Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Discrete-Time Backstepping Synchronous Generator Stabilization using a Neural Observer

Alma Y. Alanis * **Edgar N. Sanchez** ** **Alexander G. Loukianov** **

* *CUCEI, Universidad de Guadalajara, Av. Revolucion 1500, Col. Olimpica, C.P. 44430, Guadalajara, Jalisco, Mexico, e-mail: almayalanis@gmail.com*
** *CINVESTAV, Unidad Guadalajara, Apartado Postal 31-438, Plaza La Luna, Guadalajara, Jalisco, C.P. 45091, Mexico, e-mail: sanchez@gdl.cinvestav.mx*

Synchronous generators, discrete-time nonlinear observer, high-order neural network, extended Kalman filter, backstepping.

Abstract: This paper deals with adaptive tracking for discrete-time MIMO nonlinear systems in presence of bounded disturbances, based on a neural observer. 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); besides the observer is based on a recurrent high-order neural network (RHONN), which estimates the state vectors of the unknown plant dynamics. The learning algorithm for both neural networks is based on an Extended Kalman Filter (EKF). The applicability of the proposed approach is tested, via simulations, by its application to synchronous generators control.

1. INTRODUCTION

Nonlinear trajectory tracking is an important research subject (Ge et al. [2004], Chen et al. [1995], Krstic et al. [1995], Loukianov et al. [2002], Poznyak et al. [2001] and references therein). In recent adaptive and robust control literature, numerous approaches have been proposed for the design of nonlinear control systems. Among them, adaptive backstepping constitutes a major design methodology (Krstic et al. [1995]). 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. 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.

In most nonlinear control designs, it is usually assumed that all the system state is measurable. In practice, however, only parts of this state is measured directly. For this reason, nonlinear state estimation remains an important topic for study in nonlinear system theory (Poznyak et al. [2001]). Recently recurrent neural-network observers have been proposed, without requiring a precise plant model. This technique is therefore attractive and has been successfully applied to state estimation (Poznyak et al. [2001]). These works were developed mostly for continuous-time systems. Nonlinear discrete-time neural observers, on the other hand, have been seldom discussed (Alanis et al. [2006]).

The best-known training approach for recurrent neural networks (RNN) is the back propagation through time learning (Werbos [1990]). However, it is merely a first-order gradient descent method and hence its learning speed is very slow. Recently, some extended Kalman filter (EKF) based algorithms have been introduced to neural networks training (Singhal et al. [1989]). With an EKF-based algorithm, the learning convergence can be improved. Over the past decade, the EKF-based training of neural networks, both feedforward and recurrent ones, has proven to be reliable and practical for many applications (Alanis et al. [2007], Feldkamp et al. [2001], Singhal et al. [1989]).

In this paper, we propose a stabilizing control law based on the well-known backstepping methodology, which allows to track a rotor angle reference signal for a synchronous generator. First a neural observer is proposed, then using the neural model the block strict feedback decomposition is applied in order to define a number of sub-problems of lower order. Once this decomposition is achieved, the backstepping technique is used to design a suitable controller (Krstic et al. [1995]). Afterwards, this resulting controller is approximated by a High Order Neural Network (HONN) (Ge et al. [2004]). The implementation is simple and the training is performed on-line by means of an extended Kalman filter (EKF) (Sanchez et al. [2006]). The proposed control applicability is illustrated by trajectory tracking for a discrete-time synchronous generator model. The structure of this scheme is based on the separation

⋆ This work was supported by CONACYT Mexico, through Projects 46069Y and 57801Y.

principle for discrete-time nonlinear systems (Lin et al. [1994]). The main contributions of this paper are: first, the implementation of the proposed scheme, which does not require the knowledge of the generator parameters; second, simulations shows that the proposed scheme preserves stability and good performance when a short circuit fault is incepted and cleared out, and third, the stability analysis is based on the separation principle for discrete-time nonlinear systems.

