Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Discrete-Time Neural Block Control Using Sliding Modes for Induction Motors with Gears

M.Elena Antonio-Toledo * **Patricia Ramirez** *
Edgar N. Sanchez * **Alexander G. Loukianov** *

* *CINVESTAV Unidad Guadalajara, Av. Bosque 1145, CP 45019 Zapopan,Jalisco e-mail: {mantonio,pramirez,sanchez,louk}@gdl.cinvestav.mx*

Abstract: This paper proposes a control scheme based on discrete-time block control technique using sliding modes, for a system composed of a three-phase rotatory induction motor when includes a gear.The goal is tracking position trajectory. A recurrent high order neural network (RHONN) is used to identify the system, which is trained with an Extended Kalman Filter (EKF) algorithm. Its performance is illustrated via simulations.

Keywords: Induction motor, backlash, extend Kalman filter (EKF), neural block control,slide modes.

1. INTRODUCTION

Induction motors are one of the most important workhorses in industry and they are manufactured in large numbers. For a relatively long period, induction motors have mainly been deployed in constant-speed motor drives for general purpose application due to their reliability, simple construction, and relatively low cost (Menghal and Laxmi, 2013) (Mohamadian et al., 2003).
Gears are mechanical elements which, when attached, can change the behaviour of a system. For motors, gears enable the change of the movement's speed, torque and direction. Despite the many advantages of mechanical gears, there are features when must be taken into account when implementing such elements. Due to the continuous use, natural weather or design failures, these components can have or develop dead zone, hysteresis or backlash behaviours. These non-linearities are not easy to predict and are difficult to compensate when they become present in systems (Woods, 1944).
Backlash is present when there is a gap between the gear's teeth. When gears change direction, the transference between the gears becomes zero until the gap is closed; therefore the final gear's position is constant even if the motor shaft is in motion (Menghal and Laxmi, 2013).
For many non-linear systems, it is often difficult to obtain their accurate and faithful mathematical models,regarding their physically complex structures and hidden parameters. Therefore, system identification becomes a relevant issue and even necessary before system control can be considered, not only for understanding and predicting the behaviour of the system but also to obtain an effective control law. The identification problem consists on the selection of an appropriate identification model and adjusting its parameters according to an adaptive law, such that the response of the model to an input signal (or class of input signals) approximates the response of the real system to the same input (Alanis et al., 2010).
The best-known training approach for recurrent neural networks (RNN) is the back propagation through time learning. However, it is merely a first-order gradient descent method and hence its learning speed is very slow. Extended Kalman Filter (EKF) based algorithms have been introduced to the training of neural networks. With an EKF-based algorithm, the learning convergence can be improved. Over the past decade, the EKF-based training of neural networks, both feed-forward and recurrent ones, has proven to be reliable and practical for many applications (Alanis et al., 2007).
In this paper, we propose a scheme for position tracking based on the discrete-time block control technique with sliding mode, using a neural identifier based on RHONN for a class of multi input multi output (MIMO) discrete-time non-linear systems. The recurrent high-order neural network (RHONN), which estimates the state vectors of the unknown plant dynamics. The training algorithm for RHONN is based on an EKF.

2. FUNDAMENTALS

2.1 Discrete-time high order neural netwokrs

Consider a MIMO nonlinear system:

$$\begin{aligned} \chi(k+1) &= F(\chi(k), u(k)) \\ y(k) &= h(k) \end{aligned} \tag{1}$$

where states $\chi \in \Re^n$, control values $u \in \Re^m$, for some n and m, each time instant $k \in \mathbb{Z}^+$, $F \in \Re^n \times \Re^m \to \Re^n$ is a nonlinear function.

For control tasks, extensions of the first order Hopfield model, called Recurrent High Order Neural Networks (RHONN), present more interactions among the neurons.

⋆ This work is supported by CONACYT Mexico project 131678.

Additionally, the RHONN model is very flexible and allows to incorporate to the neural model a priory information about the system structure (Alanis et al., 2007). Consider the following discrete-time RHONN:

$$\hat{x}_i(k+1) = w_i^T z_i(\hat{x}(k), v(k)) \quad i = 1, ..., n \tag{2}$$

where $\hat{x}_i (i = 1, 2, ..., n)$ is the state of the i-th neuron, $w_i (i = 1, 2, ..., n)$ is the respective on-line adapted weight vector, and $z_i(\hat{x}(k), v(k))$ is given by

