INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2007; **21**:674–691
Published online 4 July 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/acs.982

Neuro-adaptive sliding-mode tracking control of robot manipulators

Andon V. Topalov[1,*,†,‡], Okyay Kaynak[2] and Gokhan Aydin[2]

[1]*Division of Electrical and Computer Engineering, College of Information Technology, Ajou University, Suwon 443-749, Republic of Korea*
[2]*Electrical and Electronic Engineering Department, Mechatronics Research and Application Center, Bogazici University, Bebek, 34342 Istanbul, Turkey*

SUMMARY

In this work, a new dynamical on-line learning algorithm for robust model-free neuro-adaptive control of a class of nonlinear systems with uncertain dynamics is proposed and experimentally tested in order to evaluate its performance and practical feasibility in industrial settings. The control application studied is the trajectory tracking control task for the first three joints of an open architecture articulated robot manipulator. The control scheme makes use of variable structure systems theory and the feedback-error-learning concept. An inner sliding motion is established in terms of the neurocontroller parameters, aiming to lead the error in its control signal towards zero. The outer sliding motion bears on the system under control, the state tracking error vector of which is simultaneously driven towards the origin of the phase space. The existing relation between the two sliding motions is shown. Experimental results illustrate that the proposed neural-network-based controller possesses a remarkable learning capability to control complex dynamical systems, virtually without requiring *a priori* knowledge of the plant dynamics and laborious start-up procedures. Copyright © 2007 John Wiley & Sons, Ltd.

Received 24 October 2006; Revised 7 May 2007; Accepted 17 May 2007

KEY WORDS: neural networks; variable-structure systems; robot manipulator; learning algorithms

*Correspondence to: Andon V. Topalov, Division of Electrical and Computer Engineering, College of Information Technology, Ajou University, Suwon 443-749, Republic of Korea.
†E-mail: topalov@tu-plovdiv.bg
‡On leave from Control Systems Department, Technical University of Sofia, campus Plovdiv, 25, Tsanko Dyustabanov Str., 4000 Plovdiv, Bulgaria.

Contract/grant sponsor: Bogazici University
Contract/grant sponsor: TUBITAK
Contract/grant sponsor: Ministry of Education and Science of Bulgaria
Contract/grant sponsor: TU Sofia

1. INTRODUCTION

The increasing complexity of today's dynamical systems is frequently coupled with unknown dynamics, modelling errors, various sorts of disturbances, uncertainties, and noise. This creates a need for advanced control design techniques that are able to overcome limitations on traditional feedback control techniques. Along these lines, numerous research activities have been reported in the literature in the area of the so-called 'universal model-free controllers' that do not require a mathematical model of the controlled system but instead are able to learn on-line the systems that they are to control so that the performance can automatically be improved.

The use of artificial neural networks (NNs) is a common suggestion in this respect and NN-based controllers have nowadays entered the mainstream of control theory as a natural extension of adaptive control to systems that are nonlinear in the tunable parameters [1]. Most of the existing training methods for NNs rely on the gradient descent methodology and involve the computation of partial derivatives, or sensitivity functions. In this respect, they can be considered as extensions of the well-known backpropagation algorithm for multilayer feedforward neural networks (MFNNs) [2] and hence they inherit some of its major drawbacks among which, in particular, is the difficulty to obtain analytical results concerning the convergence and stability of the learning schemes [3]. Recent research on the application of NNs for control has begun to address the closed-loop system structure and stability issues more rigorously [4]. The investigations in this area have been split over two main research directions. It has been shown in several works that the Lyapunov approach can be directly implemented to obtain robust training algorithms for continuous-time NNs [5–8]. Another proposed way to design a robust learning scheme is to utilize the variable structure systems (VSS) theory in constructing the parameter adaptation mechanism of the NNs [9–15] since the robustness of the variable structure control (VSC) scheme against unmodelled dynamics, disturbances, time delays and nonlinearities is well known [16].

Robotic manipulators are hard to control nonlinear systems that are frequently used as a test bed for evaluation of new control methods. Their coupled nonlinear equations with time-varying parameters and the ambiguities in the friction-related dynamics inevitably require the use of flexible control architectures. In the past decade, the applications of intelligent control techniques (fuzzy control or NN control) to the motion control for robot manipulators have received considerable attention [11, 17–21].

The motivation of this study is to design a stable neuro-adaptive control scheme for tracking control of rigid-link robot manipulator with a guaranteed error convergence and without a requirement for prior knowledge of the dynamics of the controlled plant. The control scheme makes use of VSS theory [16] and the *feedback-error-learning* concept [22, 23]. The effectiveness of the proposed approach has been experimentally tested and its potentialities from a practical point of view, in terms of robustness and stability, have been verified.

Section 2 presents the developed on-line learning algorithm, the sliding mode feedback-error-learning control scheme and introduces the equivalency constraints on the sliding control performance for the plant and the learning performance for the neural network feedback controller (NNFC). Section 3 describes an experimental real-time application of the proposed control scheme to the trajectory tracking control task, simultaneously implemented to the first three joints of a five degrees-of-freedom open architecture (OA) articulated robot manipulator (model CRS255, also known as CataLyst-5, produced by Quanser Consulting Inc.). Finally, Section 4 summarizes the results of this investigation.

DOI: 10.1002/acs

2. THE SLIDING MODE FEEDBACK-ERROR-LEARNING APPROACH

2.1. Initial assumptions and definitions

Consider the two-layered feedforward NN with a scalar output, implemented as a NNFC. Its topology is shown in Figure 1. The following definitions will be used:

$X(t)=[x_1(t),\ldots,x_p(t)]^{\mathrm{T}}$ is the vector of the time-varying input signals augmented by the bias term.
$T_H^n(t)=[\tau_{H1}^n(t),\ldots,\tau_{Hn}^n(t)]^{\mathrm{T}}$ is the vector of the output signals of the neurons in the hidden layer.
$\tau^n(t)$ is the scalar signal representing the time-varying output of the network.
$W1(t)_{(n\times p)}$ is the matrix of the time-varying connections' weights between the neurons in the input and the hidden layer, where each matrix's element $w1_{i,j}(t)$ denotes the weight of the connection of the neuron i from its input j.
$W2(t)_{(1\times n)}=[w2_1(t),\ldots,w2_n(t)]$ is the vector of the connections' weights between the neurons in the hidden layer and the output node. Both $W1(t)_{(n\times p)}$ and $W2(t)_{(1\times n)}$ are considered augmented by including the bias weight components for the neurons in the hidden layer and the output neuron, respectively.
$f(\cdot)$ is the nonlinear, differentiable, monotonously increasing activation function of the neurons in the hidden layer of the network (e.g. log-sigmoid or tan-sigmoid function).

The derivative of the activation function $f(\cdot)$ of the neuron i from the hidden layer is denoted as $A_i(t)$, where

$$0<A_i(t)=\frac{\mathrm{d}}{\mathrm{d}t}\left[f\left(\sum_{j=1}^{p} w1_{i,j}x_j\right)\right]\leqslant B_A \quad \forall i,\, j \tag{1}$$

and B_A corresponds to its maximum value.

