Submited to IEEE TNN, Special Issue on Neural Control

Discrete Time Adaptive Backstepping Nonlinear Control via High Order Neural Networks

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 adaptive tracking for discrete-time MIMO nonlinear systems in presence of bounded disturbances. In this paper, a high order neural network structure is used to approximate a control law designed by the backstepping technique, applied to a block strict feedback form (BSFF). The paper also includes the respective stability analysis, on the basis of the Lyapunov approach, for the extended Kalman filter (EKF)-based NN learning algorithm. Applicability of the scheme is illustrated via simulation for a discrete-time nonlinear model of an electric induction motor.

Keywords

High order neural networks, Extended Kalman filtering, Backstepping, Discrete-time systems, Electric induction motor.

I. Introduction

Neural networks (NN) have become a well-established methodology as exemplified by their applications to identification and control of general nonlinear and complex systems [5]; the use of high order neural networks for modelling and learning has recently increased [17]. Specifically, the problem of designing robust neural controllers for nonlinear systems with uncertainties and disturbances, which guarantee stability and trayectory tracking, has received an increasing attention lately.

Using neural networks, control algorithms can be developed to be robust to uncertainties and modelling errors. The most used NN structures are: *Feedforward* networks and *Recurrent* ones [17]. The last type offers a better suited tool to model and control nonlinear systems [15].

The best well-known training approach for recurrent neural networks (RNN) is the back propagation through time learning [22]. However, it is a first order gradient descent method and hence its learning speed could be very slow [10]. Recently the Extended Kalman Filter (EKF) based algorithms has been introduced to train neural networks [3]. With the EKF based algorithm, the learning convergence is improved [10]. 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 [3].

There already exists publications about trajectory tracking using neural networks ([3], [14], [15], [17], [18]); 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 [2] [4]. 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, adaptive backstepping constitutes a major design methodology [9]. The idea behind backstepping design is that some appropriate functions of state variables are selected recursively as virtual control inputs for lower dimension subsystems of the overall system [12]. Each backstepping stage results in a new virtual control designs from the preceding design stages. When the procedure ends, a feedback design for the true control input results, which achieves the original design objective. The backstepping technique provides a systematic framework for the design of tracking and regulation strategies, suitable for a large class of state feedback linearizable nonlinear systems [9].

In this paper, first the block strict feedback decomposition is applied in order to define a number of sub-problems of lower order. Once this decomposition is achieved, then the backstepping technique is used to design a suitable controller [9]. Afterwards, this resulting controller is approximated by a High Order Neural Network (HONN) [4]. The implementation is simple and the training is performed on-line by means of an extended Kalman Filter (EKF) [11], [16].

The method presented here has some advantages; the first one is the application to MIMO nonlinear systems in discrete-time; the second one is to guarantee the boundeness of the error in presence of disturbances and the third one is that the design can be applied to different systems. Finally, this paper also proposes the use of High Order Neural Networks (HONN), trained by EKF to approximate the control law designed by the backstepping-method [4].

The paper is organized as follows: first, the neural network model to be used is presented; then the Kalman filtering learning algorithm is discussed, followed by the controller design. To this end, the applicability of the proposed approach is illustrated by its utilization to trayectory tracking for electric induction motors, represented by discrete-time MIMO nonlinear models.

II. Mathematical preliminaries

A. Stability Definitions

This section close follows [4]. 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:

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

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 input. 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

$$u(k) = \xi(\chi(k)) \tag{2}$$

Such substitution eliminates u:

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

and yields an unforced system (3)[7].

Definition 2: The solution of (1) $-$ (2) 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 (1) 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 (2), 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 Let $V(\chi(k))$ be a Lyapunov function for a discrete-time system (1), 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 3: A subset $S \in \Re^n$ is bounded if there is $r > 0$ such that $\|\chi\| \leq r$ for all $\chi \in S$ [7].

Definition 4: The system (4) is said to be BIBO stable if for a bounded input $u(k)$, the system produces a bounded output $y(k)$ for $0 < k < \infty$ [7].

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

$$\begin{aligned}
\chi(k+1) &= A(k)\chi(k) + Bu(k) \\
y(k) &= C\chi(k)
\end{aligned} \tag{4}$$

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 (4), i.e. $\Phi(k(1), k(0)) = \prod_{k=k(0)}^{k=k_1-1} A(k)$. If $\|\Phi(k(1), k(0))\| < 1 \ \ \forall k(1) > k(0) > 0$, then the system (4) is

1) globally exponentially stable for the unforced system and

2) Bounded Input-Bounded Output (BIBO) stable [4], [19].

B. Discrete-time High Order Neural Networks

Consider the HONN described by:

$$\begin{aligned}
\phi(w, z) &= w^\top S(z) \\
S(z) &= \left[s_1^\top(z), s_2^\top(z), \cdots, s_m^\top(z)\right] \qquad\qquad i = 1, 2, \cdots, L \\
s_i(z) &= \left[\begin{array}{ccc} \prod\limits_{j \in I_1} [s(z_j)]^{d_j(i_1)} & \cdots & \prod\limits_{j \in I_1} [s(z_j)]^{d_j(i_m)} \end{array}\right]^\top
\end{aligned} \tag{5}$$

where $z = [z_1, z_2, \cdots, z_q]^\top \in \Omega_z \subset \Re^q$ are positive integer, q denotes the number of external inputs, L denotes the NN node number, $\phi \in \Re^m$, $\{I_1, I_2, \cdots, I_L\}$ is a collection of not-ordered subsets of $\{1, 2, \cdots, q\}$, $S(z) \in \Re^{L\times m}$, $d_j(i_j)$ are nonnegative integer, $w \in \Re^L$ is an adjustable synaptic weight vector, and $s(z_j)$ is chosen as the hyperbolic tangent function:

$$s(z_j) = \frac{e^{z_j} - e^{-z_j}}{e^{z_j} + e^{-z_j}} \tag{6}$$

For a desired function $u^*(z) \in \Re^m$,assume, there exist ideal weight vector $w^* \in \Re^L$ such that the smooth function vector $u^*(z)$ can be approximated by an ideal NN on a compact set $\Omega_z \subset \Re^q$

$$u^*(z) = w^{*\top} S(z) + \epsilon_z \tag{7}$$

where $\epsilon_z \in \Re^m$ is the bounded NN approximation error vector [1]; note that $\|\epsilon_z\|$ can be reduced by increasing the number of the adjustable weights. The ideal weight vector w^* is an artificial quantity required only for analytical purpose [4], [18]. In general, it is assumed that there exists unknown but constant weight vector w^*, whose estimate is $w \in \Re^L$. Hence it is possible to define:

$$\widetilde{w}(k) = w(k) - w^* \tag{8}$$

as the estimation error.

