Discrete-Time Decentralized Neural Identification and Control for a 2 DOF Robot Manipulator

R. Garcia-Hernandez [*] **E. N. Sanchez** [*] **A. G. Loukianov** [*]
E. Bayro-Corrochano [*] **V. Santibañez** [**]

[*] *CINVESTAV, Unidad Guadalajara, Apartado Postal 31-438, C.P. 44550, Guadalajara, Jalisco, Mexico.*
(e-mail:[rhernand][sanchez][louk][edb]@gdl.cinvestav.mx)
[**] *Instituto Tecnologico de la Laguna, Apartado Postal 49, Adm. 1, C.P. 27001, Torreon, Coahuila, Mexico.*
(e-mail:vsantiba@itlalaguna.edu.mx)

Abstract: This paper presents a discrete-time decentralized control scheme for identification and trajectory tracking of a 2 DOF robot manipulator. A recurrent high order neural network (RHONN) structure is used to identify the plant model and based on this model, a discrete-time control law is derived, which combines discrete-time block control and sliding modes techniques. The neural network learning is performed on line by Kalman filtering. A controller is designed for each joint, using only local angular position and velocity measurements. These simple local joint controllers allow trajectory tracking with reduced computations. The applicability of the proposed scheme is illustrated via simulations.

1. INTRODUCTION

Recently, control of robot manipulators has become a significant research area, for industrial and medical applications, due to the relevancy that these robots have acquired in performing tasks that are classified as dangerous or that require higher accuracy.

In this context, multiple control schemes have been proposed in order to guarantee efficient trajectory tracking and stability (Sanchez and Ricalde [2003]), (Santibañez et al. [2005]). The fast advance in the computational technology offers many ways for implementation of control algorithms within the approach of a centralized control design (Gourdeau [1997]). However, there is a great challenge to obtain an efficient control for this class of systems, due to its highly nonlinear complex dynamics, with strong interconnections, parameters difficult to measure, and dynamic difficult to model. Considering only the most important terms, the mathematical model obtained requires control algorithms with great number of mathematical operations, which affect the real-time implementation feasibility.

On the other hand, within the area of control systems theory, for more than three decades, an alternative approach has been developed in which a global system is conceived like a set of interconnected subsystems for which it is possible to design independent controllers, considering only inherent local variables to each subsystem: the decentralized control (Jiang [1999]), (Huang et al. [2003]). Decentralized control has been applied in robotics, mainly in cooperative multiple mobile robots and robots manipulators, where it is natural to consider each mobile robot or each manipulator as a sub-system of the whole system. In this approach, each joint is considered as a sub-system in order to develop local controllers, which just consider local angular position and angular velocity measurements, and compensate the interconnection effects, usually assumed as disturbances. The resulting controllers are easily implemented for real-time applications (Jin [1998]), (Liu [1999]).

In (Ni and Er [2000]), a decentralized control of robot manipulators is developed, decoupling the dynamic model of the manipulator in a set of linear subsystems with uncertainties and simulations for a robot of two joints are shown. In (Karakasoglu et al. [1993]), an approach of decentralized neural identification and control for robots manipulators are exposed using models in discrete-time. In (Safaric and Rodic [2000]), a decentralized control for robots manipulators is reported; it is based on the estimation of independent dynamics for each of the joints, using feedforward neural networks.

In this paper, we use an Extended Kalman Filter (EKF)-based training algorithm for a recurrent high order neural network (RHONN), in order to identify the plant model; based on this model, a discrete-time control is derived, which combines discrete-time block control and sliding modes techniques. The block control approach is used to design a nonlinear sliding surface such that the resulting sliding mode dynamics is described by a desired linear system (Alanis et al. [2006]). The proposed neural identifier and control applicability is illustrated by trajectory tracking for a 2 DOF robot manipulator.

