



Virtual lab for artificial intelligence controllers based speed control for induction motor

Serdar Paçacı¹, Okan Bingöl^{2*}

¹ Suleyman Demirel University, Department of Information Technologies Isparta, Turkey

² Applied Sciences University of Isparta, Faculty of Tecnology, Department of Electrical-Electronic Engineering, Isparta, Turkey

(İlk Geliş Tarihi 13 Temmuz 2018 ve Kabul Tarihi 24 Ekim 2018)

(DOI: 10.31590/ejosat.443601)

Abstract

Practical implementation has an important role in engineering education. Practical implementations, however, may not be possible in some situations, such as lack of physical possibilities, the presence of situations that may create risks during implementation, or place-time dependence. So, package programs are developed for the virtual practical implementation experience. On the other hand, these tools may not be flexible and interactive enough for all branches of science. Therefore, in this study a virtual laboratory tool was developed for the speed control of an induction motor fed by a three-level inverter. The user can select proportional-integral, proportional–integral–derivative, fuzzy logic, artificial neural network, and neuro-fuzzy controllers for the speed controller. Different working conditions for the induction motor can be simulated and the outcomes can be observed by the users. The virtual laboratory had a flexible interface and it was written on Microsoft Visual Studio 2015 IDE using C# programming language on Windows Presentation Foundation infrastructure.

Keywords: Virtual laboratory, multilevel inverter, neuro-fuzzy control, space vector modulation.

* Corresponding author : Suleyman Demirel University, Faculty of Tecnology, Department of Electrical-Electronic Engineering, Isparta, Turkey, okanbingol@sdu.edu.tr phone: +90 555 555 55 55

1. Introduction

With the advance of technology, modern styles of lives of people have changed. As a result of this advancement, people have started to solve problems more effectively [1]. Engineering education, which has started at university, comprises lifelong learning and progress of practical application. Essential factors of this education are mathematics, physics, experiments and practical applications [2].

Due to rapid progress and complicated nature of high-tech devices, operating laboratory has increased the cost and decreased flexibility [3]. Because of these reasons, many engineers carry out effective experiments by using information technology and simulation software. Virtual laboratory simulation software's have many advantages; for example, they save money, provide flexibility, multi access, alteration in system configuration to users [4, 5]. Hence, these applications are very useful [6–15].

Induction motors are being used more commonly in industry compared to other electrical motors due to their strong and reliable structures, low costs and high inertia momentum, ease of control, low maintenance requirements and ability to work in dirty and dangerous environments [16]. In high performance induction motor drivers, indirect field-oriented control is being commonly used. In this control method, the variables should be known correctly [17]. In order to control a system, a suitable mathematical model of the whole system should be derived. Nonlinear systems cannot be completely mathematically modeled [18]. For this reason, fuzzy logic (FL), artificial neural networks (ANN), neuro-fuzzy (NF) and other similar smart control technics are powerful alternatives and they are being commonly used [19, 20].

Fuzzy Logic is based on the fuzzy set theory, fuzzy if–then rules, and fuzzy reasoning. Fuzzy logic implements a nonlinear mapping from its input space to output space by a number of if–then rules [6, 18, 21]. ANN has the ability to learn, generalize and being adaptive. ANN has parallel and distributed information processing structures that were developed in inspiration with the human brain, which are connected to each other via weighted connections [22, 23]. Fuzzy variables can be adjusted by the expert via trial and error. Recently, these variables have been adjusted via ANN. This control type is usually named NF Controller (NFC).

Recent years, thanks to rapid progress in capacity and frequency of semiconductor power devices, a lot of changes have occurred in the systems of industrial motor driver [24]. In consequence of these changes, multilevel voltage source inverters have emerged [25]. Multilevel inverters are used in applications requiring high power and high voltage. When multilevel inverters are compared with two-level inverter at the same switching frequency, harmonic components' amplitude is low [26, 27].

In this study, a virtual laboratory software was developed which the speed control of an IM which is being fed via a three-level inverter was done via PI, PID, FL, ANN and NF controllers. SVPWM technique was used in the switching of a three-level inverter. The software was coded using C# programming language in Microsoft Visual Studio 2015 environment. The virtual laboratory software has a flexible structure and a graphical user interface. The motor and controller parameters of the driver can be easily changed and the circuit

responses to different working conditions can be observed via graphics.

2. Indirect field-oriented control of the IM

Indirect field-oriented control method is commonly used in high performance induction motor driver systems. Using this method, angular synchronous velocity of rotor flux vector can be obtained based on the summation of real angular velocity and calculated angular slip velocity [28]. Voltage equation of motor is given in equation 1, electromagnetic torque is given in equation 2, angular slip speed is given in equation 3 and the slip position of the rotor is given in equation 4.