$$z_i(x(k), \varrho(k)) = \begin{bmatrix} z_{i_1} \\ z_{i_2} \\ \vdots \\ z_{i_{L_i}} \end{bmatrix} = \begin{bmatrix} \prod_{j \epsilon I_1} \xi_{ij}^{dij(1)} \\ \prod_{j \epsilon I_2} \xi_{ij}^{dij(2)} \\ \vdots \\ \prod_{j \epsilon I_{L_i}} \xi_{ij}^{dij(L_i)} \end{bmatrix} \tag{3}$$

where L_i is the respective number of high order connections, $\{I_1, I_2, ...I_{Li}\}$ is a collection of non-ordered subsets of $\{1, 2, ..., n + m\}$, n is the state dimension, m is the number of external inputs, $d_{ij}(k)$ being a non-negative integers, and ξ_i defined as follows:

$$\xi_i = \begin{bmatrix} \xi_{i_1} \\ \vdots \\ \xi_{i_n} \\ \xi_{i_{n+1}} \\ \vdots \\ \xi_{i_{n+m}} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ \varrho_1 \\ \vdots \\ \varrho_m \end{bmatrix} \tag{4}$$

in (4), $\varrho = [\varrho_1, \varrho_2, \ldots, \varrho_m]^T$ is the input vector to the neural network, and $S(\bullet)$ is defined by

$$S(\varsigma) = \frac{1}{1 + exp(-\beta\varsigma)} \ , \ \beta > 0 \tag{5}$$

where ς is any real value variable.
Consider the problem to approximation the general discrete-time non-linear system (1), by the following discrete-time RHONN series-parallel representation:

$$x_i(k+1) = w_i^{*T} z_i(x(k), \varrho(k)) + \epsilon_{z_i}, \ i = 1, \ldots, n \tag{6}$$

where x_i is the i-th plant state, ϵ_{z_i} is a bounded approximation error, which can be reduced by increasing the number of the adjustable weights. Assume that there exists and ideal weights vector w_i^* such that $\|\epsilon_{z_i}\|$ 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. In general, it is assumed that this vector exists and is a constant but unknown. let us define its estimate as w_i and the estimation error as

$$\tilde{w}_i(k) = w_i^*(k) - w_i \tag{7}$$

The estimate w_i is used for stability analysis. Since w_i^* is a constant, then $\tilde{w}_i(k+1) - \tilde{w}_i(k) = w_i(k) - w_i(k+1)$, $\forall k \in 0 \cup \mathbb{Z}^+$ (Sanchez et al., 2008).

2.2 The EKF Training algorithm

The well known Kalman filter is a set of mathematical equations which provides an efficient computational (recursive) solution of the least-square method ; this filter estimates the state of a linear system with additive state and output white noises (Sanchez et al., 2009).
For KF-based neural network training, the network weights become the states to be estimated. In this case, the error between the neural network output and the measured plant output can be considered as additive white noise. Due to the fact that the neural network mapping is non-linear, an EKF, type is required. The training goal is to find the optimal weight values which minimize the prediction errors.
For this paper, we use a modified EKF-based training algorithm described by:

$$\begin{aligned} w_i(k+1) &= w_i(k) + \eta_i K_i(k)[y(k) - \hat{y}(k)] \\ K_i(k) &= \begin{cases} P_i(k)H_i(k)M_i(k) & if \ \ \|w_i(k)\| > c_i \\ 0 & if \ \ \|w_i(k)\| < c_i \end{cases} \\ P_i(k+1) &= P_i(k) - K_i(k)H_i^T(k)P_i(k) + Q_i(k) \end{aligned} \tag{8}$$

with

$$M_i = [R_i(k) + H_i^T(k)P_i(k)H_i(k)]^{-1} \tag{9}$$

where $c_i > 0$ is a constant used to avoid zero-crossing and $P_i \in \Re^{L_i x L_i}$ is the prediction error associated covariance matrix, $w_i \in \Re^{L_i}$ is the weight (state) vector, L_i is the total number of neural network weights, $y \in \Re^m$ is the measured output vector, $\hat{y} \in \Re^m$ is the network output, η_i is a design parameter, $K_i \in \Re^{L_i x m}$ is the Kalman gain matrix, $Q_i \in \Re^{L_i x L_i}$ is the state noise associated covariance matrix, $R_i \in \Re^{mxm}$ is the measurement noise associated covariance matrix, $H_i \in \Re^{L_i x m}$ is a matrix, in which each entry (H_{ij}) is the derivative of one of the neural network output, $(\tilde{y})$, with respect to one neural network weight, (w_{ij}), as follows:

$$H_{ij}(k) = \left[\frac{\partial \tilde{y}(k)}{\partial w_{ij}(k)} \right]_{w_i(k) = \hat{w}_i(k+1)} \tag{10}$$

where $i = 1, \ldots, n$ *and* $j = 1, \ldots, L_i$. Usually P_i, Q_i and R_i are initialized as diagonal matrices, with entries $P_i(0)$, $Q_i(0)$ and $R_i(0)$, respectively. It is important to note that $H_i(k)$, $K_i(k)$ and $P_i(k)$ for the EKF are bounded (for a detailed explanation, see (Sanchez et al., 2008).

