A Robust Adaptive Neural Network Controller Based on Variable Structure System

Nasser Sadati, Rasoul Ghadami, Mahdi Bagherpour
Intelligent Systems Laboratory, Electrical Engineering Department
Sharif University of Technology, Tehran, Iran.
sadati@sina.sharif.edu, ghadami@mehr.sharif.edu, bagherpour@ee.sharif.edu

***Abstract* - In this paper, a Robust adaptive neural network controller (RANNC) based on variable structure system for robotic manipulators is proposed to alleviate the problems met in practical implementation using classical variable structure controllers. The chattering phenomenon is eliminated by substituting single-input single-output radial-basis-function neural networks (RBFNN's), which are nonlinear and continuous, in lieu of the discontinuous part of the control signals present in classical forms. The weights of the hidden layer of the RBFNN's are updated in an online manner to compensate the system uncertainties. The key feature of this scheme is that prior knowledge of the system uncertainties is not required to guarantee the stability. Moreover, a theoretical proof of the stability and convergence of the proposed scheme using Lyapunov method is presented. To demonstrate the effectiveness of the proposed approach, a practical situation in robot control is simulated.**

Index Terms – Variable Structure Control, Adaptive Control, Lyapunov Stability, RBF Neural Network, Robotic Manipulator.

I. INTRODUCTION

Robotic manipulators are highly nonlinear, highly time-varying and highly coupled. Moreover, there always exists uncertainty in the system model such as external disturbances, parameter uncertainty, sensor errors and so on, which cause unstable performance in the robotic system. Almost all kinds of robust control schemes, including the classical variable structure control [2], have been proposed in the field of robotic control during the past decades. Classical variable structure controller design provides a systematic approach to the problem of maintaining stability in the face of modeling imprecision and uncertainty. This control scheme utilizes a high speed switching control law to drive the nonlinear predefined state trajectory onto a specified surface (called the sliding or switching surface), to attain the conventional goals of control such as stabilization and tracking.

Although classical variable structure control (VSC) is a powerful scheme for nonlinear systems with uncertainty, such as robotic manipulators [1], this control scheme has important drawbacks limiting its practical applicability, such as chattering and large control authority. Moreover, equivalent control computation requires exact knowledge of the system dynamics and parameters and, obviously, an approximate value can be achieved in partly known or uncertain systems. In addition, in order to guarantee the stability of the variable structure control systems, the boundary of the uncertainty has to be estimated. However, the estimate of the boundary is difficult to know, thus a conservative control law is selected. But this large conservative control signal causes more chattering and therefore it deteriorates the system performance.

Recently, a lot of research work has been done to use soft-computing methodologies such as artificial neural networks in order to improve the performance and alleviate the problem met in practical implementation of VSC's [3]. The use of an NN for the calculation of the equivalent term of an VSC is proposed in [4]. In [5], two NN's in parallel are used to realize the equivalent control and corrective control term of the VSC. This scheme is based on the fact that if the NN learns the equivalent control, the corrective term goes to zero and any different between them is reflected as a nonzero corrective term. In [6], by adaptively estimating the bound of system uncertainty using a multi-input single-output RBF neural network, the requirement for having prior knowledge of uncertainty is eliminated. However, there is still chattering in the control input. In [7], the gains of VSC are accepted as the weights of the NN and the weights are updated to minimize the defined cost function. The proposed adaptation scheme is MIT rule and there is no guarantee for convergence and stability.

In this paper, to control the robotic manipulator with robust characteristics, a new control scheme is developed, in such a way that the discontinuous part of the control signals in the classical variable structure controllers are substituted by single-input single-output RBF neural network functions to eliminate the chattering phenomenon. To relax the requirements for the knowledge of upper bound of the uncertainties, the weights of the hidden layer of RBF neural networks are updated in an online manner to compensate the system uncertainties and to guarantee the stability of the overall system without having any prior knowledge of the system uncertainties. The adaptive law is designed based on the Lyapunov method and mathematical proof for the stability and convergence of the overall system is provided.

The outline of this paper is as follows. Preliminaries about the model of the robotic manipulator, as partly known system, and the classical variable structure controller for robotic manipulators are summarized in section II. The robust adaptive neural network controller for robotic manipulators is presented in section III. The simulation results are given in section IV to demonstrate the effectiveness of the proposed control scheme. Finally, section V presents some concluding remarks.

II. PRELIMINARIES

A. Model of Robotic Manipulators

In the absence of friction or other disturbances, the dynamic equation of an n-link rigid robotic manipulator system is described by the following second order nonlinear vector differential equation

$$\boldsymbol{M(q)\ddot{q}} + \boldsymbol{B(q,\dot{q})\dot{q}} + \boldsymbol{G(q)} = \boldsymbol{u} \tag{1}$$

where $\boldsymbol{q} = [q_1,...,q_n]^T$ is an $n\times 1$ vector of joint angular position, $\dot{\boldsymbol{q}} = [\dot{q}_1,...,\dot{q}_n]^T$ and $\ddot{\boldsymbol{q}} = [\ddot{q}_1,...,\ddot{q}_n]^T$ are $n\times 1$ vectors of corresponding velocity and acceleration, $\boldsymbol{u}$ is an $n\times 1$ vector of applied joint torques (control inputs), $\boldsymbol{M(q)}$ is an $n\times n$ inertial matrix, $\boldsymbol{B(q,\dot{q})}$ is an $n\times n$ matrix of Coriolis and centrifugal forces and $\boldsymbol{G(q)}$ is an $n\times 1$ gravity vector.

