Design of a Morlet wavelet control algorithm using super–twisting sliding modes applied to an induction machine

Daniel A. Magallón, Carlos E. Castañeda
Centro Universitario de los Lagos
Universidad de Guadalajara
Lagos de Moreno, Jalisco, México
daniel.magallon6532@academicos.udg.mx
ccastaneda@lagos.udg.mx

Francisco Jurado
Tecnológico Nacional de México
I.T. La Laguna
Torreón
Coahuila de Zaragoza, México
fjurado@itlalaguna.edu.mx

Onofre A. Morfin
Instituto de Ingeniería y Tecnología
Universidad Autónoma de Ciudad Juárez
Ciudad Juárez, Chihuahua, México
omorfin@uacj.mx

***Abstract*—In this paper, a Morlet wavelet and super–twisting control algorithm are designed and implemented to a three–phase induction motor. The mathematical model of the squirrel–cage induction motor to be controlled is approximated by the Morlet wavelet artificial neural network, which is trained on–line with the error filtered algorithm in order to reproduce the dynamics of the plant to be controlled. The structure of the artificial neural network is proposed in series–parallel configuration and block control form to design the sliding variety, where the super–twisting control algorithm is applied indirectly. For the non–measurable state variables of the plant, state observers of the super–twisting type are proposed to feed the inputs of the artificial neural network. The simulation of the complete system in closed loop is performed where the variables to be controlled are the angular velocity and the square modulus of flux linkages. The results obtained in Matlab/Simulink validate the efficiency of the proposed neural network for the identification of states and the application of the controller.**

***Index Terms*—Morlet wavelet neural network, super–twisting controller, state observer**

I. Introduction

A machine that only has damping windings is called induction machine because the rotor voltage (which produces the current and the magnetic field of the rotor) is induced in the rotor windings instead of being physically connected through wires. The distinctive feature of an induction motor is that no direct current field is required to operate the machine [1]. In recent works, authors use the mathematical model of the squirrel–cage induction motor for describing the plant and to identify its parameters when using dynamic and steady–state tests [2]. In [3], authors work with a squirrel–cage induction motor (AC) since its design has a simple and resistant structure in contrast with other machines. In that work they control the angular velocity and flux linkages through the application of the block control technique and super–twisting sliding modes. Also [3], the authors develop state observers for the variables that are not available for measurement, such as flux linkages, since the work is developed in real time.

On the one hand, super–twisting control algorithm plays an important role in the design of second–order sliding mode controllers due to its robustness under parametric variations. The super–twisting algorithm in turn not only reduces but completely eliminates the chattering effect, however, the total elimination of the chattering significantly increases the robustness of the controller [4]. A recent work proposes the implementation of a super–twisting control algorithm based on an observer of super–twisting states. Such work presents the stabilization of the feedback output of a disturbed dual integrator system [5].

On the other hand, the function of an artificial neural network within a system is to identify the behavior of the dynamics of the plant. The problem with the neural identification is to select an appropriate model for the task as well as the adjustment of the parameters according to some adaptation law, so that the response of the neural identifier to an input signal approximates the response of the system to the same input [4]. In the work referenced above, authors identify real–time results through a high–order artificial neural network and block control transformation using high–order sliding modes. In fact, the control algorithm includes the combination of a recurrent neural network with the block control transformation using the high–order sliding mode technique as a control law.

However, there are different works where artificial neural networks are applied to the identification of the induction machine; this is the case in the work [6] but the authors use a recurrent high order neural network in discrete–time in order to control the induction machine.

Also, artificial neural networks (ANNs) represent a methodology in many disciplines such as neuroscience, mathematics, statistics, physics, computer science, engineering, and so on. ANNs find application in diverse fields such as modeling, time–series analysis, pattern recognition, signal and image processing and control by virtue of an important property, namely the ability to learn from input data, with or without training [7]. However, when it comes to dynamic systems, classical ANNs do not constitute a good option, for this reason the interest has been directed towards the use of recurrent neural networks (RNNs) by including in their structure feed-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

back loops facilitating the modeling, identification and design of observers for non–linear systems. In [8], [9], [10] the approximation and learning properties of a class of RNNs known as recurrent high–order neural networks (RHONNs) have been studied. The stability and convergence properties of a RHONN as a model of nonlinear dynamic systems were also studied, demonstrating that with a sufficiently large number of high–order connections between neurons it is possible to make the RHONN model able to approximate a large number of complex dynamics systems. Identification schemes have been developed mainly to obtain mainly the values of the unknown parameters from a system, which are commonly considered as constants, based on their measurable variables such as inputs, outputs and possible disturbances [11].

Moreover, wavelet neural networks (WNNs) are a structure that combines the concept of wavelet functions with the resource of ANNs with the objective of improving identification performance. Unlike the sigmoidal activation functions used in conventional ANNs, Morlet wavelet activation functions are used in WNNs [12]. In a recent work [15], authors use a novel wavelet neural network structure called recurrent wavelet first–order neural network (RWFONN) designed to control a robot manipulator of two degrees of freedom, that evolves in the vertical plane, through the design of a centralized neural integrator backstepping control.

The Morlet wavelet is a complex wave within a Gaussian envelope. Other types of wavelets that do not have Gaussian envelopes cannot be expected to improve on the time–frequency localization of the Morlet wavelet [13]. The Gaussian and Mexican hat wavelet, together with classical sigmoid functions, are commonly used in the structure of WNNs. There is a lack of works about the use of the Morlet wavelet in the structure of WNNs. There are a few works about the use of WNNs are focused on estimation and time–series prediction [14].

In this work, Morlet wavelets are applied in particular to the artificial neural network as an activation function to perform neuronal identification of the mathematical model of an induction motor; based on a recurrent neural network whose activation function is a sigmoid function. This is done to know the dynamics obtained with this new Morlet wavelet neural network.

Furthermore, this work presents the design and implementation of identification and control in continuous–time of an induction machine using the Morlet wavelet artificial neural network and the super–twisting algorithm applied to this neural network, and where the training algorithm used is the filtered error. For the unmeasurable variables, two–state observers are used: the flux linkages observer with first–order sliding modes and a Luenberger reduced asymptotic observer for the load torque. In the case of the flux linkages observer, the variables $\hat{\lambda}_{\alpha r}$ and $\hat{\lambda}_{\beta r}$ feed their corresponding variables of the neural network; and for the case of the estimated variable $\hat{T}_L$ for estimation of the external disturbance of the plant to be controlled.

