Acepted 2006 IEEE WCCI

Discrete-Time Recurrent Neural Induction Motor Control using Kalman Learning

Alma Y. Alanis, Edgar N. Sanchez and Alexander G. Loukianov
CINVESTAV, Unidad Guadalajara, Apartado Postal 31-438, Plaza La Luna, Guadalajara, Jalisco, C.P. 45091, Mexico, e-mail:sanchez@gdl.cinvestav.mx

Abstract— **This paper deals with the adaptive tracking problem for discrete-time induction motor model in presence of bounded disturbances. In this paper, a high order neural network structure is used to identify the plant model and based on this model, a discrete-time control law is derived, which combines discrete-time block control and sliding modes techniques. The paper also includes the respective stability analysis, for the whole system with a strategy to avoid specific adaptive weights zero-croosing. Applicability of the scheme is illustrated via simulation for a discrete-time nonlinear model of an electric induction motor.**

Keywords— **Recurrent high order neural networks, Extended Kalman filtering, Induction Motor, Discrete-time sliding mode control, Neural block control.**

I. Introduction

Induction motors are widely used in industrial applications due to their reliability, simpler construction and reduced cost with respect, for example, to d.c.motors. However, the control of induction motors can be a difficult problem since the dynamics are highly nonlinear and the parameters, mainly rotor resistance and load torque, could de considered as time varying. Provided that all state variables are measured and all parameters are known, different controllers have been proposed, including field oriented controller [7], variable structure sliding mode controller [12] and [15], and more recently, exact input-output linearizing [9] and passivity-based controllers [6].

Neural networks (NN) have become a well-established methodology as exemplified by their applications to identification and control of nonlinear and complex systems [3]; the use of high order neural networks for modelling and learning has recently increased [2]. For many nonlinear systems it is often difficult to obtain their accurate and faithful mathematical models, regarding their physically complex structures and hidden parameters as discussed in [1]. Therefore, the system identification becomes important problem and even necessary before system control can be considered not only for understanding and predicting the behavior 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 some 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 for the same input [11].

The best well-known training approach for recurrent neural networks (RNN) is the back propagation through time learning [13]. However, it is a first order gradient descent method and hence its learning speed could be very slow [13]. Recently the Extended Kalman Filter (EKF) based algorithms has been introduced to train neural networks, in order to improve the learning convergence [13]. The EKF training of neural networks, both feedforward and recurrent ones, has proven to be reliable and practical for many applications over the past ten years [13]. There already exist publications about trajectory tracking using neural networks ([2], [10], [11]); in most of them, the design methodology is based on the Lyapunov approach. However most of those works were developed for continuous-time systems. On the other hand, while extensive literature is available for linear discrete-time control system, nonlinear discrete-time control design techniques have not been discussed to the same degree. For nonlinear discrete-time systems, the control problem is more complex due to the couplings among subsystems, inputs and outputs [3]. Besides, discrete-time neural networks are better fitted for real-time implementations. In recent adaptive and robust control literature, numerous approaches have been proposed for the design of nonlinear control systems. Among these, the block control (BC) constitutes a well suited design methodology [8]. Nevertheless, as well as several feedback linearization schemes the BC may present sigularities, yielding frequently, closed-loop system instability. In this paper, we use an Extended Kalman Filter (EKF)-based training algorithm for a recurrent high order neural network (RHONN), in order to identify discrete-time nonlinear systems and to overcome the controller singularity problem; based on this model, a discrete-time control law is derived, which combines discrete-time block control and sliding modes techniques. The block control approach is used to design a nonlinear sliding surface such that the resulting sliding mode dynamics is described by a desired linear system. The proposed neural identifier and control applicability is illustrated by trajectory tracking for induction motors.

II. Discrete-Time 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 [8]

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

with

$$\begin{aligned}
\rho_1(k) &= a\left(\cos(\phi(k))\psi^{\alpha}(k) + \sin(n_p\phi(k))\psi^{\beta}(k)\right) \\
&\quad +b\left(\cos(\phi(k))i^{\alpha}(k) + \sin(\phi(k))i^{\beta}(k)\right) \\
\rho_2(k) &= a\left(\cos(\phi(k))\psi^{\alpha}(k) - \sin(\phi(k))\psi^{\beta}(k)\right) \\
&\quad +b\left(\cos(\phi(k))i^{\alpha}(k) - \sin(\phi(k))i^{\beta}(k)\right) \\
\varphi^{\alpha}(k) &= i^{\alpha}(k) + \alpha\beta T\psi^{\alpha}(k) + n_p\beta T\omega(k)\psi^{\alpha}(k) \\
&\quad -\gamma T i^{\alpha}(k) \\
\varphi^{\beta}(k) &= i^{\beta}(k) + \alpha\beta T\psi^{\beta}(k) + n_p\beta T\omega(k)\psi^{\beta}(k) \\
&\quad -\gamma T i^{\beta}(k) \\
\phi(k) &= n_p\theta(k)
\end{aligned} \tag{2}$$

where $b = (1-a)M$, $\alpha = \frac{R_r}{L_r}$, $\gamma = \frac{M^2 R_r}{\sigma L_r^2} + \frac{R_s}{\sigma}$, $\sigma = L_s - \frac{M^2}{L_r}$, $\beta = \frac{M}{\sigma L_r}$, $a = e^{-\alpha T}$, $\mu = \frac{M n_p}{J L_r}$ with 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^{α} and i^{β} represent the currents in the α and β phases, respectively; ψ^{α} and ψ^{β} represent the fluxes in the α and β phases, respectively and θ is the rotor angular displacement.

