2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Real-time neural control for discrete nonlinear systems under unknown input and state disturbances

1st Alma Y. Alanis*
CUCEI, University of Guadalajara
Guadalajara, Mexico
alma.alanis@academicos.udg.mx

2nd Jesus G. Alvarez
CUCEI, University of Guadalajara
Guadalajara, Mexico
jesus.alvarez9280@alumnos.udg.mx

3rd Oscar D. Sanchez
CUCEI, University of Guadalajara
Guadalajara, Mexico
didier.sanchez@academicos.udg.mx

4th Pavel Zuniga
CUCEI, University of Guadalajara
Guadalajara, Mexico
erasmo.martinez@academicos.udg.mx

5th Gustavo Munoz-Gomez
Tecnologico Nacional de Mexico
Nochistlan, Mexico
munozgtavo@gmail.com

***Abstract*—This paper presents the design of an intelligent controller for uncertain discrete-time nonlinear systems. The proposed controller is resilient to external unknown disturbances as well as state and input uncertainties, even though the model of the system is considered unknown. An intelligent controller is designed for an unknown discrete-time nonlinear system, this controller is model-free and it is based on sensor measurements and therefore including unknown system dynamics, actuator nonlinearities, measurement errors, noise, uncertainties, external disturbances and other phenomena associated to real-world applications. Then, using the neural model, a backstepping controller is designed to ensure a resilient performance. Finally, real-time results are included to demonstrate the effectiveness of the proposed approach using a three-phase induction motor.**

***Index Terms*—Uncertain discrete-time nonlinear system, Intelligent control, Neural control, Resilience, Experimental results, Induction motor**

I. Introduction

The term "Intelligent Control" is often used lightly and inaccurately to gain recognition or publicity. Intelligent control is essentially the convergence of two well-established disciplines, control theory and artificial intelligence. Control theory is a well-known methodology that is characterized by a strict mathematical approach that has a long and fascinating history [1]. On the other hand, artificial intelligence is considered a relatively young discipline that has proven to be an excellent technique that deals with real-world problems in an unconventional way, imitating the characteristics of biological intelligence. By combining the previously mentioned methodologies intelligent control is produced which, according to [2], can be defined as a control technique that imitates a human who has the experience to generate an appropriate control action to a particular system. In contrast, conventional control algorithms match observed or measured information with a control knowledge-base to generate the control action (controller output or control input to the system).

The authors, thank the support of CONAHCYT Mexico, through Projects FOP16-2021-01-319619 and FOP16-2021-01-319608.

By definition, intelligent control requires at least one artificial intelligence technique to be incorporated in the controller design. One of the most well-known intelligent controllers is the neural control which utilizes artificial neural networks to solve complex control problems. Direct and indirect neural control are the two main types of methodologies used in intelligent control design as described in [3]. In direct neural control, as its name implies, the artificial neural network implements the system controller directly in order to solve a regulation or reference tracking problem. Whereas Indirect neural control uses the artificial neural network to design a mathematical model for the system and subsequently uses that model to design an intelligent controller based on any preferred control algorithm. Both approaches have advantages and disadvantages depending on the application [3] and both can be designed based on measured data in a model-free approach.

Therefore, in this paper, we propose the use of a direct neural controller for a type of discrete-time nonlinear systems with a data-based model-free approach resilient to input and state uncertainties in the presence of sensor errors, external disturbances, noise, actuator nonlinearities and unknown dynamics. It is important to note that the controller described in this paper is designed with a backstepping technique.

The main contributions of this work can be summarized as follows:

1) A resilient controller for discrete-time unknown nonlinear systems with a data-based model-free approach that does not need a previously known mathematical model or an explicit estimator design of the system.
2) A biologically-inspired direct neural network control algorithm with a simple design and low computational complexity that directly implements the control law.
3) A real-time implementation using a rotatory induction motor test bench with output trajectory tracking profiles designed to stress the motor performance.