The inertial matrix $\boldsymbol{M(q)}$ is symmetric and positive definite. It is also bounded as a function of $\boldsymbol{q}$: $\mu_1 I \le \boldsymbol{M(q)} \le \mu_2 I$. $\dot{\boldsymbol{M}}\boldsymbol{(q)} - 2\boldsymbol{B(q,\dot{q})}$ is skew symmetric matrix, that is, $x^T[\dot{\boldsymbol{M}}\boldsymbol{(q)} - 2\boldsymbol{B(q,\dot{q})}]x = 0$, where x is an $n\times 1$ nonzero vector.

It is assumed that a robotic manipulator, as is described by (1), has some known parts and some unknowns and therefore, there exist uncertainty in the system model. Thus, $\boldsymbol{M(q)}$, $\boldsymbol{B(q,\dot{q})}$ and $\boldsymbol{G(q)}$ can be described by

$$\begin{aligned}\boldsymbol{M(q)} &= \hat{\boldsymbol{M}}\boldsymbol{(q)} + \Delta\boldsymbol{M(q)}\\ \boldsymbol{B(q,\dot{q})} &= \hat{\boldsymbol{B}}\boldsymbol{(q,\dot{q})} + \Delta\boldsymbol{B(q,\dot{q})}\\ \boldsymbol{G(q)} &= \hat{\boldsymbol{G}}\boldsymbol{(q)} + \Delta\boldsymbol{G(q)}\end{aligned} \tag{2}$$

where $\hat{\boldsymbol{M}}\boldsymbol{(q)}$, $\hat{\boldsymbol{B}}\boldsymbol{(q,\dot{q})}$ and $\hat{\boldsymbol{G}}\boldsymbol{(q)}$ are the known parts, $\Delta\boldsymbol{M(q)}$, $\Delta\boldsymbol{B(q,\dot{q})}$ and $\Delta\boldsymbol{G(q)}$ are the unknown parts. For simplification in notation, we avoid writing the variables in the parentheses of the above matrices and vectors.

B. Designing A Classical Variable Structure Controller

In the design of variable structure controller for robotic manipulators, the control objective is to drive the joint position $\boldsymbol{q}$ to the desired position $\boldsymbol{q_d}$. So by defining the tracking error to be in the following form

$$\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q_d} \tag{3}$$

the sliding surface can be written as

$$\boldsymbol{s} = \dot{\boldsymbol{e}} + \boldsymbol{\lambda e} \tag{4}$$

where $\boldsymbol{\lambda} = diag[\lambda_1,...,\lambda_i,...,\lambda_n]$, in which λ_i is a positive constant. The control objective can now be achieved by choosing the control input so that the sliding surface satisfies the following sufficient condition

$$\frac{1}{2}\frac{d}{dt}s_i^{\,2} \le -\eta_i |s_i| \tag{5}$$

where η_i is a positive constant. Equation (5) indicates that the energy of $\boldsymbol{s}$ should decay as long as $\boldsymbol{s}$ is not zero. To set up the control $\boldsymbol{u}$, define the reference states to be

$$\begin{aligned}\dot{\boldsymbol{q}}_{\boldsymbol{r}} &= \dot{\boldsymbol{q}} - \boldsymbol{s} = \dot{\boldsymbol{q}}_{\boldsymbol{d}} - \boldsymbol{\lambda e}\\ \ddot{\boldsymbol{q}}_{\boldsymbol{r}} &= \ddot{\boldsymbol{q}} - \dot{\boldsymbol{s}} = \ddot{\boldsymbol{q}}_{\boldsymbol{d}} - \boldsymbol{\lambda}\dot{\boldsymbol{e}}\end{aligned} \tag{6}$$

Now the control input $\boldsymbol{u}$ can be chosen to be in the following form

$$\begin{aligned}\boldsymbol{u} &= \hat{\boldsymbol{u}} - \boldsymbol{K}\,\mathrm{sgn}(\boldsymbol{s})\\ \hat{\boldsymbol{u}} &= \hat{\boldsymbol{M}}\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \hat{\boldsymbol{B}}\dot{\boldsymbol{q}}_{\boldsymbol{r}} + \hat{\boldsymbol{G}} - \boldsymbol{As}\end{aligned} \tag{7}$$

where $\boldsymbol{K} = diag[k_{11},...,k_{ii},...,k_{nn}]$ is a diagonal positive definite matrix in which k_{ii}'s are positive constants and $\boldsymbol{A} = diag[a_1,...,a_i,...,a_n]$ is a diagonal positive definite matrix in which a_i's are also positive constants. Putting (7) into (1) leads to

$$\boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{(B + A)s} = \boldsymbol{\Delta f} - \boldsymbol{K}\,\mathrm{sgn}(\boldsymbol{s}) \tag{8}$$

where $\boldsymbol{\Delta f} = \boldsymbol{\Delta M}\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{\Delta B}\dot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{\Delta G}$. It can be proved that by choosing $\boldsymbol{K}$ such that

$$k_{ii} \ge |\Delta f_i|_{bound} \tag{9}$$

where $|\Delta f_i|_{bound}$ is the boundary of $|\Delta f_i|$, the overall system is asymptotically stable.