This paper is organized as follows: in Section II, the mathematical model of the squirrel–cage induction motor is presented; Section III introduces the state observer to estimate the rotor flux linkages and the load torque; in Section IV presents the Recurrent Wavelet First Order Neural Network (RWFONN) as well as the filtered error learning algorithm; in Section V presents the feedback linearization technique by block control applied to RWFONN; the super–twisting algorithm is described in Section VI; in Section VII are shown the simulation results; conclusion and some remarks about the application of control on RWFONN are drawn in Section VIII.

II. SQUIRREL–CAGE INDUCTION MOTOR MATHEMATICAL MODEL

In this work is used the squirrel–cage induction motor mathematical model presented in [3] as well as the Clarke transformation matrix in order to refer this electrical variables abc to the $\alpha\beta$ coordinate frame. This model is represented as

$$\begin{aligned}
\frac{d}{dt}\omega_m =& K_T(i_{\beta s}\lambda_{\alpha r} - i_{\alpha s}\lambda_{\beta r}) - \frac{B_m}{J_m}\omega_m - \frac{1}{J_m}T_L \\
\frac{d}{dt}\lambda_{\alpha r} =& -\frac{1}{T_r}\lambda_{\alpha r} - \frac{P}{2}\omega_m\lambda_{\beta r} + \frac{L_m}{T_r}i_{\alpha s} \\
\frac{d}{dt}\lambda_{\beta r} =& \frac{P}{2}\omega_m\lambda_{\alpha r} - \frac{1}{T_r}\lambda_{\beta r} + \frac{L_m}{T_r}i_{\beta s} \\
\frac{d}{dt}i_{\alpha s} =& \frac{\delta}{T_r}\lambda_{\alpha r} + \frac{P}{2}\delta\omega_m\lambda_{\beta r} - \gamma i_{\alpha s} + \frac{1}{\sigma L_s}v_{\alpha s} \\
\frac{d}{dt}i_{\beta s} =& -\frac{P}{2}\delta\omega_m\lambda_{\alpha r} + \frac{\delta}{T_r}\lambda_{\beta r} - \gamma i_{\beta s} + \frac{1}{\sigma L_s}v_{\beta s}
\end{aligned} \tag{1}$$

where the states variables are: ω_m as the angular mechanical velocity, $\lambda_{\alpha r}$ and $\lambda_{\beta r}$ are the rotor flux linkages at α and β axis, respectively; $i_{\alpha s}$ and $i_{\beta s}$ are the stator currents; also with constant parameters K_T, T_r, L_m, δ, σ, γ, L_s, moreover all electrical parameter are defined in references [2], [3]; P is the number of poles; B_m is the shaft frictional coefficient, and J_m is the inertial moment. T_L is load torque and it represents a mechanical input for the system (1), $v_{\alpha s}$ and $v_{\beta s}$ are the α and β voltages and represent the control inputs.

III. STATE OBSERVERS

An asymptotic state observer is a dynamic system that provides an estimate of the internal state of a physics system using measurements of its inputs and outputs directly. It is essential to develop the state observers for the non–measurable variables of the induction motor because in the application of the ANN for identification of the states it is necessary to know all the states from the original plant.

For the purposes of this work, we use state observers for the flux linkages and load torque for the squirrel–cage induction motor, because in real–time, these variables commonly are not available for measurement, even when we only show simulation results.

A. Flux linkages observer with first-order sliding modes

In this work, it is used a flux linkages observer with first order sliding modes applied to the squirrel–cage induction motor presented in [3]. This robust observer has the from

$$\begin{aligned}\frac{d}{dt}\hat{\boldsymbol{\lambda}}_r =&\mathbf{A}_{11}\hat{\boldsymbol{\lambda}}_r + \mathbf{A}_{12}\hat{\mathbf{i}}_s - \mathbf{G}_1\mathbf{v}\\ \frac{d}{dt}\hat{\mathbf{i}}_s =&\mathbf{A}_{21}\hat{\boldsymbol{\lambda}}_r + \mathbf{A}_{22}\hat{\mathbf{i}}_s + \mathbf{B}_2\mathbf{v}_s + \mathbf{v}\end{aligned} \tag{2}$$

where $\mathbf{v} = \mathbf{N}\ \text{sign}\left(\mathbf{i}_s - \hat{\mathbf{i}}_s\right)$, $\mathbf{N} = \begin{bmatrix} N_{11} & 0 \\ 0 & N_{22}\end{bmatrix}$ and $\mathbf{G}_1 = \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22}\end{bmatrix}$, for $\mathbf{A}_{11}$, $\mathbf{A}_{12}$, $\mathbf{A}_{21}$ and $\mathbf{A}_{22}$ taken from [3], which represent some parameters from the induction motor (1), and $\hat{\boldsymbol{\lambda}}_r = \begin{bmatrix}\hat{\lambda}_{\alpha r} & \hat{\lambda}_{\beta r}\end{bmatrix}^\top$ represents the observed rotor flux vector and $\hat{\mathbf{i}}_s = \begin{bmatrix}\hat{i}_{\alpha s} & \hat{i}_{\beta s}\end{bmatrix}^\top$ is the observed stator current vector. The discontinuous function $\mathbf{v}$ is chosen such that the sliding mode is enforced in the surface $\mathbf{s} = 0$ and $\mathbf{N}$ is a diagonal matrix, where the components have high enough positive values for steering the surface toward zero in finite time. Meanwhile, matrix $\mathbf{G}_1$ must satisfy vanishing of the rotor fluxes observation error vector $\varepsilon_\lambda = \boldsymbol{\lambda}_s - \hat{\boldsymbol{\lambda}}_s = 0$ on the specified rate [3]. The validation of the observer for the flux linkages $\hat{\lambda}_{\alpha r}$ and $\hat{\lambda}_{\beta r}$ is done by means of the convergence of the currents observed $\hat{i}_{\alpha s}$ and $\hat{i}_{\beta s}$ using (2).

