Proc. of the 5th WSEAS/IASME Int. Conf. on Electric Power Systems, High Voltages, Electric Machines, Tenerife, Spain, December 16-18, 2005 (pp80-85)

Control of Induction Motor Drive by Artificial Neural Network

L.FARAH[1], N.FARAH, M.BEDDA
Centre Universitaire Souk Ahras
BP 1553 Souk Ahras
ALGERIA

Abstract: Recently there has been increasing interest in the development of efficient control strategies to improve dynamic behaviour of power inverters. These systems mainly include power supply associated with inverters and electric motors. In this paper, a method for controlling induction motor drive is presented. It is based on the use of a well known artificial neural network, the multilayer perceptron (MLP) net. This neural net is utilized to generate clean and appropriate PWM controlling signals and to eliminate unwanted harmonics as well. The MLP net is trained to learn system variations; the backpropagation algorithm is applied as an update for adjusting the net weights. To show the effectiveness of our scheme, the proposed method was simulated on an electrical system composed of a synchronous motor and its power inverter. Simulation results concerning the speed control of such a system are also given

Key-words: Artificial Neural Networks, Control, P.W.M (Pulse Width Modulation), Backpropagation.

1 Introduction

Recently there has been increasing interest in the development of efficient control strategies to improve dynamic behaviour of power inverters. The behaviour of such systems is controlled by the switching ON and OFF of components such as thyristors or transistors. Among classical controllers which have been widely used there is the well-known P.W.M (Pulse Width Modulation) approach. This technique consists of controlling the process, using mean input values [1, 2, 3]. The regulation is often achieved by a P.I.D controller.

Present development trends in PWM inverters are primary concerned with the design of real time microprocessor-based PWM wave form generators. However, instead of the natural PWM described above, a modified PWM technique known as regular sampled, PWM is used [9].

Artificial Neural Networks have been proved extremely useful in pattern recognition [7, 8] and control systems [8, 9]. In this paper we propose an optimized multi-layer neural network for the generation of PWM waveforms, and then we show how it is able to control the state of a switching circuit and to provide the control output which ensures that the trajectory is followed in the state space.

This method utilizes the neural network paradigm as a mean to generate appropriate control signals to be applied on the system.

The proposed method has been simulated on a synchronous motor and its power inverter in order to show its effectiveness in speed control. Simulation results show a good response of the inverter circuit and confirm the validity of the neural approach.

2 Structure of the Artificial Neural Network

Artificial Neural Networks can be defined as highly connected arrays of neurons [8]. The internal structure of a neuron is shown in Fig 1.

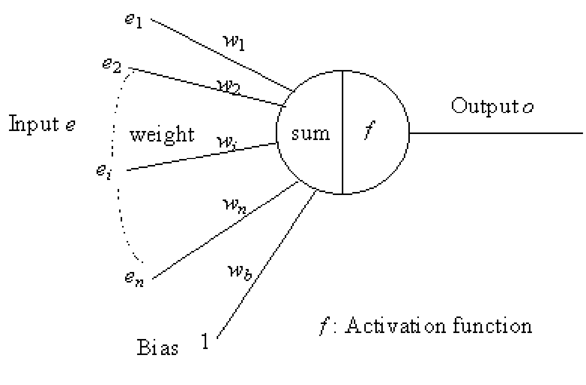


Fig. 1: Neurone Model.

The internal activity of a single neuron computes the weighted sum of the inputs e_i = (net)and passes this sum through a non-linear function, f according

Proc. of the 5th WSEAS/IASME Int. Conf. on Electric Power Systems, High Voltages, Electric Machines, Tenerife, Spain, December 16-18, 2005 (pp80-85)

to:

$$\begin{cases} net = \sum_{i}^{n} w_i e_i + w_b \\ o = f(net) \end{cases} \quad (1)$$

Another term called the bias term w_b is associated with this sum. The function used as a non-linear function is, for example, a sigmoid function given by:

$$f(net) = \frac{1}{1+e^{-net}} \quad (2)$$

A layer is a set of elementary neurons. The neural networks used here are basically layers of neurons connected in cascade, with one input layer, one or more hidden layers and one output layer. The input layer is the sensory organ for the Artificial Neural Networks. Each neuron in a layer is connected to neurons of adjacent following layer with different weights. Each neuron, except for the neurons of the input layer, receives signals from the neurons of the previous layer, weighted by the interconnect values between neurons. Consequently the output layer produces an output signal. The calculation of weights is performed with the well known learning algorithm, the backpropagation update rule which is presented in the next Section.