Proof: Consider V in equation (10) as the Lyapunov function candidate

$$V = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{Ms} \tag{10}$$

Since $\boldsymbol{M}$ is symmetric and positive definite, then for $\boldsymbol{s} \ne 0$, $V > 0$. Now taking the derivative of V with respect to $\boldsymbol{s}$, one can obtain

$$\begin{aligned}\dot{V} &= \boldsymbol{s}^{\boldsymbol{T}}[-\boldsymbol{(B + A)} + \boldsymbol{\Delta f} - \boldsymbol{K}\,\mathrm{sgn}\boldsymbol{(s)} + \boldsymbol{Bs}]\\ &= \sum_{i=1}^{n}(s_i[\Delta f_i - k_{ii}\,\mathrm{sgn}(s_i)]) - \boldsymbol{s}^T\boldsymbol{As}\end{aligned} \tag{11}$$

Using (9), when $s_i > 0$,

$$\Delta f_i - k_{ii}\,\mathrm{sgn}(s_i) = \Delta f_i - k_{ii} \le 0$$

and when $s_i < 0$

$$\Delta f_i - k_{ii}\,\mathrm{sgn}(s_i) = \Delta f_i + k_{ii} \ge 0$$

so that

$$s_i[\Delta f_i - k_{ii}\,\mathrm{sgn}(s_i)] \le 0$$

Therefore

$$\sum_{i=1}^{n}(s_i[\Delta f_i - k_{ii}\,\mathrm{sgn}(s_i)]) \le 0$$

Since $\boldsymbol{A}$ is a positive definite matrix, $-\boldsymbol{s}^T\boldsymbol{As} \le 0$. With this result, it can be proved that

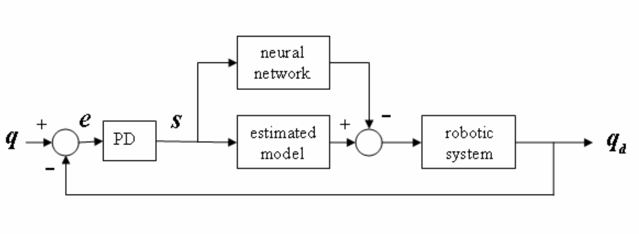

Fig. 1. Robust Neural network control (RNNC) of robotic manipulators.

$$\dot{V} = \sum_{i=1}^{n} (s_i[\Delta f_i - k_{ii}\,\mathrm{sgn}(s_i)]) - \boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} \le -\boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} \le 0 \quad (12)$$

The Lyapunov function candidate in (10) can be considered as an indicator of the energy of $\boldsymbol{s}$. As $\dot{V} \le 0$,and $\dot{V} = 0$ only when $\boldsymbol{s} = 0$. Therefore the decay of the energy of $\boldsymbol{s}$, as long as $\boldsymbol{s} \neq 0$, is guaranteed and the sufficient condition in (5) is satisfied. So the overall system is asymptotically stable.

III. Robust Adaptive Neural Network Control

There are two major disadvantages in designing the classical variable structure controllers. First, because of the control actions which are discontinuous across $\boldsymbol{s}$, there is chattering in a boundary of the surface $\boldsymbol{s}$. Such high frequency switching (chattering) might excite unmodeled dynamics and impose undue wear and tear on the actuators, so the control law would not be considered acceptable. Second, the prior knowledge of the boundary of uncertainty is required in compensators. If boundary is unknown, a large value has to be applied to the gain of discontinuous part of control signal and this large control gain may intensify the chattering on the sliding surface.

In this section, a Robust adaptive neural network controller (RANNC), to avoid the aforementioned problems, has been proposed. For this purpose single-input single-output (SISO) RBF neural networks, as continuous approximation of the elements of $\boldsymbol{K}\,\mathrm{sgn}(\boldsymbol{s})$ in the control law (7) are used. The control input is written as

$$\boldsymbol{u} = \hat{\boldsymbol{M}}\ddot{\boldsymbol{q}}_r + \hat{\boldsymbol{B}}\dot{\boldsymbol{q}}_r + \hat{\boldsymbol{G}} - \boldsymbol{A}\boldsymbol{s} - \boldsymbol{K} \quad (13)$$

where $\boldsymbol{K} = [k_1,...,k_i,...,k_n]^T$ is an $n\times1$ vector in which k_i is the output of the ith RBF neural network. The control block diagram of the RNNC is shown in Fig. 1, where the PD block indicates the computation of $d/dt + \lambda$ and the input of each RBF network is s_i and the corresponding output is k_i . The applied RBF neural networks have the following structure

$$k_i = \boldsymbol{W}_{k_i}{}^T \Phi_{k_i}(s_i) \quad (14)$$

where $\boldsymbol{W}_{k_i}$ is the $m\times1$ vector of the output layer weights and m is the number of nodes in the hidden layer and $\Phi_{k_i}(s_i) = [\varphi_{k_i}^1(s_i),...,\varphi_{k_i}^j(s_i),...,\varphi_{k_i}^m(s_i)]^T$ is the $m\times1$ vector