The rest of this paper is organized as follows: Section 2,

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

includes the controller design methodology. Section 3, presents the application of the proposed scheme to a three-phase induction motor. Section 4, includes the real-time implementation for a three-phase induction motor. Section 5 presents the conclusions.

II. Controller design

Consider a discrete-time nonlinear system in the general form:

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

it is possible to describe the system with input and state variable uncertainties as:

$$\begin{aligned} X(k) &= \chi(k) + \Delta_x(k) \\ u(k) &= \upsilon(k) + \Delta_u(k) \end{aligned}$$

where $\Delta_x(k)$ and $\Delta_u(k)$ represent state and input uncertainties, respectively, which are considered unknown and bounded. It is important to consider that the controller is designed using the neural backsteppig technique.

The mathematical model of nonlinear systems can be written in, or transformed into, a special state-form named block strict feedback form [5], as follows:

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

where $X(k) = \left[x^{1\top}(k), \cdots, x^{r\top}(k)\right]^\top$ represents the system state, $\overline{x}^i(k) = \left[x^{1\top}, x^{2\top}, \cdots, x^{i\top}\right]^\top$, $x^i \in \Re^{n_i}$ $(i = 1, 2, \cdots, r-1)$, $r \geq 2$, r is the block number, $u(k) \in \Re^m$ are the system inputs, $y(k) \in \Re^m$ is the system output, $d^i \in \Re^{n_i}$ is the disturbance vector. Moreover, a constant $\overline{d}_i$ exists such that $\|d_i(k)\| \leq \overline{d}_i$, for $0 < k < \infty$.

System (2) can be considered as a one-step ahead predictor, which can then be transformed into an equivalent maximum r-step ahead predictor, which can predict the future state values $x^1(k+r), x^2(k+r-1), \cdots, x^r(k+1)$, it is then possible to avoid the causality contradiction if the controller is designed based on this r-step ahead predictor by backstepping [6], [8]. Next, the system (2) can be rewritten as

$$\begin{aligned} x^1(k+r) &= F^1(k) + G^1(k)x^2(k+r-1) \\ &+ d^1(k+r) \\ &\vdots \\ x^{r-1}(k+2) &= F^{r-1}(k) + G^{r-1}(k)x^r(k+1) \\ &+ d^{r-1}(k+2) \\ x^r(k+1) &= f^r(k) + g^r(k)u(k) + d^r(k) \\ y(k) &= x_1(k) \end{aligned} \tag{3}$$

Now, the control objective is to design $u(k)$ to drive the system output $y(k)$ to track a desired reference signal $y_d(k)$. When, (3) is defined, the well-known backstepping technique can be applied [5]. For system (3), it is possible to 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:

$$\begin{aligned} \alpha^{1*}(k) &\triangleq x^2(k) = \varphi^1\left(\overline{x}^1(k), y_d(k+r)\right) \\ \alpha^{2*}(k) &\triangleq x^3(k) = \varphi^2\left(\overline{x}^2(k), \alpha^{1*}(k)\right) \\ &\vdots \\ \alpha^{r-1*}(k) &\triangleq x^r(k) = \varphi^{r-1}\left(\overline{x}^{r-1}(k), \alpha^{r-2*}(k)\right) \\ u^*(k) &= \varphi^r\left(X(k), \alpha^{r-1*}(k)\right) \\ y(k) &= x^1(k) \end{aligned} \tag{4}$$

where φ^j $(1 \leq j \leq r)$ are smooth nonlinear functions. Then, the desired virtual controls $\alpha^{j*}(k)$ and the ideal control $u^*(k)$ will force the output $y(k)$ to follow the desired signal $y_d(k)$ when the exact system model is known and does not present unknown disturbances nor uncertainties. It is important to note that the previous conditions are considered ideal conditions which cannot be met in real-world applications, hence, neural networks will be used to approximate both, desired virtual controls and desired practical controls, when these ideal conditions are not satisfied.

