The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Real-Time Decentralized Neural Block Controller for a Robot Manipulator

R. Garcia-Hernandez[1], E. N. Sanchez[2], V. Santibañez[3], M. A. Llama[3] and E. Bayro-Corrochano[2]

Abstract— **This paper presents a discrete-time decentralized control scheme for identification and trajectory tracking of a two degrees of freedom (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 online 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 proposed scheme is implemented in real-time to control a two DOF robot manipulator.**

I. INTRODUCTION

Robot manipulators are employed in a wide assortment of applications in industry. Today most of the applications are in manufacturing to move materials, parts, and tools of various types. Future applications will include nonmanufacturing tasks, such as construction work, exploration of space, and medical care.

In this context, multiple control schemes have been proposed to guarantee efficient trajectory tracking and stability [1], [2]. The fast advance in computational technology offers many ways for implementing control algorithms within the approach of a centralized control design [3]. 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 dynamics difficult to model. Considering only the most important terms, the mathematical model obtained requires control algorithms with great number of mathematical operations, which affect 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 considering a global system as 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 [4], [5]. Decentralized control has been applied in robotics, mainly in cooperative multiple mobile robots and robot manipulators, where it is natural to consider each mobile robot or each manipulator as a subsystem of the whole system. For robot manipulators each joint is considered as a subsystem 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 [6].

In [7], 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 [8], an approach of decentralized neural identification and control for robot manipulators is presented using models in discrete-time. In [9], a decentralized control for robot manipulators is reported; it is based on the estimation of independent dynamics for each of the joints, using feedforward neural networks.

In [10], the authors proposed a similar decentralized control strategy using a recurrent neural identifier and block control structure for both identification and control. This approach was tested only using simulations, with a two degrees of freedom robot manipulator, and with an ANAT (Articulated Nimble Adaptable Trunk) manipulator, with seven degrees of freedom.

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 robot manipulator 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 [11]. We present the real-time implementation of the proposed scheme to control a two DOF robot manipulator.

II. DISCRETE-TIME DECENTRALIZED SYSTEMS

Let consider a class of discrete-time nonlinear perturbed and interconnected system which can be presented in the nonlinear block-controllable (NBC) form [12] consisting of r blocks

$$
\begin{aligned}
\chi_i^1(k+1) &= f_i^1(\chi_i^1) + B_i^1(\chi_i^1)\chi_i^2 + \Gamma_{i\ell}^1 \\
\chi_i^2(k+1) &= f_i^2(\chi_i^1,\chi_i^2) + B_i^2(\chi_i^1,\chi_i^2)\chi_i^3 + \Gamma_{i\ell}^2 \\
&\;\;\vdots \\
\chi_i^{r-1}(k+1) &= f_i^{r-1}(\chi_i^1,\chi_i^2,\ldots,\chi_i^{r-1}) \\
&\quad + B_i^{r-1}(\chi_i^1,\chi_i^2,\ldots,\chi_i^{r-1})\chi_i^r + \Gamma_{i\ell}^{r-1} \\
\chi_i^r(k+1) &= f_i^r(\chi_i) + B_i^r(\chi_i)u_i + \Gamma_{i\ell}^r
\end{aligned}
\tag{1}
$$

This work was supported by CONACYT, Mexico under Grant 57801.
[1] Universidad Autonoma del Carmen, Facultad de Ingenieria, Cd. del Carmen, Campeche, Mexico. PhD. Student at CINVESTAV Guadalajara. e-mail:rghernandez@pampano.unacar.mx
[2] CINVESTAV Guadalajara, Apartado Postal 31-438, Plaza La Luna, C.P. 44550, Guadalajara, Jalisco, Mexico. e-mail:[sanchez,edb]@gdl.cinvestav.mx
[3]Instituto Tecnologico de la Laguna, Apartado Postal 49, Adm. 1, C.P. 27001, Torreon, Coahuila, Mexico. e-mail:vsantiba@itlalaguna.edu.mx