3. MATHEMATICAL MODEL

3.1 Induction motor model

The six-order discrete-time induction motor model in the stator fixed reference frame (α, β), under the assumptions of equal mutual inductances and linear magnetic circuit, is given by

$$\begin{aligned} \theta(k+1) &= \theta(k) + \omega(k)T \\ &+ \frac{\mu}{\alpha}[T - \frac{1}{\alpha}(1-a)]M(i^\beta(k)\psi^\alpha(k) - i^\alpha(k)\psi^\beta(k)) \\ &- \frac{T_L(k)}{2J}T^2 \\ \omega(k+1) &= \omega(k) \\ &+ \frac{\mu}{\alpha}(1-a)M(i^\beta(k)\psi^\alpha(k) - i^\alpha(k)\psi^\beta(k)) - (\frac{T}{J})T_L(k) \end{aligned} \tag{11}$$

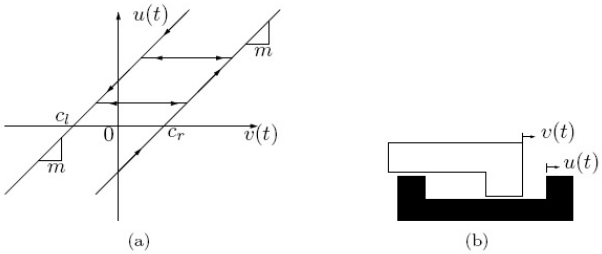

Fig. 1. a) Backlash model; b)Schematic representation.

$$\psi^{\alpha}(k+1) = cos(\eta_p\theta(k+1))\rho_1 - sin(\eta_p\theta(k+1))\rho_2$$

$$\psi^{\beta}(k+1) = sin(\eta_p\theta(k+1))\rho_1 + cos(\eta_p\theta(k+1))\rho_2$$

$$i^{\alpha}(k+1) = \varphi^{\alpha}(k) + \frac{T}{\sigma}u^{\alpha}(k)$$

$$i^{\beta}(k+1) = \varphi^{\beta}(k) + \frac{T}{\sigma}u^{\beta}(k)$$

where

$$\begin{array}{l}\rho_1(k) = a(cos(\phi(k))\psi^{\alpha}(k) + sin(\phi(k))\psi^{\beta}(k)) \\ +(1-a)M(cos(\phi(k))i^{\alpha}(k) + sin(\phi(k))i^{\beta}(k)) \\ \\ \rho_2(k) = a(cos(\phi(k))\psi^{\beta}(k) + sin(\phi(k))\psi^{\alpha}(k)) \\ +(1-a)M(cos(\phi(k))i^{\beta}(k) + sin(\phi(k))i^{\alpha}(k))\end{array} \quad (12)$$

with

$$\alpha = \frac{R_r}{L_r}, \quad \gamma = \frac{M^2R_r}{\sigma L_r^2} + \frac{R_s}{\sigma}, \quad \sigma = L_s - \frac{M^2}{L_r},$$

$$\beta = \frac{M}{\sigma L_r}, \quad a = e^{-\alpha T}, \quad \mu = \frac{3M\eta_p}{2J_rL_r},$$

$$\begin{array}{l}\varphi^{\alpha}(k) = i^{\alpha}(k) + \alpha\beta T\psi^{\alpha}(k) + \eta_p\beta T\omega(k)\psi^{\beta}(k) - \gamma T i^{\alpha}(k) \\ \varphi^{\beta}(k) = i^{\beta}(k) + \alpha\beta T\psi^{\beta}(k) + \eta_p\beta T\omega(k)\psi^{\alpha}(k) - \gamma T i^{\beta}(k) \\ \phi(k) = \eta_p\theta(k)\end{array} \quad (13)$$

where θ is the rotor angular position, ω is the rotor angular speed, ψ_α is the rotor magnetic flux in α, ψ_β is the rotor magnetic flux in β, i_α is the stator current in α, i_β is the stator current in β, u_α is the input voltage in α, u_β is the input voltage in β and T_L is the load.

whit L_s, L_r and M are the stator, rotor and mutual inductance respectively; R_s and R_r are the stator and rotor resistances respectively; and n_p is the number of pole pairs.

