A Virtual Electrical Drive Control Laboratory: Neuro-Fuzzy Control of Induction Motors

MUAMMER GÖKBULUT,1 CAFER BAL,1 BEŞİR DANDIL2
1Department of Electronic and Computer Education, Faculty of Technical Education, Firat University, Elazig, Turkey
2Department of Electrical Education, Faculty of Technical Education, Firat University, Elazig, Turkey

Received 2 December 2004; accepted 3 January 2006

ABSTRACT: Neural and fuzzy courses are widely offered at graduate and undergraduate level due to the successful applications of neural and fuzzy control to nonlinear and unmodeled dynamic systems, including electrical drives. However, teaching students a neuro-fuzzy controlled electrical drive in a laboratory environment is often difficult for schools with limited access to expensive equipment facilities. Therefore, computer simulations and dedicated software are needed to assist the students in visualizing the concepts and to provide graphical feedback during the learning process. In this article, an educational software is proposed for the neuro-fuzzy control of induction machine drives. The tool helps students learn the application of neuro-fuzzy control of electrical drives. The software has a flexible structure and graphical user interface. The neuro-fuzzy architecture, the motor and load parameters can be easily changed in the developed software. Neuro-fuzzy control performance of induction motors can be monitored graphically for various control structures and current controllers. © 2006 Wiley Periodicals, Inc. Comput Appl Eng Educ 14: 211–221, 2006; Published online in Wiley InterScience (www.interscience.wiley.com); DOI 10.1002/cae.20082

Keywords: neuro-fuzzy control; induction motors; educational software

INTRODUCTION

An induction machine is a highly coupled, nonlinear dynamic plant, and its parameters vary with time and operating conditions. The stator current components strongly interact with the rotor flux components producing the electromagnetic torque necessary to meet the angular speed requirements. High performance control of variable speed induction motor drives can be achieved by means of field-oriented control. The aim of the field-oriented control is the decoupling of the rotor flux terms. This decoupling technique permits independent control of the torque...
and flux producing components of the stator current [1,2]. Field orientation requires exact knowledge of rotor flux terms. Indirect method uses the motor model to estimate the flux position. It is much easier to implement the indirect field orientation; however, it is sensitive to parameter variations, and heating and saturation of the motor result in errors on the output torque and flux of the motor [3–5]. The conventional controllers used in induction motor drives become poor and it should be adaptive and robust, when the load is nonlinear, and uncertainties exist.

Neuro-fuzzy systems combine the advantages of neural networks and fuzzy logic systems. Neural networks provide connectionist structure and learning abilities to the fuzzy logic systems, and the fuzzy logic systems provide the neural networks with a structural framework with high-level fuzzy IF-THEN rule thinking and reasoning [6]. During the past two decades, neural networks and fuzzy logic control are successfully applied to the control of nonlinear and unmodeled systems [7–11]. Neural network-based fuzzy systems, which have the learning ability of neural networks to realize the fuzzy logic inference system, have gained popularity in the control of nonlinear systems [12–14] and electrical machinery [15–17]. Various neuro-fuzzy architectures based on fuzzy inference system and feed forward or feedback connections between the layers are proposed to increase the approximation, learning and adaptation capabilities of the neuro-fuzzy systems.

Neural and fuzzy courses are widely offered at the graduate and undergraduate level due to successful applications of neural and fuzzy systems to the control of nonlinear and unmodeled 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 limited access to expensive equipment facilities. Furthermore, 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. Therefore, computer simulations and dedicated software become very important. Many colleges are now developing educational tools for: fuzzy logic control of electrical machines [18,19], effective use of multimedia for teaching of electromagnetic education [20], virtual learning system for DSP-based control system of motor drives [21,22], and neural network control of electrical drives [23]. The well known software package, MATLAB-Simulink developed by Mathworks, Inc., offers neural-fuzzy toolboxes, in addition to many other toolboxes. It provides an environment for the development and evaluation of neural-fuzzy systems; 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, educational software is developed for neuro-fuzzy control of induction motor drives. It is a part of a virtual intelligent control laboratory project for electrical drives carried out by the authors. The software can be easily used for teaching the materials to graduate and undergraduate students. It may also be used by instructors for curriculum development. The tool has a flexible structure and graphical user interface which permits the design of the neuro-fuzzy systems. The control performance of the various neuro-fuzzy controllers (NFCs) can be monitored under the motor and load parameter variations for various current controllers. The program is developed using Borland Visual C++ Builder and it can be installed on a PC operating in Windows environment.

