

Article

Sensorless FOC Performance Improved with On-Line Speed and Rotor Resistance Estimator Based on an Artificial Neural Network for an Induction Motor Drive

Jose M. Gutierrez-Villalobos *, Juvenal Rodriguez-Resendiz, Edgar A. Rivas-Araiza and Moisés A. Martínez-Hernández

Laboratorio de Mecatrónica, Universidad Autónoma de Querétaro, Cerro de las Campanas, Col. Las Campanas, S/N, Queretaro 76010, Mexico; E-Mails: juvenal@ieee.org (J.R.-R.); erivas@uaq.mx (E.A.R.-A.); moises.martinez@uaq.mx (M.A.M.-H.)

* Author to whom correspondence should be addressed; E-Mail: marcelino.gutierrez@uaq.mx; Tel.: +52-442-192-1200 (ext. 7005).

Academic Editor: Vittorio M.N. Passaro

Received: 11 April 2015 / Accepted: 23 June 2015 / Published: 29 June 2015

Abstract: Three-phase induction motor drive requires high accuracy in high performance processes in industrial applications. Field oriented control, which is one of the most employed control schemes for induction motors, bases its function on the electrical parameter estimation coming from the motor. These parameters make an electrical machine driver work improperly, since these electrical parameter values change at low speeds, temperature changes, and especially with load and duty changes. The focus of this paper is the real-time and on-line electrical parameters with a CMAC-ADALINE block added in the standard FOC scheme to improve the IM driver performance and endure the driver and the induction motor lifetime. Two kinds of neural network structures are used; one to estimate rotor speed and the other one to estimate rotor resistance of an induction motor.

Keywords: adaptive system; neural networks; on-line identification; adjustable speed driver; parameter estimation; FOC

1. Introduction

Three-phase induction motors (IMs) have become the most employed electrical device in industry, thanks to their robustness, efficiency and low maintain. Nowadays, IMs consume 60% of electricity that industry requires, which represents the 21% of global electrical production, about 3.99 PetaWatts-hour [1]. Therefore, in order to increase the IM efficiency and raise its performance, any improvement results in a global benefit and helps the industrial economy. Current drivers are based on control schemes such as voltage-frequency (V/F), direct torque control (DTC) and field oriented control (FOC). FOC has been the most selected control method for IMs in torque and speed control applications where high performance is required. There have been many contributions towards improve it for almost 3 decades [2]. Basically, FOC sees the IM as a direct current (DC) motor and it has been classified in two types. One method has been the direct field oriented control (DFOC), which needs to know the position of the field magnetic flux and the rotor magnetic flux with hall-effect sensors inside the electric machine, which results complicated or impossible to installation of the sensors. The other scheme is indirect field oriented control (IFCO), which has two different diagrams. The sensed diagram estimates magnetic flux position and uses an encoder to measure the IM rotor speed. The sensorless scheme estimates magnetic flux as well but it does not need an encoder to know the rotor speed. Sensorless IFCO estimates the motor velocity based on the number of IM poles, and the frequency of voltages and currents applied to stator coils [3]. Currently, commercial drivers use sensorless IFCO because it does not need an encoder. It is to say, it is a simple controller with good performance and acceptable results for some applications.

As explained before, IFCO is better known only as FOC in the IM driver and controller literature. It is the control technique that has gained more interest within researchers because there have been opportunity areas for control theory, observers and estimators. Figure 1 shows the general sensorless FCO scheme. For the reason of not using sensors and estimating field magnetic flux, there have been many surveys in this area. Other, previously published reviews such as [4] analyze and describe several methods, where observers, estimators and controllers were used. Never the less, there are some works that only presented validations for short times such as [5–7], which do not really ensure that their systems work properly, after longer terms or under motor condition variations.

This work employs a combination of two different artificial neural network (ANNs) approaches. The combination of tow neural networks (NNs) helps to improve IM-driver performance, by means of reducing the mismatch produced between estimated electrical parameters values and the real ones. The ANN unit estimates motor speed (ω_r) and rotor resistance (R_r), which allows us to keep the IM controller tuned, achieving higher control over a wide speed range and load variations. Since R_r varies by up to 100% from its original value due to temperature changes and load steps [8]. Normal control schemes can present undesirable speed tracking at low speeds. It leads to decreases in IM efficiency and shortens motor drive life.

In this algorithm, a cerebellar model articulation controller (CMAC) was used to estimate ω_r , it is selected for its capability of quickly learning non-linear functions due to the local nature of its weight modification. These networks are simple to implement when compared to other types of neural network as describe in [9]. Then, for R_r estimation, a block with an adaptive linear neuron (ADALINE) structure is selected. The main advantages of this ANN are its simplicity and the ability to be trained online;

besides, the ADALINE weights can be interpreted physically, as explained in [10]. Figure 2 presents the proposed ANN scheme.