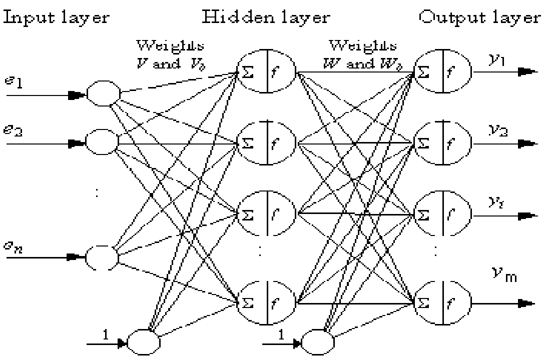


Fig 2: Scheme of an Artificial Neural Network.

The dimension of the input layer corresponds to the number of state variables to be manipulated. The output layer size is defined by the number of outputs. The choice of the number of hidden layer nodes is a compromise between efficiency and accuracy. The basic structure of a three-layer Artificial Neural Network capable to satisfactorily perform the control action is shown in Fig 2.

The propagation of the data is performed as follows. For the neuron of the output layer, the value y_i has the following shape:

$$y_i = f\left(\sum_{j=1}^{m}\left(w_{ij} f\left(\sum_{k=1}^{n} v_{jk} e_k + v_{bj}\right) + w_{bj}\right)\right) \quad (3)$$

Where:
e_k: kth input of the network.
v_{jk}: the interconnection weights between the input and hidden layers.
$v_{b,j}$: the jth bias weight of the hidden layer.
w_{ij}: the weights between the hidden and output layers.
$w_{b,i}$: the ith bias weight of the output layer.

Clearly, the matrix form of the net output is

$$y = F(W(F(Ve+V_b))+W_b) \quad (4)$$

With

$$F(NET) = [f(net_1) f(net_2) \Lambda\ f(net_m)]^T$$

Where net_z is the weighted sum of the z th neuron, and

$$NET = [net_1 net_2 \Lambda\ net_m]^T$$

$$y = \begin{bmatrix} y1 \\ y2 \\ \\ y_m \end{bmatrix},\ e = \begin{bmatrix} e1 \\ e2 \\ \\ e_n \end{bmatrix},\ W = \lfloor w_{ij} \rfloor,\ V = \lfloor v_{jk} \rfloor,$$

$W_b = \lfloor w_{bj} \rfloor$, and $V_b = \lfloor v_{bj} \rfloor$

2.1 Training Algorithm

The training problem consists of how to online adjust the weights using a set S of data. The learning strategy is based on the backpropagation algorithm [8]. The principle is to minimize for every input-output pair denoted
(e, y^d) of the set S, the quadratic criterion J defined as :

$$J = \frac{1}{2}\varepsilon_{nn}^T \varepsilon_{nn}$$

where the error ε_{nn} vector is given by the difference between the desired output y^d and the neural network output y obtained for the input e .

$$\varepsilon_{nn} = y^d - y$$

The algorithm used to minimize this criterion is based on the well-known gradient descent method, which gives for a weight w the following adaptation law:

$$\Delta w = -\eta \frac{\partial J}{\partial w}$$

With η the learning gain which influences the weights convergence speed.

Applying this algorithm to the network weights, we obtain the gradient vectors denoted δ_W and δ_V :

$$\begin{cases} \delta_W = \left(y^d - y\right) * F'\left(W\left(F\left(Ve + V_b\right)\right) + W_b\right) \\ \delta_V = \left(W^T \delta_W\right) * F'\left(Ve + V_b\right) \end{cases} \quad (5)$$

With

$$F'(NET) = \left[f'\left(net_1\right) \quad f'\left(net_2\right) \quad \cdots \quad f'\left(net_m\right)\right]^T$$

the derivative of F(NET)

Where $f'(net_z)$ *is* the derivative of f with respect to net_z :

$$f'\left(net_z\right) = \frac{\partial f\left(net_z\right)}{\partial net_z} = \frac{e^{-net_z}}{\left(1 + e^{-net_z}\right)^2} \quad (6)$$

$*$ Denote the HADAMARD product. Thus, the adaptation laws are:

$$\begin{cases} W_{new} = W_{old} + \eta\, \delta_W \left[F\left(Ve + V_b\right)\right]^T \\ W_{b,new} = W_{b,old} + \eta_b \delta_W \\ V_{new} = V_{old} + \eta\, \delta_V [e]^T \\ V_{b,new} = V_{b,old} + \eta_b \delta_V \end{cases} \quad (7)$$

The learning gain for the bias weights is η_b with $\eta_b < \eta$, so that their variations are not too large with respect to the weight variations of *W* and *V*.

