TRACKING CONTROL OF ROBOT MANIPULATORS USING SECOND ORDER NEURO SLIDING MODE

R. GARCIA-RODRIGUEZ† and V. PARRA-VEGA‡

†*Department of Electrical Engineering, Universidad de Chile, Av. Tupper 2007, Santiago, Chile*
rgarciar@ing.uchile.cl
‡*Robotics and Advanced, Manufacturing Division, CINVESTAV, México*
vparra@cinvestav.mx

Abstract— **Few works on neural networks-based robot controllers address the issue of how many units of neurons, hidden layers and inputs are necessary to approximate any functions up to a bounded approximation error. Thus, most proposals are conservative in a sense that they depend on high dimensional hidden layer to guarantee a given bounded tracking error, at a computationally expensive cost, besides that an independent input is required to stabilize the system.**

In this paper, a low dimensional neural network with online adaptation of the weights is proposed with an stabilizer input which depends on the same variable that tunes the neural network. The size of the neural network is defined by degree of freedom of the robot, without hidden layer. The neuro-control strategy is driven by a second order sliding surface which produce a chattering-free control output to guarantee tracking error convergence. To speed the response up even more, a time base generator shapes a feedback gain to induce finite time convergence of tracking errors for any initial condition. Experimental results validate our proposed neuro-control scheme.

Keywords— **Robot control, Neural networks, Second order sliding mode, Chattering-free, Neuro-sliding controller.**

I. INTRODUCTION

In robotics, one of main objectives is to design simple controllers to compensate nonlinear couplings, parameters variation and disturbances to execute complex tasks with greater precision in tracking regime.

Although previously in 1980 the computed torque controller was presented;[1] it was not until 1986 when Slotine and Li (1987) showed that a particular structure of manipulator dynamics exist to develop a simple controller avoiding measurements or estimates of the manipulator joint accelerations. Through this adaptive control scheme it is possible to compensate parameter variations to guarantee local stability of the system as well as asymptotic convergence of the tracking errors without any knowledge of the parameters, though the exact regressor is a requirement. Based on this result, many schemes in adaptive control has been developed and applied to a wide class of systems. Although the adaptive control represents a problem solution of parameter variation in the robots, the principal drawback is that it is a model-based controller, therefore the computation effort increases proportionally to degree of freedom of the robot or when the robotic system is more complex, *e.g.* in cooperative robots or mechanical hands.

At the same time, in the 80's the simple PD controller was presented, where it can compensate nonlinearity and uncertainties of robot dynamics (Arimoto, 1996). In addition it is recognized that one of the generic characteristic of robot dynamics is its open loop passivity property from torque input to velocity output as a way to exploit the robot system's physical structure and design energetically stable controllers. In this case, for stability purposes, a storage energy function arises which gives rise to stable behavior, then the challenge is to produce passivity in closed-loop via a given error velocity as its output.

In another way, as a result of the work done by many researchers started by McCulloch and Pitts, the neural networks attracted attention as networks ability to mimic basic patterns of the human brain, such as its ability to learn and respond in consequence, as if it were employed the learning capability to produce a control action.

In terms of control design, the main interest in neural networks is their capability to approximate a large class of continuous nonlinear maps from the collective action of autonomous processing units interconnected in simple ways, as well as inherent parallel and highly redundant processing architecture, that makes it possible to develop parallel adaptation update laws and reduce latency. These neural network

[1] The exact knowledge of the parameters of robot and a great computational power is required and it can not compensate parameter variations.

properties have been used in a large number of applications as adaptive system identification (Narendra and Parthasarathy, 1990) and control of complex highly uncertain dynamical systems (Lewis *et al.*, 1996; Kosmatopoulos and Christodoulou, 1994; Ge and Hang, 1998; Lee and Choi, 2004).

The adaptive neural network-based controllers have been getting the attention because they make possible to design controllers of a wide class of systems without any knowledge of the dynamic model, the regressor nor the parameters. Basically, neural network-based controllers approximate the inverse dynamics of system in a given error coordinate system. However, well known results show that a large number of nodes is required in each layer of the neural network[2] to achieve exact approximation of the unknown functional (Cotter, 1990). The amount of nodes can be prohibitory large for a simple practical systems[3].

In order to design neuro-control schemes with smooth control and low computational effort to compensate the unknown physical parameters of robot, a dynamical combination of different intelligent and control techniques as variable structure systems, passivity, model-based control and PID-like controller has been proposed (Ertugrul and Kaynak, 2000; Sanchez *et al.*, 2003; Yu, 2003; Choi *et al.*, 2001; Lin *et al.*, 2000; Lin *et al.*, 2001; Barambones and Etxebarria, 2002; Debache *et al.*, 2006; Hayakawa *et al.*,2005).

In neuro-adaptive or neuro-sliding mode control (Ge and Harris, 1994; Ertugrul and Kaynak, 2000; Lewis *et al.*, 1996; Ge and Hang, 1998) generally neural network approximate the inverse dynamics of the manipulator based on gradient descent method or adaptive control, where the main disadvantage of these schemes is that they use a great number of neurons in each layer of neural network. Sometimes it is necessary an additional and independent control term to guarantee stability and robustness in the presence of approximation error (Yamakita and Satoh, 1999; Ge and Harris, 1994; Yu, 2003; Sanchez *et al.*, 2003, Lin *et al.*, 2000; Lewis *et al.*, 1996; Sun and Sun, 1999). However, its high frequency input represent the principal disadvantage in practical applications, though to eliminate the chattering a saturation function is included (Barambones and Etxebarria, 2002; Lin *et al.*, 2001, Ertugrul and Kaynak, 2000; Chih-Min and Chun-Fei, 2002). Unfortunately, in the later case the invariant condition is not satisfied and tracking is not guaranteed.

[2]The network topology refers to the number and organization of the computing units, the types of connections between neurons, and the direction of information flow in the network. The node is the basic organizational unit of a neural network, and nodes are arranged in a series of layers to create the Artificial Neural Network (ANN). According to their location and function within the network, nodes are classified as input, output, or hidden layer nodes (Stern, 1991)

[3]In feed-forward neural networks with multilayer structure, an oversized hidden layer does not increase the accuracy of approximation, but increases the risk of overfitting, *i.e.* it may lead to a bad approximation

In other approaches based on passivity-based adaptive control, the convergence of tracking errors is guaranteed under assumption of a combination of neural network with regressor knowledge, which reduces the main advantage of the neural network (Jung and Hsia, 1996). Therefore, apparently there is not room for the main role of the neural networks as universal approximator of any continuous function if a high frequency input is included or if some part of the regressor is used. Some researchers devised the way to introduce a neural network to substitute the regressor into a classical adaptive control (Yu, 2003; Sanchez *et al.*, 2003; Lewis *et al.*, 1996; Ge and Harris, 1994), however they only guarantee stability without convergence of tracking errors.