A. Neural backstepping controller

Consider a HONN described by:

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

where $z = [z_1, z_2, \cdots, z_q]^\top \in \Omega_z \subset \Re^q$ are positive integers, q denotes the input vector dimension, L denotes the number of HONN weights, $\phi \in \Re^m$, $\{I_1, I_2, \cdots, I_L\}$ is a collection of non-ordered subsets of $\{1, 2, \cdots, q\}$, $S(z) \in \Re^{L \times m}$, $d_j(i_j)$ are non-negative integers, $w \in \Re^L$ is a vector which contains HONN weights, and $s(z_j)$ is chosen as an hyperbolic tangent function:

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

Figure 1 depicts the modelling scheme proposed in this paper, it is important to note that the neural network models the entire system, including: state uncertainties, input uncertainties, unmodelled dynamics, external disturbances, sensor errors, noise, actuator limitations, among others and they are represented by x and u, affected, respectively, by Δ_x and Δ_u which represent state and input uncertainties that, as in (1), are considered unknown and bounded.

For a desired function $\upsilon^*(z) \in \Re^m$, the existence of weight vectors w_j^* and w_r^* is assumed such that virtual and practical controllers can be approximated by ideal HONNs on a compact set $\Omega_z \subset \Re^q$

$$\alpha^{j*}(k) = w_j^{*\top} S_j(z_j) + \epsilon_z \tag{7}$$

$$\upsilon^*(z) = w_r^{*\top} S_r(z_r) + \epsilon_z \tag{8}$$

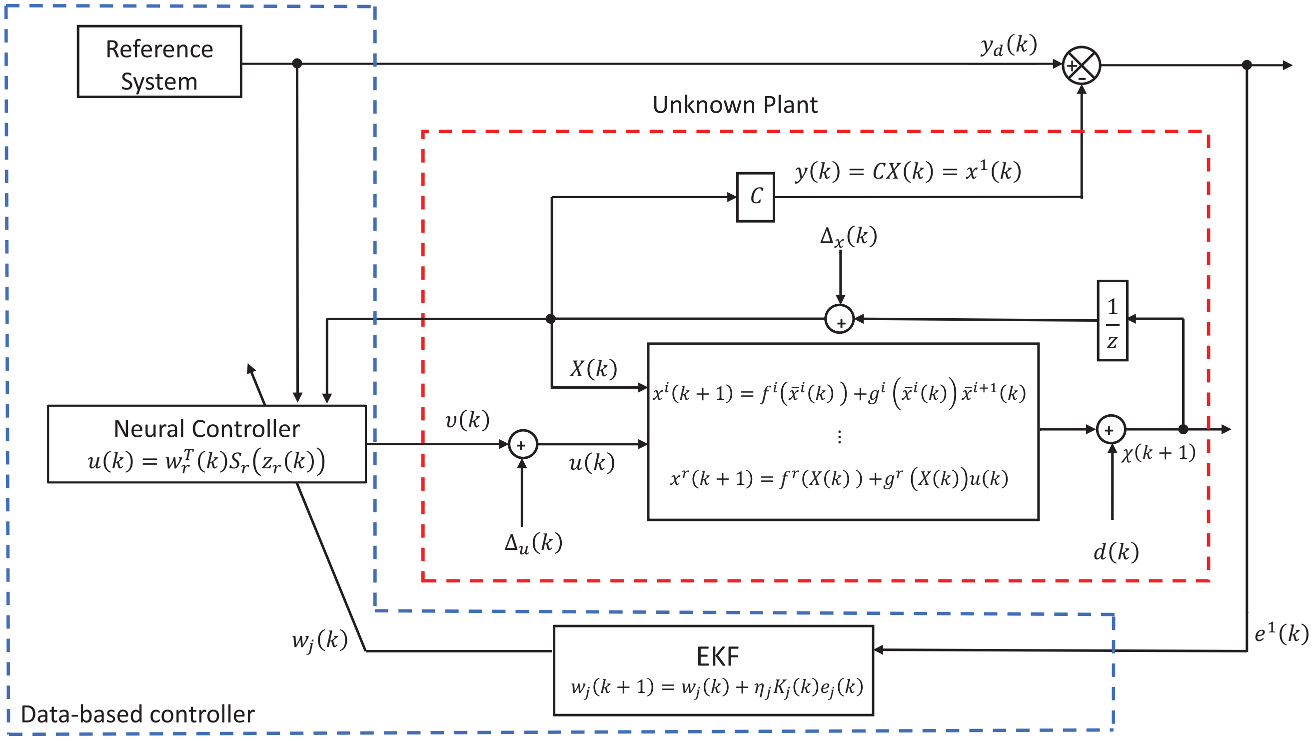

Fig. 1. Direct neural control scheme for unknown discrete-time nonlinear systems.

where $j = 1, \cdots, r-1$, r is the number of blocks as in (2) and $\epsilon_z \in \Re^m$ is the HONN approximation error vector, considered bounded [9]; 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 purposes [7], [10], which is considered unknown and constant w_l^*, whose estimate is $w_l \in \Re^L$. Hence, it is possible to define:

$$\widetilde{w}_l(k) = w_l(k) - w_l^* \tag{9}$$

as the estimation error, with $l = 1, \cdots, r$.

The training goal is to find the optimal weight values which minimize the prediction error. The modified Extended Kalman Filter (EKF) algorithm is defined by:

$$\begin{aligned} w_l(k+1) &= w_l(k) + \eta_l K_l(k) e_l(k) \\ K_l(k) &= P_l(k) H_l(k) M_l(k) \\ M_l(k) &= \left[R_l(k) + H_l^{\top}(k) P_l(k) H_l(k)\right]^{-1} \\ P_l(k+1) &= P_l(k) - K_l(k) H^{\mathsf{T}}(k) P_l(k) + Q_l(k) \end{aligned} \tag{10}$$

where $P_l \in \Re^{L_l \times L_l}$ is the prediction error associated covariance matrix, $w_l \in \Re^{L_l}$ is the weight (state) vector, L_l is the respective number of neural network weights, r is the number of blocks, $K_l \in \Re^{L_l}$ is the Kalman gain vector, $Q_l \in \Re^{L_l \times L_l}$ is the state noise associated covariance matrix, $R_l \in \Re$ is the measurement noise associated covariance, and H_l is defined as:

$$H_l(k) = \left[\frac{\partial \widehat{\nu}^l(k)}{\partial w_l(k)}\right] \tag{11}$$

where $\widehat{\nu}^l(k) \in \Re^{n_l}$ is the HONN function approximation. Usually, P_i and Q_i are initialized as diagonal matrices, with entries $P_i(0)$ and $Q_i(0)$, respectively.

Now, the backstepping controller (4) is approximated by a HONN, without the causality contradiction [6]. Let us approximate the virtual controls and practical control by the following HONN ($j = 1, \cdots, r-1$):

$$\begin{aligned} \alpha^j(k) &= w_{j^{\top}} S_j(z_j(k)) \\ v(k) &= w_{r^{\top}} S_r(z_r(k)) \end{aligned} \tag{12}$$

with

$$z^1(k) = \left[x^1(k), y_d(k+r)\right]^{\top} \tag{13}$$

$$z^j(k) = \left[\overline{x}^j(k), \alpha^{j-1}(k)\right]^{\top} \tag{14}$$

$$z^r(k) = \left[X(k), \alpha^{r-1}(k)\right]^{\top} \tag{15}$$

$$\tag{16}$$

with $j = 1, \cdots, r-1$, where $w^j \in \Re^{L_j}$ are the estimates of ideal constant weights w^{l*} ($l = 1, \cdots, r$) and $S^l \in \Re^{L_l \times n_l}$.