3 Application to a Synchronous Motor

3.1 Presentation

The system used to obtain the simulation results is composed of a permanent magnet synchronous machine fed through a PWM inverter. In this approach of the problem of motor control, a mathematical model of the system is required to simulate its behaviour. By taking the statoric currents, the angle and the velocity as state variables the mathematical model of the system is given by:

$$\begin{aligned} \frac{d}{dt} i_d &= \frac{1}{L_d} v_d - \frac{R}{L_d} i_d + \frac{L_q}{L_d} p\, \omega_r i_q \\ \frac{d}{dt} i_q &= \frac{1}{L_q} v_q - \frac{R}{L_q} i_q - \frac{L_d}{L_q} p\, \omega_r i_d - \frac{\lambda p \omega_r}{L_q} \\ T_e &= 1.5 p\, [\lambda i_q + (L_d - L_q) i_d i_q] \end{aligned} \quad (8)$$

Where all quantities are expressed with respect to the rotor reference frame):

- L_q, L_d: q and d axis inductances
- R: resistance of the stator windings
- i_q, i_d: q and d axis currents
- v_q, v_d: q and d axis voltages
- ω_r: angular velocity of the rotor
- λ: amplitude of the flux induced by the permanent magnets of the rotor in the stator phases
- p: number of pole pairs
- T_e: electromagnetic torque Mechanical System

$$\begin{aligned} \frac{d}{dt}\omega_r &= \frac{1}{J}(T_e - F\omega_r - T_m) \\ \frac{d\theta}{dt} &= \omega_r \end{aligned} \quad (9)$$

Where:

- J: Combined inertia of rotor and load
- F: Combined viscous friction of rotor and load
- θ: Rotor angular position
- T_m: Shaft mechanical torque

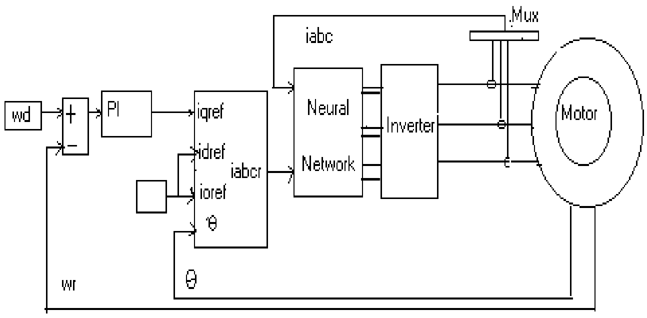


Fig. 3 Controller block diagram

The scheme proposed for speed control is shown in Fig. 3. The motor speed regulation is simply achieved with a feedforward action combined with a well-known proportional-integral (PI) controller. The ultimate gaol of the feedforward neural network is to improve the dynamic control performance. This compensator will then make it possible that satisfactory performances are reached for this application. The gains of the PI are chosen such that the response is fast and with a low overshooting for the nominal conditions.

For the application concerned, the PI controller gives satisfactory results with the gains chosen as follows:

$$K = 50 \text{ and } T_i = 2{,}6$$

The phase currents i_a, i_b, i_c and the reference currents i_{ar}, i_{br}, i_{cr}, are the inputs to the neural network. The voltages applied to the stator v_a, v_b, v_c, are the outputs of the network. The control process is now constituted by two regulation sub-systems, one for the speed control, and the other for tuning the currents.

