

An Educational Tool for Neural Network Control of Brushless DC Motors*

MUAMMER GOKBULUT and AHMET TEKIN

Firat University, Elazığ, Turkey. E-mail: mgokbulut@firat.edu.tr

This paper presents an educational tool developed for neural network (NN) control of brushless DC (BLDC) motors. Neural networks courses are widely offered at the graduate and undergraduate level due to the successful applications of neural networks to the nonlinear and unmodelled systems control. However, teaching of neural network control in a laboratory may be time consuming and an expensive task. The developed software helps students learn the application of neural network control to the electric machinery. It provides flexible structure and graphical interface which permits the design of the neural networks and its training with various learning algorithms. Using the proposed tool, neural network control performance of BLDC motors can be monitored graphically for different NN control structure under different load conditions and parameter variations.

Keywords: control; neural network; DC motors

INTRODUCTION

BRUSHLESS DC MOTORS are preferred in high performance drive applications such as, robotics, dynamic actuation, machine tools and positioning devices, due to their favourable electrical and mechanical characteristics such as high torque to volume ratio, high efficiency and low moment of inertia [1, 2]. BLDC motors have the speed and torque control characteristics similar to DC motors. The torque and flux control can be achieved by means of field oriented control (FOC) which is sometimes called decoupling control or vector control. Thus, two control loops exist in the BLDC motor control system, which are current and speed control loops [3]. A desirable control performance of the motor must be provided even when the loads, inertia and parameters are varying. The conventional controllers used in the speed and/or vector control loops of the motor become poor when the load is nonlinear and uncertainties exist. Therefore, control strategy must be adaptive, robust and accurate.

NNs have many advantageous features that include parallel and distributed processing, robustness and efficient nonlinear mapping between inputs and outputs, without knowledge of the system model [4]. In the last two decades, much research have been done on applications of artificial neural networks for the identification and control of nonlinear dynamic systems [5–7]. Various NN control structures based on adaptive control and conventional control techniques are proposed using feedforward and recurrent NNs [8–10]. Similarly, wide varieties of NN control methods are applied to BLDC motors [11, 12]

under nonlinear loads and parameter variations. Many colleges are now offering neural networks courses as a result of successful application of NN control in nonlinear systems. Therefore the quantity of the material to be taught has increased. However, this increase does not make the understanding of the course easy for the students, if the courses are not supported with the educational tools and experiments. It is often difficult for schools with their limited and often expensive equipment facilities. Therefore, computer simulations and dedicated software become very important. For this purpose, many colleges are now developing educational tools for various areas [13–20].

The present teaching approach uses simulations, which include the animation and graphics if necessary to assist the students in visualizing the concepts and to provide graphical feedback during the learning process. There are many software packages in the areas of neural networks. The well known software package, MATLAB-Simulink developed by Mathworks Inc., offers neural networks and control toolboxes, in addition to many other toolboxes [21]. These software packages provide an environment for the development and evaluation of neural networks; however, they are not well suited to electrical machines and only a few advanced students could achieve its application to electrical drives control in a limited time.

In this study, an educational tool is presented for NN identification and control of electrical drives. The software can be easily used for teaching the materials to graduate and undergraduate students. It also may be used by instructor for curriculum development. The tool has a flexible structure and graphical interface which permits the design of the NN and its training with various

* Accepted 1 April 2005.

learning algorithms. The program is prepared in DELPHI and it can be installed on a PC operating in Windows environment. In the paper, initially the dynamic model of a BLDC motor is presented and conventional control strategy is given. Then, NNs control structures and training algorithms are outlined. In the last section, the software is explained in details by showing the graphical and numerical results obtained from NNs identification and control of a BLDC motor.

BLDC MOTORS

The state space model of a BLDC motor referred to the rotor rotating reference frame is given by:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} i_q \\ i_d \end{bmatrix} = & \begin{bmatrix} -\frac{R}{L_q} & 0 \\ 0 & -\frac{R}{L_d} \end{bmatrix} \begin{bmatrix} i_q \\ i_d \end{bmatrix} \\ & + \begin{bmatrix} \frac{1}{L_q} & 0 \\ 0 & \frac{1}{L_d} \end{bmatrix} \begin{bmatrix} v_q \\ v_d \end{bmatrix} \\ & + \begin{bmatrix} -\frac{\lambda}{L_q} \omega - \frac{L_d}{L_q} \omega i_d \\ \frac{L_d}{L_d} \omega i_q \end{bmatrix} \end{aligned} \quad (1)$$