The neuron in the output layer is considered with a linear activation function.

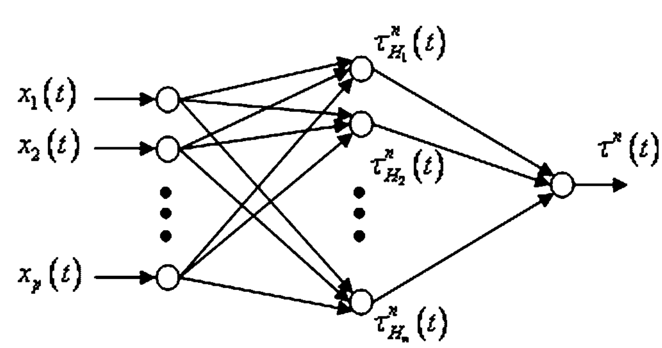

Figure 1. Multilayer feedforward neural network used as a controller.

DOI: 10.1002/acs

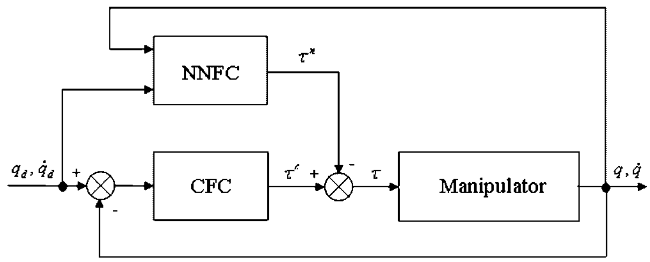

Figure 2. Block diagram of nonlinear regulator sliding mode feedback-error-learning scheme.

The output signal $\tau^n_{H_i}$ of the ith neuron from the hidden layer and the output signal of the network $\tau^n(t)$ are defined as follows:

$$\tau^n_{H_i} = f\left(\sum_{j=1}^{p} w1_{i,j} x_j\right) = f(W1_i X) \tag{2}$$

$$\tau^n(t) = \sum_{i=1}^{n} w2_i \tau^n_{H_i} = W2 T^n_H \tag{3}$$

The NNFC is assumed to operate within an adaptive control scheme, the general structure of which is presented in Figure 2. A proportional plus derivative (PD) controller (depicted as a conventional feedback controller (CFC) block in Figure 2) is provided both as an ordinary feedback controller to guarantee global asymptotic stability in compact space and as an inverse reference model of the response of the system under control.

It will be assumed (due to the existence of a CFC controller in the scheme) that the input vector of the NNFC and its time derivative are bounded, i.e.

$$\|X(t)\| = \sqrt{x_1^2(t) + \cdots + x_p^2} \leqslant B_X \quad \forall t \tag{4}$$

$$\|\dot{X}(t)\| = \sqrt{\dot{x}_1^2(t) + \cdots + \dot{x}_p^2(t)} \leqslant B_{\dot{X}} \quad \forall t \tag{5}$$

with B_X and $B_{\dot{X}}$ being known positive constants.

Due to the physical constraints, it is also assumed that the magnitude of all vectors row $W1_i(t)$ constituting the matrix $W1(t)$ and the elements of the vector $W2(t)$ are bounded, i.e.

$$\|W1_i(t)\| = \sqrt{w1_{i,1}^2(t) + w1_{i,2}^2(t) + \cdots + w1_{i,p}^2(t)} \leqslant B_{W1} \quad \forall t \tag{6}$$

$$|w2_i(t)| \leqslant B_{W2} \quad \forall t \tag{7}$$

for some known constants B_{W1} and B_{W2}, where $i = 1, 2, \ldots, n$.

$\tau(t)$ and $\dot{\tau}(t)$ are considered also as bounded signals, i.e.

$$|\tau(t)| \leqslant B_\tau, \quad |\dot{\tau}(t)| \leqslant B_{\dot{\tau}} \quad \forall t \tag{8}$$

where B_τ and $B_{\dot{\tau}}$ are positive constants.

DOI: 10.1002/acs

2.2. The VSC-based on-line learning algorithm

In the feedback-error-learning approach proposed in [23], the control signal of the CFC is used as a learning error for the NNFC. A VSC-based on-line learning algorithm is applied to the NNFC in this investigation to replace the gradient-based algorithm used earlier in [22, 23]. This leads to the establishment of an inner sliding motion in terms of the neurocontroller parameters, aiming to lead the learning error towards zero. Thus, the control of the plant is gradually transferred from CFC to NNFC. Moreover, an outer sliding motion also takes place. It bears on the system under control, the state tracking error vector of which is simultaneously driven towards the origin of the phase space. In Section 2.3, the relation between the two sliding motions will be shown. The sliding surface for the system under control $s_p(e, \dot{e})$ and the zero adaptive learning error level for the controller $s_c(\tau, \tau^n)$ are defined as follows:

$$s_p(e, \dot{e}) = \dot{e} + \lambda e \tag{9}$$

$$s_c(\tau^n, \tau) = \tau^c = \tau^n + \tau \tag{10}$$

with λ being a constant determining the slope of the sliding surface.

Definition 1
A sliding motion will have a place on a sliding surface $s_c(\tau^n, \tau) = \tau^c(t) = 0$, after a hitting time t_h, if the condition $s_c(t)\dot{s}_c(t) = \tau^c(t)\dot{\tau}^c(t) < 0$ is satisfied for all t in some nontrivial semi-open subinterval of time of the form $[t, t_h) \subset (-\infty, t_h)$.

The learning algorithm for the neurocontroller weights $W1(t)$ and $W2(t)$ has to be derived in such a way that the sliding mode condition of Definition 1 will be enforced.

Let us denote as 'sign(s_c)' the signum function, defined as follows:

$$\operatorname{sign}(s_c) = \begin{cases} 1 & \text{for } s_c(t) > 0 \\ 0 & \text{for } s_c(t) = 0 \\ -1 & \text{for } s_c(t) < 0 \end{cases} \tag{11}$$

To enable $s_c = 0$ is reached, the following theorem is introduced.

Theorem 1
If the adaptation law for the weights $W1(t)$ and $W2(t)$ of NNFC is chosen, respectively, as

$$\dot{w}1_{i,j} = -\left(\frac{w2_i x_j}{X^{\mathrm{T}} X}\right) \alpha \operatorname{sign}(s_c) \tag{12}$$

$$\dot{w}2_i = -\left[\frac{\tau^n_{H_i}}{(T^n_H)^{\mathrm{T}} T^n_H}\right] \alpha \operatorname{sign}(s_c) \tag{13}$$

with α being sufficiently large positive constant satisfying

$$\alpha > n B_A B_{W1} B_{\dot{X}} B_{W2} + B_{\dot{\tau}} \tag{14}$$

then, given an arbitrary initial condition $s_c(0)$, the learning error $\tau^c(t)$will converge to zero during a finite time t_{h} which may be estimated as

$$t_{\mathrm{h}} \leqslant \frac{|s_c(0)|}{\alpha - nB_A B_{W1} B_{\dot{X}} B_{W2} + B_{\dot{\tau}}} \tag{15}$$

and a sliding motion will be maintained on $\tau^c = 0$ for all $t > t_{\mathrm{h}}$.