2. DYNAMICS OF ROBOT MANIPULATORS

The dynamics of a serial n-link robot can be written as (Spong and Vidyasagar [1989])

$$\ddot{q} = M(q)^{-1}[\tau - C(q,\dot{q})\dot{q} - f(\dot{q}) - g(q)] \qquad (1)$$

where q is the $n \times 1$ vector of joint positions, $\dot{q}$ is the $n \times 1$ vector of joint velocities, $\ddot{q}$ is the $n \times 1$ vector of joint accelerations, τ is the $n \times 1$ vector of applied torques, $M(q)$ is the $n \times n$ symmetric positive definite manipulator inertia matrix, $C(q,\dot{q})\dot{q}$ is the $n \times 1$ vector of centripetal and Coriolis torques, $g(q)$ is the vector of gravitational torques obtained as the gradient of the robot potential energy $U(q)$ due to gravity, i.e.,

$$g(q) = \frac{\partial U(q)}{\partial (q)}$$

and $f(\dot{q})$ is the $n \times 1$ vector of friction torque. In the static models, friction is modeled by a vector $f(\dot{q}) \in \Re^n$ that depends only on the joint velocity $\dot{q}$ (Kelly et al. [2005]). Friction effects are local, that is, $f(\dot{q})$ may be written as

$$f(\dot{q}) = \begin{bmatrix} f_1(\dot{q}_1) \\ f_2(\dot{q}_2) \\ \vdots \\ f_n(\dot{q}_n) \end{bmatrix}.$$

A “classical” static friction model is one that combines the so-called viscous and Coulomb friction phenomena. This model establishes that the vector $f(\dot{q})$ is given by

$$f(\dot{q}) = F_{m1}\dot{q} + F_{m2}\,\mathrm{sgn}(\dot{q}) \qquad (2)$$

where F_{m1} and F_{m2} are $n \times n$ diagonal positive definite matrices. The elements of the diagonal of F_{m1} correspond to the viscous friction parameters while the elements of F_{m2} correspond to the Coulomb friction parameters. Furthermore in the model given by (2)

$$\mathrm{sgn}(\dot{q}) = \begin{bmatrix} \mathrm{sgn}(\dot{q}_1) \\ \mathrm{sgn}(\dot{q}_2) \\ \vdots \\ \mathrm{sgn}(\dot{q}_n) \end{bmatrix}$$

and $\mathrm{sgn}(\dot{q})$ is the sgn “function”, defined as follows

$$\mathrm{sgn}(\dot{q}) = \begin{cases} 1, & \text{if } \dot{q} > 0 \\ 0, & \text{if } \dot{q} = 0 \\ -1, & \text{if } \dot{q} < 0. \end{cases}$$

However, $\mathrm{sgn}(0)$ is undefined in the sense that one do not associate a particular real number to the “function” $\mathrm{sgn}(\dot{q})$ when $\dot{q} = 0$.

The model (1) does not completely represent the manipulator due to the inexact knowledge of the friction effects, the omission of some dynamics, and parameter uncertainties and variations (Karakasoglu et al. [1993]).

In order to apply a decentralized approach is supposed that the model (1) can be partitioned in N sub-systems, each one formulated as:

$$\begin{aligned} \ddot{q}_i &= f_i(q_i, \dot{q}_i, \tau_i) + \Gamma_i(q_1, q_2, \ldots, q_N, \\ &\quad \dot{q}_1, \dot{q}_2, \ldots, \dot{q}_N, \tau_1, \tau_2, \ldots, \tau_N) \end{aligned} \qquad (3)$$

where $i = 1, 2, \ldots, N$, $f_i(\cdot)$ depends only the local variables, and $\Gamma_i(\cdot)$ represents the interconnection effects.

The dynamics of mathematical model for robotic manipulators are bounded as

$$\begin{aligned} \|M(q)\ddot{q}\| &\le c_m \|\ddot{q}\| \\ \|C(q,\dot{q})\dot{q}\| &\le c_v \|\dot{q}\|^2 \\ \|f(\dot{q})\| &\le c_f \|\dot{q}\| \\ \|g(q)\| &\le c_g \end{aligned}$$

where c_m, c_v, c_f, and c_g are constants. This aspect is important for the possibility of using neural identification Fu [1992].

3. NEURAL IDENTIFIER

3.1 Decentralized Neural Identifier

The following recurrent high order neural network structure is proposed to identify (3):

$$\begin{aligned} x_{i1}(k+1) &= w_{i11}(k)S(\chi_{i1}(k)) + w_{i12}(k)\chi_{i2}(k) \\ x_{i2}(k+1) &= w_{i21}(k)S(\chi_{i1}(k)) + w_{i22}(k)S(\chi_{i2}(k)) \\ &\quad + w_{i23}(k)u_i(k) \end{aligned} \qquad (4)$$

where $i = 1, \ldots, N$, $x_{i1}(k)$ and $x_{i2}(k)$ are the states (neurons) of the i-th neural network, representing $\chi_{i1}(k)$ and $\chi_{i2}(k)$ (the i-th angular position and velocity) respectively; w_{ij} is the respective synaptic weight of the $S(\cdot)$ activation function, and $u_i(k)$ represents the applied torque to the i-th joint.