$$\begin{bmatrix} v_d \\ v_q \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_s + L_s p & -\omega_e L_s & L_m p & -\omega_e L_m \\ \omega_e L_s & R_s + L_s p & \omega_e L_m & L_m p \\ L_m p & s\omega_e L_m & R_r + L_r p & -s\omega_e L_r \\ s\omega_e L_m & L_m p & s\omega_e L_r & R_r + L_r p \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{bmatrix} \quad (1)$$

$$T_e = \frac{3}{2} \frac{L_m}{L_r} (\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}) \quad (2)$$

$$\omega_{sl} = \omega_e - \omega_r = \frac{L_m R_r i_{qs}}{L_r \lambda_{dr}} \quad (3)$$

$$\theta_{sl} = \theta_e - \theta_r \quad (4)$$

Where V_d, V_q are voltages, i_{ds}, i_{qs} are stator currents, i_{dr}, i_{qr} are rotor currents in the stationary, R_s, R_r are stator and rotor winding resistances, L_s, L_r are stator and rotor inductance, L_m is mutual inductance, $\lambda_{dr}, \lambda_{qr}$ is magnetic flux of the rotor, ω_{sl} is slip angular speed, ω_e is synchronous angular speed and ω_r slip rotor angular speed, θ_{sl} is slip position, θ_e is synchronous position, θ_r is rotor position and p derivative operator. Basic structure of indirect field-oriented control of IM supplied by three-level inverter is given in Figure 1.

3. Three-Level Svpwm Inverter

Three-phase systems can be transferred into two-phase plane α - β of 90° phase differences by using space vector method. SVPWM modulation is based on space vector representation of voltages in the plane of α - β [29]. This method can also be used in the control of three-phase voltage-source inverters. Three-phase voltages produced by the inverter are shown as space vector reference voltage V_{ref} on the α - β plane. This case is expressed mathematically on equation 5.

$$V_{ref} = V_\alpha + jV_\beta = \left(\frac{2}{3}\right)(V_a e^{j0} + V_b e^{j2\pi/3} + V_c e^{-j2\pi/3}) \quad (5)$$

The basic circuit diagram of the classical three-phase three-level voltage-source inverter is shown in Figure 2. Three switching variables (a, b and c) can be assigned to the inverter. There are total of ($n^3 = 3^3 = 27$) switching states in three-phase three-level inverter as each phase (leg) has three switching states. As seen in Figure 3, voltage vectors are positioned in various points of two nested hexagons and divided into four groups according to switching their states. These are (a, b, c), (ap, an, bp, bn) and (op, oo, on) voltage vectors. In Figure 3, hexagon has six sectors (A, B, C, D, E, F). In the same time,

there are four regions (1, 2, 3, 4) in each sector. The total number of regions in the hexagon is 24 [30, 31].

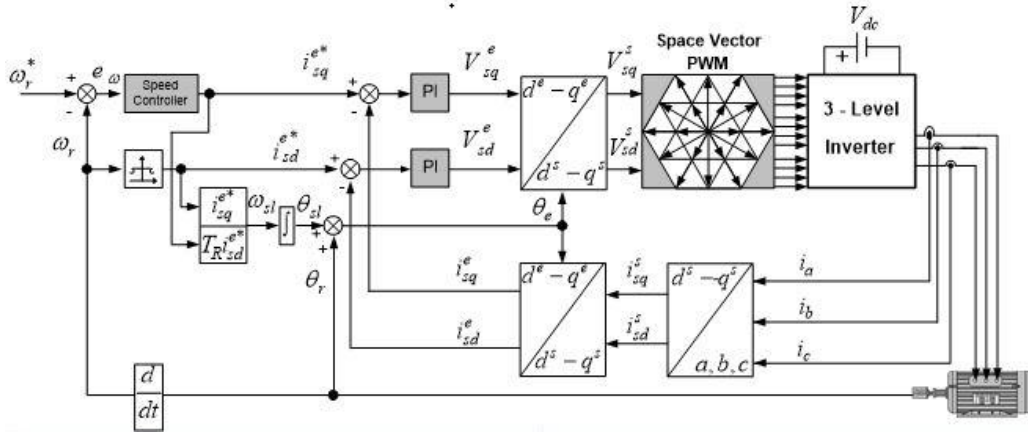


Figure 1. Indirect field-oriented control of IM.

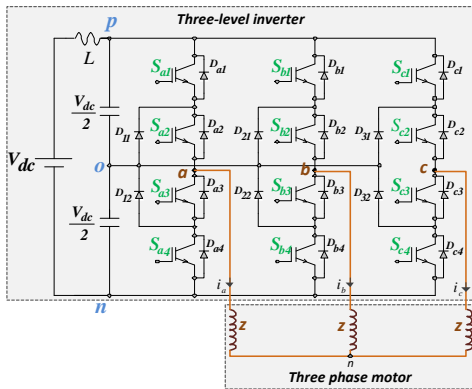


Figure 2. Three-phase three-level diode clamped voltage-source inverter.

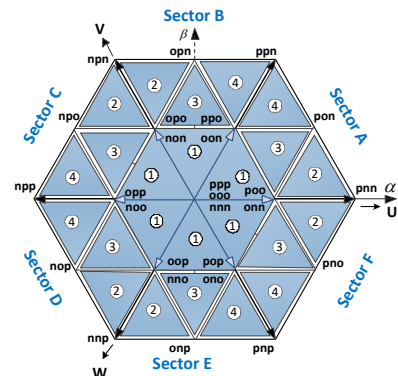


Figure 3. Switching states of three-level PWM inverter.

