported. The experimental setup is shown in Section 5. Modeling and optimization process is explained in Section 6. Results and discussion are submitted in Section 7. The conclusion is given the last section.

2. Vector controlled induction motor

The main objective of the vector control of IM 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 & Lawson, 1988) as shown in Fig. 1, the d axis is aligned with the rotor flux space vector. Under this condition,

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

In an asynchronous squirrel cage IM 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 can be written as follows:

Magnetizing current;

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

Flux speed;

$$f_{\mathrm{s}} = (n_p \cdot n) + \left(\frac{1}{T_{\mathrm{r}} w_b}\frac{I_q}{I_{\mathrm{mr}}}\right) \tag{2}$$

Flux angle;

$$\theta = \theta + w_b \cdot f_{\mathrm{s}} \cdot T \tag{3}$$

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}: 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); n_{p}: 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

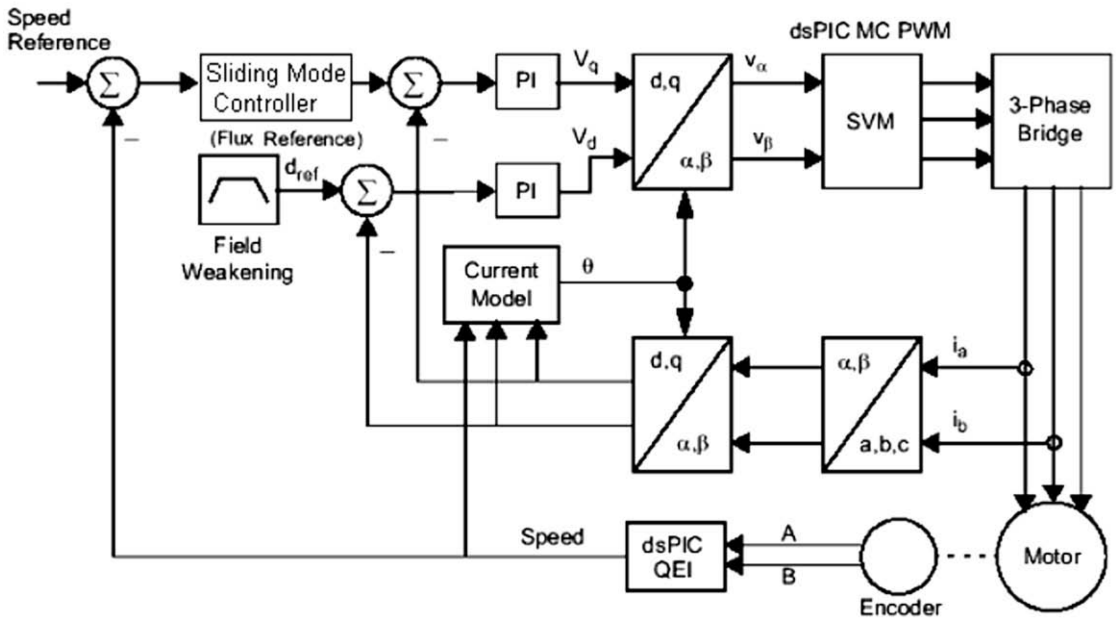

Fig. 1. The overall vector-controlled induction motor drive system.

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. Eq. (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 Eq. (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.

In general, the mechanical equation of an IM can be represented as

$$J\dot{\omega}_r(t) + B\omega_r(t) + T_l = T_e \tag{4}$$

where ω_r is the rotor angular speed; J is the total mechanical moment inertia constant; B is the total damping coefficient; T_l is the torque of external load disturbance; T_e denotes the electromagnetic torque. With the implementation of field-oriented control (Leonhard, 1996), the electromagnetic torque can be simplified as

$$T_e = K_t i^*_{qs} \tag{5}$$

With the torque constant K_t is defined

$$K_t = (3\,n_p/2)(L_m^2/L_r)\,i^*_{ds}$$

where n_p is the number of pole pairs; L_m is the magnetizing inductance per phase; L_r is the rotor inductance per phase referred to stator; i^*_{ds} and i^*_{qs} denote the flux and torque current commands. Substituting Eq. (5) into Eq. (4), the mechanical dynamic of the IM drive system can be represented as

$$\dot{\omega}_r(t) = -\frac{B}{J}\omega_r(t) + \frac{K_t}{J} i^*_{qs} - \frac{T_l}{J} \tag{6}$$

or

$$\dot{x}(t) = Ax(t) + Bu(t) + CT_l \tag{7}$$

where $x(t) = \omega_r(t), A = -B/J,\ C = -1/J, u(t) = i^*_{qs}$ is the control effort. Though the dynamic behavior of the IM is like that of a separately excited DC motor, the control performance of the IM is sill influenced seriously by the system uncertainties including electrical and mechanical parameter variation, external load disturbance, non-ideal field-oriented transient responses and unmodeled dynamics in practical applications (Karanayil et al., 2007).

