Adaptive RBFNN-HOMS Nonsingular Terminal Sliding Control of n-DOF Robotic Manipulator

Amar Rezoug*, **Bertrand Tondu****, **Mustapha Hamerlain*** and **Mohamed Tadjine*****

**DRP, Center for Development of Advanced Technologies, Baba Hassen 16303, Algiers, Algeria (Tel: +213555450740, e-mail: arezoug@ cdta.dz).*
*** INSA-LAAS, Groupe GEPETTO, Pôle RIA, 135 Avenue de Ranguil 31077, Toulouse, France (bertrand.tondu@ insa-toulouse.fr).*
****PCL, Automatic Control Department, polytechnic national school Algiers, Algeria (tadjine@yahoo.fr).*

Abstract: Hybridization of the Sliding Mode Control schemes and the Artificial Intelligence techniques is relatively a new way in the control domain. This paper proposes a robust control scheme by the adoption of Non-singular Terminal Sliding Mode Control (NTSMC), Higher Order Sliding Mode (HOSM) and Neural Network (NN) structure for n-DOF robotic manipulator. The NTSMC is used with time delay estimation method where the equivalent control term is synthesized without requirement of the robot model. In order to overcome the chattering drawback of the NTSMC, the discontinuous term is replaced by an adaptive HOSM controller. The adaptive HOSM controller consists of the Super Twisting algorithm which is estimated adaptively using Radial Based Function Neural Network (RBFNN) structure. The used RBFNN is learned online without requirement of a prior knowledge of training data. The stability is proved using a candidate Lyapunov function and the controller parameters are adjusted adaptively. The superiority and the effectiveness of the proposed approach are tested under 2-DOF robot manipulator in trajectory tracking mode and compared with NTSMC.

Keywords: Terminal Sliding Mode Control, Super Twisting Algorithm, Radial Based Function Neural Networks, Time Delay Estimation, Robot Manipulator, Lyapunov Stability.

1. INTRODUCTION

During the last decades, robotic manipulators have been widely investigated in both academy and industry domains due to some superior advantages such as higher accuracy and stiffness, speed trajectory tracking and so on (Van, *et al.* 2013, Fenga, *et al.* 2002, Jin, *et al.* 2013 Jin, *et al.* 2009, Kang, 2012, Jin, *et al.* 2011, Sadati, 2008, Rezoug, *et al.* 2012, Capisania, *et al.* 2009, Qi, *et al.* 2013, Lin, 2006, Cao, *et al.* 2012). However, robotic manipulators are complex systems with multiple nonlinearities like Friction, higher coupled dynamic, etc. Furthermore, these systems have always large structured and unstructured inherent uncertainties in their models which even cause instability. For controlling robotic manipulators, several PID schemes have been proposed due to their simple structure and ease of implement with low cost (Islam, *et* al. 2014, Moldoveanu, *et* al. 2005). Nevertheless, since the dynamic of the robotic manipulators are highly nonlinear, the performance of the linear controllers is very limited. To overcome the shortcomings of the linear controllers, many nonlinear controllers have been proposed for robotic manipulator such as feedback linearization and neural networks (Moradi, *et* al. 2013), adaptive control (Rezoug, *et* al. 2012), computed torque control (Lewis, *et* al. 2004) and sliding mode control (Capisania, *et al.* 2009, Mondal, *et al.* 2014, Kumar, *et al.* 2014, Lei, *et al.* 2014, Lin, *et al.* 2003), and others.

Among all the nonlinear control methods, Sliding Mode Control (SMC) (Fenga, *et al.* 2002,Jin, *et al.* 2013, Jin, *et al.* 2009) has found large acceptance and continues to attract a particular attention from the research community because of its simplicity. SMC is well-known as nonlinear and naturally robust control method which is suitable for controlling uncertain systems in presence of measurement errors and external disturbances, etc. (Fenga, *et al.* 2002,Jin, *et al.* 2013, Jin, *et al.* 2009 Perruquetti, 2002, Bandyopadhyay, *et al.* 2013, Shtessel, et al. 2014). SMC can be subdivided into two main classes. The first class consists of the traditional SMC called first order SMC. The principal drawback of this method is the chattering phenomenon which generates adverse effects for the controlled system (Perruquetti, 2002, Bandyopadhyay, *et al.* 2013, Shtessel, et al. 2014). In order to overcome the conventional SMC drawback, the concept of Higher Order Sliding Mode (HOSM) has been proposed as a new generation of the SMC (Perruquetti, 2002, Bandyopadhyay, *et al.* 2013, Shtessel, et al. 2014, Levant, 1993). The HOSM can be considered as the second class of the SMC. Some HOSMs have been applied to robotic manipulators such as: Twisting Algorithm and Super-Twisting Algorithm in (Kumar, *et al.* 2014, Lei, *et al.* 2014, Lin, *et al.* 2003), Sub-Optimal Sliding Mode (Capisania, *et al.* 2009). All HOSM controllers are very sensitive to the noise effects.

Terminal Sliding Mode Control (TSMC) is a particular case of the Higher Order Sliding Mode control (HOSM) which has been used also as conventional SMC (Mondal, *et al.* 2014, Rezoug, *et al.* 2011,Tang, *et al.* 2006). This technique has been studied to further improve the control performance, to achieve a best finite-time convergence with an asymptotic stability. Several terminal sliding mode controls have been applied to the robotic manipulators in (Mondal, *et al.* 2014) the Second Order Terminal Sliding Mode is proposed however this method is very sensitive to the noise effects. The Non-singular Terminal Sliding Mode Control (NTSMC) based on the robot model given in (Fenga, *et al.* 2002) this approach takes into account the parametric uncertainties of the robot manipulator model. Since the exact models of the robotic systems and, the aforementioned bounds of the controller parameters' are not always obtainable in practice, then this approach is complicated to implement.

Among the solution usually employed to overcome the robot manipulator model unknown consists of the estimation of the robot model and/or the controller by using artificial intelligence techniques (Van, *et al.* 2013, 27. Manceur, Lin, *et al.* 2003, et *al.* 2012). Neural Networks schemes are one of the artificial intelligence approaches used in the control domain for systems which have uncertain dynamics. Several works have exploited neural networks for controlling robotic systems (Tang, *et al.* 2006, Kumar, *et al.* 2014, Lei, *et al.* 2014, Lin, *et al.* 2003, T. Tai, *et al.* 2010, Fei, *et al.* 2012, Lian, *et al.* 2011). Traditional back propagation neural networks have the inconvenience of slow learning speed and local minimal convergence. RBFNN is a particular type of feed-forward artificial neural network, which contains only a single hidden layer of neurons usually characterized by Gaussian Activation Functions (GAF). Since RBFNN can be used without requirement of a training data, we can consider it as the best candidate to solve the classical neural network problems.

To the best of authors' knowledge, the hybridizations of the conventional SMC, higher order sliding mode control and artificial intelligences techniques are proposed only in (Van, *et al.* 2013, Manceur, *et al.* 2012). In this paper, we propose a new adaptive, stable, speed and robust control approach for controlling n-DOF robotic manipulator. Called Adaptive RBFNN-Super-Twisting Non Singular Terminal Sliding Mode Control (RBFNTSMSTW), this control approach particularly consists of the combination of NTSMC, STW controller and RBFNNs. Compared with other control approaches of the same category; Super-Twisting algorithm requires only the information of the sliding set for controlling in a finite time a studied system. The RBFNNs are used to estimate the two terms of super twisting algorithm in order to remove the chattering effect caused by the hitting control part of the NTSMC. Stability of the robot in close loop is guaranteed using Lyapunov theorem. The proposed control scheme allows us (1) To avoid the nonlinear modelling problems, (2) To guarantee the stability and the robustness of the robot and (3) To attenuate of the chattering effects (4) to improve the speed convergence of the state space variables. In order to prove the effectiveness of the proposed approach, simulation experiments are realized under 2-DOF robot manipulator.