3.2 Backlash model

The backlash model as well as a simple mechanical connection are both presented in Fig. 1. (Santos and Viera, 2008).
In the backlash characteristic shown in Fig. 1.a, $v(t)$ is the input, $u(t)$ is the output, and $c_r > 0$ is the right "crossing" on the $v-axis$, while $c_l < 0$ is the left "crossing" on the $v-axis$.

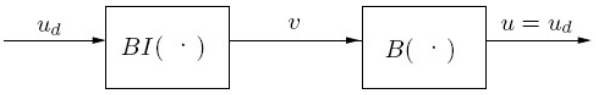

Fig. 2. Inverting a backlash.

The discrete-time model of the backlash is given by:

$$u(t) = \begin{cases} m(v(t) - c_l)\ if\ v(t) \le v_l = \dfrac{u(t-1)}{m} + c_l \\ m(v(t) - c_r)\ if\ v(t) \ge v_r = \dfrac{u(t-1)}{m} + c_r \\ u(t-1) otherwise \end{cases} \quad (14)$$

where the values v_l and v_r are the $v-axis$ projections of the intersections of the two parallel lines of slope m with the horizontal inner segment containing $u(t-1)$.
Equation (11) is the so-called friction driven hysteresis backlash model, i.e., the driven member retains its position when the backlash gap is not closed, as if kept in place by strong friction. It can be verified that (11) is a piecewise-linear dynamical system with three distinct regions of behaviour, here called upward active, downward active and gap regions.

3.3 Backlash inverse model

The desired function of a backlash inverse is to cancel the harmful effects of backlash on system performance: the delay corresponding to the time needed to traverse an inner segment of $B(.)$ and the information loss occurring on an inner segment when the output $u(t)$ remains constant while the input $v(t)$ continues is changed. That is, given a desired signal $u_d(t)$ for $u(t)$, a backlash inverse $BI(.)$ is such that $u_d(t) = B(BI(u_d(t)))$ (Fig. 2).

The discrete-time model of the backlash inverse is represented by the following mapping:

$$v(t) = \begin{cases} v(t-1) & if \quad u_d(t) = u_d(t-1) \\ \dfrac{u_d(t)}{m} + c_l & if \quad u_d(t) < u_d(t-1) \\ \dfrac{u_d(t)}{m} + c_r & if \quad u_d(t) > u_d(t-1) \end{cases} \quad (15)$$

Notice that the formulation of the backlash inverse in discrete-time does not make use of the derivatives of $u_d(t)$. Another advantage of the discrete-time formulation over the continuous-time one is that closed-loop signal bounds can be established even for the case of different slopes m.

4. NEURAL IDENTIFIER

For the induction motor identification, a recurrent high order neural network (RHONN) is used; defined as:

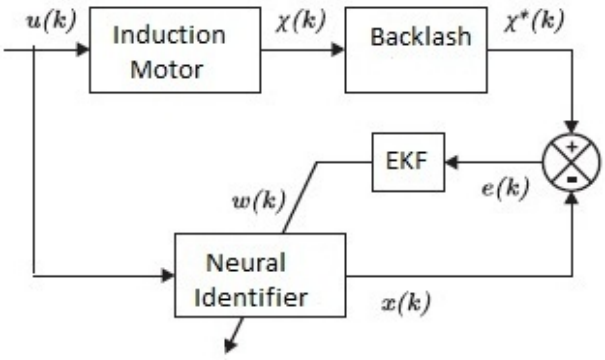

Fig. 3. Neural identifier scheme.

$$\begin{aligned}
\hat{x}_0(k+1) &= \omega_{01}(k)S_0(\theta^*(k)) + \omega_{02}(k)S_1(\theta^*)S_2(\psi^\beta(k))i^\alpha(k) \\
&\quad +\omega_{03}(k)S_1(\theta^*)S_2(\psi^\alpha(k))i^\beta(k) \\
\hat{x}_1(k+1) &= \omega_{11}(k)S(\omega^*(k)) + \omega_{12}(k)S(\omega^*)S(\psi^\beta(k))i^\alpha(k) \\
&\quad +\omega_{13}(k)S(\omega^*)S(\psi^\alpha(k))i^\beta(k) \\
\hat{x}_2(k+1) &= \omega_{21}(k)S(\theta^*(k))S(\psi^\alpha(k)) + \omega_{22}(k)i^\alpha(k) \\
\hat{x}_3(k+1) &= \omega_{31}(k)S(\theta^*(k))S(\psi^\beta(k)) + \omega_{32}(k)i^\beta(k) \\
\hat{x}_4(k+1) &= \omega_{41}(k)S(\psi^\alpha(k)) + \omega_{42}(k)S(\psi^\beta(k)) \\
&\quad +\omega_{43}(k)S(i^\alpha(k)) + \omega_{44}(k)u^\alpha(k) \\
\hat{x}_5(k+1) &= \omega_{51}(k)S(\psi^\alpha(k)) + \omega_{52}(k)S(\psi^\beta(k)) \\
&\quad +\omega_{53}(k)S(i^\beta(k)) + \omega_{54}(k)u^\beta(k)
\end{aligned} \tag{16}$$

