Control of Induction Motor Drive by Artificial Neural Network

L.FARAH¹, 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.

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

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

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

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

Where:
- $e_k$: $k$th input of the network.
- $v_{jk}$: the interconnection weights between the input and hidden layers.
- $v_{bi}$: the $i$th bias weight of the hidden layer.
- $w_{ij}$: the weights between the hidden and output layers.
- $w_{bi}$: the $i$th bias weight of the output layer.

Clearly, the matrix form of the net output is

$$y = F(W(F(VE + V_b)) + W_b)$$ (4)

With

$$F(NET) = \left[ f(net_1) f(net_2) \ldots f(net_m) \right]^T$$

Where net$_i$ is the weighted sum of the $i$th neuron, and

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

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:

$$J = \frac{1}{2} e_m^T e_m$$

where the error $e_m$ vector is given by the difference between the desired output $y^d$ and the neural network output $y$ obtained for the input $e$.
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 \( \eta \) the learning gain which influences the weights convergence speed.

Applying this algorithm to the network weights, we obtain the gradient vectors denoted \( \delta_w \) and \( \delta_V \):

\[
\begin{align*}
\delta_w &= (y^d - y) \ast F'(W(V_{eq} + V_{es}) + W_i) \\
\delta_V &= (W^T \delta_w) \ast F'(V_{eq} + V_{es})
\end{align*}
\]

(5)

With

\[
F'(\text{NET}) = \left[ f'(\text{net}_1), f'(\text{net}_2), \ldots, f'(\text{net}_m) \right]^T
\]

the derivative of \( F(\text{NET}) \)

Where \( f'(\text{net}_j) \) is the derivative of \( f \) with respect to \( \text{net}_j \):

\[
f'(\text{net}_j) = \frac{\partial f(\text{net}_j)}{\partial \text{net}_j} - \frac{e^{-w_{net_j}}}{(1 + e^{-w_{net_j}})^2}
\]

(6)

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

\[
\begin{align*}
W_{new} &= W_{old} + \eta \delta_w [F'(V_{eq} + V_{es})] \\
W_{old} &= W_{old} + \eta \delta_w \\
V_{new} &= V_{old} + \eta \delta_V \\
V_{old} &= V_{old} + \eta \delta_V 
\end{align*}
\]

(7)

The learning gain for the bias weights is \( \eta_b \) 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:

\[
\begin{align*}
\frac{d}{dt} i_d &= \frac{1}{L_d} v_d - \frac{R}{L_d} i_d + \frac{L_q}{L_d} p \omega_j i_q \\
\frac{d}{dt} i_q &= \frac{1}{L_q} v_q - \frac{R}{L_q} i_q - \frac{L_d}{L_q} p \omega_j i_d - \frac{\lambda p \omega_j}{L_q} \\
T_e &= 1.5p [\lambda j_q + (L_d - L_q) i_d i_q]
\end{align*}
\]

(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
- \( v_q, v_d \): q and d axis voltages
- \( \omega_r \): angular velocity of the rotor
- \( \lambda \): 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

\[
\begin{align*}
\frac{d}{dt} \omega_r &= \frac{1}{J}(T_e - F \omega_r - T_m) \\
\frac{d \theta}{dt} &= \omega_r
\end{align*}
\]

(9)

Where:

- \( J \): Combined inertia of rotor and load
- \( F \): Combined viscous friction of rotor and load
- \( \theta \): Rotor angular position
- \( T_m \): Shaft mechanical torque

The scheme proposed for speed control is shown in Fig. 3. The motor speed regulation is simply

![Fig. 3 Controller block diagram](image)
achieved with a feedforward action combined with a
well-known proportional-integral (PI) controller.
The ultimate goal 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:

\[
R = 2.875 \Omega \quad \phi_m = 0.175 \text{ Wb} \quad p = 4 \\
L_d = 8.5 \text{ H} \quad J = 0.8 \text{ kg.m}^2 \\
L_q = 8.5 \text{ H} \quad F = 0 \text{ N.m.s}
\]

Where: \( \phi_m \) 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 damped 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