This paper is organized as follow: In section 2, the robot manipulator model and the non-singular terminal sliding mode controller based on the time delay estimation method are presented. In section 3, the proposed Adaptive RBFNN-HOMS Non-singular Terminal Sliding Control for *n*- DOF robot manipulator is designed. In order to show the proposed control scheme superiority and effectiveness, the simulation experiments in trajectory tracking are given in section 4. Finally, in section 5, the paper is closed by a conclusion.

2. PRELIMINARY

The robot manipulator dynamic model and the non-singular terminal sliding mode control based on the Time Delay Estimation (TDE) method will be presented in this section.

2.1. *n-DOF robot manipulator model*

The standard form of the *n*-DOF robotic manipulator dynamic model is given as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(q,\dot{q}) + \tau_d = \tau \qquad (1)$$

Where:

- $M(q) \in R^{nxn}$ is the Inertial matrix which is symmetric positive non-singular and bounded by $m_{min}\|x\|^2 \le x^T M x \le m_{max}\|x\|^2 \quad \forall x \in R^n$, with m_{max} and m_{min} are minimum and maximum Eigenvalues of $M(q)$.
- $C(q,\dot{q}) \in R^{nxn}$ is the matrix of Centrifugal and Coriolis terms.
- $G(q) \in R^{nx1}$ is the vector of Gravitational force.
- $F(q,\dot{q}) \in R^{nx1}$ is the vector of Friction.
- $q \in R^{nx1}, \dot{q} \in R^{nx1}$ and $\ddot{q} \in R^{nx1}$ are the Position, the Velocity and the Acceleration vectors, respectively.
- $\tau \in R^n$ is the torque input vector.
- $\tau_d \in R^{nx1}$ is the vector of generalized input due to the disturbances (such as an add payload, etc.) , with $\|\tau_d\| < \beta_d$, and β_d is a positive real value.

2.2. *Nonsingular Terminal Sliding Mode Control based on the TDE*

In order to achieve the convergence of the angular positions of the robot to zero in finite time, the objective now consists of the control law designing. As in conventional SMC, the TSMC requires the choosing of the sliding variable. Conventional terminal sliding variable is given as: $S(X) = x_2 + \Lambda . x_1{}^{\rho}$ .Where: $X = [x_1, x_2]^T = [q - q_d, \dot{q} - \dot{q}_d]^T$ and $\rho = q/p$, with p and q are positive odd integers with $p > q$. $\Lambda = \mathrm{diag}(\lambda_{ii}) > 0$ is a diagonal matrix with real positive values obtained by "trial and error" in real applications. The use of the sliding variable $S(X) = x_2 + \Lambda . x_1{}^{\rho}$ will generate the so-called singularity problem. This problem resides in the division by zero which appeared in the general form of the TSMC (Fenga, et *al*. 2002). In order to overcome this drawback, we have adopted the Non Singular Terminal Sliding Mode Control (NTSMC) initially proposed in (Fenga, et *al*. 2002). In NTSMC, it is proposed the modification of the terminal sliding variable as follows:

$$S = x_1 + \Lambda^{-1} x_2^{\gamma} \tag{2}$$

Where: $1 < \gamma = \rho^{-1} < 2$

Let us consider the diagonal matrix $\bar{M} = \mathrm{diag}(m_{ii})$ which has positive real values with (i=1…n). Substituting $\bar{M}$ in (1), the dynamic of the *n*-DOF robotic manipulator can be rewritten as:

$$\bar{M}\ddot{q} + N(q,\dot{q},\ddot{q}) = \tau \tag{3}$$

Where:
$N(q,\dot{q},\ddot{q}) = \left[M(q) - \bar{M}\right]\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(q,\dot{q}) + \tau_d$

Sliding variable is chosen as given in equation (2). Making the time derivative of the sliding variable (equation (2)) equal to zero ($\dot{S} = 0$) and its substituting in equation (3) yields to the following control law:

$$\tau = \bar{M}\left[\ddot{q}_d + \gamma^{-1}\Lambda \dot{e}^{2-\gamma} + K_{sw}\mathrm{Sgn}(S)\right] + \hat{N}(q,\dot{q},\ddot{q}) \tag{4}$$

Where: $e = x_1$ and $\dot{e} = x_2$. $\hat{N}(q,\dot{q},\ddot{q})$ is obtained using the principle of the Time Delay Estimation method such as:

$$\hat{N}(q,\dot{q},\ddot{q}) = N(q,\dot{q},\ddot{q})_{t-L} \tag{5}$$

Where: $\hat{N}(q,\dot{q},\ddot{q})$ is the estimation of $N(q,\dot{q},\ddot{q})$ at t-L. Then, $N(q,\dot{q},\ddot{q})_{t-L}$ is the value of $N(q,\dot{q},\ddot{q})$ at t-L, and L is chosen usually as the sampling time. This assumption is available only if the sampling time is sufficiently fast. Using equations (4) and (5), we can write $N(q,\dot{q},\ddot{q})_{t-L}$ as:

$$N(q,\dot{q},\ddot{q})_{t-L} = \tau_{t-L} - \bar{M}\ddot{q}_{t-L} \tag{6}$$

The combination of equations (4),(5) and (6) yield to the NTSMC based on the TDE as:

$$\tau = \tau_{t-L} + \bar{M}\left[\ddot{q}_d - \ddot{q}_{t-L} + \gamma^{-1}.\Lambda.\dot{e}^{2-\gamma} + K_{sw}\mathrm{Sign}(S)\right] \tag{7}$$

3. PROPOSED CONTROLLER

This section is composed by two subsections. In the subsection 3.1 Non-singular Terminal Super Twisting Control terminal sliding mode controller for *n*-DOF robot manipulator is proposed where the Super Twisting is used as a solution to reduce the chattering drawback. In the subsection 3.2 intituled "Adaptive RBFNN-HOMS Nonsingular Terminal Sliding Control" the estimation of the super twisting controller parts using the radial based neural networks is proposed. Stability analysis and the parameters adjustment are also detailed in in this subsection.

3.1. *Non-singular Terminal Super Twisting Control*

In order to overcome NTSMC drawback (the chattering caused by Sign(S) in equation 7), we have opted to use the HOSM controller. Among the HOSM controller most used is the STW. This last is developed and analysed for controlling only systems having relative degree equal to one with respect to the input. The STW is used to attenuate the chattering effect. The STW law consists of the addition of two terms (Shtessel, et al. 2014, Levant, 1993, Levant, 2003), which has the nonlinear PI profile (equation 8). The advantage of the STW consists of its dependence only of the sliding variable compared with twisting and sub-optimal algorithms which are depended on the derivative of the sliding variables. The multivariable STW algorithm for n-DOF robotic manipulator can be written as:

$$U = -\beta.\|S\|^{1/2}\mathrm{sign}(S) - \alpha\int_0^{t_c}\mathrm{sign}(S)dt \tag{8}$$

Where: $\beta = \mathrm{diag}(\beta_{ii}) \in R^{n\times n}$ and $\alpha = \mathrm{diag}(\alpha_{ii}) \in R^{n\times n}$ are positive diagonal matrices.