where, i_d and i_q is the direct and quadrature components of the stator current, R is the stator resistance, L_q and L_d is the inductance, λ is the magnitude of the flux linkage established by the rotor magnet and ω is the rotor speed. The model is based on the assumptions that the air-gap is uniform, rotor induced current are neglected due to the high resistivity of magnet and stainless steel, and the motor is supplied by a three phase sinusoidal input source [1, 2]. For the vector control of electrical drives, stator currents are decomposed into the flux and torque components which can be controlled independently. Using the flux and torque current components, electromagnetic torque produced by the BLDC motor is given by Equation (2):

$$T_e = \frac{3P\lambda}{4} i_q + (L_d - L_q) i_q i_d \quad (2)$$

In the surface mounted permanent magnet motors, the d-axis inductance is assumed to be equal to the q-axis inductance. Thus, electromagnetic torque equation becomes dependent only on the q-axis current, i.e.

$$T_e = \frac{3P\lambda}{4} i_q \quad (3)$$

On the other hand, dynamic equation of the mechanical system can be expressed as:

$$\frac{d}{dt} \omega - \frac{3P^2\lambda}{8J} i_q - \frac{B}{J} \omega - \frac{P}{2J} T_L \quad (4)$$

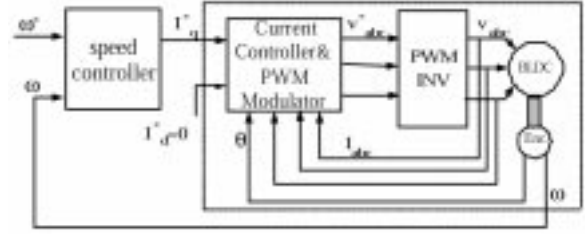


Fig. 1. Speed control block diagram of the BLDC motor.

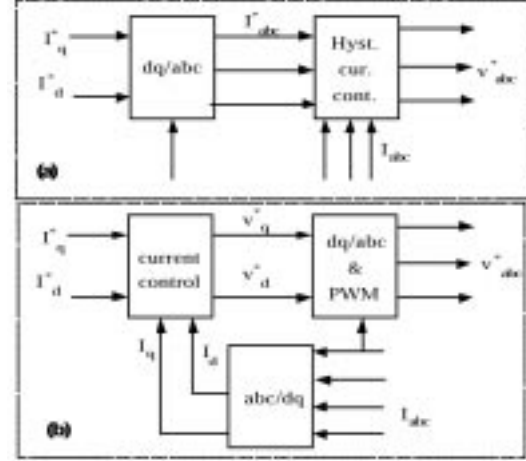


Fig. 2. (a) Hysteresis band current controller; (b) conventional current controller and PWM mod.

where ω is the electrical rotor speed, T_L is the load torque (Nm) and J and B are inertia and friction of the motor respectively.

For the vector control, the output of the speed controller is the quadratic current component (torque current) i_q^* , and direct current component (exciting current) i_d^* is set to zero to avoid demagnetization of the permanent magnet on the rotor [3]. The block diagram of the vector controlled BLDC motor is shown in Fig. 1. Figure 2(a) shows the hysteresis band current controller and Fig. 2(b) shows the conventional current controller and PWM modulator unit.

NEURAL NETWORK CONTROL OF BLDC MOTORS

As seen in Fig. 1, NN can be used for the speed control, current control and the identification of the BLDC motor. In addition, the speed and position estimation for the sensorless control of a BLDC motor is also another important application of the NN in electrical drives. The developed software is restricted only to the identification and speed control of BLDC motors using various NN control structures and learning algorithms under parameter variations.

In order to determine the NN forward identification model of a BLDC motor, shown with dashed line in Fig. 1, forward dynamics of the motor including current controllers should be

considered. If the dynamics of the fast current control loop are ignored i.e., $i_q^* = i_q$ and $i_d^* = i_d$, NN identification model can be expressed as:

$$\omega_m(k) = NN(\omega(k-1), i_q^*(k)) \quad (5)$$

which is obtained using the discrete time equivalent of Equation(4), and shows the forward dynamics of the motor since the flux current component i_d is equal to zero. If the actual current controllers are considered, it is clear that the more delayed components of the speed and torque current may be required. Hence, NN models representing the dynamics of the motor and current controller can be derived using more delayed components of the speed and torque current, i.e.,

$$\begin{aligned} \omega_m(k) &= NN(\omega(k-1), \dots, \omega(k-n), \\ i_q(k), i_q(k-1), \dots, i_q(k-r)) \end{aligned} \quad (6)$$

Forward identification scheme of the BLDC motor using NN is illustrated in Fig. 3. In a similar way, inverse identification model of the motor, used as speed controller, can be derived by using Equation (6) and it is given as:

$$\begin{aligned} i_q^*(k) &= NN^{-1}(\omega(k), \omega(k-1), \dots, \\ \omega(k-n), i_q^*(k-1), \dots, i_q^*(k-r)) \end{aligned} \quad (7)$$