C. 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 [6], [20]. 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 [16] and references therein).

The training goal is to find the optimal weight values which minimize the prediction error. The EKF-based training algorithm is described by [6]:

$$\begin{aligned} K(k) &= P(k) H(k) \left[R(k) + H^\top(k) P(k) H(k)\right]^{-1} \\ w(k+1) &= w(k) + \eta K(k) \left[y(k) - \widehat{y}(k)\right] \\ P(k+1) &= P(k) - K(k) H^\top(k) P(k) + Q(k) \end{aligned} \tag{9}$$

where $P \in \Re^{L\times L}$ is the prediction error covariance matrix, $w \in \Re^L$ is the weight (state) vector, L is the total number of neural network weights, $y \in \Re^m$ is the measured output vector, $\widehat{y} \in \Re^{m}$ is the network output, η is a design parameter, $K \in \Re^{L\times m}$ is the Kalman gain matrix, $Q \in \Re^{L\times L}$ is the state noise covariance matrix, $R \in \Re^{m\times m}$ is the measurement noise covariance matrix, $H \in \Re^{L\times m}$ is a matrix, for which each entry (H_{ij}) is the derivative of one of the neural network output, $(\widehat{y}_i)$, with respect to one neural network weight, (w_j), as follows:

$$H_{ij}(k) = \left[\frac{\partial \widehat{y}_i(k)}{\partial w_j(k)}\right]_{w(k)=\widehat{w}(k+1)}, \quad i = 1, ..., m \text{ and } \; j = 1, ..., L \tag{10}$$

Usually P, Q and R are initialized as diagonal matrices, with entries $P(0)$, $Q(0)$ and $R(0)$, respectively. It is important that $H(k)$, $K(k)$ and $P(k)$ for the EKF are bounded [20]. Therefore, there exist constants $\overline{H} > 0$, $\overline{K} > 0$ and $\overline{P} > 0$ such that:

$$\begin{aligned} \|H(k)\| &\leq \overline{H} \\ \|K(k)\| &\leq \overline{K} \\ \|P(k)\| &\leq \overline{P} \end{aligned} \tag{11}$$

This bounds would be used for controller design which will be discussed in section III.

III. Controller design

The model of many practical nonlinear systems can be expressed in (or transformed into) a special state-space form named, block strict feedback form (BSFF) [9], as follows:

$$\begin{aligned} x^i(k+1) &= f^i\left(\overline{x}^i(k)\right) + g^i\left(\overline{x}^i(k)\right)x^{i+1}(k) + d^i(k), \qquad i = 1, 2, \cdots, r-1 \\ x^r(k+1) &= f^r(X(k)) + g^r(X(k))\,u(k) + d^r(k) \\ y(k) &= x^1(k) \end{aligned} \tag{12}$$

where $X(k) = \left[x^{1\top}(k), \cdots, x^{r\top}(k)\right]^\top$ are the state variables, $\overline{x}^i(k) = \left[x^{1\top}, x^{2\top}, \cdots, x^{i\top}\right]^\top$, $x^i \in \Re^{n_i}$, $r \geq 2$, r is the number of blocks, $u(k) \in \Re^m$ are the system inputs, $y(k) \in \Re^m$ is the system output, $d^i \in \Re^{n_i}$ is the bounded unknown disturbance vector, $f^i(\bullet)$ and $g^i(\bullet)$ are unknown smooth nonlinear functions and the set of numbers $(n_1, \cdots, n_r)$ define the system structure as

$$rang\,(g_i) = n_i$$

and

$$n_1 \leq n_2 \leq \cdots \leq n_r \leq m$$

For simplicity, we consider the structure $n_1 = n_2 = \cdots = n_r = m$ moreover there exists a constant $\overline{d}_i$ such that $\|d_i(k)\| \leq \overline{d}_i$, for $0 < k < \infty$.and as in [21], in this paper the plant (12) is assumed to be BIBO.

If we consider the original system (12) as a one-step ahead predictor, then we can transform it into an equivalent maximum r-step ahead one, which can predict the future states $x^1(k+r), x^2(k+r-1), \cdots, x^r(k+1)$, then the causality contradiction is avoided when the controller is constructed based on the maximum r-step ahead prediction by backstepping [2], [4]:

$$\begin{aligned}
x^1(k+r) &= F^1\left(\bar{x}^1(k)\right) + G^1\left(\bar{x}^1(k)\right)x^2(k+r-1) + d^1(k+r)\\
&\vdots\\
x^{r-1}(k+2) &= F^{r-1}\left(\bar{x}^{r-1}(k)\right) + G^{r-1}\left(\bar{x}^{r-1}(k)\right)x^r(k+1) + d^{r-1}(k+2)\\
x^r(k+1) &= f^r(X(k)) + g^r(X(k))u(k) + d^r(k)\\
y(k) &= x^1(k)
\end{aligned} \tag{13}$$

where $F^i(\bullet)$ and $G^i(\bullet)$ are unknown functions of $f^i\left(\overline{x}^i(k)\right)$ and $g^i\left(\overline{x}^i(k)\right)$, respectively. For convenience of analysis, let us define $1 \leq i \leq r-1$

$$\begin{aligned}
F^i(k) &\triangleq F^i(\overline{x}_i(k))\\
G^i(k) &\triangleq G^i(\overline{x}_i(k))\\
f^r(k) &\triangleq f^r(X(k))\\
G^r(k) &\triangleq g^r(X(k))
\end{aligned}$$

Then, system (13) can be written as

$$\begin{aligned}
x^1(k+r) &= F^1(k) + G^1(k)x^2(k+r-1) + d^1(k+r)\\
&\vdots\\
x^{r-1}(k+2) &= F^{r-1}(k) + G^{r-1}(k)x^r(k+1) + d^{r-1}(k+2)\\
x^r(k+1) &= f^r(k) + g^r(k)u(k) + d^r(k)\\
y(k) &= x_1(k)
\end{aligned} \tag{14}$$