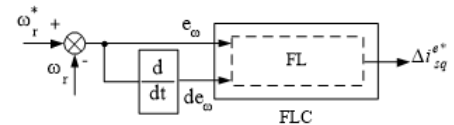
4. Fuzzy logic, artificial neural network and neuro-fuzzy controllers

4.1 Fuzzy Logic Controller

Fuzzy logic was inspired and developed by the human reasoning and decision-making capability [32]. Today, it has found a very wide range of usage in different areas. A FLC consists of four units including a fuzzification, fuzzy inference, knowledge base and a defuzzification unit [14]. The block diagram of FL is shown in Figure 4a. The FL speed controller block diagram of the IM is shown on Figure 4b.



Figure 4. a) FL block diagram



b) FL speed controller block diagram.

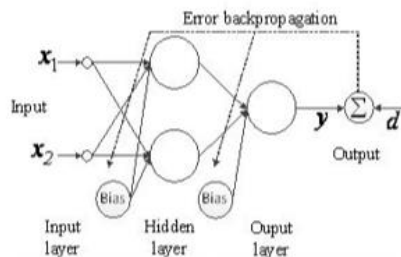
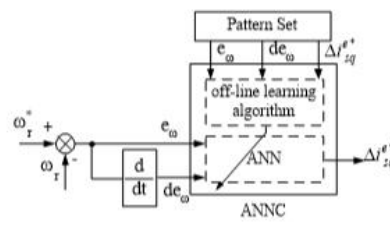


Figure 5. a) A multilayer feed-forward ANN with one hidden layer



b) ANN speed controller block diagram.

$$y_0^{IV} = t_0^{IV} = \frac{a}{b} \quad (11)$$

4.3 Neuro-fuzzy Controller

Neuro-Fuzzy concept emerged by combining the FL controller's decision-making ability and the ANNs learning ability. Via this way, the systems controlled by FL obtained the learning ability of the ANN while ANNs obtained the decision-making ability of FL controllers [34]. The NF controller presented in Figure 6 comprises 4 layers.

The first layer is the input layer. In this layer, the weights are unity and fixed. The weight values in this layer are not changing [35, 36].

The second layer is the membership layer and the nodes on this layer represent the membership functions on the FL. Membership functions are being chosen as Gaussian membership functions. To find the input values of the nodes on the second layer, the equality shown in equation 6 is being used while the equality shown in equation 7 is being used to show the output value. In this equality, m shows the center of the membership function, and σ shows the width of the membership function or its standard deviation.

$$t_{ij}^{II} = -\frac{(x_{ij}^{II} - m_{ij}^{II})^2}{(\sigma_{ij}^{II})^2} \quad (6)$$

$$y_{ij}^{II} = \exp(t_{ij}^{II}) \quad (7)$$

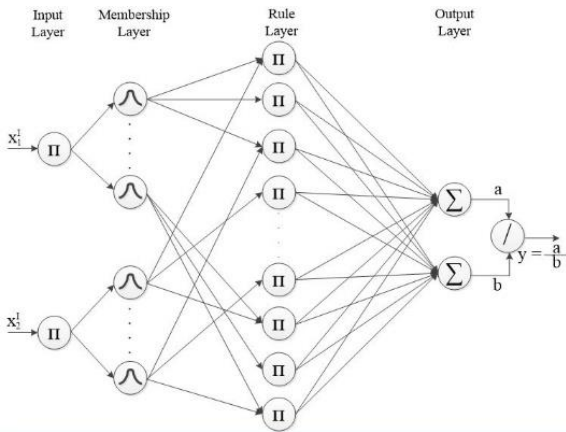


Figure 6. Neuro-fuzzy block diagram.

The third layer is the layer on which the rule base for the FL controller will be adjusted. Each node on this layer represents a rule. The smaller one of the values arriving at the nodes on the second layer is taken as nodes input expression. The expression shown in equation 8 is used to calculate the output values of the nodes on layer three.

$$y_{jk}^{III} = t_{jk}^{III} = \min(x_{1j}^{III}, x_{1k}^{III}) \quad (8)$$

The fourth layer includes the defuzzification and deduction procedure used in FL controllers. For defuzzification, weight center method was used. Parameters a and b in the block diagram shown in Figure 6 are given in equation 9 and 10. The output of the fourth layer is as shown in equation 11.

$$a = \sum_{j=1}^n \sum_{k=1}^n w_{jk}^{IV} y_{jk}^{III} \quad (9)$$

$$b = \sum_{k=1}^n y_{jk}^{III} \quad (10)$$

The main purpose in using NFC is to adjust the parameters of the FL controller. On the second layer of NFC, the center and width values of the Gaussian membership function that is used as input membership functions are adjusted. Similarly, the output membership functions and rule base on the third layer are being adjusted [35]. For the training of NFC, the error value is expressed by E . Its calculation is shown in equation 12. In the equality, d represents the desired output, and y represents the network output.

$$E = \frac{(d - y)^2}{2} = \frac{e^2}{2} \quad (12)$$

The training is conducted from the fourth layer to the first layer. Since the output of the fourth layer is the output of the network, an error value is found on this layer. The propagation value is represented by δ_0^{IV} and its calculation is shown in equation 13.

$$\delta_0^{IV} = -\frac{\partial E}{\partial t_0^{IV}} = -\frac{\partial E}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial a} \frac{\partial a}{\partial t_0^{IV}} = \frac{e}{b} \quad (13)$$