At $t=t_c$ equation (8) is substituted by:

$$U = -\beta\|S\|^{1/2}\mathrm{sign}(S) - \alpha\cdot t_c\cdot \mathrm{sign}(S) \tag{9}$$

The optimal case of the control law (9) is given as:

$$U^* = -\beta^*\|S\|^{1/2}\mathrm{sign}(S) - \alpha^*\cdot t_c\cdot \mathrm{sign}(S) \tag{10}$$

Where: $\beta^* = \mathrm{diag}(\beta^*_{ii}) \in R^{n\times n}$ and $\alpha^* = \mathrm{diag}(\alpha^*_{ii}) \in R^{n\times n}$ are positive diagonal matrices. The positive control of (10) can be written as:

$$\|U^*\| = \beta^*\|S\|^{1/2} + \alpha^*\cdot t_c \tag{11}$$

Where t_c is the convergence time which lets the controlled system to converge in the neighborhood of υ, where υ is defined as (Manceur, *et al.* 2012):

$$\|S\| \le \upsilon \tag{12}$$

With: $t_c = \begin{cases} t & \text{if } \|S\| > \upsilon \\ t_\upsilon & \text{if } \|S\| \le \upsilon \end{cases}$

t_c is chosen to obtain the final time value corresponding to $\|S\| = \upsilon$ (Manceur, *et al.* 2012).

The general form of the control law based on SMC and STW for any systems can be written as (Van, *et al.* 2013, Levant, 1993, Manceur, *et al.* 2012):

$$u = u_{eq} + u_{STW}. \tag{13}$$

Where:

u_{eq}: Is the equivalent control term.

u_{STW}: Is the super twisting control term.

Using equation (13) the control law (7) can be written as:

$$\tau = \tau_{t-L} + \bar{M}\Big[\ddot{q}_d - \ddot{q}_{t-L} + \gamma.\Lambda.\dot{e}^{2-\gamma} + \beta\|S\|^{1/2}\,\mathrm{sign}(S) + \alpha\int_0^t \mathrm{sign}(S(\tau))d\tau\Big] \tag{14}$$

The nonsingular terminal STW controller (equation 14) contains an integral discontinuous function which gives only chattering attenuation (Tang, *et al.* 2006). In addition, the STW is very sensitive to noise measurement. In order to improve the performance of the control and to eliminate the chattering drawback, we will replace the two terms of the STW controller by two Radial Based Functions Neural Networks for each joint. This proposition will be detailed in the section 3.2.

3.2. *Adaptive RBFNN Nonsingular Terminal Sliding HOMS Control*

This section is composed of three subsections: First, the estimation of the STW parts using RBFNNs is given in a general form. Second, the used RBFNNs architectures are detailed. Third, stability analysis with RBFNNs parametric adjustment is designed.

A. *Estimation of the STW Controller Using RBFNN*

In this proposed approach two RBFNNs are used to estimate the discontinuous and the integral term of the STW control respectively, in this case, equation (8) is substituted by:

$$\tilde{U} = \begin{bmatrix} \|S\|^{1/2}\, W_1^{1T} h_1^1(S) + W_1^{2T} h_1^2(S) \\ \vdots \\ \|S\|^{1/2}\, W_n^{1T} h_n^1(S) + W_n^{2T} h_n^2(S) \end{bmatrix} \tag{15}$$

Where: W_i^1 is the weight vector of the RBFNN used for the estimation of the first part of the STW controller for joint i. W_i^2 is the weight vector of the RBFNN used for the estimation of the second part of the STW controller for joint i. $h_i^1(S)$ is the Gaussian activation vector of the RBFNN used for the estimation of the STW first part for joint i. $h_i^2(S)$ is the Gaussian activation vector of the RBFNN used to estimate the STW second part for joint i.

Substituting equation (15) in the control law (14) we obtain:

$$\tau = \tau_{t-L} + \bar{M}\left[\ddot{q}_d - \ddot{q}_{t-L} + \gamma^{-1}\cdot\Lambda\cdot\dot{e}^{2-\gamma} + \tilde{U}\right] \tag{16}$$

The proposed control law (16) is called Radial Based Function Neural Network Nonsingular Terminal Sliding Super Twisting Controller (RBFNTSMSTW). In the next section the architecture of the proposed RBFNNs will be detailed.

B. *The RBFNNs Architecture*

The RBFNN can generate a map between the input and the output with any desired accuracy, which is able to effectively estimate any nonlinear dynamics (Lei, *et al.* 2014). The RBFNN can give a best estimation only with enough sampling data and time and the RBFNN parameters can be adjusted well. The RBFNN is composed of three layers as shown in the Figure 1. In order to take into account the coupling, the input layer contains the sliding mode variable of joint i and the sliding variable of joint i+1. The middle layer is called hidden layer which is composed of m neurons within each neuron is characterized by Gaussian activation functions (GAF). The output layer gives the estimation of the STW parts. We have adopted the decentralized RBFNN architecture, where for each joint; we have two RBFNN for estimating the two parts of the STW, respectively. It is very interested to note that it is possible to use a Multi-Input Multi-Output (MIMO) RBFNN for controlling a robot manipulator as reported in (Kumar, *et al.* 2014, Lu, *et* al. 2010). However, the use of this structure has some disadvantages. First, the complexity and the misunderstanding of the MIMO RBFNN dynamics increase more and more with the increasing of the number of links to be controlled (in (Lu, *et* al. 2010), the authors have limited their study on 2-DOF robot manipulator). Second, the computing time will be very important with the increasing of the joints number. Because it is required to find the best combination for several joints at the same time, the adjustment of the MIMO parameters is delicate, etc.

The inputs of the used RBFNN are the sliding variable of joint i and the sliding variable of joint i+1. Each neuron of the hidden layer is characterized by Gaussian Activation Function (GAF) which is given by:

$$h_{ji}^1 = e^{-\left(\frac{\|X_{ji} - c_{ji}\|}{2\sigma_{ji}^2}\right)} \quad \text{and} \quad h_{ri}^2 = e^{-\left(\frac{\|X_{ri} - c_{ri}\|}{2\sigma_{ri}^2}\right)} \tag{17}$$

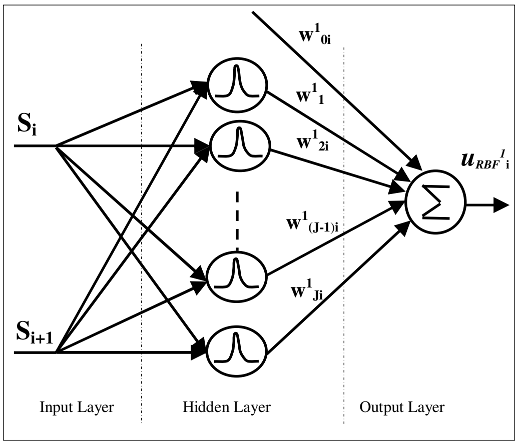
Where: c_{ji} and c_{ri} are the centers of the GAFs. $j = 1\ldots J$ and $r = 1\ldots R$, J and R are the RBFNNs sizes. σ_{ji} and σ_{ri} are the variances of the GAF, i is the number of joint.