The objective is to design a control $u(k)$ to make the system output $y(k)$ follow a desired trayectory $y_d(k)$. Once (14) is defined, we apply the well known backstepping technique [9]. For system (14), we can define the desired virtual controls $\left(\alpha^{j*}(k), \quad j=1,\cdots,r-1\right)$ and the ideal practical control $(u^*(k))$ as follows:

$$\begin{aligned}
\alpha^{1*}(k) &\triangleq x^2(k) = \varphi^1\left(\overline{x}^1(k), y_d(k+r)\right) \\
\alpha^{2*}(k) &\triangleq x^3(k) = \varphi^2\left(\overline{x}^2(k), \alpha^{1*}(k)\right) \\
&\vdots \\
\alpha^{r-1*}(k) &\triangleq x^r(k) = \varphi^{r-1}\left(\overline{x}^{r-1}(k), \alpha^{r-2*}(k)\right) \\
u^*(k) &= \varphi^r\left(X(k), \alpha^{r-1*}(k)\right) \\
y(k) &= x^1(k)
\end{aligned} \tag{15}$$

where φ^j $(1 \leq j \leq r)$ are nonlinear smooth functions. It is obvious that the desired virtual controls $\alpha^{i*}(k)$ and the ideal control $u^*(k)$ will drive the output $y(k)$ to track the desired signal $y_d(k)$ only if the exact system model is known and there are no unknown disturbances. However in practical applications these two conditions cannot be satisfied. In the following, neural networks will be used to approximate the desired virtual controls, as well as the desired practical controls, when the conditions established above are not satisfied. As in [4], we construct the virtual and practical controls via embedded backstepping without the causality contradiction [2]. Let us approximate the virtual controls and practical control by the following HONN $(1 \leq i \leq r-1)$:

$$\begin{aligned}
\alpha^i(k) &= w^{i^\top} S^i\left(z^i(k)\right) \\
u(k) &= w^{r^\top} S^r\left(z^r(k)\right), \qquad i=1,\cdots,r-1
\end{aligned} \tag{16}$$

with

$$\begin{aligned}
z^1(k) &= \left[x^1(k), y_d(k+r)\right]^\top \\
z^i(k) &= \left[\overline{x}^i(k), \alpha^{i-1}(k)\right]^\top, \qquad i=1,\cdots,r-1 \\
z^r(k) &= \left[X(k), \alpha^{r-1}(k)\right]^\top
\end{aligned}$$

where $w^j \in \Re^{L_j}$ are the estimates of ideal constant weights w^{j*} $(1 \leq j \leq r)$ and $S^j \in \Re^{L_j \times n_j}$ Define the estimation error as:

$$\widetilde{w}^j(k) = w^j(k) - w^{j*} \tag{17}$$

Then the corresponding weights updating laws are defined of the form

$$w^j(k+1) = w^j(k) + \eta^j K^j(k) e^j(k) \tag{18}$$

with

$$\begin{aligned} K^j(k) &= P^j(k) H^j(k) M^{j^{-1}}(k) \\ M^j(k) &= R^j(k) + H^{j^{\top}}(k) P^j(k) H^j(k) \\ P^j(k+1) &= P^j(k) - K^j(k) H^{j^{\top}}(k) P^j(k) + Q^j(k) \qquad (19) \\ H^j(k) &= \left[\frac{\partial \widehat{\upsilon}^j(k)}{\partial w^j(k)} \right] \qquad (20) \end{aligned}$$

and

$$e^j(k) = \upsilon^j(k) - \widehat{\upsilon}^j(k) \tag{21}$$

where $\upsilon^i(k) \in \Re^{n_i}$ is the desired signal and $\widehat{\upsilon}^i(k) \in \Re^{n_i}$ is the HONN function approximation defined, respectively, as follows

$$\begin{aligned} \upsilon^1(k) &= y_d(k) \\ \upsilon^2(k) &= x^2(k) \\ &\vdots \\ \upsilon^r(k) &= x^r(k) \end{aligned} \tag{22}$$

and

$$\begin{aligned} \widehat{\upsilon}^1(k) &= y(k) \\ \widehat{\upsilon}^2(k) &= \alpha^1(k) \\ &\vdots \\ \widehat{\upsilon}^r(k) &= \alpha^{r-1}(k) \end{aligned} \tag{23}$$

$e^j(k)$ denotes the error at each step, as

$$\begin{aligned} e^1(k) &= y_d(k) - y(k) \\ e^2(k) &= x^2(k) - \alpha^1(k) \\ &\vdots \\ e^r(k) &= x^r(k) - \alpha^{r-1}(k) \end{aligned} \tag{24}$$

Besides, it is worth to include the following remarks.

Remark 1. The NN approximation error vector ϵ_z is bounded. This is a well known neural network property [1].

Remark 2. The gain matrix of the EKF $(K(k))$ is bounded by a constant $\overline{K} > 0$, that is $\|K(k)\| \leq \overline{K}$

Before proceeding to demonstrate the main result of this paper, we need to establish the following two lemmas.

Lemma 2. (21) can be formulated as

$$e^j(k+1) = e^j(k) + \Delta e^j(k) \quad (1 \leq j \leq r) \tag{25}$$

with $\Delta e^j(k) \leq -\gamma^j e^j(k)$ and $\gamma^j = \max \left\| H^{j^\top}(k) \eta^j K^j(k) \right\|$.

Proof: From (21) and consider that $\upsilon(k)$ do not depend on the HONN parameters, we obtain

$$\frac{\partial e^i(k)}{\partial w^i(k)} = -\frac{\partial \widehat{\upsilon}(k)}{\partial w^i(k)} \tag{26}$$

Let us approximate (26) by

$$\Delta e^i(k) = \left[\frac{\partial e^i(k)}{\partial w^i(k)} \right]^\top \Delta w^i(k) \tag{27}$$

Substituting (20) and (26) in (27), yields

$$\Delta e^i(k) = -H^{i^\top}(k) \eta^i K^i(k) e^i(k) \tag{28}$$

Defining

$$\gamma^i = \max \left\| H^{i^\top}(k) \eta^i K^i(k) \right\|$$

then

$$\Delta e^i(k) \leq -\gamma^i e^i(k) \tag{29}$$

■

Lemma 3. The HONN weights updated with (18), based on the EKF algorithm (19), are bounded.