The change calculation due to the error values of the weights are expressed as shown in equation 14.

$$\Delta w_{jk}^{IV} = -\frac{\partial E}{\partial w_{jk}^{IV}} = -\frac{\partial E}{\partial t_0^{IV}} \frac{\partial t_0^{IV}}{\partial w_{jk}^{IV}} = \delta_0^{IV} y_{jk}^{III} \quad (14)$$

Arrangement of the weights is as shown in equation 15. The l symbol in equation 15 shows the sample index and μ_w show the learning ratio.

$$w_{jk}^{IV}(l) = w_{jk}^{IV}(l-1) + \mu_w \Delta w_{jk}^{IV}(l) \quad (15)$$

Since there is no weight value calculation in the third layer, only the error value coming from the fourth layer is being calculated. Equation 16 shows how to calculate δ_{jk}^{III} .

$$\delta_{jk}^{III} = -\frac{\partial E}{\partial t_{jk}^{III}} = -\frac{\partial E}{\partial t_0^{IV}} \frac{\partial t_0^{IV}}{\partial y_{jk}^{III}} \frac{\partial y_{jk}^{III}}{\partial t_{jk}^{III}} = \delta_0^{IV} (w_{jk}^{IV} - y) \quad (16)$$

The calculation of the error value δ_{ij}^{II} on the second layer is shown in equation 17.

$$\delta_{ij}^{II} = -\frac{\partial E}{\partial t_{ij}^{II}} = -\frac{\partial E}{\partial t_{jk}^{III}} \frac{\partial t_{jk}^{III}}{\partial y_{ij}^{II}} \frac{\partial y_{ij}^{II}}{\partial t_{ij}^{II}} = \sum_{j,k} \delta_{jk}^{III} y_{jk}^{III} \quad (17)$$

On the second layer, the adjustment of both the center and width of the same membership function is done. Equation 18 shows the change of the center value, equation 19 shows the change of the width value.

$$\Delta m_{ij}^{II} = -\frac{\partial E}{\partial m_{ij}^{II}} = -\frac{\partial E}{\partial t_{ij}^{II}} \frac{\partial t_{ij}^{II}}{\partial m_{ij}^{II}} = \delta_{ij}^{II} \frac{2(x_i^{II} - m_{ij}^{II})}{(\sigma_{ij}^{II})^2} \quad (18)$$

$$\Delta \sigma_{ij}^{II} = -\frac{\partial E}{\partial \sigma_{ij}^{II}} = -\frac{\partial E}{\partial t_{ij}^{II}} \frac{\partial t_{ij}^{II}}{\partial \sigma_{ij}^{II}} = \delta_{ij}^{II} \frac{2(x_i^{II} - m_{ij}^{II})^2}{(\sigma_{ij}^{II})^2} \quad (19)$$

Calculated change values and center and width values are calculated as shown in equation 20 and 21. In the equations, μ_m shows the learning ratio for center value and μ_σ shows the learning ratio for width value.

$$m_{ij}^{II}(l) = m_{ij}^{II}(l-1) + \mu_m \Delta m_{ij}^{II}(l) \quad (20)$$

$$\sigma_{ij}^{II}(l) = \sigma_{ij}^{II}(l-1) + \mu_\sigma \Delta \sigma_{ij}^{II}(l) \quad (21)$$

Figure 7 shows the NF speed controller block diagram that belongs to the IM.

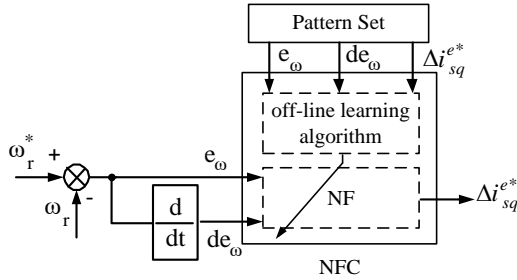


Figure 7. Neuro-fuzzy speed controller block diagram.

All controller input coefficients are defined as the speed error e_ω between the reference speed ω_r^* and the actual motor speed ω_r and as the change in speed error de_ω . Error of speed e_ω is calculated as given in equation 22. Change in speed error is calculated as given in equation 23. Output variable is chosen as the current moment component. The differential expression of the speed controllers in the computer application is given in equation 24.

$$e_\omega(t) = \omega_r^*(t) - \omega_r(t) \quad (22)$$

$$de_\omega(t) = e_\omega(t) - e_\omega(t-1) \quad (23)$$

$$I_{sq}^{e*}(t) = I_{sq}^{e*}(t-1) + \Delta I_{sq}^{e*}(t) \quad (24)$$

The reference speed required at the $\omega_r^*(t)$ t'th sampling time is the motor speed at the $\omega_r(t)$ t'th sampling time, and the motor speed error de_ω at the e_ω t'th sampling time is the change in motor speed error at the t'th sampling time and $e_\omega(t-1)$ is the change at the motor speed error at the (t-1)'th sampling time.

5. Virtual Laboratory

The main window of the program is shown in Figure 8. Above the main window, "File", "Options", "Language" and

"Help" menus are placed. Below the File menu, there are sub-menus that enable transferring the data obtained from the program to outside the program or transferring data created outside the program to the program. In the Options menu, the main parameters of the program can be arranged. In the Language menu, Turkish and English can be chosen by the user. The Help menu provides information regarding how to use the program.