where

$$\begin{aligned}
S(\zeta) &= tanh(\zeta) \\
S_0(\zeta) &= 10^4\left(\frac{1}{1+e^{-10-3\varsigma}} - 0.5\right) \\
S_1(\zeta) &= 10\left(\frac{1}{1+e^{-10-7\zeta}} + 0.5\right) \\
S_2(\zeta) &= 0.006\left(\frac{1}{1+e^{-0.1\zeta}} - 0.5\right)
\end{aligned} \tag{17}$$

where $\hat{x}_0(k)$ estimates the angular position $\theta(k)$; $\hat{x}_1(k)$ estimates the angular speed $\omega(k)$; $\hat{x}_2(k)$ and $\hat{x}_3(k)$ estimates the fluxes $\psi^\alpha(k)$ and $\psi^\beta(k)$, respectively; $\hat{x}_4(k)$ and $\hat{x}_5(k)$ estimates the currents $i^\alpha(k)$ and $i^\beta(k)$, respectively.
The training is performed with all neural network states initialized randomly. The covariances matrices are initialized as diagonals, and non-zero elements are: $P0 = 1e3, Q0 = 1e3, r0 = 1e3, P1 = 1e3, Q1 = 1e3, r1 = 4e7, P2 = 4e3, Q2 = 1e3, r2 = 1e3, P3 = 1e1, Q3 = 1e1, r3 = 4e7, P4 = 4e7, Q4 = 1e0, r4 = 1e7, P5 = 4e7, Q5 = 1e0, r5 = 4e7$. During the identification process the plant and the neural network (NN) operates in open-loop. Both of them (plan and NN) have the same input vector $[u_\alpha u_\beta]^T$; u_α and u_β are chirp functions with 200 volts of amplitude and incremental frequencies from 0Hz to 150Hz and 0Hz to 200Hz, respectively. The implementation is performed with a sampling time of 0.0005 s.

5. NEURAL BLOCK CONTROLLER DESIGN

The control objective is to track references of angular position and flux amplitude for the discrete-time induction motor, using the algorithm developed as follows. Let define the states

$$x^1(k) = \begin{bmatrix} \hat{x}_0(k) - \theta_r(k) \\ \Psi(k) - \Psi_r(k) \end{bmatrix} \tag{18}$$

$$x^2(k) = \begin{bmatrix} \hat{x}_4(k) \\ \hat{x}_5(k) \end{bmatrix} \tag{19}$$

where $\Psi(k) = \hat{x}_2^2(k) + \hat{x}_3^2(k)$ is the rotor flux estimated magnitude, $\Psi_r(k)$ and $\theta_r(k)$ are reference signals. Equation (16) can be represented in the block control form as (Sanchez et al., 2008)

$$\begin{aligned}
x^1(k+1) &= f_1(x^1(k)) + B_1(x^1(k))x^2(k) \\
x^2(k+1) &= f_2(x^1(k), x^2(k)) + B_2u(k)
\end{aligned} \tag{20}$$

with $u(k) = [\ u^\alpha(k)\ \ u_\beta(k)\ ]^T$ and

$$f_1(x^1(k)) = \begin{bmatrix} w_{01}(k)S_0(\theta^*(k)) - \theta_r(k+1) \\ f_{11} \end{bmatrix} \tag{21}$$

$$\begin{aligned}
f_{11} &= w_{21}^2(k)S^2(\theta^*(k))S^2(\psi_\alpha(k)) + w_{31}^2S^2(\theta^*(k))S^2(\psi_\beta(k)) \\
&\quad + I_m^2(k) - \Psi_r(k+1)
\end{aligned}$$

$$I_m = \sqrt{w_{22}^2(k)i_\alpha^2(k) + w_{32}^2(k)i_\beta^2(k)}$$

$$B_1(x^1(k)) = \begin{bmatrix} b_{11}(k) & b_{12}(k) \\ b_{21}(k) & b_{22}(k) \end{bmatrix} \tag{22}$$