3. ANN and GA

Multi-layer perceptrons (MLPs) are the simplest and therefore most commonly used neural network architectures. The backpropagation algorithm is the most commonly adopted MLP training algorithm. The backpropagation neural network is the most popular feedforward predictive network deployed in process industries. The backpropagation network assumes that all processing elements and connections are somewhat responsible for the difference between the expected output and the actual output (GARCIA, 2007). 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. The model can be used to produce the output when the desired output is unknown. The ANN model structure of the system is shown in Fig. 2, where f, C, and ϕ are fitness function and C–ϕ coefficients, respectively.

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.

GAs are search algorithms that use operations found in natural genetic to guide through a search space (NG et al., 1995). GAs use a direct analogy of behavior. They work with a population of chromosomes, each one representing a possible solution to a given

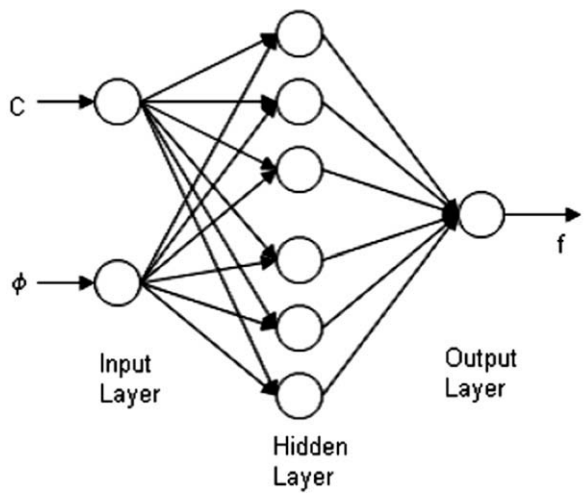

Fig. 2. The ANN model structure of the system.

problem. Each chromosome has assigned a fitness score according to how good solution to the problem it is. GA is theoretically and empirically proven to provide robust search in complex spaces, giving a valid approach to problem requiring efficient and effective searching (Cordon & Herrera, 1995; Velasco & Magedalena, 1995).

Any GA starts with a population of randomly generated solutions, chromosomes, and advances toward better solutions by applying genetic operators, modeled on the genetic processes occurring in nature. In each generation, relatively good solutions reproduce to give offspring that replace the relatively bad solutions which die. An evaluation or fitness function plays the role of the environment to distinguish between good and bad solutions. The process of going from the current population to the next population constitutes in the execution of GA. Although there are many possible variants of simple GA, the fundamental underlying mechanism operates on a population of chromosomes and consists of three operations:

- Evaluation of individual fitness,
- Formation of gene pool (intermediate population)
- Recombination and mutation.

The next procedure shows the structure of a simple GA (Hazzab, Khalil Bousserhane, & Kamli, 2004).

Structure of standard genetic algorithm

- Begin (1)
- t = 1
- Initialize Population(t)
- Evaluate fitness Population(t)
- While (Generations < Total Number) do
- Begin (2)
- Select Population(t+1) out of Population(t)
- Apply Crossover on Population(t+1)
- Apply Mutation on Population(t+1)
- Evaluate fitness Population(t+1)
- t = t + 1
- End (2)
- End (1)

A fitness function must be devised for each problem to be solved. Given a particular chromosome, a solution, the fitness function returns a single numerical fitness, which is supposed to be proportional to the utility or adaptation of the individual which that chromosome represents.

In GAs, a crossover operator combines the features of two parent structures to form two similar offspring. It is applied with a probability of performance, the crossover probability. A mutation operator arbitrary alters one or more components of a selected structure so as to increase the structural variability of the population. Each position of each solution vector in the population undergoes a random change according to a probability defined by a mutation rate, the mutation probability (Hazzab et al., 2004).