B. Load torque Luenberger observer

The mechanical part of the induction motor, including the load torque equation with smooth movement, has the following representation [3]

$$\begin{aligned}\frac{d}{dt}\omega_m &= K_T\boldsymbol{\lambda}_r^\top\mathbf{M}\mathbf{i}_s - \frac{1}{T_m}\omega_m - \frac{1}{J_m}T_L\\ \frac{d}{dt}T_L &= 0\end{aligned} \tag{3}$$

where K_T is a constant parameter, $\boldsymbol{\lambda}_r^\top = [\lambda_{\alpha r}\ \lambda_{\beta r}]$, $\mathbf{M} = \begin{bmatrix} 0 & 1 \\ -1 & 0\end{bmatrix}$ and $\mathbf{i}_s = [i_{\alpha s}\ i_{\beta s}]$. Based on the work made in [3], the observer of the load torque has the following representation

$$\begin{aligned}\frac{d}{dt}\hat{\omega}_m =&K_T\boldsymbol{\lambda}_r^\top\mathbf{M}\mathbf{i}_s - \frac{1}{T_m}\hat{\omega}_m - \frac{1}{J_m}\hat{T}_L + l_1(\omega_m - \hat{\omega}_m)\\ \frac{d}{dt}\hat{T}_L =&l_2(\omega_m - \hat{\omega}_m)\end{aligned} \tag{4}$$

where $\hat{\omega}_m$ is the observed angular mechanical velocity, $\hat{T}_L$ is the observed load torque, l_1 and l_2 are constants, which are selected to ensure the asymptotic convergence to zero at a specified rate of the angular velocity observation error $\varepsilon_\omega = \omega_m - \hat{\omega}_m$. To validate the load torque observer $\hat{T}_L$, this work shows the convergence of the angular velocity observer $\hat{\omega}_m$ using (4).

IV. RECURRENT WAVELET FIRST ORDER NEURAL NETWORK

Recurrent neural networks (RNNs) are characterized by having feedback loops between neurons, which distinguishes them from pre–fed neuronal commands, where the output of each neuron is connected only to the neurons of the next layer. In the simplest case of a RNN [16], the state of each neuron can be determined by a differential equation of the form [10]

$$\dot{x}_i = -a_i x_i + b_i\sum_{k=1}^{L} w_{ik}y_k \tag{5}$$

where x_i is the state of the i–th neuron, a_i and b_i are positive real constants, w_{ik} is the synaptic weight connecting the k–th state to the i–th neuron. Each y_i is the external input or the state of a neuron passed through a sigmoid function, i.e., $y_k = s(x_k)$, where $s(\cdot)$ represents a sigmoid function. In a second–order RNN, the input to the neuron is not only the linear combination of its components y_k but also of their products $y_k y_j$ to include high order interactions represented by triplets $y_k y_j y_l$, quadruplets, etc., forming the Recurrent High Order Neural Network (RHONN)

$$\dot{x}_i = -a_i x_i + b_i\sum_{k=1}^{L} w_{ik}z_k \tag{6}$$

From the theory for RHONN model [17], replacing now the vector z by a wavelet vector ψ and considering that higher–order terms will not be used, the RHONN model (6) can be rewritten as

$$\dot{x}_i = -a_i x_i + b_i\sum_{k=1}^{L} w_{ik}\psi_{ik} \tag{7}$$

Defining the adjustable parameter vector as $w_i = b_i\left[w_{i1}\ w_{i2}\ ...\ w_{iL}\right]^\top$, so (7) becomes [18]

$$\dot{x}_i = -a_i x_i + (w_{ik})^\top\psi_{ik} \tag{8}$$

where the vectors w_i represent the adjustable weights of the network while the coefficients a_i for $i = 1, 2, ..., n$ are part of the underlying network architecture and are fixed during training. The structure in the form (8) here is called as Recurrent Morlet Wavelet Neural Network (RMWNN) [18]. Using the theory of [10], [16]–[18], [20] for the construction of a novel neural network structure called Recurrent Wavelet First Order Neural Network (RWFONN), where a neuron with a single connection of first order, the sigmoidal activation function $s(\cdot)$ is replaced by the real version of the modified Morlet wavelet [20] of the form $\psi(x) = e^{-x^2/\beta}\cos(\lambda x)$, with parameters β and λ representing expansion and dilation, respectively. Thus, it is called of first order because in this novel neural network structure the high order terms of the RHONN structure are eliminated.

A. RWFONN Structure

The proposed structure RWFONN with the form (8) of the neural network identification for the squirrel–cage induction motor states (1) is as follows

$$\begin{aligned}
\dot{x}_1 =& - a_1 x_1 + b_1 w_1 y_1(\chi_1) + b_{13} w_{13} y_{13}(\hat{\chi}_3) \\
&+ b_{14} w_{14} y_{14}(\chi_4) + b_2(x_2 x_4 - x_3 x_5) \\
\dot{x}_2 =& - a_2 x_2 + b_2 w_2 y_2(\hat{\chi}_2) + x_4 \\
\dot{x}_3 =& - a_3 x_3 + b_3 w_3 y_3(\hat{\chi}_3) + x_5 \\
\dot{x}_4 =& - a_4 x_4 + b_4 w_4 y_4(\chi_4) + v_{\alpha s} \\
\dot{x}_5 =& - a_5 x_5 + b_5 w_5 y_5(\chi_5) + v_{\beta s}
\end{aligned} \tag{9}$$

where χ_1, χ_4, and χ_5 represent the states ω_m, $i_{\alpha s}$, and $i_{\beta s}$ respectively, from the induction motor (1); $\hat{\chi}_2$ and $\hat{\chi}_3$ are the estimated states $\hat{\lambda}_{\alpha r}$ and $\hat{\lambda}_{\beta r}$, from (2), of the state variables $\lambda_{\alpha r}$ and $\lambda_{\beta r}$, from (1), respectively. The neural network states x_1 to x_5 identify their corresponding state variables χ_1, $\hat{\chi}_2$, $\hat{\chi}_3$, χ_4, and χ_5, respectively. a_i and b_i, $(i = 1, 2, ..., 5)$ are constant parameters, w_i are the synaptic weights which are adjusted on–line using the filtered error algorithm for the neural network and y_i are the Morlet wavelet activation functions. Finally, $v_{\alpha s}$ and $v_{\beta s}$ represent the voltage control inputs to the induction motor in the α and β axis.

