Adaptive Neural Network Sliding Mode Control For Electrically-Driven Robot Manipulators

S. Sefriti*, J. Boumhidi, R. Naoual*, I. Boumhidi***

**LESSI Laboratory, Department Of Physics, Faculty Of Sciences,*
Sidi Mohammed Ben Abdellah University, Fez, Morocco
e-mail: sefriti.salma@ hotmail.com
***L IIAN Laboratory, Department of Informatics, Faculty of Sciences,*
Sidi Mohammed Ben Abdellah University, Fez, Morocco
e-mail: jboumhidi@yahoo.fr

Abstract: In this paper a method for neural network sliding mode control design (NNS) is proposed for the robust tracking control of the electrically-driven two-links robot manipulators. The aim of this study is to overcome some shortcomings of the standard sliding mode controller (SMC) such as the produced higher amplitude of chattering, due to the higher switching gain required in the presence of large uncertainties. In the proposed NNS, the sliding mode control with a boundary layer approach is combined with the neural network (NN) to control the electrically-driven two-links robot. The NN is used for the prediction of the model unknown parts and hence it enables a lower switching gain to be used in the presence of large uncertainties. The stability is shown by the Lyapunov Theory and the control action used did not exhibit any chattering behavior. As a result, a high-precision position tracking performance is obtained without any oscillatory behavior. The effectiveness of the designed NNS is illustrated by simulations.

Keywords: Adaptive Control, Neural Network, Robot Manipulators, Sliding Mode Control.

1. INTRODUCTION

The robot manipulator is a complex nonlinear system, whose dynamic parameters are difficult to forecast precisely. In fact, it is almost impossible to obtain exact dynamic models as the system is described by a nominal model with large uncertainties. To deal with parameters uncertainties, various methods have been proposed, including the Sliding Mode Control (Slotine, 1984;Utkin, 1992), and the neural network based controls (Patino et al, 2002; Hussain and Ho, 2004; Liu et al, 2003). The SMC is a nonlinear control strategy that is well known for its strong robustness and accuracy. The main feature of this method is to drive the system states on a user-specified surface in the state space (switching surface), and to maintain the states on the surface for all subsequent time. However, in the presence of large uncertainties, the controller has a higher switching gain and produces higher amplitude of chattering. As a result, it is impossible to achieve in practical systems. One possible method to eliminate this chattering problem is based on the boundary layer solution (Slotine and Sastry, 1983; Slotine, 1984). Though, this method can resolve the problem for systems with small uncertainties only.

The NN-based controls (Ciliz, 2005; Sun et al, 2011; Sun et al, 2000) have been closely scoped out in the NN applications in robot tracking control. Most of the papers get the results that the tracking errors can be uniformly ultimately bounded as in (Sun et al, 2000) or asymptotically converge to zero as in (Ciliz, 2005; Sun et al, 2011). However, the considered uncertainties are small or some gains parameters are sufficiently large in the case of large uncertainties, which lead to the oscillatory behaviour. In this paper, a neural network structure is proposed to estimate the unknown parts of the two-links robot model, so that the system uncertainties can be kept small and hence enable a lower switching gain to be used. The network weights are adjusted during the online implementation by using the gradient descent method (GD) (Rumelhart et al, 1986). The proposed control consists of the so-called equivalent control added to robust control term, the NN predicted terms are incorporated in the equivalent control component, enabling the robust component to be used with a small gain which is responsible of compensating only the network errors prediction. As a result, the responses will be fast and smooth without any oscillatory behaviour. The stability is shown by using the Lyapunov theory.

The rest of the paper is divided into five sections. In Section 2, the system model is presented. In Section 3, the proposed neural network sliding mode controller is shown. Section 4 presents the simulations results. Finally, a conclusion is given in section 5.

2. SYSTEM MODEL OF THE ELECTRICALLY-DRIVEN TWO-LINKS ROBOT

The dynamic model of the electrically-driven two-links robot control may be expressed as follows (Dawson et al, 1992):

$$\begin{cases} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \\ \dot{\tau} = Ji - B\tau - E\dot{q} \end{cases} \quad (1)$$

where $q, \dot{q}, \ddot{q}$ denote respectively the joints position, which is the controlled output of the system, velocity, and acceleration vectors.

