

# Robust Adaptive Control of Robot Manipulator Using General Regression Neural Network

*Abstract:* In this paper, a robust adaptive neural network sliding mode controller for robotic manipulators with uncertain load is presented. The proposed approach remedies the previous problems met in practical implementation of classical sliding mode controllers. An adaptive General Regression Neural Network (GRNN) is used to calculate each element of the control gain vector; discontinuous part of control signal, in a classical sliding mode controller. The key feature of this scheme is that prior knowledge of the system uncertainties is not required to guarantee the stability. Also the chattering phenomenon is completely eliminated. To demonstrate the effectiveness of the proposed approach, a three link SCARA robot is simulated in the presence of uncertainties.

*Index Terms:* Robust, adaptive, neural network, sliding mode, control, uncertain, robot manipulator.

**Jyoti Ohri**

National Institute of Technology,  
Kurukshetra,  
(e-mail: ohrijyoti@rediffmail.com).

**Sourav Dutta**

M Tech. Student, Department of  
Electrical Engineering,  
National Institute of Technology,  
Kurukshetra  
(e-mail: souravready@gmail.com).

## 1. INTRODUCTION

The problem of motion control for robotic manipulators has attracted the attention of many researchers over the past decade. Two basic facts about the robot manipulator dynamics make the control problem a challenging one. First, the dynamics are described by a set of second order, nonlinear, and coupled differential equations. Second, the parameters of the model are partially unknown, due to errors in modeling, and varying payload [1]. The typical structure of a robust controller is composed of a nominal part, similar to a feedback linearization or inverse control law, and additional terms aimed at dealing with uncertainties. Almost all kinds of robust control schemes, including the classical sliding mode control [2], have been proposed in the field of robotic control during the past decades. Classical sliding mode controller design provides a systematic approach to the problem of maintaining stability in the face of modeling imprecision and uncertainty. Although classical sliding mode control is a powerful scheme for nonlinear systems with uncertainty, such as robotic manipulators [1,3], this control scheme has important drawbacks limiting its practical applicability, such as chattering and large control authority. Moreover, in order to guarantee the stability of the sliding mode control systems, the boundary of the uncertainty has to be estimated. Recently, much research works have been done to use soft-computing methodologies such as artificial neural networks in order to improve the performance and

remedy the problems met in practical implementation of sliding mode controllers [6].

The use of neural network (NN) for calculation of the equivalent term of a sliding mode controller (SMC) is proposed in [7]. In [8] two NNs in parallel are used to realize the equivalent control and corrective control terms of an SMC. This scheme is based on the fact that if the NN learns the equivalent control, the corrective term goes to zero and any difference between them is reflected as a nonzero corrective term. In [9] the gains of an SMC 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, the combination of neural network and sliding mode control are used for controlling the robotic manipulator with robust characteristics. The discontinuous part of the control signals in the classical sliding mode controllers are substituted by the output of General Regression Neural Network (GRNN), which are nonlinear and continuous, to eliminate the chattering phenomenon.

## 2. PRELIMINARIES

### A. Model of robotic manipulators

The dynamic equation of an n-link rigid robotic manipulator system can be described by the following second-order nonlinear vector differential equation [10]

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) - u(t) \quad (1)$$

Where  $q, \dot{q}, \ddot{q} \in R^n$  are joint position, velocity and acceleration vectors respectively,  $M(q) \in R^{n \times n}$  denotes the inertia matrix,  $C(q, \dot{q}) \in R^{n \times n}$  expresses the coriolis and centrifugal torques,  $F(q, \dot{q}) \in R^{n \times 1}$  is the unstructured uncertainties of the dynamics including friction and other disturbances,  $G(q) \in R^n$  is the gravity vector and  $u(t) \in R^{n \times 1}$  is the actuator torque vector acting on joints.

It is assumed that a robotic manipulator, as described in (1), has some known and some unknown parts. These unknown parts include uncertainty in robot parameters due to the unknown load.

Thus  $M(q), C(q, \dot{q})$  and  $G(q)$  respectively, can be described as

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

where  $\hat{M}(q), \hat{C}(q, \dot{q}), \hat{G}(q)$  are the known parts or the estimated parameters and  $\Delta M(q), \Delta C(q, \dot{q}), \Delta G(q)$  are the unknown parts. For the simplification of notations, from now on we avoid writing the variables in the parentheses of the above matrices and vectors.

### B. Classical sliding mode controller (SMC)

In the design of SMC for a robotic manipulator, the control objective is to drive the joint position  $q$  to the desired position  $q_d$ . So by defining the tracking error to be in the following form:

$$e = q - q_d \quad (3)$$

The sliding surface can be written as

$$s = \dot{e} + \lambda e \quad (4)$$

Where  $\lambda = \text{diag}[\lambda_1, \dots, \lambda_n]$  and  $\lambda_i$  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:

$$\frac{1}{2} \frac{ds_i^2}{dt} \leq -\eta_i |s_i|$$

where  $\eta_i$  is a positive constant. Eq. (5) indicates that the energy of  $s$  should decay as long as  $s$  is not zero. Now to set up the control input  $u$ , we can define the reference states to be in the following forms:

$$\begin{aligned} \dot{q}_r &= \dot{q} - s = \dot{q}_d - \lambda e \\ \text{and} \\ \ddot{q}_r &= \ddot{q} - \dot{s} = \ddot{q}_d - \lambda \dot{e} \end{aligned} \quad (6)$$