where $\chi_i \in \Re^{n_i}$, $\chi_i = \left[\chi_i^{1T}\ \chi_i^{2T}\ \dots\ \chi_i^{rT}\right]^T$ and $\chi_i^j \in \Re^{n_{ij}\times 1}$, $\chi_i^j = \left[\chi_{i1}^j\ \chi_{i2}^j\ \dots\ \chi_{il}^j\right]^T$, $i = 1,\dots,N$; $j = 1,\dots,r$; $l = 1,\dots,n_{ij}$; N is the number of subsystems, $u_i \in \Re^{m_i}$ is the input vector, the rank of $B_i^j = n_{ij}$, $\sum_{j=1}^r n_{ij} = n_i$, $\forall \chi_i^j \in D_{\chi_i^j} \subset \Re^{n_{ij}}$. We assume that f_i^j, B_i^j and Γ_i^j are smooth and bounded functions, $f_i^j(0) = 0$ and $B_i^j(0) = 0$. The integers $n_{i1} \le n_{i2} \le \dots \le n_{ij} \le m_i$ define the different subsystem structures. The unmatched and matched interconnection terms are given by

$$\begin{aligned}
\Gamma_{i\ell}^1 &= \sum_{\ell=1,\ \ell\neq i}^N \gamma_{i\ell}^1(\chi_\ell^1)\\
\Gamma_{i\ell}^2 &= \sum_{\ell=1,\ \ell\neq i}^N \gamma_{i\ell}^2(\chi_\ell^1,\chi_\ell^2)\\
&\ \ \vdots\\
\Gamma_{i\ell}^{r-1} &= \sum_{\ell=1,\ \ell\neq i}^N \gamma_{i\ell}^{r-1}(\chi_\ell^1,\chi_\ell^2,\dots,\chi_\ell^{r-1})\\
\Gamma_{i\ell}^r &= \sum_{\ell=1,\ \ell\neq i}^N \gamma_{i\ell}^r(\chi_\ell)
\end{aligned} \tag{2}$$

where χ_ℓ represents the state vector of the ℓ-th subsystem with $1 \le \ell \le N$ and $\ell \neq i$.

The terms (2) reflect the interaction between the i-th and the others subsystems.

III. Neural Identifier

A. Decentralized Neural Identifier

The following decentralized recurrent high order neural network (RHONN) model is proposed to identify (1):

$$\begin{aligned}
x_i^1(k+1) &= w_i^1(k)S(\chi_i^1(k)) + w_i^{'1}\chi_i^2(k)\\
x_i^2(k+1) &= w_i^2(k)S(\chi_i^1(k),\chi_i^2(k)) + w_i^{'2}(k)\chi_i^3(k)\\
&\ \ \vdots\\
x_i^{r-1}(k+1) &= w_i^{r-1}(k)S(\chi_i^1(k),\chi_i^2(k),\dots,\chi_i^{r-1}(k))\\
&\quad + w_i^{'(r-1)}\chi_i^r(k)\\
x_i^r(k+1) &= w_i^r(k)S(\chi_i^1(k),\dots,\chi_i^r(k)) + w_i^{'r}u_i(k)
\end{aligned} \tag{3}$$

where $x_i^j = \left[x_i^1\ x_i^2\ \dots\ x_i^r\right]^T$ is the i-th block neuron state $i = 1,\dots,N$ and $j = 1\dots,r$, $w_i^j(k)$ are the adjustable weights, $w_i^{'j}$ are the fixed weights with $rank(w_i^{'j}) = n_{ij}$, $S(\cdot)$ is the activation function, and $u_i(k)$ represents the input vector.

It is worth to note that (3), constitutes a series-parallel identifier and fulfills the conditions of the nonlinear block-controllable form [13].