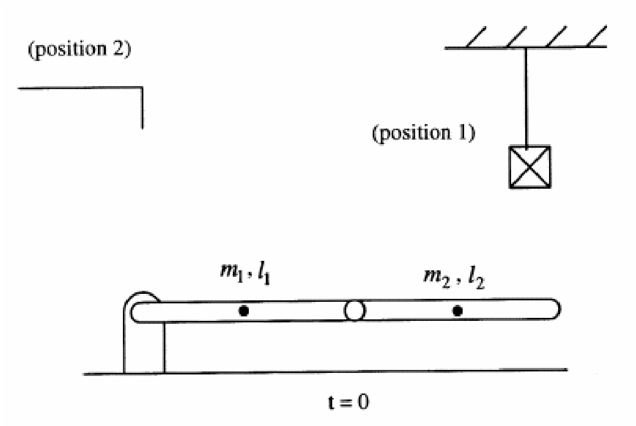

Fig. 2. Illustration of the situation depicted in the simulation.

of outputs of the hidden layer nodes, whose elements (basis functions) are chosen as Gaussian-type function, expressed by

$$\varphi_{k_i}^j(s_i) = \alpha_i \exp(-\left\|s_i - \mu_i^j\right\|^2 / 2\sigma_i^{j\,2}) \quad (15)$$

where μ_i^j and σ_i^j are the center and variance of the jth basis function of the ith RBFN, respectively and α_i is a positive constant.

In continue, an adaptive law is designed to guarantee that k_i 's can compensate the system uncertainties. $\boldsymbol{W}_{k_i}$ is chosen as the parameter to be updated and therefore is called the parameter vector, and $\Phi_{k_i}(s_i)$ is called the basis function vector which can be regarded as the weight of the parameter vector. Substituting (13) into (1) leads to

$$\boldsymbol{M}\dot{\boldsymbol{s}} = -(\boldsymbol{B} + \boldsymbol{A})\boldsymbol{s} + \boldsymbol{\Delta f} - \boldsymbol{K} \quad (16)$$

where the definition of $\boldsymbol{\Delta f}$ is the same as that in part A of section I. It has been proven ([8]) that the RBF neural networks are universal approximators if their basis functions are chosen as a scaled version of Gaussian functions, which means that these neural networks are capable of approximating any real continuous function on a compact set to arbitrary accuracy.

Defining $\boldsymbol{W}_{k_{id}}$ so that $k_i = \boldsymbol{W}_{k_{id}}{}^T \Phi_{k_i}(s_i)$ is the optimal compensation for Δf_i . According to the property of universal approximation of RBF neural networks, there exists $\delta_i > 0$ satisfying

$$\left|\Delta f_i - \boldsymbol{W}_{k_{id}}{}^T \Phi_{k_i}(s_i)\right| \le \delta_i \quad (17)$$

where δ_i is arbitrary and can be chosen as small as possible. Defining

$$\tilde{\boldsymbol{W}}_{k_i} = \boldsymbol{W}_{k_i} - \boldsymbol{W}_{k_{id}} \quad (18)$$

It can be shown that by choosing the adaptive law as

$$\dot{\tilde{\boldsymbol{W}}}_{k_i} = s_i \Phi_{k_i}(s_i) \quad (19)$$

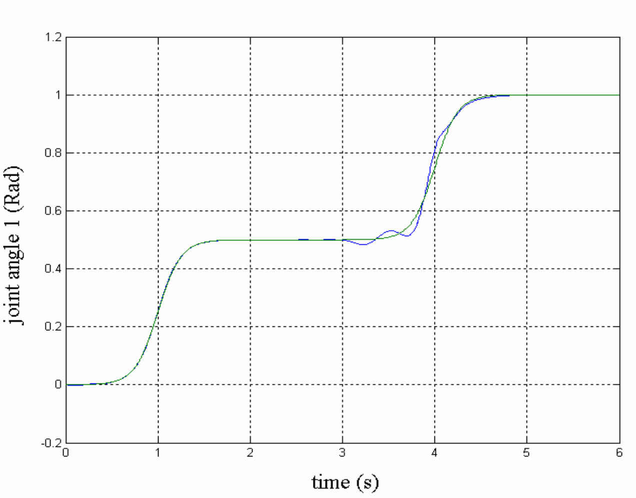

Fig. 3. Tracking of joint 1 in RANNC.