In this paper is presented a combination of second order sliding mode control with low dimensional neural network based on adaptive linear elements (Adalines) to guarantee tracking errors convergence with a smooth controller where the regressor of robot it is not required. The closed loop system renders a sliding mode for all time, whose solution converges in finite time and hence a perfect tracking is obtained. In addition, it is presented an alternative solution to address the issue of how many units of neurons are necessary to approximate any continuous function as well as the input set. Experimental results on a robot manipulators are presented that verify the closed loop stability properties.

The paper is organized as follows. Section II shows the robot dynamics and its properties while Section III presents the open loop error dynamic system. In Section IV presents the main properties and characteristics of the neural network used in this paper. The proposed control scheme and its stability analysis is given in Section V . Section VI presents the experimental results on planar robot manipulator and conclusions are given in Section VII.

II. ROBOT DYNAMICS

The dynamic model of a rigid serial n-link robot manipulator with all revolute joints is described as follows

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \qquad (1)$$

where $q, \dot{q}, \ddot{q} \in \Re^n$ are the generalized joint coordinates, $H(q) \in \Re^{nxn}$ denotes a symmetric positive definite inertial matrix, the second term on the left represents the Coriolis and centripetal forces $C(q,\dot{q}) \in \Re^{nxn}$, $g(q) \in \Re^n$ models the gravity forces, and $\tau \in \Re^n$ stands for the torque input.

Some useful properties of robot dynamic are:

Property 1: (Arimoto, 1996) Matrix $H(q)$ is symmetric and positive definite.

Property 2: Matrix $\frac{1}{2}\dot{M}(q) - C(q,\dot{q})$ is skew symmetric and hence satisface (Arimoto, 1996):

$$\dot{q}^T[\frac{1}{2}\dot{M}(q) - C(q,\dot{q}]\dot{q} = 0 \qquad \forall q, \dot{q} \in \Re^n \qquad (2)$$

Property 3: There exists positive scalar $\beta_i (i = 0, \ldots, 5)$ such that

$$\begin{aligned} \| H(q) \| &\geq \lambda_m(H(q)) > \beta_0 > 0 \\ \| H(q) \| &\leq \lambda_M(H(q)) < \beta_1 > \infty \\ \| C(q,\dot{q}) \| &\leq \beta_2 \| \dot{q} \| \\ \| g(q) \| &\leq \beta_3 \\ \| \dot{q}_r \| &\leq \beta_4 + \alpha \| \Delta q \| + \gamma \| \sigma \| \\ \| \ddot{q}_r \| &\leq \beta_5 + \alpha \| \Delta \dot{q} \| \end{aligned} \tag{3}$$

with $\lambda_m(A)$, $\lambda_M(A)$ stand the minimum and maximum eigenvalues of matrix $A \in \Re^{nxn}$, respectively. The norm of a vector x is defined as $\|x\| = \sqrt{x^T x}$ and $\|A\| = \sqrt{\lambda_M(A^T A)}$ as induced Frobenius norm.

Property 4: The left-hand side of (1) is linear in terms of suitable selected set of robot and load parameters, i.e.

$$Y\Theta = H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) \tag{4}$$

where $Y = Y(q, \dot{q}, \dot{q}, \ddot{q}) \in \Re^{nxp}$ and $\Theta \in \Re^p$ containing the unknown robot manipulator and load parameters.

Property 5: The robot dynamics is passive in open-loop, from torque input to velocity output, with the Hamiltonian as its storage function. If viscous friction were considered, the energy dissipates and the system is strictly passive.

Due to linear parametrization property, (1) can be written in terms of a nominal reference $\dot{q}_r$ and its derivative $\ddot{q}_r$ as (Lewis and Abdallaah, 1994):

$$H(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + g(q) = Y_r\Theta \tag{5}$$

where the regressor is defined as $Y_r = Y_r(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \in \Re^{nxp}$. Using Property 3 in (5) we have that

$$\begin{aligned} Y_r\Theta &= \|H(q)\|\|\ddot{q}_r\| + \|C(q,\dot{q})\|\|\dot{q}_r\| + \|g(q)\| \\ &\leq \beta_1\alpha\|\Delta\dot{q}\| \\ &\quad + \beta_2\|\dot{q}\| * (\alpha\|\Delta q\| + \gamma\|\sigma\| + \beta_4) + \bar{\beta}_3 \\ &\leq \eta(t) \end{aligned} \tag{6}$$

where $\bar{\beta}_3 = \beta_1\beta_5 + \beta_3$ and $\eta(t) = f(\Delta q, \Delta\dot{q}, \sigma, \beta_i, t)$ is a state-dependent function. If we add and subtract (5) into (1) we obtain the open loop error equation

$$H(q)\dot{S}_r + C(q,\dot{q})S_r = \tau - Y_r\Theta \tag{7}$$

where the extend error S_r carries out a change of coordinates through $(\dot{q}_r, \ddot{q}_r)$, defined by

$$S_r = \dot{q} - \dot{q}_r \tag{8}$$

The question is how to design a smooth τ without knowledge of $Y_r\Theta$. To that end, it is useful to design a second order nominal reference $\dot{q}_r$.

III. ERROR MANIFOLDS AND ERROR DYNAMICS

Let the nominal reference $\dot{q}_r$ be

$$\dot{q}_r = \dot{q}_d - \alpha\Delta q + S_d - K_i \int_{t_0}^{t} sign(S_q(\zeta))d\zeta \tag{9}$$

where

$$\begin{aligned} S_q &= S - S_d & (10) \\ S &= \Delta\dot{q} + \alpha(t)\Delta q & (11) \\ S_d &= S(t_0)exp^{-kt} \qquad k > 0. & (12) \end{aligned}$$

with $\Delta q = q - q_d$ the position tracking error, with subscript d denoting the desired reference value, $\alpha > 0, K_i = K_i^T > 0$ and function sign(x) stands for the signum function of x. Substituting (9) into (8), we obtain the extended error S_r, which depends in turn of a second order sliding surface S_q defined as

$$S_r = S_q + K_i \int_{t_0}^{t} sign(S_q(\zeta))d\zeta \tag{13}$$

Remark 1: Sliding Surface with an integral term.
Based on seminal work (Slotine and Spong, 1985) several approaches have reported sliding surface with an integral term (Jager, 1996; Stepanenko *et al.*, 1998) to provide some robustness in the controller. However, in our paper, we use a integral with an entirely different purpose: the extend error uses the integral of the sliding surface to induce second order sliding mode *at* the sliding surface, without using integral term *in* the sliding surface S_q. That is, it is shown that the integral of $sign(S_q)$ satisfies the sliding condition for S_q. In this way, $sgn(*)$ is used without introducing chattering, avoiding to use boundary layer. Furthermore notice that $S_d \in C^1$ is given as desired reference of S in the phase plane $(\Delta\dot{q}, \Delta q)$ and it is designed to eliminate the reaching phase. Then S_d converges monotonously to zero with initial conditions $S_d(t_0) = S(t_0)$ at time $t = t_d > 0$ $(S_d(t_0) = 0)$.