The HONN (12) is trained with an EKF (10), with

$$\begin{aligned} e^1(k) &= y_d(k) - y(k) \\ e^2(k) &= x^2(k) - \alpha^1(k) \\ &\vdots \\ e^r(k) &= x^r(k) - \alpha^{r-1}(k) \end{aligned} \tag{17}$$

The previous statement implies that the proposed controller allows us to obtain bounded errors for output trajectory tracking and weights estimation. The norm of the bounded error for output trajectory tracking depends on the uncertainties and disturbances affecting the system and its measurements. For the weight estimations bounded errors, the bound of the weights estimation depends on the inability of the neural network to accurately approximate the dynamics of the system with the available information [3]. It is well known that the previously mentioned error can be minimized with an infinite number of connections [10].

It is important to note, that this controller is model-free and the control law is defined by a HONN trained with measured data, therefore such measurements contain uncertainties and noise. In a similar way, the control applied to the system is not necessarily equal to the one calculated by the algorithm, and therefore suffers from uncertainties, noise and actuator inaccuracies, all these phenomena are to be compensated online by the controller for accurate trajectory tracking of the system.

III. Application to a three-phase induction motor

This work focuses on controlling rotor speed and flux magnitude for trajectory tracking of time-varying reference signals, this type of control is of significance for different applications, especially for mechatronic and transportation systems [4], [11]–[16]. Mathematical models are a central tool to design modern control techniques, however, an accurate model for complex systems is not always available. Therefore, this work proposes a discrete-time resilient neural controller for a three-phase induction motor using a HONN trained with an extended Kalman filter to compute the control law. The controller solves the trajectory tracking problem for time-varying signals with input and state uncertainties under unknown external disturbances, and under the assumption that the induction motor model is unknown. The neural controller is designed using the neural backstepping technique previously described.

It is well known that electromechanical systems can be written in the BSFF (3) [11]–[14], with $r = 2$, then. Then, it is possible to design a neural controller (12) trained with an EKF-based algorithm (10). The following state variables are defined

$$\begin{aligned} x^1(k) &= \begin{bmatrix} \omega(k) \\ \Psi(k) \end{bmatrix}; \; x^2(k) = \begin{bmatrix} i^\alpha(k) \\ i^\beta(k) \end{bmatrix} \\ u(k) &= \begin{bmatrix} u^\alpha(k) \\ u^\beta(k) \end{bmatrix}; \; y_d(k) = \begin{bmatrix} \omega_d(k) \\ \Psi_d(k) \end{bmatrix} \\ y(k) &= x^1(k) \end{aligned} \tag{18}$$

where $\Psi(k) = \psi^{\alpha^2}(k) + \psi^{\beta^2}(k)$ is the rotor flux magnitude, and $\omega_d(k)$ and $\Psi_d(k)$ are the reference signals. The control objective is to drive the output $y(k)$ to follow the time-varying reference signal $y_d(k)$. Using (18), the induction motor can be described in two blocks in the BSFF:

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

where $f^1\left(x^1(k)\right), g^1\left(x_1(k)\right)$, $f_2\left(\overline{x}_2(k)\right)$ and $g_2\left(\overline{x}_2(k)\right)$ are considered unknown for this application, and $d_1(k)$ is the unknown bounded disturbance applied as external load torque. Now we use the HONN to approximate the desired virtual controls and the ideal practical controls described as:

$$\begin{aligned} \alpha^{1*}(k) &\triangleq x^2(k) = \varphi^1\left(x^1(k), y_d(k+2)\right) \\ \upsilon^*(k) &= \varphi^2\left(x^1(k), x^2(k), \alpha^{1*}(k)\right) \\ y(k) &= x^1(k) \end{aligned} \tag{19}$$