2. MATHEMATICAL PRELIMINARIES

Let k denote the sampling step, $k \in 0 \cup \mathbb{Z}^{+}$, $|\cdot|$ be the absolute value, and $\|\cdot\|$ be the Euclidian norm for vectors and an adequate norm for matrices.

Following (Ge et al. [2004]), consider a MIMO nonlinear system,

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

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

where $x \in \Re^{n}$, $u \in \Re^{m}$, and $F \in \Re^{n} \times \Re^{m} \rightarrow \Re^{n}$ is a nonlinear map.

For system (1), after selecting the input u as a feedback function of the state:

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

one can obtain

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

which yields an unforced system

$$x(k+1) = \widetilde{F}(x(k)) \tag{5}$$

Let us define, the following discrete-time RHONN:

$$\widehat{x}_i(k+1) = w_i^{\top} z_i(\widehat{x}(k), u(k)), \quad i = 1, \cdots, n \tag{6}$$

where $\widehat{x}_i$ $(i = 1, 2, \cdots, n)$ is the state of the i-th neuron, L_i is the respective number of higher-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 $z_i(\widehat{x}(k), u(k))$ is given by

$$z_i(x(k), u(k)) = \begin{bmatrix} z_{i_1} \\ \vdots \\ z_{i_{L_i}} \end{bmatrix} = \begin{bmatrix} \Pi_{j \in I_1} \psi_{i_j}^{d i_j(1)} \\ \vdots \\ \Pi_{j \in I_{L_i}} \psi_{i_j}^{d i_j(L_i)} \end{bmatrix} \tag{7}$$

with $d_{j_i}(k)$ being a nonnegative integers, and

$$\psi_i = \begin{bmatrix} \psi_{i_1} \\ \vdots \\ \psi_{i_1} \\ \psi_{i_{n+1}} \\ \vdots \\ \psi_{i_{n+m}} \end{bmatrix} = \begin{bmatrix} S(\widehat{x}_1) \\ \vdots \\ S(\widehat{x}_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix} \tag{8}$$

in which $u = [u_1, u_2, \ldots, u_m]^{\top}$ is the input vector to the neural network, and $S(\cdot)$ is defined by

$$S(x) = \frac{1}{1 + \exp(-b\widehat{x})} \tag{9}$$

where $b > 0$ is a constant.

Now, consider the problem of approximating the general discrete-time nonlinear system (1), which is supposed to be observable, by the following discrete-time RHONN parallel representation (Rovithakis et al. [2001]):

$$x_i(k+1) = w_i^{*\top} z_i(x(k), u(k)) + \epsilon_{z_i} \tag{10}$$

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 (Rovithakis et al. [2001]).

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 analysis (Rovithakis et al. [2001]). In general, it is assumed that this vector exists and is an unknown constant. Define its estimate as w_i and the estimation error as

$$\widetilde{w}_i(k) = w_i(k) - w_i^* \tag{11}$$

The estimate w_i is used for stability analysis, which will be discussed later. Since w_i^* is constant, one has

$$\widetilde{w}_i(k+1) - \widetilde{w}_i(k) = w_i(k+1) - w_i(k)$$

3. THE EKF TRAINING ALGORITHM

The Kalman filter (KF) estimates the state of a linear system with additive state and output white noises (Grover et al. [1992], Song et al. [1995]). 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. Since the neural network mapping is nonlinear, an EKF is applied (see Sanchez et al. [2006] and references therein). The training goal is to find the optimal weight values, which minimize the prediction errors.

