Robust Neural Sliding Mode Control of Robot Manipulators

Nguyen Tran Hiep*. Pham Thuong cat.**

** Le Quy Don University Hanoi – Viet nam*
e-mail: hiepnguyentran@vnn.vn
***Institute of Information Technology, Academy of Science and Technology Vietnam*
e-mail: ptcat@ioit.ac.vn

Abstract: This paper proposes a robust neural sliding mode control method for robot tracking problem to overcome the noises and large uncertainties in robot dynamics. The Lyapunov direct method has been used to prove the stability of the overall system. Simulation results are given to illustrate the applicability of the proposed method

Keywords: Robot control, neural network, PD sliding mode

1. INTRODUCTION

Robot control is a domain that has been developed continuously since the last 50 years [1] [13] [14]. The dynamics of robot manipulator is a highly nonlinear system with several uncertain factors. In many applications the robot must move quickly from a position to the other or track accurately a desired trajectory in a 3D space. Traditional control methods like PID, computed torque or optimal control [3] [13] do not always bring reasonable results because these methods often require knowing accurately the robot dynamics. Sliding mode control [12] is a robust control method. However the main disadvantage of sliding mode control is the self chattering around the sliding surface in both amplitude and frequency. In order to reduce those bad effects many authors has proposed different methods as bringing the integral part into the sliding surface, using saturated functions, or estimation of uncertain parts of robot dynamics [5][7][8]. Neural networks have been used widely in control because of advantages as parallel processing, highly capable of self learning and self adaptation. Neural networks can approximate nonlinear functions from practice data without knowing exactly parameters and structure of the functions [4] [6]. Many researches used this property of neural networks to approximate unknown nonlinearities of robot and presented the convergence through simulation results [10] [11]. Anyway most of them did not show the close theory to prove the stability of the whole robot control system using neural networks.

This paper proposes a new trajectory tracking robot control method with neural networks to approximate uncertainties and compensate noises. The stability of the whole system is proved by the Lyapunov direct method. The paper is divided into 6 parts. The 1st part deals with the problems of robot control with many uncertainties. The 2nd part describes the problem to solve. The 3rd part describes the proposed control scheme consisting of a PD sliding mode control block, a nonlinear feed forward block and a neural network block with weights learning online. The system stability is also proved in this part. The 4th part shows some evaluations on the accuracy and the convergence rate of the method. Computer simulation results are demonstrated in the 5th part and finally some summary and conclusions are given in the last part.

2. PROBLEMS OF ROBOT CONTROL WITH MANY UNCERTAINTIES

The dynamic equation of a robot system with n joints can be described by the following nonlinear MIMO equation:

$$\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) + \mathbf{d}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau} \qquad (1)$$

where $\mathbf{q} = [q_1, q_2, ...q_n]^T$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ are vectors *nx1* representing the position, velocity and acceleration of related joints, $\boldsymbol{\tau} = [\tau_1, \tau_2, ...\tau_n]^\mathbf{T}$ is a vector *nx1* representing the torques acting on the joints, $\hat{\mathbf{M}}(\mathbf{q})$ is the n*x*n inertia matrix, $\hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}})$ is the nxn matrix of Coriolis and centripetal effects, $\hat{\mathbf{g}}(\mathbf{q})$ is the *nx1* torque vector representing the gravity effect, $\mathbf{d}(\mathbf{q},\dot{\mathbf{q}})$ is the *nx1* vector of friction and noises acting on the joints.

The parameters $\hat{\mathbf{M}}(\mathbf{q})$, $\hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}})$, $\hat{\mathbf{g}}(\mathbf{q})$ often can not be identified exactly. We can describe them as follow:

$$\hat{\mathbf{M}}(\mathbf{q}) = \mathbf{M}(\mathbf{q}) + \Delta\mathbf{M}(\mathbf{q}) \qquad (2a)$$

$$\hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{B}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\mathbf{B}(\mathbf{q},\dot{\mathbf{q}}) \qquad (2b)$$

$$\hat{\mathbf{g}}(\mathbf{q}) = \mathbf{g}(\mathbf{q}) + \Delta\mathbf{g}(\mathbf{q}) \qquad (2c)$$