The RBFNN output of joint i is computed by two terms as:

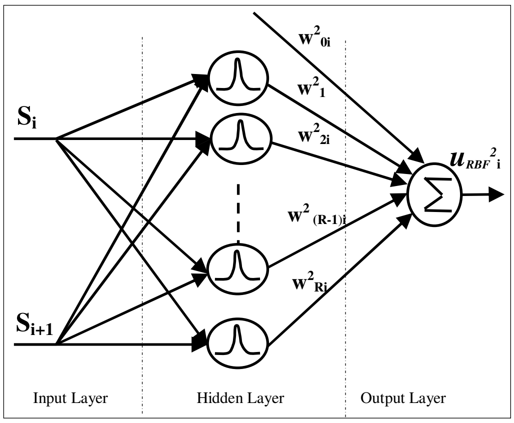
$$u^1_{RBFi} = \sum_{j=1}^{J} w^1_{ji} \times h^1_{ji} = W^1_i h^1_i$$
$$u^2_{RBFi} = \sum_{r=1}^{R} w^2_{ri} \times h^2_{ri} = W^2_i h^2_i \qquad (18)$$

Where: u^1_{RBFi} and u^2_{RBFi} are the estimated parts of the two terms of the STW for joint i. w^1_{ji} and w^2_{ri} are weights associated with the output for joint i. h^1_{ji} and h^2_{ri} are Gaussian functions. W^1_i and W^2_i are the weight vectors. The output of the neuron for given data is a radial function of the distance between the neuron center C_{ji} and C_{ri} and the sliding variables S for joint i and i+1 (Rezoug, *et al.* 2012) while:

$$\left\| S - C_{ji} \right\| = (S_i - c_{ji1})^2 + (S_{i+1} - c_{j(i+1)2})^2$$
$$\left\| S - C_{ri} \right\| = (S_i - c_{ri1})^2 + (S_{i+1} - c_{r(i+1)2})^2 \qquad (19)$$



(a)



(b)

Fig.1. Structure of the used RBFNNs (a) RBFNN for the estimation of the first part of the STW and (b) RBFNN for the estimation of the second part of the STW.

C. *Stability Analysis and Parameters Adjustment of the RBFNTSMSTW Controller*

For the *n*-DOF robotic manipulator controlled by the control law (16), we have chosen a candidate Lyapunov function as:

$$V = \frac{1}{2} S^T S + \sum_{i=1}^{n} \frac{1}{2\delta^1_i} \tilde{W}^{1T}_i \tilde{W}^1_i + \sum_{i=1}^{n} \frac{1}{2\delta^2_i} \tilde{W}^{2T}_i \tilde{W}^2_i \qquad (20)$$

Where: δ^1_i and δ^2_i are positive real values for each joint i. $\tilde{W}^1_i = W^1_i - W^{1*}_i$ and $\tilde{W}^2_i = W^2_i - W^{2*}_i$ .with W^{1*}_i and W^{2*}_i are the optimal weight vectors for each joint i.

The time derivative of equation (20) yields:

$$\dot{V} = S^T \dot{S} + \sum_{i=1}^{n} \frac{1}{\delta^1_i} \tilde{W}^{1T}_i \dot{W}^1_i + \sum_{i=1}^{n} \frac{1}{\delta^2_i} \tilde{W}^{2T}_i \dot{W}^2_i \qquad (21)$$

The time derivative of the sliding variable is given by the following equation

$$\dot{S} = \dot{e} + \gamma \Lambda^{-1} \operatorname{diag}(e^{\gamma-1}) \ddot{e} \qquad (22)$$

Substituting equation (8) in the sliding variables time derivative $\dot{S}$ and yields:

$$\dot{S} = \Big\{ \dot{e} + \gamma \Lambda^{-1} \operatorname{diag}(e^{1-\gamma})$$
$$\times \left[-\gamma \Lambda^{-1} \dot{e}^{2-\gamma} \left(-\beta \|S\|^{1/2} \operatorname{sign}(S) - \alpha \int_0^t \operatorname{sign}(S(\tau)) d\tau + \varepsilon \right) \right] \Big\}$$
$$= \gamma \Lambda^{-1} \dot{e}^{\gamma-1} \left(-\beta \|S\|^{1/2} \operatorname{sign}(S) - \alpha \int_0^t \operatorname{sign}(S(\tau)) d\tau + \varepsilon \right) \qquad (23)$$

Where: $\varepsilon \overset{\Delta}{=} \bar{M}^{-1}(N - N_{t-T})$.

Replacing STW controller (equation 8) by the RBFNN given in equation (15), $\dot{S}$ becomes:

$$\dot{S} = \gamma \Lambda^{-1} \dot{e}^{\gamma-1} \left(-\tilde{U} + \varepsilon \right)$$
$$= \gamma \Lambda^{-1} \dot{e}^{\gamma-1} \left(-\begin{bmatrix} \|S\|^{1/2} W^{1T}_1 h^1_1(S) + W^{2T}_1 h^2_1(S) \\ \vdots \\ \|S\|^{1/2} W^{1T}_n h^1_n(S) + W^{2T}_n h^2_n(S) \end{bmatrix} + \varepsilon \right) \qquad (24)$$

Let us denote $\tilde{U}^* = \begin{bmatrix} \|S\|^{1/2} W^{1*T}_1 h^1_1(S) + W^{2*T}_1 h^2_1(S) \\ \vdots \\ \|S\|^{1/2} W^{1*T}_n h^1_n(S) + W^{2*T}_n h^2_n(S) \end{bmatrix}$ as the optimal case of the equation (15), if we add and subtract $\tilde{U}^*$ from the equation (24), in this case, $\dot{S}$ becomes:

$$\dot{S}=\gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-\begin{bmatrix}\|S\|^{1/2}\left(W_1^1-W_1^{1*}\right)^T h_1^1(S)+\left(W_1^2-W_1^{2*}\right)^T h_1^2(S)\\ \vdots \\ \|S\|^{1/2}\left(W_n^1-W_n^{1*}\right)^T h_n^1(S)+\left(W_n^2-W_n^{2*}\right)^T h_n^2(S)\end{bmatrix}-\tilde{U}^*+\varepsilon\right)$$
$$=\gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-\begin{bmatrix}\|S\|^{1/2}\,\tilde{W}_1^{1T}h_1^1(S)+\tilde{W}_1^{2T}h_1^2(S)\\ \vdots \\ \|S\|^{1/2}\,\tilde{W}_n^{1T}h_n^1(S)+\tilde{W}_n^{2T}h_n^2(S)\end{bmatrix}-\tilde{U}^*+\varepsilon\right) \tag{25}$$