It is worth to note that (4), as defined, constitutes a series-parallel identifier and fulfills the conditions of the Block Controllable Form (BCF) (Utkin [2000]), with two blocks of dimension one. This is also a particular case of the more general one discussed in (Benitez et al. [2003]).

3.2 EKF Training Algorithm

It is known, that Kalman filtering (KF) estimates the state of a linear system with additive state and output white noises (Grover and Hwang [1992]), (Haykin [2001]). For KF-based NN training, the network weights become the states to be estimated. In this case, the error between the NN output and the measured plant output can be considered as additive white noise. Due to the fact that NN mapping is nonlinear, an EKF-type is required.

The training goal is to find the optimal weight values which minimize the prediction error. We use a decoupled EKF-based training algorithm described by:

$$\begin{aligned} K_j(k) &= P_j(k)H_j(k)M_j(k) \\ w_j(k+1) &= w_j(k) + \eta_j K_j(k) e_j(k) \\ P_j(k+1) &= P_j(k) - K_j(k)H_j^T(k)P_j(k) + Q_j(k) \\ j &= 1, \ldots, n \end{aligned} \qquad (5)$$

with

$$\begin{aligned} M_j(k) &= [R_j(k) + H_j^T(k)P_j(k)H_j(k)]^{-1} \\ e_j(k) &= [\chi_j(k) - x_j(k)] \end{aligned} \qquad (6)$$

where $e_j(k) \in \Re^m$ is the identification error, $P_j(k) \in \Re^{L_j \times L_j}$ is the prediction error covariance matrix, $w_j(k) \in$

$\Re^{L_j}$ is the weight (state) vector, η_j is the rate learning parameter such that $0 \le \eta_j \le 1$, L_j is the respective number of NN weights, $\chi_j(k) \in \Re^m$ is the j-th plant state, $x_j(k) \in \Re^m$ is the j-th neural network state, n is the number of states, $K_j(k) \in \Re^{L_j \times m}$ is the Kalman gain matrix, $Q_j(k) \in \Re^{L_j \times L_j}$ is the measurement noise covariance matrix, $R_j(k) \in \Re^{m \times m}$ is the state noise covariance matrix, and $H_j(k) \in \Re^{L_j \times m}$ is a matrix, in which each entry (H_{jk}) is the derivative of j-th NN state $(x_j(k))$, with respect to jk-th neural network weight (w_{jk}), as follows

$$H_{jk}(k) = \left[\frac{\partial x_j(k)}{\partial w_{jk}(k)} \right]^T_{w_j(k)=w_j(k+1)}, \tag{7}$$

where $j = 1, \ldots, n$ and $k = 1, \ldots, L_j$. Usually P_j and Q_j are initialized as diagonal matrices, with entries $P_j(0)$ and $Q_j(0)$, respectively. It is important to remark that $H_j(k)$, $K_j(k)$, and $P_j(k)$ for the EKF are bounded (Song and Grizzle [1995]).

4. DESIGN CONTROLLER

Consider the nonlinear system defined as

$$\begin{aligned} x(k+1) &= f(x(k)) + B(x(k))u(k) + d(k) \\ y(k) &= Cx(k) \end{aligned} \tag{8}$$

where $x(k) \in \Re^n$ 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, the vector $f(\cdot)$, the columns of $B(\cdot)$ and $d(\cdot)$ are smooth vector fields, and $d(k)$ is a disturbance vector.

By means of a non-singular transformation (Utkin et al. [1999]), system (8) can be represented in the block controllable form consisting of r blocks (Loukianov [2003]):

$$\begin{aligned} x_i(k+1) &= f_i(\overline{x}_i(k)) + B_i(\overline{x}_i(k))x_{i+1}(k) + d_i(k) \\ x_r(k+1) &= f_r(x(k)) + B_r(x(k))u(k) + d_r(k) \\ y(k) &= x_1(k), i = 1, \ldots, r-1 \end{aligned} \tag{9}$$

where $x(k) = [x_1(k) \ldots x_i(k) \ldots x_r(k)]^T$, $d(k) = [d_1(k) \ldots d_i(k) \ldots d_r(k)]^T$ and $\overline{x}_i(k) = [x_1(k) \ldots x_i(k)]^T$, and the sets of numbers $(n_1, n_2, \ldots n_r)$, which define the structure of system (9), satisfy $n_1 \le n_2 \le \ldots \le n_r \le m$.