The forward and inverse NN models of BLDC

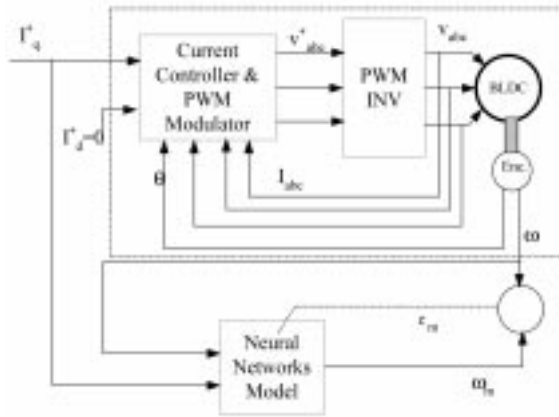


Fig. 3. Forward identification of BLDC motors with current controller using NN.

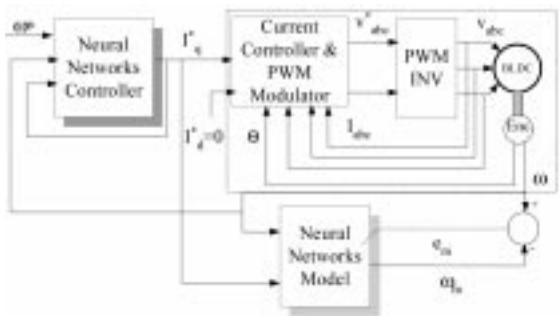


Fig. 4. Indirect adaptive control of BLDC motors using NNs.

motors, using speed and torque current delays, can be trained in the developed software. Indirect adaptive control scheme of the BLDC motor using the both NN forward and inverse model of the motor is shown in Fig. 4.

Direct adaptive control scheme is obtained if the forward NN model is removed from the system shown in Fig. 4. Wide varieties of NN control structures are proposed for identification and control of nonlinear systems and can be accessed them in the literature [5–10]. In this software, direct and indirect NN control schemes are simulated and conventional PID control is also included for comparison.

Training of the neural network

Multilayer neural network is used in the proposed software for both identification and control purposes. Forward mathematical relation of one hidden layer NN can be given as:

$$v_j = \varphi \left\{ \sum_{i=1}^n W_{ji} x_i + b_j \right\}, \quad \omega_m = \sum_{j=1}^m \theta_j v_j + q \quad (8)$$

where, ω_m is the output of the NN model, x_i s are the inputs of the NN, v_j s are the outputs of the hidden layer, W_{ji} s are the weights between the input and the hidden layer neuron and θ_j s are the weights of the output layer. It means that the training of the NN is to find a procedure for the adaptation of the NN weights, which minimize the selected performance criteria depending on the error between NN output and teaching output. Various derivatives based training algorithms including pattern and batch learning are given in the literature [10, 22]. For pattern learning, the error (e_m) between the NN model and actual motor speed and the square of the error as performance criteria (E) are given:

$$e_m = \omega - \omega_m \quad E = \frac{1}{2} e_m^2 \quad (9)$$

The first derivative of the performance criteria E is used by backpropagation algorithm for the updates the weights (W and θ). Thus, the correction to be applied to any weight in the hidden layer and output layer can be expressed respectively as:

$$\Delta W = \mu \nabla E_W \quad \Delta \theta = \mu \nabla E_\theta \quad (10)$$

where, μ is learning rate, ∇E_W are ∇E_θ the gradients of the performance criteria. Note that Equations (9) and (10) are for the training of the NN model. Similar equations can be derived for the training of the NN controller. If the gradient descent direction is obtained by using the second derivatives of the performance criteria, if available, the NNs can be trained by Newton's methods and it can be defined as:

$$\Delta W = \mu H^{-1} \nabla E_W \quad \Delta \theta = \mu H^{-1} \nabla E_\theta \quad (11)$$

where, H is the Hessian matrix obtained by second derivative of the performance criteria. Classical

Newton's method defined in Equation (11) requires much less training time than the back-propagation method. However, the main disadvantage of Newton's method is that calculating the inverse of the Hessian matrix is computationally intensive and may introduce numerical problems due to rounding of errors. In addition, if the first weights of the NNs selected randomly are remote from the optimum weights, this method may not yield descent direction due to the truncated higher-order terms in a Taylor series expansion of the performance criteria [22]. Levenberg-Marquardt modification of the Newton's method can be obtained by adding a positive definite matrix to the Hessian matrix to make it positive definite and it can be defined as:

$$\Delta W = \mu(H + \lambda I)^{-1} \nabla E_w \quad (12a)$$

$$\Delta \theta = \mu(H + \lambda I)^{-1} \nabla E_\theta \quad (12b)$$

where, λ is a positive value and I is the identity matrix.

