

Neural Identifier-Control Scheme for Nonlinear Discrete Systems with Input Delay

Jorge D. Rios, Alma Y. Alanís^(✉), Nancy Arana-Daniel,
and Carlos López-Franco

Universidad de Guadalajara, Centro Universitario de Ciencias Exactas e
Ingenierías, Blvd. Marcelino García Barragán 1421, Guadalajara, Mexico
almayalanis@gmail.com

Abstract. This work presents a scheme based on a discrete recurrent high order neural network identifier and a block control based on sliding modes for nonlinear discrete-time systems with input delays in real-time. The identifier is trained with an extended Kalman Filter based algorithm and the block control is used for trajectory tracking. Experimental results are included using a linear induction motor prototype with added delays to its input signals.

Keywords: Extended Kalman filter training · Neural block control · Neural identification · Real-time · Time-delay

1 Introduction

Delays in systems are a source of instability and poor performance, also, they make system analysis a more complex task [1, 2]. Time delay systems mainly inherit delay from their components and examples can be found easily in areas like chemical industry, hydraulic systems, metallurgical processing and network systems [1, 2].

System identification is a process to obtain a mathematical model of a system from data obtained from a practical experiment with the system [3]. Among the many techniques for system identification, neural networks stand up [3, 4].

Recurrent high order neural networks (RHONNs) internal connections allow them to capture the response of complex nonlinear systems and to have characteristics like robustness against noise, on-line and off-line training, and the possibility of incorporating a priori information about the system to identified [5–7]. On the other hand, training of neural networks with Kalman filter algorithms has proved to be reliable and practical, also, it offers advantages for the improvement of learning convergence and computational efficiency compared to backpropagation methods [5, 6].

Neural block control is a methodology which uses a neural identifier of the block controllable form of a system, then, based on this model a discrete control law is designed combining discrete-time block-control and sliding modes technique [5].

There are a number of methodologies which work with systems with input delay [8–11]. The main disadvantages of these methodologies are that they need a lot of information about the system, in our methodology, there is not necessary to know the

model of the system because it works with the model obtained in the neural identification process. Moreover, most of them work in continuous-time which could be seen as a disadvantage due to the tendency towards digital rather than analog systems [12]. In this way, we present a rather simple to work with a scheme for discrete-time systems with input delays which can be used in real time even if the model of the system is unknown or incomplete.

On the other hand, compared with some of our previous works [13–15] this paper differs in that none of them treat the case of input delay in the system and some do not even consider any kind of delay.

The paper outline: Sect. 2 is dedicated to neural identification using RHONNs and extended Kalman filter (EKF) training. Then, the block control is in Sect. 3, results are shown in Sect. 4. Finally, the conclusions are included in Sect. 5.

2 Neural Identification

Neural identification is a process to obtain a mathematical model of a system by selecting a neural network and an adaptation law, in a way that the neural network responds in the same way to an input as the system to be identified [3].

2.1 Recurrent High Order Neural Network Identification

In this work, we use the following RHONN series-parallel model:

$$\widehat{x}_i(k + 1) = \omega_i^T z_i(x(k), u(k - l)) \quad i = 1, \dots, n \tag{1}$$

where n is the state dimension, \widehat{x} is the neural network state vector, ω is the weight vector, x is the plant state vector, and u is the input vector to the neural network, l is the unknown time delay and $z_i(\cdot)$ is defined as follows:

$$\begin{aligned} z_i(x(k), u(k - l)) &= [z_{i_1} \quad z_{i_2} \quad \dots \quad z_{i_{L_i}}]^T \\ &= \left[\prod_{j \in I_1} \zeta_{ij}^{d_{ij}(1)} \quad \prod_{j \in I_1} \zeta_{ij}^{d_{ij}(2)} \quad \dots \quad \prod_{j \in I_{L_i}} \zeta_{ij}^{d_{ij}(L_i)} \right]^T \end{aligned} \tag{2}$$