τ is the torque and the considered control law is the current i applied to the servo motors.

$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$: is an inertia matrix that is symmetric and positive definite where:

$$M_{11} = I_1 + I_2 + 4m_2 l_1^2 + 4m_2 l_1 l_2 \cos(q_2)$$
$$M_{12} = I_2 + 2m_2 l_1 l_2 \cos(q_2),$$
$$M_{21} = I_2 + 2m_2 l_1 l_2 \cos(q_2), \text{ and } M_{22} = I_2.$$

with $q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T$ as the positions, l_1, l_2 as the lengths, m_1, m_2 are the masses and I_1, I_2 as the respective inertias of the first and second segment.

$C(q,\dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$: represents the centrifugal forces where:

$$C_{11} = -2m_2 l_1 l_2 \dot{q}_2 \sin(q_2)$$
$$C_{12} = -2m_2 l_1 l_2 (\dot{q}_1 + \dot{q}_2)\sin(q_2)$$
$$C_{21} = 2m_2 l_1 l_2 \dot{q}_1 \sin(q_2) \text{ and } C_{22} = 0$$

$G(q)$ is the coriolis matrix given as:

$$G(q) = \begin{bmatrix} m_2 g l_2 \sin(q_1 + q_2) + m_1 g l_1 \sin(q_1) \\ m_2 g l_2 \sin(q_1 + q_2) \end{bmatrix}$$

J, B and E are constant, positive definite and diagonal matrices.

The system model can be written as the following state-space form:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = h_{1n}(\underline{x}, \underline{u}) + \xi_1(\underline{x}, t) \\ \dot{x}_4 = x_5 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = h_{2n}(\underline{x}, \underline{u}) + \xi_2(\underline{x}, t) \end{cases} \quad (2)$$

where: $h_{1n}(\underline{x}, \underline{u})$ and $h_{2n}(\underline{x}, \underline{u})$ are the nominal representation of the system with respectively the unknown parts $\xi_1(\underline{x}, t)$ and $\xi_2(\underline{x}, t)$.

$$h_{1n}(\underline{x}, \underline{u},) = f_{1n}(\underline{x}) + g_{11n}(x_1, x_4)u_1 + g_{12n}(x_1, x_4)u_2$$
$$h_{2n}(\underline{x}, \underline{u},) = f_{2n}(\underline{x}) + g_{21n}(x_1, x_4)u_1 + g_{22n}(x_1, x_4)u_2$$

$i = \underline{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$ is the input control of the system

$q = \begin{bmatrix} x_1 & x_4 \end{bmatrix}^T$: is the controlled output position

$\dot{q} = \begin{bmatrix} x_2 & x_5 \end{bmatrix}^T$, $\ddot{q} = \begin{bmatrix} x_3 & x_6 \end{bmatrix}^T$,

$\underline{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^T$

$$f_n(\underline{x}) = \begin{bmatrix} f_{1n}(\underline{x}) & f_{2n}(\underline{x}) \end{bmatrix}^T = -M^{-1}\{[\dot{M} + C + BM]\ddot{q} + [\dot{C} + BC + E]\dot{q} + [\dot{G} + BG]\}$$

and $M^{-1}(\underline{x})J = g_n(\underline{x}) = \begin{pmatrix} g_{11n}(\underline{x}) & g_{12n}(\underline{x}) \\ g_{21n}(\underline{x}) & g_{22n}(\underline{x}) \end{pmatrix}$.

3. NEURAL NETWORK SLIDING MODES CONTROL DESIGN

3.1 Controller Design

Let's define some variables as:

$$\xi(\underline{x}, t) = \begin{bmatrix} \xi_1(\underline{x}, t) & \xi_2(\underline{x}, t) \end{bmatrix}^T \quad (3)$$

This represents the unknown parts of the system.

$$e = q - q_d = \begin{bmatrix} x_1 - x_{1d} & x_4 - x_{4d} \end{bmatrix}^T \quad (4)$$

is the output tracking error with:

$q_d = \begin{bmatrix} x_{1d} & x_{4d} \end{bmatrix}^T$: is the desired output.