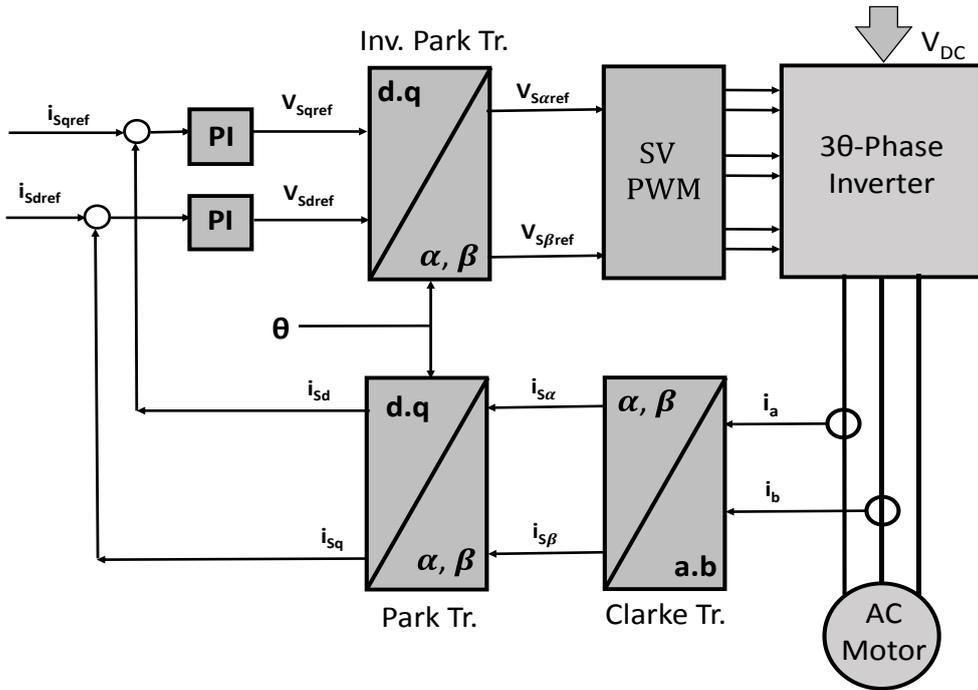


Figure 1. General sensorless indirect field oriented control (IFCO) scheme.

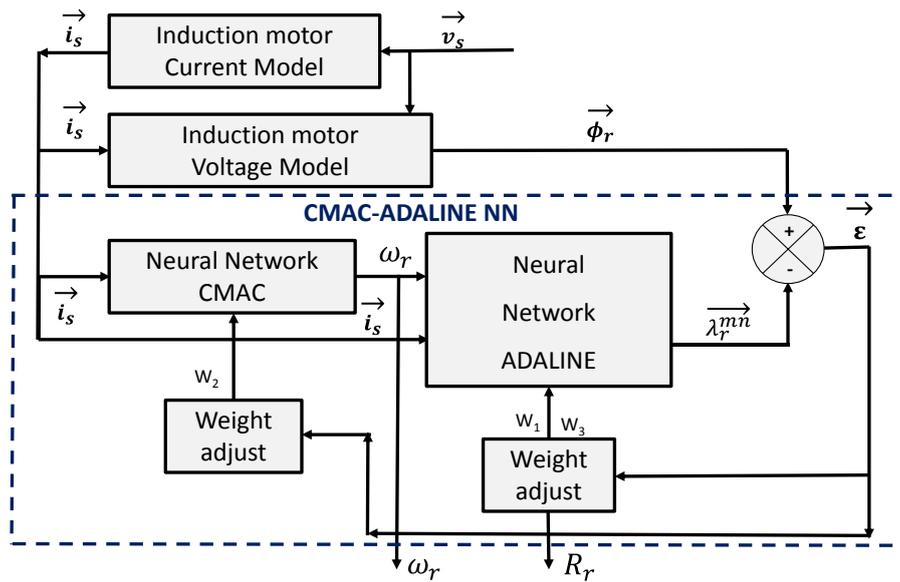


Figure 2. Cerebellar model articulation controller (CMAC)-adaptive linear neuron (ADALINE) structure for the parameter estimator.

This survey is divided into four sections. The first part, which describes how this algorithm was implemented and its outcomes, the second part presents the ANN structures employed in the research and their behavior, the third part explains how CMAC-ADALINE unit performs on-line parameter estimation and improves drive speed control performance, the fourth part shows conclusive results and the final section contains the conclusions and the possible future applications for this work.

1.1. CMAC Description and Behavior

The CMAC structure was developed (such as, the human cerebellum), and it is a convenient learning structure for on-line and real-time modeling and controlling induction motors, IMs are non-linear systems. CMAC is capable of learning non-linear functions; it can be defined as a look-up-table. In this structure, each state variable is quantized and mapped, see Figure 3. The space vector is divided into discrete states. The output of each state can be obtained from the sum of the information stored in the associate memories.

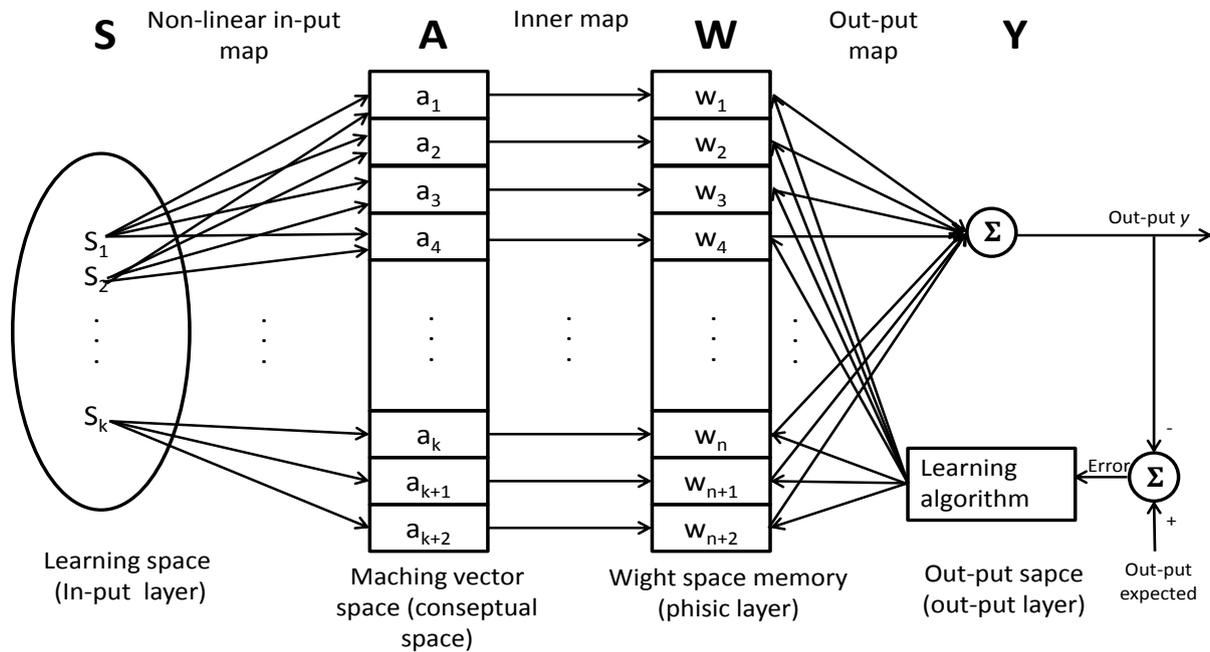


Figure 3. Artificial neural network (ANN) CMAC model.