Define the following transformation

$$\begin{aligned} z_1(k) &= x_1(k) - x_1^d(k) \\ z_2(k) &= x_2(k) - x_2^d(k) \\ z_2(k) &= x_2(k) - [B_1(x_1(k))]^{-1}(K_1 z_1(k) \\ &\quad -(f_1(x_1(k)) - d_1(k))) \\ &\vdots \\ z_r(k) &= x_r(k) - x_r^d(k) \end{aligned} \tag{10}$$

with $y_d(k) = x_1^d(k)$ as the desired trajectory for tracking. x_i^d is the desired value for x_i $(i = 1, \ldots, r)$, and $K_i = diag$ $\{k_{i_1}, \ldots, k_{i_n}\}$ with $|k_{i_n}| < 1$.

$$\begin{aligned} z_1(k+1) &= K_1 z_1(k) + B_1 z_2(k) \\ &\vdots \\ z_{r-1}(k+1) &= K_{r-1} z_{r-1}(k) + B_{r-1} z_r(k) \\ z_r(k+1) &= f_r(x(k)) + B_r(x(k))u(k) \\ &\quad + d_r(k) - x_r^d(k+1). \end{aligned} \tag{11}$$

To design the control law, we use the sliding mode block control technique. The surface is derived from the block control procedure, and a natural selection for the sliding surface $S_D(k) = 0$ is $S_D(k) = z_r(k) = 0$. Thus, system (11) is represented as

$$\begin{aligned} z_1(k+1) &= K_1 z_1(k) + B_1 z_2(k) \\ &\vdots \\ z_{r-1}(k+1) &= K_{r-1} z_{r-1}(k) + B_{r-1} S_D(k) \\ S_D(k+1) &= f_r(x(k)) + B_r(x(k))u(k) \\ &\quad + d_r(k) - x_r^d(k+1). \end{aligned} \tag{12}$$

Once the sliding surface is selected, next step is to define $u(k)$, as

$$u(k) = \begin{cases} u_{eq}(k) & \text{for } \|u_{eq}(k)\| \le u_0 \\ u_0 \dfrac{u_{eq}(k)}{\|u_{eq}(k)\|} & \text{for } \|u_{eq}(k)\| > u_0 \end{cases} \tag{13}$$

where the equivalent control is calculated from $S_D(k+1) = 0$, as

$$\begin{aligned} u_{eq}(k) &= [B_r(x(k))]^{-1} \\ &\quad (-f_r(x(k)) + x_r^d(k+1) - d_r(k)) \end{aligned} \tag{14}$$

and u_0 is the control resources that bound the control.

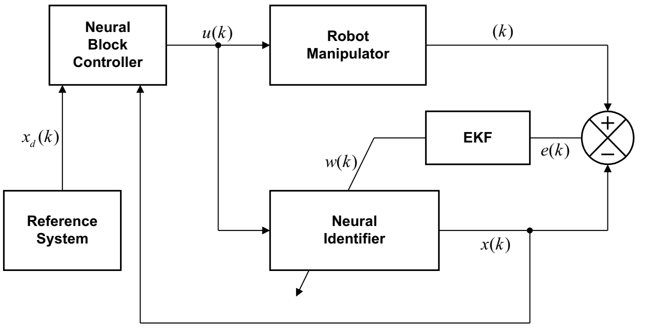

Fig. 1. Neural block control scheme

The whole proposed identification and control scheme is displayed in Fig. 1. The stability analysis to prove that the closed-loop system motion over the surface is stable is presented in (Alanis et al. [2006]).