Proof
Consider $V_c = \frac{1}{2}s_c^2$ as a Lyapunov function candidate. Then, differentiating V_c yields

$$\begin{aligned}
\dot{V}_c &= s_c\dot{s}_c = s_c(\dot{\tau}^n + \dot{\tau}) = s_c\left\{\frac{\mathrm{d}}{\mathrm{d}t}\left[\sum_{i=1}^{n} w2_i f\left(\sum_{j=1}^{p} w1_{i,j}x_j\right)\right] + \dot{\tau}\right\} \\
&= s_c\left[\sum_{i=1}^{n} \dot{w}2_i\tau_{H_i}^n + \sum_{i=1}^{n} w2_i A_i \sum_{j=1}^{p}(\dot{w}1_{i,j}x_j + w1_{i,j}\dot{x}_j) + \dot{\tau}\right] \\
&= s_c\left[-\sum_{i=1}^{n}\frac{\tau_{H_i}^n}{(T_H^n)^{\mathrm{T}}T_H^n}\alpha\,\mathrm{sign}(s_c)\tau_{H_i}^n\right. \\
&\quad \left. +\sum_{i=1}^{n} A_i\sum_{j=1}^{p}\left(-\left(\frac{w2_i x_j}{X^{\mathrm{T}}X}\right)\alpha\,\mathrm{sign}(s_c)x_j w2_i + w1_{i,j}\dot{x}_j w2_i\right) + \dot{\tau}\right] \\
&= s_c\left(-\alpha\,\mathrm{sign}(s_c) - \sum_{i=1}^{n} A_i\alpha w2_i^2\,\mathrm{sign}(s_c) + \sum_{i=1}^{n} A_i w2_i\sum_{j=1}^{p} w1_{i,j}\dot{x}_j + \dot{\tau}\right) \\
&= -\left(\alpha + \alpha\sum_{i=1}^{n} A_i w2_i^2\right)|s_c| + \left(\sum_{i=1}^{n} A_i w2_i\sum_{j=1}^{p} w1_{i,j}\dot{x}_j + \dot{\tau}\right)s_c \\
&\leqslant -\alpha|s_c| + s_c\left(\sum_{i=1}^{n} A_i w2_i\sum_{j=1}^{p} w1_{i,j}\dot{x}_j + \dot{\tau}\right) \\
&\leqslant -\alpha|s_c| + (nB_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{\tau}})|s_c| \\
&= |s_c|(-\alpha + nB_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{\tau}}) < 0 \quad \forall s_c \neq 0
\end{aligned} \tag{16}$$

Inequality (16) means that by applying the adaptation law for the weights of the NNFC as in (12) and (13) the controlled trajectories of the learning error $s_c(t)$ will converge to zero in a stable manner since the negative definiteness of the time derivative of the above Lyapunov function is ensured. □

It is possible now to be shown that such a convergence will take place in finite time. The differential equation that is satisfied by the controlled error trajectories $s_c(t)$ is as

DOI: 10.1002/acs

follows:

$$\dot{s}_c(t) = -\left(1 + \sum_{i=1}^{n} A_i w2_i^2\right)\alpha\,\mathrm{sign}(s_c) + \sum_{i=1}^{n} A_i w2_i \sum_{j=1}^{p} w1_{i,j}\dot{x}_j + \dot{\tau} \tag{17}$$

For any $t \leqslant t_\mathrm{h}$, the solution, $s_c(t)$, of this equation, with initial condition $s_c(0)$ at $t=0$, satisfies

$$\begin{aligned} s_c(t) - s_c(0) &= \int_0^t \dot{s}_c(\sigma)\,\mathrm{d}\sigma \\ &= \int_0^t \Bigg[-\alpha\,\mathrm{sign}(s_c(\sigma))\left(1 + \sum_{i=1}^{n} A_i(\sigma) w2_i^2(\sigma)\right) \\ &\quad + \sum_{i=1}^{n} A_i(\sigma) w2_i(\sigma) \sum_{j=1}^{p} w1_{i,j}(\sigma)\dot{x}_j(\sigma) + \dot{\tau}(\sigma)\Bigg]\mathrm{d}\sigma \end{aligned} \tag{18}$$

At time $t = t_\mathrm{h}$ the solution takes zero value and, therefore,

$$\begin{aligned} -s_c(0) &= -\alpha\,\mathrm{sign}(s_c(0))\left[t_\mathrm{h} + \int_0^{t_\mathrm{h}} \left(\sum_{i=1}^{n} A_i(t) w2_i^2(t)\right)\mathrm{d}t\right] \\ &\quad + \int_0^{t_\mathrm{h}} \left(\sum_{i=1}^{n} A_i(t) w2_i(t) \sum_{j=1}^{p} w1_{i,j}(t)\dot{x}_j(t) + \dot{\tau}(t)\right)\mathrm{d}t \end{aligned} \tag{19}$$

By multiplying both sides of the equation with $-\mathrm{sign}(s_c(0))$ the estimate of t_h in (15) can be found using the following inequality:

$$\begin{aligned} |s_c(0)| &= \alpha t_\mathrm{h} + \alpha \int_0^{t_\mathrm{h}} \left(\sum_{i=1}^{n} A_i(t) w2_i^2(t)\right)\mathrm{d}t \\ &\quad - \mathrm{sign}(s_c(0)) \int_0^{t_\mathrm{h}} \left(\sum_{i=1}^{n} A_i(t) w2_i(t) \sum_{j=1}^{p} w1_{i,j}(t)\dot{x}_j(t) + \dot{\tau}(t)\right)\mathrm{d}t \\ &\geqslant \alpha\left(t_\mathrm{h} + \int_0^{t_\mathrm{h}} \left(\sum_{i=1}^{n} A_i(t) w2_i^2(t)\right)\mathrm{d}t\right) - (n B_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{\tau}}) t_\mathrm{h} \\ &\geqslant [\alpha - (n B_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{\tau}})] t_\mathrm{h} \end{aligned} \tag{20}$$

Obviously, for all times $t < t_\mathrm{h}$, taking into account the sliding mode controller gain α in (14), it follows from (17) that

$$\begin{aligned} s_c(t)\dot{s}_c(t) &= -\alpha |s_c(t)| \left(1 + \sum_{i=1}^{n} A_i(t) w2_i^2(t)\right) \\ &\quad + \left(\sum_{i=1}^{n} A_i(t) w2_i(t) \sum_{j=1}^{p} w1_{i,j}(t)\dot{x}_j(t) + \dot{\tau}(t)\right) s_c(t) \\ &\leqslant (-\alpha + n B_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{\tau}}) |s_c(t)| < 0 \end{aligned} \tag{21}$$

and a sliding motion is sustained on $s_c(t) = 0$ for $t > t_\mathrm{h}$.