Due to (5) involves the derivative of (9) we have that

$$\ddot{q}_r = \ddot{q}_d - \alpha\Delta\dot{q} + \dot{S}_d - K_i sign(S_q) \tag{14}$$

which is discontinuous. However since neural networks can not approximate discontinuous signals, we need to avoid introducing discontinuous signals in the function $Y_r\Theta$. To solve this, $\ddot{q}_r$ is decompose into continuous and discontinuous terms, as follows

$$\ddot{q}_r = \ddot{q}_{cont} + K_i Z \tag{15}$$

where

$$\begin{aligned} \ddot{q}_{cont} &= \ddot{q}_d - \alpha\Delta\dot{q} + \dot{S}_d - K_i \tanh(\lambda S_q) \\ Z &= \tanh(\lambda S_q) - sign(S_q), \end{aligned} \tag{16}$$

for the vector $\tanh(x) = [\tanh(x_1), \ldots, \tanh(x_k)]^T$ as the continuous hyperbolic tangent function of $X \in \Re^k$, $\lambda = \lambda^T \in \Re^{nxn} > 0$ and the equation (16) is bounded and it has the following properties: $Z \geq -1$, $Z \leq 1$, $Z_{S_q \to 0^-} = -1$, $Z_{S_q \to 0^+} = +1$ and $Z_{S_q \to \pm\infty} = 0$.

Substituting (15), (9) in (5) the parametrization $Y_r\Theta$ now is defined as

$$H(q)\ddot{q}_{cont} + C(q,\dot{q})\dot{q}_r + g(q) = Y_{cont}\Theta - \tau_d \qquad (17)$$

where the regressor $Y_{cont} = Y_r(q, \dot{q}, \dot{q}_r, \ddot{q}_{cont})$ is continuous due to $(\dot{q}_r, \ddot{q}_{cont}) \in C^1$, and $\tau_d = H(q)K_iZ$ models bounded discontinuous high frequency signals and it is considered as bounded disturbance in the controller design. This representation of robot dynamics in terms of nominal reference and its derivative will be of great importance in the next section.

Adding and subtracting (17) into (1) yields the following open-loop error dynamics

$$H(q)\dot{S}_r = -C(q,\dot{q})S_r + \tau - Y_{cont}\Theta - \tau_d \qquad (18)$$

If the regressor were known, adaptive control would suffices.

Remark 2: Adaptive-like control, with Y_{cont}. In the case when the regressor is known, it is very well known that it suffices to design an adaptive-like controller as

$$\begin{aligned} \tau &= -K_dS_r + Y_{cont}\hat{\Theta} + \tau_d &(19)\\ \dot{\hat{\Theta}} &= -\Gamma Y_{cont}S_r &(20) \end{aligned}$$

where K_d and Γ are positive definite gains of appropriate dimensions which produces an asymptotically stable closed-loop system. However, if the regressor Y_{cont} is assumed unknown then (19)-(20) cannot be implemented, *i.e.* we do neither known anything about physical structure Y_{cont} nor the parameters Θ of robot manipulator.

A great variety of approaches exist in the literature to approximate $Y_{cont}\Theta$ with neural networks, however its size and topology is fundamental to ensure a given approximation error and careful analysis is required to pick up the right neural network. In this paper a low dimensional neural network is proposed to yield bounded approximation error altogether with a smooth second order sliding mode term to finally ensure convergence, in contrast to some algorithms which ultimately bounded tracking, that is stable behavior only, not asymptotic stability.

IV. NEURAL NETWORK APPROXIMATOR

To approximate continuous regressor Y_{cont} a tree network structure that satisfied Stone-Weierstrass theorem (Cotter, 1990) is used, i.e many neurons on one layer feed a single neuron on the next layer. The input-output relationship for this generic architecture is given as $y_i = \phi(\sum_{i=1}^{n} x_iw_i)$ where $x_i \in \Re^n$ is the input to network, w_i is the weight of connections and n is the number of inputs. It is important to notice that the tree structure could have one or more hidden layers where the linear activation function is used as last stage of a multilayer neural network.

Based on this network structure, in this work we use an ADAptive LINear Element (ADALINE) proposed by Widrow and Hoff (Widrow and Hoff, 1960) which consist of a single neuron of the McCulloch-Pitts type. The Adaline expressed in matrix format is given as $y = \phi(\mathbf{X}^T\mathbf{W})$ where input vector $\mathbf{X}$ corresponds to the set of the input stimuli of the neuron, weight vector $\mathbf{W}$ corresponds to the set of synaptic strengths of the neuron[4] and the activation function $\phi(\cdot)$ represents the behavior of the neuron core[5]. When a neuron is excited it produces the output y which depends on its input and on the state of the weight vector. The weight vector $\mathbf{W}$ may be constantly modified during the training.

Definition: Let $\mathbf{K}$ be a closed bounded subset of $\Re^n$and a real vector valued functions $f(*)$ be defined on $\mathbf{K}$ as $f : \mathbf{K} \subset \Re^n \to \Re$.

Based on the Stone-Weierstrass theorem, in Cotter (1990) is showed that any smooth function $f(x) \in C^m(\mathbf{S})$, where $\mathbf{S}$ is a compact set simply connected set of $\Re^n$, can be approximated by a sufficiently large dimensional neural network, given as

$$f(x) = \phi(\mathbf{X}^T\mathbf{W}_1) \qquad (21)$$

where $\mathbf{X}$ belongs to a compact set $\mathbf{K} \subset \Re^{2n}$, that is $\mathbf{S} := \{x : \|x\| \leq \mathbf{S}\}$ and the ideal weights required for (21) are bounded by known values $\|W\| \leq W_{max}$ (Lewis *et al.*, 1996). When approximation function is done with a low dimensional neural network, a bounded functional reconstruction error $\epsilon(x)$ appears

$$\hat{f}(x) = \phi(\mathbf{X}^T\mathbf{W}_2) + \epsilon(x) \qquad (22)$$

where $\mathbf{W}_2$ is a subset of $\mathbf{W}_1$ and $\|\epsilon(x)\| \leq \epsilon_N$ with $\epsilon_N > 0$.