B. Error filtered learning algorithm

This section describes an adjustable law for the weights under the assumption that the unknown system can be modeled exactly by an architecture of the RHONN type [17]. The development of this identification scheme starts from the differential equation that describes the unknown system

$$\dot{\chi}_i = -a_i \chi_i + \boldsymbol{w}_i^{*\top} \boldsymbol{z} \tag{10}$$

where χ_i represents each state of the unknown dynamic system. From (10), the identifier is chosen as:

$$\dot{x}_i = -a_i x_i + \boldsymbol{w}_i^{\top} \boldsymbol{z} \tag{11}$$

which identifies the states of the system; x_i represents each state of the neural network and $\boldsymbol{w}_i$ is the estimate of the unknown vector $\boldsymbol{w}_i^*$. In this case, the identification error is defined as $\xi_i = x_i - \chi_i$, and its respective derivative is given as

$$\begin{aligned}
\dot{\xi}_i = \dot{x}_i - \dot{\chi}_i =& - a_i x_i + \boldsymbol{w}_i^{\top} \boldsymbol{z} - (-a_i \chi_i + \boldsymbol{w}_i^{*\top} \boldsymbol{z}) \\
=& - a_i x_i + \boldsymbol{w}_i^{*\top} \boldsymbol{z} + a_i \chi_i - \boldsymbol{w}_i^{*\top} \boldsymbol{z} \\
=& - a_i (x_i - \chi_i) + (\boldsymbol{w}_i^{\top} - \boldsymbol{w}_i^{*\top})
\end{aligned} \tag{12}$$

which can be rewritten as

$$\dot{\xi}_i = -a_i \xi_i + \widetilde{\boldsymbol{w}}_i^{\top} \boldsymbol{z} \tag{13}$$

where $\widetilde{\boldsymbol{w}} = \boldsymbol{w}_i - \boldsymbol{w}_i^*$. The weights $\boldsymbol{w}_i$ for $i = 1, 2, ..., n$ are adjusted according to the learning law [9]

$$\dot{\boldsymbol{w}}_i = -\boldsymbol{\Gamma}_i \boldsymbol{z} \xi_i \tag{14}$$

where the learning rate $\boldsymbol{\Gamma}_i \in \mathbb{R}^{L \times L}$ is a definite positive matrix. In the special case that $\boldsymbol{\Gamma}_i = \gamma_i \boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix and $\gamma_i > 0$ is a scalar, then $\boldsymbol{\Gamma}_i$ can be replaced by γ_i yielding

$$\dot{\boldsymbol{w}}_i = -\boldsymbol{\gamma}_i \boldsymbol{z} \xi_i \tag{15}$$

Equations (14) and (15) the so-called error filtered. Stability and convergence properties of the adjustable weight law given by (14) are well known in the adaptive control literature [17] [19].

V. BLOCK CONTROL LINEARIZATION APPLIED TO A RWFONN

The block control linearization consists of transforming the mathematical model of the plant to be controlled (in this case the structure of the RWFONN (9)) to the block control form, which is as follows [3]

$$\begin{aligned}
\dot{\mathbf{z}}_1 =& - \mathbf{k}_1 \mathbf{z}_1 + \mathbf{E}_{11} \mathbf{z}_2 + \varphi_1(\mathbf{z}_1, t) \\
\dot{\mathbf{z}}_i =& - \mathbf{k}_i \mathbf{z}_i + \mathbf{E}_{i1} \mathbf{z}_{i+1} + \varphi_i(\mathbf{z}_i, t), \ i = 2, ..., r-1 \\
\dot{\mathbf{z}}_r =& \bar{\mathbf{f}}_r(\mathbf{z}, t) + \overline{\mathbf{B}}_r(\mathbf{z}, t) \mathbf{u} + \overline{\varphi}_r(\mathbf{z}, t)
\end{aligned} \tag{16}$$

where $\mathbf{z} = [z_1, ..., z_r]^{\top}$ is the tracking error vector, $\bar{\mathbf{f}}_r(\mathbf{z}, t)$ is a bounded function, the rank$(\mathbf{B}(\mathbf{z}, t)) = m$, $\varphi(\mathbf{z}, t)$ characterizes the external disturbances and parameter variations, $\mathbf{E}_{i1}$ is a matrix with constant values, and $\mathbf{u}$ is the input vector; the matrix $\overline{\mathbf{B}}_r(\mathbf{z}, t)$ represents a constant matrix to establish an affine control. In order to apply the block control linearization to the structure of the RWFONN we rewrite (9) in vector form as follows

$$\begin{aligned}
\dot{x}_1 =& - a_1 x_1 + b_1 w_1 y_1(\chi_1) + b_{13} w_{13} y_{13}(\hat{\chi}_{13}) \\
&+ b_{14} w_{14} y_{14}(\chi_4) + b_2 \mathbf{x}_2^{\top} \mathbf{i}_1 \mathbf{x}_3 \\
\dot{\mathbf{x}}_2 =& \mathbf{C}_{11} \mathbf{x}_2 + \mathbf{C}_{12} \mathbf{i}_2 + \mathbf{i}_3 \mathbf{x}_3 \\
\dot{\mathbf{x}}_3 =& \mathbf{C}_{21} \mathbf{x}_3 + \mathbf{C}_{22} \mathbf{i}_2 + \mathbf{i}_3 \mathbf{v}
\end{aligned} \tag{17}$$

with $r = 3$, so the state vectors are $\mathbf{x}_2 = [x_2 \ x_3]^{\top}$ where x_2 and x_3 represent the states of the neural network that identify the flux linkages estimated $\hat{\lambda}_{\alpha r}$ and $\hat{\lambda}_{\beta r}$ respectively from (2), $\mathbf{x}_3 = [x_4 \ x_5]^{\top}$ and $\mathbf{v} = [v_{\alpha s} \ v_{\beta s}]^{\top}$. The parametric matrices are defined as

$$\mathbf{C}_{11} = \begin{bmatrix} -a_2 & 0 \\ 0 & -a_3 \end{bmatrix}; \ \mathbf{C}_{12} = \begin{bmatrix} b_2 w_2 y_2 & 0 \\ 0 & b_3 w_3 y_3 \end{bmatrix};$$