5. SIMULATIONS RESULTS

5.1 Experimental Setup

In order to evaluate via numerical simulations the performance of proposed control algorithm, we use a dynamical model of a 2 DOF robot manipulator moving in the vertical plane as shown in Fig. 2 (Kelly and Santibañez [2003]). The robot manipulator consists of two rigid links; high-torque brushless direct-drive servos are used to drive the joints without gear reduction. This kind of joints presents reduced backlash and significantly lower joint friction as compared to the actuators with gear drives. The motors used in the experimental arm are DM1200-A and DM1015-B from Parker Compumotor, for the shoulder and elbow joints, respectively. Angular information is obtained from incremental encoders located on the motors, which have a resolution of 1,024,000 pulses/rev for the first motor and 655,300 for the second one (accuracy 0.0069° for both motors), and the angular velocity information is computed via numerical differentiation of the angular position signal.

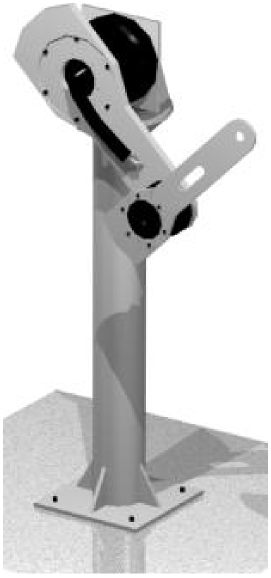

Fig. 2. Robot manipulator

5.2 Design Parameters

The goal is to track a desired reference signal, which is achieved by designing a control law based on the sliding mode technique described in section 4. The tracking error is defined as

$$z_{i1}(k) = x_{i1}(k) - x_{i1}^d(k) \tag{15}$$

where x_{i1}^d is the desired trajectory signal. Using (4) and introducing the desired dynamics for $z_{i1}(k)$ we have

$$\begin{aligned} z_{i1}(k+1) &= w_{i11}(k)S(\chi_{i1}(k)) + w_{i12}(k)\chi_{i2}(k) \\ &\quad -x_{i1}^d(k+1) = k_{i1}z_{i1}(k). \end{aligned} \tag{16}$$

The desired value $x_{i2}^d(k)$ for $x_{i2}(k)$ is calculated from (16) as

$$\begin{aligned} x_{i2}^d(k) &= \frac{1}{w_{i12}(k)}[-w_{i11}(k)S(\chi_{i1}(k)) \\ &\quad +x_{i1}^d(k+1) + k_{i1}z_{i1}(k)]. \end{aligned} \tag{17}$$

At the second define a new variable as

$$z_{i2}(k) = x_{i2}(k) - x_{i2}^d(k). \tag{18}$$

Taking one step ahead, we have

$$\begin{aligned} z_{i2}(k+1) &= w_{i21}(k)S(\chi_{i1}(k)) + w_{i22}(k)S(\chi_{i2}(k)) \\ &\quad +w_{i23}(k)u_i(k) - x_{i2}^d(k+1) \\ &= k_{i2}z_{i2}(k). \end{aligned} \tag{19}$$

The manifold for the sliding mode is chosen as $\mathbf{S}_i(k) = z_{i2}(k) = 0$. The control law is given by

$$u_i(k) = \begin{cases} u_{eq_i}(k) & \text{for } \|u_{eq_i}(k)\| \le \tau_i^{\max} \\ \tau_i^{\max}\dfrac{u_{eq_i}(k)}{\|u_{eq_i}(k)\|} & \text{for } \|u_{eq_i}(k)\| > \tau_i^{\max} \end{cases} \tag{20}$$

where $u_{eq_i}(k)$ is obtained from $\mathbf{S}_i(k+1) = 0$ as

$$\begin{aligned} u_{eq_i}(k) &= \frac{1}{w_{i23}(k)}[-(w_{i21}(k)S(\chi_{i1}(k)) \\ &\quad +w_{i22}(k)S(\chi_{i2}(k))) + x_{i2}^d(k+1)]. \end{aligned} \tag{21}$$

According to the actuators manufacturer, the direct-drive motors are able to supply torques within the following bounds:

$$\begin{aligned} |\tau_1| \le \tau_1^{\max} &= 150 \text{ [Nm]} \\ |\tau_2| \le \tau_2^{\max} &= 15 \text{ [Nm]}. \end{aligned}$$

For simulations we choose the following discrete-time trajectories as

$$\left.\begin{aligned} x_{11}^d(k) &= b_1(1-e^{d_1kT^3}) + c_1(1-e^{d_1kT^3})\sin(\omega_1kT) \\ x_{21}^d(k) &= b_2(1-e^{d_2kT^3}) + c_2(1-e^{d_2kT^3})\sin(\omega_2kT) \end{aligned}\right.\text{[rad]}$$

where $b_1 = \pi/4$, $c_1 = \pi/18$, $d_1 = -2.0$, and $\omega_1 = 7.5$ [rad/s] are parameters of the desired position trajectory for the first joint, whereas $b_2 = \pi/3$, $c_2 = 25\pi/36$, $d_2 = -1.8$, and $\omega_2 = 1.75$ [rad/s] are parameters of the desired position trajectory for the second joint with a sampling period $T = 2.5$ milliseconds.