In this paper, we use a decoupled EKF-based training algorithm described by

$$\begin{aligned} w_i(k+1) &= w_i(k) + \eta_i K_i(k) e(k), \quad i = 1, \cdots, n \\ K_i(k) &= P_i(k) H_i(k) M_i(k) \\ P_i(k+1) &= P_i(k) - K_i(k) H_i^{\top}(k) P_i(k) + Q_i \end{aligned} \tag{12}$$

with

$$M_i(k) = \left[R_i + H_i^{\top}(k) P_i(k) H_i(k)\right]^{-1} \tag{13}$$

$$e(k) = y(k) - \widehat{y}(k) \tag{14}$$

where $e(k) \in \Re^{p}$ is the observation error and $P_i(k) \in \Re^{L_i \times L_i}$ is the weight estimation 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, $y \in \Re^{p}$ is the plant output, $\widehat{y} \in \Re^{p}$ is the NN output, n is the number of states, $K_i \in \Re^{L_i \times p}$ is the Kalman gain matrix, $Q_i \in \Re^{L_i \times L_i}$ is the NN weight estimation noise covariance matrix, $R_i \in \Re^{p \times p}$ is the error noise covariance, and $H_i \in \Re^{L_i \times p}$ is a matrix, in which each entry (H_{ij}) is the derivative of the i-th neural output with respect to ij-th neural network weight, (w_{ij}), given as follows:

$$H_{ij}(k) = \left[\frac{\partial \widehat{y}(k)}{\partial w_{ij}(k)}\right]^{\top} \tag{15}$$

where $j = 1, ..., L_i$ and $i = 1, ..., n$. 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, see Song et al. [1995]).

4. NEURAL OBSERVER DESIGN

This observer is first proposed in (Alanis et al. [2006]). A brief description is included for the sake of completeness.

In this section, we consider the estimation of the states of an observable discrete-time nonlinear system given by

$$\begin{aligned} x(k+1) &= F(x(k), u(k)) + d(k) \\ y(k) &= Cx(k) \end{aligned} \tag{16}$$

where $x \in \Re^n$ is the state vector of the system, $u(k) \in \Re^m$ is the input vector, $y(k) \in \Re^p$ is the output vector, $C \in \Re^{p \times n}$ is a known output matrix, $d(k) \in \Re^n$ is a disturbance vector, and $F(\cdot)$ is a smooth vector field with entries $F_i(\cdot)$. Hence, (16) can be rewritten as:

$$\begin{aligned} x(k) &= [\, x_1(k) \ \dots \ x_i(k) \ \dots \ x_n(k) \,]^\top \\ d(k) &= [\, d_1(k) \ \dots \ d_i(k) \ \dots \ d_n(k) \,]^\top \\ x_i(k+1) &= F_i(x(k), u(k)) + d_i(k) \ \ , i = 1, \cdots, n \\ y(k) &= Cx(k) \end{aligned} \tag{17}$$

For system (17), a recurrent neural Luenberger observer (RHONO) is proposed, with the following structure:

$$\begin{aligned} \widehat{x}(k) &= [\, \widehat{x}_1(k) \ \dots \ \widehat{x}_i(k) \ \dots \ \widehat{x}_n(k) \,]^\top \\ \widehat{x}_i(k+1) &= w_i^\top z_i(\widehat{x}(k), u(k)) + L_i e(k) \\ \widehat{y}(k) &= C\widehat{x}(k), \quad i = 1, \cdots, n \end{aligned} \tag{18}$$

with $L_i \in \Re^p$, w_i and z_i as in (6). The weight vectors are updated on-line with a decoupled EKF $(12) - (15)$; the output error is defined by

$$e(k) = y(k) - \widehat{y}(k) \tag{19}$$

and the state estimation error is

$$\widetilde{x}(k) = x(k) - \widehat{x}(k) \tag{20}$$

Then the dynamic of (20) can be expressed as

$$\begin{aligned} \widetilde{x}_i(k+1) &= \widetilde{w}_i(k) z_i(x(k), u(k)) \\ &\quad + \epsilon_{z_i} - L_i C \widetilde{x}(k) \end{aligned} \tag{21}$$

On the other hand the dynamic of (11) is

$$\widetilde{w}_i(k+1) = \widetilde{w}_i(k) - \eta_i K_i(k) e(k) \tag{22}$$

By summarizing (12)-(20), we obtain the first main result as follows.