In this paper $f(x) = Y_{cont}\Theta$ is estimated using a low dimensional neural network, where ϕ is proposed as linear function and the training is performed on line. The unknown linear function $\hat{f}(x)$ is parameterized by static adaline neural network and it is given as

$$\hat{f}(x) = Y_{cont}\hat{\Theta} \equiv \mathbf{X}^T\mathbf{W}_2 + \epsilon(x) \qquad (23)$$

where input to the neural network $\mathbf{X}^T \in \Re^{nxp}$ is independent of the dynamics parameters and linear parameters are estimated by neural network weights $\mathbf{W}_2 \in \Re^p$. See Fig 1.

[4]The number of nodes in the input layer is equal to the number of independent variables entered into the network. The number of output nodes corresponds to the number of variables to be predicted.

[5]The adaline neuron uses linear activation function so the output of the neuron is simply the weighted inputs

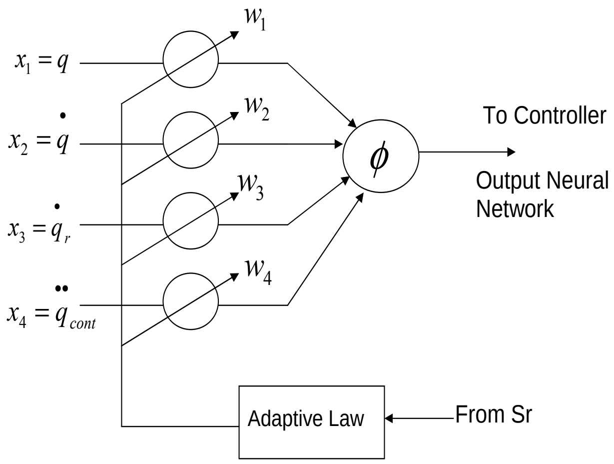

Figure 1: Proposed neural network structure

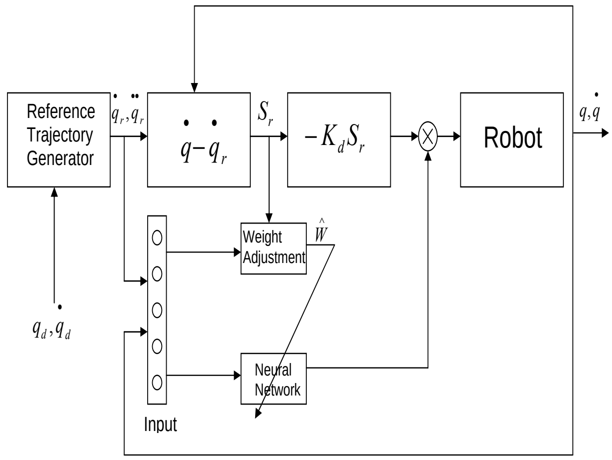

Figure 2: Joint space neuro controller structure

Remark 3: It is important to notice that the size of the neural network $2n$, can be obtained roughly by checking carefully the dynamics of a general n-link rigid arm[6]. Due to regressor Y_{cont} is formed by independent dynamic parameters, the input to the neural network is defined as

$$\mathbf{X}^T = [q, \dot{q}, \dot{q}_r, \ddot{q}_{cont}] \tag{24}$$

Remark 4: The neural network provides an approximation of Y_{cont} without worrying about its accuracy and only linear part of robot dynamics is approximated, i.e. the neural network can be considered as minimal architecture to approximate the robot dynamics taking into account the regressor elements. This architecture takes more relevance when the neural network it is driven by a second order sliding mode, as will be shown in the next section.

Remark 5: An extension of Stone-Weierstrass theorem to bounded measurable functions applying Lusin's theorem show that $\hat{f}(x)$ converge to $f(x)$ almost everywhere. The practical consequence of this result is that an infinitely large neural network can model any continuous functions while finite network might only accurately model such functions over a subset of the domain. Then extending the size and the layers of the proposed neural network a generic architecture is obtained as is reported in the literature (Lewis *et al.*, 1996; Ge and Hang, 1998).

Now we are ready to design the neuro-adaptive controller.

V. NEURO-CONTROLLER DESIGN

Substituting (23) into (18), we have

$$H(q)\dot{S}_r = -C(q,\dot{q})S_r + \tau - \mathbf{X}^T\mathbf{W} - \epsilon(x) - \tau_d \tag{25}$$

Now consider the following control law

$$\tau = -K_d S_r + \mathbf{X}^T\hat{\mathbf{W}} \tag{26}$$

[6] Without of generality, in the rest of the paper we refer $\mathbf{W}_2$ as $\mathbf{W}$, omitting its subindex.

where $K_d = K_d^T \in \Re^{nxn}$ is a gain matrix.

Substituting (26) in (25) gives rise to the closed-loop error dynamics

$$H(q)\dot{S}_r = -C(q,\dot{q})S_r - K_d S_r - \mathbf{X}^T\tilde{\mathbf{W}} - \epsilon(x) - \tau_d \tag{27}$$

where $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$.

Finally, we have the following result.

Theorem 1. Exponential Stability: Consider robot dynamics (1) in the closed-loop with the control law (26) and neuro-adaptive law

$$\dot{\hat{\mathbf{W}}} = -\Gamma \mathbf{X} S_r \tag{28}$$

where $\Gamma = \Gamma^T \in \Re^{pxp}$. Then exponential convergence of tracking errors is assured if K_d is large enough for any initial conditions and with the weight vector bounded for $t \geq \frac{|S_q(t_0)|}{\mu}$.

Proof. It is organized in three parts as follows.

Part 1: Boundedness of Closed-loop Trajectories. Consider the following Lyapunov function

$$V = \frac{1}{2}S_r^T H S_r + \frac{1}{2}\tilde{\mathbf{W}}^T\Gamma^{-1}\tilde{\mathbf{W}}, \tag{29}$$

whose total derivative along its solution (27) is as follows

$$\begin{aligned} \dot{V} &= S_r^T H \dot{S}_r + \frac{1}{2}S_r^T \dot{H} S_r + \tilde{\mathbf{W}}^T\Gamma^{-1}\dot{\tilde{\mathbf{W}}} \\ &= -S_r^T K_d S_r - S_r^T\epsilon(x) - S_r^T\tau_d + \tilde{\mathbf{W}}^T(-\mathbf{X}S_r - \Gamma^{-1}\dot{\hat{\mathbf{W}}}) \\ &= -S_r^T K_d S_r - S_r^T\epsilon(x) - S_r^T\tau_d \end{aligned} \tag{30}$$