These trajectories present the following characteristics: a) Incorporate a sinusoidal term to evaluate the performance before relatively fast periodic signals, where the non-linearities of the robot dynamics are really important and b) Present a term that smoothly grow for maintenance the robot in an operation state without saturating actuators whose limit are in 150 [Nm] and 15 [Nm], respectively. The trajectory tracking results are presented in Fig. 3 and Fig. 4. The identification and tracking performance can be verified for the neural identifier and the plant outputs, respectively.

Figs. 5 and 6 display the angular position identification errors for joints 1 and 2. Figs. 7 and 8 shown the angular position tracking errors for joints 1 and 2.

The applied torques to each joint are shown in Figs. 9 and 10. It is easy to see that both control signals are always inside of the prescribed limits given by the actuators manufacturer, that is, their absolute values are smaller than the limits $\tau_1^{\max}$ and $\tau_2^{\max}$, respectively.

6. CONCLUSIONS

A decentralized neural identification and control scheme for trajectory tracking is applied to a discrete-time 2 DOF

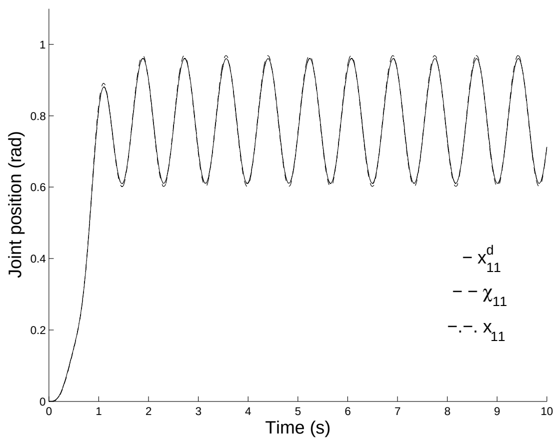

Fig. 3. Identification and tracking for joint 1 $x^d_{11}(k)$ (solid line), $\chi_{11}(k)$ (dashed line), and $x_{11}(k)$ (dashdot line)

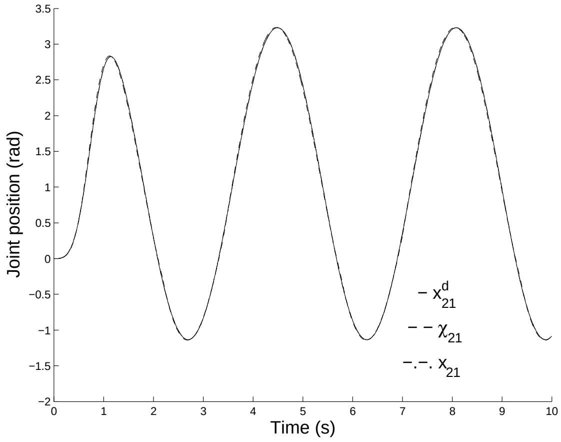

Fig. 4. Identification and tracking for joint 1 $x^d_{21}(k)$ (solid line), $\chi_{21}(k)$ (dashed line), and $x_{21}(k)$ (dashdot line)

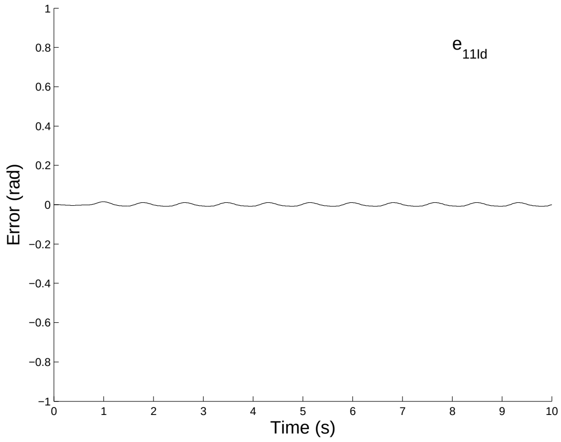

Fig. 5. Identification error for joint 1

robot manipulator model. The training of the recurrent high-order neural network is performed on-line using an extended Kalman filter in a series-parallel configuration. Simulation results illustrate the applicability of the proposed control methodology. The research proceeds to test in real-time the whole scheme.