B. EKF Training Algorithm

It is known that Kalman filtering (KF) estimates the state of a linear system with additive state and output white noises [14], [15]. For KF-based neural network (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 $w_i^j(k)$ which minimize the prediction error. We use an EKF-based training algorithm described by:

$$\begin{aligned}
K_i^j(k) &= P_i^j(k)H_i^j(k)M_i^j(k)\\
w_i^j(k+1) &= w_i^j(k) + \eta_i^j K_i^j(k)e_i^j(k)\\
P_i^j(k+1) &= P_i^j(k) - K_i^j(k)H_i^{jT}(k)P_i^j(k) + Q_i^j(k)
\end{aligned} \tag{4}$$

with

$$\begin{aligned}
M_i^j(k) &= [R_i^j(k) + H_i^{jT}(k)P_i^j(k)H_i^j(k)]^{-1}\\
e_i^j(k) &= [\chi_i^j(k) - x_i^j(k)]
\end{aligned} \tag{5}$$

where $e_i^j(k)$ is the identification error, $P_i^j(k)$ is the prediction error covariance matrix, $w_i^j(k)$ is the j-th weight (state) of the i-th subsystem, η_i^j is the rate learning parameter such that $0 \le \eta_i^j \le 1$, $\chi_i^j(k)$ is the j-th plant state, $x_i^j(k)$ is the j-th neural network state, n is the number of states, $K_i^j(k)$ is the Kalman gain matrix, $Q_i^j(k)$ is the measurement noise covariance matrix, $R_i^j(k)$ is the state noise covariance matrix, and $H_i^j(k)$ is a matrix, in which each entry of (H_i^j) is the derivative of j-th neural network state ($x_i^j(k)$), with respect to all adjustable weights (w_i^j), as follows

$$H_i^j(k) = \left[\frac{\partial x_i^j(k)}{\partial w_i^j(k)}\right]^T_{w_i^j(k)=w_i^j(k+1)}, \tag{6}$$

where $i = 1,\dots,N$ and $j = 1,\dots,n$. Usually P_i^j and Q_i^j are initialized as diagonal matrices, with entries $P_i^j(0)$ and $Q_i^j(0)$, respectively [15]. It is important to remark that $H_i^j(k)$, $K_i^j(k)$, and $P_i^j(k)$ for the EKF are bounded [16].

IV. Controller Design

A. Neural Block Controller

The proposed block control is given in the following equations, we begin defining the tracking error as

$$z_i^1(k) = x_i^1(k) - x_{i\mathrm{d}}^1(k) \tag{7}$$

where $x_{i\mathrm{d}}^1(k)$ is the desired trajectory signal.

Once defined the first new variable (7), one step ahead is taken

$$z_i^1(k+1) = w_i^1(k)S(\chi_i^1(k)) + w_i^{'1}\,\chi_i^2(k) - x_{i\mathrm{d}}^1(k+1). \tag{8}$$

Equation (8) is viewed as a block with state $z_i^1(k)$ and the state $\chi_i^2(k)$ is considered as a pseudo-control input, where desired dynamics can be imposed. This can be solved with the anticipation of the desired dynamics for this block as follows:

$$\begin{aligned}
z_i^1(k+1) &= w_i^1(k)S(\chi_i^1(k)) + w_i^{'1}\,\chi_i^2(k) - x_{i\mathrm{d}}^1(k+1)\\
&= k_i^1 z_i^1(k)
\end{aligned} \tag{9}$$

where $\left|k_i^1\right| < 1$, in order to assure stability of (9). $\chi_i^2(k)$ is computed as

$$x_{i\mathrm{d}}^2(k) = \frac{1}{w_i^{'1}}[-w_i^1(k)S(\chi_i^1(k)) + \chi_{i\mathrm{d}}^1(k+1) + k_i^1 z_i^1(k)]. \tag{10}$$

Note that the calculated value of state $x_{i\mathrm{d}}^2(k)$ in (10) is not the real value of such state, instead, represents the desired behavior for $\chi_i^2(k)$. So, to avoid confusions this desired value of $\chi_i^2(k)$ is referred as $x_{i\mathrm{d}}^2(k)$ in (10).