The state vector is mapped in S^{th} elements onto another vector called A , which is formed by n^{th} elements. In the next map, the output P is the sum of the weights whose activation function will be activated. Its representative mathematical equations is given as follows:

$$f : s \rightarrow A$$

$$g : A \xrightarrow{w} P$$

Now s is an input vector of k -dimension, A is the N -dimension association vector containing M active elements, P is the output vector and W is the corresponding weight vector in which input data of P is stored, as explained in [11].

$$P_k = g(a_k^*) = A_k W_k \tag{1}$$

In this equation, k is associated to k^{th} states, a_k^* means the activation function of the k^{th} state, and A_k is the grouping of cell vector of k^{th} state. The form CMAC weights are adapted is given by:

$$W_k(j+1) = W_k(j) + \Delta W_k(j) = W_k(j) + \frac{\alpha}{M} (Y_{d,k} - P_k(j)) \tag{2}$$

$$W_k(j+1) = W_k(j) + \frac{\alpha}{M} (Y_{d,k} - A_k W_k(j)) \tag{3}$$

where W_k means the weight vector at k -th state, j means j -th iteration, α is the learning rate is selected, M is the number of activated association cells, and $Y_{d,k}$ is the desired output at k -th state. The learning rule of CMAC is to distribute the error equally to the corresponding weights, as well explained in [11]. In the learning process, the input of the controlled plant U (induction motor) is treated as a desired output to modify the contents of CMAC stored at location $Y(k)$ and $Y(k + 1)$, where $Y(k + 1)$ is the actual system output at time step $k + 1$. How precisely the CMAC can approximate a controlled plant is mainly determined by the quantization in each dimension of the input vector and the generalization width.

1.2. ADALINE Description and Behavior

In this section, The ADALINE structure is explained in [10]. The ANN-ADALINE structure is equivalent to one neuron which is composed of an input vector X_k , a weight matrix W_k , and an activation function $f(v)$. The weights vector $W_k = [W_{0k}, W_{1k} \dots W_{nk}]^T$ corresponds to the whole neuron synaptic forces. The input vector $X_k = [X_{0k}, X_{1k} \dots X_{nk}]$ corresponds to the whole neuron input stimulus. The activation function $f(v)$ specifies the neuron behavior. Various activation functions can be used in the ANN theory. However, the ADALINE uses the linear activation function $f(v) = v$ Consequently, the ADALINE output y_k is given by: $y_k = X_k W_k$. When the ADALINE is excited, it produces the output y_k which depends on the input vector. The basic scheme of an ADALINE structure is shown in Figure 4. The weights vector W_k is continuously modified during the network learning process with the purpose of approaching as close as possible the desired output d_k .

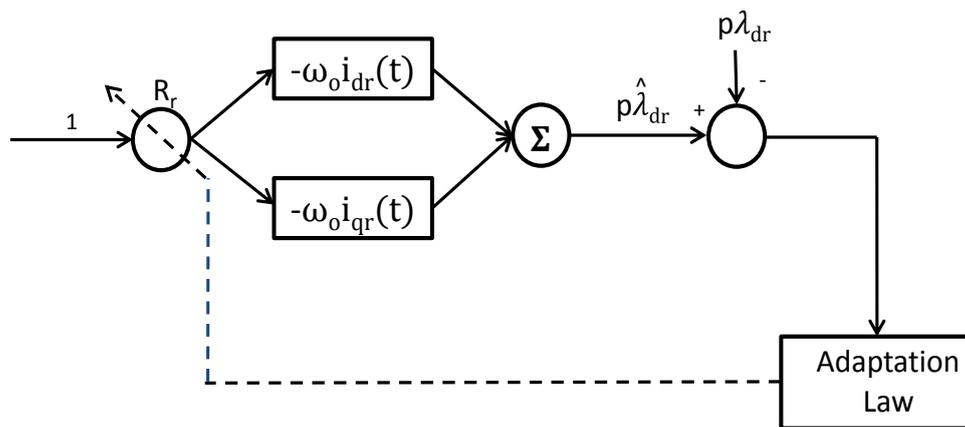


Figure 4. ANN ADALINE model.

It is a linear combination of its in-put vector $X(k) = [X_1, X_2 \dots X_n]^T$ at any time k^{th} that is multiplied by the weight vector $W(k) = [W_1, W_2 \dots W_n]$, finally, their products are summed to obtain a linear output represented by:

$$\hat{Y}(k) = X(k)^T W(k) \tag{4}$$

And adaptation rule is based on the Least-Square-Method (LMS), which is known as the Widro-Hoff delta rule and is given by:

$$\hat{Y}(k) = W(k) + \alpha \frac{X(k)e(k)}{X(k)^T X(k)} \tag{5}$$

The main drawback with the ADALINE structure consists in selecting the correct set of weights so that the input-output ADALINE behavior gets closer to the desired input-output data. In order to update ADALINE weights, the LMS algorithm was used to minimize the error $e(k)$. The learning process for ADALINE is well explained in [12]. This structure requires supervised learning. The error is obtained by using the energy function $E = (1/2)e^2(k)$ and is calculated by:

$$e(k) = d_k - y_k = d_k - X_k^T W_k \quad (6)$$

The stepping method is described by the expression:

$$W_{k+1} = W_k + \eta(-\bar{\nabla}_k) = W_k - \eta \frac{\partial e_k^2}{\partial W_k} \quad (7)$$

where η is used to control the stability and the convergence rate; $\bar{\nabla}_k$ is the gradient values at $W = W_k$.