THE EDUCATIONAL TOOL

The software is prepared using DELPHI and it works in the MS Windows environment. It is developed to help students improve their understanding of neural network modelling and control. Using the software, learning and generalization performance of the various NN control structures of BLDC motors under the parameter and load variations can be studied and various learning algorithms can be used for the training. Both graphical and numerical results can be seen on a

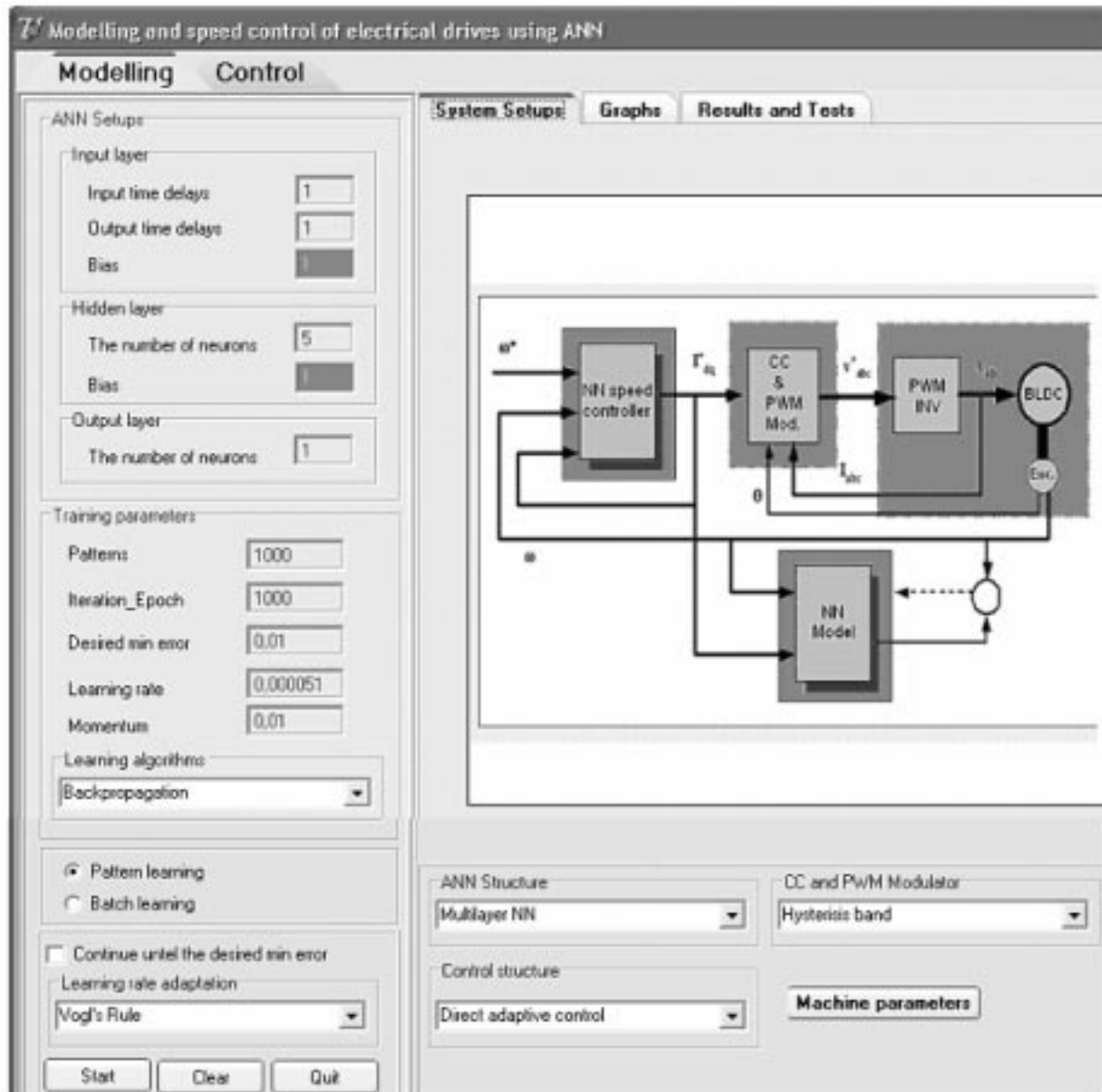


Fig. 5. The main window of the control tabsheet.