Proceeding in the same way as for the first block, a second variable in the new coordinates is defined as

$$z_i^2(k) = x_i^2(k) - x_{i\mathrm{d}}^2(k).$$

Taking one step ahead in $z_i^2(k)$ yields

$$z_i^2(k+1) = x_i^2(k+1) - x_{i\mathrm{d}}^2(k+1).$$

The desired dynamics for this block are imposed as

$$\begin{aligned} z_i^2(k+1) &= w_i^2(k)S(\chi_i^1(k), \chi_i^2(k)) + w_i^{'2}\, x_i^3(k) \\ &\quad - x_{i\mathrm{d}}^2(k+1) \\ &= k_i^2 z_i^2(k) \end{aligned} \tag{11}$$

where $\left|k_i^2\right| < 1$.

These steps are taken iteratively. At the last step, the known desired variable is $x_{i\mathrm{d}}^r(k)$, and the last new variable is defined as

$$z_i^r(k) = x_i^r(k) - x_{i\mathrm{d}}^r(k).$$

As usually, taking one step ahead yields

$$\begin{aligned} z_i^r(k+1) &= w_i^r(k)S(\chi_i^1(k), \ldots, \chi_i^r(k)) \\ &\quad + w_i^{'r} u_i(k) - x_{i\mathrm{d}}^r(k+1). \end{aligned} \tag{12}$$

The system (3) can be represented in the new variables $z_i = \left[z_i^{1T} z_i^{2T} \cdots z_i^{rT}\right]$ of the form

$$\begin{aligned} z_i^1(k+1) &= k_i^1 z_i^1(k) + w_i^{'1} z_i^2(k) \\ z_i^2(k+1) &= k_i^2 z_i^2(k) + w_i^{'2} z_i^3(k) \\ &\vdots \\ z_i^{r-1}(k+1) &= k_i^{r-1}\, z_i^{r-1}(k) + w_i^{'(r-1)} z_i^r(k) \\ z_i^r(k+1) &= w_i^r(k)S(\chi_i^1(k), \ldots, \chi_i^r(k)) + w_i^{'r} u_i(k) \\ &\quad - x_{i\mathrm{d}}^r(k+1). \end{aligned} \tag{13}$$

For a sliding mode control implementation [17], when the control resources are bounded by u_{0i} as

$$|u_i(k)| \leq u_{0i} \tag{14}$$

a sliding manifold and a control law that will drive the states toward such manifold must be designed. The sliding manifold is chosen as $S_{D_i}(k) = z_i^r(k) = 0$. The system (12) is rewritten as follows:

$$\begin{aligned} S_{D_i}(k+1) &= w_i^r(k)S(\chi_i^1(k), \ldots, \chi_i^r(k)) + w_i^{'r} u_i(k) \\ &\quad - x_{i\mathrm{d}}^r(k+1). \end{aligned} \tag{15}$$

Once defined the sliding manifold, the next step is to find a control law that takes into consideration the bound (14), therefore, the control $u_i(k)$ is selected of the following form [18]:

$$u_i(k) = \begin{cases} u_{eq_i}(k) & \text{for } \|u_{eq_i}(k)\| \leq u_{0i} \\ u_{i0} \frac{u_{eq_i}(k)}{\|u_{eq_i}(k)\|} & \text{for } \|u_{eq_i}(k)\| > u_{0i} \end{cases} \tag{16}$$

where the equivalent control $u_{eq_i}(k)$ is calculated from $S_{D_i}(k+1) = 0$ as

$$u_{eq_i}(k) = \frac{1}{w_i^{'r}}[-w_i^r(k)S(\chi_i^1(k), \ldots, \chi_i^r(k)) + x_{i\mathrm{d}}^r(k+1)]. \tag{17}$$

The whole proposed identification and control scheme for the system is displayed in Fig. 1.