The control problem is to find a current control law so that the state $q(t)$ can track the desired trajectory q_d.

The relative degree $r = 3$, then the sliding variable can be defined as:

$$S = \ddot{e} + \gamma\dot{e} + \beta e \quad (5)$$

γ, β are diagonal matrices defined as follows:

$$\gamma = \begin{pmatrix} \gamma_{11} & 0 \\ 0 & \gamma_{22} \end{pmatrix}, \ \beta = \begin{pmatrix} \beta_{11} & 0 \\ 0 & \beta_{22} \end{pmatrix}$$

γ and β are selected such that the roots of the following characteristic polynomial are specified in the open left half of the complex plane :

$$\begin{aligned} & s^{(3)} + \gamma_{11} s^{(2)} + \beta_{11} s^{(1)} = 0 \\ & s^{(3)} + \gamma_{22} s^{(2)} + \beta_{22} s^{(1)} = 0 \end{aligned} \tag{6}$$

The sliding variable derivative is:

$$\dot{S} = f + \xi(\underline{x}, t) + gu + \begin{pmatrix} \dot{x}_{3d} \\ \dot{x}_{6d} \end{pmatrix} + \gamma\ddot{e} + \beta\dot{e} \tag{7}$$

To ensure that a sliding mode exists on a switching surface and that this switching surface can be reached in finite time, the condition given below has to be satisfied:

$$S^T \dot{S} < 0 \tag{8}$$

The control law that satisfies (8) is given by (Alaoui et al, 2007):

$$\underline{u} = g_n^{-1}(\underline{x}) \left(-f_n(\underline{x}) + \begin{pmatrix} \dot{x}_{3d} \\ \dot{x}_{6d} \end{pmatrix} - \gamma\ddot{e} - \beta\dot{e} - k\,sign(S) \right) \tag{9}$$

where sign(.) is the sign function given by:

$$sign(S) = \begin{cases} 1 & \text{if } S > 0 \\ 0 & \text{if } S = 0 \\ -1 & \text{if } S < 0 \end{cases}$$

The positive switching gain to compensate the uncertainties is : k which is designed as:

$$B < k \tag{10}$$

with B as the upper bound of the uncertainties given by:

$$\left\| \xi(\underline{x}, t) \right\| < B \tag{11}$$

To eliminate the chattering effect caused by the discontinuous control law, the boundary layer approach can be used. The control becomes as follows:

$$\underline{u} = g_n^{-1}(\underline{x}) \left(-f_n(\underline{x}) + \begin{pmatrix} \dot{x}_{3d} \\ \dot{x}_{6d} \end{pmatrix} - \gamma\ddot{e} - \beta\dot{e} - k\,sat(S) \right) \tag{12}$$

Where sat is the saturation function, given by:

$$sat(S) = \begin{cases} S/\delta & \text{if } \|S\| < \delta \\ \text{sgn}(S) & \text{otherwise} \end{cases} \tag{13}$$

with δ is the boundary layer thickness.

This method can resolve the problem for systems with small uncertainties. For systems with large uncertainties, we propose in this study the use neural networks to model the unknown parts of the two-links robot nonlinear functions given in (3), so that the system uncertainties can be kept small.

Let's denote the prediction of the unknown non linear functions parts as:

$$\hat{\xi}(\underline{x}, t) = \left[\hat{\xi}_1(\underline{x}, t) \quad \hat{\xi}_2(\underline{x}, t) \right]^T \tag{14}$$

where $\hat{\xi}_1(\underline{x})$, $\hat{\xi}_2(\underline{x})$ are the expressions for the network outputs given in a later section.

$$\varepsilon(\underline{x}, t) = \xi(\underline{x}, t) - \hat{\xi}(\underline{x}, t) \tag{15}$$

And $\left\| \xi(\underline{x}, t) \right\| < \varepsilon^*$ (16)

where ε^* is the upper bound of the network error prediction.