In the article, initially the neuro-fuzzy control of an induction motor is presented. Then, NFCs and training algorithm are outlined. In the last section, the software is explained in details by showing the graphical and numerical results obtained from neuro-fuzzy control of an induction motor.

NEURO-FUZZY CONTROL OF INDUCTION MOTORS

The fundamentals of field oriented control implementation can be explained with the help of motor model represented in a synchronously rotating d–q axis reference frame. Stator and rotor voltage equations of an induction motor in the synchronously rotating reference frame can be written as follows [1].

\[
v_{qs} = R_{se} i_{qs} + \frac{d\psi_{qs}}{dt} + \omega_r \psi_{ds} \quad (1)
\]

\[
v_{ds} = R_{se} i_{ds} + \frac{d\psi_{ds}}{dt} + \omega_r \psi_{qs} \quad (2)
\]

\[
v_{qr} = 0 = R_{se} i_{qr} + \frac{d\psi_{qr}}{dt} + (\omega_e - \omega_r) \psi_{dr} \quad (3)
\]

\[
v_{dr} = 0 = R_{se} i_{dr} + \frac{d\psi_{dr}}{dt} - (\omega_e - \omega_r) \psi_{qr} \quad (4)
\]

where, the subscript s shows the stator variables, subscript r shows the rotor variables, R is the resistance, \(\psi_q\) and \(\psi_d\) are the flux linkages, \(\omega_e\) is the
synchronous speed, and \( \omega_r \) is the rotor speed. The stator and rotor flux linkages are defined as,

\[
\begin{bmatrix}
\psi_{qs} \\
\psi_{ds} \\
\psi_{qr} \\
\psi_{dr}
\end{bmatrix} = \begin{bmatrix}
L_s & 0 & L_m & 0 \\
0 & L_s & 0 & L_m \\
L_m & 0 & L_r & 0 \\
0 & L_m & 0 & L_r
\end{bmatrix} \begin{bmatrix}
\psi_{qs} \\
\psi_{ds} \\
\psi_{qr} \\
\psi_{dr}
\end{bmatrix}
\] (5)

The electromagnetic torque produced by induction motor is,

\[
T_e = 3 PL_m \left( \psi_{qs}^* \psi_{qs}^* - \psi_{qr} \psi_{dr} \right)
\] (6)

The rotor flux orientation implies that the \( \psi_{dr} = \psi_r \), and \( \psi_{qr} = 0 \) [2]. Furthermore, rotor currents, which are inaccessible, can be obtained from Equation (5). Substituting \( \psi_{dr} = \psi_r \), and \( \psi_{qr} = 0 \) into Equations (3) and (4) and eliminating the rotor currents, Equations (3) and (4) can be rewritten as follows [1]:

\[
\omega_d = \left( \omega_e - \omega_r \right) = \frac{L_m}{\tau_r \cdot \omega_r} \psi_{iq}
\] (7)

\[
\tau_r = \frac{d \psi_r}{dt} + \psi_r = L_m \cdot i_{ids}
\] (8)

where \( \omega_d \) is the slip frequency, which is the difference between synchronous speed and rotor speed, and \( \tau_r \) is the rotor circuit time constant. Substituting \( \psi_{dr} = \psi_r \), and \( \psi_{qr} = 0 \) in Equation (6), electromagnetic torque equation reduces to

\[
T_e = 3 PL_m \frac{\psi_{qs}^* \psi_{qs}^*}{4L_r}
\] (9)

The rotor angular speed \( \omega_r \) can be derived from the mechanical dynamics of the motor [1,2] as follows:

\[
\frac{d \omega_r}{dt} = \frac{B}{J} \omega_r + \frac{1}{J} \left( T_e - T_L \right)
\] (10)

where \( J \) is the moment of inertia, \( B \) is the viscous friction of the motor, and \( T_L \) is the load torque. Using the slip frequency calculated from Equation (7) and rotor speed obtained from Equation (10), the flux position \( \theta_e \) used for coordinate transformation can be obtained as

\[
\theta_e = \int \omega_e dt = \int \left( \omega_r + \omega_d \right) dt
\] (11)

Once \( \theta_e \) is obtained, the stator reference current vector can be produced and thus the vector control of induction motor, which includes the neuro-fuzzy speed controller, can be implemented as shown in Figure 1.

The current controller and PWM modulator block in Figure 1 can be different depending on the types of current controllers to ensure good current regulation. Figure 2a shows the hysteresis band current controller and Figure 2b shows the conventional current controller and PWM modulator unit.