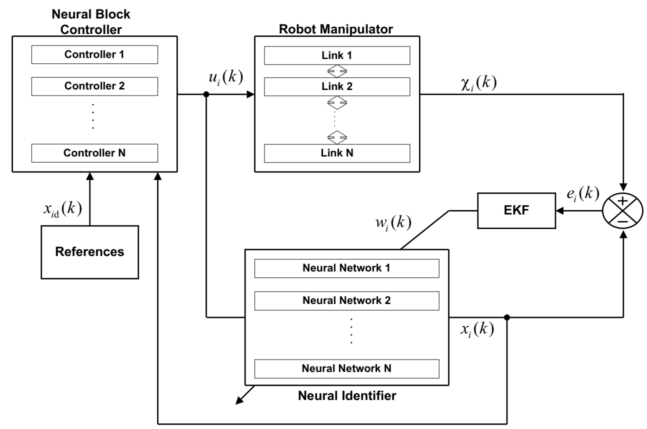

Fig. 1. Neural block control scheme

V. Two DOF Robot Manipulator Application

A. Robot Description

In order to evaluate via real-time implementation the performance of the proposed control algorithm, we use a two DOF robot manipulator moving in the vertical plane as shown in Fig. 2. 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 present 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.

B. Control Objective

To identify the robot model, from (1) and (3) we propose the following decentralized series-parallel neural network

$$\begin{aligned} x_i^1(k+1) &= w_{i1}^1(k)S(\chi_i^1(k)) + w_i^{'1}\chi_i^2(k) \\ x_i^2(k+1) &= w_{i1}^2(k)S(\chi_i^1(k)) + w_{i2}^2(k)S(\chi_i^2(k)) \\ &\quad + w_i^{'2} u_i(k) \end{aligned} \tag{18}$$

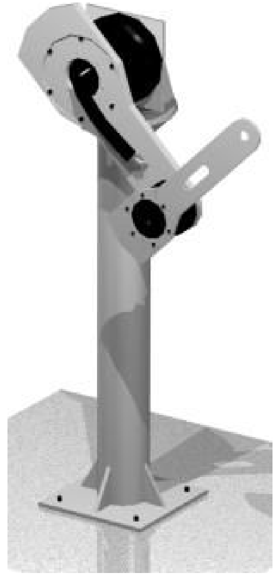

Fig. 2. Robot manipulator

where $x_i^1(k)$ and $x_i^2(k)$ identify $\chi_i^1(k)$ and $\chi_i^2(k)$; $i = 1, 2$, respectively; w_{ip}^j are the adjustable weights, p is the number of adjustable weights with $p = 1$; $j = 1$ for the first NN state and $p = 1, 2$; $j = 1, 2$ for the second one; $w_i^{'1}$ and $w_i^{'2}$ are fixed parameters.

Due to the time varying of RHONN weights, we need to guarantee the controllability of the system by assuring the weights $w_i^{'1}$ and $w_i^{'2}$ are not zero, which are so-called controllability weights [19].

To update the weights w_{ip}^j, an EKF-based training algorithm (4) is implemented.

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

$$z_i^1(k) = x_i^1(k) - x_{i\mathrm{d}}^1(k) \tag{19}$$

where $x_{i\mathrm{d}}^1$ is the desired trajectory signal. Using (18) and introducing the desired dynamics for $z_i^1(k)$ we have

$$\begin{aligned} z_i^1(k+1) &= w_{i1}^1(k)S(\chi_i^1(k)) + w_i^{'1}\chi_i^2(k) \\ &\quad - x_{i\mathrm{d}}^1(k+1) \\ &= k_i^1 z_i^1(k). \end{aligned} \tag{20}$$