Theorem 1: For system (17), the nonlinear observer (18) trained with the EKF-based algorithm (12), ensures that the output error (14) and the estimation error (20) are semi-globally uniformly ultimately bounded.

Proof. For the complete proof, we refer the reader to (Alanis et al. [2006]) ■

5. 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) (Krstic et al. [1995]), as follows:

$$\begin{aligned} x^i(k+1) &= f^i\left(\overline{x}^i(k)\right) + g^i\left(\overline{x}^i(k)\right) x^{i+1}(k) + d^i(k) \\ x^r(k+1) &= f^r(X(k)) + g^r(X(k)) u(k) + d^r(k) \\ y(k) &= x^1(k) \end{aligned} \tag{23}$$

where $X(k) = \left[x^{1\top}(k), \cdots, x^{r\top}(k)\right]^\top$ are the state variables, $\overline{x}^i(k) = \left[x^{1\top}, x^{2\top}, \cdots, x^{i\top}\right]^\top$, $x^i \in \Re^{n_i}$ $(i = 1, 2, \cdots, r-1)$, $r \geq 2$, 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. Moreover there exists a constant $\overline{d}_i$ such that $\|d_i(k)\| \leq \overline{d}_i$, for $k = 1, 2, \cdots$. If we consider the original system (23) 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 (Chen et al. [1995], Ge et al. [2004]):

$$\begin{aligned} x^1(k+r) &= F^1\left(\bar{x}^1(k)\right) \\ &\quad + G^1\left(\bar{x}^1(k)\right) x^2(k+r-1) + d^1(k+r) \\ &\ \ \vdots \\ x^{r-1}(k+2) &= F^{r-1}\left(\bar{x}^{r-1}(k)\right) \\ &\quad + G^{r-1}\left(\bar{x}^{r-1}(k)\right) x^r(k+1) \\ &\quad + d^{r-1}(k+2) \\ x^r(k+1) &= f^r(X(k)) + g^r(X(k)) u(k) + d^r(k) \\ y(k) &= x^1(k) \end{aligned} \tag{24}$$

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 $i = 1, \cdots, r-1$

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

Then, system (24) can be written as

$$\begin{aligned} x^1(k+r) &= F^1(k) + G^1(k) x^2(k+r-1) \\ &\quad + d^1(k+r) \\ &\ \ \vdots \\ x^{r-1}(k+2) &= F^{r-1}(k) + G^{r-1}(k) x^r(k+1) \\ &\quad + d^{r-1}(k+2) \\ x^r(k+1) &= f^r(k) + g^r(k) u(k) + d^r(k) \\ y(k) &= x_1(k) \end{aligned} \tag{25}$$

The objective is to design a control $u(k)$ to force the system output $y(k)$ to track a desired trajectory $y_d(k)$. Once (25) is defined, we apply the well known backstepping technique (Krstic et al. [1995]). For system (25), we can define the desired virtual controls $\left(\alpha^{j*}(k), \ \ j = 1, \cdots, r-1\right)$ and the ideal practical control $(u^*(k))$ as follows:

$$\begin{aligned}\alpha^{1*}(k) &\triangleq x^2(k) = \varphi^1\left(\overline{x}^1(k), y_d(k+r)\right)\\ \alpha^{2*}(k) &\triangleq x^3(k) = \varphi^2\left(\overline{x}^2(k), \alpha^{1*}(k)\right)\\ &\vdots\\ \alpha^{r-1*}(k) &\triangleq x^r(k) = \varphi^{r-1}\left(\overline{x}^{r-1}(k), \alpha^{r-2*}(k)\right)\\ u^*(k) &= \varphi^r\left(X(k), \alpha^{r-1*}(k)\right)\\ y(k) &= x^1(k) \end{aligned} \tag{26}$$

where φ^j $(j = 1, \cdots, r)$ 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 (Ge et al. [2004]). 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 (Ge et al. [2004]), we construct the virtual and practical controls via embedded backstepping without the causality contradiction (Chen et al. [1995]). Let us approximate the virtual controls and practical control by the following High Order Neural Network (HONN) (Alanis et al. [2007]):

$$\begin{aligned}\alpha^i(k) &= w^{i^\top} S^i\left(z^i(k)\right), \qquad i = 1, \cdots, r-1\\ u(k) &= w^{r^\top} S^r\left(z^r(k)\right)\end{aligned} \tag{27}$$