PC monitor by choosing appropriate windows. The software consists of two tabsheets: modelling and control. Each tabsheet includes three windows which are system setups, graphs and results-&-tests windows. A system setups window including the neural network design section is the main window.

There are some negligible differences between modelling and control tabsheets, and the user can aware of them easily in the software. Hence, all of the windows are not given in order to avoid the repetition of similar figures. The main window of the control tabsheet is given in Fig. 5 and it is divided into two sections. The left side of the main window is used for designing of NNs. Dimension of input vector, hidden layer units and outputs can be defined by users. Training parameters such as the number of training patterns, training iterations, desired minimum error as stopping criteria and learning rate can be specified as well. In addition, one of the various learning algorithms and pattern or batch learning mode, and learning rate adaptation rule can be selected by users. On the right side of the main window, three menus and related windows are placed, which are system setups, graphs and results and test windows. When the setups menu is selected, modelling or control configuration appears on the right side of the main window, which permits the changing of NN control structures, vector control methods and motor parameters, using the related buttons at the bottom of the window. Submenus of control structure are shown in Fig. 6. Submenus of the current controller&PWM modulator are given in Fig. 7. The settings in Figs 6 and 7 allow users to test the NN control performance of motor for various current controllers and NN control structures.

Figure 8 shows the parameter settings window. When the graphs menu is selected, graphs window appears on the right side of the main window (as shown in Fig. 9), which permits the variables of the controller and the motor to be seen using the related buttons on the right side of the window. Two graphics can be shown at a time and more variables can be selected to see more variables on the same graph. If desired, graphs can be printed out and/or copied and used in other softwares. Speed tracking performance of the NN controller for the sinusoidal reference input is given in the first graph in Fig. 9, and the control input (the torque current) is presented in the second graph in Fig. 9.

As seen in Fig. 10, on the upper half of the results and test window, the numerical values of the neural networks weights, motor and NN outputs, the iteration number required for training and sum of the squared error can be monitored. The lower half of the window is used for testing of trained NN model or controller, using the different test inputs such as square wave, step and triangular wave. Sum of the squared errors for the selected test input can be obtained and modelling or control performance of the trained NNs can be

illustrated graphically. Identification performance of the trained NN for the square test input is illustrated by the graph in Fig. 10.

Although many different simulation results can be obtained using the developed software in addition to results given in Fig. 9 and Fig. 10, some simulation examples are also presented. The sum of the squared errors curve obtained by training of the NN model is given in Fig. 11 and the error between the actual motor speed and the output of the NN model trained using the Levenberg-Marquardt learning algorithm is shown in Fig. 12.

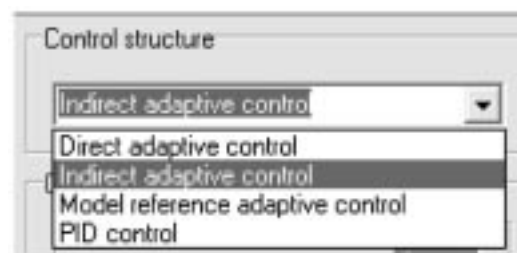


Fig. 6. Submenus of the control structure.

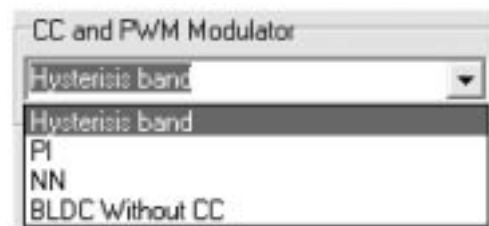


Fig. 7. Submenus of the current controller.

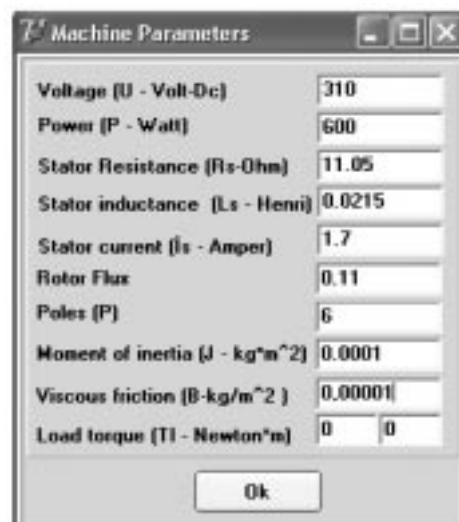


Fig. 8. Motor parameter settings.