The main window consists of two parts. This part provides the schematics for a diode-clamped three-level inverter circuits. On the circuit chart, which semiconductor power keys will be conducting and which ones will be cutting off during the simulation will be shown using different colors. Also, there is the SVPWM graphic which shows the reference voltage vector and PWM signals during the simulation. Furthermore, in this section, PI, PID, FL, ANN and NF options for choosing the controller and simulation control keys can be found.

On the right of the main window; there is an interface where the IM and inverter parameter values can be entered, motor's reference speed signal and the load values to be applied to the motor can be arranged. For the reference speed signal, there can be two choices: stable and variable. Furthermore, there is a similar option to choose how to load or unload the motor.

Two different graphics windows were prepared for the simulation graphics window. For the first window, as shown in Figure 9a, PI, FL, ANN and NF controllers' simulation graphics window is shown. In this window, the current, voltage, reference speed – real speed and moment graphics of the motor can be found in their respective order. The second simulation window, as can be seen in Figure 9b, is prepared for the FL controller. In this window reference speed – real speed, motor's current graphics, speed-error-input-membership-function usage, speed-error-change-input-membership-function usage, defuzzification and rule base usage graphics can be found, respectively.

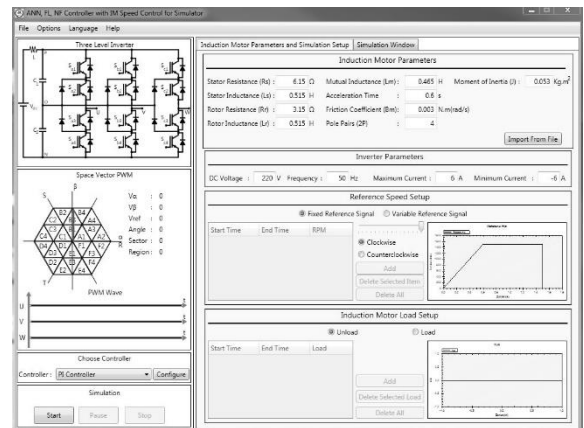


Figure 8. Main window.

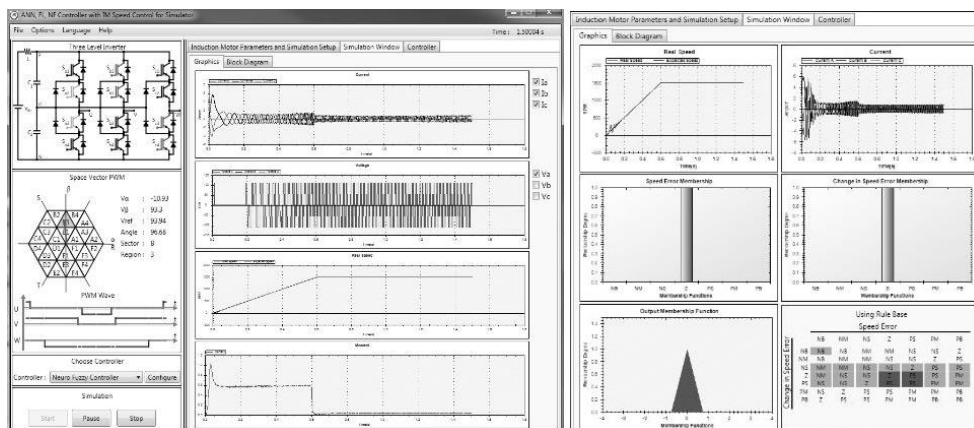


Figure 9. a) Simulation graphics window

b) FL simulation graphics window.

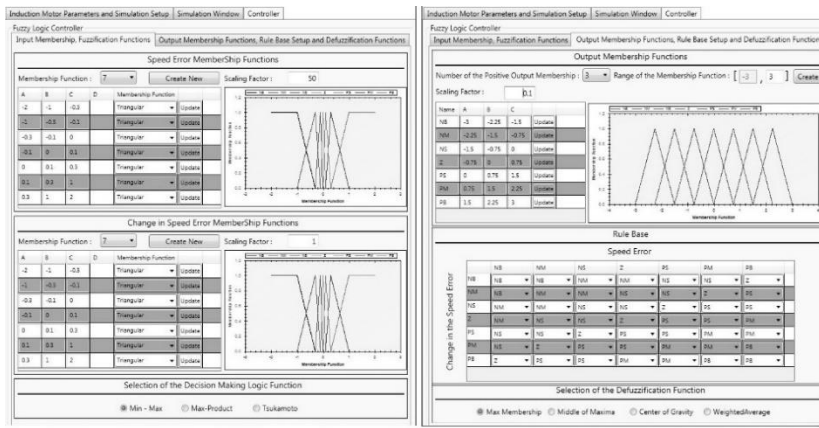


Figure 10. FL a) Input parameters b) Output parameters.