Differentiating the equation above and introducing the linear error we obtain:

$$W_{k+1} = W_k - \eta \frac{\partial e_k^2}{\partial W_k} = W_k - 2\eta e_k \frac{\partial e_k}{\partial W_k} \quad (8)$$

and

$$W_{k+1} = W_k - 2e_k \eta \frac{\partial (d_k - X_k^T W_k)}{\partial W_k} \quad (9)$$

which results in

$$W_{k+1} = W_k - 2\eta e_k X_k^T \quad (10)$$

Finally, the constant η determines stability and convergence rate, this value is typically less than one. The training coefficient η is used to accelerate the algorithm convergence and it has to be variable. It has to be high at the beginning of the training process; then, it has to decrease to present a small value at the end. The value of η can be selected by the expression Equation (11) as well explain in [13].

$$\eta = \eta_i \left(\frac{\eta_f}{\eta_i} \right)^{\frac{t}{t_{max}}} \quad (11)$$

where η_i is the initial value of the learning rate and η_f is the final value, and t_{max} is the maximum training time. Never the less, this parameter has to be first obtained by simulation, then the ANN response has to be analyzed to determine the correct value for the physical system.

2. Speed and Rotor Resistance Estimator Implementation

Thus, it is important to consider the state-variable model of the induction machines referred to an arbitrarily rotating reference frame [11]. Which is written as

$$\frac{d}{dt} \begin{bmatrix} i_{S_d}^c \\ i_{S_q}^c \\ \psi_{R_d}^c \\ \psi_{R_q}^c \end{bmatrix} = \begin{bmatrix} -\frac{R_S}{L_S\sigma} - \frac{R_S(1-\sigma)}{L_S\sigma} & \frac{\omega_S}{L_S\sigma} - \frac{R_S(1-\sigma)}{L_S\sigma} & \frac{MR_S}{\sigma L_R L_R^2} & \frac{M\omega_R}{\sigma L_R L_R} \\ & -\omega_S & \frac{M\omega_R}{\sigma L_R L_R} & \frac{MR_S}{\sigma L_R L_R^2} \\ \frac{MR_S}{L_R} & 0 & \frac{R_R}{L_R} & \omega_S - \omega_R \\ 0 & \frac{MR_S}{L_R} & -\frac{R_r}{L_R} & -\frac{R_r}{L_R} \end{bmatrix} \begin{bmatrix} i_{S_d}^c \\ i_{S_q}^c \\ \psi_{R_d}^c \\ \psi_{R_q}^c \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ \frac{1}{L_S\sigma} & \frac{1}{L_S\sigma} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{S_d}^c \\ v_{S_q}^c \end{bmatrix} \tag{12}$$

Besides, this mathematical model is described in [14]. For being able to implement this combined ANNs, it is necessary to consider the IM current model given by Equation (13) and the IM voltage model given by Equation (14). The Equation (14) is required to adjust the weights, and these equations are shown in the following lines.

$$\begin{bmatrix} \frac{d\lambda_{dr}^{vm}}{dt} \\ \frac{d\lambda_{qr}^{vm}}{dt} \end{bmatrix} = \frac{L_r}{L_m} \left\{ \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix} - R_s \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} - \sigma L_s \begin{bmatrix} \frac{di_{ds}}{dt} \\ \frac{di_{qs}}{dt} \end{bmatrix} \right\} \tag{13}$$

$$\begin{bmatrix} \frac{d\lambda_{dr}^{im}}{dt} \\ \frac{d\lambda_{qr}^{im}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_r} & -\omega_r \\ \omega_r & -\frac{1}{T_r} \end{bmatrix} \begin{bmatrix} \lambda_{dr}^{im} \\ \lambda_{qr}^{im} \end{bmatrix} + \frac{L_m}{T_r} \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} \tag{14}$$

The current model of Equation (14) can also be rewritten as:

$$\overrightarrow{\lambda}_r^{im} = \left(\frac{-1}{T_r} I + \omega_r J \right) \overrightarrow{\lambda}_r^{im} + \frac{L_m}{T_r} \overrightarrow{i}_s \tag{15}$$

where

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \overrightarrow{i}_s = \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix}$$

$$\overrightarrow{v}_s = \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix}, \quad \overrightarrow{\lambda}_r^{im} = \begin{bmatrix} \lambda_{dr}^{im} \\ \lambda_{qr}^{im} \end{bmatrix}, \quad \overrightarrow{\lambda}_r^{vm} = \begin{bmatrix} \lambda_{dr}^{vm} \\ \lambda_{qr}^{vm} \end{bmatrix}$$

The discrete data model of Equation (15) can also be expressed as:

$$\overrightarrow{\lambda}_r^{im}(k) = (W_1 I + W_2 J) \overrightarrow{\lambda}_r^{im}(k-1) + W_3 \overrightarrow{i}_s(k-1) \tag{16}$$

Considering

$$W_1 = 1 - \frac{T_s}{T_r}; \quad W_2 = \omega_r T_s; \quad W_3 = \frac{L_m}{T_r} T_s$$

and T_s is the sampling period. Equation (16) can also be written as:

$$\overrightarrow{\lambda_r^{nm}}(k) = W_1 X_1 + W_2 X_2 + W_3 X_3 \quad (17)$$

Therefore

$$X_1 = I \overrightarrow{\lambda_r^{nm}}(k-1) = \begin{bmatrix} \lambda_{dr}^{nm}(k-1) \\ \lambda_{qr}^{nm}(k-1) \end{bmatrix}$$

$$X_2 = J \overrightarrow{\lambda_r^{nm}}(k-1) = \begin{bmatrix} -\lambda_{dr}^{nm}(k-1) \\ \lambda_{qr}^{nm}(k-1) \end{bmatrix}$$

$$X_3 = I \overrightarrow{i_s}(k-1) = \begin{bmatrix} i_{ds}(k-1) \\ i_{qs}(k-1) \end{bmatrix}$$

Then the weight adjustment is obtained by trying to minimize the error

$$\bar{\varepsilon}(k) = \overrightarrow{\phi_r}(k) - \overrightarrow{\lambda_r^{nm}}(k) \quad (18)$$

The weights between neurons are tuned by minimizing the energy function

$$E = \frac{1}{2} \bar{\varepsilon}^2(k) = \frac{1}{2} \left\{ \overrightarrow{\phi_r}(k) - \overrightarrow{\lambda_r^{nm}}(k) \right\}^2 \quad (19)$$