Substituting equation (25) in (21) the time derivative of the Lyapunov function becomes:

$$\dot{V}=S^T\left[\gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-\begin{bmatrix}\|S\|^{1/2}\,\tilde{W}_1^{1T}h_1^1(S)+\tilde{W}_1^{2T}h_1^2(S)\\ \vdots \\ \|S\|^{1/2}\,\tilde{W}_n^{1T}h_n^1(S)+\tilde{W}_n^{2T}h_n^2(S)\end{bmatrix}-\tilde{U}^*+\varepsilon\right)\right]$$
$$+\sum_{i=1}^{n}\frac{1}{\delta_i^1}\tilde{W}_i^{1T}\dot{W}_i^1+\sum_{i=1}^{n}\frac{1}{\delta_i^2}\tilde{W}_i^{2T}\dot{W}_i^2$$
$$=S^T\left[\gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-\tilde{U}^*+\varepsilon\right)\right]$$
$$+\gamma\Lambda^{-1}S^T\dot{e}^{\gamma-1}\left[-\begin{bmatrix}\|S\|^{1/2}\,\tilde{W}_1^{1T}h_1^1(S)+\tilde{W}_1^{2T}h_1^2(S)\\ \vdots \\ \|S\|^{1/2}\,\tilde{W}_n^{1T}h_n^1(S)+\tilde{W}_n^{2T}h_n^2(S)\end{bmatrix}\right]$$
$$+\sum_{i=1}^{n}\frac{1}{\delta_i^1}\tilde{W}_i^{1T}\dot{W}_i^1+\sum_{i=1}^{n}\frac{1}{\delta_i^2}\tilde{W}_i^{2T}\dot{W}_i^2$$
$$=S^T\left[\gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-\tilde{U}^*+\varepsilon\right)\right]$$
$$+\sum_{i=1}^{n}\tilde{W}_i^{1T}\left(\frac{1}{\delta_i^1}\dot{W}_i^1-\gamma\lambda_i^{-1}\dot{e}_i^{\gamma-1}S_i\|S\|^{1/2}h_i^1(S)\right)$$
$$+\sum_{i=1}^{n}\tilde{W}_i^{2T}\left(\frac{1}{\delta_i^2}\dot{W}_i^2-\gamma\lambda_i^{-1}\dot{e}_i^{\gamma-1}S_ih_i^2(S)\right) \tag{26}$$

We take:

$$\dot{W}_i^1=\gamma\delta_i^1\lambda_i^{-1}\dot{e}_i^{1-\gamma}S_i\|S\|^{1/2}h_i^1(S)$$
$$\dot{W}_i^2=\gamma\delta_i^2\lambda_i^{-1}\dot{e}_i^{1-\gamma}S_ih_i^2(S) \tag{27}$$

Using equation (27) the time derivative of the Lyapunov candidate function becomes:

$$\dot{V}=S^T\left[\gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-U^*+\varepsilon\right)\right]$$
$$=\gamma\sum_{i=1}^{n}S_i\lambda_i^{-1}\dot{e}_i^{\gamma-1}\left(-\left(u_i^{1*}+u_i^{2*}\right)+\varepsilon_i\right) \tag{28}$$

We can write equation (28) as:

$$\dot{V}\le\gamma\sum_{i=1}^{n}S_i\lambda_i^{-1}\dot{e}_i^{\gamma-1}\left(-\left(\|u_i^{1*}\|+\|u_i^{2*}\|\right)+\varepsilon_i\right)$$
$$\le\gamma\sum_{i=1}^{n}S_i\lambda_i^{-1}\dot{e}_i^{\gamma-1}\left(-\left(\alpha_i^*t_{ci}+\beta_i^*\|S_i\|^{1/2}\right)+\varepsilon_i\right) \tag{29}$$

If we take $\alpha_i^*t_{ci}+\beta_i^*\|S_i\|^{1/2}\ge\varepsilon_i$, the derivative of Lyapunov function is negative; then, the stability of the controlled system is confirmed**.**

The control system structure is composed of nominal control term and two STW-RBFNNs as shown in Figure 2.

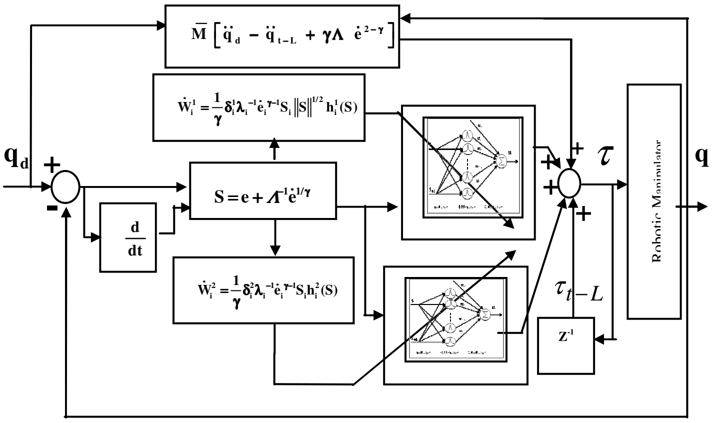


Fig. 2. Control structure scheme of RBFNTSMSTW.

4. SIMULATION RESULTS

In order to make in evidence the proposed control approach, simulation experiments have been realized using 2-degree of freedom robot manipulator depicted in figure 3. Trajectory tracking modes was adopted. The 2-DOF robot model is given by the equations (30), (31) and (32). Simulation has been coded using MATLAB environment under the ODE45 solver. Joint one and two are simulated for a time of 10 second with sampling time equal to 10ms.

The robot parameters are given as m_1= 1 *kg***,** m_2=2 *kg***,** l_1=1 *m***,** l_2=2 *m* and g=9.8 m/sec^{2}. The control parameters are: K_{sw}=(10.5,21.5), $\bar{M}=\mathrm{diag}(6.7,0.2)$, *q*=9 and *p*=11 for all joints, $\Lambda=\mathrm{diag}(1.1,2)$ $\delta_1^1=40,\delta_2^1=0.4$ and $\delta_1^2=15,\delta_2^2=2$. The used RBFNNs sizes are: RBFNN1 for joint 1 has 18 nodes, RBFNN 2 for joint 1 has 25 nodes, RBFNN 1 for joint 2 has 15 nodes and RBFNN 2 for joint 2 has 25 nodes. The central positions of the Gaussian functions $\mathbf{c_i}$ are selected from [-1,1] for all RBFNNs, and the spread factors are taken to be $\sigma_i^1=0.2$ $\sigma_i^2=0.2$ with i is the number of joints.

$$M(q)=\begin{bmatrix}(m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos(q_2) & m_2l_2^2+m_2l_1l_2\cos(q_2)\\ m_2l_2^2+m_2l_1l_2\cos(q_2) & m_2l_2^2\end{bmatrix} \tag{30}$$

$$C(q,\dot{q}) = \begin{bmatrix} -m_2 l_1 l_2 \sin q_2 (2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \\ -m_2 l_1 l_2 \sin q_2 \dot{q}_1 \dot{q}_2 \end{bmatrix} \tag{31}$$

$$G(q) = \begin{bmatrix} -(m_1 + m_2) g l_1 \sin q_1 - m_2 g l_2 \sin(q_1 + q_2) \\ -m_2 g l_2 \sin(q_1 + q_2) \end{bmatrix} \tag{32}$$

In order to avoid the zero division if $\dot{e}_i(t) = 0$ the RBFNTSMSTW controller is replaced by the super-twisting terminal sliding mode given by equation (14). The stability analysis and the RBFNNs parameters adjustment are detailed in the APPENDIX. A.

For all simulation, we have chosen to present the angular position, the errors in angular position and the control signals. The RBFNTSMSTW is presented by red lines and at the same time NTSMC is presented by the blue lines. The joint one and two move according to the references (equation 33).

