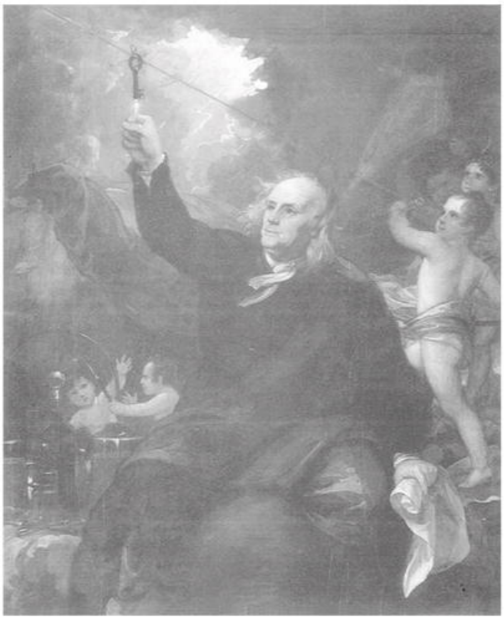
Volume 347 Issue 7 September 2010 ISSN 0016-0032

ELSEVIER

Journal
*of* The Franklin Institute
ENGINEERING AND APPLIED MATHEMATICS



Available online at www.sciencedirect.com
ScienceDirect
Journal of the Franklin Institute 347 (2010) 1253–1265
Journal of The Franklin Institute
www.elsevier.com/locate/jfranklin

# Discrete-time recurrent high order neural networks for nonlinear identification

Alma Y. Alanis[a,*], Edgar N. Sanchez[b], Alexander G. Loukianov[b], Esteban A. Hernandez[c]

[a]*CUCEI, Universidad de Guadalajara, Apartado Postal 51-71, Col. Las Aguilas, C.P. 45080, Zapopan, Jalisco, Mexico*
[b]*CINVESTAV, Unidad Guadalajara, Apartado Postal 31-438, Plaza La Luna, Guadalajara, Jalisco, C.P. 45091, Mexico*
[c]*The Hamilton Institute, NUI Maynooth, Co. Kildare, Ireland*

Received 1 December 2008; received in revised form 15 February 2010; accepted 29 May 2010

**Abstract**

This paper focuses on the problem of discrete-time nonlinear system identification via recurrent high order neural networks. It includes the respective stability analysis on the basis of the Lyapunov approach for the NN training algorithm. Applicability of the proposed scheme is illustrated via simulation for a discrete-time nonlinear model of an electric induction motor.


## 1. Introduction

Modern control systems usually require a very structured knowledge about the system to be controlled; such knowledge should be represented in terms of differential or difference equations. This mathematical description of the dynamic system is named as the model. There can be several motives for establishing mathematical descriptions of dynamic systems, such as simulation, prediction, fault detection, and control system design.

Basically there are two ways to obtain a model; it can be derived in a deductive manner using physics laws, or it can be inferred from a set of data collected during a practical experiment. The first method can be simple, but in many cases is excessively time-consuming; it would be unrealistic or impossible to obtain an accurate model in

*Corresponding author.
*E-mail address:* almayalanis@gmail.com (A.Y. Alanis).

0016-0032/$32.00 © 2010 The Franklin Institute. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.jfranklin.2010.05.018

this way. The second method, which is commonly referred as system identification, could be a useful short cut for deriving mathematical models. Although system identification not always results in an accurate model, a satisfactory model can be often obtained with reasonable efforts. The main drawback is the requirement to conduct a practical experiment, which brings the system through its range of operation [2,11].

Regardless of the fact that all systems in principle are nonlinear, the major part of the literature on system identification deals with the identification of linear systems. However, the nonlinearities are frequently of such severe character that employing a nonlinear model in the control design can enhance the performance of the control system greatly. Obviously, nonlinear modelling is therefore important [11,12]. On the other hand it is well known that, even with one hidden layer, neural networks can uniformly approximate any continuous function over a compact domain, provided the network has sufficient number of units, or neurons [13]. Then it results in a natural alternative, the use of neural networks to construct model structures for nonlinear systems identification.

Neural networks have grown to be a well-established methodology [6], which allows for solving very difficult problems in engineering, as exemplified by their applications to identification and control of general nonlinear and complex systems [11,12]. Neural networks consist of a number of interconnected processing elements or neurons. The way in which the neurons are interconnected determines its structure. For identification and control, the most used structures are *feedforward* and *recurrent* ones [5,4,6,11]. In feedforward neural networks, the neurons are grouped into layers; signals flow from the input to the output via unidirectional connections, the network exhibits high degree of connectivity, contains one or more hidden layers of neurons and the activation function of each network is smooth, generally a sigmoid function. For recurrent neural networks the outputs of the neuron are feedback to the same neuron or neurons in the preceding layers; signals flow in forward and backward directions [14].