The weight variation is given by

$$\Delta W_n(k) \propto -\frac{\partial E}{\partial W_n} = -\frac{\partial E}{\partial \overrightarrow{\lambda_r^{nm}}(k)} \frac{\partial \overrightarrow{\lambda_r^{nm}}(k)}{\partial W_n} = -\bar{\delta} X_n \quad (20)$$

where

$$\bar{\delta} = \frac{\partial E}{\partial \overrightarrow{\lambda_r^{nm}}(k)} = \left[\overrightarrow{\lambda_r^{im}}(k) - \overrightarrow{\lambda_r^{nm}}(k) \right]^T \quad (21)$$

Another coefficient (α) is added to determine the effect of the changes of the past weights and increase in the learning rate. Finally, weight adjustment of W_1 and W_3 can be represented by the next form [15]:

$$W_1(k) = W_1(k-1) - \eta \bar{\delta} X_1 + \alpha \Delta W_1(k-1) \quad (22)$$

$$W_3(k) = W_3(k-1) - \eta \bar{\delta} X_3 + \alpha \Delta W_3(k-1) \quad (23)$$

The Equation (11) was used to select a proper value for η , several values had to be tested in order to determine what value is the most convenient.

and considering W_2

$$\Delta W_2(k) = \left(\overrightarrow{\phi_r}(k) - \overrightarrow{\lambda_r^{nm}}(k) \right)^T J \overrightarrow{\lambda_r^{nm}}(k-1) \quad (24)$$

Then the weight W_2 value is calculated as follows:

$$W_2(k) = W_2(k-1) + \frac{\alpha}{M} \Delta W_2(k) \tag{25}$$

By using the Equation (26) and solving the speed from the weight W_2 , the estimated rotor speed is given by combining equations above, the estimated rotor speed is represented as follows:

$$\omega_r(k) = \omega_r(k-1) + \frac{\alpha}{M} \frac{\Delta W_2(k)}{T_s} \tag{26}$$

and rotor resistance can be found from either W_1 or W_3 using Equations (27) or (28).

$$R_r = \frac{L_r W_3}{L_m T_s} \tag{27}$$

$$R_r = \frac{L_r (1 - W_1)}{T_s} \tag{28}$$

Both equations above were tested, but finally Equation (28) was eventually selected to be used because it presented less variation and better response due to it was only affected by L_r . It is important to know how the proposed algorithm is exactly incorporated to the general FCO scheme as shown in Figure 5.

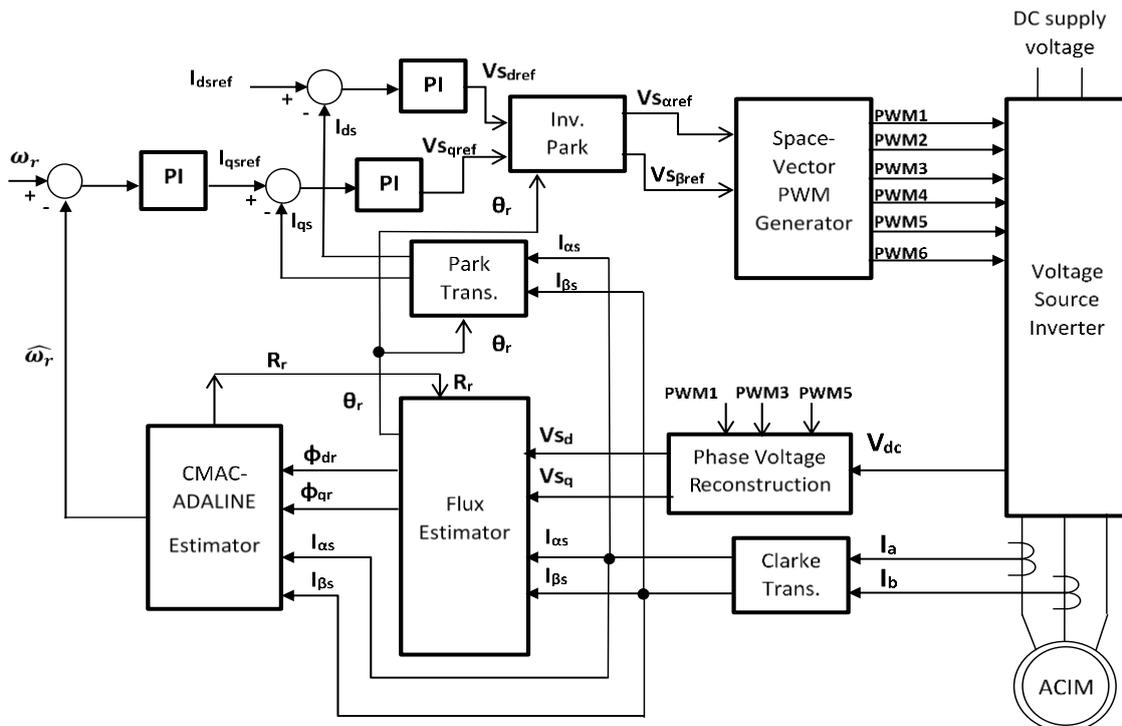


Figure 5. CMAC-ADALINE block connected to the FCO scheme.

3. Experimental Results

This part presents the results obtained from a comparative between standard FOC scheme and the FOC one including the proposed estimating algorithm. The initial values were $\eta = 0.05$ and $\alpha = 0.01$,

$T_s = 0.001$ s, this block was implemented and validated in a real-time embedded system TI tool set TMDSHVMTRPFCKIT. Table 1 shows the parameter of the IM.

Table 1. Parameters of the induction motor.

1/4 HP, 1725 R.P.M., 208–230 V, 1.2–1.3 A, 3 Phase, 4 Pole, 60 Hz
Stator Resistance R_s : 11.05 Ω
Stator Inductance L_s : 0.316423 H
Rotor Resistance R_r : 6.11 Ω
Rotor Inductance L_r : 0.316423 H
Mutual Inductance L_m : 0.293939 H

The IM is shaft-to-shaft connected to a DC motor, which simulates a changing load. That load can be set or adjusted to validate control performance at different speeds and loads. Estimated ω_r and R_r are real-time calculated and they get adjusted at every millisecond in the Code Composer control panel included in the TI software. This is suitable to track IM speed, store and validate real and estimated parameter values. The DC motor is controlled with a servo amplifier that has an inner current control loop to drive torque applied to the IM. DC motor features are described in Table 2.