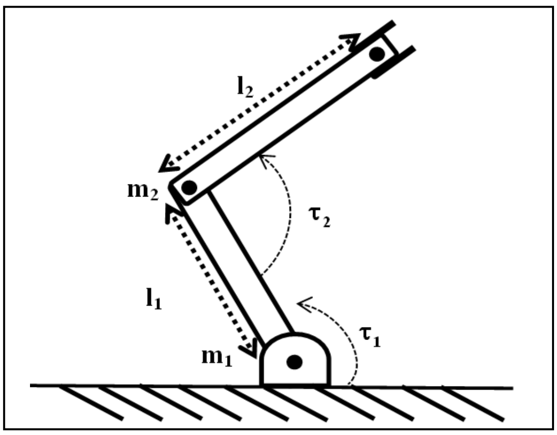
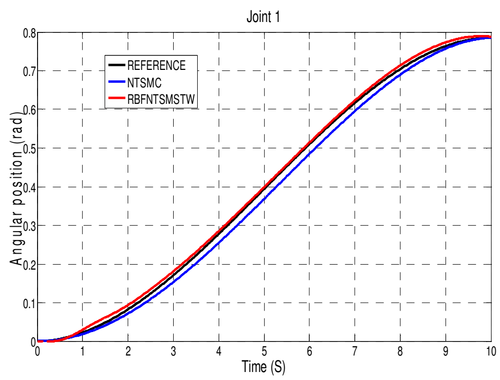
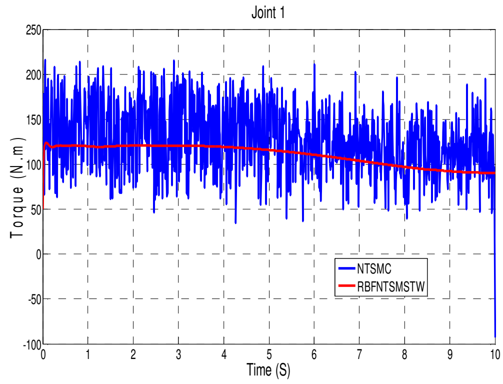


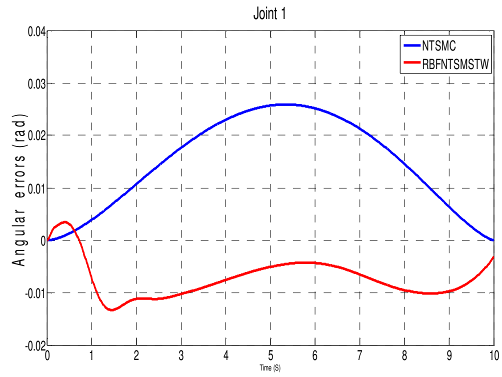
Fig. 3. 2-DOF robotic manipulator.

$$q_{d1} = \frac{\pi}{2000} t^3 - \frac{\pi}{400} t^2 \qquad q_{d1}(0) = 0 \qquad q_{d1}(10) = \frac{\pi}{4}$$

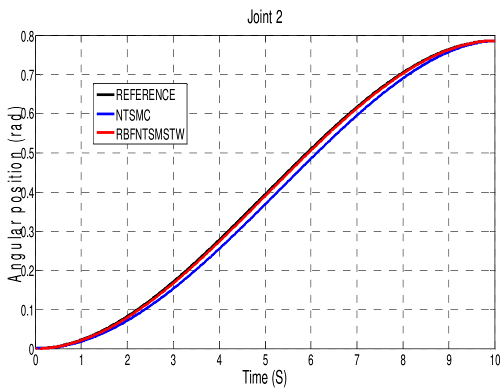
$$q_{d2} = \frac{\pi}{2000} t^3 - \frac{\pi}{400} t^2 \qquad q_{d2}(0) = 0 \qquad q_{d2}(10) = \frac{\pi}{4} \tag{33}$$

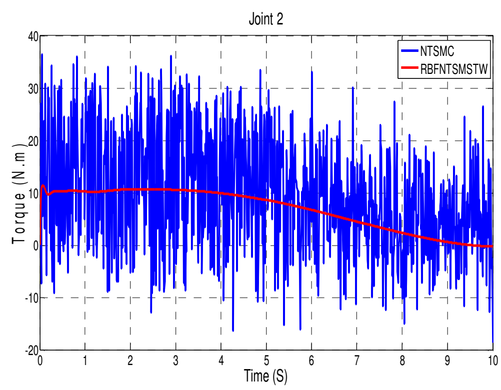


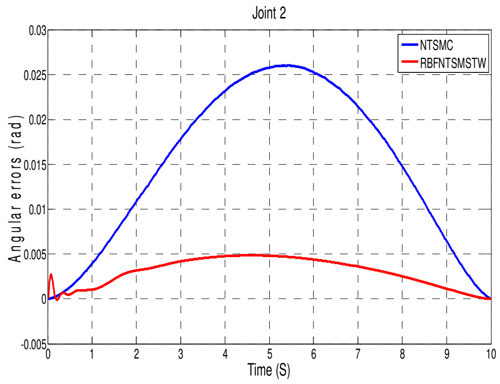




Joint 1







Joint 2

Fig.4. Simulation results for the application of RBFNTSMSTW and NTSMC to all joints in trajectory tracking mode.

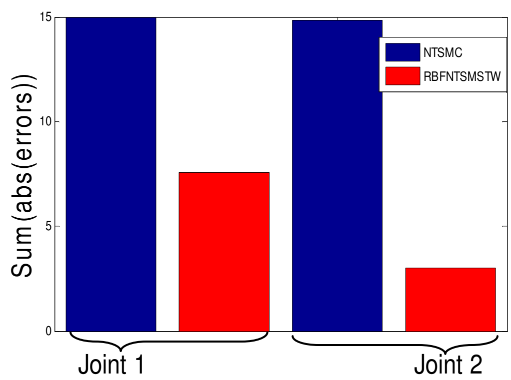
For all results (joint 1 and 2) RBFNTSMSTW presents better results compared with NTSMC. This is shown in the position responses, where we obtain a good amelioration of static errors (stabilization). In addition, the control signal is important in the case of NTSMC and presents higher chattering compared with RBFNTSMSTW. In the case of RBFNTSMSTW, we have relative smooth control signals which are explained by the effect of the RBFNN. The chattering is moved in all trajectory tracking. We can consider that this time delay is acceptable in the proposed case. Because the robot is placed against the gravitational effect, the control of the joint 1 is delicate compared with the joint 2.

A quantitative comparison is effectuated her in order to analyze which is the better controller. In this case we will calculate the following criterions:

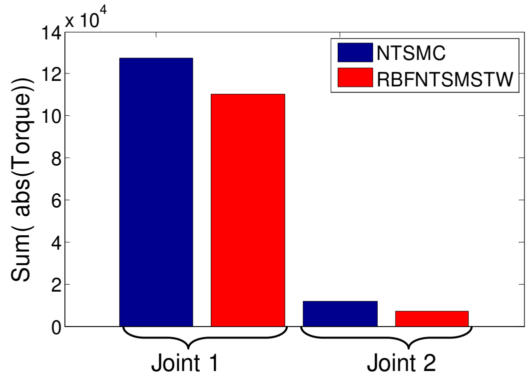
$$\mathrm{IAE} = \int_{t=0}^{t=t_f} |e| dt \tag{34}$$

$$\mathrm{IATorque} = \int_{t=0}^{t=t_f} |\tau| dt \tag{35}$$

Figure 5 presents the results using the criterions (34) and (35).



(a)



(b)

Fig.5. Quantitative comparisons between NTSMC and RBFNTSMSTW (a) sum of absolute errors (b) sum of control torques.

All errors and consummated energy are reduced in the case RBFNTSMSTW compare with NTSMC. We can illustrate from the results shown in the figure 5 that the RBFNTSMSTW is given better results compared with the NTSMC.