Let us now analyse the case when a bounded noise is added to the NNFC inputs. It will be assumed that $H(t)=[\eta_1(t),\ldots,\eta_p(t)]^{\mathrm{T}}$ is an augmented vector-valued norm-bounded external perturbation input which is added to the input vector $X(t)$. Its last element $\eta_p(t)$ is considered to be equal to zero. This means that the constant input to the bias weight is regarded as a fixed value without an influence of perturbation signals on it.

It is also assumed that the perturbation input $H(t)$ is not 'larger' than the input $X(t)$, i.e.

$$\|H(t)\|=\sqrt{\eta_1^2(t)+\cdots+\eta_p^2(t)}\leqslant B_H<B_X \quad \forall t \tag{22}$$

$\dot{H}(t)$ is assumed to be also bounded, i.e.

$$\|\dot{H}\|=\sqrt{\dot{\eta}_1^2(t)+\cdots+\dot{\eta}_p^2(t)}\leqslant B_{\dot{H}} \quad \forall t \tag{23}$$

The perturbed output signal $\hat{\tau}^n_{H_i}$ of the ith neuron from the hidden layer and the perturbed output signal of the neurocontroller $\hat{\tau}^n$ are defined as follows:

$$\hat{\tau}^n_{H_i}=f\left[\sum_{j=1}^{p} w1_{i,j}(x_j+\eta_j)\right]=f[W1_i(X(t)+H(t))] \tag{24}$$

$$\hat{\tau}^n=\sum_{i=1}^{n} w2_i\hat{\tau}^n_{H_i}=W2\hat{T}^n_H \tag{25}$$

where $\hat{T}^n_H=(\hat{\tau}^n_{H_1},\ldots,\hat{\tau}^n_{H_n})$.

Hence, the learning error can now be expressed as follows:

$$\hat{s}_c=\hat{\tau}^n+\tau=\sum_{i=1}^{n} w2_i f\left[\sum_{j=1}^{p} w1_{i,j}(x_j+\eta_j)\right]+\tau \tag{26}$$

Let us consider the following perturbed adaptation law for the weights in NNFC:

$$\dot{w}1_{i,j}=-\left(\frac{w2_i(x_j+\eta_j)}{(X+H)^{\mathrm{T}}(X+H)}\right)\alpha\,\mathrm{sign}(\hat{s}_c) \tag{27}$$

$$\dot{w}2_i=-\left[\frac{\hat{\tau}_{Hi}}{(\hat{T}^n_H)^{\mathrm{T}}\hat{T}^n_H}\right]a\,\mathrm{sign}(\hat{s}_c) \tag{28}$$

It is easy to be verified that the weights adaptation law (27), (28) results in the following perturbed learning error dynamics:

$$\dot{\hat{s}}_c(t)=-\left(1+\sum_{i=1}^{n}\hat{A}_i w2_i^2\right)\alpha\,\mathrm{sign}(\hat{s}_c)+\sum_{i=1}^{n}\hat{A}_i w2_i\sum_{j=1}^{p} w1_{i,j}(\dot{x}_j+\dot{\eta}_j)+\dot{\tau} \tag{29}$$

where

$$0<\hat{A}_i(t)=\frac{\mathrm{d}}{\mathrm{d}t}\left\{f\left[\sum_{j=1}^{p} w1_{i,j}(x_j+\eta_j)\right]\right\}\leqslant B_A \quad \forall i,j$$

DOI: 10.1002/acs

The robustness result is summarized in the following theorem whose proof is similar to that of Theorem 1.

Theorem 2
A sliding motion will have place on the zero learning error manifold of the NNFC, including a perturbed input vector, if the adaptation law for the weight vectors $w1_{i,j}$ and $w2_i$ is chosen as in (27) and (28) with α being a positive constant satisfying

$$\alpha > nB_A B_{W1} B_{W2}(B_{\dot{X}} + B_{\dot{H}}) + B_{\dot{\tau}} \tag{30}$$

For any arbitrary initial condition $\hat{s}_c(0)$, the perturbed learning error will converge to zero in $\hat{t}_{\mathrm{h}}$, estimated by

$$\hat{t}_{\mathrm{h}} \leqslant \frac{|\hat{s}_c(0)|}{\alpha - nB_A B_{W1} B_{W2}(B_{\dot{X}} + B_{\dot{H}}) + B_{\dot{\tau}}} \tag{31}$$

despite all possible assumed (bounded) values of the perturbation inputs and their time derivatives. Moreover, a sliding motion is sustained on $\hat{s}_c(t) = 0$ for all $t > \hat{t}_{\mathrm{h}}$.

2.3. Relation between the VSC-based learning of the controller and the sliding motion in the behaviour of the system

A relation between the sliding manifold s_p, as defined in (9), and the zero adaptive learning error level s_c in (10), can be established when the slope constant λ is taken as

$$\lambda = \frac{k_P}{k_D} \tag{32}$$

It is then determined by the following equation:

$$s_c = \tau^c = k_D \dot{e} + k_P e = k_D \left(\dot{e} + \frac{k_P}{k_D} e \right) = k_D s_p \tag{33}$$

where k_D and k_P are the gains of the PD controller.

The tracking performance of the system under control can be analysed by introducing $V_p = \frac{1}{2} s_p^2$ as a Lyapunov function candidate.

Theorem 3
If the adaptation strategy for the adjustable parameters of the NNFC is chosen as in Equations (12) and (13), then the negative definiteness of the time derivative of the above Lyapunov function is ensured.

Proof
Evaluating the time derivative of the Lyapunov function V_p yields:

$$\begin{aligned} \dot{V}_p &= \dot{s}_p s_p = \frac{1}{k_D^2} \dot{s}_c s_c \\ &\leqslant \frac{1}{k_D^2} |s_c| (-\alpha + nB_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{\tau}}) < 0 \quad \forall s_c, s_p \neq 0 \end{aligned} \tag{34}$$

□

DOI: 10.1002/acs

Remark 1

The obtained results mean that, assuming the VSC task is achievable, utilization of τ^c as the learning error for the NNFC together with the tuning law of (12) and (13) enforces the desired reaching mode followed by the sliding regime for the system under control. It is straightforward to prove that the hitting occurs in finite time (see the second part of the proof of Theorem 1).

3. TRAJECTORY TRACKING CONTROL OF AN ARTICULATED ROBOT MANIPULATOR

The effectiveness of the proposed sliding mode neuro-adaptive control approach has been experimentally tested by implementing the control scheme in real-time simultaneously for trajectory tracking control of the first three joints of an OA articulated robot (CRS255/CataLyst-5, produced by Quanser Consulting Inc.).

3.1. The CRS255/CataLyst-5 open architecture articulated robot manipulator

The CRS255 is five degrees-of-freedom articulated robot manipulator (see Figure 3). It has five joints powered by five motors. Figure 4 specifies the joints of the robot while it is in its home position. The system is supplied with a CRS C500 controller and has all the capabilities of an industrial robot.