4. Sliding mode control of induction motor

Let us consider that a class of nonlinear system is defined as

$$\dot{x}(t) = f(x) + b(x)u(t) \tag{8}$$

Here $x(t) = [x, \dot{x}, x, \dots, \ddot{x}^{(n-1)}]^{\mathrm{T}}$ is the state vector of the system, $u(t)$ are the control inputs, $b(x)$ is an $(n-1)x1$ unknown control vector function and $f(x)$ is an $(n-1)x1$ unknown nonlinear dynamic vector function.

The desired state vector can be defined as $x_d(t) = [x_d, \dot{x}_d, \ddot{x}_d, \dots, x_d^{(n-1)}]^{\mathrm{T}}$. The tracking error is defined as $e(t) = x_d(t) - x(t)$, and the tracking error vector be defined as

$$e(t) = x_d(t) - x(t) = [e, \dot{e}, \ddot{e}, \dots, e^{(n-1)}]^{\mathrm{T}} \tag{9}$$

The design of SMC involves two tasks. The first one is to select the sliding surface $S(t)$ for prescribing the desired dynamic characteristics of the controlled system. The second one is to design the discontinuous control such that the system enters the sliding surface $S(t) = 0$ and retains in it forever.Most of the sliding surfaces are defined as

$$S(t) = c^{\mathrm{T}} e(t) \tag{10}$$

where $c = [c_1, c_2, \dots c_{n-1}, c_n]^{\mathrm{T}}$ is chosen such that $c_n = 1$ and the coefficients $c_1, c_2, \dots, c_{n-1}$ are describing the dynamics of the sliding surface $S(t) = 0$ Sarwer et al., 2005.

The most commonly cited approach to reduce the effects of "chattering" has been the so called piecewise linear or smooth approximation of the switching element in a boundary layer of the sliding manifold (Slotine & Sastry, 1983). Inside the boundary layer, the switching function is approximated by a linear gain. In order for the system behavior to be close to that of the ideal sliding mode, particularly when a significant unknown disturbance is to be rejected, sufficiently high gain is needed.

The vector controlled IM model can be expressed by Eq. (11)

$$\dot{\omega}_{\mathrm{r}} = \frac{1}{J}(K_{\mathrm{t}} i_{qs}^{*} - T_1 - B\omega_{\mathrm{r}}) \tag{11}$$

$$\ddot{\omega}_{\mathrm{r}} = \frac{K_{\mathrm{t}}}{J\tau} u - \frac{B}{J}\dot{\omega} \tag{12}$$

where the integration constant of control action τ is represented. Since T_1 is considered to be constant, it does not appear in (12). The variable state representation can be simplified as follows:

$$x_1 = \omega_{\mathrm{r}} - \omega_{\mathrm{ref}} \tag{13}$$

$$x_2 = \dot{x}_1 \tag{14}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} \mu \tag{15}$$

where $a = B/J$, $b = K_t/J \cdot \tau$.

The trajectory, which the SMC forces the system to slide along, is a straight line described in Eq. (16)

$$s = Cx_1 + x_2 = 0 \tag{16}$$

The dynamics described in Eq. (8) is a first-order response with a defined speed response time constant. Various control laws can be used to force the system response.

Where C is a strictly positive real number that guarantees the stability of the sliding motion. Block diagram of the proposed control system is shown in Fig. 3.

Because of the discontinue component (Signum (s) function) in SMC, the chattering is unavoidable. One effective solution is to introduce a boundary layer around the sliding surface (Lin, Shyu, & Lin, 1999). To eliminate the chattering, usually a boundary layer is introduced neighboring the sliding surface. The sat(s/ϕ) for boundary layer is written as

$$\mathrm{sat}(s/\phi) = \begin{cases} \mathrm{sgn}(s/\phi), & \text{if} \quad |s/\phi| \geqslant 1 \\ s/\phi, & \text{if} \quad |s/\phi| < 1 \end{cases} \tag{17}$$

where $\phi > 0$ *represents* the boundary layer thickness, sat(s/ϕ) is a saturation function.

It is obvious that, if $\phi \geqslant s$, (10) is satisfied and the system states move toward the sliding surface, if $|s| < \phi$, the control changes linearly and chattering is reduced. But with this boundary introduction, a steady-state error appears. The smaller ϕ is, the less the steady-state error is and the more serious the chattering is, and vice versa. Since the switching term changes continuously, fast dynamic response is achieved and the chattering phenomenon is completely eliminated.