**NEURO-FUZZY CONTROLLERS**

Neuro-fuzzy system can be defined as a neural network structure which is functionally equivalent to fuzzy inference system, which is called ANFIS (adaptive neuro fuzzy inference system) in some literature [6]. Three common neuro-fuzzy architectures based on fuzzy inference system, such as Sugeno, Tsukamoto, and Mamdani ANFIS are used in identification and control of nonlinear systems [12–16]. Furthermore, recurrent ANFIS structure which has self-feedbacks in hidden layers and time delayed ANFIS which has the inputs including delayed components of its outputs are also studied [17]. However, common approach in neuro-fuzzy
control is to use the error and the derivative of the error as neuro-fuzzy inputs.

In this software, two-input first-order Sugeno and Mamdani NFCs which have not recurrence are studied for the speed control of induction motors. For a first order Sugeno ANFIS, a common rule set with two fuzzy if-then rules is the following:

\[
\begin{align*}
    & \text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1, \text{ then } f_1 = p_1 x_1 + q_1 x_2 + r_1 \\
    & \text{if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2, \text{ then } f_2 = p_2 x_1 + q_2 x_2 + r_2
\end{align*}
\]

For the Mamdani ANFIS, a common rule set with two fuzzy if-then rules is the following:

\[
\begin{align*}
    & \text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1, \text{ then } o_1 \\
    & \text{if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2, \text{ then } o_2
\end{align*}
\]

where \( x \) is the neuro fuzzy input vector. Sugeno type neuro fuzzy architecture used in this study is shown in Figure 3.

In this study, three types of NFCs are defined depending on the neuro-fuzzy inputs and outputs. The first type is called PD-NFC which uses the speed error and the derivative of error as inputs and the desired torque current of the motor as output. The second type is called PD-I-NFC which uses the speed error and the derivative of error as inputs and the derivative of the desired torque current of the motor as output. The output of this type of NFC should be integrated to eliminate the steady state error which is seen when the load torque is applied. The third type of controller proposed in this study is called PI-NFC which uses the speed error and the integral of error as inputs and the desired torque current of the motor as output. Thus, the steady state error is eliminated by the same way as in PD-I-NFC, without an external compensator. The layers of the three types of NFCs defined above perform the fuzzification, inference, and defuzzification processes of the fuzzy systems, respectively. The first layer includes the membership functions and calculates the degree of membership functions for the input values. Membership functions are selected as bell and sigmoid functions, and the output of the first

Figure 2  (a) Hysteresis current controller; (b) conventional current controller.

Figure 3  Neuro-fuzzy architecture for Sugeno fuzzy model.
layer \( y_{n1} \) from input \( r \) to output \( i \) can be expressed as
\[
y_{n1}^i = \frac{1}{1 + e^{-a_i(x_n - c_n)}}
\]
where \( a, b, \) and \( c \) are the parameters of the membership functions to be determined by the training. The second layer of the NFC includes the fuzzy rule base and the nodes in this layer represented by \( \Pi \) determine the fuzzy rules. The output of this layer (for three membership functions \( i = 1, 2, 3 \) and \( j = 1, 2, 3 \)) is given as
\[
y_{j2}^i = y_{i1}^j \cdot y_{i2}^j
\]

The third layer is a normalization layer and it computes the certainty of the fuzzy rules. The normalized output of any node in this layer is the division of the certainty of the fuzzy rule to the sum of the rules.
\[
y_{j3}^i = \frac{y_{j2}^i}{\sum_j y_{j2}^i}
\]

The fourth layer gives the certainty of a rule and the output of this layer is the product of the normalized certainty of a rule and the related function. The output of the fourth layer and the function \( f \) (for \( n = 1, 2, \ldots, i^n j \)) are defined as
\[
y_{j4}^i = y_{j3}^i f_{ij}
\]
where \( p, q, \) and \( r \) are the output function parameters of the NFC to be determined by the training of the NFC. The fifth layer is the output layer and produces an output as the summation of incoming signals.
\[
y_{o} = \sum_i \sum_j y_{j4}^i
\]

### Training of the Neuro-Fuzzy Controller

NFC has two groups of parameters to be trained, which are the premise parameters of the membership layer and the consequent parameters of the output layer. In order to obtain the desired control performance, these parameters should be adapted. For the on-line training of the NFC, the speed tracking error in discrete time is
\[
e(k) = \omega^* - \omega(k)
\]
and the cost function to be minimized is
\[
E(k) = \frac{1}{2} e^2(k)
\]
where \( \omega^* \) is the reference speed, \( \omega \) is the actual shaft speed, and \( e \) is the speed error. If \( \eta \) is the learning rate and \( \theta_i \) is any consequent parameter in Equation (15) for the output of the NFC, back propagation learning algorithm can be defined as
\[
\theta_i(k) = \theta_i(k-1) - \eta \frac{\partial E(k)}{\partial \theta_i}
\]