III. Mathematical definitions

A. Stability Definitions

This section close follows [3]. Through this paper, we use k as the step sampling, $k \in 0 \cup \mathbb{Z}^+$, $|\bullet|$ as the absolute value and, $\|\bullet\|$ as the Euclidian norm for vectors and as any adequate norm for matrices. Consider a MIMO nonlinear system

$$\chi(k+1) = F(\chi(k), u(k)) \tag{3}$$

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 (3) is said to be forced, or to have input. In contrast, the system described by an equation without explicit presence of an input u, that is

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

is said to be unforced.

Definition 2: The system (4), can be obtained after selecting the input u as a feedback function

$$u(k) = \xi(\chi(k)) \tag{5}$$

Such substitution eliminates u:

$$\chi(k+1) = F(\chi(k), \xi(\chi(k))) \tag{6}$$

and yields the unforced system (6).

Definition 3: [3] A solution of $(3)-(5)$ is semiglobally uniformly ultimately bounded (SGUUB), if for any Ω, 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 (3) is said to be SGUUB if, for any a priory given (arbitrarily large) bounded set Ω and any a priory given (arbitrarily small) set Ω_0, which contains $(0,0)$ as an interior point, there exists a control (5), such that every trayectory of the closed loop system starting from Ω enters the set $\Omega_0 = \{\chi(k) \mid \|\chi(k)\| < \varepsilon\}$, in a finite time and remains in it thereafter.

Theorem 1 [3] Let $V(\chi(k))$ be a Lyapunov function for the discrete-time system (3), which satisfies the following properties:

$$\begin{aligned}
\gamma_1(\|\chi(k)\|) &\leq V(\chi(k)) \leq \gamma_2(\|\chi(k)\|) \\
V(\chi(k+1)) - V(\chi(k)) &= \Delta V(\chi(k)) \\
&\leq -\gamma_3(\|\chi(k)\|) + \gamma_3(\zeta)
\end{aligned}$$

where ζ 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

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

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 4: [5] A subset $S \in \Re^n$ is bounded if there is $r > 0$ such that $\|\chi\| \leq r$ for all $\chi \in S$.

Definition 5: [5] The system (7) is said to be BIBO stable if for a bounded input $u(k)$, the system produces a bounded output $y(k)$ for $0 < k < \infty$.

Lemma 1: Consider a linear time varying discrete-time system given by

$$\begin{aligned}
\chi(k+1) &= A(k)\chi(k) + Bu(k) \\
y(k) &= C\chi(k)
\end{aligned} \tag{7}$$

where $A(k)$, B and C are appropriately dimensional matrices, $\chi \in \Re^n$, $u \in \Re^m$ and $y \in \Re^p$.

Let $\Phi(k(1), k(0))$ be the state-transition matrix corresponding to $A(k)$ for system (7), i.e. $\Phi(k(1), k(0)) = \prod_{k=k(0)}^{k=k_1-1} A(k)$. If $\|\Phi(k(1), k(0))\| < 1 \quad \forall k(1) > k(0) > 0$, then the system (7) is

1) globally exponentially stable for the unforced system and

2) Bounded Input-Bounded Output (BIBO) stable [3].

B. Discrete-time Recurrent High Order Neural Networks

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

$$x_i(k+1) = w_i^\top z_i(x(k), u(k)), \quad i = 1, \cdots, n \qquad (8)$$

where x_i $(i = 1, 2, \cdots, n)$ is the state of the ith neuron, L_i is the respective number of higer-order connections, $\{I_1, I_2, \cdots, I_{L_i}\}$ is a collection of non-ordered subsets of $\{1, 2, \cdots, n\}$, n is the state dimension, w_i $(i = 1, 2, \cdots, n)$ is the respective on-line adapted weight vector, and $z_i(x(k), u(k))$ is given by

$$z_i(x(k), u(k)) = \begin{bmatrix} z_{i_1} \\ z_{i_2} \\ \vdots \\ z_{i_{L_i}} \end{bmatrix} = \begin{bmatrix} \Pi_{j \in I_1} y_{i_j}^{d_{ij}(1)} \\ \Pi_{j \in I_2} y_{i_j}^{d_{ij}(2)} \\ \vdots \\ \Pi_{j \in I_{L_i}} y_{i_j}^{d_{ij}(L_i)} \end{bmatrix} \qquad (9)$$

with $d_{j_i}(k)$ being a nonnegative integers, and y_i is defined as follows:

$$y_i = \begin{bmatrix} y_{i_1} \\ \vdots \\ y_{i_1} \\ y_{i_{n+1}} \\ \vdots \\ y_{i_{n+m}} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix} \qquad (10)$$