Table 2. Direct current motor parameters.

Power: 1/4 Hp
R.P.M.: 1725
Voltage (dc): 180 V
Current: 2.5 A
Torque constant (Kt): 1 N m/A
Torque Max: 2.5 N

The servo amplifier is connected to a National Instruments (NI) data acquisition board (DAQ) by means of an USB connection, then a human-machine interface (HMI) in LabVIEW is implemented to monitor and control variable torque applied to the AC motor. A load from 0 to 2.5 N m can be emulated by this system. A FUTEK shaft-to-shaft rotary torque sensor (TRS300) is used to measure the mechanical torque applied to the AC motor, and its signal is connected to an NI DAQ to create the load control system as shown in Figure 6. There were three speeds applied: low speed (LS) at 10% of nominal RPM value, medium speed (MS) at 50% and high speed (HS) at 90%.

Sensorless FOC without estimator algorithm was first tested with a specific velocity profile, which consists of speed steps applied at 180, 540, 860, 1260 and 1440 R.P.M. at 1 N m torque, generated with the DC motor. This created disturbances that directly affected IM speed. Those effects decreased or increased when load was applied or removed. Figure 7a shows the error between the estimated velocity and the real one at low revolutions, however, as speed increases, estimated velocity gets closer to the real value. Figure 7b shows the estimated speed and the desire one with load applied, which consists of speed steps applied at 180 and 540 R.P.M. at 1 N m torque, using the ANN block, introduced to the standard FOC. The real velocity was measured with a QD200 encoder connected directly to the IM shaft.

Thus, R_r tends to vary within a range close to the manufacturer value, approximately 6.11 Ohms. These changes precisely occur at speed variations. These mismatches affect the controller performance

because, as it was described in Section 3, R_r is part of the IM mathematical model that controller uses. Figure 8a shows the speed steps applied to the IM and Figure 8b presents the R_r is estimated with the other ADALINE structure. These changeable values are the ones that are fed back to the controller scheme in order to keep it tuned.

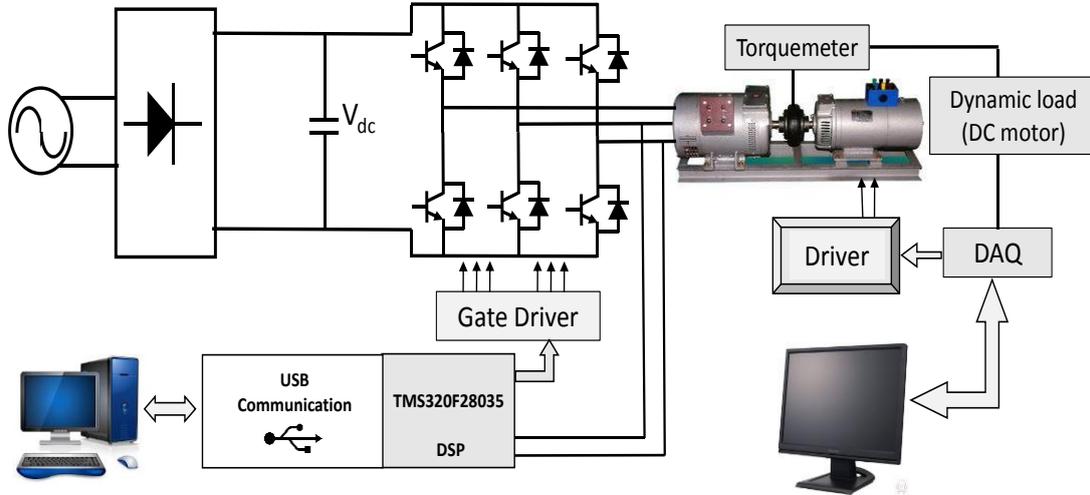
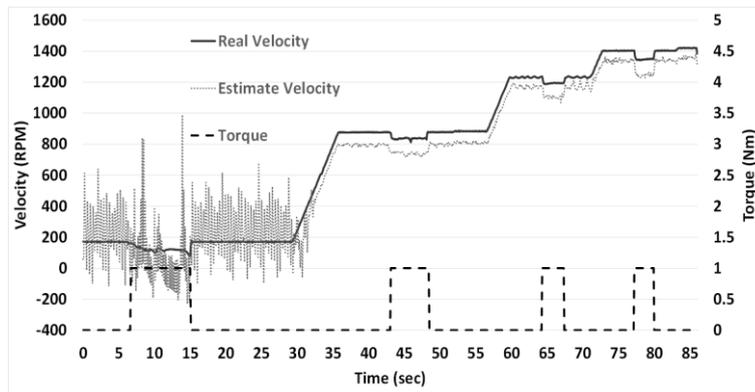
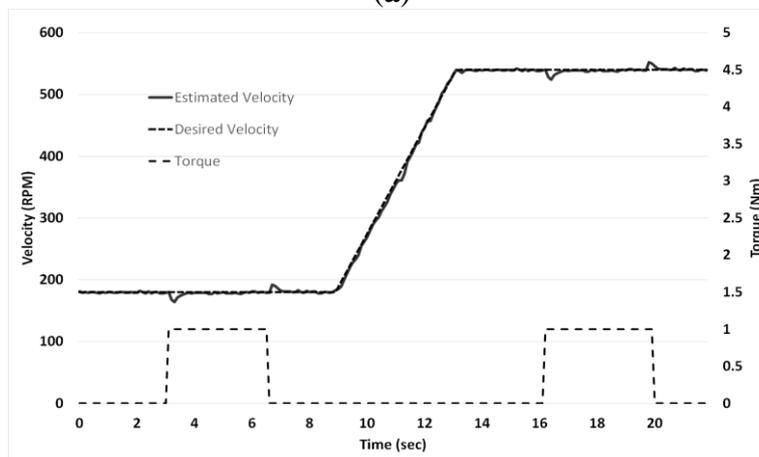


Figure 6. Complete system scheme used to test and validate the algorithm and testbench to simulate different load resistances.



(a)



(b)

Figure 7. (a) FOC without ANN estimator response and (b) FOC with the ANN-block.