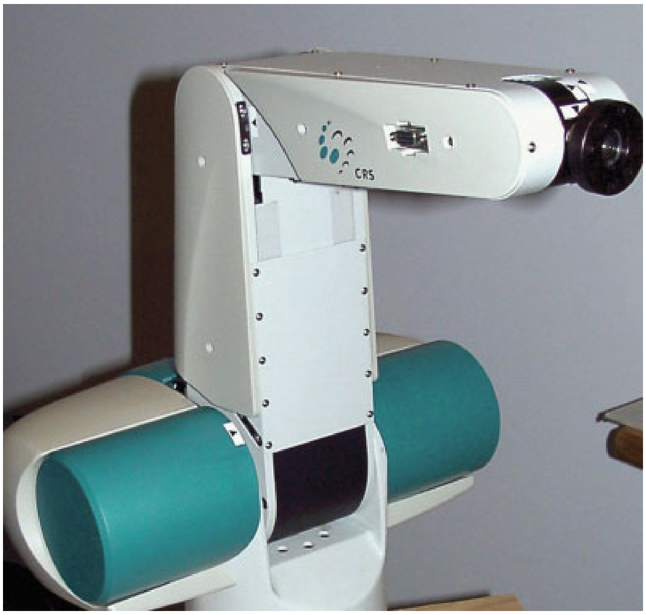

Figure 3. The CRS255/CataLyst-5 manipulator.

DOI: 10.1002/acs

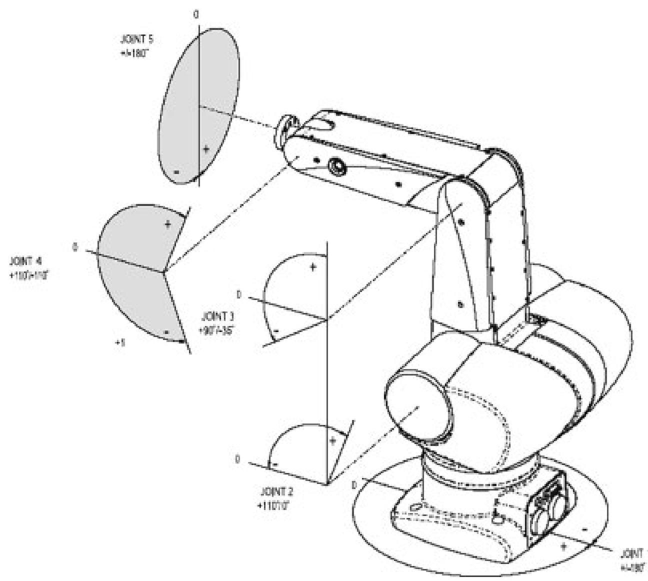

Figure 4. A detailed view of the axes of rotations and their ranges.

An OA means that there exists an additional possibility to control the manipulator from a personal computer (PC), equipped with a data acquisition and control board, by directly sending voltages to control the motors. The system uses the MultiQ-3 data acquisition and control board produced by Quanser Consulting Inc. The OA system allows the user to switch back and forth from its own controller to the C500 controller seamlessly. The WinCon software package, supplied with the system, allows the user to run controllers directly from Matlab/Simulink environment. It is a real-time Windows application that runs Simulink generated code using the Matlab Real-Time Workshop package.

The block diagram in Figure 5 shows the OA approach. Through the analog switches all signals can be routed to and from the MultiQ board as opposed to the CRS C500 controller. This allows full control over the motors and direct measurements from the robot encoders and other signals of relevance.

3.2. Experiments carried on with a smoothed square wave reference signal

Three identical NNs (one per joint) have been used as NNFCs. The implemented network structure of each neurocontroller has been with an input layer consisting of four neurons, a hidden layer with five neurons and one neuron in the output layer. The initial weights have been randomly generated by using the Matlab command *randn* with a subsequent division by 15. The parameter α of the sliding mode learning algorithm has been selected as $\alpha = 0.5$.

The square wave signals used as reference signals to be followed have been smoothed using a filter consisting of two consecutively connected blocks each with a transfer function $G(s) = 25/(s^2 + 10s + 25)$. The parameters of the reference signals for each controlled joint have

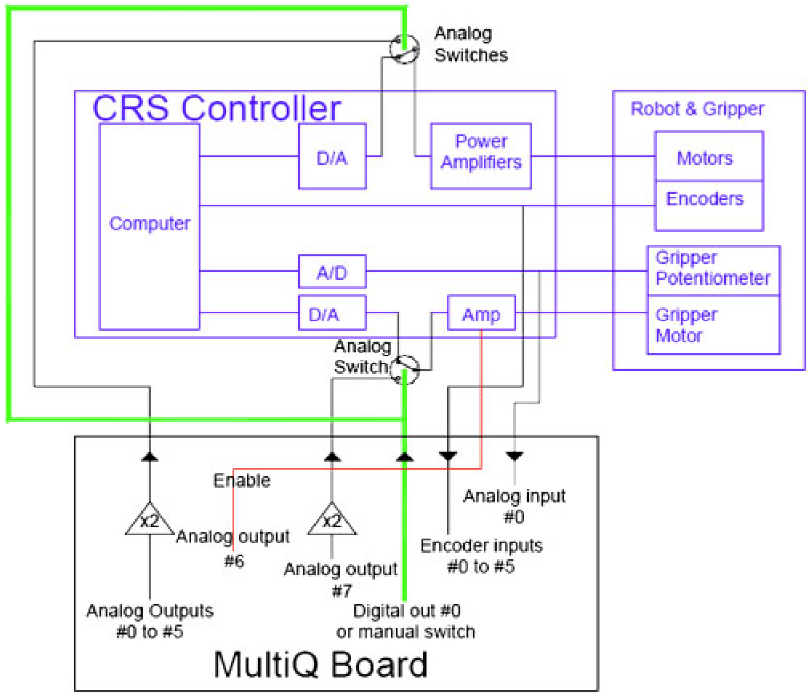

Figure 5. Open architecture operation using the MultiQ data acquisition and control board.

been chosen as follows:

Joint 1: amplitude of the signal -8, bias 7.7, and frequency $\pi/2.4$ rad/s.
Joint 2: amplitude of the signal 4, bias -20, and frequency $\pi/3.6$ rad/s.
Joint 3: amplitude of the signal 12, bias 30, and frequency $\pi/4.8$ rad/s.

The *ode1* ordinary differential equation solver, implementing Euler numerical integration method with a fixed time step of 0.001 s has been applied when running the experiments in Matlab/Simulink environment.