Note that the term τ_d is radially unbounded only when $S_r \to \infty$ and for bounded signals it is zero only at $S_r = 0$. This arguments implies that $\|S_r^T\tau_d\| \leq \xi\|S_r\|$ where $\xi = \|H(q)\|\ \|K_i\|$. Then, Eq. (30) becomes

$$\dot{V} \leq -S_r^T K_d S_r + \|S_r\|\|\epsilon\| + \xi\|S_r\| \tag{31}$$

Since S_r is a function of $\Delta q, \Delta\dot{q}$ and initial conditions, then for sufficiently small initial errors belonging to a neighborhood ε with radius $r > 0$ centered in the equilibrium $S_r = 0$ and invoking Lyapunov arguments, there exists a large enough feedback gain K_d such that $K_d > \|\eta(t)\|$ and therefore, S_r converge into a bounded set ε. The boundedness of tracking error can be concluded $S_r \to \varepsilon$ as $t \to \infty$. In this way, the upper bounded of S_r is defined as

$$S_r \in \mathcal{L}_\infty \tag{32}$$

then $\|S_r\| < \varepsilon_1$ with $\varepsilon_1 > 0$. Boundedness of S_r implies the boundedness of the state of the closed loop system and that $S_q \in \mathcal{L}_\infty$ and since desired trajectories are C^2 and feedback gains are bounded, we have that $(\dot{q}_r, \ddot{q}_{cont}) \in \mathcal{L}_\infty$, which implies that $Y_{cont} \in \mathcal{L}_\infty$ and $\hat{\mathbf{W}} \in \mathcal{L}_\infty$. In this way, from Eq. (31) render $\|S_r\|\|\epsilon\| \leq \varepsilon_2$ with $\varepsilon_2 > 0$ is bounded. By virtue of $H(q)$ is positive definite and upper bounded and exists a constant $\bar{\eta}$ such that $\bar{\eta} \geq \|\eta(t)\|$, from (27) we have that

$$\begin{aligned} \dot{S}_r &= -H(q)^{-1}\{(C(q,\dot{q}) + K_d)S_r \\ &\quad -\tilde{\mathbf{W}}^T X - \epsilon(x) - \tau_d\} \\ &\leq \lambda_M(H(q)^{-1})\{(\beta_2\|\dot{q}\| + \lambda_M(K))\epsilon_1 \\ &\quad +\bar{\eta} + \epsilon_N + \xi\} \\ \|\dot{S}_r\| &\leq \zeta(t) \end{aligned} \tag{33}$$

where the bounded function $\zeta(t)$ does not depend on acceleration measurement. So far, we conclude the boundedness of all closed-loop error signals.

Part 2. Sliding Mode: Now, we show that a sliding mode at $S_q = 0$ arises for all time. If we multiply the derivative of S_r by S_q^T, and rearranging we obtain the sliding mode condition

$$\begin{aligned} S_q^T \dot{S}_q &= S_q^T(\dot{S}_r - K_i sign(S_q)) \\ &\leq |S_q^T||\dot{S}_r| - \gamma_m|S_q| \\ &\leq \zeta_{sup}|S_q| - \gamma_m|S_q| \\ &\leq -\mu|S_q| \end{aligned} \tag{34}$$

where $\mu = \gamma_m - \zeta_{sup}$ with $\gamma_m = \lambda_m(K_i)$ and ζ_{sup} is the supremum of ζ. Thus, in order to prove that $S_q \to 0$ in finite time, we can always choose $\gamma_m > \zeta_M$, in such a way that $\mu > 0$, guarantees the existence of a sliding mode at $S_q = 0$ at time $t_q \leq \frac{|S_q(t_0)|}{\mu}$. However, notice that for any initial condition $S_q(t_0) = 0$, and hence $t_q \equiv 0$ implies that a sliding mode in $S_q(t) = 0$ is enforced for all time without reaching phase and then (10) renders $S = S_d \; \forall \; t$.

Part 3: Exponential Convergence. If k in (12) is tuned large enough such that $S_d \approx 0$ for some small time $0 < t_d \ll 1$ then (10) yields

$$S = 0 \; \forall \; t \geq t_d > 0. \tag{35}$$

guarantee exponential stability of tracking errors since the solution of $S = 0$ goes to zero exponentially.

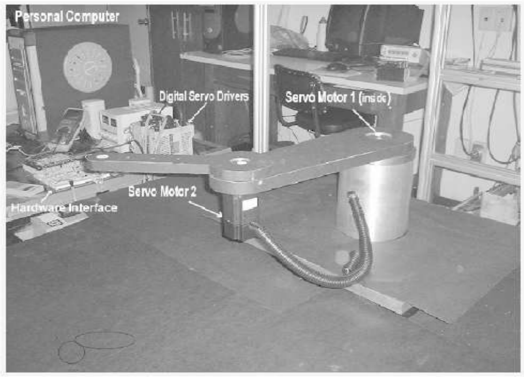

Figure 3: High performance planar manipulator

Remark 6: Passivity and Dissipativity. Given the structure and properties of the proposed neural network it is possible to show that the neuro-adaptive weight vector guarantee passivity properties on the low dimensional neural network as well as in the closed loop.

The passivity analysis of the closed loop system can be obtained as in Parra-Vega *et al.* (2003) where (29) qualifies as Lyapunov candidate function.

Now, let the feedforward block that mapping $-S_r \longrightarrow \mathbf{X}^T\mathbf{W}$ in (28). Then, we have that

$$\begin{aligned} -\int_0^t S_r^T \mathbf{X}^T \mathbf{W} d\tau &= \int_0^t \dot{\tilde{\mathbf{W}}}^T \Gamma^{-1} \tilde{\mathbf{W}} d\tau \\ &= \frac{1}{2}\int_0^t \frac{d}{dt}(\tilde{\mathbf{W}}^T \Gamma^{-1} \tilde{\mathbf{W}}) d\tau \\ = \frac{1}{2}\tilde{\mathbf{W}}^T(t)\Gamma^{-1}\tilde{\mathbf{W}}(t) &- \frac{1}{2}\tilde{\mathbf{W}}^T(0)\Gamma^{-1}\tilde{\mathbf{W}}(0) \\ &\geq -\frac{1}{2}\tilde{\mathbf{W}}^T(0)\Gamma^{-1}\tilde{\mathbf{W}}(0) \end{aligned}$$

and the mapping is passive.

By other hand, for large enough K_i the dissipativity block is established from the mapping $S_q \longrightarrow \dot{S}_q$.

Remark 7: Model-Free Control Structure. Notice that the control synthesis does not depend on any knowledge of the robot dynamics -model-free; and it keeps a very simple structure. The principal advantages of second order sliding mode with respect to others schemes (Barambones and Etxebarria, 2002; Jager, 1996; Stepanenko *et al.*,1998) is that we can guarantee smooth control input and to compensate some component of high frequency while in others approaches it is necessary chattering attenuation or chattering reduction (Lee and Choi, 2004).