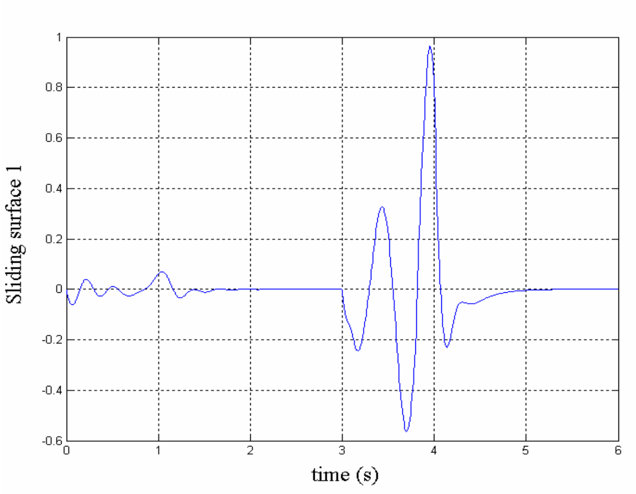

Fig. 5. Sliding surface of joint 1 in RANNC.

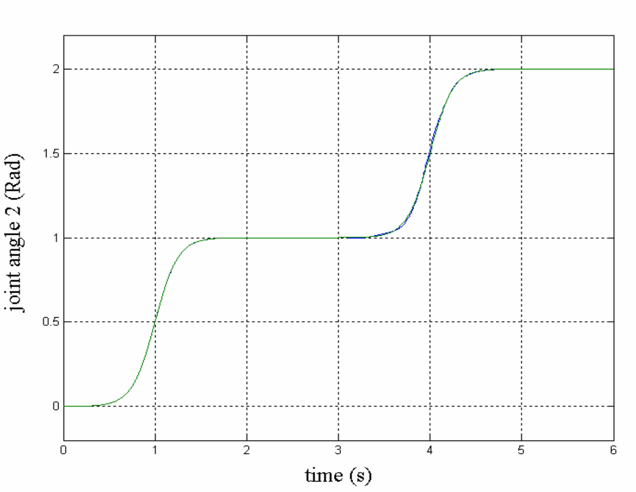

Fig. 4. Tracking of joint 2 in RANNC.

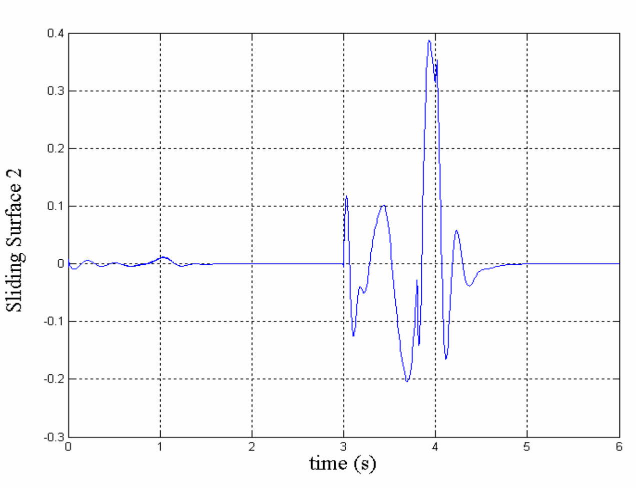

Fig. 6. Sliding surface of joint 2 in RANNC.

the overall system is asymptotically stable with respect to $\boldsymbol{s}$ and the actual joint angular positions converge to the desired ones.

Proof: Choose the Lyapunov candidate as

$$V = \frac{1}{2}\boldsymbol{s}^T \boldsymbol{M}\boldsymbol{s} + \frac{1}{2}\sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T \tilde{\boldsymbol{W}}_{k_i} \qquad (20)$$

where $\boldsymbol{M}$ is symmetric positive matrix and $\tilde{\boldsymbol{W}}_{k_i}{}^T \tilde{\boldsymbol{W}}_{k_i} > 0$, therefore V is positive definite. Now Consider the derivative of V

$$\dot{V} = \frac{1}{2}[\dot{\boldsymbol{s}}^T \boldsymbol{M}\boldsymbol{s} + \boldsymbol{s}^T \dot{\boldsymbol{M}}\boldsymbol{s} + \boldsymbol{s}^T \boldsymbol{M}\dot{\boldsymbol{s}}] + \frac{1}{2}\sum_{i=1}^{n}[\dot{\tilde{\boldsymbol{W}}}_{k_i}{}^T \tilde{\boldsymbol{W}}_{k_i} + \tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}]$$

$$= \frac{1}{2}[\boldsymbol{s}^T \dot{\boldsymbol{M}}\boldsymbol{s} + 2\boldsymbol{s}^T \boldsymbol{M}\dot{\boldsymbol{s}}] + \frac{1}{2}\sum_{i=1}^{n} 2\tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}$$

$$= \boldsymbol{s}^T[\boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{B}\boldsymbol{s}] + \sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}$$

$$= \boldsymbol{s}^T[-(\boldsymbol{B}+\boldsymbol{A})\boldsymbol{s} + \boldsymbol{\Delta f} - \boldsymbol{K} + \boldsymbol{B}\boldsymbol{s}] + \sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}$$

$$= \boldsymbol{s}^T[-\boldsymbol{A}\boldsymbol{s} + \boldsymbol{\Delta f} - \boldsymbol{K}] + \sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}$$

$$= -\boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} + \boldsymbol{s}^T[\boldsymbol{\Delta f} - \boldsymbol{K}] + \sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}$$

$$= -\boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} + \sum_{i=1}^{n}(s_i[\Delta f_i - k_i]) + \sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}$$