$$\mathbf{C}_{21} = \begin{bmatrix} -a_4 & 0 \\ 0 & -a_5 \end{bmatrix}; \ \mathbf{C}_{22} = \begin{bmatrix} b_4 w_4 y_4 & 0 \\ 0 & b_5 w_5 y_5 \end{bmatrix};$$

$$\mathbf{i}_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; \ \mathbf{i}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \ \mathbf{i}_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Now, applying the block control linearization technique to the structure of the neural network (17) and using the state observer of flux linkages (2), the error vector $\mathbf{z}_1$ is defined as:

$$\mathbf{z}_1 = \begin{bmatrix} \omega_{mref} - x_1 \\ \varphi_{ref} - \varphi_r \end{bmatrix} \tag{18}$$

where x_1 represents the identified angular velocity ω_m, φ_r is the square modulus of rotor flux linkages, ω_{mref} is the desired angular velocity, and φ_{ref} is the desired square modulus for

the rotor flux linkages. In order to control the flux linkages estimated by the state observer (2), we use the methodology from [3] in order to find the square modulus of rotor flux linkages

$$\varphi_r = |\mathbf{x}_2|^2 = \mathbf{x}_2^\top \mathbf{x}_2 = x_2^2 + x_3^2 \tag{19}$$

In the same way, φ_{ref} is obtained using the methodology from [3] such that

$$\varphi_{ref} = \frac{2\mathbf{x}_2^\top \mathbf{x}_3}{(a_2 + a_3)} \tag{20}$$

Then, the dynamics of the tracking error vector (18), that implies the neural network model (17) and introduces a specified steady first order dynamics, takes the form:

$$\begin{aligned}\dot{\mathbf{z}}_1 = &\begin{bmatrix} \dot{\omega}_{mref} + a_1 x_1 \\ \dot{\varphi}_{ref} + (a_2 + a_3)\varphi_r \end{bmatrix} - \begin{bmatrix} b_2 \mathbf{x}_2^\top \mathbf{i}_1 \\ 2\mathbf{x}_2^\top \end{bmatrix} \mathbf{x}_3 - \\ &\begin{bmatrix} b_1 w_1 y_1(\chi_1) + b_{13} w_{13} y_{13}(\hat{\chi}_3) + b_{14} w_{14} y_{14}(\chi_4) \\ 0 \end{bmatrix} = \\ &-\mathbf{K}_1 \mathbf{z}_1 \end{aligned} \tag{21}$$

where the adjustment matrix $\mathbf{K}_1$ is adjusted to vanish the vector of the tracking error variables such as

$$\mathbf{K}_1 = \begin{bmatrix} K_{11} & 0 \\ 0 & K_{22} \end{bmatrix}$$

Using the block control linearization technique in the neural network system (17), two first–order subsystems are created which can be solved independently. In this way, it is possible to develop a control law through the state $\mathbf{x}_3$ from (17), where the reference vector is defined from (21), as follows

$$\begin{aligned}\mathbf{x}_{3ref} = &\begin{bmatrix} b_2 \mathbf{x}_2^\top \mathbf{i}_1 \\ 2\mathbf{x}_2^\top \end{bmatrix}^{-1} \begin{bmatrix} \dot{\omega}_{mref} + a_1 x_1 \\ \dot{\varphi}_{ref} + (a_2 + a_3)\varphi_r \end{bmatrix} - \\ &\begin{bmatrix} b_1 w_1 y_1(\chi_1) + b_{13} w_{13} y_{13}(\hat{\chi}_3) + b_{14} w_{14} y_{14}(\chi_4) \\ 0 \end{bmatrix} \\ &+\mathbf{K}_1 \mathbf{z}_1 \end{aligned} \tag{22}$$

Therefore, the second error vector is defined as

$$\mathbf{z}_2 = \mathbf{x}_{3ref} - \mathbf{x}_3 \tag{23}$$

with dynamics

$$\dot{\mathbf{z}}_2 = \dot{\mathbf{x}}_{3ref} - \dot{\mathbf{x}}_3 \tag{24}$$

where

$$\dot{\mathbf{z}}_2 = \dot{\mathbf{x}}_{3ref} - (\mathbf{C}_{21}\mathbf{x}_3 + \mathbf{C}_{22}\mathbf{i}_2 + \mathbf{i}_3 \mathbf{v}) \tag{25}$$

Finally, using (21), (25) and one steady state equation from the neural network (9) as zero dynamics, the representation of the model (17) in the block control form (16) is obtained as

$$\begin{aligned}\dot{\mathbf{z}}_1 =& -\mathbf{K}_1 \mathbf{z}_1 + \begin{bmatrix} b_2 \mathbf{x}_2^\top \mathbf{i}_1 \\ 2\mathbf{x}_2^\top \end{bmatrix} \mathbf{z}_2 \\ \dot{\mathbf{z}}_2 =& \dot{\mathbf{x}}_{3ref} - (\mathbf{C}_{21}\mathbf{x}_3 + \mathbf{C}_{22}\mathbf{i}_2 + \mathbf{i}_3 \mathbf{v}) \\ \dot{x}_2 =& -a_2 x_2 + b_2 w_2 y_2(\hat{\chi}_2) + x_4 \end{aligned} \tag{26}$$

The state equation x_2 in (26) represents an internal dynamics to complete the order of the neural network and to consider the zero dynamics of the system. From (26), it is important to mention that the application of the block control technique is to decouple the control vector $\mathbf{v}$ and then to apply the super–twisting control strategy.

VI. SUPER–TWISTING CONTROLLER

In order to design a super–twisting control algorithm that induces the sliding surface $\mathbf{s} = 0$ and its respective derivative $\dot{\mathbf{s}} = 0$, the super-twisting control algorithm is applied [3]

$$\mathbf{v}_s = \lambda |\mathbf{s}|^{\frac{1}{2}} \text{sign}(\mathbf{s}) + \mathbf{v}_{1s} \tag{27}$$

where $\dot{\mathbf{v}}_{1s} = \boldsymbol{\alpha}\ \text{sign}(s)$, $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_\alpha & 0 \\ 0 & \lambda_\beta \end{bmatrix}$ and $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_\alpha & 0 \\ 0 & \alpha_\beta \end{bmatrix}$.