5. Experimental setup

The experimental setup consisted of a motor and generator. The motor used was a 0.55 kW, 1.34 A, 50 Hz, $\cos\theta = 0.84$, three phase squirrel-cage IM. 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. 4. The stator voltage and frequency were adjusted using a Space Vector PWM (SVPWM) technique.

Error is calculated from difference between reference speed and actual speed taken from incremental encoder. Then, PI generates new control data according to this error. Amplitude and speed values are generated using the control data. Required values for PWM output of the DSP controller are calculated by using two values (amplitude and speed) and SVPWM technique. PWM time base is 200 µs for this application. The control loop is carried out once during each 10 PWM time base. Dead time is formed by the controller. The value of dead time determined by a register is taken 7 µs. 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.

6. Modeling and optimization

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 in Fig. 5.

The used ANN parameters for modeling the system are given in Table 1.

The system was worked for different C and ϕ coefficients. Maximum overshoot and settling time are obtained as experimental. From this data fitness function is calculated. Data used for the ANN model of the system is given in Table 2.

Thirty-eight sets of input–output data taken from the application circuit are given in Table 2. The coefficients of the ANN are trained using data in Table 1. Change in the error in training process is given in Fig. 6.

As shown in Fig. 7, the error values reduce acceptable values when iteration number is 12,000. Therefore, the training process was finished at 12,000 iterations. Then, the best C and ϕ for the whole system are obtained by using genetic algorithm program.

GAs is 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

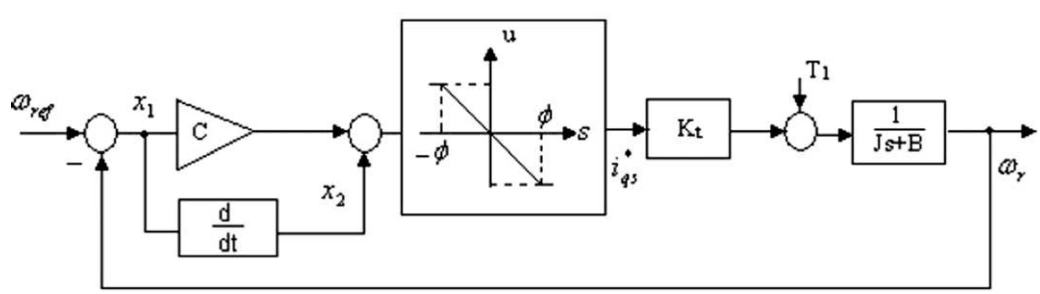

Fig. 3. Block diagram of the proposed control system.

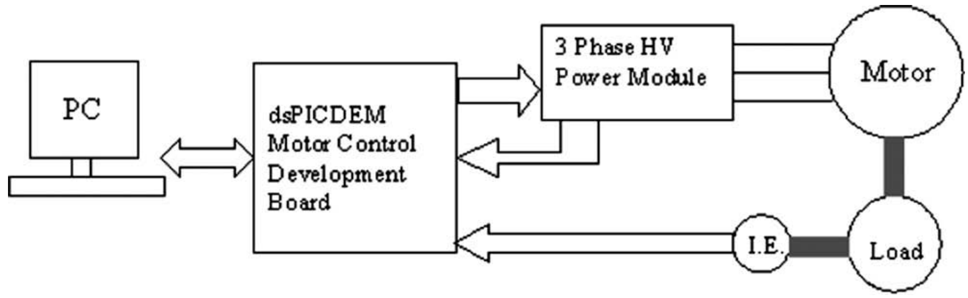

Fig. 4. The block diagram of the application circuit.

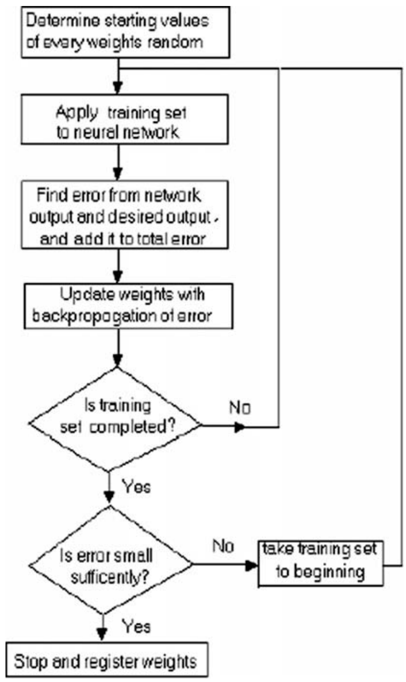

Fig. 5. The flow chart of training process.