Proof: (18) can be written as

$$
\begin{aligned}
\widetilde{w}^{i^{\top}}(k+1) &= \widetilde{w}^{i^{\top}}(k) - \eta^i K^{i^{\top}}(k)\, S\left(z^i(k)\right) \widetilde{w}^{i^{\top}}(k) + \eta^i K^{i^{\top}}(k)\, \epsilon'_{z^i} \\
&= A^i(k)\, \widetilde{w}^{i^{\top}}(k) + B^i v_{z^i}(k) \\
i &= 1, \cdots, r
\end{aligned}
$$

with

$$
\begin{aligned}
A^i(k) &= \left[I - \eta^i K^{i^{\top}}(k)\, S\left(z^i(k)\right)\right] \\
B^i &= \eta^i \\
v_{z^i}(k) &= K^{i^{\top}}(k)\, \epsilon'_{z^i}(k) \\
\epsilon'_{z^i}(k) &= \epsilon_{z^i} + d^i(k) \\
i &= 1, \cdots, r
\end{aligned} \tag{30}
$$

Considering remark 1 and remark 2, ϵ_{z^i} and, $S\left(z^i(k)\right)$ are bounded. Hence selecting η^i appropriately $A^i(k)$ satisfies $\|\Phi(k(1), k(0))\| < 1$. By applying Lemma 1, $\widetilde{w}^i(k)$ is bounded. ∎

Considering $(12) - (24)$, we establish the main result in the following theorem.

Theorem 2: For the system (12), the HONN (16) trained with the EKF-based algorithm (19) to approximate the control law (15), ensures that the tracking error (24) is semiglobally uniformly ultimately bounded (SGUUB); moreover, the HONN weights remain bounded.

Proof: For the first block of system(12), with the virtual control $\alpha^{1*}(k)$ approximated by the HONN $\left(\alpha^1(k) = w^1 S^1\left(z^1(k)\right)\right)$ and $e^1(k)$ defined as in (24), consider the Lyapunov function candidate

$$
V^1(k) = e^{1^{\top}}(k)\, e^1(k) + \widetilde{w}^{1^{\top}}(k)\, \widetilde{w}^1(k) \tag{31}
$$

whose first difference is

$$
\begin{aligned}
\Delta V^1(k) &= V^1(k+1) - V^1(k) \\
&= e^{1^{\top}}(k+1)\, e^1(k+1) + \widetilde{w}^{1^{\top}}(k+1)\, \widetilde{w}^1(k+1) - e^{1^{\top}}(k)\, e^1(k) - \widetilde{w}^{1^{\top}}(k)\, \widetilde{w}^1(k)
\end{aligned} \tag{32}
$$

From (17) and (18), then

$$
\widetilde{w}^1(k+1) = \widetilde{w}^1(k) + \eta^1 K^1(k)\, e^1(k) \tag{33}
$$

Let us define

$$
\begin{aligned}
& \left[\widetilde{w}^{1}(k)+\eta^{1} K^{1}(k) e^{1}(k)\right]^{\top} \times\left[\widetilde{w}^{1}(k)+\eta^{1} K^{1}(k) e^{1}(k)\right] \\
= \;& \widetilde{w}^{1^{\top}}(k) \widetilde{w}^{1}(k)+2 \widetilde{w}^{1^{\top}}(k) \eta^{1} K^{1}(k) e^{1}(k)+\left(\eta^{1} K^{1}(k) e^{1}(k)\right)^{\top} \eta^{1} K^{1}(k) e^{1}(k)
\end{aligned} \tag{34}
$$

From (24), then

$$
\begin{aligned}
e^{1}(k+1) &= e^{1}(k)+\Delta e^{1}(k) \\
e^{1^{\top}}(k+1) e^{1}(k+1) &= e^{1^{\top}}(k) e^{1}(k)+e^{1^{\top}}(k) \Delta e^{1}(k)+\Delta e^{1^{\top}}(k) e^{1}(k)+\Delta e^{1^{\top}}(k) \Delta e^{1}(k) \\
e^{1^{\top}}(k+1) e^{1}(k+1)-e^{1^{\top}}(k) e^{1}(k) &= e^{1^{\top}}(k) \Delta e^{1}(k)+\Delta e^{1^{\top}}(k) e^{1}(k)+\Delta e^{1^{\top}}(k) \Delta e^{1}(k)
\end{aligned}
$$

where $\Delta e^{1}(k)$ is the error difference. Substituting (33) and (34) in (32) results in

$$
\begin{aligned}
\Delta V^{1}(k) = \;& e^{1^{\top}}(k) \Delta e^{1}(k)+\Delta e^{1^{\top}}(k) e^{1}(k)+\Delta e^{1^{\top}}(k) \Delta e^{1}(k) \\
& +2 \widetilde{w}^{1^{\top}}(k) \eta^{1} K^{1}(k) e^{1}(k)+\left(\eta^{1} K^{1}(k) e^{1}(k)\right)^{\top} \eta^{1} K^{1}(k) e^{1}(k)
\end{aligned} \tag{35}
$$

From Lemma 2, substituting (29) , we obtain

$$
\begin{aligned}
\Delta V^{1}(k) \leq \;& -2 \gamma^{1} e^{1^{\top}}(k) e^{1}(k)+\gamma^{1^{2}} e^{1^{\top}}(k) e^{1}(k)+2 \widetilde{w}^{1^{\top}}(k) \eta^{1} K^{1}(k) e^{1}(k) \\
& +\left(\eta^{1} K^{1}(k) e^{1}(k)\right)^{\top} \eta^{1} K^{1}(k) e^{1}(k) \\
\leq \;& -2 \gamma^{1}\left\|e^{1}(k)\right\|^{2}+\gamma^{1^{2}}\left\|e^{1}(k)\right\|^{2}+2\left\|\widetilde{w}^{1^{\top}}(k) \eta^{1} K^{1}(k)\right\|\left\|e^{1}(k)\right\| \\
& +\left\|\eta^{1} K^{1}(k)\right\|^{2}\left\|e^{1}(k)\right\|^{2}
\end{aligned} \tag{36}
$$

where $\gamma^{1}=\max \left\|H^{1^{\top}}(k) \eta^{1} K^{1}(k)\right\|$. From Lemma 3, it follows that $\widetilde{w}^{1}(k)$ is bounded; then there is $\eta^{1}>0$ such that:

$$
\Delta V^{1}(k) \leq 0, \quad \text{once} \quad \left|e^{1}(k)\right|>\kappa^{1} \tag{37}
$$

with κ^{1} defined as

$$
\kappa^{1}=\frac{2 \eta^{1} \overline{w}^{1^{\top}} \overline{K}^{1}}{2 \gamma^{1}-\gamma^{1^{2}}-\eta^{1^{2}} \overline{K}^{1^{2}}}
$$

where $\overline{w}^{1}$ is the upper bound of $\widetilde{w}^{1}$ and $\overline{K}^{1}$ is the upper bound of $K^{1}(k)$ [20]. From (37) it follows the boundness of $V^{1}(k)$ for $k \geq k_{T}$, that leads to the SGUUB of $e^{1}(k)$.