In (10), $u = [u_1, u_2, \ldots, u_m]^\top$ is the input vector to the neural network, and $S(\bullet)$ is defined by

$$S(x) = \frac{1}{1 + \exp(-\beta x)} \qquad (11)$$

Consider the problem to approximate the general discrete-time nonlinear system (3), by the following discrete-time RHONN serie-parallel representation [11]:

$$\chi_i(k+1) = w_i^{*\top} z_i(x(k), u(k)) + \epsilon_{z_i}, \quad i = 1, \cdots, n \quad (12)$$

where χ_i is the ith plant state, ϵ_{z_i} is a bounded approximation error, which can be reduced by increasing the number of the adjustable weights [11]. Assume that there exists 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 [11]. 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

$$\widetilde{w}_i(k) = w_i(k) - w_i^* \qquad (13)$$

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

IV. The EKF Training Algorithm

It is known, that Kalman filtering (KF) estimates the state of a linear system with additive state and output white noises [4], [14]. 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 nonlinear, an EKF-type is required (see [10] and references therein). The training goal is to find the optimal weight values which minimize the prediction error. In this paper, we use a modified EKF-based training algorithm described by

$$\begin{aligned} w_i(k+1) &= w_i(k) + \eta_i K_i(k) e_i(k) \qquad (14) \\ K_i(k) &= \begin{cases} P_i(k) H_i(k) M_i(k) & \text{if } \|w_i(k)\| > c_i \\ 0 & \text{if } \|w_i(k)\| < c_i \end{cases} \\ P_i(k+1) &= P_i(k) - K_i(k) H_i^\top(k) P_i(k) + Q_i(k) \\ i &= 1, \cdots, n \end{aligned}$$

with

$$\begin{aligned} M_i(k) &= \left[R_i(k) + H_i^\top(k) P_i(k) H_i(k)\right]^{-1} \qquad (15) \\ e_i(k) &= \chi_i(k) - x_i(k) \qquad (16) \end{aligned}$$

where $c_i > 0$ is a constrain used to avoid the zero-croosing, $e_i(k) \in \Re$ is the respective identification error, $P_i(k) \in \Re^{L_i \times L_i}$ is the prediction error covariance matrix at step k, $w_i \in \Re^{L_i}$ is the weight (state) vector, L_i is the respective number of neural network weights, χ_i is the ith plant state, x_i is the ith neural network state, n is the number of states, $K_i \in \Re^{L_i}$ is the Kalman gain vector, $Q_i \in \Re^{L_i \times L_i}$ is the NN weight estimation noise covariance matrix, $R_i \in \Re$ is the error noise covariance; $H_i \in \Re^{L_i}$ is a vector, in which each entry (H_{ij}) is the derivative of one of the neural network state, (x_i), with respect to one neural network weight, (w_{ij}), as follows

$$H_{ij}(k) = \left[\frac{\partial x_i(k)}{\partial w_{i_j}(k)}\right]^\top_{w_i(k) = w_i(k+1)} \qquad (17)$$

where $i = 1, ..., n$ and $j = 1, ..., L_i$. If we select $c_i = 0$ the modified EKF (MEKF) (14) becomes the standard extended Kalman Filter [4]. Usually P_i and Q_i are initialized as diagonal matrices, with entries $P_i(0)$ and $Q_i(0)$, respectively. It is important to remark that $H_i(k)$, $K_i(k)$ and $P_i(k)$ for the EKF are bounded; for a detailed explanation of this fact see [14].

V. Identification

In this section we consider the problem to identify the nonlinear system (3) using a RHONN (8) trained with a MEKF algorithm (14) and we establish the main result in the following theorem.

Theorem 2: The RHONN (8) trained with the MEKF-based algorithm (14) to identify the nonlinear plant (8), ensures that the identification error (16) is semiglobally uniformly ultimately bounded (SGUUB); moreover, the RHONN weights remain bounded.

Proof: Case 1. $\|w_i(k)\| > c_i$. Consider the Lyapunov function candidate

$$V_i(k) = \frac{1}{2}e_i^2(k) + \widetilde{w}_i^\top(k)\,\widetilde{w}_i(k) \tag{18}$$

whose first difference is:

$$\Delta V_i(k) = V_i(k+1) - V_i(k) \tag{19}$$

From (13) and (14), then

$$\widetilde{w}_i(k+1) = \widetilde{w}_i(k) + \eta_i K_i(k)\,e_i(k) \tag{20}$$

Let us define

$$\begin{aligned} & [\widetilde{w}_i(k) + \eta_i K_i(k)\,e_i(k)]^\top \\ & \times [\widetilde{w}_i(k) + \eta_i K_i(k)\,e_i(k)] \\ = & \ \widetilde{w}_i^\top(k)\,\widetilde{w}_i(k) + 2\eta_i e_i(k)\,\widetilde{w}_i^\top(k)\,K_i(k) \\ & + \eta_i^2 e_i^2(k)\,K_i^\top(k)\,K_i(k) \end{aligned} \tag{21}$$