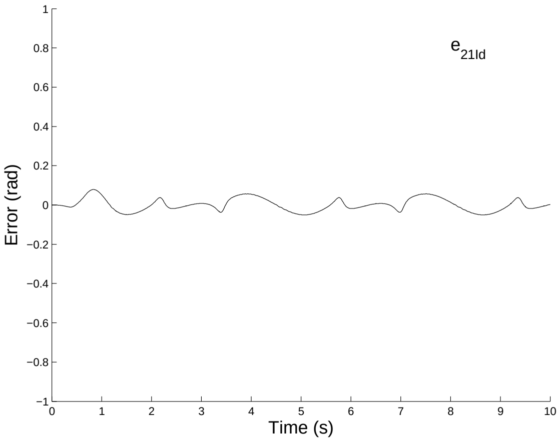

Fig. 6. Identification error for joint 2

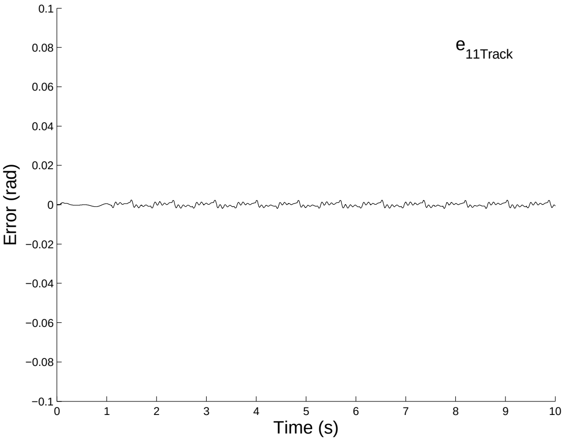

Fig. 7. Tracking error for joint 1

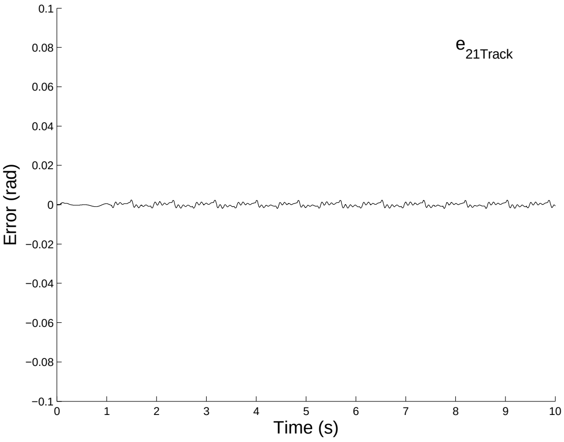

Fig. 8. Tracking error for joint 2

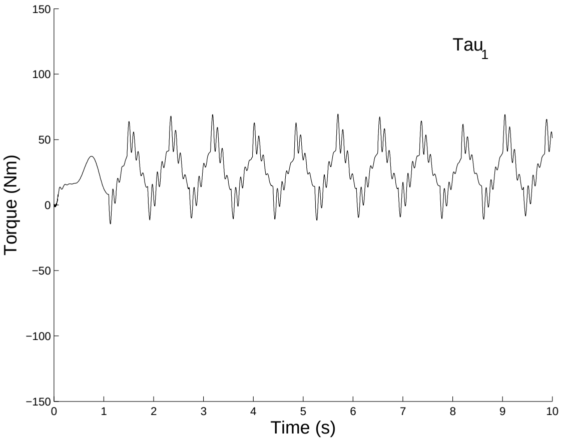

Fig. 9. Applied torque to the joint 1

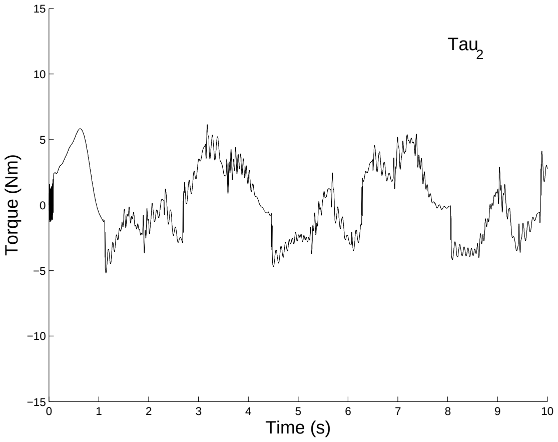

Fig. 10. Applied torque to the joint 2

ACKNOWLEDGEMENTS

The first author thanks to Universidad Autonoma del Carmen (UNACAR) and the PROMEP for supporting this research.