Since $k_i = \tilde{\boldsymbol{W}}_{k_i}{}^T \Phi_{k_i}(s_i) + \boldsymbol{W}_{k_{id}}{}^T \Phi_{k_i}(s_i)$, then

$$\dot{V} = -\boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} + \sum_{i=1}^{n}(s_i[\Delta f_i - (\tilde{\boldsymbol{W}}_{k_i}{}^T \Phi_{k_i}(s_i) + \boldsymbol{W}_{k_{id}}{}^T \Phi_{k_i}(s_i))])$$

$$+ \sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}$$

$$= -\boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} + \sum_{i=1}^{n}(s_i[\Delta f_i - \boldsymbol{W}_{k_{id}}{}^T \Phi_{k_i}(s_i)])$$

$$+ \sum_{i=1}^{n}(-s_i \tilde{\boldsymbol{W}}_{k_i}{}^T \Phi_{k_i}(s_i)) + \sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T \dot{\tilde{\boldsymbol{W}}}_{k_i}$$

$$= -\boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} + \sum_{i=1}^{n}(s_i[\Delta f_i - \boldsymbol{W}_{k_{id}}{}^T \Phi_{k_i}(s_i)])$$

$$+ \sum_{i=1}^{n} \tilde{\boldsymbol{W}}_{k_i}{}^T[-s_i \Phi_{k_i}(s_i)) + \dot{\tilde{\boldsymbol{W}}}_{k_i}]$$

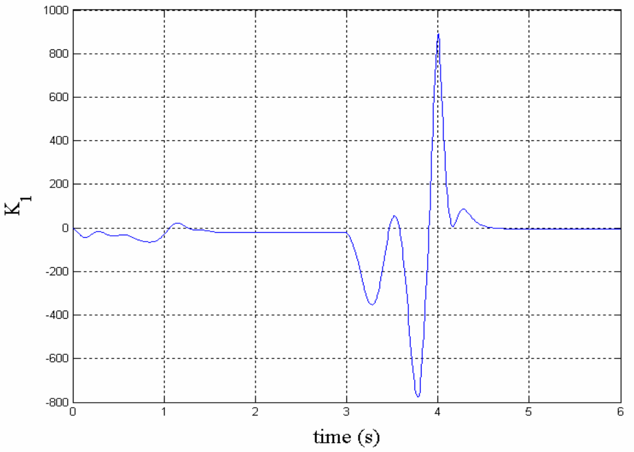

Fig. 7. Control gain of joint 1 in RANNC.

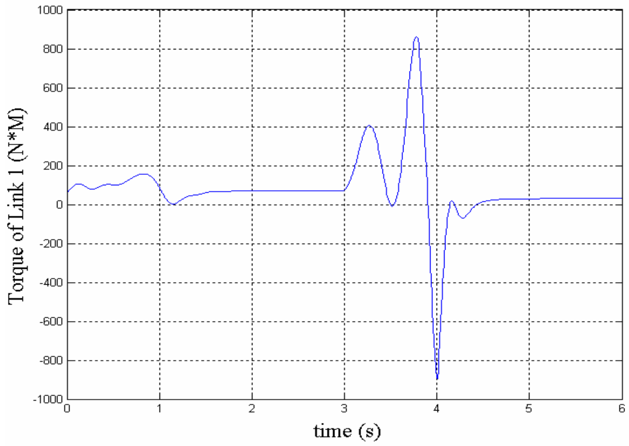

Fig. 9. Control torque of joint 1 in RANNC.

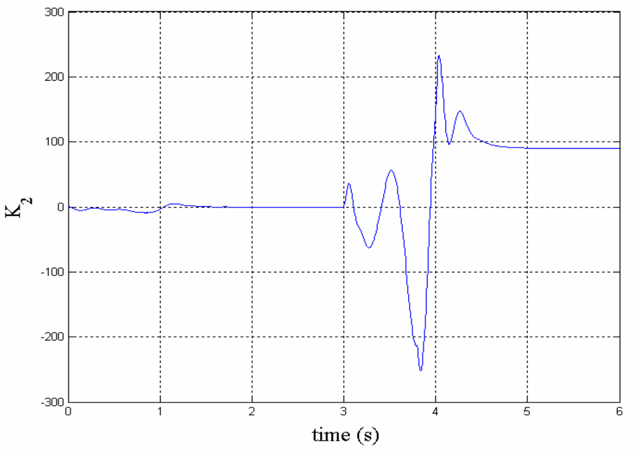

Fig. 8. Control gain of joint 2 in RANNC.

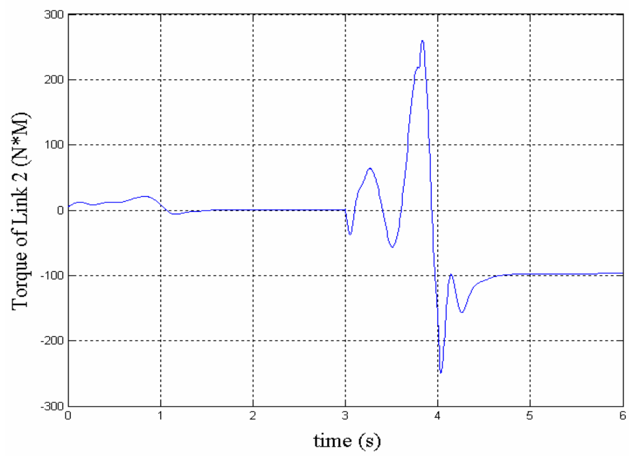

Fig. 10. Control torque of joint 2 in RANNC.