From (16), it follows

$$\begin{aligned} e_i(k+1) &= e_i(k) + \Delta e_i(k) \\ e_i^2(k+1) &= e_i^2(k) + 2e_i(k)\,\Delta e_i(k) + (\Delta e_i(k))^2 \end{aligned} \tag{22}$$

where $\Delta e_i(k)$ is the difference error. Substituting (21) and (22) in (18) and then in (19) yields

$$\begin{aligned} \Delta V_i(k) &= e_i(k)\,\Delta e_i(k) + \frac{1}{2}(\Delta e_i(k))^2 \\ & \quad + 2\eta_i e_i(k)\,\widetilde{w}_i^\top(k)\,K_i(k) \\ & \quad + \eta_i^2 e_i^2(k)\,K_i^\top(k)\,K_i(k) \end{aligned} \tag{23}$$

From (16), we obtain

$$\frac{\partial e_i(k)}{\partial w_i(k)} = -\frac{\partial x_i(k)}{\partial w_i(k)} \tag{24}$$

Let as approximate (24) by

$$\Delta e_i(k) = \left[\frac{\partial e_i(k)}{\partial w_i(k)}\right]^\top \Delta w_i(k) \tag{25}$$

Using (12), (17) and (24) in (25), yields

$$\Delta e_i(k) = -\eta_i H_i^\top(k)\,K_i(k)\,e_i(k) \tag{26}$$

Defining

$$g_i = \max\left\|H_i^\top(k)\,P_i(k)\,H_i(k)\,M_i(k)\right\|$$

with $M_i(k)$ as in (15), (26) can be rewritten as

$$\Delta e_i(k) \le -\eta_i g_i e_i(k) \tag{27}$$

Using (27) in (23), then

$$\begin{aligned} \Delta V_i(k) \le & -\eta_i g_i |e_i(k)|^2 + \frac{1}{2}\eta_i^2 g_i^2 |e_i(k)|^2 \\ & + 2\eta_i |e_i(k)| \left\|\widetilde{w}_i^\top(k)\right\| \|K_i(k)\| \\ & + \eta_i^2 |e_i(k)|^2 \|K_i(k)\|^2 \end{aligned} \tag{28}$$

The weight adaptation dynamics (20) can be written as

$$\begin{aligned} \widetilde{w}_i^\top(k+1) &= \widetilde{w}_i^\top(k) - \eta_i K_i^\top(k)\,z_i(k)\,\widetilde{w}_i^\top(k) \\ & \quad + \eta_i K_i^\top(k)\,\epsilon_{z_i} \\ &= A_i(k)\,\widetilde{w}_i^\top(k) + B_i(k)\,v_{z_i}(k) \end{aligned}$$

with

$$\begin{aligned} A_i(k) &= \left[I - \eta_i K_i^\top(k)\,z_i(k)\right] \\ B_i(k) &= \eta_i,\ \ v_{z_i}(k) = K_i^\top(k)\,\epsilon_{z_i}(k) \end{aligned} \tag{29}$$

As in [16], in this paper the plant (3) is assumed to be BIBO, ϵ_{z_i} is also assumed to be bounded and $z_i(k)$ is bounded too. Hence $A_i(k)$ always satisfies $\|\Phi(k(1), k(0))\| < 1$. By applying Lemma 1, $\widetilde{w}_i(k)$ is bounded. Then in (28) $\Delta V_i(k) \le 0$, once $|e_i(k)| > \kappa_i$. with $\kappa_i = \frac{\left|4w_i^\top K_i\right|}{\left|2g_i - \eta_i g_i^2 - 2\eta_i \|K\|_i^2\right|}$; furthermore $g > 0$ and $\eta_i > 0$. This implies the boundness of $V_i(k)$ for $k \ge 0$ which leads to the SGUUB of $e_i(k)$.

Case 2. $\|w_i(k)\| < c_i$. Consider the same Lyapunov function candidate as in case 1 (19) if $K_i = 0$ this implies that $\Delta V_i(k) = 0$ then the identification error and the weights are bounded. ■

The constraint c_i allows to eliminate the controller singularities for specific weights zero-crossing.

VI. Neural Block Control (NBC)

In this paper, the control law is based on the neural network identifier (8) updated with the MEKF (14). The control scheme is based on the following proposition

Proposition 1. Given a desired output trajectory y_r, a dynamic system with output y_p, and a neural network with output y_n, then it is possible to establish the following inequality [2]:

$$\|y_r - y_p\| \le \|y_n - y_p\| + \|y_r - y_n\|$$

where $y_r - y_p$ is the system output tracking error, $y_n - y_p$ is the output identification error, and $y_r - y_n$ is the RHONN output tracking error.

Based on this proposition, it is possible to divide the tracking error in two parts [2]:
1. Minimization of $y_n - y_p$,which can be achieved by the proposed on-line identification algorithm (14) on the basis of theorem 2.
2. Minimization of $y_n - y_r$, for that a tracking algorithm is developed on the basis of the neural identifier (8). This can be reached by designing a control law based on the RHONN model. To design such controller we propose to use the NBC methodology [2], [8].