Remark 8: Adaptive Neural Network. In the proposed scheme the neural network is used to compensate unknown or time varying parameters due to payloads while the second order sliding mode stabilize unmodelled dynamics and disturbances. It is fundamental to notice that the neural network is tuned by extended error S_r. Then, bounded of the weights can be assured when S_r is bounded. Furthermore it is not necessary in the control law a component to suppress the neural network reconstruction error for closed loop stability (Kwan *et al.*, 2001). In Ertugrul and Kaynak (2000) it is presented a neuro-sliding mode scheme based on two parallel neural networks. To approximate equivalent control and corrective control, respectively. The number of neurons in each neural network are determined through the design of the first order sliding mode. The chattering is eliminated defining a boundary layer nevertheless outside of boundary layer as in the reaching phase, high frequency transient may arise and the error increases.

Although the Adalines was a first one neural networks reported in the literature, recently in several approaches due to simplicity has been used to solve many problems as current compensator to achieve the selective compensation of harmonics currents in three-phase electric systems with neutral conductor (Villalva and Filho, 2006) and as data driven function approximation based on generalized adalines (Wu *et al.*,2006).
Remark 9: Comparisons. Some characteristics of the proposed scheme in comparison to other well known approaches are the following: i)The discontinuity associated to the sliding mode present in $S_r = 0$ is relegated to the first order time derivative of the $\dot{S}_r$. Furthermore, discontinuous dynamics that is imposing through $sgn(S_q)$ satisfies the sliding condition for S_q not for S_r and it avoids to use boundary layer. Then, it is guaranteed the sliding mode without chattering and without knowledge of the regressor in contrast to first order sliding mode control; ii)In contrast to adaptive control, the proposed scheme is faster and more robust given that sliding mode is induced without reaching phase, without any knowledge of the regressor and without any overparametrization; and finally iii) In contrast to adaptive (first order) sliding mode control, the proposed scheme induces a sliding mode for all time, thus it is faster and robust without any knowledge of the regressor.

VI. EXPERIMENTAL RESULTS

In this section we present the experimental results carried out on 2 degree of freedom planar robot arm (Fig. 3). The experiments were developed under LabWindows 5.0 on Pentium 4, 1.0 GHz with 256 Mb RAM under Windows 2000. Each run has an average running of 12 s. for 1 ms. sampling time. Planar manipulator control system used to demonstrate usefulness of our controller is shown in Fig. 4. The parameters of the planar robot are $m_1 = 7.19Kg$, $m_2 = 1.89Kg$, $l_1 = 0.5m$, $l_2 = 0.35m$, $l_{c1} = 0.19m$, $l_{c2} = 0.12m$, $I_1 = 0.02Kgm^2$, $I_2 = 0.016Kgm^2$ for first and second link, respectively.

The objective of these experiments is to give a desired task and the end effector follows it in finite time. The desired task is defined as a circle of radius 0.1m in 2.5s whose center located at X=(0.55,0)m in the Cartesian workspace. For each experiment we have

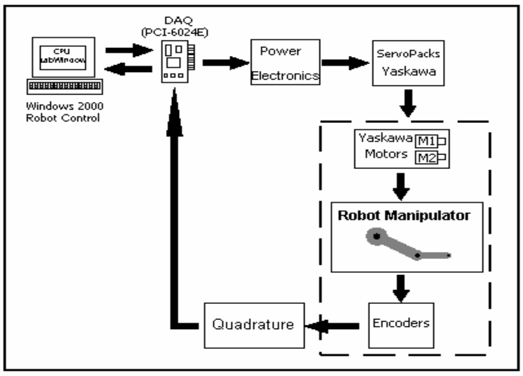

Figure 4: Two-link planar manipulator control system

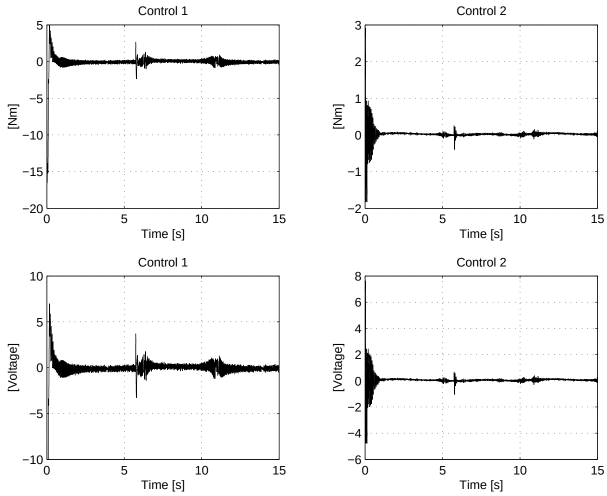

Figure 5: Theorem 1: Control input for both joints.

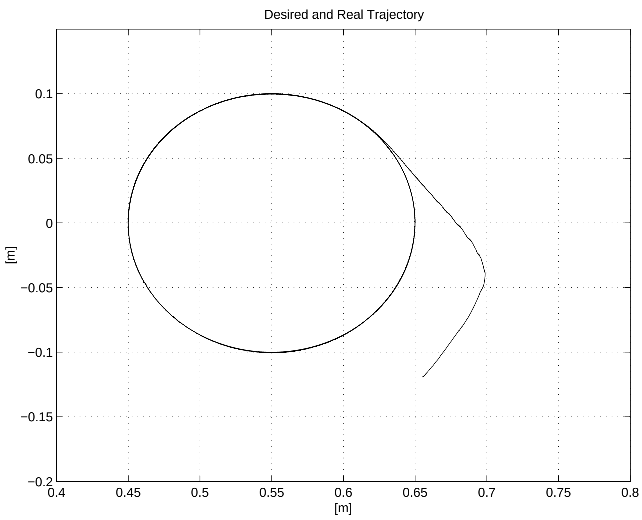

Figure 6: Theorem 1: End effector trajectory tracking.