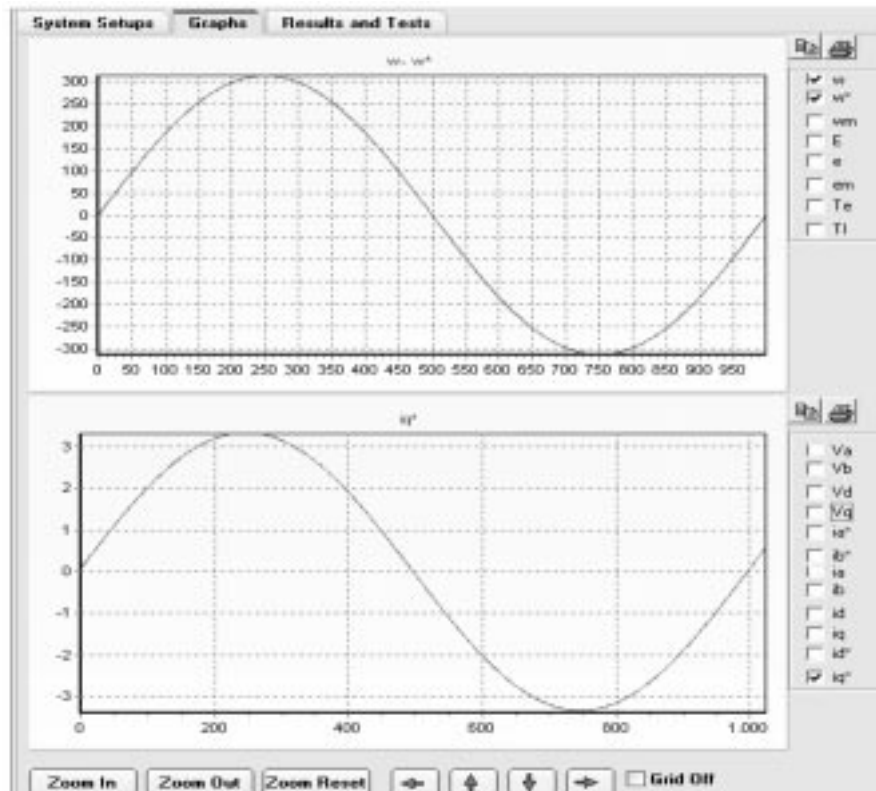


Fig. 9. Graphs window and some results obtained NN control of BLDC motors.