Where $\mathbf{M}(\mathbf{q})$, $\mathbf{B}(\mathbf{q},\dot{\mathbf{q}})$, $\mathbf{g}(\mathbf{q})$ are estimated values, $\Delta\mathbf{M}(\mathbf{q})$, $\Delta\mathbf{B}(\mathbf{q},\dot{\mathbf{q}})$, $\Delta\mathbf{g}(\mathbf{q})$ are unknown. However we can assume $\|\Delta\mathbf{M}(\mathbf{q})\| < m_0, \|\Delta\mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\| < b_0, \|\Delta\mathbf{g}(\mathbf{q})\| < g_0$, where m_0, b_0, g_0 are known. Equation (1) can be rewritten as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau} \qquad (3a)$$

CP1107, *Intelligent Systems and Automation, 2nd Mediterranean Conference*
edited by L. Beji, S. Otmane, and A. Abichou

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) \quad (3b)$$

$$\boldsymbol{\tau}_0 = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (3c)$$

$$\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) = \Delta\mathbf{M}(\mathbf{q}) + \Delta\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q}) + \mathbf{d}(\mathbf{q}, \dot{\mathbf{q}}) \quad (3d)$$

Where $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) \in R^n$ is the sum of uncertainties of the robot dynamics, friction and noises that are acting on the robot. It is limited by $\|\mathbf{f}\| \le f_0$ with f_0 is estimable. We use following properties of robot dynamics in finding control algorithm in the paper:

Inertia matrix $\mathbf{M}(\mathbf{q})$ is symmetric positive definite.

Matrix $(\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}))$ is skewing symmetric, that is:

$$\mathbf{x}^{\mathbf{T}}\dot{\mathbf{M}}(\mathbf{q})\mathbf{x} = 2\mathbf{x}^{\mathbf{T}}\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{x} \quad (4)$$

with $\forall \mathbf{x} \in R^n$.

The goal of control is to choose torque τ so the robot follows the desired trajectory $\mathbf{q_d}$ that means the error $\mathbf{e} = (\mathbf{q} - \mathbf{q_d}) \to 0$; $\dot{\mathbf{e}} = (\dot{\mathbf{q}} - \dot{\mathbf{q}}_\mathbf{d}) \to 0$ and $\ddot{\mathbf{e}} = (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_\mathbf{d}) \to 0$. Here, $\mathbf{e}, \dot{\mathbf{e}}, \ddot{\mathbf{e}}$ are errors on position, velocity and acceleration.

3. NUEURAL SLIDING MODE CONTROL ALGORITHM

Sliding mode control is widely used in control of uncertain MIMO nonlinear systems. For the robot system (1), a sliding surface is chosen in PD form:

$$\mathbf{s}(t) = \dot{\mathbf{e}} + \mathbf{Ce} \quad (5)$$

C is often chosen as a positive diagonal matrix and $\mathbf{s} = [s_1, s_2, \ldots\ldots s_n]^T$. Equation (5) shows the close relation between $(\mathbf{e}, \dot{\mathbf{e}})$ and s. Therefore Equation (3b) could be written as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \mathbf{f}(\mathbf{s}) \quad (6)$$

The unknown f(s) is the main reason to reduce the control quality. If we can compensate this effect, the control quality may improve. According to Stone-Weierstrass theorem [2] we can choose an appropriate Artificial Neural Network (ANN) with limited number of nodes that can approximate an unknown nonlinear function with given accuracy. For approximating function $\mathbf{f(s)}$ we choose the following simple structure:

$$\mathbf{f(s)} = \mathbf{W}\boldsymbol{\sigma} + \varepsilon \quad (7a)$$

or

$$\mathbf{f(s)} = \hat{\mathbf{f}} + \varepsilon \quad (7b)$$

where the approximated part of f(s) is $\hat{\mathbf{f}} = [\hat{f}_1, \hat{f}_2, \ldots\ldots \hat{f}_n]^T = \mathbf{W}\boldsymbol{\sigma}$; $\boldsymbol{\varepsilon}$ is the approximation error. With $\|\mathbf{f(s)}\| < f_0$, we can have a limit ε_0 of $\boldsymbol{\varepsilon}$: $\|\boldsymbol{\varepsilon}\| \le \varepsilon_0$

Let $\mathbf{w}_i$ be the column vector i of matrix W, we will have:

$$\hat{\mathbf{f}} = \mathbf{W}\boldsymbol{\sigma} = [\mathbf{w}_1, \mathbf{w}_2, \ldots\ldots \mathbf{w}_\mathbf{n}]\boldsymbol{\sigma} = \sum_{i=1}^{n} \mathbf{w}_i \boldsymbol{\sigma}_i \quad (8)$$

We build a RBF (Radial Basic Function).neural network having one hidden layer as shown in Figure 1.