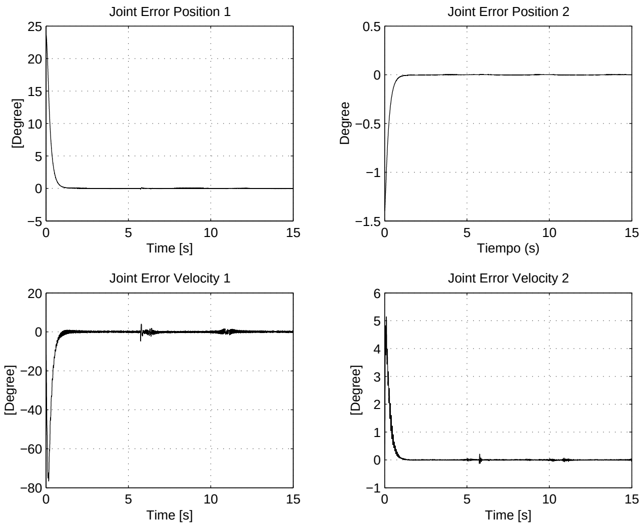

Figure 7: Theorem 1: Position and velocity tracking errors

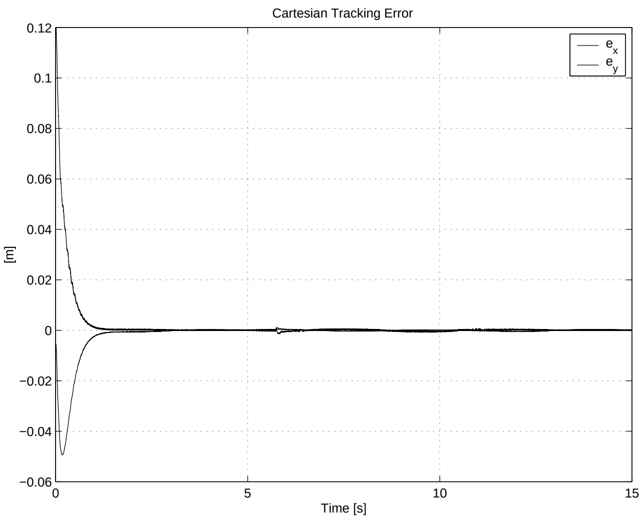

Figure 8: Theorem 1: Cartesian tracking error.

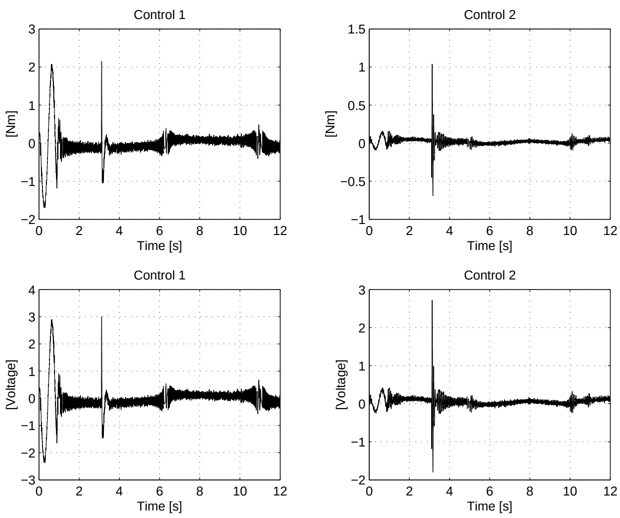

Figure 9: Control input for both joints for $t_g = 1.5s$

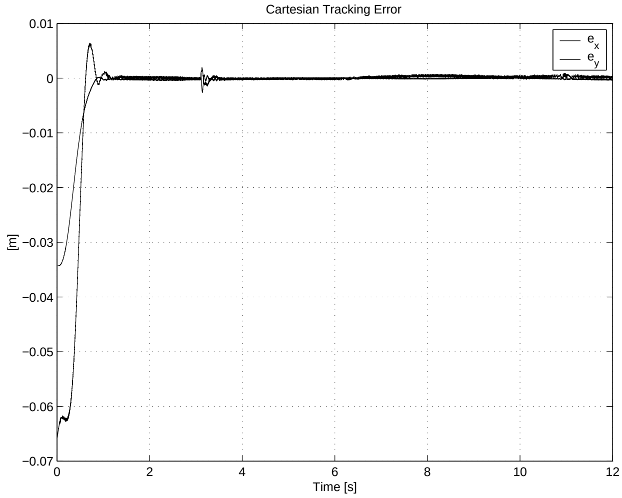

Figure 10: Cartesian tracking error for $t_g = 1.5s$

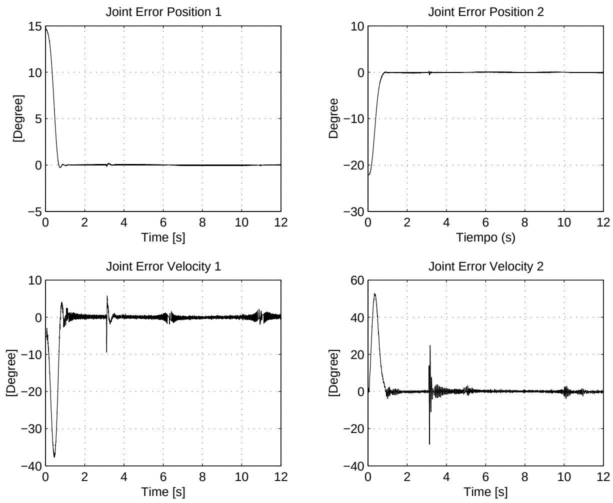

Figure 11: Position and velocity tracking errors for $t_g = 1.5s$

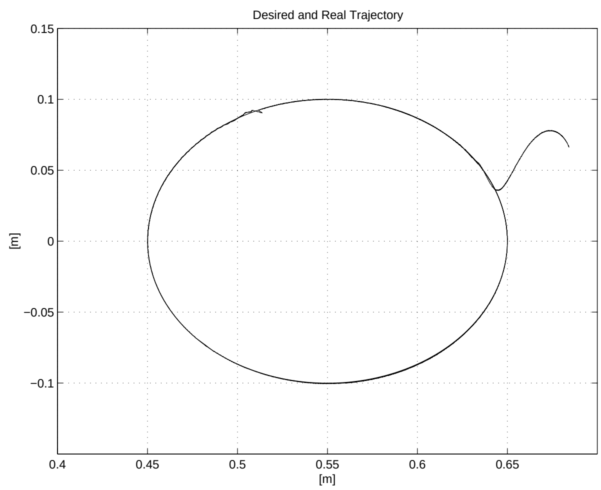

Figure 12: Phase plane for $t_g = 1.5s$

Table 1: Feedback gains

K_{d1}	K_{d2}	α_1	α_2	K_1	K_2	Γ	Figures
30	1.8	5	5	0.01	0.01	10	5-8
15	1.5	3	3	0.01	0.01	10	9-12

different initial conditions, the initializing neural network weights are zero, zero initial velocity and there is 100% of parametric uncertain i.e. neural network approximate the regressor matrix only based in the states of this function (extend error).

The performance of the proposed controller in Theorem 1 is depicted in Fig. 6. It can be seen from Fig. 5 the smooth and chattering free control input controller. Figure 7 and Fig. 8 show the position/velocity and cartesian tracking error, respectively.

In order to increase the convergence velocity of tracking error a time base generator (TBG) to induce well-posed terminal attractors is proposed in Parra-Vega (2001). The TBG sliding surface yields finite time convergence of tracking errors and it allows to obtain small error at any given time that is generally defined by the user. Defining the finite time convergence in $t_g = 1.5$ the end effector follows exactly the desired trajectory, Fig. 12. Smooth control effort is showed in Fig. 9-this frequency is normal in direct drive robots. Furthermore, the convergence of the cartesian and joint tracking errors are showed in Fig. 10 and Fig. 11, respectively.