Then, the gradient of the cost function can be determined as
\[
\frac{\partial E(k)}{\partial \theta_i} = \frac{\partial E(k)}{\partial e(k)} \frac{\partial e(k)}{\partial \omega(k)} \frac{\partial \omega(k)}{\partial y_o(k)} \frac{\partial y_o(k)}{\partial \theta_i}
\]
\[
= -e(k) \frac{\partial \omega(k)}{\partial y_o(k)} \frac{\partial y_o(k)}{\partial \theta_i}
\]
where \( \frac{\partial \omega}{\partial y} \) is easily calculated from the Equations (15–16). However, \( \frac{\partial \omega}{\partial y} \) should be calculated using the motor dynamics. The approximate value of this gradient can be used (i.e., discrete derivative of the system output with respect to the input or sign of the discrete derivative). Hence, the local error \( \delta \) at the output of the NFC can be approximated as
\[
\delta = \frac{\partial E(k)}{\partial y_o} \frac{\partial y_o}{\partial \omega} \frac{\partial \omega}{\partial y_o}
\]
\[
= -e(k) \text{sgn} \left( \frac{\omega(k) - \omega(k-1)}{y_o(k) - y_o(k-1)} \right)
\]

In similar way, if the premise parameter vector of the NFC is defined as \( W \), the gradient of the cost function for the parameter \( W_i \) of the membership functions can be written as
\[
\frac{\partial E(k)}{\partial W_i} = \delta \frac{\partial y_o}{\partial y_{j4}^i} \frac{\partial y_{j4}^i}{\partial y_{j3}^i} \frac{\partial y_{j3}^i}{\partial y_{j2}^i} \frac{\partial y_{j2}^i}{\partial \theta_i}
\]

### THE SOFTWARE FOR NEURO-FUZZY CONTROL OF INDUCTION MOTORS

The software is developed using Borland C++ Builder and it works in Windows environment. It is developed to help students improve their understanding of neuro-fuzzy control of induction motors. Using the developed software, induction motor control performance of the various neuro-fuzzy structures can be studied under the parameter and load variations of the motor. Both graphical and numerical results can be observed.
on a PC monitor by choosing appropriate windows. When the program is run, the main window of the software which includes the control structure is seen as shown in Figure 4. Options in each block in Figure 4 and parameter settings sub-windows are come up by clicking the related block symbol. After completing all of the block settings, simulation options such as simulation time, switching frequency of the inverter, and load torque of the motor, which are shown at the left bottom of the main window, are defined and then the simulation can be started. Sampling frequency of the current control loops is equal to switching frequency of the inverter. Hence, speed/current sampling ratio in the simulation options of the main window will determine the sampling frequency of the speed control loop of the motor.

When the blocks on the main window (which are speed controller, current controller, PWM generator, motor, and graphics blocks) are selected, options of the selected block and related parameter settings are seen in sub-windows. Initially, motor parameters shown in the window in Figure 5 can be given, which is obtained by choosing the motor symbol. Default button in this window loads the motor parameters used in this study.

PWM generation block, as shown in Figure 6a, includes three buttons showing the current control and PWM generation method of the induction motor drives.

Tolerance band should be given when the hysteresis band current controller and PWM generation is selected. However, conventional PI current controllers should be designed for sinusoidal and space vector PWM generation as shown in Figure 6b. Current controller outputs are limited to avoid the excessive voltages at the terminals of the motor which is the case encountered in the practical applications.
Figure 6  (a) PWM generation block and submenus. (b) Current controller block and related settings.

Figure 7  (a) Speed controller block and PI controller design; (b) neuro fuzzy controller selection.

Figure 8  Neuro fuzzy controller setup window.
Instead of speed dependent field weakening, d-axis reference current ($i_{dref}$) can be changed by the user to see the effects of the d-axis current on the motor speed and torque.

After defining the motor parameters, current controller and PWM generation settings, the speed controller block can be designed. Speed reference and torque current limits are defined by clicking the related buttons as shown in Figure 7a. PI control of induction motor is also included for the purpose of comparison. When the NFC is selected, a popup message as in Figure 7b is seen. Default button in this window permits the user to simulate the system using the NFC parameters obtained by our simulation studies. Training button permits the user to train the NFC and then simulate the system using random initial NFC parameters.