This structure has been proved to satisfy the Stone-Weierstrass Theorem [2]. Choosing the acting function σ_i with Gaussian distribution we have:

$$\sigma_i = \exp - \frac{(s_i - c_i)^2}{\lambda_i^2} \quad (9a)$$

Where c_i is the centre and λ_i is the deviation parameter, freely chosen. So we have

$$\hat{f}_\mathbf{i} = \sum_{j=1}^{n} \mathbf{w}_{ji} \boldsymbol{\sigma}_j \text{ , with } i = 1, 2 \ldots n \quad (9b)$$

$\mathbf{w}_{ij}$ is the weights of the approximating neural network.

The control problem is now to find the control torque $\boldsymbol{\tau}$ with learning algorithm w_{ij} of neural network (8) so s → 0 and the system will slide toward the co-ordinate origin e = 0 granting q(t) → q_d.

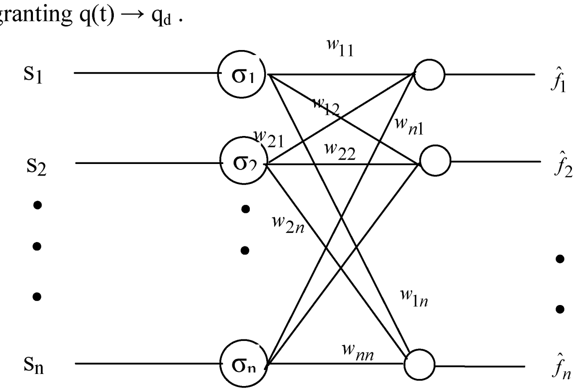

Figure 1: RBF Neural Network used to approximate uncertainties of the robot

Theorem: The dynamic system (1) of n DOF robot with given neural network (8) and sliding surface (5) will follow the desired trajectory q_d with error $\mathbf{e} = (\mathbf{q} - \mathbf{q_d}) \to 0$ if we apply the control torque $\boldsymbol{\tau}$ and the learning algorithm $\dot{\mathbf{w}}_i$ as follows:

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_\mathbf{d} + \mathbf{B}\dot{\mathbf{q}}_\mathbf{d} + \mathbf{g} - \mathbf{MC}\dot{\mathbf{e}} - \mathbf{BCe} - \mathbf{Ks} - \gamma \frac{\mathbf{s}}{\|\mathbf{s}\|} + (1 + \eta)\mathbf{W}\boldsymbol{\sigma} \quad (10)$$

$$\dot{\mathbf{w}}_i = -\eta \mathbf{s} \sigma_i \quad (11)$$

Where matrix $\mathbf{K} = \mathbf{K}^\mathbf{T} > 0$ is a freely chosen symmetric positive definite matrix, and $\eta, \gamma > 0$.

The structure of the proposed control system is showed in the diagram on Figure 2. Moment $\boldsymbol{\tau}$ comprises of three main parts: a nonlinear feed-forward compensator $\boldsymbol{\tau}_{ff} = \mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{B}\dot{\mathbf{q}}_d + \mathbf{g} - \mathbf{MC}\dot{\mathbf{e}} - \mathbf{BCe}$, sliding part $\boldsymbol{\tau}_s = -\mathbf{Ks} - \gamma \frac{\mathbf{s}}{\|\mathbf{s}\|}$ and is a RBF neural network with online learning $\hat{\mathbf{f}}$.

This theorem can be proved by the Lyapunov direct method granting the asymptotical stability as follow:

Choose Lyapunov candidate function as:

$$V = \frac{1}{2}\left[\mathbf{s}^{\mathbf{T}}\mathbf{Ms} + \sum_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{w}_{\mathbf{i}}^{\mathbf{T}}\mathbf{w}\right] \tag{12}$$

Because M is the inertia matrix of the robot that is symmetric positive definite so we have V > 0 for all $(\mathbf{s}^T, \mathbf{w}^T) \neq 0$ and V = 0 if and only if $(\mathbf{s}^T, \mathbf{w}^T) = 0$. Function V satisfies other conditions of the Lyapunov stability method as V → ∞ when s → ∞, $w_i \to \infty$. If we can identify the control torque $\boldsymbol{\tau}$ granting $\dot{V} < 0$ then according to the Lyapunov stability method s → 0 or the system will approach and stay on the sliding surface. The stability of the whole system will be held.