Table 1
The used ANN parameters for modelling the system

Parameter	Value
Number of neurones for input layer	2
Number of neurones 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	25,000
Error limit	0.0001
Training coefficient	0.75
Momentum coefficient	0.9

Table 2
Data used for the ANN model of the system

Data set	C	ϕ	=1/(1 + Mo(rpm) +2 * Ts(ms))
1	1	1	0002551
2	1	75	0.002564
3	5	1	0.002632
4	5	75	0.000497
5	5	150	0.002591
6	5	225	0.000498
7	5	300	0.002591
8	5	400	0.000481
9	10	1	0.002639
.	.	.	.
.	.	.	.
37	25	300	0.002786
38	30	400	0.002817

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 et al., 1995). Parameters of GA used in this application are given in Table 3.

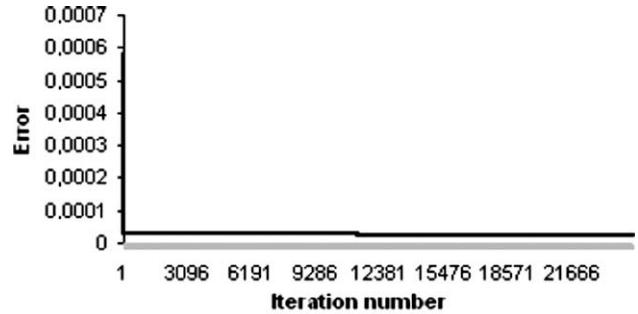

Fig. 6. The error values according to iteration number.

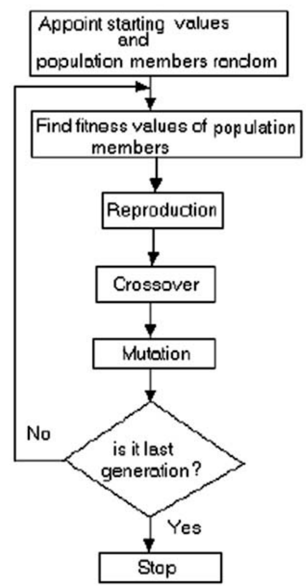

Fig. 7. The flow chart of the GA.

Table 3
Parameters of GA used in the application

Population size	30
Chromosome length	30 bits (15 each for C and ϕ)
Number of generations	100
Selection scheme	Combination of Roulette wheel selection and Elitism
Crossover operator	Double point crossover
Crossover probability (crossover rate)	0.85
Mutation probability (mutation rate)	0.08
Termination criterion	100 generations

Table 4
Fitness values of the members in the first generation

Parameters	Values
Fitness of member 1	0.267399
Fitness of member 2	0.267258
Fitness of member 3	0.266495
Fitness of member 4	0.266342
Fitness of member 5	0.265848
Fitness of member 6	0.265249
Fitness of member 7	0.264923
Fitness of member 8	0.264663
Fitness of member 9	0.264158
Fitness of member 9	0.264258
The best fitness values:	0.267399

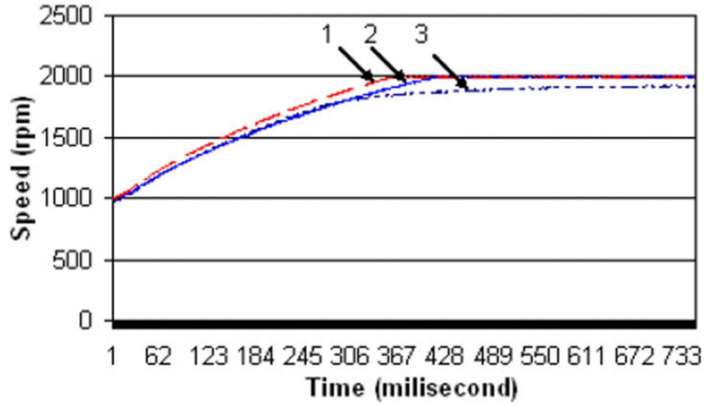

Fig. 8. Neuro-genetic SMC.