Since the seminal paper [10], there has been continuously increasing interest in applying neural networks to identification and control of nonlinear systems, specially recurrent high order neural networks (RHONN) due to their excellent approximation capabilities, using few units. High-order networks are expansions of the first-order Hopfield and Cohen–Grossberg models that allow higher-order interactions between neurons [13]. These kind of neural networks, compared to the first order ones, are more flexible and robust when faced with new and or noisy data patterns [4]. Besides high order neural networks performed better than the multilayer first order ones using a small number of free parameters [5,13]. Furthermore, different authors have demonstrated the feasibility and the advantages of using these architectures in applications for system identification and control; however, most of those works were developed for continuous-time systems [10,13,14], the discrete-time case has not been discussed to the same degree [3,16]. The main motivation for using RHONN in system identification, is because of, the model obtained with the RHONN has a physical meaning, which is not possible with feedforward neural networks. By using RHONN for the identification of dynamic systems, the state of each neuron represents a state variable of the system to be modelled, which means a considerable reduction in the number of neurons in the network [10].

In this paper, a RHONN is used to identify the plant model, whose mathematical model is assumed to be unknown, under the assumption of all the state is available for measurement. The learning algorithm for the RHONN is implemented using a novel learning algorithm. The respective stability analysis, on the basis of the Lyapunov approach, for the whole scheme, is included. The applicability of these schemes is illustrated by simulations for an electric three phase induction motor.

## 2. Mathematical preliminaries

In this section, important mathematical preliminaries, required in future sections, are presented. This section close follows [3].

### 2.1. Stability definitions

Through this paper, we use $k$ as the sampling step, $k \in 0 \cup \mathbb{Z}^+$, $|\bullet|$ as the absolute value and, $\|\bullet\|$ as the Euclidian norm for vectors and as any adequate norm for matrices. Consider a MIMO nonlinear system:

$$\chi(k+1) = F(\chi(k), u(k)) \tag{1}$$

$$y(k) = h(\chi(k)) \tag{2}$$

where $\chi \in \Re^n$, $u \in \Re^m$, and $F \in \Re^n \times \Re^m \to \Re^n$ is nonlinear function.

**Definition 1.** The system (1) is said to be forced, or to have inputs. In contrast the system described by an equation without explicit presence of an input $u$, that is,

$$\chi(k+1) = F(\chi(k))$$

is said to be unforced. It can be obtained after selecting the input $u$ as a feedback function of the state

$$u(k) = \xi(\chi(k)) \tag{3}$$

Such substitution eliminates $u$:

$$\chi(k+1) = F(\chi(k), \xi(\chi(k))) \tag{4}$$

and yields an unforced system (4) [7].

**Definition 2.** The solution of Eqs. (1)–(3) is semiglobally uniformly ultimately bounded (SGUUB), if for any $\Omega$, a compact subset of $\Re^n$ and all $\chi(k_0) \in \Omega$, there exists an $\varepsilon > 0$ and a number $N(\varepsilon, \chi(k_0))$ such that $\|\chi(k)\| < \varepsilon$ for all $k \geq k_0 + N$.

In other words, the solution of Eq. (1) is said to be SGUUB if, for any a priori given (arbitrarily large) bounded set $\Omega$ and any a priori given (arbitrarily small) set $\Omega_0$, which contains (0,0) as an interior point, there exists a control (3) such that every trajectory of the closed loop system starting from $\Omega$ enters the set $\Omega_0 = \{\chi(k) | \|\chi(k)\| < \varepsilon\}$, in a finite time and remains in it thereafter.

**Theorem 1.** *Let $V(\chi(k))$ be a Lyapunov function for a discrete-time system* (1), *which satisfies the following properties*:

$$\gamma_1(\|\chi(k)\|) \leq V(\chi(k)) \leq \gamma_2(\|\chi(k)\|)$$

$$V(\chi(k+1)) - V(\chi(k)) = \Delta V(\chi(k)) \leq -\gamma_3(\|\chi(k)\|) + \gamma_3(\zeta)$$

*where $\zeta$ is a positive constant, $\gamma_1(\bullet)$ and $\gamma_2(\bullet)$ are strictly increasing functions, and $\gamma_3(\bullet)$ is a continuous, nondecreasing function. Thus if*

$$\Delta V(\chi) < 0 \quad \text{for } \|\chi(k)\| > \zeta$$

*then $\chi(k)$ is uniformly ultimately bounded, i.e. there is a time instant $k_T$, such that $\|\chi(k)\| < \zeta, \forall\ k < k_T$.*

**Definition 3.** A subset $S \in \Re^n$ is bounded if there exists $r>0$ such that $\|\chi\| \leq r$ for all $\chi \in S$ [7].

### 2.2. Discrete-time high order neural networks

The use of multilayer neural networks is well known for pattern recognition and for modelling of static systems. The NN is trained to learn an input–output map. Theoretical works have proven that, even with just one hidden layer, a NN can uniformly approximate any continuous function over a compact domain, provided that the NN has a sufficient number of synaptic connections.

For control tasks, extensions of the first order Hopfield model called recurrent high order neural networks (RHONN), which present more interactions among the neurons, are proposed in [10,13]. Additionally, the RHONN model is very flexible and allows to incorporate to the neural model a priori information about the system structure.