The desired value $x_{i\mathrm{d}}^2(k)$ for $\chi_i^2(k)$ is calculated from (20) as

$$\begin{aligned} x_{i\mathrm{d}}^2(k) &= \frac{1}{w_i^{'1}}[-w_{i1}^1(k)S(\chi_i^1(k)) \\ &\quad + x_{i\mathrm{d}}^1(k+1) + k_i^1 z_i^1(k)]. \end{aligned} \tag{21}$$

At the next step, let define a new variable as

$$z_i^2(k) = x_i^2(k) - x_{i\mathrm{d}}^2(k). \tag{22}$$

Taking one step ahead, we have

$$\begin{aligned} z_i^2(k+1) &= w_{i1}^2(k)S(\chi_i^1(k)) + w_{i2}^2(k)S(\chi_i^2(k)) \\ &\quad + w_i^{'2}u_i(k) - x_{i\mathrm{d}}^2(k+1) \\ &= k_i^2 z_i^2(k). \end{aligned} \tag{23}$$

The manifold for the sliding mode is chosen as $S_{D_i}(k) = z_i^2(k) = 0$. The control law is given by

$$u_i(k) = \begin{cases} u_{eq_i}(k) & \text{for} \quad \|u_{eq_i}(k)\| \le \tau_i^{\max} \\ \tau_i^{\max}\frac{u_{eq_i}(k)}{\|u_{eq_i}(k)\|} & \text{for} \quad \|u_{eq_i}(k)\| > \tau_i^{\max} \end{cases} \tag{24}$$

where $u_{eq_i}(k)$ is obtained from $S_{D_i}(k+1) = 0$ as

$$\begin{aligned} u_{eq_i}(k) &= \frac{1}{w_i^{'2}}[-(w_{i1}^2(k)S(\chi_i^1(k)) \\ &\quad + w_{i2}^2(k)S(\chi_i^2(k))) + x_{i\mathrm{d}}^2(k+1)] \end{aligned} \tag{25}$$

and the control resources are bounded by $\tau_i^{\max}$.

C. Real-Time Results

For the experiments we choose the following discrete-time trajectories as

$$x_{1\mathrm{d}}^1(k) = b_1(1 - e^{d_1 kT^3}) + c_1(1 - e^{d_1 kT^3})\sin(\omega_1 kT)[\mathrm{rad}]$$
$$x_{2\mathrm{d}}^1(k) = b_2(1 - e^{d_2 kT^3}) + c_2(1 - e^{d_2 kT^3})\sin(\omega_2 kT)[\mathrm{rad}]$$

where $b_1 = \pi/4$, $c_1 = \pi/18$, $d_1 = -2.0$, and $\omega_1 = 5$ [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 $\omega_2 = 1.0$ [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 grows for maintaining the robot in an operation state without saturating actuators whose limit are in 150 [Nm] and 15 [Nm], respectively. The figures 3 and 4 display the identification and trajectory tracking results for each joint. In the real system the initial conditions are the same that those of the neural identifier both are restricted to be equal to zero, therefore does not exist transient errors.

According to the figures 3 and 4, the identification error for joint 1 and 2 presents a good behavior and remain bounded as shown in Fig. 5. The tracking errors for each joint are presented in Fig. 6. The applied torques to each joint are shown in Fig. 7. The control signals present oscillations at some time instants due to the gains and fixed parameters chosen for each controller. 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 bounds $\tau_1^{\max}$ and $\tau_2^{\max}$, respectively.