$$\zeta_i = [\zeta_{i_1} \quad \dots \quad \zeta_{i_n} \quad \zeta_{i_{n+1}} \quad \dots \quad \zeta_{i_{n+m}}]^T = [S(x_1) \quad \dots \quad S(x_n) \quad u_1 \quad \dots \quad u_m]^T \tag{3}$$

with L_i as the respective number of high-order connections, $\{I_1, I_2, \dots, I_{L_i}\}$ is a collection of non-ordered subsets of $\{1, 2, \dots, n + m\}$, $d_{ij}(k)$ being non-negative integers and $1/(1 + e^{-\beta v})$ with $\beta > 0$ and v is any real value variable.

EKF Training Algorithm. The training goal is to find the optimal weight vector which minimizes the prediction error. In this way, the weights ω become the states to be estimated by the Kalman filter, and the identification error between x and \widehat{x} is

considered as additive white noise [4]. The training algorithm is based on the EKF due to the nonlinearity of the neural network (1) and is defined in (4).

$$\begin{aligned} \omega_i(k+1) &= \omega_i(k) + \eta K_i(k)(x_i(k) - \hat{x}_i(k)) \\ K_i(k+1) &= P_i(k)H_i(k) [R_i(k) + H_i(k)^T P_i(k)H_i(k)]^{-1} \\ P_i(k+1) &= P_i(k) - K_i(k)H_i(k)^T P_i(k) + Q_i(k) \end{aligned} \tag{4}$$

where $\omega_i \in \mathcal{R}^{L_i}$ is the adapted weight vector, $\eta \in \mathcal{R}$ is the learning rate, \hat{x}_i is the i -th state variable of the neural network, $K_i \in \mathcal{R}^{L_i}$ is the Kalman gain vector, $R_i \in \mathcal{R}$ is the error noise covariance, $H_i \in \mathcal{R}^{L_i}$ is vector with entries $H_{ij} = [\partial \hat{x}_i(x) / \partial \omega_{ij}(k)]^T$ and $P_i \in \mathcal{R}^{L_i \times L_i}$ is the weight estimation error covariance matrix, $Q_i \in \mathcal{R}^{L_i \times L_i}$ is the estimation noise covariance matrix. P_i and Q_i are initialized as diagonal matrices with entries $P_i(0)$ and $Q_i(0)$, respectively.

RHONN Identification. Consider the following Nonlinear Discrete-Time System with input delay:

$$\begin{aligned} x(k+1) &= F(x(k), u(k-l)) \\ y(k) &= h(x(k)) \end{aligned} \tag{5}$$

where $x \in \mathcal{R}^n$, $u \in \mathcal{R}^m$, $F \in \mathcal{R}^n \times \mathcal{R}^m \rightarrow \mathcal{R}^n$ is a nonlinear function and $l = 1, 2, \dots$ is the unknown delay. Then, our identification process consists of approximating the system (5) with the RHONN (1) trained online with the EKF algorithm (4).

This identification process is validated achieving a small error between the system outputs and the identifier outputs for the same inputs.

3 Neural Block Control

The model of many practical nonlinear systems can be transformed in the block controllable form [5]:

$$\begin{aligned} x_j(k+1) &= f_j(\bar{x}_j(k)) + B_j(\bar{x}_j(k))x_{j+1}(k) + d_j(k) \\ x_r(k+1) &= f_r(x(k)) + B_r(x(k))u(k) + d_r(k) \\ y(k) &= x_1(k) \end{aligned} \tag{6}$$

where $j = 1, \dots, r-1$, $x \in \mathcal{R}^n$ is the state variable vector with $x(k) = [x_1(k) \cdots x_r(k)]^T$, $\bar{x}_j = (k)[x_1(k) \cdots x_j(k)]^T$, $r \geq 2$ is the number of blocks, $u \in \mathcal{R}^m$, $d(k) = [d_1(k) \cdots d_j(k) \cdots d_r(k)]^T$ is the bounded unknown disturbance vector and $f_j(\cdot)$ and $B_j(\cdot)$ are smooth nonlinear functions. Consider the following transformation [5]:

$$\begin{aligned}
\chi_1(k) &= x_1(k) - x_1^d(k) \\
\chi_2(k) &= x_2(k) - x_2^d(k) \\
&= x_2(k) - [B_1(x_1(k))]^{-1}(K_1 z_1(k) - (f_1(x_1(x)) - d_1)) \\
&\quad \vdots \\
\chi_r(k) &= x_r(k) - x_r^d(k)
\end{aligned} \tag{7}$$

where x_1^d is the tracking reference, x_i^d is the desired value for x_i ; and K is a Shur matrix. Using (7) and selecting $S_D(k) = \chi_r(k) = 0$ system (6) can be rewritten as (8):

$$\begin{aligned}
\chi_1(k+1) &= K_1 \chi_1(k) + B_1 \chi_2(k) \\
&\quad \vdots \\
\chi_{r-1}(k+1) &= K_{r-1} \chi_{r-1}(k) + B_{r-1} \chi_r(k) \\
\chi_r(k+1) &= f_r(x(k)) + B_r(x(k))u(k) + d_r(k) - x_r^d(k+1)
\end{aligned} \tag{8}$$

then, $u(k)$ is defined in (9), where u_{eq} is calculated from $S_D(k+1) = 0$ and u_0 it is the control resources that bound the control.

$$\begin{aligned}
u(k) &= \begin{cases} u_{eq}(k) & \text{if } \|u_{eq}(k)\| \leq u_0 \\ u_0 \frac{u_{eq}(k)}{\|u_{eq}(k)\|} & \text{if } \|u_{eq}(k)\| > u_0 \end{cases} \\
U_{eq}(k) &= [B_r(x(k))]^{-1}(-f_r(x(k)) + x_r^d(k+1) - d_r(k))
\end{aligned} \tag{9}$$

Hence, the first step of the process is to design a RHONN identifier in a block controllable form for the system to be identified and then obtained the $u(k)$ as in (9).

4 Results

Test Description. Using the Linear Induction Motor prototype (Fig. 1) which is based in a dSPACE® board RTI1104 and a MATLAB®/Simulink® interface the neural block control is implemented in a Simulink model with communication to the prototype by the dSPACE® tools. A subsystem to induce delays in the system input is created in Simulink®. The subsystem consists of that 4 s after the prototype starting the control signal is switched to a version with random time-delay. This is achieved using the block “*Variable Transport Delay*” with the configuration *variable time delay*

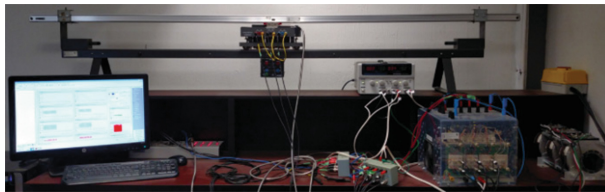


Fig. 1. Linear induction motor prototype

which receives as input at each time a random number from 0 to 10 multiplied by the sampling time set to 0.0003 s.

Experimental Results. Table 1 shows the identification errors for all variable states and Fig. 2 shows two graphs the first one shows the velocity tracking, the second one shows the tracking of the flux magnitude which is defined as:

$$flux\ magnitude = Alpha\ Flux^2 + Beta\ Flux^2$$

Table 1. Root mean square errors of identification

State variable	RMSE	State variable	RMSE
Position	5.86×10^{-5}	Beta flux	4.91×10^{-5}
Velocity	1.43×10^{-4}	Alpha current	1.46
Alpha flux	4.78×10^{-5}	Beta current	1.005