Consider the following discrete-time recurrent high order neural network (RHONN):

$$x_i(k+1) = w_i^\top z_i(x(k),u(k)), \quad i = 1,\ldots,n \tag{5}$$

where $x_i$ $(i = 1,2,\ldots,n)$ is the state of the $i$-th neuron, $n$ is the state dimension, $w_i$ $(i = 1,2,\ldots,n)$ is the respective on-line adapted weight vector, and $z_i(x(k), u(k))$ is given by

$$z_i(x(k),u(k)) = \begin{bmatrix} z_{i_1} \\ z_{i_2} \\ \vdots \\ z_{i_{L_i}} \end{bmatrix} = \begin{bmatrix} \Pi_{j\in I_1} \xi_{i_j}^{di_j(1)} \\ \Pi_{j\in I_2} \xi_{i_j}^{d_{i_j}(2)} \\ \vdots \\ \Pi_{j\in I_{L_i}} \xi_{i_j}^{di_j(L_i)} \end{bmatrix} \tag{6}$$

with $L_i$ is the respective number of high-order connections, $\{I_1, I_2, \ldots, I_{L_i}\}$ is a collection of non-ordered subsets of $\{1,2,\ldots,n+m\}$, $m$ is the number of external inputs, $d_{j_i}(k)$ being nonnegative integers, and $\xi_i$ defined as follows:

$$\xi_i = \begin{bmatrix} \xi_{i_1} \\ \vdots \\ \xi_{i_1} \\ \xi_{i_{n+1}} \\ \vdots \\ \xi_{i_{n+m}} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix} \tag{7}$$

In Eq. (7), $u = [u_1, u_2, \ldots, u_m]^\top$ is the input vector to the neural network, and $S(\bullet)$ is defined by

$$S(\varsigma) = \frac{1}{1+\exp(-\beta\varsigma)}, \quad \beta > 0 \tag{8}$$

where $\varsigma$ is any real value variable.

From Eq. (5) three possible models can be derived:

- Parallel model

$$x_i(k+1) = w_i^\top z_i(x(k),u(k)), \quad i = 1,\ldots,n \tag{9}$$

- Series–parallel model

$$x_i(k+1) = w_i^\top z_i(\chi(k),u(k)), \quad i = 1,\ldots,n \tag{10}$$

- Feedforward model (HONN)

$$x_i(k) = w_i^\top z_i(u(k)), \quad i = 1,\ldots,n$$

where $x$ is the NN state vector, $\chi$ is the plant state vector and $u$ is the input vector to the NN.

Consider the problem to approximate the general discrete-time nonlinear system (1), by the discrete-time RHONN series–parallel representation depicted in Fig. 1, which mathematical model is given by [13]:

$$\chi_i(k+1) = w_i^{*\top} z_i(\chi(k),u(k)) + \varepsilon_{z_i}, \quad i = 1,\ldots,n \tag{11}$$

where $\chi_i$ is the $i$-th plant state,$\varepsilon_{z_i}$ is a bounded approximation error, which can be reduced by increasing the number of the adjustable weights [13].

**Assumption 1.** Assume that there exists ideal weights vector $w_i$* such that $\|\varepsilon_{z_i}\|$ can be minimized on a compact set $\Omega_{z_i} \subset \Re^{L_i}$. The ideal weight vector $w_i$* is an artificial quantity required for analytical purpose [13]. In general, it is assumed that this vector exists and is constant but unknown.

Let us define its estimate as $w_i$ and the estimation error as

$$\tilde{w}_i(k) = w_i^* - w_i(k) \tag{12}$$

The estimate $w_i$ is used for stability analysis, which will be discussed later. Since $w_i$* is constant, then $\tilde{w}_i(k+1) - \tilde{w}_i(k) = w_i(k+1) - w_i(k)$, $\forall k \in 0 \cup \mathbb{Z}^+$.

## 3. Neural identification

In this section, we consider the problem to identify the nonlinear system

$$\chi(k+1) = F(\chi(k),u(k)) + d(k) \tag{13}$$

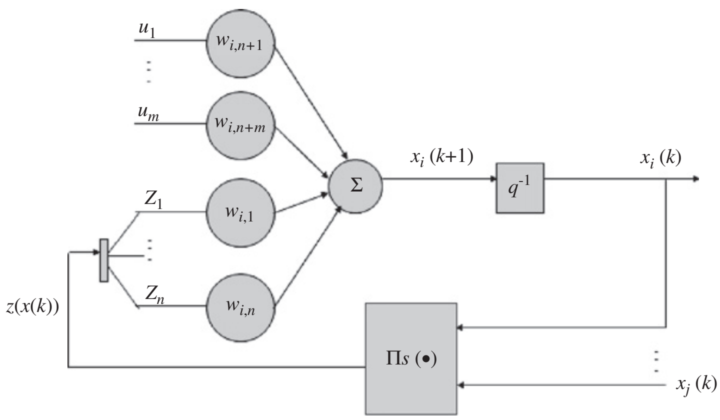



Fig. 1. RHONN structure.

where $\chi \in \Re^n$, $u \in \Re^m$, $d \in \Re^n$ is the disturbance vector and $F \in \Re^n \times \Re^m \rightarrow \Re^n$ is nonlinear function. To identify the system (13), we use a RHONN defined as

$$x_i(k+1) = w_i^\top z_i(\chi(k),u(k)), \quad i = 1,\ldots,n \tag{14}$$

as defined in Eq. (5).