Then, the HONN proposed for this application is designed as follows:

$$\begin{aligned} \alpha^1(k) &= w_1^\top S_1\left(z^1(k)\right) \\ \upsilon(k) &= w_2^\top S_2\left(z^2(k)\right) \end{aligned} \tag{20}$$

with

$$z^1(k) = \left[x^1(k), y_d(k+2)\right]^\top \tag{21}$$

$$z^2(k) = \left[x^1(k), x^2(k), \alpha^1(k)\right]^\top \tag{22}$$

HONN weights are updated using the EKF as follows:

$$\begin{aligned} w_i(k+1) &= w_i(k) + \eta_i K_i(k) e_i(k) \qquad (23) \\ K_i(k) &= P_i(k) H_i(k) M_i(k) \\ M_i(k) &= \left[R_i(k) + H_i^\top(k) P_i(k) H_i(k)\right]^{-1} \\ P_i(k+1) &= P_i(k) - K_i(k) H_i^\top(k) P_i(k) + Q_i(k) \end{aligned}$$

with $i = 1, 2$ and

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

The training is performed online, using a series-parallel configuration. All the HONN state variables are initialized randomly and associated covariances matrices are initialized as diagonal ones with nonzero elements selected as: $P_1(0) = P_2(0) = 10000$; $Q_1(0) = Q_2(0) = 5000$ and $R_1(0) = R_2(0) = 10000$.

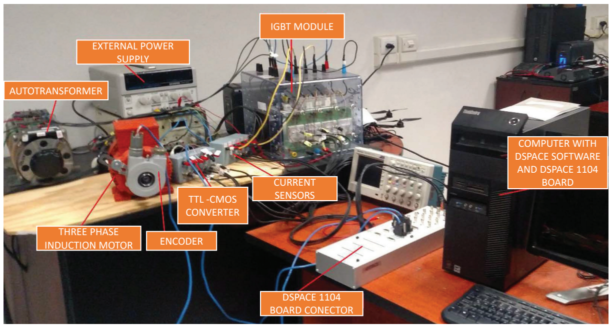

Fig. 2. Three-phase induction motor test bench.

IV. Real-time implementation

The experiments are performed using an induction motor test bench that includes a PC for supervision; a pulse-width modulated (PWM) three-phase IGBT inverter module as the power stage; a dSPACE DS1104 board (dSPACE is a registered trademark of dSPACE GmbH, Germany) that allows the applications designed in this work for data acquisition and control to be downloaded directly from Simulink (Matlab and Simulink are registered trademarks of MathWorks Inc., USA); and a three-phase induction motor as the plant to be controlled with the following characteristics: 220 V, 60 Hz, 0.19 kW, 1660 rpm, and 1.3 A [12]. The picture of the test bench is shown in Figure 2 and which shows the entire test bench used for experiments. The motor is controlled by the neural scheme explained in Section 2 with an unknown load torque as disturbance, as well as input uncertainties, state uncertainties and noise.

Figure 3 presents the trajectory tracking for a time-varying speed reference signal, it is important to note that during $8.4\text{s} \leq t \leq 8.9\text{s}$ a heavy load variation is applied that stops the rotor movement, however, the disturbance is compensated immediately resulting in a fast recovery of the trajectory tracking performance of the system which shows the resilience capability of the proposed controller. In Figure 6 the norm of online NN weights evolution is presented. Finally, Figure 7 shows the rotor speed trajectory tracking for a trapezoidal rotor speed trapezoidal profile. This profile is of importance because it is designed online. This is similar to a user defining a required speed for a desired task in real-time.

V. Conclusions

This work presents a direct neural controller for unknown discrete-time nonlinear systems with input and state uncertainties. The proposed controller has been experimentally tested with a three-phase induction motor to show that the trajectory tracking errors are small enough for time-varying reference signals. Despite all this, the proposed neural control scheme exhibits an excellent real-time performance with a sample time equal to 0.5ms. As seen from the results, the tracking of the reference signal is effectively achieved even though a sinusoidal and trapezoidal waveform represents a difficult task for induction motors.