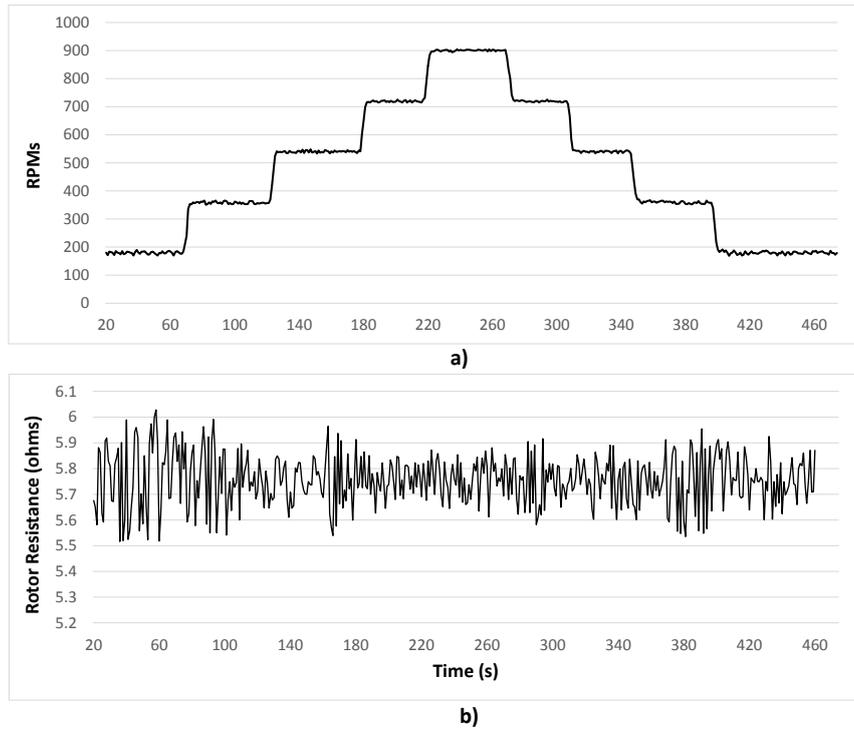


Figure 8. (a) Speed changing profile from 180 to 900 R.P.M with 1 N m. load and (b) Comparison between manufacturer motor resistance and estimated one.

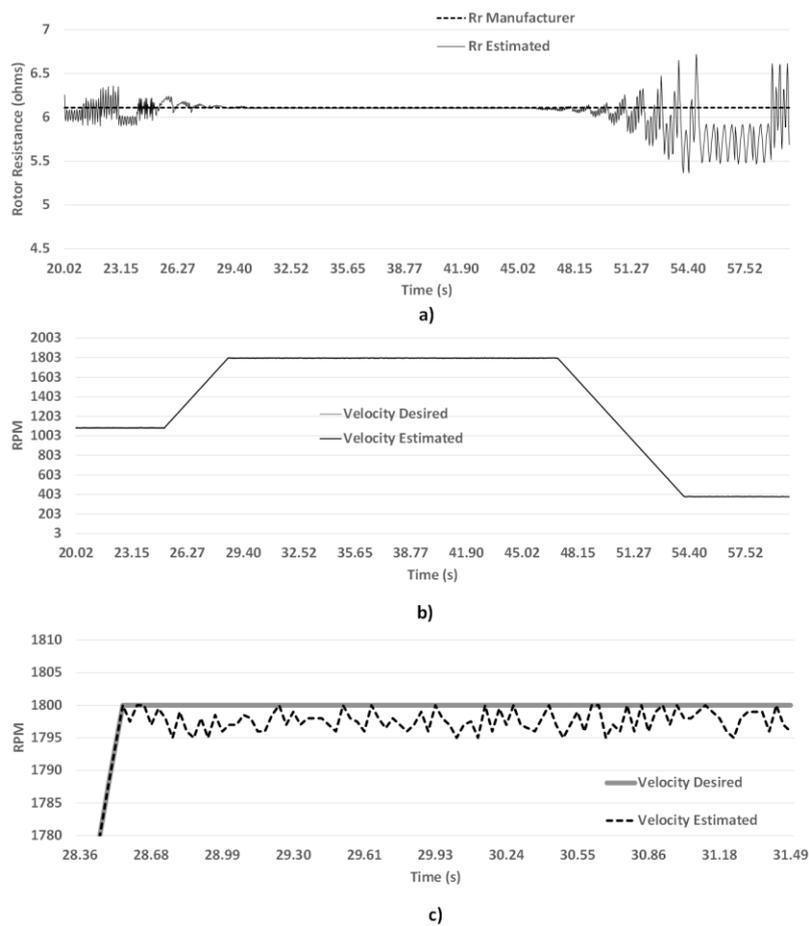


Figure 9. (a) Estimated R_r vs. Desired R_r ; (b) Estimated ω_r vs. Desired ω_r ; (c) Zoom to the velocity.

As it can be observed in Figure 9, ω_r and R_r value vary at speed changes because the currents and the magnetic fluxes are changed depending on IM electric demand, affecting the controller function. Never the less, the proposed algorithm updates these parameters and introduces them to the FOC scheme and improves driver performance. But, at constant speeds, resistance and speed values keep almost at the same value. Figure 9a shows the R_r and Figure 9b presents speed estimation respectively, finally, Figure 9c presents a zoom applied to the speed estimation.

4. Conclusions

The main contribution of this work is proposing a new kind of estimator for control schemes for neural network and control theory. Along, in this article, the IM speed and rotor resistance estimation was validated and the IM driver performance was improved, due to the inclusion of the CMAC-ADALINE block into the standard FOC scheme. A new hybrid neural network estimator was designed based on two types of neural network structures and resulted as expected.

This estimator adjusts its set of weights in order to update these two values (speed and rotor resistance), which improves FOC algorithm behavior. The implementation of this algorithm was easy to program, on a real-time application over a three-phase IM, and good speed and resistance tracking with a minimum error was achieved. As it can be observed, that results are very satisfactory considering the IM was connected to a dynamic load system, which was modified during the test.