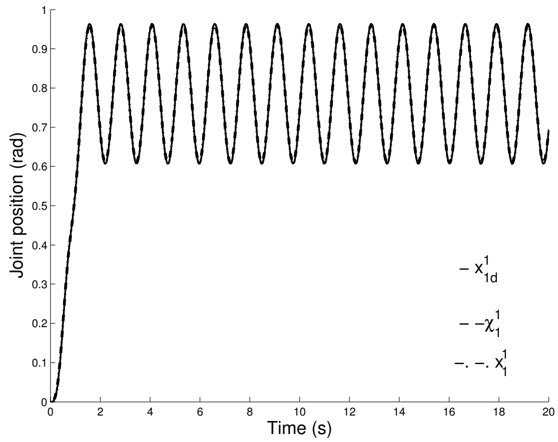

Fig. 3. Identification and tracking for joint 1 $x^1_{1\mathrm{d}}(k)$ (solid line), $\chi^1_1(k)$ (dashed line), and $x^1_1(k)$ (dashdot line)

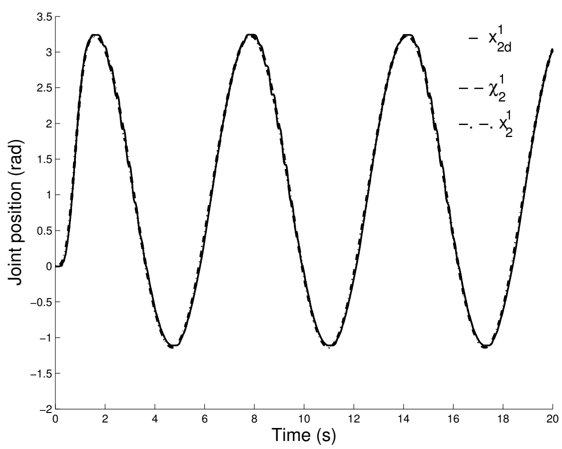

Fig. 4. Identification and tracking for joint 2 $x^1_{2\mathrm{d}}(k)$ (solid line), $\chi^1_2(k)$ (dashed line), and $x^1_2(k)$ (dashdot line)

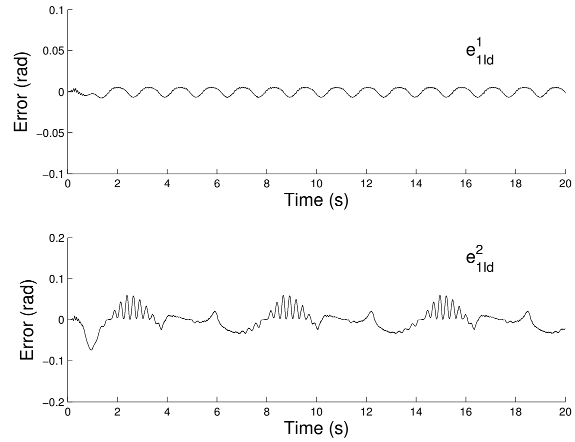

Fig. 5. Identification errors for joints 1 and 2

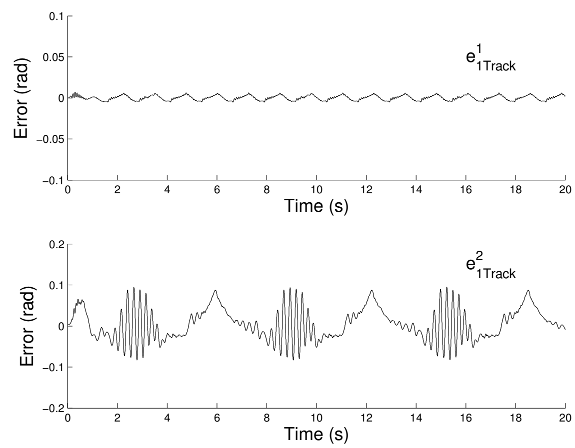

Fig. 6. Tracking errors for joints 1 and 2

VI. Conclusions

A decentralized neural identification and control scheme is proposed which is able to identify each subsystem dynamics. The training of each neural network is performed online using an extended Kalman filter in a series-parallel configuration. The obtained real-time results present good performance for trajectory tracking when applied to a two DOF robot manipulator. Currently we are carrying out experiments using a real five DOF ANAT robot [20]. The results will be published in near future.

Acknowledgment

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

References

[1] 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*, Portland, Oregon, USA, 2003, pp. 359–364.