Theorem: Consider the robot manipulator modelled by (2) in the presence of large uncertainties. If the system control is designed as:

$$\underline{i} = \underline{u} = \hat{\underline{u}}_e + \underline{u}_s$$

$$\hat{\underline{u}}_e = g_n^{-1}(\underline{x}) \left(-(f_n(\underline{x}) + \hat{\xi}(\underline{x}, t)) + \begin{pmatrix} \dot{x}_{3d} \\ \dot{x}_{6d} \end{pmatrix} - \gamma\ddot{e} - \beta\dot{e} \right)$$

$$\underline{u}_s = -k\,g_n^{-1}(\underline{x})\,sat(S)$$

with $\varepsilon^* < k$

The trajectory tracking errors will converge, in finite time, to the vicinity of S = 0 as $\|S\| < \delta$, with δ is the small boundary layer thickness.

Proof. Consider the candidate Lyapunov function: $V = \frac{1}{2} S^T S$ then $\dot{V} = S^T \dot{S}$

Replacing the expression of $\dot{S}$ given in (7) we get:

$$\dot{V} = S^T \left(f + \xi(\underline{x}, t) + g\underline{u} + \begin{pmatrix} \dot{x}_{3d} \\ \dot{x}_{6d} \end{pmatrix} + \gamma\ddot{e} + \beta\dot{e} \right)$$

By replacing the expression of u given in the theorem we get:

$$\begin{aligned} \dot{V} &= S^T (\xi(\underline{x}, t) - \hat{\xi}(\underline{x}, t) - ksat(S)) \\ &= S^T \varepsilon(\underline{x}, t) - kS^T sat(S) \le \left\| S^T \right\| \left\| \varepsilon(\underline{x}, t) \right\| - kS^T sat(S) \end{aligned}$$

$$\leq \left\| S^T \right\| \varepsilon^* - kS^T sat(S)$$

By choosing $\varepsilon^* < k$, with k as a small gain, which is responsible only for compensating the network errors prediction, we get:

For any small $\delta > 0$, if $\|S\| \geq \delta$, $sat(S) = sign(S)$, the function $\dot{V} = (\varepsilon^* - k)\|S\| < 0$. However, in a small δ-vicinty of the origin (boundary layer), $sat(S) = \frac{S}{\delta}$ is continuous, the system trajectories are confined to a boundary layer of a sliding mode manifold $S = 0$, then the high tracking precision $\|S\| < \delta$ is obtained .

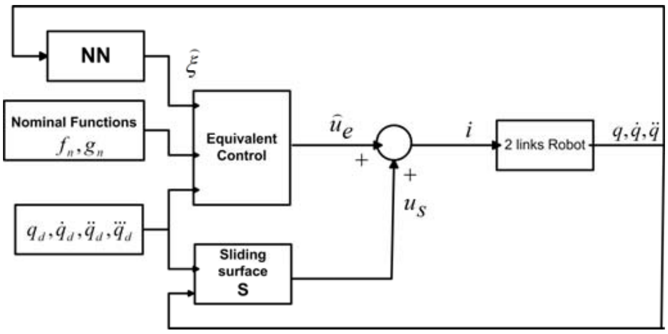

Fig. 1. NNS controller scheme.

3.2 Neural Network Design

In this paper, we consider a NN with two layers of adjustable weights (Lewis et al, 1999) (Fig. 1). $\underline{x}$: is the state input variables and the output variables are:

$y_1 = \hat{\xi}_1(\underline{x}, t)$, $y_2 = \hat{\xi}_2(\underline{x}, t)$

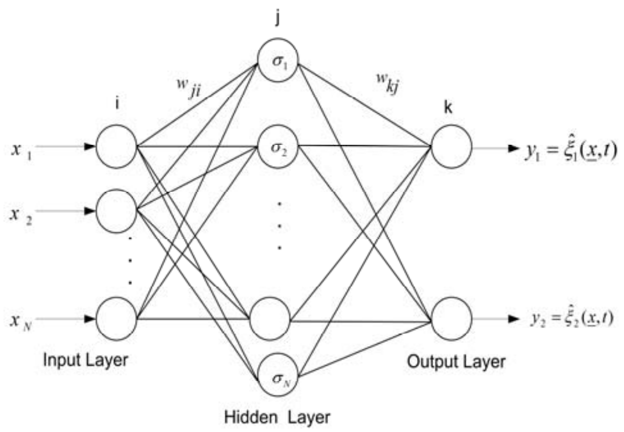

Fig. 2. The architecture of a multilayer neural network for the prediction of uncertain parts.