The employed kit was only for test and validation of the FOC scheme added with the ANN estimator and it is not for industrial usage yet, a higher power capacity platform must be built to meet the requirements based on the power demand of the application. Future projects establish the need to develop a more power capacity station and improve of this control scheme to meet industry standards, to command heavier loads and to have the correct accuracy.

Acknowledgments

The author would like to thank Comite Nacional de Ciencia y Tecnologia (CONACyT), the Graduate Department of the Universidad Autonoma de Queretaro and the Engineering faculty of the UAQ.

Author Contributions

Jose M. Gutierrez-Villalobos: This work starts with the idea that I had, coming from my doctoral thesis, which basically consisted in developing an adaptive system for a field oriented control for an induction motor, which consists of a standard vector control improved with this artificial neural network. Since, industry uses a lot of induction motors we thought this work would have some impact in adjustable speed driver construction. Beside, other consideration such as power factor correction and total harmonic distortion reduction that are part of this same work, as well. And I developed part of the mathematical analysis.

Juvenal Rodriguez-Resendiz: The experience I have with induction motors was important to this research, because of the challenge that the construction of the test bench represented in order to connect both (DC and AC) motors. My knowledge about induction motor advantages and drawbacks was helpful to determine the suitable conditions to accomplish the testing task.

Edgar A. Rivas-Araiza: My contribution to this survey was the part to implement the control for the DC motor and the human-device interface, which basically consisted in connecting the DC motor and the national instruments products to implement the control system and drive the DC motor. This allowed us to test the induction motor at different loads and speeds. The experience I have with national instruments has led me to develop quickly motor control systems validate their performances.

Moises A. Martinez-Hernández: I have worked with artificial neural networks since I studied my master degree that is the reason I was invited to collaborate to this research, the experience I have with their training process and selecting parameters. I helped with the implementation of this algorithm in the TI platform to proof this new neural network programmed in C.

Conflicts of Interest

The authors declare no conflict of interest.

Nomenclature

V_{ds}, V_{qs} : d-q axes stator voltages.

i_{ds}, i_{qs} : d-q axes stator currents.

ϕ_{ds}, ϕ_{qs} : d-q axes voltage model rotor fluxes in stationary reference frame.

$\lambda_{ds}, \lambda_{qs}$: d-q axes current model rotor fluxes in stationary reference frame.

$\sigma = 1 - \frac{L_m^2}{L_s L_r}$: Leakage coefficient.

ω_r : Electrical rotor angular velocity.

$T_r = \frac{L_r}{R_r}$: Rotor time constant.

References

1. Brunner, C.U.; Waide, P.; Jakob, M. Harmonized Standards for Motors and Systems Global Progress Report and Outlook. In Proceedings of the 7th International Conference EEMODS'11 Energy Efficiency in Motor Driven Systems; Publications Office of the European Union: Luxembourg, 2013; pp. 444–456.
2. Gómez-Espinosa, A.; Hernández-Guzman, V.M.; Bandala-Sanchez, M.; Jiménez-Hernández, H.; Rivas-Araiza, E.A.; Rodríguez-Resendiz, J.; Herrera-Ruíz, G. A New Adaptive Self-Tuning Fourier Coefficients Algorithm for Periodic Torque Ripple Minimization in Permanent Magnet Synchronous Motors (PMSM). *Sensors* **2013**, *13*, 3831–3847.
3. Mouna, B.H.; Lassaad, S. Neural network speed controller for direct vector control of induction motors. *Int. J. Eng. Sci. Technol.* **2010**, *2*, 7470–7480.
4. Gutierrez-Villalobos, J.M.; Rodriguez-Resendiz, J.; Rivas-Araiza, E.A.; Mucinob, V.H. A review of parameter estimators and controllers for induction motors based on artificial neural networks. *Neurocomputing* **2013**, *118*, 87–100.
5. Rubaai, A.; Kotaru, R.; Kankam, M.D. Online training of parallel neural network estimators for control of induction motors. *IEEE Trans. Ind. Appl.* **2001**, *37*, 1512–1521.

6. Ye, X.; Zhang, Q.; Zhang, T. Speed estimation of induction motor based on neural network. In Proceedings of the International Conference on Intelligent Control and Information Processing (ICICIP), Harbin, China, 25–28 July 2011; pp. 619–623.
7. Chen, T.-C.; Sheu, T.-T. Model reference neural network controller for induction motor speed control. *IEEE Trans. Energy Convers.* **2002**, *17*, 157–163.
8. Korlinchak, C.; Comanescu, M. Sensorless field orientation of an induction motor drive using a time-varying observer. *IET Electr. Power Appl.* **2012**, *6*, 353–361.
9. Roncero-Sánchez, P.; Garcia-Cerradab, A.; Feliu-Batllea, V. Rotor-resistance estimation for induction machines with indirect-field orientation. *Control Eng. Pract.* **2007**, *15*, 1119–1133.
10. Marei, M.I.; Shaaban, M.F.; El-Sattar, A.A. A speed estimation unit for induction motors based on adaptive linear combiner. *Energy Convers. Manag.* **2009**, *50*, 1664–1670.
11. Tsai, C.-H. CMAC-based speed estimator design for induction motor drive. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), Taipei, Taiwan, 8–11 October 2006; pp. 2608–2614.
12. Bechouche, A.; Sediki, H.; Abdeslam, D.O.; Haddad, S. A Novel Method for Identifying Parameters of Induction Motors at Standstill Using ADALINE. *IEEE Trans. Energy Convers.* **2012**, *27*, 105–116.
13. Sedikia, H.; Bechouchea, A.; Abdeslamb, D.O.; Haddada, S. ADALINE approach for induction motor mechanical parameters identification. *Math. Comput. Simul.* **2013**, *9*, 86–97.
14. Brandstetter, P.; Cajka, R.; Skuta, O. Rotor time constant adaptation with ANN application. In proceedings of the European Conference on Power Electronics and Applications, Aalborg, Denmark, 2–5 September 2007; pp. 1–10.
15. Liu, H.-K.; Tsai, C.-H.; Lu, H.-C. CMAC neural network application for induction motor drives. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Anchorage, AK, USA, 9–12 October 2011; pp. 2279–2284.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).