For the following i-th $(i = 2, \cdots, r-1)$ equation of the system (12), with the virtual control $\alpha^{i*}(k)$ approximated by the HONN $\alpha^i(k) = w^i S^i\left(z^i(k)\right)$ and $e^i(k)$ defined in (24), consider the Lyapunov function candidate.

$$V^i(k) = e^{i^\top}(k)\, e^i(k) + \widetilde{w}^{i^\top}(k)\, \widetilde{w}^i(k) \tag{38}$$

whose first difference is

$$\begin{aligned}\Delta V^i(k) &= V^i(k+1) - V^i(k) \\ &= e^{i^\top}(k+1)\, e^i(k+1) + \widetilde{w}^{i^\top}(k+1)\, \widetilde{w}^i(k+1) - e^{i^\top}(k)\, e^i(k) - \widetilde{w}^{i^\top}(k)\, \widetilde{w}^i(k)\end{aligned} \tag{39}$$

From (17) and (18), then

$$\widetilde{w}^i(k+1) = \widetilde{w}^i(k) + \eta^i K^i(k)\, e^i(k) \tag{40}$$

Let us define

$$\begin{aligned}&\left[\widetilde{w}^i(k) + \eta^i K^i(k)\, e^i(k)\right]^\top \times \left[\widetilde{w}^i(k) + \eta^i K^i(k)\, e^i(k)\right] \\ =\ & \widetilde{w}^{i^\top}(k)\, \widetilde{w}^i(k) + 2\widetilde{w}^{i^\top}(k)\, \eta^i K^i(k)\, e^i(k) + \left(\eta^i K^i(k)\, e^i(k)\right)^\top \eta^i K^i(k)\, e^i(k)\end{aligned} \tag{41}$$

From (24), then

$$\begin{aligned}e^i(k+1) &= e^i(k) + \Delta e^i(k) \\ e^{i^\top}(k+1)\, e^i(k+1) &= e^{i^\top}(k)\, e^i(k) + e^{i^\top}(k)\, \Delta e^i(k) + \Delta e^{i^\top}(k)\, e^i(k) + \Delta e^{i^\top}(k)\, \Delta e^i(k) \\ e^{i^\top}(k+1)\, e^i(k+1) - e^{i^\top}(k)\, e^i(k) &= e^{i^\top}(k)\, \Delta e^i(k) + \Delta e^{i^\top}(k)\, e^i(k) + \Delta e^{i^\top}(k)\, \Delta e^i(k)\end{aligned}$$

where $\Delta e^i(k)$ is the error difference. Substituting (40) and (41) in (39) results in

$$\begin{aligned}\Delta V^i(k) &= e^{i^\top}(k)\, \Delta e^i(k) + \Delta e^{i^\top}(k)\, e^i(k) + \Delta e^{i^\top}(k)\, \Delta e^i(k) \\ &\quad + 2\widetilde{w}^{i^\top}(k)\, \eta^i K^i(k)\, e^i(k) + \left(\eta^i K^i(k)\, e^i(k)\right)^\top \eta^i K^i(k)\, e^i(k)\end{aligned} \tag{42}$$

From Lemma 2, substituting (29), we obtain

$$\begin{aligned}\Delta V^i(k) &\le -2\gamma^i e^{i^\top}(k)\, e^i(k) + \gamma^{i^2} e^{i^\top}(k)\, e^i(k) + 2\widetilde{w}^{i^\top}(k)\, \eta^i K^i(k)\, e^i(k) \\ &\quad + \left(\eta^i K^i(k)\, e^i(k)\right)^\top \eta^i K^i(k)\, e^i(k) \\ &\le -2\gamma^i \left\|e^i(k)\right\|^2 + \gamma^{i^2} \left\|e^i(k)\right\|^2 + 2\left\|\widetilde{w}^{i^\top}(k)\, \eta^i K^i(k)\right\| \left\|e^i(k)\right\| \\ &\quad + \left\|\eta^i K^i(k)\right\|^2 \left\|e^i(k)\right\|^2\end{aligned} \tag{43}$$

where $\gamma^i = \max \left\| H^{i^\top}(k)\, \eta^i K^i(k) \right\|$. From Lemma 3, it follows that $\widetilde{w}^i(k)$ is bounded; then there is $\eta^i > 0$ such that:

$$\Delta V^i(k) \leq 0, \quad \text{once} \quad \left| e^i(k) \right| > \kappa^i \tag{44}$$

with κ^i defined as

$$\kappa^i = \frac{2\eta^i \overline{w}^{i^\top} \overline{K}^i}{2\gamma^i - \gamma^{i2} - \eta^{i2} \overline{K}^{i2}}$$

where $\overline{w}^i$ is the upper bound of $\widetilde{w}^i$ and $\overline{K}^i$ is the upper bound of $K^i(k)$ [20]. From (44) it follows the boundness of $V^i(k)$ for $k \geq k_T$, that leads to the SGUUB of $e^i(k)$. ■

IV. Applications

In this section, we apply the above developed scheme to control a three-phase induction motor. Induction motor is one of the most used actuators for industrial applications due to its reliability, ruggedness and relatively low cost. The control of the induction motor is challenging, since its dynamics is described by multivariable, coupled, and highly nonlinear system [13]. Early works on control of induction motors was focused on the field oriented control (FOC) [8], exact input-output linealization, adaptive input output linearization, direct torque control (DTC) [8]; however most of those works were developed for continuous-time model of the motor. In [13] it was proposed discrete-time model and control algorithm assuming that the parameters and load torque of the motor model are known. Now we consider the control problem assuming that the it is only known the rotor flux dynamics.