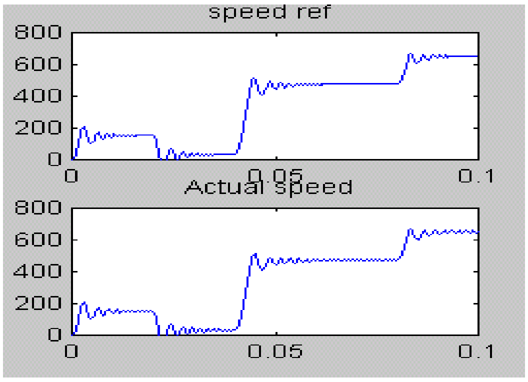
4 Simulation Results

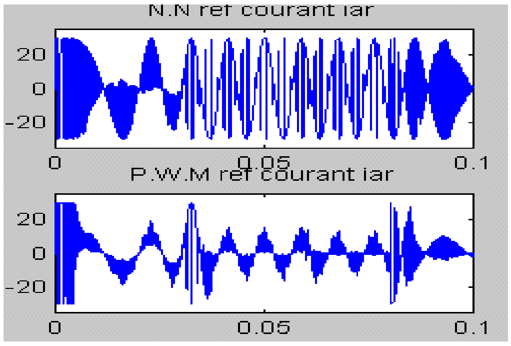
The MATLAB-SIMULINK simulation software has been used to study the response of the electrical system. The equations of the complete drive have been resolved by the fifth Runge-Kutta order method for the numerical integration. The three-phase inverter has been simulated by considering ideal switches and the synchronous machine has been represented by the state equation (14) with the following parameters:

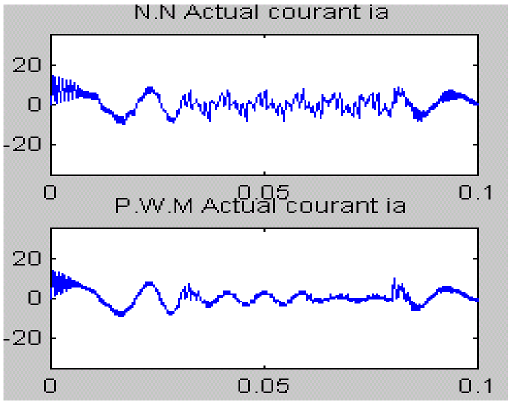
R= 2.875 Ω ϕ_m= 0.175 Wb   p= 4
Ld = 8.5 H   J= 0.8 kg.m^2
Lq = 8.5 H   F= 0 N.m.s

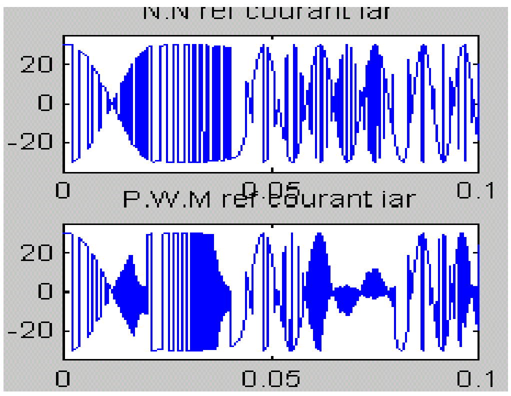
Where: ϕ_m is the flux induced by magnets

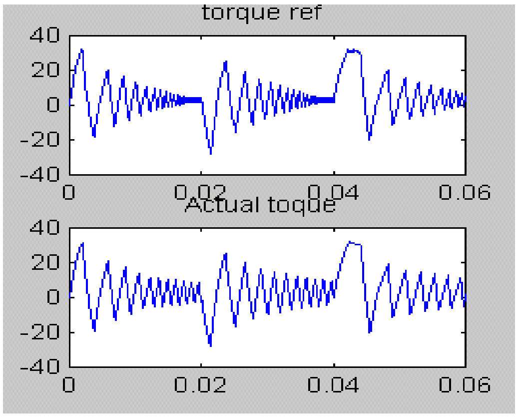
The structure of the neural network used is a 6-12-6 structure (six inputs, twelve neurons in the hidden layer, and six neurons in the output layer). The sampling time of the neural network is 0.2 ms, which corresponds to a 5 KHz switching frequency. This sampling period is chosen according to the dynamics of the system and the frequency limitation of the components