$$\begin{aligned}
b_{11} &= w_{02}(k)S_1(\theta^*(k))S_2(\psi_\beta(k)) \\
b_{12} &= w_{03}(k)S_1(\theta^*(k))S_2(\psi_\alpha(k)) \\
b_{21} &= 2w_{21}(k)w_{22}(k)S(\theta^*(k))S(\psi_\alpha(k)) \\
b_{22} &= 2w_{31}(k)w_{32}(k)S(\theta^*(k))S(\psi_\beta(k))
\end{aligned}$$

$$f_2(x^2(k)) = \begin{bmatrix} f_{21}\ (k) \\ f_{22}\ (k) \end{bmatrix} \tag{23}$$

$$B_2 = \begin{bmatrix} w_{44}(k) & 0 \\ 0 & w_{54}(k) \end{bmatrix} \tag{24}$$

$$\begin{aligned}
f_{21}(k) &= w_{41}(k)S(\psi_\alpha(k)) + w_{42}(k)S(\psi_\beta(k)) \\
&\quad + w_{43}(k)S(i_\alpha(k))
\end{aligned}$$

$$\begin{aligned}
f_{22}(k) &= w_{51}(k)S(\psi_\alpha(k)) + w_{52}(k)S(\psi_\beta(k)) \\
&\quad + w_{53}(k)S(i_\beta(k))
\end{aligned}$$

Applying the block control technique (Loukianov, 2002), we define the following vector $z_1(k) = x^1(k)$. Then

$$z_1(k+1) = f_1(x^1(k)) + B_1(x^1(k))x^2(k) = Kz_1(k) \tag{25}$$

where $K = diag\{k_1, k_2\}$, with $|k_i| < 1 (i = 1, 2)$; the desired value $x^{2d}(k)$ of $x^2(k)$ is calculated from as

$$x^{2d}(k) = B_1^{-1}(x^1(k))[-f_1(x^1(k)) + Kz_1(k)] \tag{26}$$

It is desired that $x^2(k) = x^{2d}(k)$. Hence, a new error vector is defined as $z_2(k) = x^2(k) - x^{2d}(k)$, then

$$z_2(k+1) = f_3(x^1(k)) + B_2(k)u(k) \tag{27}$$

with

$$f_3(x^1(k)) = f_2(x^2(k)) - B_1^{-1}(x^1(k))[-f_1(x^1(k)) + Kz_1(k)] \tag{28}$$

Let us select the manifold for the sliding mode as $S_D(k) = z_2(k)$. In order to design a control law, a discrete-time sliding mode version is implemented as

$$u(k) = \begin{cases} u_{eq}(k) & if \quad \|u_{eq}\| \leq u_0 \\ u_o(k)\dfrac{u_{eq}}{\|u_{eq}\|} & if \quad \|u_{eq}\| > u_0 \end{cases} \tag{29}$$

where $u_{eq}(k) = -B_2^{-1}(k)f_3(x^1(k))$ is calculated from $S_D(k) = 0$ and u_0 is the control resources that bound the control. Due the time varying of RHONO weights, we need to guarantee that $B_1(\bullet)$ and $B_2(\bullet)$ are not singular; then it is necessary to avoid the zero-crossing of the weights $w_{13}(k)$, $w_{22}(k)$, $w_{32}(k)$, $w_{44}(k)$ and $w_{54}(k)$, which are so-called controllability weights. It is important to remark that in this application only the weights $w_{44}(k)$ and $w_{54}(k)$ tend to cross zero.

6. SIMULATIONS RESULTS

Simulations are performed for system (16), using the parameters given in Table 1. The identification results are included as follows: Fig. 4 shows the position estimation, with $\hat{x}_0(k)$ which estimates the the motor position $\theta(k)$; Fig. 5 shows the speed motor, with $\hat{x}_1(k)$ which estimates the speed motor $w(k)$; Fig. 6 shows the alpha flux estimation, with $\hat{x}_2(k)$ which estimates the alpha flux motor $\psi^\alpha(k)$; Fig. 7 shows the alpha current estimation, with $\hat{x}_4(k)$ which estimates the alpha current motor $i^\alpha(k)$. The result for ψ^β and i^β are similar to ψ^α and i^α. Finally, tracking results for the angular position and for the flux magnitude are present in Fig. 8 and Fig. 9; Fig. 10 shows the control law in phases α and β.