Different crossover and mutation rates are used for processing of optimization of genetic algorithms. Ten of the obtained fitness values of the members of the first generation are listed from the largest fitness value to the smallest fitness value, in Table 4. The flow chart of the GA is shown in Fig. 7.

The optimum values for the C and ϕ were obtained using a computer program written in C++ language for the GA. This process executes with three different operators at bit level. Thirty-eight of the C and ϕ were determined at random. Each of C and ϕ consisted of 15 bits. These C and ϕ were entered to ANN model as input. The fitness values were obtained from the ANN outputs. These values were used as the fitness function.

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 0–1 and 1–0. 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 C and ϕ values chosen lay range of (1–30) and (1–400), respectively. The fitness function is defined as

$$f = \frac{1}{M_o + 2 * T_s + 1} \tag{18}$$

7. Results and discussion

To prove the efficiency of the proposed method, the designed controller is applied to the control of the IM. A comparison between the speed responses of IM by conventional SMC and neuro-genetic SMC is presented in Fig. 8. The optimum sliding surface slope and boundary layer width are found as $C = 26$, $\phi = 368$. Fig. 8 shows the speed response for the optimum C and ϕ.

As shown in Fig. 8, curve 1 demonstrates the speed response for the optimum C and ϕ ($C = 26$, $\phi = 368$). The second curve indicates the speed response for selected random C and ϕ ($C = 20$, $\phi = 150$). Curve 3 presents the speed response for the values of $C = 1$ and $\phi = 400$.

This comparison shows clearly that the Neuro-genetic SMC gives good performance. The controller is applied to the system when the motor speed is about 1000 rpm. That is, the speed is increased from 1000 rpm to 2000 rpm by using the proposed controllers. The system is worked to 1000 rpm as open-loop control.

8. Conclusion

In this study, the design method of sliding mode with a boundary layer has been presented. The proposed control structure combines a sliding-mode and a neuro-genetic based controller. The control dynamics of the proposed hierarchical structure has been experimentally investigated. This controller has been applied implemented for IM speed control. The whole system is modeled using ANN. GA is used for determining of the system parameters (sliding surface slope and boundary layer width). The experiments show that the dynamic response of the system using the proposed controller is better than a classical SMC.

Acknowledgments

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

References

Bartolini, G., Ferrara, A., Usai, E., & Utkin, V. I. (2001). On multiinput chattering-free second-order sliding mode control. *IEEE Transactions on Automatic Control, 45*(9), 1711–1718.

Benchaib, A., Rachid, A., & Audrezet, E. (1999). Sliding mode input–output linearization and field orientation for real-time control of induction motors. *IEEE Transactions on Power Electronics, 14*(1), 3–8.

Burton, B., Harley, R. G., Diana, G., & Rodgerson, J. L. (1998). Implementation of a neural network to adaptively identify and control VSI-fed induction motor stator currents. *IEEE Transactions on Industry Applications, 34*(3), 1998). May–June 1998.

Chang, F. J., Twu, S. H., & Chang, S. (1992). Tracking control of DC motors via an improved chattering alleviation control. *IEEE Transactions on Industry Electronics, 39*, 25–29. February 1992.

Cordon, O., & Herrera, F. (1995). A general study on genetic fuzzy systems. In J. Periaux, G. Winter, M. Galan, & P. Custa (Eds.), *Genetic algorithms in engineering and computer science*. John Wiley and Sons.

Dimeo, R., & Lee, K.Y. (1995). The use of a genetic algorithm in power plant control system design. In *IEEE proceeding of the 34th conference on decision & control* (pp. 737–742).

Eksin, I., Guzelkaya, M., & Tokat, S. (2002). Sliding surface slope adjustment in fuzzy sliding mode controllers. In *Proceedings of the 10th mediterranean conference on control and automation – MED2002*, July 9–12.

GARCIA, R. F. (2007). On fault isolation by neural-networks-based parameter estimation techniques. *Expert Systems, 24*, 47–63.

Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning*. Reading, Ma: Addison-Wesley.

Hazzab, A., Khalil Bousserhane, I., & Kamli, M. (2004). Design of a fuzzy sliding mode controller by genetic algorithms for induction machine speed control. *International Journal of Emerging Electric Power Systems, 1*(2).

Ho, E. Y. Y., & Sen, P. C. (1988). Decoupling control of induction motor drive. *IEEE Transactions on Industry Electronics, 35*(2), 253–262.