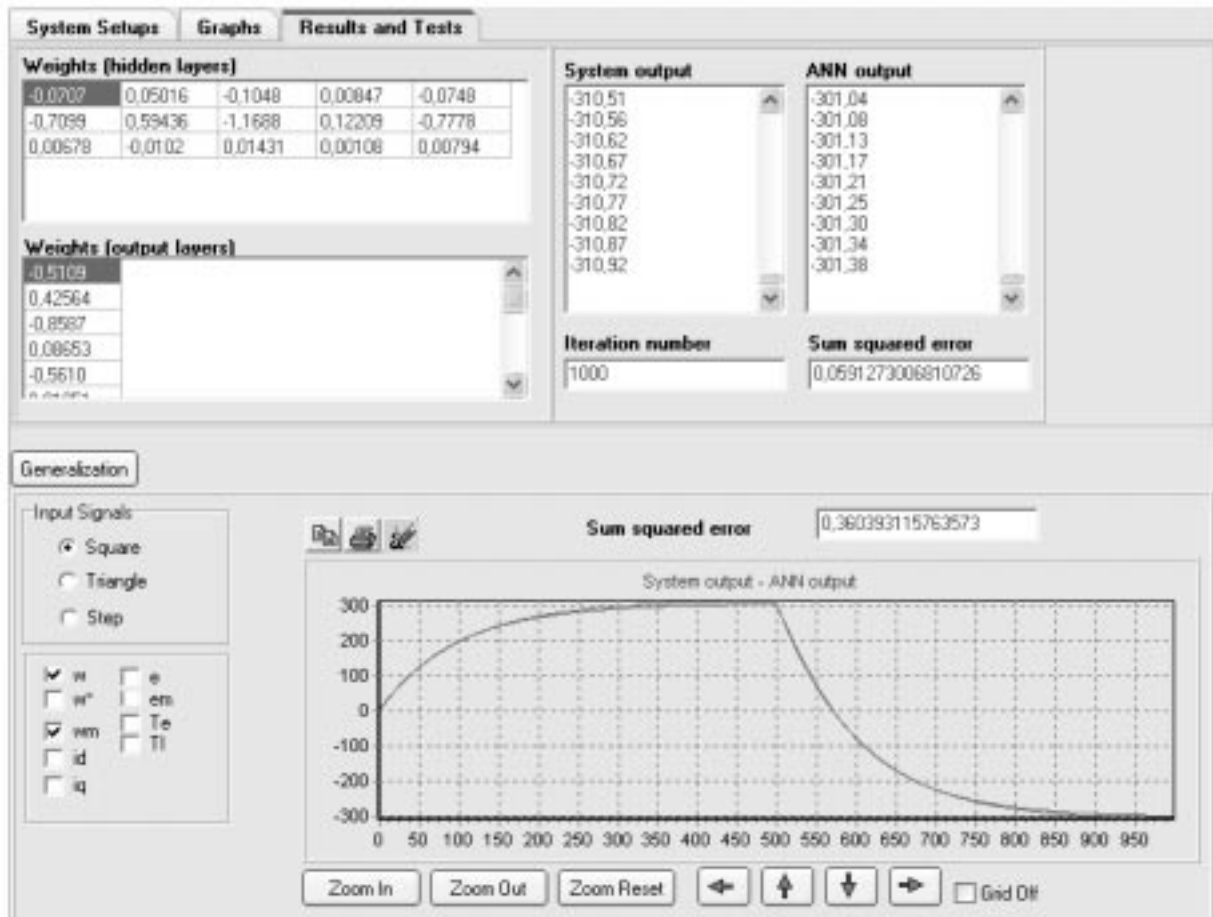


Fig. 10. Numerical results and test window.

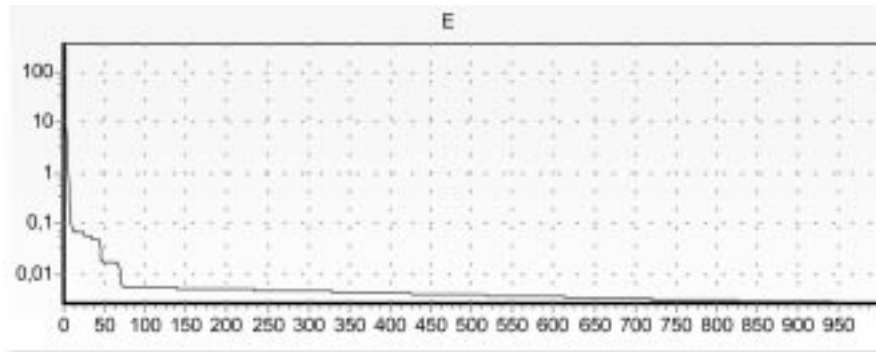


Fig. 11. Sum of the squared errors curve.

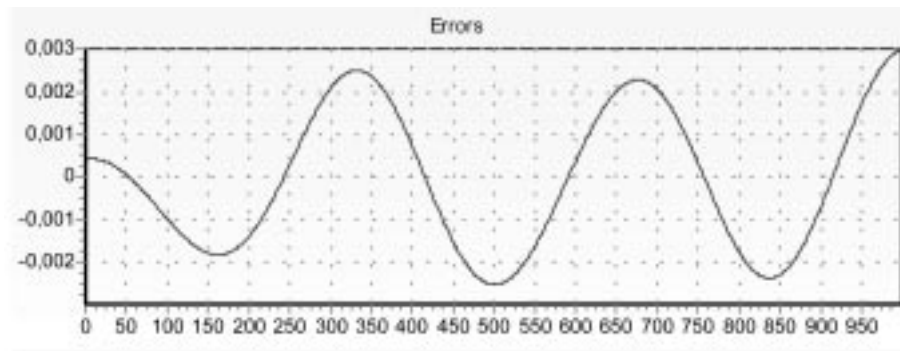


Fig. 12. Speed error between the motor and NN model.

CONCLUSIONS

In this paper, an educational tool is developed for neural network control of brushless DC motors. It aims to help students improve their understanding of neural network control of electrical drives. The tool has a flexible structure and graphical interface. It enables the user to design neural networks and to change the motor parameters. Motor and controller variables can be monitored for various NN control structures under the various load conditions.

The software is prepared in a Delphi environment and the setup files are available for downloading at <http://imgokbulut.firat.edu.tr>. To benefit from the tool, students should have *a priori* knowledge of electrical machines, neural networks and control techniques. The tool has been used by graduate students who have the required background in the department of electronics and

computer education at Firat University. The critics of the graduate students in the neural network control courses are very positive. Recently, the software has started to be used in laboratory experiments of the undergraduate neural network course. We have the impression that the tool should be adapted for the undergraduate level. Therefore, it is planned to assess the educational advantages of the tool with more students through evaluation sheets.

It is expected that the tool achieves the following educational goals. The students should be able to:

- Gain a comprehensive understanding on neural network control techniques of drives.
- Do virtual experiments on a PC to be ready for real lab experiments.
- Relate drive parameters to response of the drive.
- Improve his/her knowledge on neural network control in a short time.