Both selections (default or training) in Figure 7 give the neuro fuzzy controller setup window in Figure 8. However, only the upper half of the NFC setup window is active when the default button is selected, since the default NFC parameters are used. If

![Figure 9](image_url)  
**Figure 9** Training performance of the NFC at the end of the training epoch.

![Figure 10](image_url)  
**Figure 10** Consequent parameters of the trained neuro-fuzzy controller.
the training button in Figure 7b is selected, all buttons in Figure 8 will be active. NFC setup window is used to select the type of NFC and pattern or batch learning mode. Furthermore, the training parameters to train the NFC such as learning rate, training epoch, the number of training patterns, and desired minimum error as stopping criteria can also be defined. Then, the training can be initialized. Training performance of the NFC can be monitored at the end of every training epoch as shown in Figure 9.

If the training performance of the NFC depending on the mean square error and tracking error of sinusoidal reference is acceptable, then the training can be stopped, otherwise it can be continued for an additional epoch. After the completion of the training, consequent parameters of the trained NFC are seen numerically as in Figure 10.

After the parameter settings and the training of the NFC are completed, the system is ready to simulate using the start simulation button on the main window. If the graphics block on the main window is selected, the lists of the variables which can be monitored graphically are seen as in Figure 11.

Although many different simulation results can be obtained using the developed software, some simulation examples are given in Figures 12–14, in order to monitor the speed control performance of the various controllers. Speed tracking performance of the conventional PI controller designed by trial and error is given in Figure 12a for step speed reference. There is an overshoot of approximately 10% and small deviations from the reference are observed where the load torque is applied and removed. Figure 12b shows the reference and actual torque current variation with the hysteresis current controller.

Speed tracking performance of the PD type NFC is given in Figure 13a for step speed reference. There is no overshoot; however, there is steady state error where the load torque is applied. Figure 13b shows the three phase stator currents of the motor. Speed tracking performance of the PI type NFC proposed in this study is given in Figure 14a for step speed reference. There is an overshoot of approximately 3%, there is no steady state error; however, very small deviations are observed where the load torque is applied.

Figure 11 The list of the graphically monitored variables.
applied and removed. Figure 14b shows the reference q-axis current produced by speed controller and actual current with the hyteresis current controller.

CONCLUSION

In this article, an educational tool developed for the neuro-fuzzy control of induction motors is described. The main aim is to help students improve their understanding of neuro-fuzzy control of electrical drives. The tool has a flexible structure and graphical user interface. It enables the user to design neuro-fuzzy system and change the motor parameters. Motor and controller variables can be monitored for various NFCs under the various load conditions.

REFERENCES

Neural Networks 6 (1995), 144–156.
[8] K. S. Narendra and K. Parthasarthy, Gradient methods for the optimization of dynamical systems containing
[16] A. Rubai, D. Ricketts, and D. Kanham, Development and implementation of an adaptive fuzzy-neural net-
[18] C. Elmas and M. A. Akcayol, Virtual electrical machinery laboratory: A fuzzy logic controller for
[19] M. A. Akcayol, C. Elmas, O. A. Erdem, and M. Kurt, An educational tool for fuzzy logic controller and
(2002), 1500–1505.
[22] H. Guldemir and Z. H. Akpolat, Modelling of induction motor dynamic behaviour using mathcad,
[23] M. Gokbulut and A. Tekin, An educational tool for

BIOGRAPHIES

Muammer Gökbulut received the BSc and MSc degrees in electrical and electronics education from Gazi University, Ankara, Turkey, in 1980 and 1988, respectively, and the PhD degree in electronics engineering from Erciyes University, Kayseri, Turkey, in 1998. He is currently a professor in the Department of Electronics and Computer Education, Faculty of Technical Education, Fırat University, Elazığ, Turkey. His research interests include research and development of intelligent control systems (neural networks and fuzzy logic), electrical drives control, and adaptive control systems.

Besir Dandil received the BSc, MSc, and PhD degrees in electrical and electronic engineering from Fırat University, Elazığ, Turkey, in 1992, 1998, and 2004, respectively. He is currently an assistant professor at the same university. His research interests include high performance control of induction motors and identification and control of nonlinear systems including artificial neural network and fuzzy control.

Cafer Bal received the BSc and MSc degrees in electronic and computer education from Firat University, Elazığ, Turkey, in 1997 and 2002, respectively. He is currently working towards to the PhD degree in electrical and electronic engineering and is a lecturer at the same university. His research interests include speed sensorless control of electrical drives, artificial neural networks and fuzzy control, educational tools, and virtual laboratories.