Due to space limitations and given that the convergence of the term $\|y_r - y_n\|$ has been done before [2], [8], we do not include it in this paper.

VII. Induction Motor Control

In this section we apply the above developed scheme to control a three-phase induction motor, which model is described in section II.

A. Neural network identification

The RHONN proposed for this application is as follows:

$$\begin{aligned} x_1(k+1) &= w_{11}(k)S(\omega(k)) \\ &\quad + w_{12}(k)S(\omega)S\left(\psi^{\beta}(k)\right)i^{\alpha}(k) \\ &\quad + w_{13}(k)S(\omega)S(\psi^{\alpha}(k))i^{\beta}(k) \\ x_2(k+1) &= w_{21}(k)S(\omega(k))S\left(\psi^{\beta}(k)\right) \\ &\quad + w_{22}(k)i^{\beta}(k) \\ x_3(k+1) &= w_{31}(k)S(\omega(k))S(\psi^{\alpha}(k)) \\ &\quad + w_{32}(k)i^{\alpha}(k) \\ x_4(k+1) &= w_{41}(k)S(\psi^{\alpha}(k)) + w_{42}(k)S\left(\psi^{\beta}(k)\right) \\ &\quad + w_{43}(k)S(i^{\alpha}(k)) + w_{44}(k)u^{\alpha}(k) \\ x_5(k+1) &= w_{51}(k)S(\psi^{\alpha}(k)) + w_{52}(k)S\left(\psi^{\beta}(k)\right) \\ &\quad + w_{53}(k)S\left(i^{\beta}(k)\right) + w_{54}(k)u^{\beta}(k) \end{aligned} \tag{30}$$

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

B. Neural Block Controller design

Given full state measurements, the control objective is to develop velocity and flux amplitude tracking for the discrete-time induction motor model (30), using the discrete-time block control and sliding mode techniques. Let us define the following states as

$$x^1(k) = \begin{bmatrix} x_1(k) - \omega_r(k) \\ \Psi(k) - \Psi_r(k) \end{bmatrix}, \quad x^2(k) = \begin{bmatrix} i^{\alpha}(k) \\ i^{\beta}(k) \end{bmatrix} \tag{31}$$

where $\Psi(k) = x_2^2(k) + x_3^2(k)$ is the rotor flux identify magnitude, $\Psi_r(k)$ and $\omega_r(k)$ are reference signals. Then

$$\begin{aligned} \Psi(k+1) = {} & w_{21}^2(k)S^2(\omega(k))S^2\left(\psi^{\beta}(k)\right) \\ & + w_{22}^2(k)i^{\beta^2}(k) + w_{32}^2(k)i^{\alpha^2}(k) \\ & + w_{31}^2(k)S^2(\omega(k))S^2(\psi^{\alpha}(k)) \\ & + 2w_{21}(k)S(\omega(k))S\left(\psi^{\beta}(k)\right)w_{22}(k)i^{\beta}(k) \\ & + 2w_{31}(k)S(\omega(k))S(\psi^{\alpha}(k))w_{32}(k)i^{\alpha}(k) \end{aligned}$$

Using (31), (30) can be represented in the block control form consisting of two blocks

$$\begin{aligned} x^1(k+1) &= f_1\left(x^1(k)\right) + B_1\left(x^1(k)\right)x^2(k) \\ x^2(k+1) &= f_2\left(x^1(k), x^2(k)\right) + B_2 u(k) \end{aligned} \tag{32}$$

with $u(k) = \begin{bmatrix} u^{\alpha}(k) & u^{\beta}(k) \end{bmatrix}^{\top}$ and

$$\begin{aligned} f_1\left(x^1(k)\right) &= \begin{bmatrix} w_{11}(k)S(\omega(k)) - \omega_r(k+1) \\ f_{11}(k) \end{bmatrix} \\ f_{11}(k) &= w_{21}^2(k)S^2(\omega(k))S^2\left(\psi^{\beta}(k)\right) \\ &\quad + w_{31}^2(k)S^2(\omega(k))S^2(\psi^{\alpha}(k)) \\ &\quad + w^2 I_m^2(k) - \Psi_r(k+1) \\ I_m(k) &= \sqrt{w_{22}^2(k)i^{\alpha^2}(k) + w_{32}^2(k)i^{\beta^2}(k)} \\ B_1\left(x^1(k)\right) &= \begin{bmatrix} b_{11}(k) & b_{12}(k) \\ b_{21}(k) & b_{22}(k) \end{bmatrix} \\ b_{11}(k) &= w_{12}(k)S(\omega)S\left(\psi^{\beta}(k)\right) \\ b_{12}(k) &= w_{13}(k)S(\omega)S(\psi^{\alpha}) \\ b_{21}(k) &= 2w_{31}(k)w_{32}(k)S(\omega(k))S(\psi^{\alpha}(k)) \\ b_{22}(k) &= 2w_{21}(k)w_{22}(k)S(\omega(k))S\left(\psi^{\beta}(k)\right) \\ f_2\left(x^2(k)\right) &= \begin{bmatrix} f_{21}(k) \\ f_{22}(k) \end{bmatrix}, B_2 = \begin{bmatrix} w_{44}(k) & 0 \\ 0 & w_{54}(k) \end{bmatrix} \\ f_{21}(k) &= w_{41}(k)S(\psi^{\alpha}(k)) \\ &\quad + w_{42}(k)S\left(\psi^{\beta}(k)\right) + w_{43}(k)S(i^{\alpha}(k)) \\ f_{22}(k) &= w_{51}(k)S(\psi^{\alpha}(k)) \\ &\quad + w_{52}(k)S\left(\psi^{\beta}(k)\right) + w_{53}(k)S\left(i^{\beta}(k)\right) \end{aligned}$$