The diagonal matrices $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}$ have components that are selected such that the sliding surface $\mathbf{s} = \mathbf{z}_2$ converges to zero in a finite time. By applying the super–twisting control algorithm (27) in the block control structure (26), the neuronal controller is obtained as follows

$$\begin{aligned}\dot{\mathbf{z}}_1 =& -\mathbf{K}_1 \mathbf{z}_1 + \begin{bmatrix} b_2 \mathbf{x}_2^\top \mathbf{i}_1 \\ 2\mathbf{x}_2^\top \end{bmatrix} \mathbf{z}_2 \\ \dot{\mathbf{z}}_2 =& \dot{\mathbf{x}}_{3ref} - \mathbf{C}_{21}\mathbf{x}_3 - \mathbf{C}_{22}\mathbf{i}_2 - \mathbf{i}_3(\boldsymbol{\lambda}|\mathbf{s}|^{\frac{1}{2}} \text{sign}(\mathbf{s}) + \mathbf{v}_{1s}) \\ \dot{\mathbf{v}}_{1s} =& \boldsymbol{\alpha}\ \text{sign}(s) \\ \dot{x}_2 =& -a_2 x_2 + b_2 w_2 y_2(\hat{\chi}_2) + x_4 \end{aligned} \tag{28}$$

VII. SIMULATION RESULTS

In order to analyze the dynamics of the neural identification and estimation of the non–measurable variables by the control scheme in closed–loop, the simulation in Matlab/Simulink has been implemented. The values and parameters of the squirrel–cage induction motor are shown in Table I [2].

TABLE I
RATED VALUES AND MOTOR MODEL PARAMETERS.

Parameter	Value
Frequency	$60\ Hz$
Rated Voltage	$127/220\ V$
Power	$0.25\ HP$
Poles	4
Rated current	$1.5\ A$
Rated angular velocity	$1750\ r.p.m.$
Stator resistance R_s	$12\ \Omega$
Rotor resistance R_r	$8.1\ \Omega$
Stator inductance L_s	$0.7066\ H$
Magnetizing inductance L_m	$0.678\ H$
Rotor inductance L_r	$0.7066\ H$
Inertial moment J_m	$0.00324\ N \cdot m \cdot s^2$
Frictional coefficient B_m	$0.00194\ N \cdot m \cdot s$

A. States observer and neural identification results

To perform the neural identification of the squirrel–cage induction motor dynamics, the model (9) is considered using the parameters: $a_1 = 5$, $a_2 = a_3 = a_4 = a_5 = 100$, $b_1 = 5$, $b_{13} = b_{14} = b_4 = b_5 = 100$ and $b_2 = b_3 = 150$.

The parameters for the learning rate γ for the error filtered algorithm are $\gamma_1 = 500$, $\gamma_{13} = \gamma_{14} = 49000$, $\gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = 49000$, and the parameters of expansion β and dilation λ used for the Morlet wavelet activation function are given by $\beta_1 = 10000$, $\beta_{13} = \beta_{14} = \beta_2 = \beta_3 = 20$, $\beta_4 = \beta_5 = 30$, $\lambda_1 = 10000$, $\lambda_{13} = \lambda_{14} = \lambda_2 = \lambda_3 = 0.01$, $\lambda_4 = \lambda_5 = 0.001$.

Fig. 1 shows the estimated torque obtained, where it can be seen that the load torque is given as a constant = 1, and the estimated $\hat{T}_L$ converges to the value given in approximately 0.06 s.

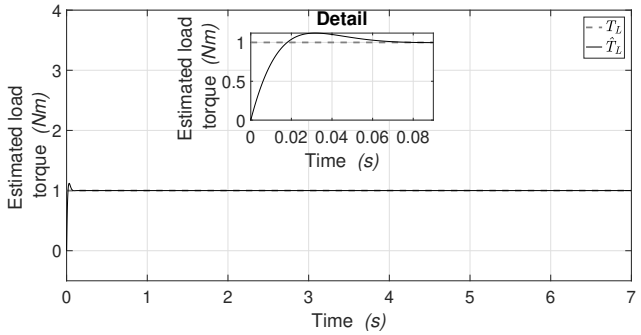

Fig. 1. Load torque and torque observer: T_L (gray dashed line), $\hat{T}_L$ (continuous thin line).

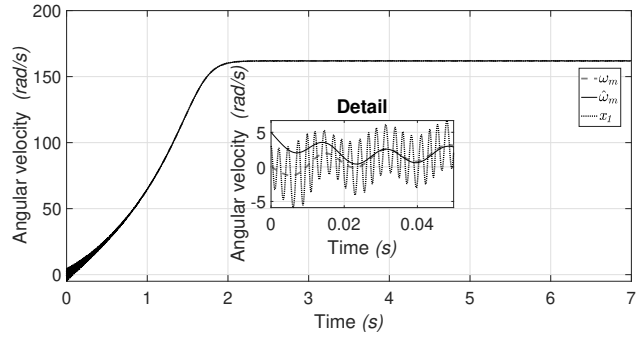

Fig. 2. Angular velocity, state observer and neural identification behavior: ω_m (gray dashed line), $\hat{\omega}_m$ (continuous thin line) and x_1 (dotted tick line).

Figs. 2-6 the results of the wavelet neural identification are presented graphically. In Fig. 2 are shown the angular velocity ω_m, as well as the observed $\hat{\omega}_m$ and the identification x_1 results. In order to show the convergence of these results, the initial conditions are given as $\omega_m = 0$, $\hat{\omega}_m = 5$, and $x_1 = 3$. Note that the identification convergence is given in $1s$ approximately. As it was mentioned before, the estimated result $\hat{\omega}_m$ is for validating the load torque $\hat{T}_L$ estimation.

The estimated of the flux observer $\hat{\lambda}_{\alpha r}$, the neural identification x_2, and $\lambda_{\alpha r}$ are shown in Fig. 3, the initial conditions are $\hat{\lambda}_{\alpha r} = 0.1$, $x_2 = 0.2$, $\lambda_{\alpha r} = 0$. The same for Fig. 4, that shows the flux linkages $\hat{\lambda}_{\beta r}$, x_3, and $\lambda_{\beta r}$, the initial conditions are $\hat{\lambda}_{\beta r} = 0.2$, $x_3 = 0.1$, and $\lambda_{\beta r} = 0$. The convergence of the estimated states and the identification of the neural network indicate that the estimation and approximation errors tend to zero in finite time. The gains for the flux linkages observer are $N_{11} = 1000$, $N_{22} = 900$, $G_{11} = 0.0015$ and $G_{22} = 0.0028$.