REFERENCES

A.Y. Alanis, E.N. Sanchez, A.G. Loukianov, and Guanrong Chen. Discrete-time output trajectory tracking by recurrent high-order neural network control. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6367–6372, San Diego, CA, USA, 2006.

V.H. Benitez, A.G. Loukianov, and E.N. Sanchez. Neural identification and control of a linear induction motor using α-β model. In *Proceedings of the American Control Conference*, pages 4041–4046, Denver, Colorado, USA, 2003.

L.-Ch. Fu. Robust adaptive decentralized control of robot manipulators. *IEEE Transactions on Automatic Control*, 37(1):106–110, 1992.

R. Gourdeau. Object-oriented programming for robotic manipulator simulation. *IEEE Robotics and Automation*, 4(3):21–29, 1997.

R. Grover and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons, Inc, New York, 1992.

S. Haykin. *Kalman Filtering and Neural Networks.* John Wiley and Sons, Inc, New York, 2001.

S. Huang, K.K. Tan, and T.H. Lee. Decentralized control design for large-scale systems with strong interconecctions using neural networks. *IEEE Transactions on Automatic Control*, 48(5):805–810, 2003.

Z.-P. Jiang. New results in decentralized adaptive nonlinear control with output feedback. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 4772–4777, Phoenix, Arizona, USA, 1999.

Y. Jin. Decentralized adaptive fuzzy control of robot manipulators. *IEEE Transactions on Systems, Man, and Cybernetics,Part B*, 28(1):47–57, 1998.

A. Karakasoglu, S.I. Sudharsanan, and M.K. Sundareshan. Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators. *IEEE Transactions on Neural Networks*, 4(6):919–930, 1993.

R. Kelly and V. Santibañez. *Control de Movimiento de Robots Manipuladores.* Pearson Prentice Hall, Madrid, 2003.

R. Kelly, V. Santibañez, and A. Loría. *Control of Robots Manipulators in Joint Space.* Springer-Verlag, London, 2005.

M. Liu. Decentralized control of robot manipulators: nonlinear and adaptive approaches. *IEEE Transactions on Automatic Control*, 44(2):357–363, 1999.

A.G. Loukianov. Robust block decomposition sliding mode control design. *Mathematical Problems in Engineering*, 8(4-5):349–365, 2003.

M.-L. Ni and M.J. Er. Decentralized control of robot manipulators with coupling and uncertainties. In *Proceedings of the American Control Conference*, pages 3326–3330, Chicago, Illinois, USA, 2000.

R. Safaric and J. Rodic. Decentralized neural-network sliding-mode robot controller. In *Proceedings of 26th Annual Conference on the IEEE Industrial Electronics Society*, pages 906–911, Nagoya, Japan, 2000.

E.N. Sanchez and L.J. Ricalde. Trajectory tracking via adaptive recurrent neural control with input saturation. In *Proceedings of International Joint Conference on Neural Networks 2003*, pages 359–364, Portland, Oregon, USA, 2003.

V. Santibañez, R. Kelly, and M.A. Llama. A novel global asymptotic stable set-point fuzzy controller with bounded torques for robot manipulators. *IEEE Transactions on Fuzzy Systems*, pages 362–372, June 2005.

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*, 5(1):59–78, 1995. Birkhauser-Boston.

M. Spong and M. Vidyasagar. *Robot Dynamics and Control.* Wiley, New York, 1989.

V. Utkin. Block control principle for mechanical systems. *Journal of Dynamic Systems, Measurement, and Control*, 122(1):1–10, 2000.

V. Utkin, J. Guldner, and J. Shi. *Sliding Mode Control in Electromechanical Systems.* Francis and Taylor, Philadelphia, 1999.