A. 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 [13]

$$
\begin{aligned}
\omega(k+1) &= \omega(k) + \frac{\mu}{\alpha}(1-\alpha) \times M\left(i^{\beta}(k)\psi^{\alpha}(k) - i^{\alpha}(k)\psi^{\beta}(k)\right) - \left(\frac{T}{J}\right)T_L(k) \\
\psi^{\alpha}(k+1) &= \cos(n_p\theta(k+1))\rho_1(k) - \sin(n_p\theta(k+1))\rho_2(k) \\
\psi^{\beta}(k+1) &= \sin(n_p\theta(k+1))\rho_1(k) + \cos(n_p\theta(k+1))\rho_2(k) \\
i^{\alpha}(k+1) &= \varphi^{\alpha}(k) + \frac{T}{\sigma}u^{\alpha}(k) + d_1(k) \\
i^{\beta}(k+1) &= \varphi^{\beta}(k) + \frac{T}{\sigma}u^{\beta}(k) + d_2(k) \\
\theta(k+1) &= \theta(k) + \omega(k)T + \frac{\mu}{\alpha}\left[T - \frac{(1-a)}{\alpha}\right] \times M\left(i^{\beta}(k)\psi^{\alpha}(k) - i^{\alpha}(k)\psi^{\beta}(k)\right) - \frac{T_L(k)}{J}T^2 \qquad (45)
\end{aligned}
$$

with

$$
\begin{aligned}
\rho_1(k) = a\left(\cos(\phi(k))\psi^{\alpha}(k) + \sin(\phi(k))\psi^{\beta}(k)\right) + b\left(\cos(\phi(k))i^{\alpha}(k) + \sin(\phi(k))i^{\beta}(k)\right) \\
\rho_2(k) = a\left(\cos(\phi(k))\psi^{\alpha}(k) - \sin(\phi(k))\psi^{\beta}(k)\right) + b\left(\cos(\phi(k))i^{\alpha}(k) - \sin(\phi(k))i^{\beta}(k)\right)
\end{aligned}
$$

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

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

B. Block-Strict-Feedback-Form (BSFF) for the induction motor

Define the following states:

$$
\begin{aligned}
x^1(k) &= \begin{bmatrix} \omega(k) \\ \Psi(k) \end{bmatrix}; \quad x^2(k) = \begin{bmatrix} i^{\alpha}(k) \\ i^{\beta}(k) \end{bmatrix} \\
u(k) &= \begin{bmatrix} u^{\alpha}(k) \\ u^{\beta}(k) \end{bmatrix}; \quad y_d(k) = \begin{bmatrix} \omega_d(k) \\ \Psi_d(k) \end{bmatrix} \\
y(k) &= x^1(k) \qquad (46)
\end{aligned}
$$

where $\Psi(k) = \psi^{\alpha^2}(k) + \psi^{\beta^2}(k)$ is the rotor flux magnitude, $\omega_d(k)$ and $\Psi_d(k)$ are the reference signals. The objective of control is to drives the output $y(k)$ to track the reference $y_d(k)$. Using (46) the system (45), can be represented in the BSFF consisting of two blocks

$$\begin{aligned} x^1(k+1) &= f^1\left(x^1(k)\right) + g^1\left(x^1(k)\right)x^2(k) + d^1(k) \\ x^2(k+1) &= f^2\left(\overline{x}^2(k)\right) + g^2\left(\overline{x}^2(k)\right)u(k) \end{aligned}$$

where $f^1\left(x^1(k)\right), g^1(x_1(k))$, $f_2(\overline{x}_2(k))$ and $g_2(\overline{x}_2(k))$ are assumed to be unknown and $d_1(k)$ is the unknown bounded disturbances; in this case this disturbance is the load torque. Now we use the HONN to approximate the desired virtual controls and the ideal practical controls described as

$$\begin{aligned} \alpha^{1*}(k) &\triangleq x^2(k) = \varphi^1\left(x^1(k), y_d(k+2)\right) \\ u^*(k) &= \varphi^2\left(x^1(k), x^2(k), \alpha^{1*}(k)\right) \\ y(k) &= x^1(k) \end{aligned} \tag{47}$$

The HONN proposed for this application is as follows:

$$\begin{aligned} \alpha^1(k) &= w^{1^\top}S^1\left(z^1(k)\right) \\ u(k) &= w^{2^\top}S^2\left(z^2(k)\right) \end{aligned} \tag{48}$$

with

$$\begin{aligned} z^1(k) &= \left[x^1(k), y_d(k+2)\right]^\top \\ z^2(k) &= \left[x^1(k), x^2(k), \alpha^1(k)\right]^\top \end{aligned}$$

The weights are updated using the EKF as follows:

$$\begin{aligned} w^i(k+1) &= w^i(k) + \eta^i K^i(k)e^i(k) \qquad (1 \leq i \leq 2) \\ K^i(k) &= P^i(k)H^i(k)\left[R^i(k) + H^{i^\top}(k)P^i(k)H^i(k)\right]^{-1} \\ P^i(k+1) &= P^i(k) - K^i(k)H^{i^\top}(k)P^i(k) + Q^i(k) \end{aligned} \tag{49}$$

with

$$\begin{aligned} e^1(k) &= y_d(k) - y(k) \\ e^2(k) &= x^2(k) - \alpha^1(k) \end{aligned}$$

The training is performed on-line, using a series-parallel configuration. All the NN states are initialized in a random way. The covariances matrices are initialized as diagonals, and the nonzero elements are: $P_1(0) = P_2(0) = 10000$; $Q_1(0) = Q_2(0) = 5000$ and $R_1(0) = R_2(0) = 10000$, respectively. The simulation is performing under the presence of the disturbances $d^1(k)$ as shown in Fig. 3 and parametric variations (Fig. 8).

C. Reduced order nonlinear observer

The last control algorithm works with the full state and parameters measurement assumption [13]. However, the rotor fluxes measurement is a difficult task. Here, a reduced order nonlinear observer is designed for fluxes with rotor speed and currents measurements only. The flux dynamics in (45) can be written as

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

with:

$$\begin{aligned} G(k) &= \begin{bmatrix} \cos(n_p T\omega(k)) & -\sin(n_p T\omega(k)) \\ \sin(n_p T\omega(k)) & \cos(n_p T\omega(k)) \end{bmatrix} \\ \mathbf{I}(k) &= \begin{bmatrix} i^\alpha(k) \\ i^\beta(k) \end{bmatrix} \end{aligned} \tag{50}$$

The proposed observer for the system (45), assumes speed and current measurements, and an unknown load.