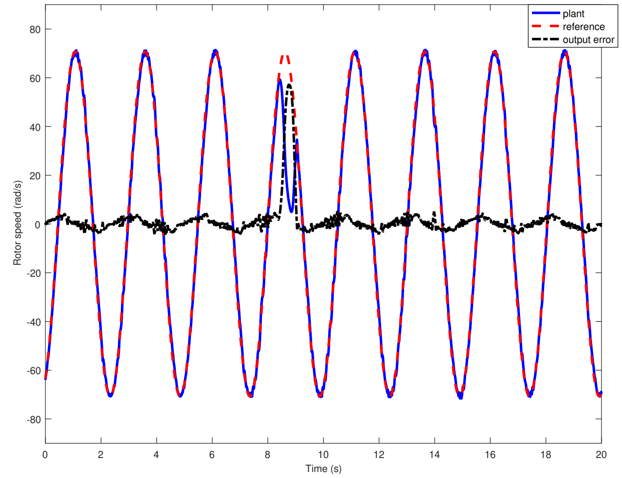

Fig. 3. Rotor speed trajectory tracking.

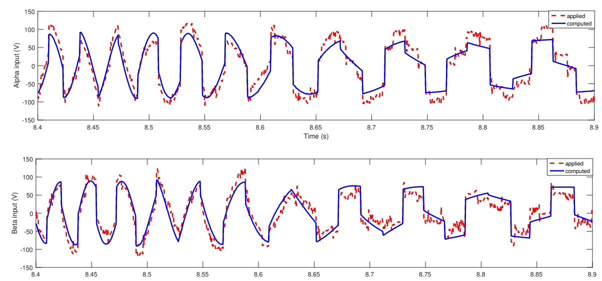

Fig. 4. Applied and computed control law.

Acknowledgements

The authors thank the University of Guadalajara for its support in this work.

References

[1] Dorf RC, Bishop RH. *Modern control systems*. Upper Saddle River, NJ: Prentice Hall; 2017.

[2] Antsaklis P. *Defining intelligent control Report of the Task Force on Intelligent Control*. 1994.

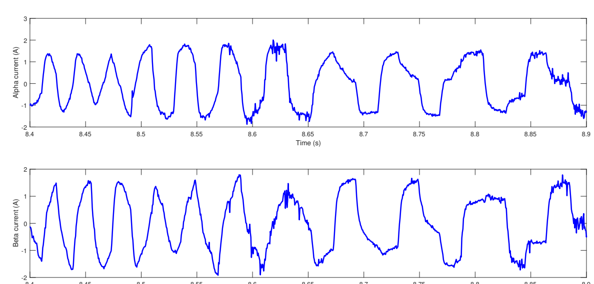

Fig. 5. Measured currents in $\alpha - \beta$ axis.

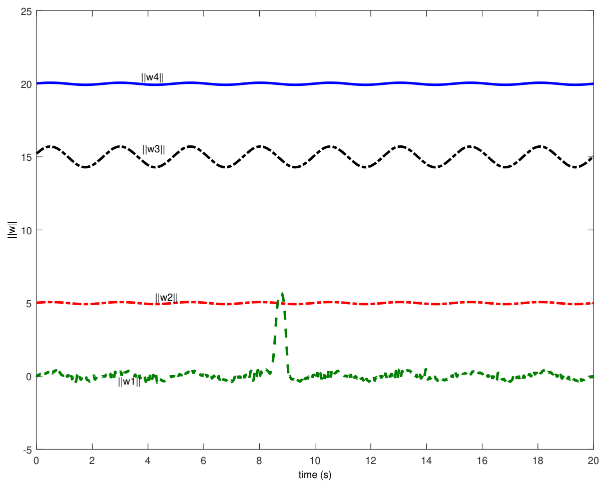

Fig. 6. Norm neural network weights evolution.