$$y_k(\underline{x}) = W_k^T \sigma(W_j^T \underline{x}) \quad k = 1,2 \tag{17}$$

where:

$\sigma(.)$ represents the hidden-layer activation function considered as a sigmoid function given by:

$$\sigma(s) = \frac{1}{1+e^{-s}} \tag{18}$$

$W_k = \begin{bmatrix} w_{k1} & w_{k2} & \ldots & w_{kN} \end{bmatrix}^T$ and

$W_j = \begin{bmatrix} w_{j1} & w_{j2} & \ldots & w_{jN} \end{bmatrix}^T$ are respectively the interconnection weights between the hidden and the output layers and between the input and the hidden layers.

The actual output $y_{dk}(\underline{x})$ (desired output which is the difference between the actual and nominal functions) is:

$$y_{dk}(\underline{x}) = y_k(\underline{x}) + \varepsilon_k(\underline{x}) \tag{19}$$

Where: $\varepsilon_k(\underline{x})$ is the NN approximation error.

Remark: Before incorporating the networks into the proposed sliding mode control strategy, the networks were trained offline. The objective of offline training is to let the networks learn the functional nonlinearities to a certain degree of accuracy before implementing into the controller, and thus can give faster online adaptation as needed. After the pre-training step, we would have reasonably good initial values of the network weights.

The network weights are adjusted during the online implementation. The method used is based on the gradient descent method (GD), which is a simple and fast method for online adaptation.

The essence of the GD consists of iteratively adjusting the weights in the direction opposite to the gradient of E, so as to reduce the discrepancy according to:

$$\frac{\partial w_{kj}}{\partial t} = -\eta_k \frac{\partial E}{\partial w_{kj}} \tag{20}$$

Where $\eta_k > 0$ is the usual learning rate. The gradient terms $\frac{\partial E}{\partial w_{kj}}$ can be derived using the backpropagation algorithm (Rumelhart et al, 1986). The cost function E is defined as the error index and the least square error criterion is often chosen as follows:

$$E = \frac{1}{2}\sum_{k=1}^{2} \varepsilon_k^2 \tag{21}$$

4. SIMULATION RESULTS

In this section, we test the proposed control algorithm on a two-links robot described by the model (2). The control objective is to maintain the system in order to track the desired angle trajectory:

$x_{1d} = (\pi/3)\cos(t)$ and $x_{4d} = \pi/2 + (\pi/3)\sin(t)$

The parameters are considered to be $m_1 = 0.6$ and $m_2 = 0.4$.

The considered sampling period is 0.01s.

The considered uncertainties are a vector random noise with the magnitude equal to unity.

$$E = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix},\; B = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \text{ and } J = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$$

The switching functions coefficients are defined as:
$\gamma_{11} = \gamma_{22} = \beta_{11} = \beta_{22} = 4$.

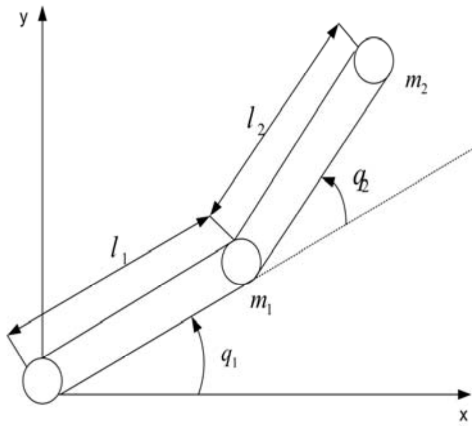

Fig. 3. Two-links robot manipulator.

From figures 4 and 7, it can be seen that the tracking performance is obtained without any oscillatory behaviour even in the presence of large uncertainties. The corresponding control current signals are given in Fig. 5 and 8. The figures 6 and 9 show the adjusted torques of joints 1 and 2.