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*} $(j = 1, \cdots, r)$ and $S^j \in \Re^{L_j \times n_j}$ Define the estimation error as:

$$\widetilde{w}^j(k) = w^{j*} - w^j(k) \tag{28}$$

Then the corresponding weights updating laws are defined of the form of (12), with

$$e(k) = \begin{bmatrix} y_d(k) - y(k)\\ x^2(k) - \alpha^1(k)\\ \vdots\\ x^r(k) - \alpha^{r-1}(k)\end{bmatrix} \tag{29}$$

Considering $(23) - (29)$, we establish the second main result in the following theorem.

Theorem 2: For the system (23), the HONN (27) trained with the EKF-based algorithm (12) to approximate the control law (26), ensures that the tracking error (29) is semiglobally uniformly ultimately bounded (SGUUB); moreover, the HONN weights remain bounded.

Proof. Due to space limitations, we refer the reader to (Alanis et al. [2007]) ■

The third main result of this paper is the following.

Proposition 1. Given a desired output trajectory y_d, a dynamic system with output y, and a neural network with output $\widehat{y}$, the following inequality holds:

$$\|y_d - y\| \leq \|\widehat{y} - y\| + \|y_d - \widehat{y}\|$$

where $y_d - y$ is the system output tracking error, and $\widehat{y} - y$ is the output estimation error and $y_d - \widehat{y}$ is the output tracking error of the nonlinear observer.

Based on this proposition, it is possible to divide the tracking objective into two parts:

(1) Minimization of $\widehat{y} - y$, which can be achieved by the proposed on-line nonlinear observer algorithm (12) as established in Theorem 1.
(2) Minimization of $y_d - \widehat{y}$. For this, a tracking algorithm is developed on the basis of the nonlinear observer (6). This minimization is obtained by designing the control law (26), as stated in Theorem 2.

It is possible to establish *Proposition 1* due to the separation principle for discrete-time nonlinear systems (Lin et al. [1994]) as follows.

Theorem 3. (Separation Principle) (Lin et al. [1994]): The asymptotic stabilization problem of the system (1)-(2) is solvable via estimated state feedback, if and only if, the system (1)-(2) is asymptotically stabilizable and exponentially detectable.

Corollary 1 (Lin et al. [1994]): There is an exponential observer for a Lyapunov stable discrete-time nonlinear system (1)-(2) with $u = 0$ if, and only if, the linearized system of the system (1)-(2) is detectable.

6. SYNCHRONOUS GENERATOR CONTROL

In this section, we apply the previous control technique to a discrete-time synchronous generator model (De Leon-Morales et al. [2003], Orta et al. [2001]). We consider a synchronous generator connected through purely reactive transmission lines to the rest of the network, which is represented by an infinite bus. The discrete-time model of the synchronous generator is given by (De Leon-Morales et al. [2003], Orta et al. [2001]):

$$\begin{aligned} x_1(k+1) &= f^1\left(\overline{x}^1(k)\right) + \tau x_2(k)\\ x_2(k+1) &= f^2\left(\overline{x}^2(k)\right) + \tau m_2 x_3(k)\\ x_3(k+1) &= f^3\left(\overline{x}^3(k)\right) + \tau m_6 u(k)\end{aligned} \tag{30}$$