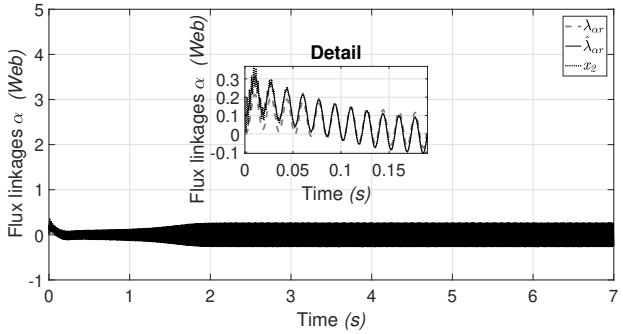

Fig. 3. Flux linkage, state observer and neural identification behavior: $\lambda_{\alpha r}$ (gray dashed line), $\hat{\lambda}_{\alpha r}$ (continuous thin line), and x_2 (dotted tick line).

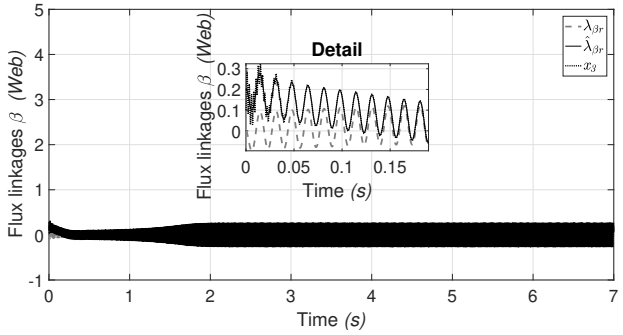

Fig. 4. Flux linkage, state observer and neural identification behavior: $\lambda_{\beta r}$ (gray dashed line), $\hat{\lambda}_{\beta r}$ (continuous thin line), and x_3 (dotted tick line).

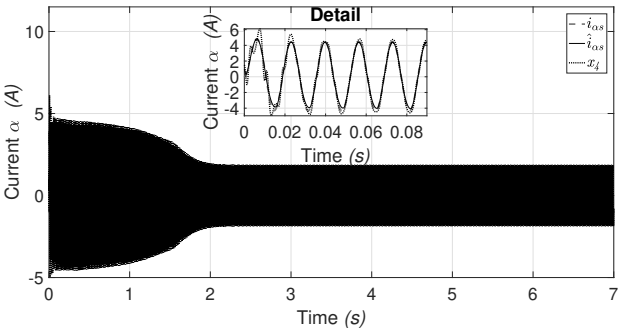

Fig. 5. Current, state observer and neural identification behavior: $i_{\alpha s}$ (gray dashed line), $\hat{i}_{\alpha s}$ (continuous thin line), and x_4 (dotted tick line).

The estimation and neural identification results for the currents $i_{\alpha s}$ and $i_{\beta s}$ are shown in Figs. 5 and 6, respectively, where for the Fig. 5 are $i_{\alpha s}$, $\hat{i}_{\alpha s}$ and x_4, the initial condition are $i_{\alpha s} = 0$, $\hat{i}_{\alpha s} = 1$ and $x_4 = 1.5$ and for Fig. 6 are $i_{\beta s}$, $\hat{i}_{\beta s}$ and x_5; the initial condition are: $i_{\beta s} = 0$, $\hat{i}_{\beta s} = 1$ and $x_5 = 1.5$. Note that these results are very good due to the estimation and identification errors tend to zero when the time tends to infinity. Also, the estimation results $\hat{i}_{\alpha s}$ and $\hat{i}_{\beta s}$ are shown for validation of the estimation results $\hat{\lambda}_{\alpha r}$ and $\hat{\lambda}_{\beta r}$.

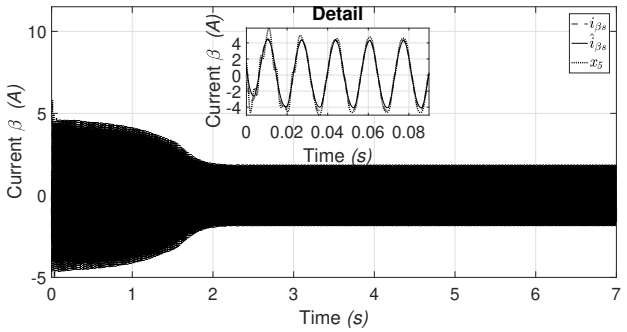

Fig. 6. Current, state observer and neural identification behavior: $i_{\beta s}$ (gray dashed line), $\hat{i}_{\beta s}$ (continuous thin line), and x_5 (dotted tick line).

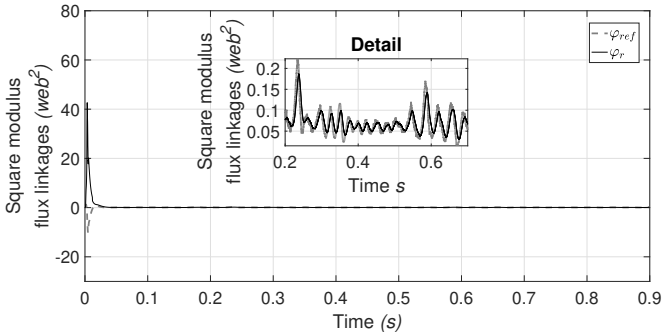

Fig. 8. Rotor flux squared modules linkages performance for trajectory tracking reference: φ_{ref} (gray dashed line), φ_r (continuous thin line).

B. Neural controller

As it was mentioned before, the controller designed in this work is to control the state variables of the angular velocity and the square modulus of flux linkages of the squirrel–cage induction motor through the RWFONN in an indirect way. The simulation of the wavelet neural controller was made using the following adjustable gains $K_{11} = 300$ and $K_{22} = -0.4$. The angular velocity ω_m tracks a constant of 180 rad/s as reference signal, see Fig. 7. It can be noted that the velocity tracking has an effective performance when the system is in closed–loop.