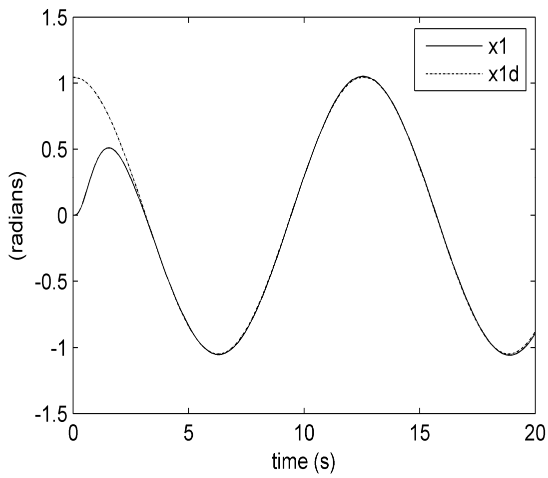

Fig. 4. Angle response x_1 and desired trajectory x_{1d}.

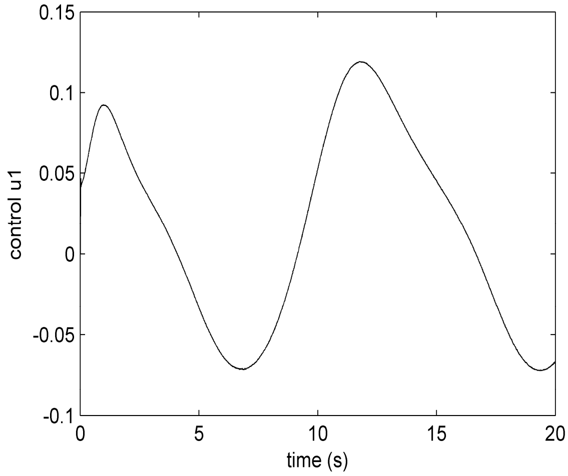

Fig. 5. Control u_1 (input current of join actuator 1).

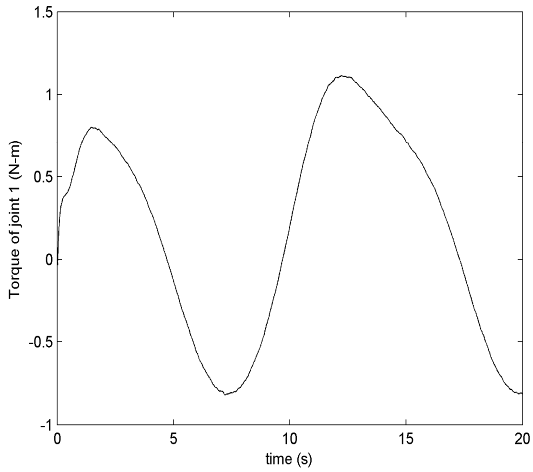

Fig. 6. Torque of joint 1 (N-m).

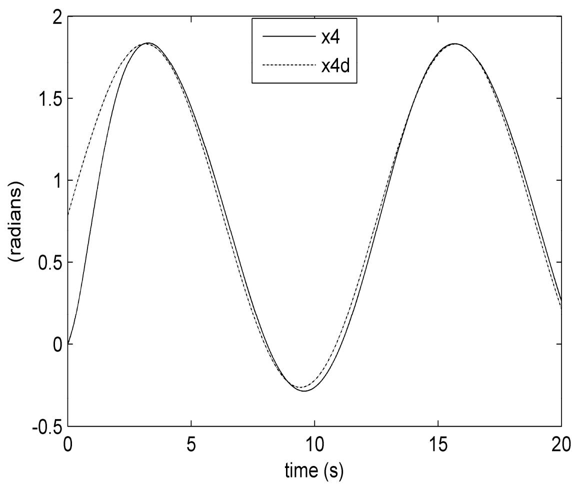

Fig. 7. Angle response x_4 and desired trajectory x_{4d}.