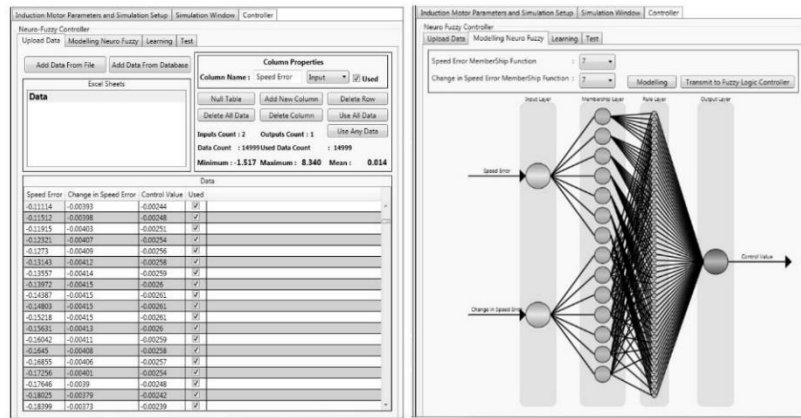


Figure 11. NF a) Data input window b) Modeling window.

Figure 10a shows the window designed to adjust the FL controllers input membership functions and choosing fuzzification method. For each membership function, four different membership function type which is triangle, trapezoid, gaussian and sigmoid.

Scaling factor parameters of the input membership functions are defined in this screen as well. Finally, in this section three different fuzzification methods that are Min-Max, Max-Product and Tsukamoto can be chosen. For output membership function of FL controller, adjustment of the rule base and choosing the defuzzification method, the window seen in Figure 10b is designed. In this window, the output membership functions are created and rule base is adjusted. Also, either one of maximum membership, average of the maximums, weight center and weight average is chosen as the defuzzification method.

To adjust the NF controller, the data input window shown in Figure 11a is used. The “Add Data from File” button on this window is used to obtain training and/or testing data from files with txt, csv, xls and xlsx or from Microsoft Access data base. It is possible to transfer data from MS Sql Server to the program via the “Add Data from Database” button. The columns in the transferred data can be erased or additional columns can be added, the lines can be erased or new lines can be added. Furthermore, the data to be used in the training or in the tests can be chosen by the user. After the last column of the added data, a column named “Used” is automatically added. The value of this column expresses whether this data will be used in the training or not. It is not necessary to add data from outside the program. “Null Table” but-ton enables creation of training and/or test data over the program.

After the data input is done, by using the window shown in Figure 11b, NF’s modelling is done. By using the “Model” button, the NF controller is modeled and is showed graphically. “Transmit to Fuzzy Logic Controller” button transfers all adjusted parameters as a result of the training to the FL controller.

After the NF controller is modeled, the training procedure is carried out via the maximum iteration number and MSE (Mean Square Error) value chosen by the user. Figure 12a shows the window designed for the training of the NF controller. If the MSE value was not decreased as targeted at the end of the training, “Resume Learning” option is selected and “Start Learning” button is clicked to resume training with the parameters from the previous training. When the training is competed using the learning window designed for the NF controller, when the modelling window is returned to, the shape of the membership function with respect to the center and width values in the nodes that are located on the second layer and show the input membership functions in the FL controller is shown to the user. Furthermore, the weight values of the connections between the rule layer which is the third layer and output layer which is the fourth and the last layer is shown to the user when hovered over with the cursor of the mouse. And after the training stage, the response of the NF controller to the training and test data and error values can be seen in the test screen shown in Figure 12b.

Menus that are similar to the NF controller are pre-pared for the ANN controller as well. Data input procedure, which is the first step in adjusting the ANN controller is done via the screen that is designed in the same way as the data input menu of the NF controller. For the second step, the menu shown in Figure 13

is used to model ANN. Here the user can add as many hidden layers as he/she likes and can also add as many cells to each

hidden layer added. For each layer, either one of the sigmoid and bipolar sigmoid or linear activation functions can be chosen.

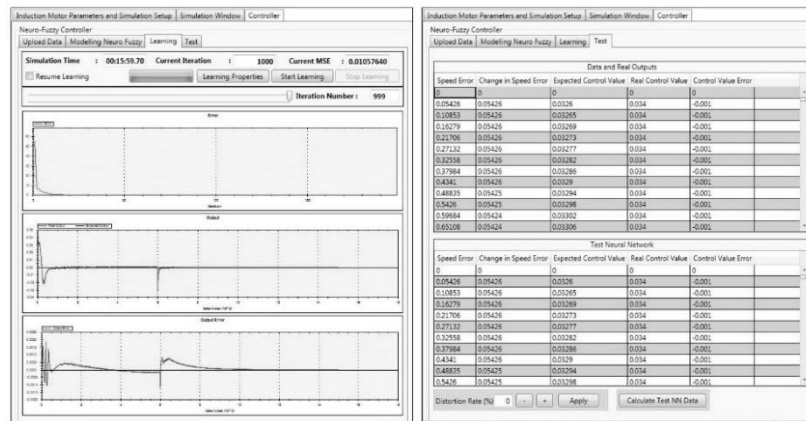


Figure 12. NF NF a) Training window

b) Test window.

After the ANN controller is modeled, the training procedure is carried on by using the user chosen maximum iteration number, MSE (Mean Square Error) value, learning coefficient and momentum coefficient. By using the training menu designed for the ANN controller, after the training is done, when the modelling screen is returned to, the nodes show the parameter values of the nodes when hovered over with the cursor of the mouse, and the weight connections show the weight values in the same way. After this stage, the user can test the trained ANN by using the “Test” tab.