5. CONCLUSION

In this work, a new adaptive robust approach called Radial Based Function Neural Network Nonsingular Terminal Sliding Super Twisting Controller (RBFNTSMSTW) is proposed for controlling *n*-DOF robot manipulator. The NTSMC with time delay estimation method is used to design the equivalent control without requirement of model knowledge. RBFNNs based on STW is used to remove the chattering phenomenon and to improve the control performance. The feasibility and the effectiveness of the proposed approach have been proven through simulation experiments. The proposed RBFNTSMSTW presents better performance compared with the NTSMC. Then, the control can be achieved without the robot model knowledge with better performance and robustness.

REFERENCES

Astrom, K. J. and Hugglund, T. (1984). Automatic Tuning of Simple Regulators With Specifications on Phase And Amplitude Margins. Automatica. volume (20), issue (5) 645-651.

Bandyopadhyay, B. Janardhanan, S. and Spurgeon, S. K. (2013). Advances in Sliding Mode Control. Lecture Notes in Control and Information Sciences. Volume (440), Springer.

Cao, L. Chen, Y. Chen, X. Zhao, Y. and Fan, J. (2012). The Design and Analysis of Non-singular Terminal Adaptive Fuzzy Sliding-mode Controller. Proc. of the 2012 American Control Conference, 4625-4630, Canada.

Capisania, L. M. Ferraraa, A. and Magnania, L. (2009). Design and Experimental Validation of a Second-Order

Sliding-Mode Motion Controller for Robot Manipulator. Int. Journal of Control. Volume (82), Issue (2), 365-377.

Fei, J., H. Ding, S. Hou, Wang, S. and Xin, M. (2012). “Robust Adaptive Neural Sliding Mode Approach for Tracking Control of a MEMS Triaxial Gyroscope. Int J Adv Robotic Sy. Volume (9), 1-8.

Fenga, Y. Yu, X. Manc Z. (2002). “Non-Singular Terminal Sliding Mode Control of Rigid Manipulators,” Automatica Volume (38), Issue (2002), 2159 - 2167.

Jia, T. and Kang, G. (2012). An RBF Neural Network-Based Nonsingular Terminal Sliding Mode Controller for Robot Manipulators. 3th Int. Conf. on Intelligent Control and Information Processing. 72 – 76, China.

Jin, M. Lee, J. Chang, P. H. and Choi, C. (2009). Practical Nonsingular Terminal Sliding-Mode Control of Robot Manipulators for High-Accuracy Tracking Control. IEEE Trans. On Industrial Electronics. Volume (56), issue (9), 3593-3601.

Jin, M. Jin, Y. Chang, P. H. and Choi, C. (2011). High-Accuracy Tracking Control of Robot Manipulators Using Time Delay Estimation and Terminal Sliding Mode. Int. Journal. Adv Robotic Sy. Volume (8), issue (4), 65-78.

Jin, Y. Chang, P. H. Maolin, J. and Gweon, D. G. (2013) Stability Guaranteed Time Delay Control of Manipulators Using Nonlinear Damping and Terminal Sliding Mode. IEEE Trans. On Industrial Electronics. Volume (60), Issue (8), 3304-3317.

Islam, R. Iqbal, J. and Khan, Q. (2014). Design and Comparison of Two Control Strategies for Multi-DOF Articulated Robotic Arm Manipulator. journal of control engineering and applied informatics, Volume (16), issue (2), 28-39.

Moldoveanu, F. Comnac, V. Floroian, D. Boldisor C. (2005), Trajectory Tracking Control Of a Two-Link Robot Manipulator Using Variable Structure System Theory, journal of control engineering and applied informatics, Volume (7), issue (3), 56-62.

Kumar, N. Panwar, V. Borm, J.H. and Chai, J. (2014). Enhancing Precision Performance of Trajectory Tracking Controller for Robot Manipulators Using RBFNN and Adaptive Bound. Applied Mathematics and Computation. Volume (231), 320-328.

Lei, X. Ge, S. S. and Fang, J. (2014). Adaptive Neural Network Control of Small Unmanned Aerial Rotorcraft. J. Intell Robot Syst. Volume (75), 331–341.

Levant, A. (1993). Sliding Order And Sliding Accuracy In Sliding Mode Control. Int. J. Control, Volume (58), 1247-1263.

Levant, A. (2003). Higher Order Sliding Modes, Differentiation and Output Feedback Control. Int. J. Control, Volume (76), Issue (9/10), 924-941.

Lewis, F.L. Dawson, D.M. and Abdallah, C.T. (2004). Robot Manipulator and Control -Theory and Practice. Marcel Dekker. New York.

Lian, R.J. (2011). Intelligent Controller for Robotic Motion Control. IEEE Tran. On Industrials Electronics. Volume (58), issue (11), 5220–5230.

Lin, C. M. and Mon, Y. J. (2003). Hybrid Adaptive Fuzzy Controllers with Application to Robotic Systems. Fuzzy Sets and Systems. Volume (139), 151-165.

Lin, C. K. (2006). Nonsingular Terminal Sliding Mode Control of Robot Manipulators Using Fuzzy Wavelet Networks. IEEE Trans. On Fuzzy Systems. Volume (14), issue (6), 849-859.

Lu, H.C., Tsai, C.H., Chang, M.H., (2010). Radial Basis Function Neural Network with Sliding Mode Control for Robotic Manipulators. IEEE Int. Conf. on Systems Man and Cybernetics, 1209 – 1215. Turkey.

Manceur, M. Essounbouli, N. and Hamzaoui, A. (2012). Second Order Sliding Fuzzy Interval Type-2 Control for an Uncertain System With Real Application. IEEE Tran. On Fuzzy Systems. Volume (20), issue (2), 262 - 275.

Mondal, S. and Mahanta, C. (2014). Adaptive Second Order Terminal Sliding Mode Controller for Robotic Manipulators. Journal of the Franklin Institute. Volume (351), Issue (4), 2356-2377.

Moradi M. and Malekizade, H. (2013). Neural Network Identification Based Multivariable Feedback Linearization Robust Control for a Two-Link Manipulator”, J Intell Robot Syst. Volume (72),167-178.

Perruquetti, W. and Barbot, J.P. (2002). Sliding Mode Control in Engineering. Marcel Dekker, New York.

Qi, L. and Shi, H. (2013). Adaptive position tracking control of permanent magnet synchronous motor based on RBF fast terminal sliding mode control. Neurocomputing. Volume (115), 23–30.

Rezoug, A. Mahjoub, S. Hamerlain, M. and Tadjine, M. (2011). Fuzzy Terminal Sliding Mode Controller for Robot Arm Actuated by Pneumatic Artificial Muscles. 8th Int. Multi-Conference on Systems, Signals and Devices, 22-25, Tunisia.

Rezoug, A. Hamerlain, M. and Tadjine, M. (2012). Adaptive RBFNN type-2 fuzzy sliding mode controller for robot arm with pneumatic muscles. 2012 IEEE Int. Conf. on Robotics and Biomimetics. 1287-1292, Chine.

Rezoug, A. Tondu, B. Hamerlain, M. and Tadjine, M. (2013). Adaptive Nonsingular Terminal Sliding Mode Control For Robot Manipulator Actuated By Pneumatic Artificial Muscles. 2013 IEEE Int. Conf. on Robotics and Biomimetics, 334 – 339, Chine.

Sadati, N. Ghadami, R. (2008). Adaptive Multi-Model Sliding Mode Control of Robotic Manipulators Using Soft Computing. Neurocomputing. Volume (71), 2702-2710.

Shtessel, Y. Edwards, C. Fridman, L. and Levant, A. (2014). Sliding Mode Control and Observation. Springer, New York.