Consider the problem to approximate the general discrete-time nonlinear system (13), by the following discrete-time RHONN series–parallel representation [13]:

$$\chi_i(k+1) = w_i^{*\top} z_i(\chi(k),u(k)) + \varepsilon_{z_i}, \quad i = 1,\ldots,n \tag{15}$$

as explained in Eq. (11) with the estimation error (12) under Assumption 1.

Now, we establish the first main result of this paper in the following theorem.

**Theorem 2.** *The RHONN* (14) *trained with a weights updating laws selected as*

$$w_i(k+1) = \eta_i w_i(k) + K_i(k) z_i(\chi(k),u(k)) e(k) \tag{16}$$

*with $K_i$ as an adaptive learning rate for the i-th state of the RHONN and $\eta_i$ as its design parameter, used to identify the nonlinear plant* (13), *ensures that the identification error*

$$e_i(k) = \chi_i(k) - x_i(k) \tag{17}$$

*is semiglobally uniformly ultimately bounded* (*SGUUB*); *moreover, the RHONN weights remain bounded besides the dynamics of the identification error* (17) *can be expressed as*

$$e_i(k+1) = \tilde{w}_i(k) z_i(\chi(k),u(k)) + \varepsilon_{z_i} \tag{18}$$

*and the dynamics of* Eq. (12) *can be written as*

$$\tilde{w}_i(k+1) = w_i^* - \eta_i w_i(k) - K_i(k) z_i(\chi(k),u(k)) e(k) \tag{19}$$

**Proof.** Consider the Lyapunov function candidate

$$V_i(k) = \eta_i \tilde{w}_i^T(k) \tilde{w}_i(k) + \eta_i e_i^2(k) \tag{20}$$

$$\begin{aligned} \Delta V_i(k) &= V(k+1) - V(k) \\ &= \eta_i \tilde{w}_i^T(k+1) \tilde{w}_i(k+1) + \eta_i e_i^2(k+1) - \eta_i \tilde{w}_i^T(k) \tilde{w}_i(k) - \eta_i e_i^2(k) \end{aligned}$$

Using Eqs. (18) and (19) in Eq. (20)

$$\begin{aligned} \Delta V_i(k) = {} & \eta_i (\tilde{w}_i(k) z_i(\chi(k),u(k)) + \varepsilon_{z_i})^2 + \eta_i [w_i{*} - \eta_i w_i(k) - K_i(k) z_i(\chi(k),u(k)) e(k)]^T \\ & \times [w_i{*} - \eta_i w_i(k) - K_i(k) z_i(\chi(k),u(k)) e(k)] - \eta_i \tilde{w}_i(k) \tilde{w}_i(k) - \eta_i e_i^2(k) \end{aligned}$$

$$\begin{aligned} \Delta V_i(k) = {} & \eta_i (\tilde{w}_i(k) z_i(\chi(k),u(k)))^2 + 2\eta_i \tilde{w}_i(k) z_i(\chi(k),u(k)) \varepsilon_{z_i} + \eta_i \varepsilon_{z_i}^2 \\ & + \eta_i w_i^{*^T} w_i{*} - \eta_i^2 w_i^{*^T} w_i(k) - \eta_i w_i^{*^T} K_i(k) z_i(\chi(k),u(k)) e(k) - \eta_i^2 w_i^T(k) w_i{*} \\ & + \eta_i^3 w_i^T(k) w_i(k) + \eta_i^2 w_i^T(k) K_i(k) z_i(\chi(k),u(k)) e(k) \\ & - \eta_i e(k) z_i^T(\chi(k),u(k)) K_i^T(k) w_i{*} + \eta_i^2 e(k) z_i^T(\chi(k),u(k)) K_i^T(k) w_i(k) \\ & + \eta_i e^2(k) z_i^T(\chi(k),u(k)) K_i^T(k) K_i(k) z_i(\chi(k),u(k)) - \eta_i \tilde{w}_i^T(k) \tilde{w}_i(k) - \eta_i e_i^2(k) \end{aligned} \tag{21}$$

Using Assumption 1 and the inequalities

$$X^TX + Y^TY \geq 2X^TY$$

$$X^TX + Y^TY \geq -2X^TY$$

which are valid $\forall X, Y \in \Re^n$, and used in positive quadratic terms, then Eq. (21) can be rewritten as

$$\begin{aligned}\Delta V_i(k) \leq & 2\eta_i(\tilde{w}_i(k)z_i(\chi(k),u(k)))^2 + 2\eta_i\varepsilon_{z_i}^2 + 3\eta_i w_i^{*^T} w_i{*} + 3\eta_i^3 w_i^T(k)w_i(k) \\ & +3\eta_i e_i^2(k)z_i^T(\chi(k),u(k))K_i^T(k)K_i(k)z_i(\chi(k),u(k)) - \eta_i\tilde{w}_i^T(k)\tilde{w}_i(k) - \eta_i e_i^2(k) \\ \leq & 2\eta_i\|\tilde{w}_i(k)\|^2\|z_i(\chi(k),u(k))\|^2 + 2\eta_i|\varepsilon_{z_i}|^2 + 3\eta_i\|w_i{*}\|^2 + 3\eta_i^3\|w_i(k)\|^2 \\ & +3\eta_i|e_i(k)|^2|K_i(k)z_i(\chi(k),u(k))|^2 - \eta_i\|\tilde{w}_i(k)\|^2 - \eta_i|e_i(k)|^2 \end{aligned} \quad (22)$$