The authors declare that they have no conflict of interest.

References

- [1] Deperlioglu, Ö., Köse, U. 2011. An educational tool for artificial neural networks. *Computers & Electrical Engineering*, 37(3): 392–402, 2011. DOI:10.1016/j.compeleceng.2011.03.010
- [2] Sevgi, L. 2006. Modeling and simulation concepts in engineering education: virtual tools. *Turkish Journal of Electrical Engineering & Computer Sciences*, 14(1): 113–127, 2006.
- [3] Öztürk, N., Çelik, E. 2014. An educational tool for the genetic algorithm-based fuzzy logic controller of a permanent magnet synchronous motor drive. *International Journal of Electrical Engineering Education*, 51(3): 218–231, 2014. DOI:10.7227/ijeee.51.3.4
- [4] Potkonjak V., Gardner, M., Callaghan, V., Mattila, P., Guetl, C., Petrović, VM., Jovanović, K. 2016. Virtual laboratories for education in science, technology, and engineering: A review. *Computers & Education*, 95: 309–327, 2016. DOI:10.1016/j.compedu.2016.02.002
- [5] Keyhani, MN., Marwali, LE., Higuera, G., Athalye, G., Baumgartner. 2002. An integrated virtual learning system for the development of motor drive systems. *IEEE Trans.on Power Systems*, 17(1): 1–6, 2002. DOI:10.1109/59.982185
- [6] Köse, U., Deperlioglu, Ö. 2015. FL-LAB v2: Design and Development of an Easy-to-Use, Interactive Fuzzy Logic Control Software System. *Applied Mathematics & Information Sciences*, 9(2): 883–897, 2015. DOI:10.12785/amis/090237
- [7] Avouris, NM., Tselios, N., Tatakis, EC. 2001. Development and evaluation of a computer-based laboratory teaching tool. *Computer Applications in Engineering Education*, 9(1): 8–19, 2001. DOI:10.1002/cae.1001
- [8] Kayışlı, K., Tuncer, S., Poyraz, M. 2013. An educational tool for fundamental DC–DC converter circuits and active power factor correction applications. *Computer Applications in Engineering Education*, 21(1): 113–134, 2013. DOI:10.1002/cae.20455
- [9] Yigit, T., Elmas, Ç. 2008. An educational tool for controlling SRM. *Computer Applications in Engineering Education*, 16(4): 268–279, 2008. DOI:10.1002/cae.20148
- [10] Koku, AB., Kaynak, O. 2001. An internet-assisted experimental environment suitable for the reinforcement on undergraduate teaching of advanced control techniques,

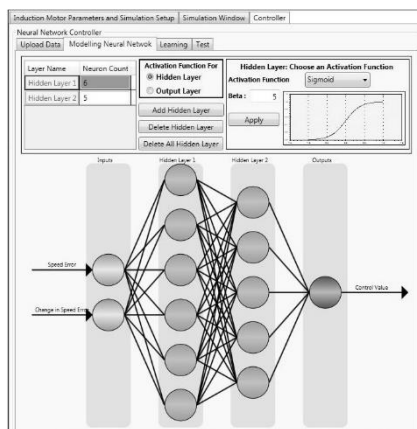


Figure 13. ANN modeling window

6. Conclusions

In this study, a virtual laboratory set was developed for the speed control of an IM being fed by three-level inverter. As a speed controller, PI, PID, FL, ANN and NF controllers were provided to the user to choose from. Before the user works with this set, he/she needs to have theoretical information regarding IM, SVPWM and speed controllers. But no detailed information; regarding how to use the set, is required before using the set. So, the users do not waste time in learning how to use the set. The set has a flexible graphics interface. The user can change the IM, inverter and controller parameters with ease and simulate IM's working under different working conditions. The set can be installed under Windows based operating systems (XP SP3 and later) and operate.