During the tracking process of the reference trajectories, the control signals of all three robot joints have been simultaneously switched from using only conventional PD controllers to the proposed sliding mode feedback-error-learning neurocontrol schemes. The gains of the implemented PD controllers have been taken to be very close to those proposed by Quanser Consulting Inc. for the initial set-up of the manipulator. The recorded experimental results are presented in Figures 6–9. The response driven by the linear controller only is characterized by a larger value of the tracking error because of the existing nonlinearity, time-varying inertia and gravitational loads, and joint friction model uncertainties. Since the NNFC can compensate for these phenomena through learning, the actual response approached better the desired response. It can be seen that after the performed switching to the proposed neurocontrol scheme, the CFC torque signals are suppressed by the NNFCs, the tracking errors are significantly decreased and the joint outputs closely follow the required trajectories demonstrating a very good tracking performance of the investigated control scheme.

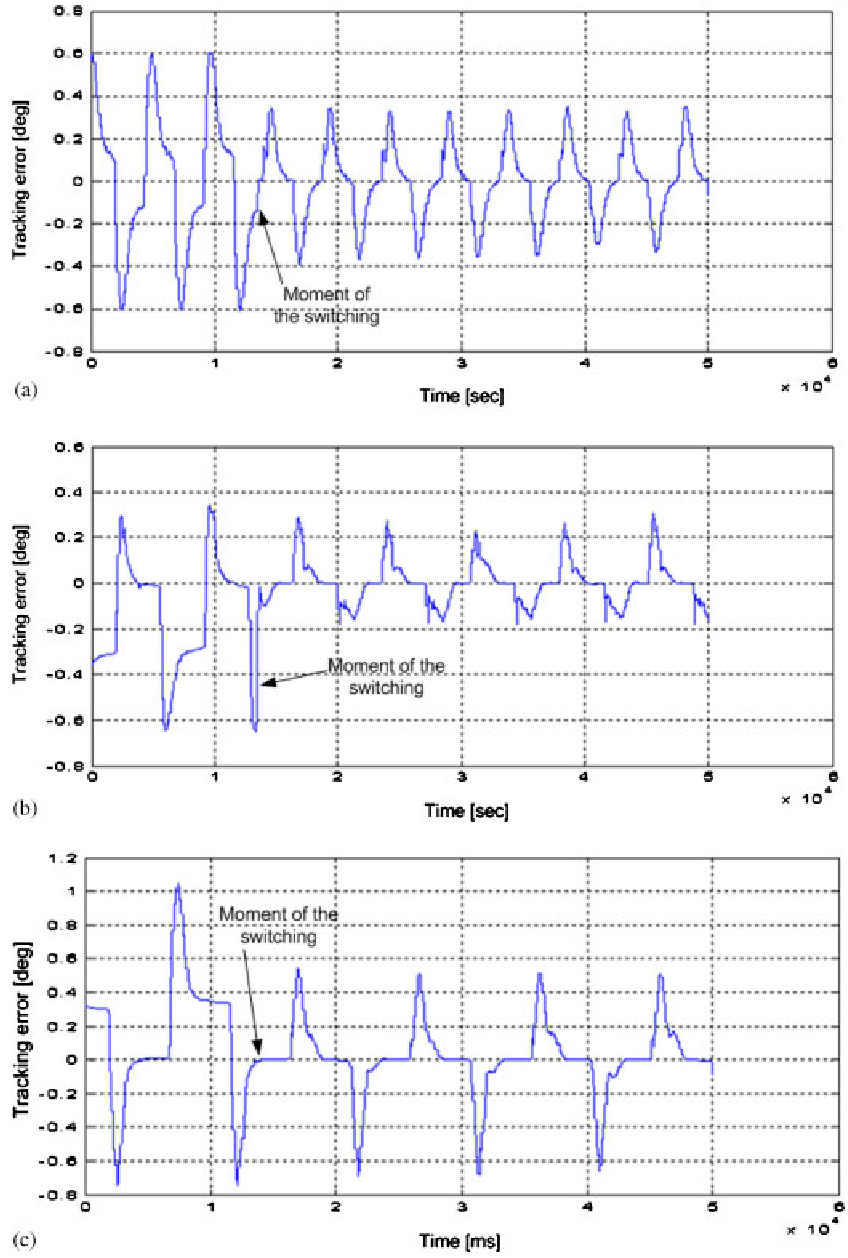

Figure 6. Changes in the tracking error (in degrees) for (a) joint 1, (b) joint 2 and (c) joint 3 when switching from conventional PD to sliding mode neuro-adaptive control.