with

$$\begin{aligned} f^1\left(\overline{x}^1(k)\right) &= x_1(k)\\ f^2\left(\overline{x}^2(k)\right) &= x_2(k)\\ +\tau &\left[m_1 + \left(m_2 E_q^{'*} + m_3\cos(\widetilde{x}_1)\right)\sin(\widetilde{x}_1)\right]\\ f^3\left(\overline{x}^3(k)\right) &= x_3(k)\\ +\tau &\left[m_4\left(x_3(k) + E_q^{'*}\right) + m_5\cos(\widetilde{x}_1) + m_6 E_{fd}^*\right]\end{aligned}$$

and $\widetilde{x}_1 = x_1(k) + \delta^*$, $m_1 = \frac{T_m}{M}$, $m_2 = \frac{-V}{MX_d'}$, $m_3 = \frac{V^2}{M}\left(\frac{1}{X_d'} - \frac{1}{X_q}\right)$, $m_4 = -\frac{X_d}{T_{do}'X_d'}$, $m_5 = -\left(\frac{X_d' - X_d}{T_{do}'X_d'}\right)V$, $m_6 = \frac{1}{T_{do}'}$, $x_1 = \Delta\delta = \delta - \delta^*$, $x_2 = \Delta\omega = \omega - \omega^*$, $x_3 = \Delta E_q' = E_q' - E_q^{'*}$. where δ is the generator rotor angle referred to the infinite bus (also called power angle), ω is the rotor angular speed and E_q' is the stator voltage which is proportional to flux linkages. M is the per unit inertia constant, T_m is the constant mechanical power supplied by the turbine, and $T_{do}^{'}$ is the transient open circuit time constant. $X_d = x_d + x_L$ is the augmented reactance, where x_d is the direct axis reactance and x_L is the line reactance, $X_d^{'}$ is the transient augmented reactance and V is the infinite bus voltage which is fixed. P_g is the generated power while E_{fd} is the stator equivalent voltage given by field voltage v_f.

$$P_g = \frac{1}{X_d^{'}} E_q^{'} V \sin(\delta) + \frac{1}{2}\left(\frac{1}{X_q} - \frac{1}{X_d^{'}}\right) V^2 \sin(2\delta)$$
$$E_{fd} = \frac{\omega_s M_f}{\sqrt{2} R_f} v_f$$

where v_f is the scaled field excitation voltage, $x_d^{'}$ is the transient direct axis reactance, x_q is the quadrature axis reactance, M_f is the mutual inductance between stator coils, R_f is the field resistance and ω_s is the synchronous speed. As in (De Leon-Morales et al. [2003], Orta et al. [2001]), we only consider the case where the dynamics of the damper windings are neglected, *i. e.* $D = 0$. Through this work, the analysis and design are done around an operation point $\left(\delta^*, \omega^*, E_q^{'*}\right)$.

To this end, we use the proposed RHONO for the discrete-time synchronous generator model (18), which is described as:

$$\begin{aligned}
\widehat{x}_1(k+1) &= w_{11} S(\widehat{x}_1(k)) + w_{12} S(\widehat{x}_2(k)) \\
\widehat{x}_2(k+1) &= w_{21} S(\widehat{x}_1(k))^6 + w_{22} S(\widehat{x}_2(k))^2 \\
&\quad + w_{23} S(\widehat{x}_3(k)) \\
\widehat{x}_3(k+1) &= w_{31} S(\widehat{x}_1(k))^2 + w_{32} S(\widehat{x}_2(k)) \\
&\quad + w_{33} S(\widehat{x}_3(k))^2 + w_{34} S(u(k)) \qquad (31)
\end{aligned}$$

where $\widehat{x}_i$ estimates x_i $(i = 1, 2, 3)$. The training is performed on-line, using a parallel configuration, with an EKF (12). 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.