Tai, N. T., Ahn, K. K. 2010. A RBF Neural Network Sliding Mode Controller for SMA Actuator. Int. Journal of Control, Automation, and Systems. Volume (8), issue (6), 1296-1305.

Tang, Y. Sun, F. and Sun, Z. (2006). Neural Network Control of Flexible-Link Manipulators Using Sliding Mode. Neurocomputing. Volume (70), 288-295.

Van, M., Kang, H.J., Suh, Y.S. (2013). A Novel Fuzzy Second-Order Sliding Mode Observer- Controller for a T-S Fuzzy System With an Application for Robot Control. Int. J. of Precision Engineering and Manufacturing. Volume (14), issue (10), 1703-1711.

APPENDIX. A.

In the case of 2-DOF robot manipulator, the control law is given by equation (23) with i=1,2. In this case, we have four (4) RBFNNs, the general form of Lyapunov candidate in the case of 2-DOF robot manipulator (i=1, 2) is:

$$\begin{aligned} V &= \frac{1}{2}S^TS \\ &+ \frac{1}{2\delta_1^1}\tilde{W}_1^{1T}\tilde{W}_1^1 + \frac{1}{2\delta_2^1}\tilde{W}_2^{1T}\tilde{W}_2^1 + \frac{1}{2\delta_1^2}\tilde{W}_1^{2T}\tilde{W}_1^2 + \frac{1}{2\delta_2^2}\tilde{W}_2^{2T}\tilde{W}_2^2 \end{aligned} \quad (36)$$

The time derivative of the Lyapunov function (36) is given as:

$$\begin{aligned} \dot{V} &= S^T\dot{S} \\ &+ \frac{1}{\delta_1^1}\tilde{W}_1^{1T}\dot{W}_1^1 + \frac{1}{\delta_2^1}\tilde{W}_2^{1T}\dot{W}_2^1 + \frac{1}{\delta_1^2}\tilde{W}_1^{2T}\dot{W}_1^2 + \frac{1}{\delta_2^2}\tilde{W}_2^{2T}\dot{W}_2^2 \end{aligned} \quad (37)$$

Developing equation (37) as in equation (23), we have :

$$\begin{aligned} \dot{V} &= S^T\left[\dot{e} + \gamma\Lambda^{-1}\mathrm{diag}\left(e^{\gamma-1}\right)\ddot{e}\right] \\ &= S^T\Big\{\dot{e} + \gamma\Lambda\mathrm{diag}\left(e^{1-\gamma}\right) \\ &\times\left[-\gamma\Lambda^{-1}\dot{e}^{2-\gamma}\left(-\beta\|S\|^{1/2}\,\mathrm{sign}(S) - \alpha\int_0^t \mathrm{sign}(S(\tau))d\tau + \varepsilon\right)\right]\Big\} \\ &+ \frac{1}{\delta_1^1}\tilde{W}_1^{1T}\dot{W}_1^1 + \frac{1}{\delta_2^1}\tilde{W}_2^{1T}\dot{W}_2^1 + \frac{1}{\delta_1^2}\tilde{W}_1^{2T}\dot{W}_1^2 + \frac{1}{\delta_2^2}\tilde{W}_2^{2T}\dot{W}_2^2 \end{aligned} \quad (38)$$

The time derivative of the sliding variables is given as:

$$\begin{aligned} \dot{S} &= \gamma\Lambda^{-1}\dot{e}^{2-\gamma}\left(-\beta\|S\|^{1/2}\,\mathrm{sign}(S) - \alpha\int_0^t \mathrm{sign}(S(\tau))d\tau + \varepsilon\right) \\ &= \gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(\tilde{U} + \varepsilon\right) \\ &= \gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-\begin{bmatrix}\|S\|^{1/2}\,W_1^{1*}h_1^1(S) + W_1^{2T}h_1^2(S) \\ \|S\|^{1/2}\,W_2^{1T}h_2^1(S) + W_2^{2T}h_2^2(S)\end{bmatrix} + \varepsilon\right) \\ &= \gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-\begin{bmatrix}\|S\|^{1/2}\left(W_1^1 - W_1^{1*}\right)^T h_1^1(S) + \left(W_1^2 - W_1^{2*}\right)^T h_1^2(S) \\ \|S\|^{1/2}\left(W_2^1 - W_2^{1*}\right)^T h_2^1(S) + \left(W_2^2 - W_2^{2*}\right)^T h_2^2(S)\end{bmatrix} - \tilde{U}^* + \varepsilon\right) \\ &= \gamma\Lambda^{-1}\dot{e}^{\gamma-1}\left(-\begin{bmatrix}\|S\|^{1/2}\,\tilde{W}_1^{1T}h_1^1(S) + \tilde{W}_1^{2T}h_1^2(S) \\ \|S\|^{1/2}\,\tilde{W}_2^{1T}h_2^1(S) + \tilde{W}_2^{2T}h_2^2(S)\end{bmatrix} - \tilde{U}^* + \varepsilon\right) \end{aligned} \quad (39)$$

Substituting the last equation of (39) in (37) yields:

$$\begin{aligned} \dot{V} &= S^T\left[\gamma\Lambda^{-1}\dot{e}^{\gamma-1}(-\tilde{U}^* + \varepsilon)\right] \\ &= S_1\gamma\lambda_1^{-1}\dot{e}_1^{\gamma-1}(-(\tilde{u}_1^{1*} + \tilde{u}_2^{1*}) + \varepsilon_1) \\ &+ S_2\gamma\lambda_2^{-1}\dot{e}_2^{\gamma-1}(-(\tilde{u}_2^{1*} + \tilde{u}_2^{2*}) + \varepsilon_2) \\ &= S_1\gamma\lambda_1^{-1}\dot{e}_1^{\gamma-1}(-((\alpha_1^* t_{c1} + \beta_1^*\|S_1\|^{1/2})\mathrm{sign}(S)) + \varepsilon_1) \\ &+ S_2\gamma\lambda_2^{-1}\dot{e}_2^{\gamma-1}(-(\alpha_2^* t_{c2} + \beta_2^*\|S_2\|^{1/2})\mathrm{sign}(S) + \varepsilon_2) \end{aligned} \quad (40)$$

If we have:

$$\begin{aligned} &\alpha_1^* t_{c1} + \beta_1^*\|S_1\|^{1/2} \ge \varepsilon_1 \\ &\alpha_2^* t_{c2} + \beta_2^*\|S_2\|^{1/2} \ge \varepsilon_2 \end{aligned} \quad (41)$$

The time derivative of Lyapunov function is negative, consequently the stability of the controlled system is confirmed, and the adaptive RBFNNs parameters are given as:

$$\begin{aligned} \dot{W}_1^1 &= \gamma\delta_1^1\lambda_1^{-1}\dot{e}_1^{\gamma-1}S_1\|S_1\|^{1/2}\,h_1^1(S) \\ \dot{W}_1^2 &= \gamma\delta_1^2\lambda_1^{-1}\dot{e}_1^{\gamma-1}S_1h_1^2(S) \\ \dot{W}_2^1 &= \gamma\delta_2^1\lambda_2^{-1}\dot{e}_2^{\gamma-1}S_2\|S_2\|^{1/2}\,h_2^1(S) \\ \dot{W}_2^2 &= \gamma\delta_2^2\lambda_2^{-1}\dot{e}_2^{\gamma-1}S_2h_2^2(S) \end{aligned} \quad (42)$$