Defining

$$A_i(k) = \eta_i - 3\eta_i|K_i(k)z_i(\chi(k),u(k))|^2$$

$$B_i(k) = \eta_i - 2\eta_i\|z_i(\chi(k),u(k))\|^2$$

$$C_i(k) = 2\eta_i|\varepsilon_{z_i}|^2 + 3\eta_i\|w_i^*\|^2 + 3\eta_i^3\|w_i(k)\|^2$$

then, there exists $\eta_i$, such that $A_i > 0$, $B_i > 0$ and $C_i > 0$, $\forall k$, then, Eq. (22) can be expressed as

$$\Delta V_i(k) \leq -|e_i(k)|^2(k)A_i(k) - \|\tilde{w}_i(k)\|^2 B_i(k) + C_i(k)$$

Hence $\Delta V_i(k) < 0$ when

$$\|\tilde{w}_i(k)\| > \sqrt{\frac{C_i(k)}{A_i(k)}} \equiv \kappa_1 \quad (23)$$

or

$$|e_i(k)| > \sqrt{\frac{C_i(k)}{B_i(k)}} \equiv \kappa_2 \quad (24)$$

Therefore, according to Theorem 1, the solution of Eqs. (18) and (19) are SGUUB along Eqs. (23) or (24) [3]. □

It is important to remark that the proposed learning algorithm for the RHONN is simpler than the based on an extended Kalman filter [1,16].

## 4. Induction motor applications

In this section, we apply the above developed schemes to a three-phase induction motor, which is one of the most used actuators for industrial applications due to its reliability, ruggedness and relatively low cost. The modelling of an induction motor is challenging, since its dynamics is described by multivariable, coupled, and highly nonlinear system [9,15]. Early works on control of induction motors were focused on the field oriented control (FOC) [8], exact input–output linearization, adaptive input output linearization, direct torque control (DTC) ([8] and references therein). However, most of those works were developed stabilized controllers for continuous-time model of the motor. In [9], a discrete-time model is proposed, as well as a control algorithm, assuming that the parameters and load torque of the motor model are known. Moreover all these controllers are designed based on the physical model of the motor, which result in sensitive control with respect to plant parameters variations. To this end, we consider the control problem assuming that some of the plant parameters as well as external disturbances (load torque) are unknown.

### 4.1. Motor model

The sixth-order discrete-time induction motor model in the stator fixed reference frame $(\alpha,\beta)$, under the assumptions of equal mutual inductances and linear magnetic circuit, is given by [9]

$$
\begin{aligned}
\omega(k+1) &= \omega(k) + \frac{\mu}{\alpha}(1-\alpha)M(i^{\beta}(k)\psi^{\alpha}(k) - i^{\alpha}(k)\psi^{\beta}(k)) - \left(\frac{T}{J}\right)T_L(k)\\
\psi^{\alpha}(k+1) &= \cos(n_p\theta(k+1))\rho_1(k) - \sin(n_p\theta(k+1))\rho_2(k)\\
\psi^{\beta}(k+1) &= \sin(n_p\theta(k+1))\rho_1(k) + \cos(n_p\theta(k+1))\rho_2(k)\\
i^{\alpha}(k+1) &= \varphi^{\alpha}(k) + \frac{T}{\sigma}u^{\alpha}(k) + d_1(k)\\
i^{\beta}(k+1) &= \varphi^{\beta}(k) + \frac{T}{\sigma}u^{\beta}(k) + d_2(k)\\
\theta(k+1) &= \theta(k) + \omega(k)T + \frac{\mu}{\alpha}\left[T - \frac{(1-a)}{\alpha}\right]M(i^{\beta}(k)\psi^{\alpha}(k) - i^{\alpha}(k)\psi^{\beta}(k)) - \frac{T_L(k)}{J}T^2
\end{aligned}
\tag{25}
$$

with

$$
\begin{aligned}
\rho_1(k) &= a(\cos(\phi(k))\psi^{\alpha}(k) + \sin(\phi(k))\psi^{\beta}(k)) + b(\cos(\phi(k))i^{\alpha}(k) + \sin(\phi(k))i^{\beta}(k))\\
\rho_2(k) &= a(\cos(\phi(k))\psi^{\alpha}(k) - \sin(\phi(k))\psi^{\beta}(k)) + b(\cos(\phi(k))i^{\alpha}(k) - \sin(\phi(k))i^{\beta}(k))\\
\varphi^{\alpha}(k) &= i^{\alpha}(k) + \alpha\beta T\psi^{\alpha}(k) + n_p\beta T\omega(k)\psi^{\alpha}(k) - \gamma T i^{\alpha}(k)\\
\varphi^{\beta}(k) &= i^{\beta}(k) + \alpha\beta T\psi^{\beta}(k) + n_p\beta T\omega(k)\psi^{\beta}(k) - \gamma T i^{\beta}(k)\\
\phi(k) &= n_p\theta(k)
\end{aligned}
$$

with $b=(1-a)M, \alpha=R_r/L_r, \gamma=M^2R_r/\sigma L_r^2+R_s/\sigma, \sigma=L_s-M^2/L_r, \beta=M/\sigma L_r, a=e^{-\alpha T}, \mu=Mn_p/JL_r$ where $Ls$, $L_r$ and $M$ are the stator, rotor and mutual inductance, respectively; $R_s$ and $R_r$ are the stator and rotor resistances, respectively; $n_p$ is the number of pole pairs; $i^{\alpha}$ and $i^{\beta}$ represents the currents in the $\alpha$ and $\beta$ phases, respectively; $\psi^{\alpha}$ and $\psi^{\beta}$ represents the fluxes in the $\alpha$ and $\beta$ phases, respectively, and $\theta$ is the rotor angular displacement. Due to space limitations the windings diagram for the induction motor is not included in this paper, however, it can be found in [8].