Moreover, since the adaptive law in (19) is chosen as

$$\dot{\tilde{\boldsymbol{W}}}_{k_i} = s_i \Phi_{k_i}(s_i)$$

then

$$\dot{V} = -\boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} + \sum_{i=1}^{n} (s_i[\Delta f_i - \boldsymbol{W}_{k_{id}}{}^T \Phi_{k_i}(s_i)]) \qquad (21)$$

From (17), there exist

$$\left|\Delta f_i - \boldsymbol{W}_{k_i}{}^T \Phi_{k_i}(s_i)\right| \le \delta_i$$

where δ_i can be chosen as small as possible. Now by assuming

$$\left|\Delta f_i - \boldsymbol{W}_{k_i}{}^T \Phi_{k_i}(s_i)\right| \le \delta_i \le \rho_i |s_i|$$

where $0 < \rho_i < 1$, the second term at the right side of (21) satisfies

$$s_i[\Delta f_i - \boldsymbol{W}_{k_i}{}^T \Phi_{k_i}(s_i)] \le \rho_i |s_i|^2 = \rho_i s_i{}^2$$

Therefore

$$\dot{V} \le -\boldsymbol{s}^T \boldsymbol{A}\boldsymbol{s} + \sum_{i=1}^{n} \rho_i s_i{}^2$$

$$\le \sum_{i=1}^{n} (-a_i s_i{}^2 + \rho_i s_i{}^2)$$

Now by simply choosing $a_i > \rho_i$

$$\dot{V} \le \sum_{i=1}^{n} (\rho_i - a_i) s_i{}^2 \le 0 \qquad (22)$$

In (22) since $(\rho_i - a_i) < 0$, $\dot{V} = 0$ only when $s_i = 0$. Thus, the overall system with the adaptive law in (19) is asymptotically stable with respect to $\boldsymbol{s}$. In other words

$$\lim_{t\to\infty} \boldsymbol{s} = \lim_{t\to\infty} (\dot{\boldsymbol{e}} + \boldsymbol{\lambda}\boldsymbol{e}) = 0 \qquad (23)$$

or equivalently

$$\lim_{t\to\infty} \boldsymbol{e} = 0 \Rightarrow \lim_{t\to\infty} \boldsymbol{q} = \boldsymbol{q}_d$$
$$\lim_{t\to\infty} \dot{\boldsymbol{e}} = 0 \Rightarrow \lim_{t\to\infty} \dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_d \qquad (24)$$

Therefore, it is proved that the robust adaptive neural network control input (13) drives the actual joint positions to their desired values.

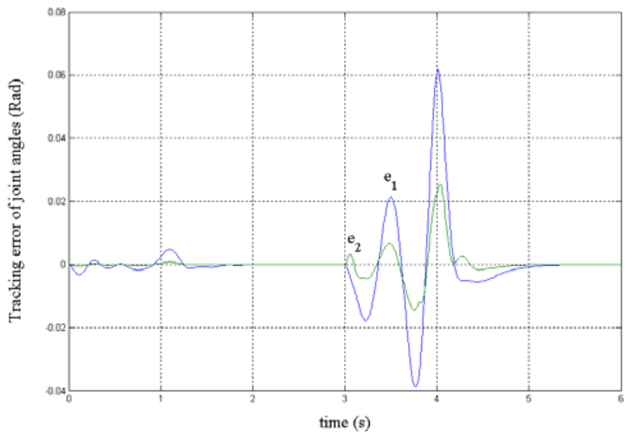

Fig. 11. Tracking error of joint angles in RANNC.

IV. Simulation Results

In this section, the proposed Robust adaptive neural network controller (RANNC) is used on a two-link robotic manipulator, whose parameter matrices are as follows [9]

$$\boldsymbol{M(q)} = \begin{bmatrix} (\frac{1}{4}m_1 + m_2)l_1^{\,2} + \frac{1}{4}m_2 l_2^{\,2} + m_2 l_1 l_2 \cos q_2 & \frac{1}{4}m_2 l_2^{\,2} + \frac{1}{2}m_2 l_1 l_2 \cos q_2 \\ \frac{1}{4}m_2 l_2^{\,2} + \frac{1}{2}m_2 l_1 l_2 \cos q_2 & \frac{1}{4}m_2 l_2^{\,2} \end{bmatrix}$$

$$\boldsymbol{B(q,\dot{q})} = \begin{bmatrix} -m_2 l_1 l_2 \dot{q}_2 \sin q_2 & -\frac{1}{2}m_2 l_1 l_2 \dot{q}_2 \sin q_2 \\ \frac{1}{2}m_2 l_1 l_2 \dot{q}_1 \sin q_2 & 0 \end{bmatrix}$$