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

Let us define

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

Then:

$$e^\Psi(k+1) = aG(k)e^\Psi(k) \tag{51}$$

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

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

with

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) = e^{\Psi^\top}(k-1)e^\Psi(k+1) - e^{\Psi^\top}(k)e^\Psi(k) \\ &= e^{\Psi^\top}(k)\left(a^2G^\top(k)G(k) - I\right)e^\Psi(k) \end{aligned} \tag{52}$$

where

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

By (50), $G^\top(k)\, G(k) = I$ 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. So the increment of the Lyapunov function (52) 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 developed above.

D. Simulation Results

The simulation is performing using the system (45) with the follow 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 performance can be verified for the two plant outputs. The Fig. 3 display the load torque applied as an external disturbance. Fig. 4 displays a parametric variation introduced in the rotor resistance (R_r) as an increment. Fig. 5 shows the weight evolution. Fig. 6 and Fig. 7 displays the control law signals. Fig. 8 and Fig. 9 portrays the fluxes and their estimates.

Comment 1. The purpose of this paper is to improve the tracking performance for a class of MIMO nonlinear systems in discrete-time, by means of the use of the EKF as the neural network learning algorithm; this approach

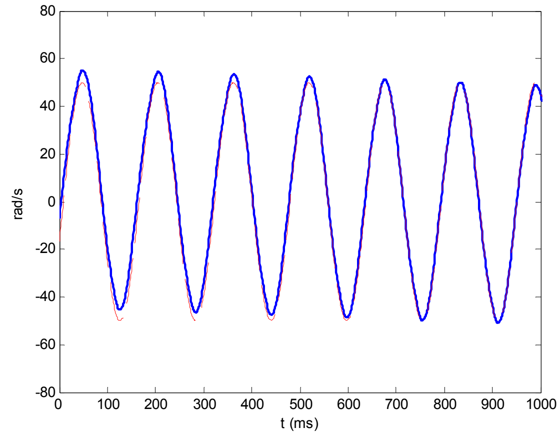

Fig. 1. Tracking performance $\omega(k)$ (solid line) and $\omega_d(k)$ (dashed line).

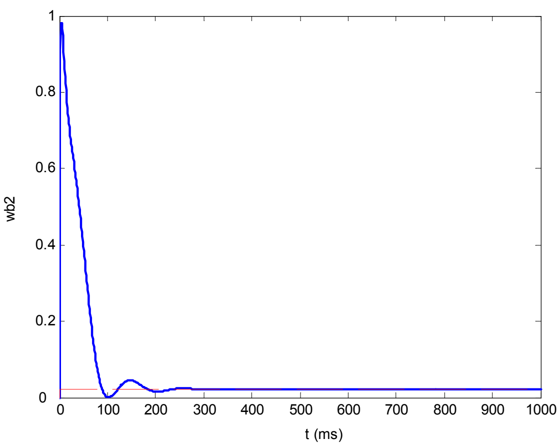

Fig. 2. Tracking performance $\Psi(k)$ (solid line and $\Psi_d(k)$ (dashed line).

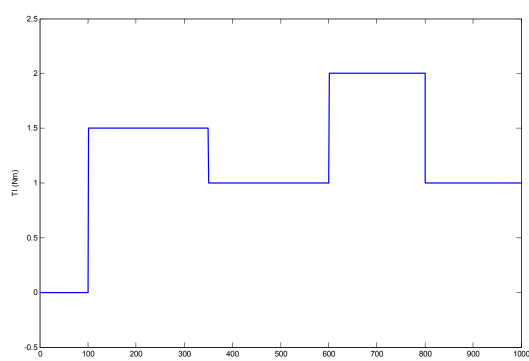

Fig. 3. Load torque $T_L(k)$

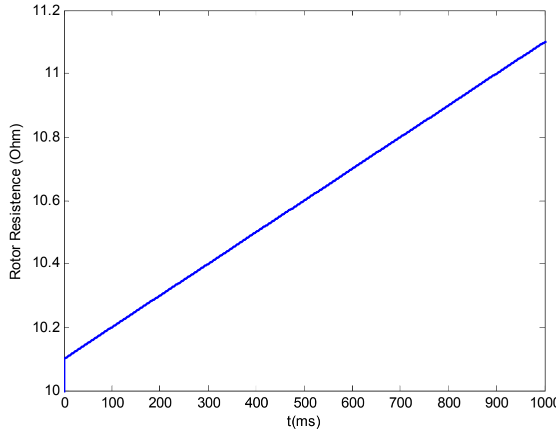

Fig. 4. Rotor resistance variation (R_r)