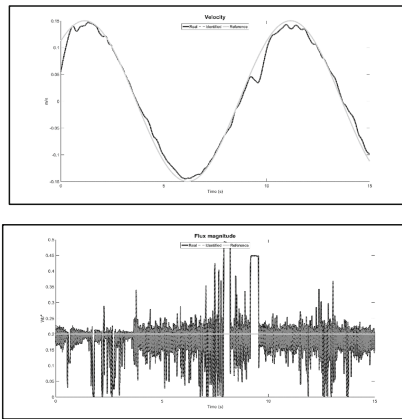


Fig. 2. Tracking reference real-time performance

5 Conclusions

In general, it is seen that the proposed scheme adapts itself quickly even in the presence of real-time disturbances and the added delays in the input to the system. More specifically, the errors shown in Table 1 are small for all variable states, even for alpha and beta currents considering that they real values can be as high as 30 A. Figure 2 shows the velocity tracking with a good performance and a flux magnitude which is maintained around its reference. Also, when the added time delay starts at 4 s it is noticeable that the performance changes, however, the presented RHONN identifier – control scheme is still capable of maintaining the dynamic of the desired trajectory. Moreover, it is important to note that for real-time tests a series of external parameters

are involved like imperfections of the prototype components which induce noise to the lecture of the signals. We are working on improving our scheme and available equipment to test it.

Acknowledgments. The authors thank the support of CONACYT Mexico, through Projects CB256769 and CB258068 (“*Project supported by Fondo Sectorial de Investigación para la Educación*”).

References

1. Boukas, E., Liu, Z.: *Deterministic and Stochastic Time-Delay Systems*. Birkhauser, Boston (2012)
2. Mahmoud, M.: *Switched Time-Delay Systems: Stability and Control*. Springer, Heidelberg (2014)
3. Norgaard, M., Ravn, O., Poulsen, N., Hansen, L.: *Neural Networks for Modelling and Control of Dynamic System*. Springer, New York (2000)
4. Fu, L., Li, P.: The research survey of system identification. In: *5th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, pp. 397–401. IEEE (2013)
5. Sanchez, E., Alanis, A., Loukianov, A.: *Discrete-Time High Order Neural Control*. Springer, Berling (2008)
6. Haykin, S.: *Neural Networks and Learning Machines*. Prentice Hall, Upper Saddle River (2009). International
7. Rovithakis, G., Christodoulou, M.: *Adaptive Control with Recurrent High-order Neural Networks: Theory and Industrial Applications*. Springer, London (2012)
8. Liu, M.: Neural network based fault tolerant control of a class of nonlinear systems with input time delay. In: *Advances in Neural Networks - ISNN 2004: International Symposium on Neural Networks*, Dalian, China, pp. 91–96. Springer (2004)
9. Spandan, R., Indra, K.: Adaptive robust tracking control of a class of nonlinear systems with input delay. *Nonlinear Dyn.* **85**(2), 1124–1139 (2016)
10. Li, L., Niu, B.: Adaptive neural network tracking control for switched strict-feedback nonlinear systems with input delay. In: *2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP)*, Wuhan, China (2015)
11. Li, H., Wang, L., Du, H., Boulkrone, A.: Adaptive fuzzy backstepping tracking control for strict-feedback systems with input delay. *IEEE Trans. Fuzzy Syst.* **25**, 642–652 (2017)
12. Ogata, K.: *Discrete-time Control Systems*. Prentice Hall, Upper Saddle River (1995). International
13. Alanis, A., Rios, J., Rivera, J., Arana-Daniel, N., Lopez-Franco, C.: Real-time discrete neural control applied to a linear induction motor. *Neurocomputing* **164**, 240–251 (2015)
14. Alanis, A., Rios, J., Arana-Daniel, N., Lopez-Franco, C.: Neural identifier for unknown discrete-time nonlinear delayed systems. *Neural Comput. Appl.* **27**(8), 2453–2464 (2016)
15. Rios, J., Alanis, A., Lopez-Franco, M., Lopez-Franco, C., Arana-Daniel, N.: Real-time neural identification and inverse optimal control for a tracked robot. *Adv. Mech. Eng.* **9**(3), 1–18 (2017)