REFERENCES

1. S. A. Nasar., I. Boldea and L. E. Unnewhr., Permanent magnet, reluctance and self-synchronous motors, CRC Pres, New York , (1993).
2. M. A. Rahman and P. Zhou, Analysis of brushless permanent magnet synchronous motors, *IEEE Trans. Industrial Electronics*, **43**, 1996, pp. 256–267.
3. S. E. Lyshevski and A. Menozzi, Control of Permanent Magnet Brushless DC Motors *Proc. of the American Control Conference*, Arlington, 2001, pp. 25–27.
4. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Comp. Inc, (1994).

5. K. S. Narendra and K. Parthasarathy, Identification and control of dynamical system using neural networks, *IEEE Trans. Neural Networks*, 1, 4–27, (1990).
6. J. Sjöberg, Q. Zhang and L. Ljung, Nonlinear black-box modeling in system identification, *Automatica*, 31, 1995, pp. 1691–1724.
7. K. S. Narendra and S. Mukhopadhyay, Adaptive control using neural networks and approximate models, *IEEE Trans. Neural Networks*, 8, 1997, pp. 475–485.
8. K. J. Hunt, D. Sbarbaro and R. Zbikowski, Neural networks for control systems—a survey, *Automatica*, 28, 1992, pp. 1083–1112.
9. C. Chee and K. Y. Lee, Diagonal recurrent neural networks for dynamic systems control, *IEEE Trans. Neural Networks*, 6, 1995, pp. 144–156.
10. K. S. Narendra and K. Parthasarathy, Gradient methods for the optimization of dynamical systems containing neural networks, *IEEE Trans. on Neural Networks*, 2, 1991, pp. 252–262.
11. M. A. El-sharkawi, A. A. El-Samahy and M. L. El-Sayed, High performance drive of DC brushless motors using neural network, *IEEE Trans. Energy Conv.*, 9, 1994, pp. 317–322.
12. A. Rubai, R. Kotaru and M. D. Kankam, A continually online-trained neural network controller for brushless DC motor drives, *IEEE Trans. Industry App.*, 36, 2000, pp. 475–483.
13. C. Elmas and M. A. Akcayol, Virtual electrical machinery laboratory: a fuzzy logic controller for induction motor drive, *Int. J. Eng. Educ.*, 20, 2004, pp. 226–233.
14. A. Şengür and H. Guldemir, An educational interface for automatic recognition of analog modulated signals, *J. Applied Sciences*, 5, 2005, pp. 513–517.
15. M. F. Iskander, Technology-based electromagnetic education, *IEEE-Trans. Microwave Theory and Techniques*, 50, 2002, pp. 1500–1505.
16. M. A. Akcayol, A. Cetin and C. Elmas, An educational tool for fuzzy logic-controlled BDCM., *IEEE-Trans. Education*, 45, 2002, pp. 33–42.
17. A. Keyhani, M. N. Marwali, L. E. Higuera, G. Athalye and G. Baumgartner, An integrated virtual learning system for the development of motor drive systems, *IEEE Trans. Power Systems*, 17, 2002, pp. 1–6.
18. E. G. Roselló, J. B. G. P. Schofield, J. G. Dacosta and M. P. Cota, Neuro-Lab: A highly reusable software-based environment to teach artificial neural networks, *Comput. Appl. Eng. Educ.*, 11, 2003, pp. 93–102.
19. J. Fulcher, Laboratory support for the teaching of neural networks, *Int. J. Electr. Eng. Educ.*, 35(1), 1998, pp. 29–36.
20. A. Tekin, A packet program for neural network control of induction motors, MSc thesis, Firat University, (2003).
21. *Neural Network Toolbox*, The Math Works Inc., Natick, MA (2001).
22. J. S. R. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall (1997).

Muammer Gokbulut received the B.Sc. degree in electrical and electronics education and the M.Sc. degree in electrical education from Gazi University, Ankara, TURKEY, in 1980 and 1988 respectively, and the Ph.D. degree in electronics engineering from Erciyes University, Kayseri, Turkey, in 1998. He is currently associate professor and the head of the department of electronics and computer education at Firat University, Faculty of Technical Education, Elazig, Turkey. His research interests include neural networks, fuzzy-neural networks, electrical drives control and adaptive control systems.

Ahmet Tekin received the B.Sc. and M.Sc. degrees in electronics and computer education from Firat University, Elazig, Turkey, in 1996 and 2003 respectively. He is currently studying PhD and teaching at Firat University, department of informatics. His research interests include artificial intelligence, web-based distance learning and virtual laboratories and education tools.