## 4.2. Neural identifier design

To this end, we apply the RHONN (Fig. 2), developed in Section 3, to identify the above mentioned three-phase induction motor (25). To identify that system, we use the RHONN (15) with $n=5$ trained with Eq. (16). The RHONN proposed for this application is as follows:

$$x_1(k+1) = w_{11}(k)S(\omega(k)) + w_{12}(k)S(\omega(k))S(\psi^{\beta}(k))i^{\alpha}(k) + w_{13}(k)S(\omega(k))S(\psi^{\alpha}(k))i^{\beta}(k)$$

$$x_2(k+1) = w_{21}(k)S(\omega(k))S(\psi^{\beta}(k)) + w_{22}(k)i^{\beta}(k)$$

$$x_3(k+1) = w_{31}(k)S(\omega(k))S(\psi^{\alpha}(k)) + w_{32}(k)i^{\alpha}(k)$$

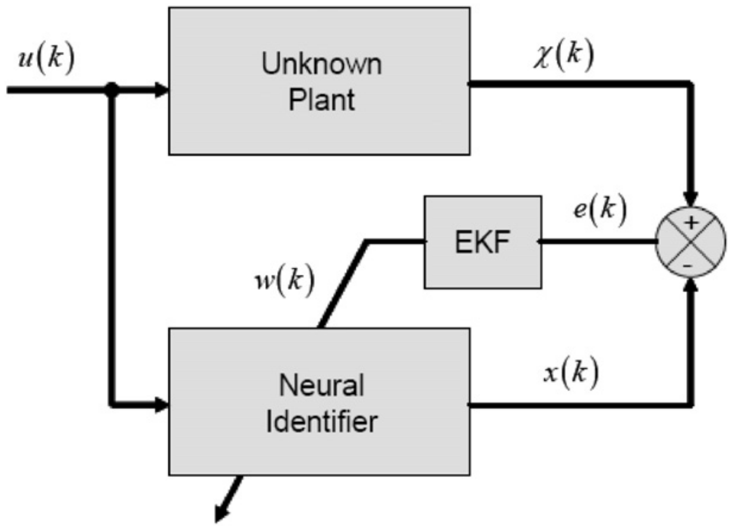



Fig. 2. Neural identifier scheme.

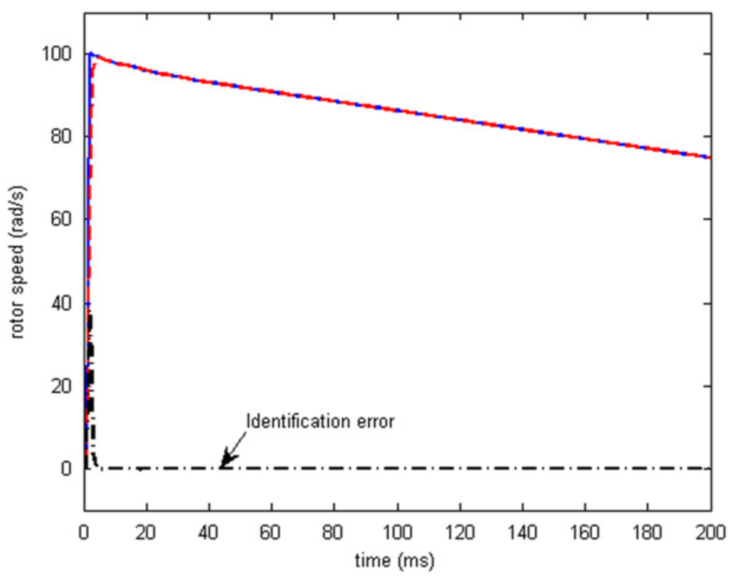



Fig. 3. Rotor speed identification (plant signal in solid line and neural signal in dashed line).

$$x_4(k+1) = w_{41}(k)S(\psi^\alpha(k)) + w_{42}(k)S(\psi^\beta(k)) + w_{43}(k)S(i^\alpha(k)) + w_{44}(k)u^\alpha(k)$$

$$x_5(k+1) = w_{51}(k)S(\psi^\alpha(k)) + w_{52}(k)S(\psi^\beta(k)) + w_{53}(k)S(i^\beta(k)) + w_{54}(k)u^\beta(k) \quad (26)$$

where $x_1$ identify the angular speed $\omega$; $x_2$ and $x_3$ identify the fluxes $\psi^\alpha$ and $\psi^\beta$, respectively; finally $x_4$ and $x_5$ identify the currents $i^\alpha$ and $i^\beta$, respectively. The inputs $u^\alpha$ and $u^\beta$ are selected as chirp functions, in order to modelling most of the plant dynamics.