Take derivation of V in time and using (4) we have:

$$\begin{aligned} \dot{V} &= \mathbf{s}^{\mathbf{T}}\mathbf{M}\dot{\mathbf{s}} + \frac{1}{2}\mathbf{s}^{\mathbf{T}}\dot{\mathbf{M}}\mathbf{s} + \sum_{\mathbf{i}=1}^{n} \mathbf{w}_i^{\mathbf{T}}\dot{\mathbf{w}} = \\ &= \mathbf{s}^{\mathbf{T}}\mathbf{M}\dot{\mathbf{s}} + \mathbf{s}^{\mathbf{T}}\mathbf{Bs} + \sum_{i=1}^{n} \mathbf{w}_i^{\mathbf{T}}\dot{\mathbf{w}} \end{aligned} \tag{13}$$

Taking into account that:

$$\mathbf{M}\dot{\mathbf{s}} + \mathbf{Bs} = \mathbf{M}(-\ddot{\mathbf{q}}_d + \mathbf{C}\dot{\mathbf{e}}) + \mathbf{B}(-\dot{\mathbf{q}}_d + \mathbf{Ce}) + \mathbf{B}\dot{\mathbf{q}} + \mathbf{M}\ddot{\mathbf{q}} \tag{14}$$

and from Equations (1), (3a) we can conclude

$$\mathbf{B}\dot{\mathbf{q}} + \mathbf{M}\ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{g(q)} - \mathbf{f(s)} \tag{15}$$

Replace (14), (15), (7b) into (14) we have:

$$\dot{V} = \mathbf{s}^{\mathbf{T}}\begin{bmatrix} -\mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{MC}\dot{\mathbf{e}} - \mathbf{B}\dot{\mathbf{q}}_d + \\ +\mathbf{BCe} + \boldsymbol{\tau} - \mathbf{g} - \mathbf{W}\boldsymbol{\sigma} - \boldsymbol{\varepsilon} \end{bmatrix} + \sum_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{w}_i^{\mathbf{T}}\dot{\mathbf{w}}_i \tag{16}$$

Apply $\boldsymbol{\tau}$ chosen from (10) into equation (16) we get:

$$\dot{V} = \mathbf{s}^T\left[-\mathbf{Ks} - \gamma\frac{\mathbf{s}}{\|\mathbf{s}\|} + \eta\mathbf{W}\sigma - \boldsymbol{\varepsilon}\right] + \sum_{i=1}^{n} \mathbf{w}_i^T\dot{\mathbf{w}}_i \tag{17}$$

With learning algorithm (11) the last part of (17) could be rewritten in:

$$\sum_{i=1}^{n} \mathbf{w}_i^{\mathbf{T}}\dot{\mathbf{w}}_i = -\eta\sum_{i=1}^{n} \mathbf{w}_i^{\mathbf{T}}\mathbf{s}\sigma_i = -\eta\mathbf{s}^{\mathbf{T}}\sum_{i=1}^{n} \mathbf{w}_i\sigma_i = -\eta\mathbf{s}^{\mathbf{T}}\mathbf{W}\boldsymbol{\sigma} \tag{18}$$

Replace (18) into (17) finally we get:

$$\dot{V} = -\mathbf{s}^{\mathbf{T}}\mathbf{Ks} - \gamma\mathbf{s}^{\mathbf{T}}\frac{\mathbf{s}}{\|\mathbf{s}\|} - \mathbf{s}^{\mathbf{T}}\boldsymbol{\varepsilon} \tag{19}$$

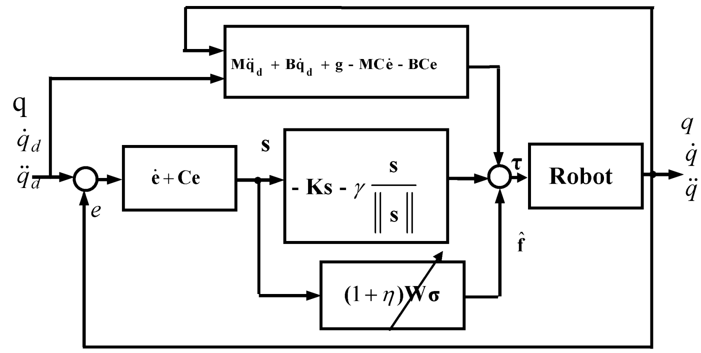

Figure 2: Structure of proposed robust neural sliding mode control for robot manipulators with many uncertainties

With differentiable **s** (5), let's examine $\frac{\mathrm{s}}{\|\mathrm{s}\|}$ in Equations (10) and (19).