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

$$z_1(k+1) = f_1\left(x^1(k)\right) + B_1\left(x^1(k)\right)x^2(k) = Kz_1(k) \tag{33}$$

where $K = diag\{k_1, k_2\}$, with $k_1 > 0$ and $k_2 > 0$; then the desired value $x^{2d}(k)$ of $x^2(k)$ is calculated from (33) as

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

It is desired that and $x^2(k) = x^{2d}(k)$. In this way, it is defined a second new error vector

$$z_2(k) = x^2(k) - x^{2d}(k)$$

Then

$$z_2(k+1) = f_3\left(x^1(k)\right) + B_2(k)\,u(k)$$

with

$$\begin{aligned} f_3\left(x^1(k)\right) &= f_2\left(x^2(k)\right) \\ &\quad -B_1^{-1}\left(x^1(k)\right)\left[-f_1\left(x^1(k)\right) + Kz_1(k)\right] \end{aligned}$$

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

$$u(k) = \begin{cases} u_{eq}(k) & \text{if } \left\|u_{eq}(k)\right\| \leq u_0 \\ u_0 \frac{u_{eq}(k)}{\left\|u_{eq}(k)\right\|} & \text{if } \left\|u_{eq}(k)\right\| > u_0 \end{cases}$$

where $u_{eq}(k) = -B_2^{-1}(k)\,f_3\left(x^1(k)\right)$ is calculated from $S(k) = 0$ and u_0 is the control resources that bound the control. Due the time varying of RHONN weights, we need to guarantee that $B_1(\bullet)$ and $B_2(\bullet)$ are not singulars; 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 the so-called controllability weights [2]. It is important to remark that in this application only the weights $w_{44}(k)$ and $w_{54}(k)$ tend to cross zero.

C. Reduced order nonlinear observer

The last control algorithm requires the full state measurement assumption [8]. However, rotor fluxes measurement is a difficult task. Here, a reduced order nonlinear observer is designed for fluxes on the basis of rotor speed and currents measurements. The flux dynamics in (30) can be written as

$$\Psi(k+1) = aG(k)\,\Psi(k) + (1-a)\,MG(k)\,\mathbf{I}(k)$$

with $\mathbf{I}(k) = \left[i^\alpha(k), i^\beta(k)\right]^\top$ and:

$$G(k) = \begin{bmatrix} \cos\left(n_p T\omega(k)\right) & -\sin\left(n_p T\omega(k)\right) \\ \sin\left(n_p T\omega(k)\right) & \cos\left(n_p T\omega(k)\right) \end{bmatrix} \quad (34)$$

The proposed observer for the system (30), assuming speed and current measurements, and an unknown load is given in [8]

$$\widehat{\Psi}(k+1) = aG(k)\,\widehat{\Psi}(k) + (1-a)\,MG(k)\,\mathbf{I}(k)$$

Let us define

$$e^\Psi(k) = \Psi(k) - \widehat{\Psi}(k)$$

Then:

$$e^\Psi(k+1) = aG(k)\,e^\Psi(k) \quad (35)$$

A Lyapunov candidate function to proof stability of $e^\Psi(k)$ is:

$$V(k) = e^{\Psi^\top}(k)\,e^\Psi(k)$$

with

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) \\ &= e^{\Psi^\top}(k-1)\,e^\Psi(k+1) - e^{\Psi^\top}(k)\,e^\Psi(k) \\ &= e^{\Psi^\top}(k)\left(a^2 G^\top(k)\,G(k) - I\right)e^\Psi(k) \quad (36) \end{aligned}$$

where

$$a^2 G^\top(k)\,G(k) - I < 0$$

By (34), $G^\top(k)\,G(k) = I$ then the condition is reduced to

$$\begin{bmatrix} a^2 & 0 \\ 0 & a^2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} < 0$$

Then $a < 1$ where $a = e^{-\alpha T}$ and I is an identity matrix. This condition is satisfied due to the fact that T and α are always positive. Hence the increment of the Lyapunov function (36) is always negative, implying that the tracking error tends asymptotically to zero. Now, we use $\widehat{\psi}^\alpha$ and $\widehat{\psi}^\beta$ to implement the control algorithm described above.

D. Simulation results

Simulations are performed for the system (30), using the following parameters.

Table 1. Induction motor parameters.