$$\boldsymbol{G(q)} = \begin{bmatrix} (\frac{1}{2}m_1 + m_2) g l_1 \cos q_1 + \frac{1}{2}m_2 g l_2 \cos(q_1 + q_2) \\ \frac{1}{2}m_2 g l_2 \cos(q_1 + q_2) \end{bmatrix}$$

where m_1 and m_2 are the masses, and l_1 and l_2 are the length of the links 1 and 2, respectively. They are chosen as $m_1 = 4$, $m_2 = 2$, $l_1 = 2$, $l_2 = 1$. The situation that is simulated is shown in Fig. 2. In this example, the manipulator is expected to take a load from position one to position two. In the first stage, the manipulator moves from the initial position to the position 1 along a predefined trajectory during 2 s. It stays there for 1 s to take the load and start to move from position one to position two at t=3 s. During the second stage, a disturbance is added to link 1 at t=3.8 s and disappears at t=4 s. From the above description, there are totally three dynamic changes in the whole process because of the added load, disturbance and coming back to be normal when the disturbance disappears. In this simulation, the system model for the control input is estimated by applying a factor to the corresponding parameter matrices of the original system, i.e., $\hat{\boldsymbol{G}} = 0.7\boldsymbol{G}$, $\hat{\boldsymbol{B}} = 0.8\boldsymbol{B}$, $\hat{M}_{11} = 0.9M_{11}$, $\hat{M}_{12} = 0.9M_{12}$, $\hat{M}_{21} = 0.9M_{21}$, $\hat{M}_{22} = M_{22}$. The control input $\boldsymbol{u}$ is chosen as in (13), where $\boldsymbol{A} = diag[50,50]$, $\boldsymbol{\lambda} = diag[10,10]$ and each element of $\boldsymbol{K}$ is constructed by an RBF neural network with 10 nodes in its hidden layer. The initial parameters of each RBF neural network are evaluated by gradient descent algorithm to approximate a continuous quasi-signum function. Also the weights of RBF neural networks are updated during on-line control with the adaptive law given by (19).

The simulation results are shown in Fig. 3 - Fig. 11. As is seen in Fig. 3 and Fig. 4, the joint angles track the desired trajectories and the RANNC drive the robotic manipulator to its desired positions. Moreover, there is no chattering in the sliding surface as shown in Fig. 5 and 6 and the values of $\boldsymbol{K}$ converge to constant in the steady-state as shown in Fig. 7 and 8. In Fig. 9 and 10, it is also shown that there is no chattering in the control torques of RANNC. Finally, Fig. 11 shows that the tracking errors converge to zero.

V. Conclusion

In this paper a robust adaptive neural network controller based on variable structure system is proposed for robotic manipulators. The discontinuous parts of the classical variable structure controller are replaced by RBF neural networks, which are nonlinear and continuous, to avoid the chattering. The weights of the output layer of RBF neural networks are updated in an online manner to compensate the system uncertainties and to guarantee the system stability without any prior knowledge of the system uncertainties. In addition, the RBF networks used in the controller are SISO systems. Therefore the learning process is simple compared to that of multi-input systems developed previously. Also the design and implementation of the controller is simplified. In addition, the stability and convergence of the overall system are proved by the lyapunov method. Finally, simulation results situation are provided to show the effectiveness of the proposed scheme.

References

[1] J. E. Slotine, W. Li, *Applied Nonlinear Control*. Prentice-Hall Inc. New Jersy, 1991.

[2] K. D. Yong, "Controller design for a manipulator using theory of variable structure system," *IEEE Trans. Systs., Man, Cybern*., vol SMC-8, no .2, pp. 210-218, Feb. 1978.

[3] O. Kaynak, K. Erbartur, M. Ertugrul. "The Fusion of computationally intelligent methodologies and sliding mode control-a survey," *IEEE Trans. on Industrial Electronics*, vol. 48, no. 1, pp. 4-17, Feb. 2001.

[4] K. Zerenik, M. Rodic, R. Safaric, B. Curk, "Neural network sliding mode robot control," *Robotica*, vol. 15, pp. 23-30, 1997.

[5] M. Ertugrul, O. Kaynak, "Neuro sliding mode control of robotic manipulators," *Mechatron*, vol. 10, no 1-2, pp. 243-267, 2000.

[6] Man Zihang, X. H. Yo, K.Eshraghian, M. Panasiwami. "A robust adaptive sliding mode tracking control using an RBF neural network for robotic manipulator," in Proc. *IEEE International Conference on Neural Networks*, vol. 5,.1995, pp. 2403-2408.

[7] M. Ertugrul, O. Kaynak, "Neural network adaptive sliding mode control and its application to SCARA type robot manipulator," in Proc. *IEEE ICRA'97*, 1997, pp. 2932-2937.

[8] J. -S. Roger Jang, C. –T. Sun, "Functional equivalence between radial basis function networks and fuzzy systems," *IEEE Trans. on Neural Networks*. vol. 4, no. 1, pp. 156-159, Jan. 1993.

[9] J. J. Craig, *Introduction to Robotic*. Reading, MA: Addison-Wesley, 1989.