Table 1. Induction motors parameters

Parameter	Value	Description
R_s	14Ω	Stator resistance
L_s	400 mH	Stator inductance
M	377 mH	Mutual inductance
R_r	101 Ω	Rotor resistance
L_r	412.8 mH	Rotor inductance
n_p	2	Number of poles pairs
J	0.01 Kg m^2	Moment of inertia
w_n	168.5 rad s^{-1}	Nominal speed
T_{L_n}	1.1 Nm	Nominal load
T	0.0005 s	Sampling period

7. CONCLUSIONS

This paper has presented the application of recurrent high order neural network to design a discrete-time block control with sliding modes techniques for a class of discrete-time non-linear systems. The RHONN is used to perform system identification; the training of the neural networks using an extended Kalman filter, which is implemented on-line. Simulations results illustrate the effectiveness of the proposed scheme for position trajectory tracking. Presently, we are implementing in real-time this scheme for a three-phase induction motor, which has an attached gear.

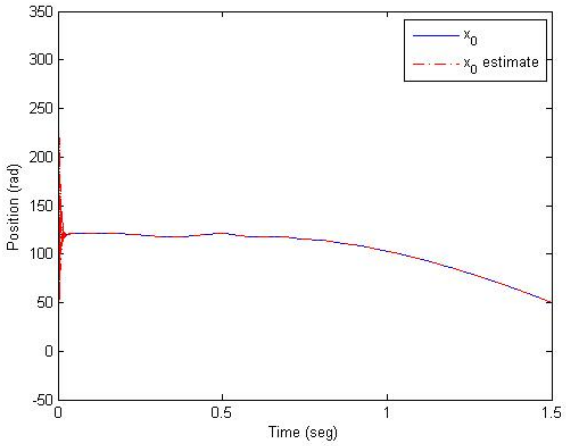

Fig. 4. Evolution of $\theta(k)$ and its estimate $\hat{x}_0(k)$.

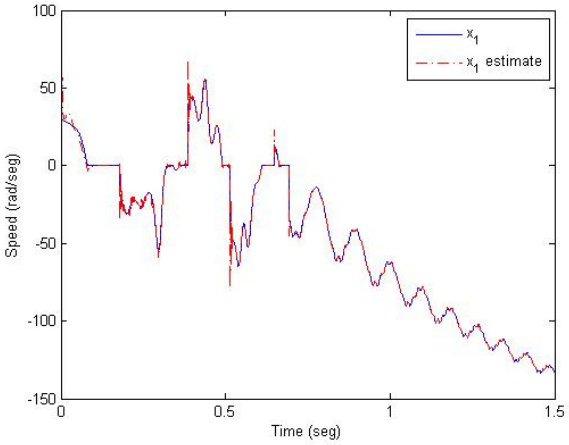

Fig. 5. Evolution of $\omega(k)$ and its estimate $\hat{x}_1(k)$.

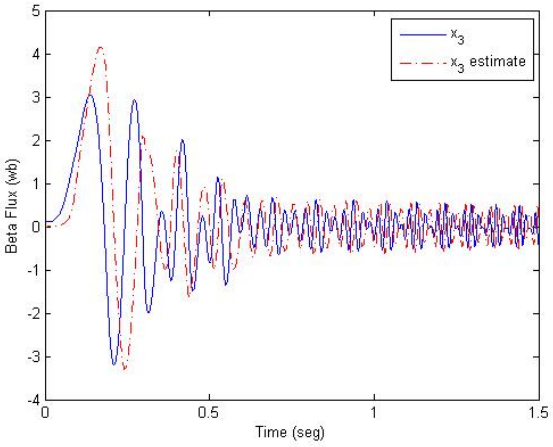

Fig. 6. Evolution of $\psi^\alpha(k)$ and its estimate $\hat{x}_2(k)$.

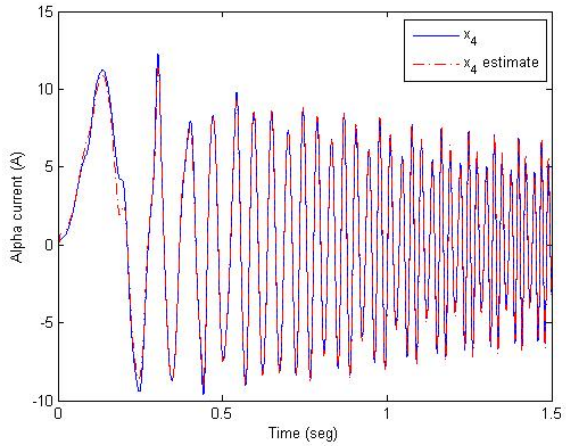

Fig. 7. Evolution of $i^{\alpha}(k)$ and its estimate $\hat{x}_4(k)$.