In a n dimensional space, estimating $\lim_{\mathrm{s}\to 0}\frac{\mathrm{s}}{\|s\|}$, where $\mathrm{s} = [s_1, s_2, \ldots s_n]^{\mathrm{T}}$ and $\|s\| = \sqrt{s_1^2 + s_2^2 + \ldots + s_n^2}$, means that $\frac{\mathrm{s}}{\|s\|} = \left[\frac{s_1}{\|s\|}; \frac{s_2}{\|s\|}; \ldots; \frac{s_n}{\|s\|}\right]^{\mathrm{T}}$, so

$$\lim_{\mathrm{s}\to 0}\frac{\mathrm{s}}{\|s\|} = \left[\lim_{s_1\to 0}\frac{s_1}{\|s\|}; \lim_{s_2\to 0}\frac{s_2}{\|s\|}; \ldots; \lim_{s_n\to 0}\frac{s_n}{\|s\|}\right]^{\mathrm{T}} = 0$$

is finite. Therefore τ and $\dot{V}$ in Equations (10) and (19) are continuous when $\mathrm{s} \to 0$.

Chosen $\gamma = \delta + \varepsilon_0$ with δ > 0

$$\dot{V} = -\mathbf{s}^{\mathbf{T}}\mathbf{Ks} - \delta\|\mathbf{s}\| - (\varepsilon_0\|\mathbf{s}\| + \mathbf{s}^{\mathbf{T}}\boldsymbol{\varepsilon}) \leq \mathbf{0} \tag{20}$$

Because $\|\varepsilon\| \leq \varepsilon_0$ so that $\dot{V} < 0$ for all s ≠ 0 and $\dot{V} = 0$ if and only if s = 0. According to the Lyapunov direct stability theorem, we have $\mathbf{s} \to 0$, and from Equation (5) we get e → $0, \dot{\mathbf{e}} \to 0$, in other words robot asymptotically follows the desired trajectory with error e → 0. So the theorem as well as the stability of the overall sliding mode control system using neural network described in Figure 2 has been proved.

4. ABOUT THE ACCURACYAND CONVERGENCE

The $\mathbf{\Delta M(q), \Delta B(q, \dot{q}), \Delta g(q)}$ in robot dynamics, are unknown changing quantities, causing errors and reducing the convergence of the control algorithm. However we can estimate the varying range of them from robot dynamic parameters and their effect can be compensated by a properly chosen neural network. The neural network chosen here has the number of neurons in the hidden layer equal to the number of DOF of the robot. The hidden neurons use acting function with Gaussian distribution. The signals on the nodes of the output layer of the neural network are the linear sum of output functions of the neurons in the hidden layer. The accuracy of the neural approximation depends on chosen parameters c_i, λ_i of the Gaussian distribution functions. The neural network has to cover the whole varying range of the uncertainties. The convergence rate of the neural network depends on the on-line learning rule (11) with the participation of the sliding surface s and learning factor η. If we choose a large slope sliding surface (large C, K) and large factor η the convergence rate is faster. However, these may cause overshoot and reduce the control quality. Optimally chosen parameters c_i, λ_i, η will bring best approximation results and optimal convergence rate. Simulation method or Genetic Algorithms (GA) may be used to identify these optimal parameters.

The control algorithm (11) when using function $\mathbf{s}\|\mathbf{s}\|^{-1}$ will create a continuous control signal and therefore can eliminate chattering.

5. SIMULATIONS

To demonstrate the proposed control algorithm we simulated the motion of a robot following a line trajectory in the Cartesian space. The chosen robot has 2 DOF [13] described in Figure 3 with technical parameters from Table 1. The task of the robot is to move its hand following a line from point A to point B in the Cartesian plane (Oxy), in time T with trapezoidal speed. That means the speed increases from zero at point A to a section with constant speed and later decreases to zero at the point B.

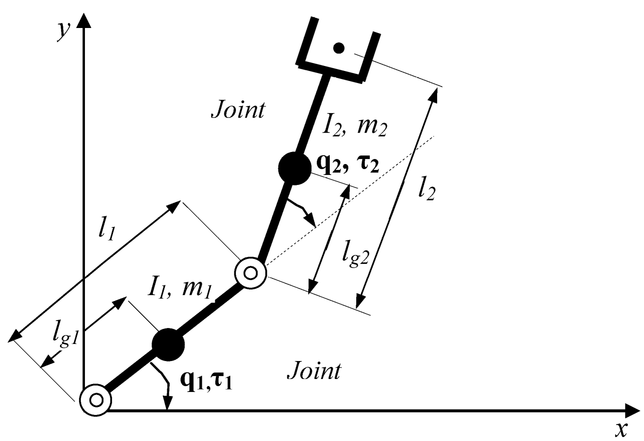

Figure 3: 2 DOF Robot

Let $m_i = m_{mi} + m_{li}$; the dynamic equation of this robot has the form $\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) + \mathbf{d}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau}$ with:

$$\hat{\mathbf{M}}(\mathbf{q}) = \begin{bmatrix} m_1 l_{g1}^2 + I_1 + I_2 + m_2(l_1^2 + l_{g2}^2 + 2 l_1 l_{g2}\cos q_2) & m_2(l_{g2}^2 + l_1 l_{g2}\cos q_2) + I_2 \\ m_2(l_{g2}^2 + l_1 l_{g2}\cos q_2) + I_2 & m_2 l_{g2}^2 + I_2 \end{bmatrix}$$

$$\hat{\mathbf{g}}(\mathbf{q}) = \begin{pmatrix} (m_1 l_{g1} + m_2 l_1) g\cos(q_1) + m_2 l_{g2} g\cos(q_1+q_2) \\ m_2 l_{g2} g\cos(q_1+q_2) \end{pmatrix}$$

$$\hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}}) = m_2 l_1 l_{g2}\begin{pmatrix} -2\dot{q}_2 & -\dot{q}_2 \\ \dot{q}_1 & 0 \end{pmatrix}\sin q_2$$

We choose control parameters for moment **τ** and correction algorithm for the weights in the neural network as follow:

$$\boldsymbol{\tau} = \hat{\mathbf{M}}\ddot{\mathbf{q}}_{\mathbf{d}} + \hat{\mathbf{B}}\dot{\mathbf{q}}_{\mathbf{d}} + \hat{\mathbf{g}} - \hat{\mathbf{M}}\mathbf{C}\dot{\mathbf{e}} - \hat{\mathbf{B}}\mathbf{C}\mathbf{e} - \mathbf{K}\mathbf{s} - \gamma\frac{\mathbf{s}}{\|\mathbf{s}\|} + (\mathbf{1}+\eta)\mathbf{W}\boldsymbol{\sigma}$$

$\dot{\mathbf{w}}_i = -\eta\mathbf{s}\sigma_i$; $\quad$ *i = 1, 2.*

With $\mathbf{K} = \begin{bmatrix} 75 & 0 \\ 0 & 75 \end{bmatrix}$; γ =10; η = 0.1;

The initial state of the robot is $q_{10} = -0.4$; $q_{20} = 1.85$.

Gaussian function parameters of the neural network: $\lambda_1 = \lambda_2 = 10$; $c_1 = 0.1$; $c_2 = 0.3$;

Table 1: Technical parameters of the 2 DOF robot

	First Joint	Second Joint
Mass of link m_{li} [kg]	50.0	50.0
Mass of the motor in joint m_{mi} [kg]	5.0	5.0
Inertia moment of link I_i [kg.m^2]	10.0	10.0
Length of the joint l_i [m]	1.0	1.0
Distance from centre point of gravity to joint l_{gi} [m]	0.5	0.5

The uncertainty of the robot dynamics is up to 20% of real value.

ΔM = 20%M; ΔB = 20%B; Δg = 20%g

Assumed unknown friction and noises:

$$\mathbf{d}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{d}(\mathbf{t}) = \begin{pmatrix} 10\sin(20t) + 1 + 4\dot{q}_1 \\ 10\cos(20t) + 4\dot{q}_2 \end{pmatrix}$$

Matrix **C** of the sliding surface is chosen as: $\mathbf{C} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$.

Point A has co-ordinates (1; 0.5) and Point B (0.5; 1) m. Time T is chosen 10 s. The desired trajectory is designed by parametric method described in [13]. With acceleration $a = 0.05 m/s^2$ we get the constant speed starting at $t_c = 1.705 s$.

Simulation results: Simulation results are shown on Figures 4, 5, 6, 7,8 and 9.