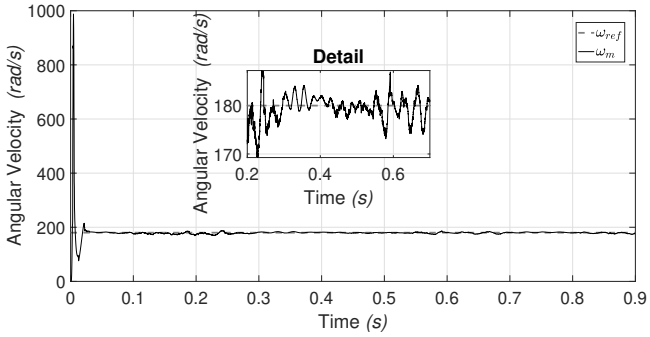

Fig. 7. Angular velocity tracking performance for a constant signal as the trajectory tracking reference: ω_{mref} (gray dashed line), x_1 (continuous thin line).

Fig. 8 shows the tracking performance of rotor flux square modulus, where this tracking is made with good results.

VIII. Conclusion

This paper shows the application of a neural controller on a three–phase induction motor. The controller is composed of a block controller and the application of a high–order controller of super–twisting sliding modes. In addition, this application includes the identification of the states of the plant to be controlled. The state observers are designed to estimate the variables that are not available for their measurement, where the results of these estimations feed their corresponding neural inputs. Also, it is designed the load torque observer, where this variable feeds the plant as external disturbance. The goal of the application of the neural controller is, that it is robust against parametric variations of the induction motor model or modeling errors, in this way to be able to control the angular velocity and the square modulus of flux linkages of the induction motor through the artificial neural network. It should be highlighted that there are works that apply neural networks to approximate mathematical models but here we are trying with the application of a novel wavelet neural network, namely an RWFONN on a model for an energy storage system. As future work, we are working with its implementation in real–time as well as with its comparison with respect to other neural network architectures, in order to provide a baseline neural network, to measure its performance.

Acknowledgment

The first author acknowledges support from Consejo Nacional de Ciencia y Tecnologia (CONACYT) and Universidad de Guadalajara Centro Universitario de los Lagos.

References

[1] S. Chapman, *Electric Machinery*, *McGraw-Hill Education; Edición: 5th Revised ed, ISBN-10: 9780073529547* 2011.

[2] O. Morfín, C. Castañeda, R. Cruz, F. Valenzuela, M. Murillo, A. Quezada and N. Padilla, *The Squirrel-Cage Induction Motor Model and Its Parameter Identification Via Steady and Dynamic Tests*, *Electric Power Components and Systems* 2018.

[3] O. Morfin, F. Valenzuela, R. Betancour, C. Castañeda, R. Cruz and A. Gonzalez, *Real-time SOSM Super-twisting combined with block control for regulating induction motor velocity*, *IEEE Access, 10.1109/ACCESS.2018.2812187* 2018.

[4] S. Rodríguez, C. Castañeda, O. Morfin, F. Jurado, and P. Esquivel, *Real-time result for high order neural identification and block control transformation form using high order sliding modes*, 2016.

[5] A. Chalanga, S. Kamal, L. Fridman, B. Bandyopadhyay, J. Moreno, *Implementation of Super-Twisting Control: Super-Twisting and Higher Order Sliding Mode Observer Based Approaches*, 2015.

[6] A. Alanis, E. Sanchez, and A. Loukianov, *Discrete-time recurrent neural induction motor control using Kalman learning*, *The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp 1993-2000* 1992.

[7] S. Haykin, *Neural networks: a comprehensive foundation*, *Prentice Hall PTRUpper Saddle River, NJ United States, ISBN:978-0-02-352761-6* 1994.

[8] Kosmatopoulos, E. B., Ioannou, P. A., and Christodoulou, M. A, *Identification of nonlinear systems using new dynamic neural network structures*,*Proceedings of the 31st IEEE Conference on Decision and Control, pp 20-25* 1992,

[9] Kosmatopoulos, E. B., Polycarpou, M. M., Christodoulou, M. A., and Ioannou, P. A, *High-order neural network structures for identification of dynamical systems*, *IEEE Transactions on Neural Networks, 6(2):422-431.* 1995.

[10] Kosmatopoulos, E. B., Christodoulou, M. A., and Ioannou, P. A, *Dynamical neural networks that ensure exponential identification error convergence*, *Neural Networks, 10(2):299-314* 1997.

[11] M. Norgaard, O. Ravn, N. Poulsen, and L. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*, *A Practitioner's Handbook Springer London* 2003.

[12] L. Vázquez and F. Jurado, *Continuous-time decentralized wavelet neural control for a 2 dof robot manipulator*, *In 2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp 1-16* 2014.

[13] L. Watkins, *Continuous wavelet transforms. In Phase estimation in optical interferometry*, *(eds P Rastogi, E Hack), pp. 69–120. Boca Raton, FL: CRC Press* 2015.

[14] A. Alexandridis, A. Zapranis, *Wavelet neural networks with applications in financial engineering, chaos, and classification*, *Hoboken, NJ: JohnWiley Sons, Inc* 2014.

[15] L. Vázquez and F. Jurado, C. Castañeda, A. Alanis *Real-Time Implementation of a Neural Integrator Backstepping Control via Recurrent Wavelet First Order Neural Network*, *Neural Processing Letters Springer Science+Business Media, LLC, part of Springer Nature* 2018.

[16] J. Hopfield, *Neurons with graded response have collective computational properties like those of two-state neurons*, *Proceedings of the National Academy of Sciences of the United States of America, 81(10):3088-3092* 1984.

[17] G. Rovithakis and A. Christodoulou, *Adaptive control with recurrent high-order neural networks: Theory and industrial applications*, *Springer, London* 2012.

[18] F. Jurado and S. Lopez, *A wavelet neural control scheme for a quadrotor unmanned aerial vehicle*, *The Royal Society Publishing* 2018.

[19] K. Narendra, and A. Annaswamy, *Stable adaptive systems*, *Prentice-Hall* 2012.

[20] P. Addison, J. Watson, and T. Feng, *Low-oscillation complex wavelets*, *Journal of Sound and Vibration, 254(4):733762* 2002.