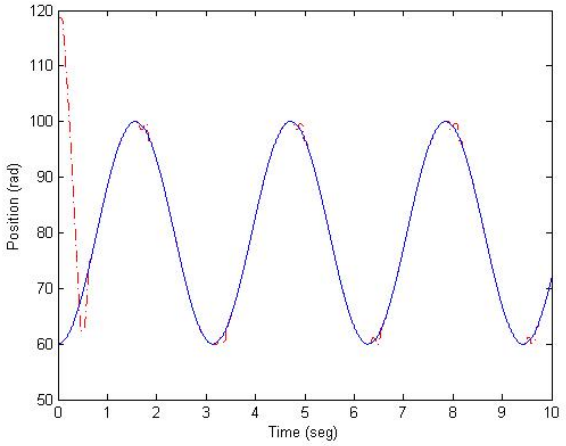

Fig. 8. Angular position tracking $\theta(k)$ (dashed line), and reference $\theta_r(k)$ (solid line).

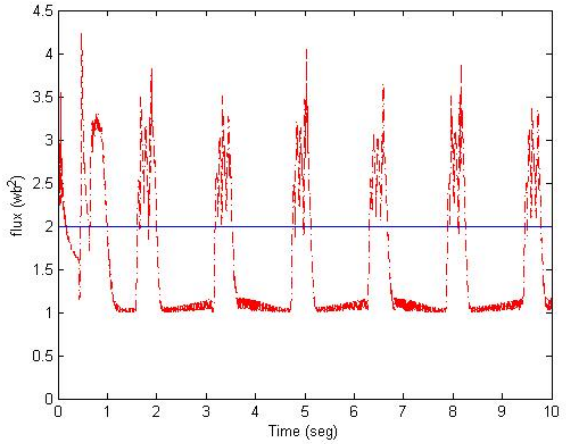

Fig. 9. Flux magnitude tracking $\Psi(k)$ (dashed line), and reference $\Psi_r(k)$ (solid line).

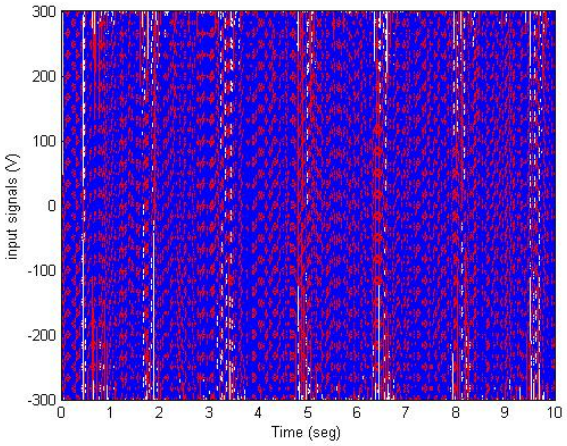

Fig. 10. Control law signals $u^{\alpha}(k)$ (solid line) and $u^{\beta}(k)$ (dashed line)

REFERENCES

Alanis, A.Y., Sanchez, E.N., and Loukianov, A.G. (2007). Discrete-time output trajectory tracking for induction motor using a neural observer. In *Intelligent Control, 2007. ISIC 2007. IEEE 22nd International Symposium on*, 584–589.

Alanis, A.Y., Sanchez, E.N., Loukianov, A.G., and Perez-Cisneros, M. (2010). Real-time discrete neural block control using sliding modes for electric induction motors. *Control Systems Technology, IEEE Transactions on*, 18(1), 11–21.

Loukianov, A.G. (2002). Robust block decomposition sliding mode control design. *Mathematical Problems in Engineering*, 8(4-5), 349–365.

Menghal, P. and Laxmi, A. (2013). Neural network based dynamic simulation of induction motor drive. In *Power, Energy and Control (ICPEC), 2013 International Conference on*, 566–571.

Mohamadian, M., Nowicki, E., Ashrafzadeh, F., Chu, A., Sachdeva, R., and Evanik, E. (2003). A novel neural network controller and its efficient dsp implementation for vector-controlled induction motor drives. *Industry Applications, IEEE Transactions on*, 39(6), 1622–1629.

Sanchez, E.N., Alanis, A.Y., and Loukianov, A.G. (2008). *Discrete-Time High Order Neural Control.* Springer Verlag, Berlin,Germany.

Sanchez, E., Loukianov, A., and Ruiz, R. (2009). Neural network identification of a double fed induction generator prototype. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, 2778–2783.

Santos, T. and Viera, F. (2008). An adaptive control approach for discrete-time systems with unknown backlash at the input. *Tendencias em matematicas aplicadas e computacional*.

Woods, G.M. (1944). Gear ratio and its effect on traction motors. *American Institute of Electrical Engineers, Transactions of the*, 63(2), 65–68.