Jansen, P. L., Lorenz, R. D., & Novotny, D. W. (1994). Observer based direct field orientation: analysis and comparison of alternative methods. *IEEE Transactions on Industry Applications, 30*(4), 945–952.

Jeon, S. H., Oh, K. K., & Choi, J. Y. (2002). Flux observer with online tuning of stator and rotor resistances for induction motors. *IEEE Transactions on Industry Electronics, 49*(3), 653–665.

Karanayil, B., Rahman, M. F., & Grantham, C. (2007). Online stator and rotor resistance estimation scheme using artificial neural networks for vector controlled speed sensorless induction motor drive. *IEEE Transactions on Industry Electronics, 54*(1), 12–86. 2007, February.

Leonhard, W. (1996). *Control of electrical drives*. New York: Springer-Verlag.

Lin, F. J., Chiu, S. L., & Shyu, K. K. (1998). Novel sliding mode controller for synchronous motor drive. *IEEE Transactions on Aerospace and Electronic Systems, 34*(2), 532–542.

Lin, F.-J., Shyu, K.-K., & Lin, Y.-S. (1999). Variable structure adaptive control for PM synchronous servo motor drive. *IEE Proceedings-Electric Power Applications, 146*(2), 173–185.

Lorenz, R. D., & Lawson, D. b. (1988). A simplified approach to continuous on-line tuning of field oriented induction machine drives. *IEEE Transactions on Industry Applications, 26*(3). May–June 90, 1988.

Mondal, S. K., Pinto, J. O. P., & Bose, B. K. (2002). A neural network based space-vector PWM controller for a three voltage-fed inverter induction motor drive. *IEEE Transactions on Industry Applications, 38*(3), 660–669. May, 2002.

Moreira, J. C., & Lipo, T. A. (1993). A new method for rotor time constant tuning in indirect field oriented control. *IEEE Transactions on Power Electronics, 8*, 626–631.

NG, K.C., LI, Y., Murray-Smith, D.J., & Sharman, K.C. (1995). Genetic algorithms applied to fuzzy sliding mode controller design. In *Proceedings of the first IEE/IEEE international conference on genetic algorithms in engineering systems, innovations and applications* (pp. 220–225).

Park, T. G., & Lee, K. S. (1998). SMC-based adaptive input-output linear control of induction motors. *IEE Proceedings Control Theory and Applications, 145*(1), 55–62.

Sarwer, M. G., Rafiq, Md. A., Data, M., Ghosh, B. C., & Komada, S. (2005). Chattering free neuro-sliding mode control of DC drives. *IEEE PEDS 2005*, 1101–1106.

Schauder, C. (1992). Adaptive speed identification for vector control of induction motors without rotation transducers. *IEEE Transactions on Industry Applications, 28*(5), 1054–1061.

Shiau, L. G., & Lin, J. L. (2001). Stability of sliding-mode current control for high performance induction motor position drives. *IEE Proceedings-Electric Power Applications, 148*(1), 69–75.

Slotine, J. J. E., & Li, W. (1991). *Applied nonlinear control*. New Jersey: Prenitce Hall: Englewood Cliffs.

Slotine, J.-J., & Sastry, S. S. (1983). Tracking control of nonlinear systems using sliding surfaces with application to Robot manipulator. *International Journal of Control, 38*(2), 465–492.

Utkin, V. I. (1993). Sliding mode control design principle and application to electric drives. *IEEE Transactions on Industry Electronics, 40*(1), 23–36.

Velasco, J. R., & Magedalena, L. (1995). Genetic algorithm in fuzzy control systems. In J. Periaux & G. Winter (Eds.), *Genetic algorithms in engineering and computer science*. John Wiley and Sons.

Wai, R. J. (2001). Adaptive sliding-mode control for induction servo motor drive. *IEE Proceedings-Electric Power Applications, 147*(6), 533–562.

Wai, R. J., Lin, K.M., & Lin, C.Y. (2004). Total sliding-mode speed control of field-oriented induction motor servo drive. In proceedings of the 5th Asian control conference, July 20–23. Australia.

Young, K. D., Utkin, V. I., & Ozguner, U. (1999). A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology, 7*, 328–341. May 1999.

Zhang, J., & Barton, T. (1991). Robustness enhancement of dc drives with a smooth optimal sliding-mode control. *IEEE Transactions on Industry Electronics, 27*, 686–693. August 1991.