Parameter	Value	Description
R_s	14Ω	Stator resistance
L_s	$400mH$	Stator inductance
M	$377mH$	Mutual inductance
R_r	10.1Ω	Rotor resistance
L_r	$412.8mH$	Rotor inductance
n_p	2	Number of pole pairs
J	$0.01Kgm^2$	Moment of inertia
ω_n	$168.5rad/s$	Nominal speed
T_{L_n}	$1.1Nm$	Nominal load
T	$0.001s$	Sampling period

The tracking results are presented in Fig. 1, and Fig. 2. There the tracking and identification performance can be verified for the two plant outputs. Fig. 3 displays the load torque applied as an external disturbance. Fig. 4 presents the parametric variation introduced in the rotor resistance (R_r) as a variation of 1 Ohm per second. Fig. 5 shows the weight evolution. Fig. 6 and Fig. 7 portray the fluxes and their estimates.

VIII. Conclusions

This paper has presented the application of recurrent high order neural networks to design a block control for a class of discrete-time nonlinear systems. The training of the neural networks is performed on-line using an extended Kalman filter. The boundness of the identification error is established on the basis of the Lyapunov approach. Simulation results illustrate the applicability of the proposed control methodology.

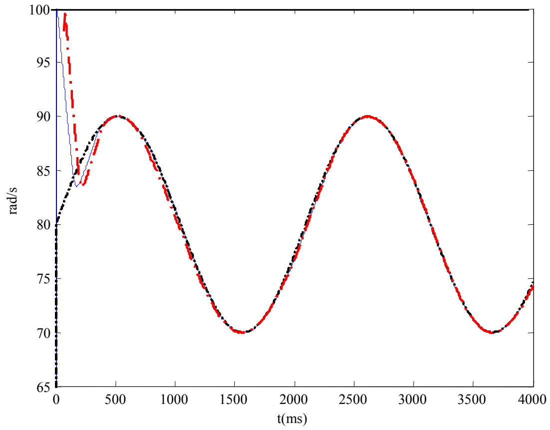

Fig. 1. Tracking performance $\omega(k)$ (solid line), $x_1(k)$ (dash-dot line) and $\omega_r(k)$ (dashed line).

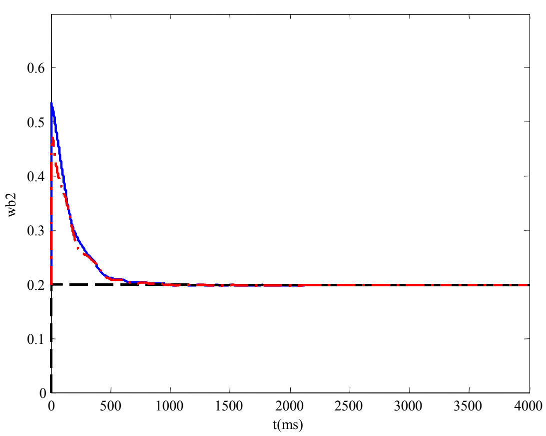

Fig. 2. Tracking performance $\Psi(k)$ (solid line), $x_2^2+x_3^2$ (dash-dot line) and $\Psi_d(k)$ (dashed line).

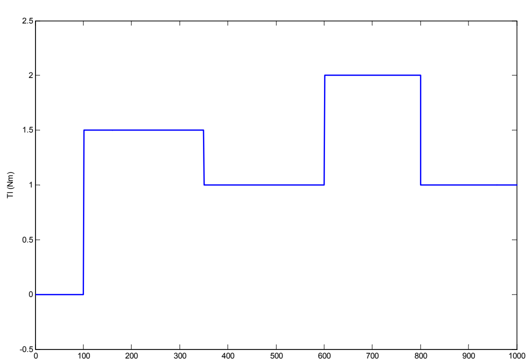

Fig. 3. Load torque $T_L(k)$

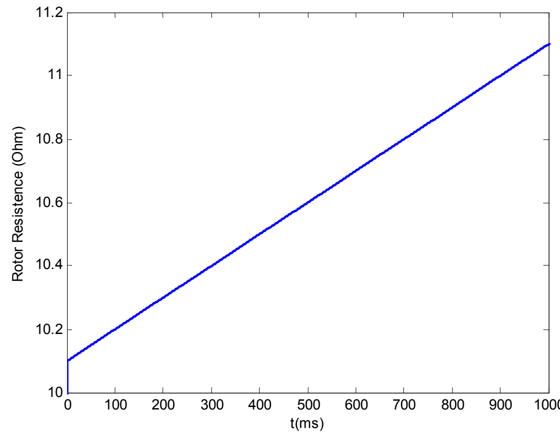

Fig. 4. Rotor resistance variation (R_r)