DOI: 10.1002/acs

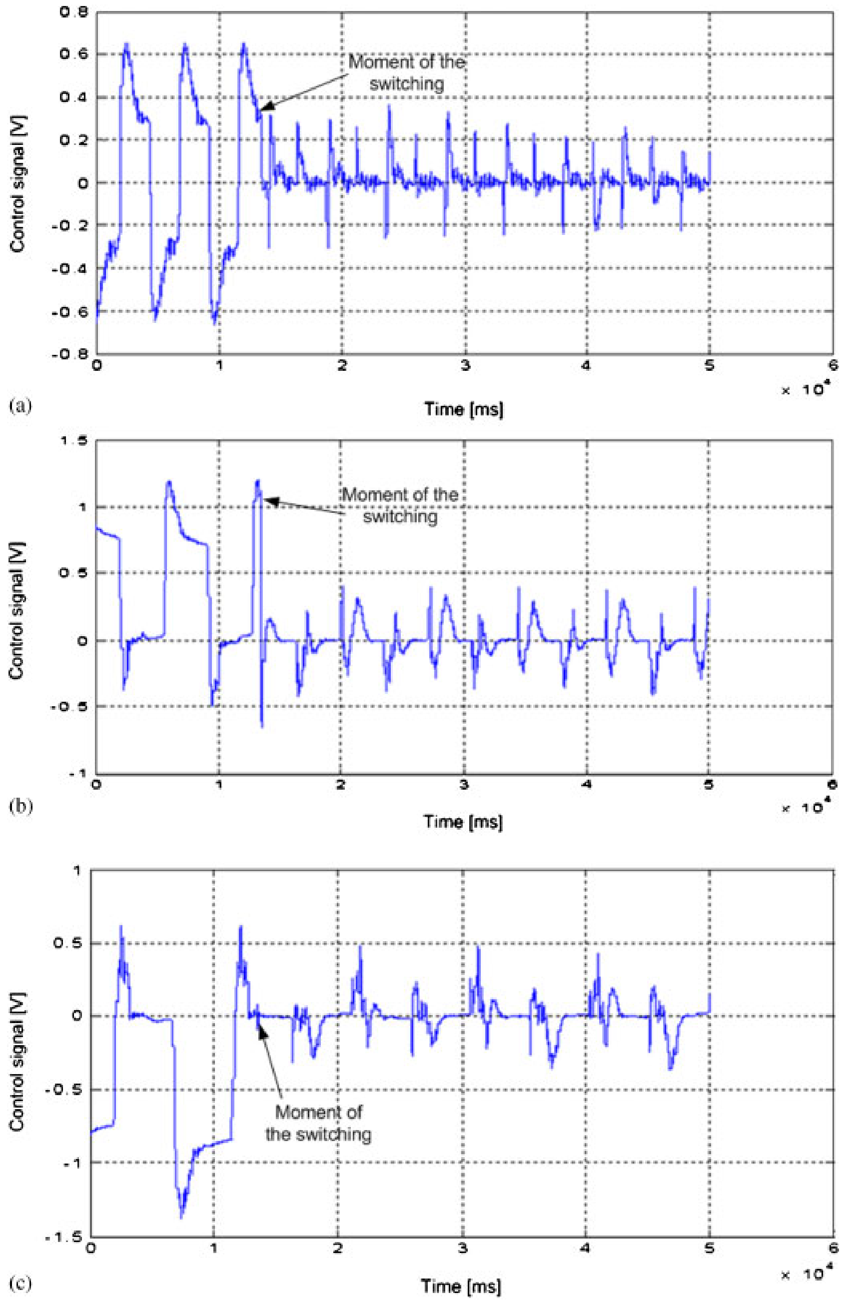

Figure 7. Changes of the PD control signal (torque) for (a) joint 1, (b) joint 2 and (c) joint 3 when switching to sliding mode neuro-adaptive control.

DOI: 10.1002/acs

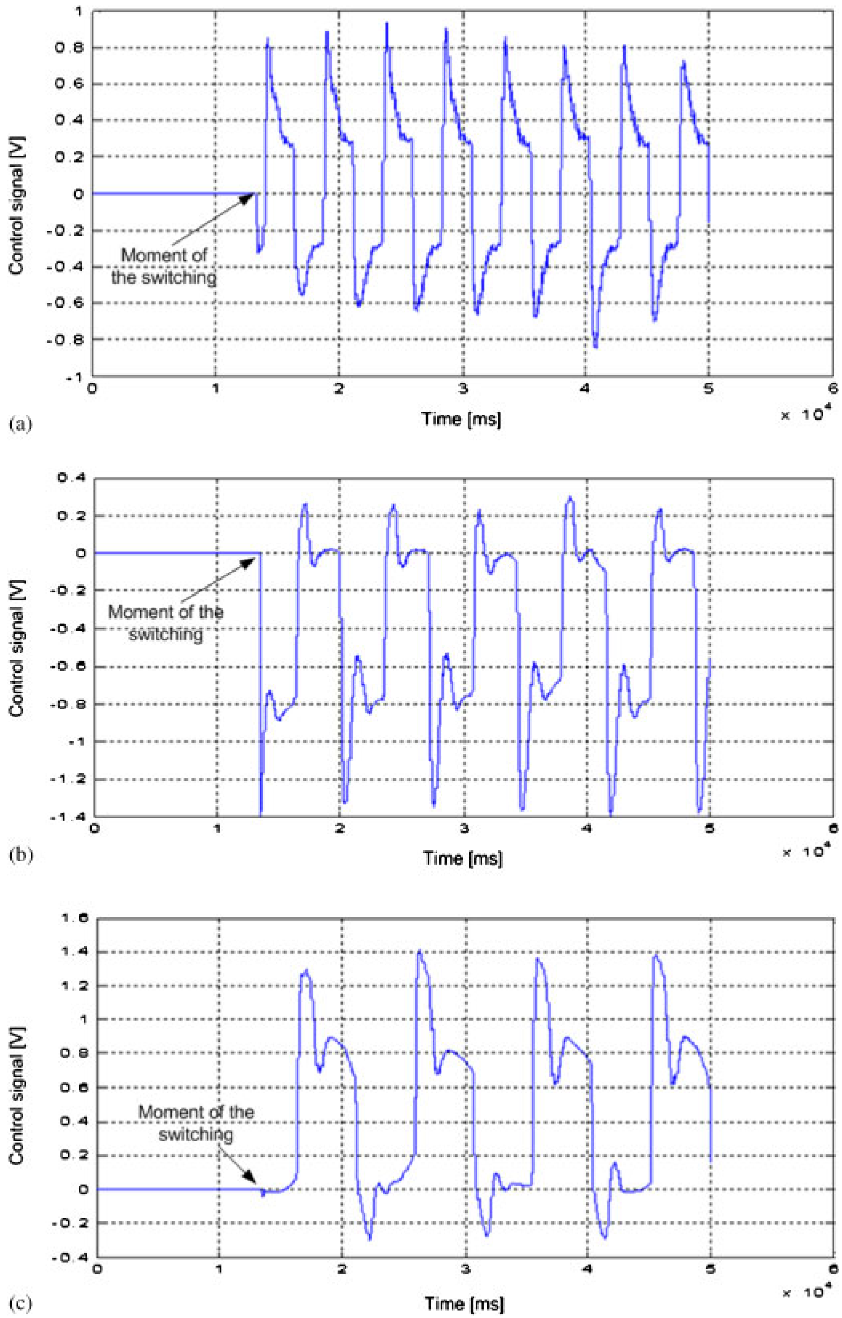

Figure 8. Changes in the neurocontroller output signal (torque) for the (a) joint 1, (b) joint 2 and (c) joint 3 when switching to sliding mode neuro-adaptive control.