Now we use the HONN to approximate the desired virtual controls and the ideal practical controls for system (31), described as follows:

$$\begin{aligned}
\alpha^1(k) &= w^{1^\top} S^1\left(z^1(k)\right) \\
\alpha^2(k) &= w^{2^\top} S^2\left(z^2(k)\right) \\
u(k) &= w^{3^\top} S^3\left(z^3(k)\right) \qquad (32)
\end{aligned}$$

with

$$\begin{aligned}
z^1(k) &= \left[\widehat{x}_1(k), y_d(k+3)\right]^\top \\
z^2(k) &= \left[\widehat{x}_1(k), \widehat{x}_2(k), \alpha^1(k)\right]^\top \\
z^3(k) &= \left[\widehat{x}_1(k), \widehat{x}_2(k), \widehat{x}_3(k), \alpha^2(k)\right]^\top
\end{aligned}$$

The controller training is performed on-line, with an EKF (12). All the NN states are initialized randomly. The simulation is performing using system (30) with the following parameters (per unit) $T_m = 1$, $M = 0.033$, $\omega_s = 1$, $T_{do}^{'} = 0.033$, $X_q = X_d = 0.9$, $X_d^{'} = 0.3$, $V = 1$, $E_{fd}^* = 1.1773$, $\delta^* = 0.870204$ and $\omega^* = 1$. We analyze the response of the system to a short circuit generated by the connection of small impedance between the machine terminals and the ground incepted at time of $300ms$; this impedance is disconnected after $50ms$, calling the clearing time. Then, the system goes back to its pre-disturbance state. Figures 1-3 present the performance of the whole control scheme. Fig. 4 displays the neural networks weights evolution. The equilibrium point is considered fixed. It can be seen that the state of the observer (dashed line) quickly converges to the state of the system (solid line) and it can be seen as well, that the respective state variables converge to the desired equilibrium point.

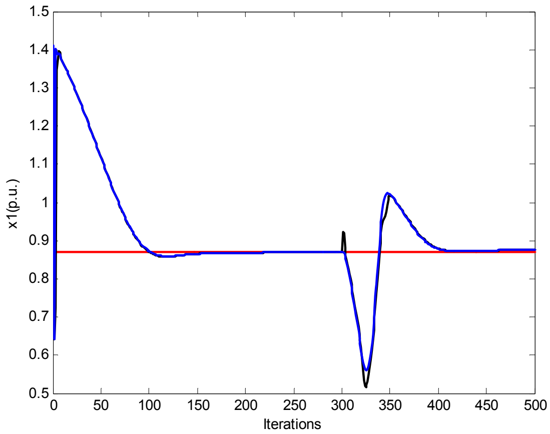

Fig. 1. Time evolution for $x_1(k)$ using the proposed observer-controller scheme (plant state $x_1(k)$ in solid line, estimated state $\widehat{x}_1(k)$ in dash-dot line, desired $x_1^d(k)$ signal in dashed line)

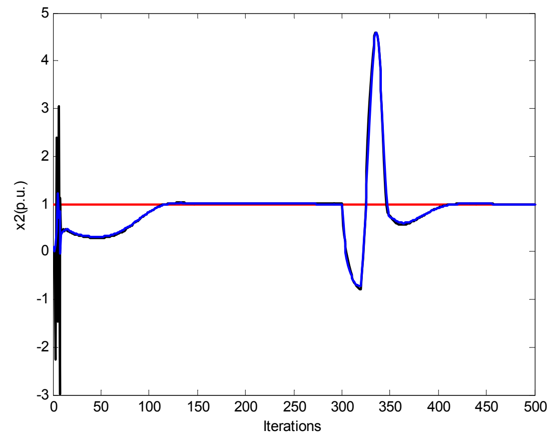

Fig. 2. Time evolution for $x_2(k)$ using the proposed observer-controller scheme (plant state $x_2(k)$ in solid line, estimated state $\widehat{x}_2(k)$ in dash-dot line, desired $x_2^d(k)$ signal in dashed line)

7. 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 the backstepping technique. It also uses a RHONN to develop a recurrent neural observer. The training for both neural networks is performed on-line using an extended Kalman filter. The boundness of the tracking error, the neural weights, and the observa-