[2] 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*, vol. 13, no. 3, pp. 362–372, Jun. 2005.

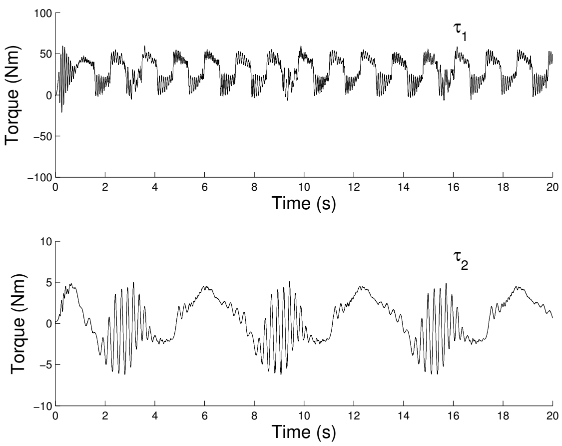

Fig. 7. Applied torques to joint 1 and 2

[3] R. Gourdeau, "Object-oriented programming for robotic manipulator simulation," *IEEE Robotics and Automation*, vol. 4, no. 3, pp. 21–29, 1997.

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

[5] S. Huang, K. K. Tan, and T. H. Lee, "Decentralized control design for large-scale systems with strong interconnections using neural networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 805–810, 2003.

[6] M. Liu, "Decentralized control of robot manipulators: nonlinear and adaptive approaches," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 357–363, 1999.

[7] M. L. Ni and M. J. Er, "Decentralized control of robot manipulators with coupling and uncertainties," in *Proceedings of the American Control Conference*, Chicago, Illinois, USA, 2000, pp. 3326–3330.

[8] 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*, vol. 4, no. 6, pp. 919–930, 1993.

[9] R. Safaric and J. Rodic, "Decentralized neural-network sliding-mode robot controller," in *Proceedings of 26th Annual Conference on the IEEE Industrial Electronics Society*, Nagoya, Aichi, Japan, 2000, pp. 906–911.

[10] E. N. Sanchez, A. Gaytan, and M. Saad, "Decentralized neural identification and control for robotics manipulators," in *Proceedings of the IEEE International Symposium on Intelligent Control*, Munich, Germany, Oct 2006, pp. 1614–1619.

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

[12] V. Utkin, "Block control principle for mechanical systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, no. 1, pp. 1–10, 2000.

[13] V. H. Benitez, E. N. Sanchez, and A. G. Loukianov, "Decentralized adaptive recurrent neural control structure," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1125–1132, 2007.

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

[15] S. Haykin, *Kalman Filtering and Neural Networks*. New York, USA: John Wiley & Sons, Inc, 2001.

[16] Y. Song and J. W. Grizzle, "The extended Kalman filter as local asymptotic observer for discrete-time nonlinear systems," *Journal of Mathematical Systems, Estimation and Control*, vol. 5, no. 1, pp. 59–78, 1995.

[17] V. Utkin, "Sliding mode control design principles and applications to electric drives," *IEEE Transactions on Industrial Electronics*, vol. 40, no. 1, pp. 23–36, Feb. 1993.

[18] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electromechanical Systems*. Philadelphia, USA: Francis & Taylor, 1999.

[19] C. E. Castañeda, E. N. Sanchez, A. G. Loukianov, and B. Castillo-Toledo, "Discrete-time recurrent neural DC motor control using Kalman learning," in *Proceedings of International Joint Conference on Neural Networks*, Hong Kong, China, Jun 2008, pp. 1930–1937.

[20] R. Garcia-Hernandez, E. N. Sanchez, M. Saad, and E. Bayro-Corrochano, "Discrete-time decentralized neural block controller for a five dof robot manipulator," in *Proceedings of International Joint Conference on Neural Networks*, Atlanta, Georgia, USA, Jun 2009, pp. 925–931.