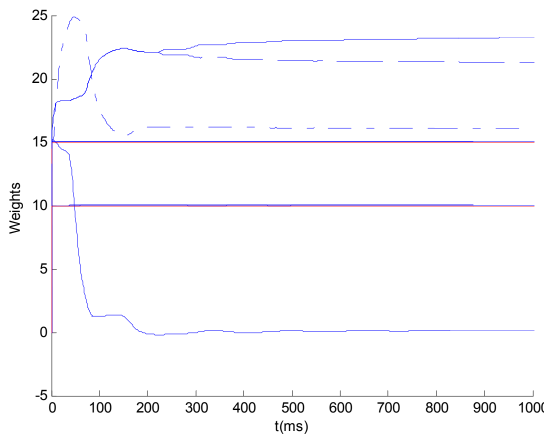

Fig. 5. Weight evolution

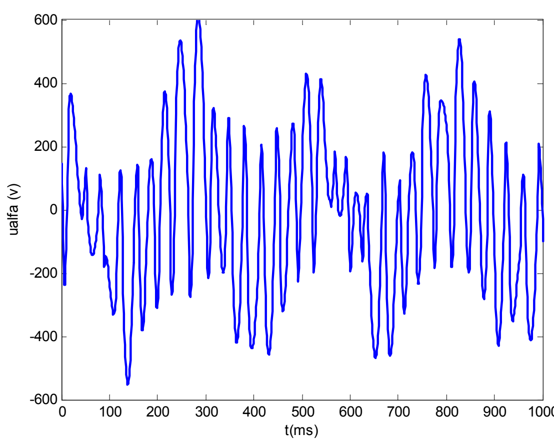

Fig. 6. Control law signal $u^{\alpha}(k)$

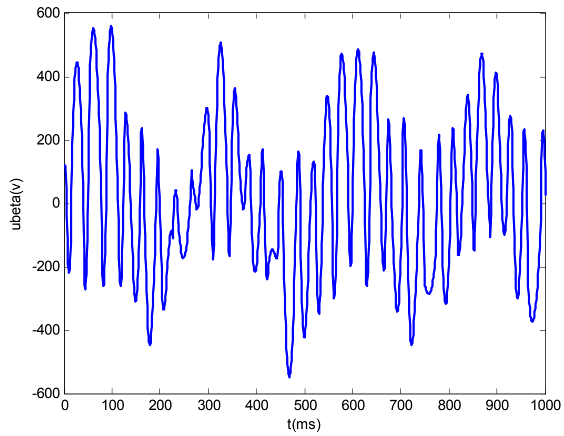

Fig. 7. Control law signal $u^{\beta}(k)$

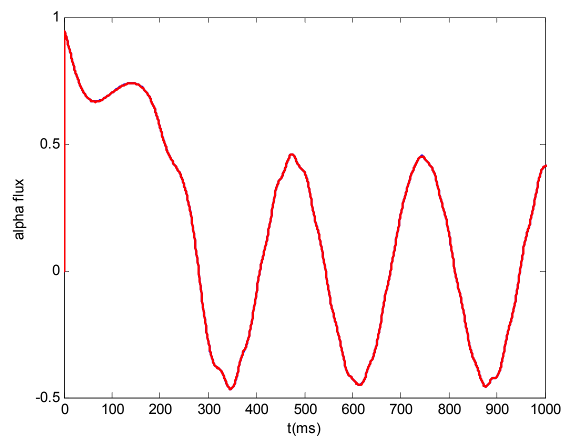

Fig. 8. Time evolution of $\psi^{\alpha}(k)$ and its estimate (real in solid line and estimated in dashed line)

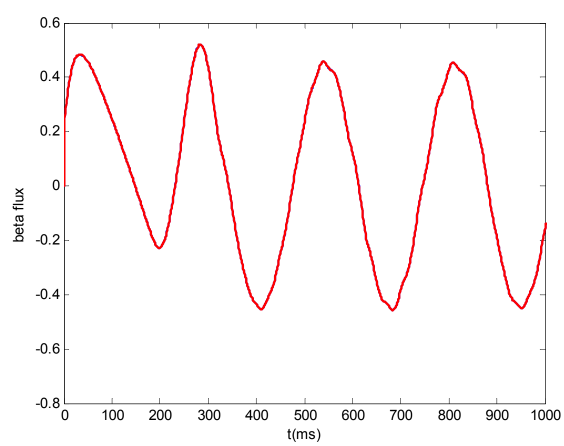

Fig. 9. Time evolution of $\psi^{\beta}(k)$ and its estimate (real in solid line and estimated in dashed line)

is validated by the simulation results presented above.

Comment 2. In this paper, the causality contradiction is avoided due to the fact that the controller is constructed based on the maximum r-step ahead predictor by the backstepping technique.

Comment 3. In literature there are few results that present both external disturbances (load torque) and parametric changes (resistance variations) as in this paper.

V. Conclusions

This paper has presented the application of HONN to solve the tracking problem for a class of MIMO nonlinear systems in discrete-time using backstepping technique. The training of the neural network is performed on-line using an extended Kalman filter. The boundness of the tracking error is established on the basis of the Lyapunov approach. The HONN training with the learning algorithm based in EKF presents good performance even in presence of larger bounded disturbances such as load torque variations and change on the plant parameters (resistance change).

References

[1] G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals and Systems*, vol. 2, pp. 304-314, 1989.

[2] F. Chen and H. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks", *IEEE Transactions on Automatic Control*, vol. 40, No. 5, pp. 791-801, 1995.

[3] L. A. Feldkamp, D. V. Prokhorov and T. M. Feldkamp, "Simple and conditioned adaptive behavior from Kalman filter trained recurrent networks", *Neural Networks*, vol. 16, pp. 683–689, 2003.

[4] 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.

[5] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control for Robotic Manipilators*, World Scientific, Singapore, 1998.

[6] 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.

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

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

[9] M. Krstic, I. Kanellakopoulos and P. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley and Sons, New York, USA, 1995.

[10] C. Leunga, and L. Chan, "Dual extended Kalman filtering in recurrent neural networks", *Neural Networks*, vol 16,pp. 223–239, 2003.

[11] A. G. Loukianov, B. Castillo-Toledo, and S. Dodds, "Robust stabilization of a class of uncertain system via block decomposition and VSC", *Int. J. Robust Nonlinear Control*, vol 12, pp. 1317–1338, 2002.

[12] Y. Li, S. Quiang, X. Zhuang and O. Kanak, "Robust and adaptive backstepping control for nonlinear systems using RBF neural networks", *IEEE Transactions on. Neural Networks*, vol. 15, pp. 693–701, May. 2004.

[13] 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.

[14] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on. Neural Networks*, vol. 1, pp. 4–27, Mar. 1990.

[15] A. S. Poznyak, E. N. Sanchez and W. Yu, *Differential Neural Networks for Robust Nonlinear Control*, World Scientific, Singapore, 2001.

[16] E. N. Sanchez, A. Y. Alanis and G. Chen, "Recurrent neural networks trained with Kalman filtering for discrete chaos reconstruction", *Dynamics of Continuous, Discrete and Impulsive Systems Series B (DCDIS_ B)*, To appear 2006.

[17] E. N. Sanchez and L. J. Ricalde, "Trajectory tracking via adaptive recurrent neural control with input saturation", *Proceedings of International Joint Conference on Neural Networks'03*, Portland, Oregon, USA, July, 2003.

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

[19] W. J. Rugh, *Linear System Theory*, 2nd ed, Prentice Hall, Upper Saddle River, N. J., USA, 1996.

[20] 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. 59-78,1995.

[21] 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.

[22] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks", *Neural Computation*, vol. 1, pp. 270–280, 1989.