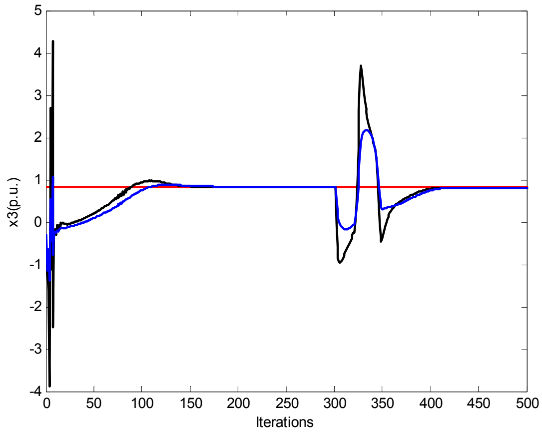

Fig. 3. Time evolution for $x_3(k)$ using the proposed observer-controller scheme (plant state $x_3(k)$ in solid line, estimated state $\widehat{x}_3(k)$ in dash-dot line, desired $x_3^d(k)$signal in dashed line)

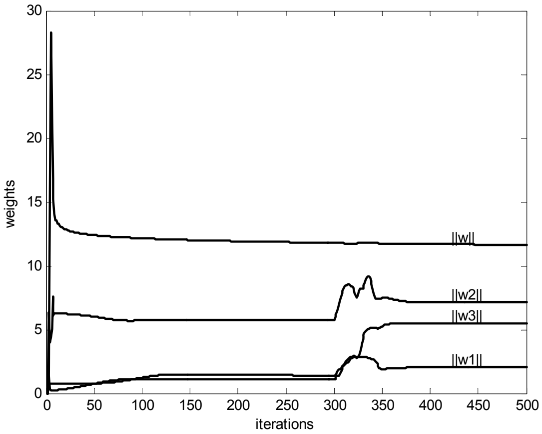

Fig. 4. Weight evolution

tion error are established on the basis of the Lyapunov approach. Simulation results illustrate the applicability of the proposed control scheme.

ACKNOWLEDGEMENTS

The authors thank the support of CONACYT Mexico, through Projects 46069Y and 57801Y. The first author also thanks the support of "CONACYT Fondo Institucional".

REFERENCES

A. Y. Alanis, E. N. Sanchez and A. G. Loukianov. Discrete-time nonlinear recurrent high order neural observer. *Proceedings of the IEEE International Symposium on Intelligent Control'06*, Munich, Germany, October, 2006.

A. Y. Alanis, E. N. Sanchez and A. G. Loukianov. Discrete time adaptive backstepping nonlinear control via high order neural networks. *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 1185-1195, July, 2007.

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.

L. A. Feldkamp, T. M. Feldkamp, and D. V. Prokhorov. Neural network training with the nprKF. *Proceedings of the IEEE International Joint Conference on Neural Networks'01*, 109-114, 2001.

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.

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.

Y. H. Kim and F. L. Lewis. *High-Level Feedback Control with Neural Networks.* World Scientific, Singapore, 1998.

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

J. De León-Morales, O. Huerta-Guevara, L. Dugard and J. M. Dion. Discrete-time nonlinear control scheme for synchronous generator. *Proceedings of the IEEE Conference on Decision and Control'03*, Maui, Hawaii, USA, December, 2003.

W. Lin and C. I. Byrnes. Design of discrete-time nonlinear control systems via smooth feedback. *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2340-2346, 1994.

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 and Nonlinear Control*, vol. 12, pp. 1317–1338, 2002.

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

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

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

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*, Part-B, vol. 6, no. 13, pp. 1-17, December, 2006.

A. E. Sanchez-Orta, C. Guerra-Torres and J. De León-Morales. Discretization scheme for a synchronous generator. *Proceedings of the IEEE Conference on Decision and Control'01*, Orlando, Florida, USA, December, 2001.

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.

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

P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550 - 1560, 1990.