The training is performed on-line, using a series–parallel configuration. All the NN states are initialized randomly as well as the weights vectors. It is important to remark that the initial conditions of the plant are completely different from the initial conditions for the NN.

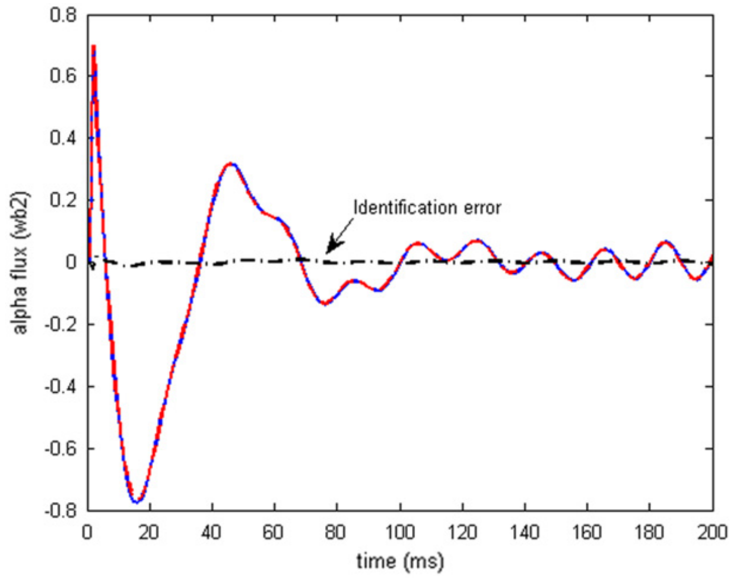



Fig. 4. Alpha flux identification (plant signal in solid line and neural signal in dashed line).

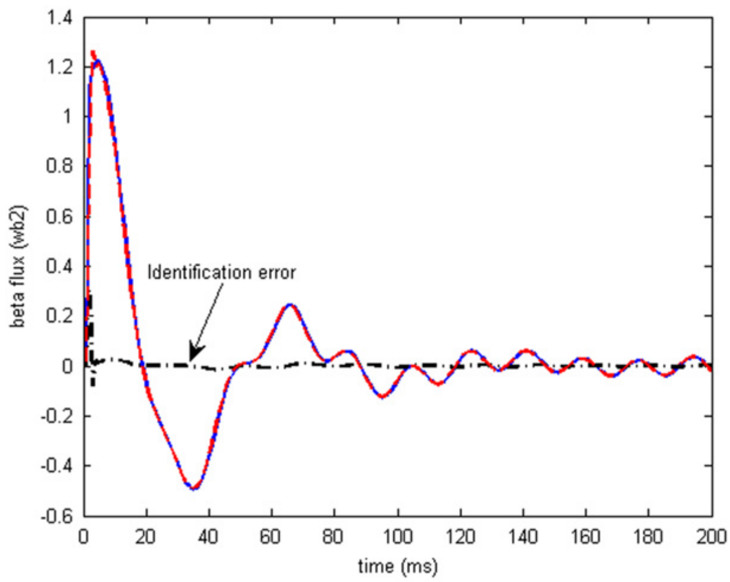



Fig. 5. Beta flux identification (plant signal in solid line and neural signal in dashed line).

### 4.3. Simulation results

The results of the simulation are presented in Figs. 3–7. Fig. 3 shows the identification performance for the speed rotor; Figs. 4 and 5 present the identification performance for the fluxes in phases $\alpha$ and $\beta$, respectively. Finally, Figs. 6 and 7 portray the identification performance for currents in phases $\alpha$ and $\beta$, respectively.

It is important to consider that for the proposed identification scheme it is necessary to have complete access to the plan state. However, in the proposed scheme the plant model it is assumed to be unknown and working under the presence of external disturbances and parametric variations, both of them are considered to be unknown [13]. Then the proposed

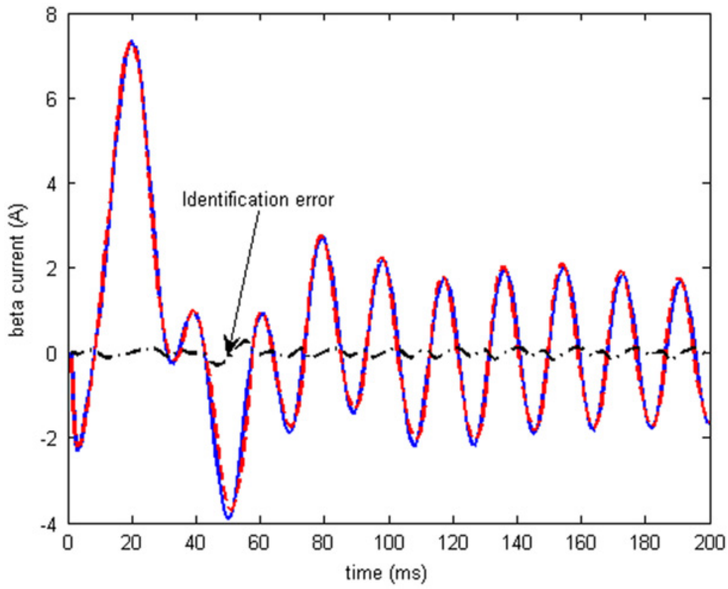



Fig. 6. Alpha current identification (plant signal in solid line and neural signal in dashed line).