It is important to note that the overshooting present in the both experiments in $t = 6.5s$ and $t = 3s$ approximately do not have relation with the experiment and possibly it is a problem with the data acquisition system. The feedback gains used in these experiments are given in Table 1. Finally, the feedback gains are tuned in trial-and-error basis according to the interplay of each gain in the closed loop system.

VII. CONCLUSION

A neuro-sliding controller that uses a simple continuous second order change of coordinates is presented to guarantee convergence of tracking errors. The controller uses few nodes to approximate the regressor online, and chattering is eliminated by means of the second order sliding mode surface. The experimental results demonstrate the stability properties and robustness of control scheme proposed.

REFERENCES

Arimoto, S., *Control Theory of Non-linear Mechanical Systems*, Oxford University Press (1996).

Barambones, O. and V. Etxebarria, “Robust Neural Network for Robotic Manipulators,” *Automatica,* **38,** 235-242 (2002).

Chih-Min, L. and H. Chun-Fei, “Neural Network-Based Adaptive Control for Induction Servomotor Drive System,” *IEEE Trans.on Industrial Electronics,* **49**, 115-123 (2002).

Choi, Y., M. Lee, S. Kim and Y. Kay, “Design and Implementation of an Adaptive Neural-Network Compensator for Control Systems,” *IEEE Trans. on Industrial Electronics,* **48**, 416-423 (2001).

Cotter, N.E., “The Stone-Weierstrass Theorem and Its Application to Neural Network,” *IEEE Trans. on Neural Networks,* **1**, 290-295 (1990).

Debbache, A., A. Bennia and N. Goléa, “Neural Network-based MRAC Control of Dynamic Nonlinear Systems,” *Int. J. Appl. Math. Comput. Sci.,* **16**, 219-232 (2006).

Ertugrul, M. and O. Kaynak, “Neuro Siliding Mode Control of Robotic Manipulators,” *Mechatronics,* **10**, 239-263 (2000).

Ge, S.S. and C.C. Hang, “Estructural Network Modeling and Control of Rigid Body Robots,” *IEEE Trans. on Robotics and Automation,* **14**, 823-827 (1998).

Ge, S.S. and T.H. Harris, *Adaptive Neural Network Control of Robotic Manipulators,* World Scientific, 1994.

Hayakawa, T., W. Haddad, J.W. Bailey and N. Hovakimyan, “Passivity-Based Neural Network Adaptive Outout Feedback Control for Nonlinear Nonnegative Dynamical Systems,” *IEEE Trans. on Neural Networks,* **16**, 387-398 (2005).

Jager, B., “Adaptive Robot Control with Second Order Sliding Component,” *13th IFAC Triennial World Congress,* San Francisco, USA, 271-276 (1996).

Jung, S. and T.C. Hsia, “A Study on Neural Network Control of Robot Manipulators,” *Robotica,* **14**, 7-15 (1996).

Kosmatopoulos, E.B. and M.A. Christodoulou, “Filtering, Prediction, and Learning Properties of ECE Neural Networks,” *IEEE Trans. Syst. Man Cybern.,* **24**, 971-981 (1994).

Kwan, C., D.M. Dawson and F.L. Lewis, “Robust Adaptive Control of Robots using Neural Network: Global Stability,” *Asian Journal of Control,* **3**, 111-121 (2001).

Lee, M.-J. and Y.-K. Choi, “An Adaptive Neurocontroller Using RBFN for Robot Manipulators,” *IEEE Trans. on Industrial Electronics,* **51**, 711-717 (2004).

Lewis, F.L. and C.T. Abdallah, *Control of Robot Manipulators,* Macmillan (1994).

Lewis, F.L., A. Yessildirek and K. Liu, "Multilayer Neural Net Robot Controller with Guaranteed Tracking Performance," *IEEE Trans. on Neural Networks,* **7**, 388-399 (1996).

Lin, C.H., W.D. Chou and F.J. Lin, "Stable Adaptive Control with Neural Network," *Automatica,* **36**, 5-22 (2000).

Lin, C.H., W.D. Chou and F.J. Lin, "Adaptive Hibrid Control usign a Recurrent Neural Network for a linear Synchronous Motor Servo-drive System," *IEEE Proc. Control Theory Appl.,* **148**, 156-168 (2001).

Narendra, K.S. and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. on Neural Networks,* **1**, 4-27 (1990).

Parra-Vega, V., "Second Order Sliding Mode Control for Robot Arms with Time Bse Generator for Finite-Time Tracking," *Dynamics and Control,* **11,** 174-186 (2001).

Parra-Vega, V., S. Arimoto, Y.H. Liu, G. Hirzinger and P. Akella, "Dynamic Sliding PID Control for Tracking of Robot Manipulators: Theory and Experiments," *IEEE Trans. on Robotics and Automation,* **19**, 967-976 (2003).

Sanchez, E.N., A.G. Loukianov and R.A. Felix, "Recurrent Neural Block form Control," *Automatica,* **39,** 1275-1282 (2003).

Slotine, J.J.E. and W. Li, "On the Adaptive Control of Robot Manipulator," *Int. Journal of Robotics Research,* **6**, 49-59 (1987).

Slotine, J.J.E. and M.W. Spong, "Robust Robot Control with Bounded Input Torques," *Journal of Robotics Systems,* **2**, 329-352 (1985).

Stepanenko, Y., Y. Cao and A.C. Su, "Variable Structure Control of Robotic Manipulator with PID Sliding Surfaces," *Int. Journal of Robust and Nonlinear Control,* **8,** 79-90 (1998).

Stern, H.S., "Neural Networks in Applied Statistics," *Proc. of the Statistical Computing Section,* American Statistical Association, 150-154 (1991).

Sun, F.C. and Z.Q. Sun, "Stable Neuro-Adaptive Control For Robots with the Upper Bound Estimation on the Neural Approximation Errors," *J. Intelligent and Robotic System,* **26,** 91-100 (1999).

Villalva, M.G. and E.R. Filho, "Control of a Shunt Power Filter with Neural Network-Theory and Practical Results," *IEEE Trans. on Industry Applications,* **126**, 946-953 (2006).

Widrow, B. and M.E. Hoff, "Adaptive Switching Circuits," *Institute of Radio Engineering IRE WESCON Convention Record,* NY, **4,** 96-104 (1960).

Wu, J., Z. Lin and P. Hsu, "Function Approximation Using Generalized Adalines," *IEEE Trans. on Neural Networks,* **17,** 541-558 (2006).

Yamakita, M. and T. Satoh, "Adaptive ANN Control of Robot Arm Using Structure of Lagrange Equation," *Proc. of the American Control Conference,* San Diego, 2834-2836 (1999).

Yu., W., "Passivity Analysis for Dynamic Neuro Identifier," *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications,* **50**, , 173-178 (2003).

Received: March 11, 2008
Accepted: October 10, 2008
Recommended by Subject Editor: José Guivant