DOI: 10.1002/acs

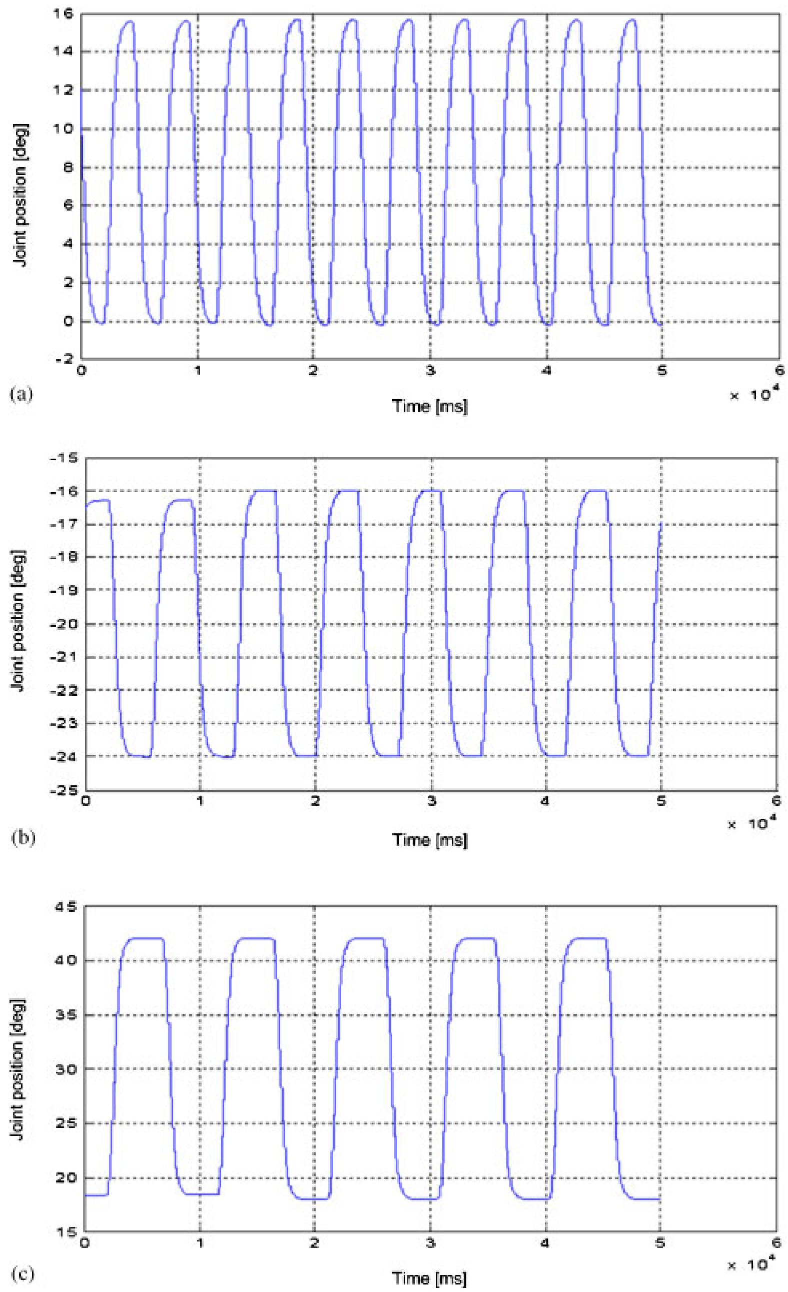

Figure 9. The tracked trajectory (in degrees) of (a) joint 1, (b) joint 2 and (c) joint 3.

DOI: 10.1002/acs

4. CONCLUSIONS

A novel approach for generating and maintaining sliding motion in the behaviour of a system with uncertainties in its dynamics is introduced. The system under control is under a closed loop simultaneously with a conventional PD controller and an adaptive variable structure neural controller. The experimental results obtained from a real-time trajectory tracking control of the first three joints of an articulated five degrees-of-freedom robot manipulator demonstrate that the proposed NN-based controller possesses a remarkable learning capability to control complex dynamical systems, virtually without requiring *a priori* knowledge of the plant dynamics and laborious start-up procedures. Another prominent feature that should be emphasized is the computational simplicity of the proposed approach.

ACKNOWLEDGEMENTS

This work was supported in part by Bogazici University Research Fund Project 03A202, TUBITAK Project 100E042, Ministry of Education and Science of Bulgaria Research Fund Project BU-TH-108/05 and TU Sofia, campus Plovdiv Research Fund Project 703/06.

REFERENCES

1. Lewis FL, Ge SS. *Neural Networks in Feedback Control Systems*. Mechanical Engineer's Handbook. Wiley: New York, 2005.
2. Rumelhart D, Hinton D, Williams G. Learning internal representations by error propagation. In *Parallel Distributed Processing*, Rumelhart D, McClelland F (eds), vol. 1. MIT Press, Cambridge, MA, 1986.
3. Yu XH, Efe MO, Kaynak O. A general backpropagation algorithm for feedforward neural networks learning. *IEEE Transactions on Neural Networks* 2002; **13**(1):251–254.
4. Efe MO, Kaynak O. Stabilizing and robustifying the learning mechanisms of artificial neural networks in control engineering applications. *International Journal of Intelligent Systems* 2000; **15**(5):365–388.
5. Ge SS, Hang CC, Lee TH, Zhang T. *Stable Adaptive Neural Network Control*. Kluwer: Boston, 2001.
6. Kosmatopoulos EB, Polycarpou MM, Christodoulou MA, Ioannou PA. High-order neural network structures for identification of dynamical systems. *IEEE Transactions on Neural Networks* 1995; **6**(2):431–442.
7. Suykens JAK, Vandewalle J, De Moor B. Lur'e systems with multilayer perceptron and recurrent neural networks; absolute stability and dissipativity. *IEEE Transactions on Automatic Control* 1999; **44**:770–774.
8. Yu W, Poznyak AS, Li X. Multilayer dynamic neural networks for nonlinear system on-line identification. *International Journal of Control* 2001; **74**(18):1858–1864.
9. Sira-Ramirez H, Colina-Morles E. A sliding mode strategy for adaptive learning in Adalines. *IEEE Transactions on Circuits and Systems I*: *Fundamental Theory and Applications* 1995; **42**(12):1001–1012.
10. Yu X, Zhihong M, Rahman SMM. Adaptive sliding mode approach for learning in a feedforward neural network. *Neural Computing and Applications* 1998; **7**:289–294.
11. Efe MO, Kaynak O, Yu X. Sliding mode control of three degrees of freedom anthropoid robot by driving the controller parameters to an equivalent regime. *Journal of Dynamic Systems*, *Measurement and Control* (ASME) 2000; **122**:632–640.
12. Parma GG, Menezes BR, Braga AP. Sliding mode algorithm for training multilayer artificial neural networks. *Electronics Letters* 1998; **34**(1):97–98.
13. Topalov AV, Kaynak O. On-line learning in adaptive neurocontrol schemes with a sliding mode algorithm. *IEEE Transactions on Systems*, *Man and Cybernetics*, *Part B* 2001; **31**(3):445–450.
14. Topalov AV, Kaynak O. Neural network modeling and control of cement mills using a variable structure systems theory based on-line learning mechanism. *Journal of Process Control* 2004; **14**:581–589.
15. Shakev NG, Topalov AV, Kaynak O. Sliding mode algorithm for on-line learning in analog multilayer feedforward neural networks. In *Artificial Neural Networks and Neural Information Processing*, Kaynak *et al.* (eds), Lecture Notes in Computer Science. Springer: Berlin, Heidelberg, New York, 2003; 1064–1072.

16. Utkin VI. *Sliding Modes in Control and Optimization*. Springer: Berlin, Heidelberg, New York, 1992.
17. Huang SJ, Lee JS. A stable self-organizing fuzzy controller for robotic motion control. *IEEE Transactions on Industrial Electronics* 2000; **47**(2):421–428.
18. Kim YH, Lewis FL. Optimal design of CMAC neural-network controller for robot manipulators. *IEEE Transactions on Systems*, *Man and Cybernetics*, *Part C* 2000; **30**(1):22–31.
19. Lewis FL, Yesildirek A, Liu K. Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks* 1996; **7**(2):388–399.
20. Yoo BK, Ham WC. Adaptive control of robot manipulator using fuzzy compensator. *IEEE Transactions on Fuzzy Systems* 2000; **8**(2):186–199.
21. Wai R-J. Tracking control based on neural network strategy for robot manipulator. *Neurocomputing* 2003; **51**:425–445.
22. Kawato M, Uno Y, Isobe M, Suzuki RA. Hierarchical model for voluntary movement and with application to robotics. *IEEE Control Systems Magazine* 1988; **8**(2):8–16.
23. Gomi H, Kawato M. Neural network control for a closed-loop system using feedback-error-learning. *Neural Networks* 1993; **6**:933–946.

DOI: 10.1002/acs