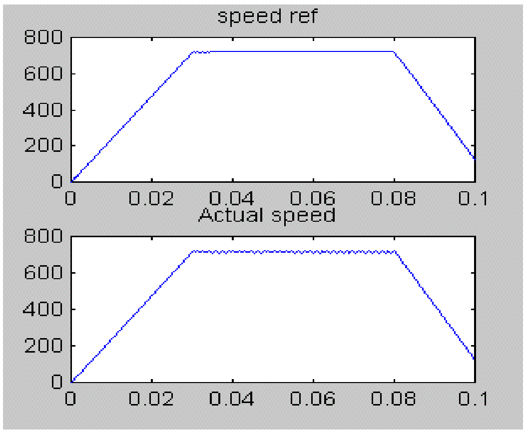


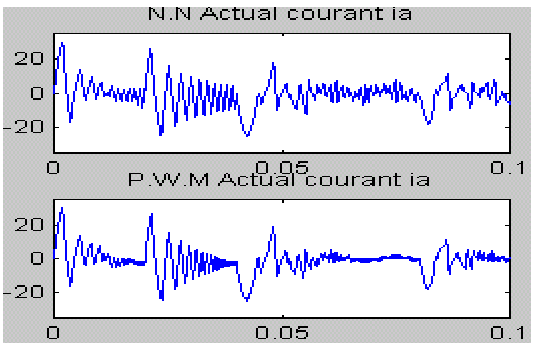












5 Conclusions

In this paper an artificial neural network is proposed for controlling nonlinear switching systems. The network learning is based on the backpropagation algorithm, which is a relatively low cost computing method, and so, it is easy to implement in real time. The obtained outcome is strongly dumped harmonics and then a constant speed.

The effectiveness of this approach is confirmed by simulation results obtained on a synchronous motor taken as an application system. The advantage of the neural network is to control the system without exact knowledge of its model. Online adaptation considerably improves the robustness of the system with respect to parametric changes. In conclusion, the proposed artificial neural network shows high performance and good control accuracy for switching systems.

References

[1] Rahman, M. F. and Zhong, L., **A Current-forced Reversible Rectifier Fed Single-Phase Variable Speed Induction Motor Drive**, Proceedings of PESC, Vol.1, 1996, pp. 114-119.

[2] Ismail. E.H. and Erickson, R., **Single-Switch 3 PWM Low Harmonic Rectifiers**, IEEE Powers Electronics, March 1996, pp. 338-346.

[3] Nonaka, S. and Neba, Y., **Analysis of A PWM GTO Current Source Inverter Fed Induction Motor Drive System**, IEEE Transactions on Industry Applications, Vol. 23, No 2, 1987.

[4] Sira-Ramirez, H., **Sliding Regimes in General Non Linear Systems : A Relative Degree Approach**, Int. J. Control, Vol. 50, No. 4, 1989, pp.1487-1506.

[5] Slotine, J. E., **Sliding Controller for Non-linear Systems**, Int. J. Control, Vol. 40, No. 2, 1984, pp. 421-434.

[6] Sabanovic, A. and Izosimov, D. B., **Application of Sliding Modes to Induction Motor Control**, IEEE Transactions on Industry Applications, Vol. 17, No. 1, 1981, pp. 41-49.

[7] Johnson, G. E., **Mimic Nets**, IEEE Transactions on Neural Networks, Vol. 4, No. 5, September 1993, pp. 803-815.

[8] Patterson, D. W., **Artificial Neural Networks**, Prentice Hall, Singapore, 1996.

[9] Narendra, K. S. and. Parthasarathy, K., **Identification and Control of Dynamical Systems Using Neural Networks**, IEEE Transactions on Neural Networks, Vol. 1, No. 1, 1990.

[10] ABADIE, V. and DAUPHIN-TANGUY, G., **Opened Loop control of switching linear system**, JOURNAL OF THE FRANKLIN INSTITUTE, Vol. 330, No. 5, 1993,pp. 799-813.

[11] Holderbaum, W., Dauphin-Tanguy, G . and Borne, P., **Boolean Control for Linear System**, Proceedings of International Symposium on Intelligent Automation and Control, ISIAC WAC'98 Anchorage, USA, 1998, pp. 212.1-212.6 .

[12] Holderbaum, W., Dauphin-Tanguy, G .and Borne, P., **Tracking Control Problem for Switching Linear System**, Proceedings of CESA'98 IMACS-IEEE/SMC Conference, Hammamet Tunisia, Vol. 1, 1998, pp. 935-940.

[13] Ducreux, J. P., Castelain, A., Dauphin-Tanguy, G. and Rombaut, C., **Power Electronics and Electrical Machines Modelling Using Bond-Graph** , in G. Dauphin-Tanguy and P. Breedveld (Eds.) IMACS Transactions on Bond-Graph for Engineers, Elsevier, NewYork, 1992.

[14] Leonhard, W., **Control of Electrical Drives**, Springer, Berlin , 1985.