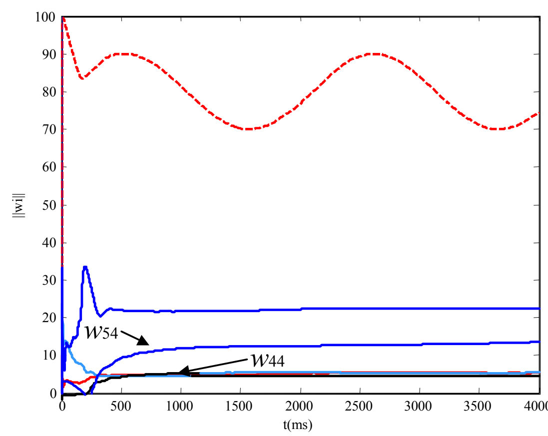

Fig. 5. Weight evolution

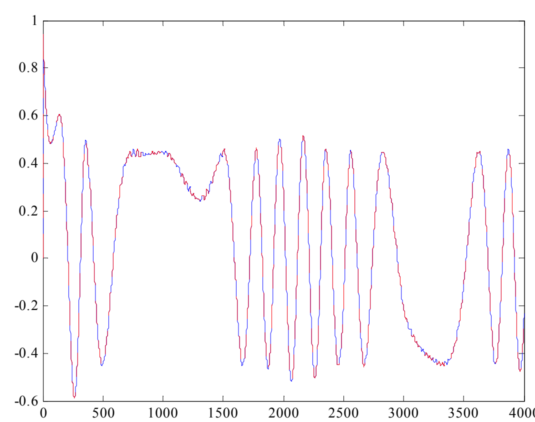

Fig. 6. Time evolution of $\psi^{\alpha}(k)$ and its estimate (real in solid line and estimated in dashed line)

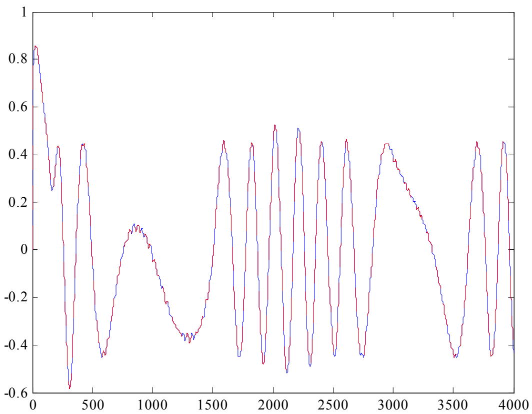

Fig. 7. Time evolution of $\psi^{\beta}(k)$ and its estimate (real in solid line and estimated in dashed line)

References

[1] C. K. Chui and G. Chen, *Kalman Filtering with Real-Time Applications,* 3rd ed., Springer-Verlag, New York, USA, 1998.

[2] R. A. Felix, E. N. Sanchez and A.G. Loukianov, "Avoding controller singularities in adaptive recurrent neural control", *Proceedings IFAC'05*, Praga, July, 2005.

[3] S. S. Ge, J. Zhang and 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, Vol. 34, No. 4, August, 2004.

[4] R. Grover and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2nd ed., John Wiley and Sons, New York, USA, 1992.

[5] H. Khalil, *Nonlinear Systems*, 2nd ed, Prentice Hall, Upper Saddle River, N. J., USA, 1996.

[6] F. Khorrami, P. Krishnamurthy and H. Melkote, *Modeling and Adaptive Nonlinear Control of Electric Motors*, Springer-Verlag, New York, USA, 2003.

[7] Leonard W. (1986), "Microcomputer control of high dynamic performance AC-drives- a survey", *Automatica*, Vol.22, pp. 1-19.

[8] A. G. Loukianov, J. Rivera and J. M. Cañedo, "Discrete-time sliding mode control of an induction motor", *Proceedings IFAC'02*, Barcelone, Spain, July, 2002.

[9] Luca A. D. and G. Ulivi. (1989). "Design of exact nonlinear controller for induction motors", *IEEE Trans. Automat. Control*, Vol.34, pp. 1304-13

[10] E. N. Sanchez, A. Y. Alanis and G. Chen, "Recurrent neural networks trained with Kalman filtering for discrete chaos reconstruction", *Proceedings of Asian-Pacific Workshop on Chaos Control and Synchronization'04*, Melbourne, Australia, July, 2004.

[11] G. A. Rovithakis and M. A. Chistodoulou, *Adaptive Control with Recurrent High -Order Neural Networks*, Springer Verlag, New York, USA, 2000.

[12] Shabanovic A. and D. Izosimov. (1981) "Application of sliding modes to induction motor control", *IEEE Trans.Ind.Applicat.*, Vol. 17, pp. 41-49.

[13] S. Singhal and L. Wu, *Training multilayer perceptrons with the extended Kalman algorithm*, in D. S. Touretzky (ed), Advances in Neural Information Processing Systems 1, pp. 133-140, Morgan Kaufmann, San Mateo, CA, USA, 1989.

[14] Y. Song and J. W. Grizzle, "The extended Kalman filter as local asymptotic observer for discrete-time nonlinear systems", *Journal of Mathematical systems*, Estimation and Control,Vol. 5, No. 1, pp. 59-78, Birkhauser-Boston, 1995.

[15] Utkin V. A. (1993). "AC drives control problems", *Avtomatica i Telemekhanika*, No. 12, pp. 53-65.

[16] W. Yu and X. Li, "Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms", *Information Sciences,* Vol. 158, pp. 131-147, 2004.