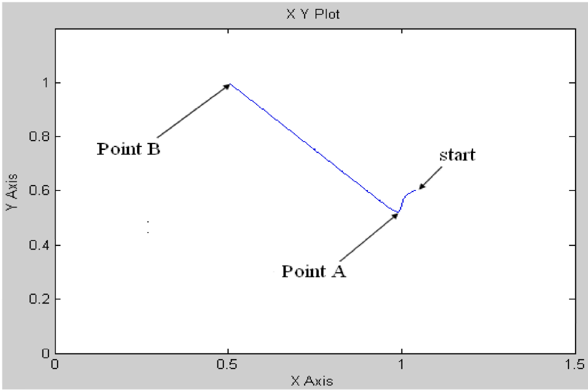

Figure 4: Actual path of robot hand in Cartesian space

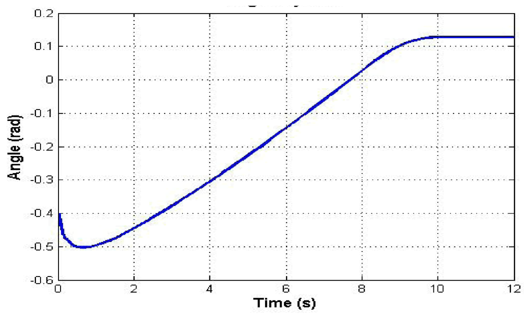

Figure 5a: Angle of joint 1

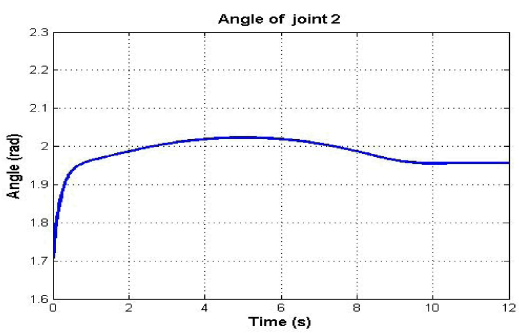

Figure 5b: Angle of joint 2

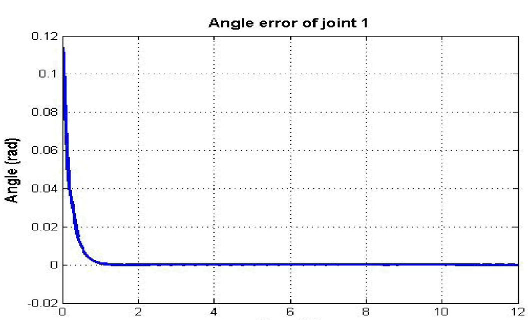

Figure 6a: Angle error of joint 1, at T =10s is 1x10^{-4} rad.

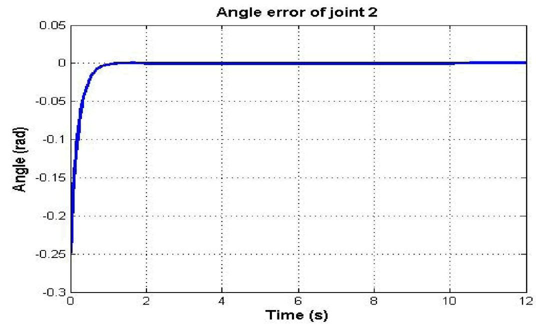

Figure 6b: Angle error of joint 2, at T =10s is 5x10^{-4} rad.