Acknowledgements

This study was funded by SDUBAP-2465-YL-10

Conflict of Interest

- IEEE Trans. on Education, 44(1): 24–28, 2001. DOI: 10.1109/13.912706
- [11] Altas, IH., Aydar, H. 2008. A real time computer controlled simulator for control systems. *Computer Applications in Engineering Education*, 16(2): 115–126, 2008. DOI:10.1002/cae.20130
- [12] Gökbulut, M., Bal, C., Dandıl, B. 2006. A virtual electrical drive control laboratory: neuro–fuzzy control of induction motors. *Computer Applications in Engineering Education*, 14(3): 211–221, 2006. DOI:10.1002/cae.20130
- [13] Bingöl, O., Paçacı, S. 2012. A virtual laboratory for neural network controlled DC motors based on a DC-DC buck converter. *The International Journal of Engineering Education*, 28(3): 713–723, 2012.
- [14] Bingöl, O., Paçacı, S. 2010. A virtual laboratory for fuzzy logic controlled DC motors. *International Journal of Physical Sciences*, 5(16): 2493–2502, 2010.
- [15] Sobczuk, DL. 2007. Internet based teaching of pulse width modulation for three-level converters. *EUROCON The International Conference on “Computer as a Tool” Warsaw*, September 9-12: 2479–2484, 2007. DOI:10.1109/eurcon.2007.4400592
- [16] Boldea İ., Nasar, SA. 1992. *Vector control of AC drives*. CRC Press, New York.
- [17] Wai RJ., Chang, HH. 2004. Backstepping wavelet neural network control for indirect field-oriented induction motor drive. *IEEE Trans. on Neural Networks*, 15(2): 367–382, 2004. DOI:10.1109/tnn.2004.824411
- [18] Akçayol, MA., Çetin, A., Elmas, Ç. 2002. An educational tool for fuzzy logic-controlled BDCM. *IEEE Trans. on Education*, 45(1): 33–42, 2002. DOI:10.1109/13.983219
- [19] Kaiser, MS., Chowdhury, ZI., Al Mamun, S., Hussain, A., Mahmud, M. 2016. A neuro-fuzzy control system based on feature extraction of surface electromyogram signal for solar-powered wheelchair. *Cognitive Computation*, 8(5), 946-954, 2016. DOI:10.1007/s12559-016-9398-4
- [20] Sun, J., Li, Z. 2015. Development and Implementation of a wheeled inverted pendulum vehicle using adaptive neural control with extreme learning machines. *Cognitive Computation*, 7(6), 740-752, 2015. DOI:10.1007/s12559-015-9363-7
- [21] Öztürk, N., Çelik, E. 2012. Speed control of permanent magnet synchronous motors using fuzzy controller based on genetic algorithms. *International Journal of Electrical Power & Energy Systems*, 43(1): 889–898, 2012. DOI:10.1016/j.ijepes.2012.06.013
- [22] Weerasooriya, S., El-Sharkawi, M. 1991. Identification and control of a dc motor using back-propagation neural networks. *IEEE Trans. Energy Conversion*, 6(4): 663–669, 1991. DOI:10.1109/60.103639
- [23] Narendra, KS., Parthasarathy, K. 1990. Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks*, 1(1): 1–27, 1990. DOI:10.1109/72.80202
- [24] Lee YH., Suh, BS., Hyun, DS. 1994. A novel PWM scheme for a three-level voltage source inverter with GTO thyristors. *IEEE Trans. on Industry Applications*, 32(2): 260–268, 1994. DOI:10.1109/ias.1994.377573
- [25] Lai, JS., Peng, FZ. 1996. Multilevel converters-A new breed of power converters. *IEEE Trans. on Industry Applications*, 32(3): 509–517, 1996. DOI:10.1109/ias.1995.530601
- [26] Nabae, A., Takahashi, I., Akagi, H. 1981. A new neutral-point-clamped PWM inverter. *IEEE Trans. on Industry Applications*, IA-17: 518–523, 1981. DOI:10.1109/tia.1981.4503992
- [27] Rodriguez, J., Lai, JS., Peng, FZ. 2002. Multilevel inverters: A survey of topologies, controls, and applications. *IEEE Trans. on Industrial Electronics*, 49(4): 724–738, 2002. DOI:10.1109/tie.2002.801052
- [28] Lin, BR., Lu, HH. 1999. Multilevel AC/DC/AC converter for AC drives. *Electric Power Applications, IEE Proceedings*. 146(4): 397–406, 1999. DOI:10.1049/ip-epa:19990253
- [29] VanDer Broeck, HW., Skudelny, HC., Stanke, GV. 1988. Analysis and realization of a pulse width modulator based on voltage space vectors. *IEEE Trans. on Industry Applications*, 24(1): 142–150, 1988. DOI:10.1109/28.87265
- [30] Celanovic, N., Boroyevich, D. 2000. A comprehensive study of neutral-point voltage balancing problem in three-level neutral-point-clamped voltage source PWM inverters. *IEEE Trans. on Power Electronics*, 15(2): 242–249, 2000. DOI:10.1109/apec.1999.749733
- [31] Yamanaka, K., Hava, AM., Kirino, H., Tanaka, Y., Koga, N., Kume, T. 2002. A novel neutral point potential stabilization technique using the information of output current polarities and voltage vector. *IEEE Trans. on Industry Applications*, 38(6): 1572–1580, 2002. DOI:10.1109/ias.2001.955552
- [32] Zadeh, LA. 1965. Fuzzy sets, in *Information and Control*. New York: Academic, 8: 338–353, 1965. DOI:10.1016/s0019-9958(65)90241-x
- [33] Elmas, Ç., Sağiroğlu, Ş., Çolak, İ., Bal, G. 1994. Nonlinear modelling of a switched reluctance drive based on neural networks. *IEEE Melecon 94, 7th Mediterranean Electrotechnical Conference*, 2: 809–812, 1994. DOI:10.1109/melcon.1994.380979
- [34] Elmas, Ç. 2011. *Artificial Intelligence Applications: Artificial Neural Networks, Fuzzy Logic and Genetic Algorithm*, Seçkin Press, Ankara, 1–424, 2011.
- [35] Elmas, Ç., Üstün, O., Sayan, HH. 2008. A neuro-fuzzy controller for speed control of a permanent magnet synchronous motor drive. *Expert Systems with Applications*, 34(1): 657–664, 2008. DOI:10.1016/j.eswa.2006.10.002
- [36] Elmas, Ç., Üstün, O. 2008. A hybrid controller for the speed control of a permanent magnet synchronous motor drive. *Control Engineering Practice*, 16(3): 260–270, 2008. DOI:10.1016/j.conengprac.2007.04.016