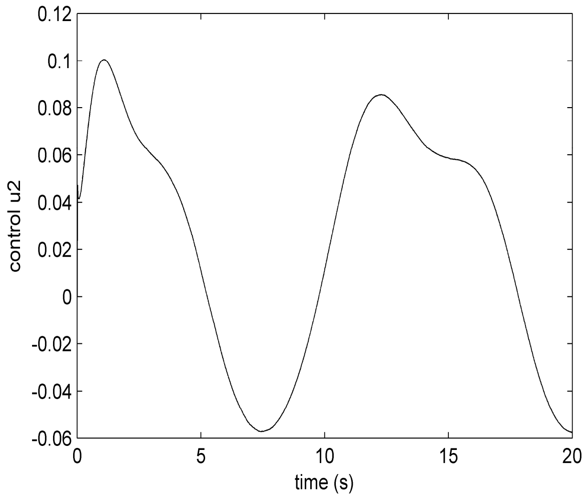

Fig. 8. Control u_2 (input current of join actuator 2).

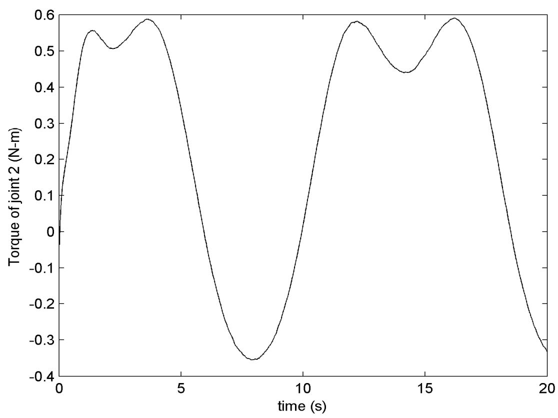

Fig. 9. Torque of joint 2 (N-m).

5. CONCLUSIONS

This paper addressed the robust trajectory tracking problem for a robot manipulator in the presence of large uncertainties without any chattering behaviour. The designed method is a combination of the sliding mode control with a boundary layer approach and the neural network employed to approximate the nonlinear model functions unknown parts with online adaptation of parameters. Simulation results have shown a good performance of the proposed method to track the desired trajectory without any oscillatory behaviour.

REFERENCES

Alaoui, C., Boumhidi, J., and Mrabti, M. (2007). Sliding surface controller design based on the LMIs with application to a two- links robot, *IJ-STA, Special Issue, CSC*, 300-311.

Ciliz, M.K. (2005). Adaptive control of robot manipulators with neural network based compensation of frictional uncertainties, *Robotica*, (23), 159–167.

Dawson, D.M., Qu, Z., and Carroll, J.J. (1992) Tracking control of rigid-link electrically-driven robot manipulators, *International Journal of Control* (56), 991–1006.

Hussain, M.A., and Ho, P.Y. (2004) Adaptive sliding mode control with neural network based hybrid models, *Journal of Process Control*, (14), 157–176.

Kim, Y., Seok., J., Noh., I and Won, S. (2008) An adaptive disturbance observer for a two-links robot manipulator, *International Conference on Control Automation and Systems,* 141–145.

Lewis, V.F.L., Jagannathan, S., and Yesildirek, A. (1999) *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor and Francis, London.

Liu, P.X., Zuo, M.J., and Meng, M.Q.H. (2003). Using neural network function approximation for optimal design of continuous-state parallel –series systems, Computers and operations Research, (30), 339-352.

Patino, H.D., Carelli, R., and Kuchen, B.R. (2002). Neural networks for advanced control of robot manipulators, *IEEE Transactions on Neural Networks,* (13), 343–354.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning internal representations by error propagation, *D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing*,MIT Press, Cambridge, MA, (1), 318–362.

Slotine, J.J., and Sastry, S.S. (1983) Tracking control of nonlinear systems using sliding surfaces with applications to robot manipulators, *International Journal of Control*, (39), 465–492.

Slotine, J.J. (1984) Sliding controller design for non-linear systems, *International Journal of Control,* (40), 421–434.

Sun, F.C., Sun, Z.Q., Zhang, R.J., and Chen, Y.B. (2000) Neural adaptive tracking controller for robot manipulators with unknown dynamics, *IEE Proceedings control Theory and Application, (*147), 366-370.

Sun, T., Pei, H., Pan, Y., Zhou, H., and Zhang, C. (2011) Neural network-based sliding mode adaptive control for robot manipulators, *Neurocomputing, (*74), 2377–2384.

Utkin, V.I. (1992) *Sliding modes in control optimization, Springer-Verlag*.