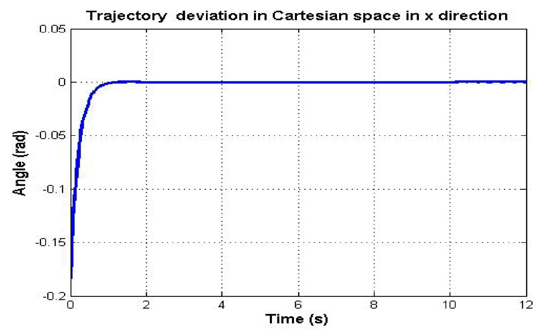

Figure 7a: Trajectory deviation in Cartesian space in x direction

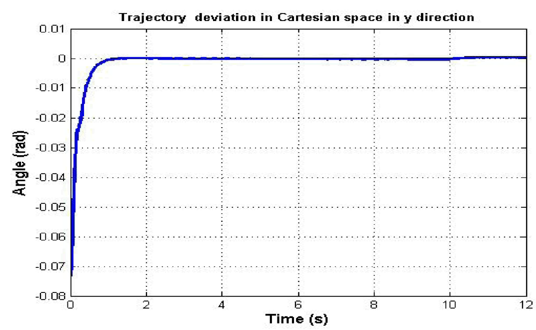

Figure 7b: Trajectory deviation in Cartesian space in y direction

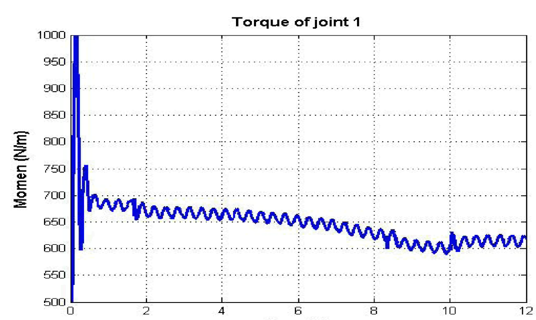

Figure 8a: Torque of joint 1

Figure 8a: Torque of joint 2

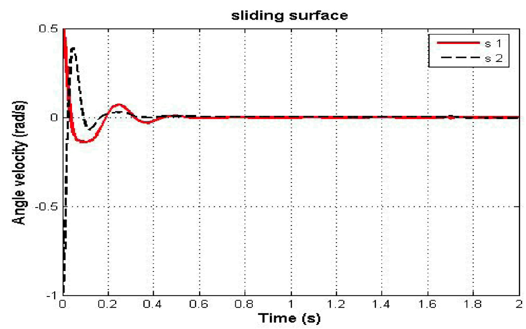

Figure 9: Sliding surface s

The simulation shows that the robot has tracked very close to the desired trajectory while the model uncertainty is up to 20% and unknown friction as well as noises acting on its joints. We have run the simulation several times with different uncertainties, frictions, noises, initial states and freely chosen c_i, λ_i, η, **C, K** parameters. The simulation results are very robust. These demonstrate the correctness and applicability of the proposed method.

6. SUMARY AND CONCLUSIONS

This paper has proposed a robust sliding mode control model using a neural network to compensate for uncertainties of the robot and proved the stability of the whole system by Lyapunov stability theorem. The neural network allows the further reduction of error caused by the uncertainties, friction and noises acting on the robot. Simulation on a 2 DOF robot following a desired trajectory in a Cartesian space brought good results corresponded to the theory of the proposed method. During the simulation with different levels of uncertainties and noises as well as different chosen parameters in sliding surface and in on-line learning algorithm we always got convergence results that demonstrated the stability and robustness of the method. The optimal parameters for fast convergence and good transient quality may be found using Genetic Algorithms.

REFERENCES

[1] John J. Cr. Caig *"Robotics mechanism and Control"* Second Edition, 1955.

[2] N.E Cotter – *The Stone- Weierstrass and Its Application to Neural Networks*. IEEE Trans on Neural Networks, Vol. 1, N4 1990 pp 290 – 295.

[3] J. Somlo - B. Lantos - P. T. Cat - *Advanced Robot Control*. Akademiai Kiado. Budapest 1997.

[4] Psaltis, D., Sideris, A and Yamamura, A – *Neural Controllers*, Proc IEEE 1st Int. Conf. Neural Networks, San Diego, Jun 1987.

[5] Ahmet Karakasoglu, M. K. Sundareshan – *A recurrent Neural network-based Adaptive Variable Structure Model-following Control of Robotic Manipulators*. Automatica Vol 31 No 10 pp 1495 – 1507. 1995.

[6] Chin - Teng Lin and C.S George Lee - *Neural Fuzzy systems* - Book is to the Chiao-Tung University Centennial 1996.

[7] Ricardo Carelli,H. Daniel Patino, and Benjamin R. Kunchen – *Neural Networks for advanced control of robot manipulators*. IEEE Transactions on neural networks, Vol. 13. No 2, pp 343-353, 2002..

[8] K. Jezernik, M.Rodic, R. Safaric – *Neural network sliding mode robot control*, Roboticka, Vol. 15, pp 23-30, 1997.

[9] Okyay Kaynak, M. On der Efe, ., *Sliding mode Control of a Three Degrees of Freedom Anthropoid Robot by Driving the Controller Parameters to an Equivalent Regime.* Bogazici University, Electrical and Electronic Eng. Dep. Bebek, 80815 Istanbul Turkey, pp 632 to 639, Vol.122, December 2000.

[10] Uchikawa, A. Ishiguro and T. Fururashi - *A neural network compensation for uncertainties of robotic manipulator.* IEEE Trans on Neural Networks, 7(2) 1996, pp 388-399.

[11] F.C Sun, Z.Q Sun and P.Y Woo "*Neural network – based adaptive controller design of robotic manipulator with observer*" IEEE Trans on Neural Networks, 12(1) 2001, 54-57.

[12] Utkin, V.L - *Sliding modes in control and optimization - 1992.*

[13] Lozenzo Sciavicco, Bruno Sicilliano - Modeling *and Control of robot manipulators*, International Edition 1996.

[14] H. Asada, J. J. E. Slotine – *Robot Analysis and Control*, Massachusetts Institute of Technology, 1985.