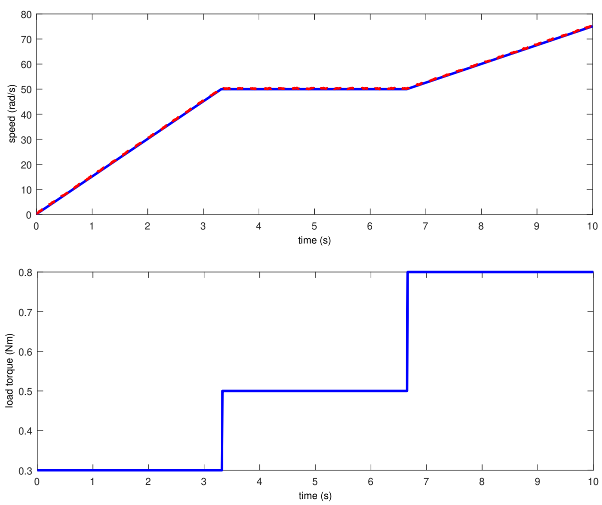

Fig. 7. Rotor speed trajectory tracking for an online time-varying reference profile.

[3] Norgaard M. *Neural Networks for Modelling and Control of Dynamic Systems: A Practitioner's Handbook*. Berlin, Germany: Springer; 2003.

[4] Rios JD, Alanis AY, Arana-Daniel N, Lopez-Franco C. Real-time neural observer-based controller for unknown nonlinear discrete delayed systems. *Int J Robust Nonlinear Control*. 2020;30(8):8402– 8429. https://doi.org/10.1002/rnc.5250

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

[6] Chen F, Khalil H. Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Transactions on Automatic Control*. 1995;40(5):791-801. https://doi.org/10.1109/9.384214.

[7] Ge SS, Zhang J, Lee TH. 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*. 2004;34(4);1630-1645. https://doi.org/10.1109/TSMCB.2004.826827.

[8] Ge SS, Lee TH, Harris CJ. *Adaptive Neural Network Control for Robotic Manipilators*. Singapore: World Scientific; 1998.

[9] Cybenko G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*. 1989;2:304-314. https://doi.org/10.1007/BF02551274

[10] Rovithakis GA, Chistodoulou MA. *Adaptive Control with Recurrent High -Order Neural Networks*. New York, USA: Springer Verlag; 2000.

[11] Alanis AY, Sanchez EN, Loukianov AG. Discrete-Time Adaptive Backstepping Nonlinear Control via High-Order Neural Networks. IEEE Transactions on Neural Networks. 2007;18(4):1185-1195. https://doi.org/10.1109/TNN.2007.899170.

[12] Alanis AY, Sanchez EN, Loukianov AG, Perez-Cisneros MA. Real-Time Discrete Neural Block Control Using Sliding Modes for Electric Induction Motors. IEEE Transactions on Control Systems Technology. 2010;18(1):11-21. https://doi.org/10.1109/TCST.2008.2009466.

[13] Quintero-Manriquez E, Sanchez EN, Felix RA. Induction motor torque control via discrete-time sliding mode, *2016 World Automation Congress (WAC)*; Rio Grande, USA; 2016. https://doi.org/10.1109/10.1109/WAC.2016.7582984.

[14] Khorrami F, Krishnamurthy P, Melkote H. *Modeling and Adaptive Nonlinear Control of Electric Motors*. Berlin, Germany: Springer-Verlag; 2003.

[15] Blanchini F, Casagrande D, Miani S, Viaro U. Robust linear parameter-varying control of induction motors. *Int. J. Robust Nonlinear Control*. 2015;25:1783–1800. https://doi.org/10.1002/rnc.3174.

[16] Castillo-Toledo B, Di Gennaro S, Loukianov AG, Rivera J. Hybrid Control of Induction Motors via Sampled Closed Representations. *IEEE Transactions on Industrial Electronics*. 2008;55(10):3758-3771. https://doi.org/10.1109/TIE.2008.928117

[17] Kumar A, Ramesh T. Direct Field Oriented Control of Induction Motor Drive. *Second International Conference on Advances in Computing and Communication Engineering*; Dehradun, India; 2015. https://doi.org/10.1109/ICACCE.2015.55.