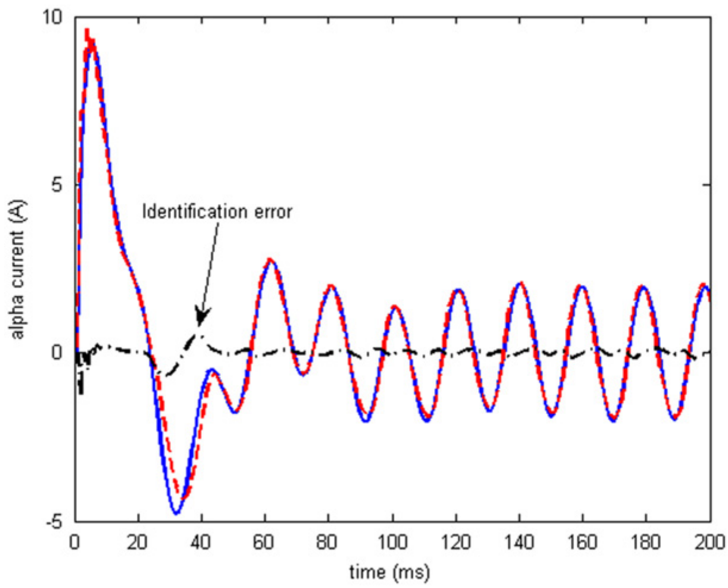



Fig. 7. Beta current identification (plant signal in solid line and neural signal in dashed line).

neural identifier provides a mathematical model of the unknown plant which can be used for control, prediction or simulation purposes.

## 5. Conclusions

A RHONN structure is used to design a neural identifier for a class of MIMO discrete-time nonlinear system. The RHONN identifier is trained with a novel algorithm, which is implemented on-line in a series parallel configuration. The boundedness of the state and weight estimation errors is established on the basis of the Lyapunov approach. The RHONN training algorithm presents good performance. Simulation results show the effectiveness of the proposed schemes, as applied to an electric three-phase squirrel cage induction motor. This paper deals only with identification for uncertain discrete-time nonlinear systems without any restriction in the weights evolution as the zero crossing problem, however, such approach is considered as a future work.

## Acknowledgement

The authors thank the support of CONACYT Mexico, through Projects 57801Y and 103191Y. The authors also thank the useful comments of the anonymous reviewers, which help to improve this paper.

## References

[1] A.Y. Alanis, E.N. Sanchez, A.G. Loukianov, Discrete time adaptive backstepping nonlinear control via high order neural networks, IEEE Transactions on Neural Networks 18 (4) (2007) 1185–1195.
[2] J.A. Farrell, M.M. Polycarpou, Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches, John Wiley and Sons, NY, USA, 2006.
[3] S.S. Ge, J. Zhang, T.H. Lee, Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time, IEEE Transactions on Systems, Man and Cybernetics, Part B 34 (4) (2004).
[4] J. Ghosh, Y. Shin, Efficient high-order neural networks for classification and function approximation, International Journal of Neural Systems 3 (4) (1992) 323–350.
[5] R. Ghazali, A. Hussain, M. Merabti, Higher order neural networks and their applications to financial time series prediction, in: Proceedings of the IASTED Artificial Intelligence and Soft Computing, Palma De Mallorca, Spain, August 2006.
[6] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice-Hall, Upper Saddle River, NJ, USA, 1999.
[7] H. Khalil, Nonlinear Systems, second ed., Prentice-Hall, Upper Saddle River, NJ, USA, 1996.
[8] F. Khorrami, P. Krishnamurthy, H. Melkote, Modeling and Adaptive Nonlinear Control of Electric Motors, Springer-Verlag, Berlin, Germany, 2003.
[9] A.G. Loukianov, J. Rivera, J.M. Cañedo, Discrete-time sliding mode control of an induction motor, in: Proceedings of the IFAC'02, Barcelona, Spain, July 2002.
[10] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks 1 (March) (1990) 4–27.
[11] M. Norgaard, O. Ravn, N.K. Poulsen, L.K. Hansen, Neural Networks for Modelling and Control of Dynamic Systems, Springer Verlag, New York, USA, 2000.
[12] A.S. Poznyak, E.N. Sanchez, W. Yu, Differential Neural Networks for Robust Nonlinear Control, World Scientific, Singapore, 2001.
[13] G.A. Rovithakis, M.A. Chistodoulou, Adaptive Control with Recurrent High-Order Neural Networks, Springer Verlag, Berlin, Germany, 2000.

[14] E.N. Sanchez, L.J. Ricalde, Trajectory tracking via adaptive recurrent neural control with input saturation, in: Proceedings of the International Joint Conference on Neural Networks'03, Portland, OR, USA, July 2003.

[15] V. Utkin, J. Guldner, J. Shi, Sliding Mode Control in Electromechanical Systems, Taylor and Francis, Philadelphia, USA, 1999.

[16] W. Yu, X. Li, Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms, Information Sciences 158 (2004) 131–147.