The control input  $u$  can now be chosen as

$$u = \hat{u} - As - K \text{sgn}(s) \quad (7)$$

where  $\hat{u} = \hat{M}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{G}$  and  $K = \text{diag}[k_{11}, \dots, k_{11}, \dots, k_{mm}]$  is a diagonal positive definite matrix in which  $k_{ii}$  s are positive constants and  $A = \text{diag}[a_1, \dots, a_1, \dots, a_n]$  is a diagonal positive definite matrix in which  $a_i$  s are also positive constants. Now, substituting (7) into (1) yields

$$Ms + (C + A)s = \Delta f - K \text{sgn}(s) \quad (8)$$

where  $\Delta f = \Delta M\ddot{q}_r + \Delta C\dot{q}_r + \Delta G + F$ . It has been proven in [2] that by considering the Lyapunov function candidate as

$$V = \frac{1}{2} s^T Ms \quad (9)$$

And choosing  $K$  such that

$$k_{ii} \geq |\Delta f_i|_{\text{bound}} \quad (10)$$

where  $|\Delta f_i|_{\text{bound}}$  is the boundary of  $|\Delta f_i|$ , the overall system is asymptotically stable. Therefore, the decay of the energy of  $s$ , as long as  $s \neq 0$ , is guaranteed and the sufficient condition in (5) is satisfied.

### 3. ADAPTIVE SLIDING MODE CONTROL USING GRNN

There are major disadvantages in designing the classical SMCs. First, because of the control actions which are discontinuous across  $s$ , there is chattering in a boundary of the surface  $s$ . 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 the following section, an adaptive SMC using soft computing, to avoid the aforementioned problems, has been proposed. General regression neural network (GRNN) is applied to construct the control gain.

Since the chattering is caused by the constant value of  $K$  and the discontinuous function  $\text{sgn}(s)$ , let the control gain  $K \text{sgn}(s)$  be replaced by a gain  $K$  which is constructed by GRNN, as described in the following sections. The new control input is then can be written as

$$u = \hat{u} - As - K \quad (11)$$

where  $K = [k_1, \dots, k_i, \dots, k_n]^T$  is an  $n \times 1$  vector in which  $k_i$  is the output of the GRNN.

#### 4. COMPENSATION OF UNCERTAINTIES USING GENERAL REGRESSION NEURAL NETWORK (GRNN)

The GRNN paradigm has been proposed [11] as an alternative to the popular back-propagation training algorithm for feedforward neural networks. It is closely related to the probabilistic neural network [12]. Regression can be thought of as the least-mean-squares estimation of the value of a variable based on available data. The GRNN is based on the estimation of a probability density function from observed samples using Parzen window estimation [13]. It utilizes a probabilistic model between the independent vector random variable  $X$  with dimension  $D$ , and dependent scalar random variable  $Y$ . Assume that  $x$  and  $y$  are the measured values for  $X$  and  $Y$  variables, respectively. If  $f(X, Y)$  represents the known joint continuous probability density function, and if  $f$  is known, the expected value of  $Y$  given  $x$  (the regression of  $Y$  on  $x$ ) can be estimated as

$$E[Y | x] = \frac{\int_{-\infty}^{\infty} Y f(x, Y) dY}{\int_{-\infty}^{\infty} f(x, Y) dY} \quad (12)$$

Based on  $p$  sample observations that are available, i.e., on the training set given by  $x$  and  $y$ , further assuming that the underlying density is continuous and the first partial derivatives of the function evaluated at any  $x$  are small, the probability estimator  $\hat{f}(x, y)$  can be written as

$$\hat{f}(x, y) = \frac{1}{(2\pi)^{D+1/2} \sigma^{D+1}} \frac{1}{p} \times \sum_{i=1}^p \left[ \exp\left(-\frac{(x-x_i)^T (x-x_i)}{2\sigma^2}\right) \exp\left(-\frac{(y-y_i)^2}{2\sigma^2}\right) \right] \quad (13)$$

where  $x_i$  and  $y_i$  are the  $i$ th training set data, and  $x_i$  denotes the vector form of variable  $x$ . A physical interpretation of the probability estimate  $\hat{f}(x, y)$  is that it assigns a sample probability of width  $\sigma$  for sample  $x_i$  and  $y_i$ , after that, the probability estimate is the sum of those sample probabilities. Substituting Equation (13) into Equation (12), the desired conditional mean of  $Y$  given  $x$ ,  $\hat{y}$ , can be calculated as

$$\hat{y}(x) = E[Y | x] = \frac{\sum_{i=1}^n [y_i \exp(d_i)]}{\sum_{i=1}^n \exp(d_i)} \quad (14)$$

where  $d_i$  is given by the distance function of the input space.

Now let us consider each element of the vector  $K$  namely  $k_i$ , to be estimated by an individual GRNN. If the weighted average approach is used to construct the output of GRNN, then each can be written as

$$\hat{k}_i = \frac{\sum_{j=1}^m [k_j \exp(d_j)]}{\sum_{j=1}^m \exp(d_j)} \quad (15)$$

where  $d_j$ , the distance function and here can be written as

$$d_j = \left[ -\left(\frac{s-s_j}{\sigma}\right)^2 \right] \quad (16)$$

In the above expression  $s$  is the new input and  $s_j$  is the stored input,  $\sigma$  is the spread factor. In equation (15)  $k_j$  is the stored output corresponding to  $s_j$  and  $\hat{k}_j$  implies the estimated value of true  $k_j$ .

In continuation, an adaptive law is designed to guarantee that can compensate the system uncertainties. According to the property of universal approximation, there exists such that

$$|\Delta f_j - \hat{k}_j| < \delta_j \quad (17)$$

where  $\hat{k}_j$  is the output of the GRNN and  $\delta_j$  can be chosen as small as possible. A good estimation of  $k_j$  depends on the selection of spread factor  $\sigma$ .

## 5. SIMULATION

In this section, the proposed adaptive SMC is used on a three-link SCARA robot, with parameter matrices given by [14].

$$M(q) = \begin{pmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & M_{33} \end{pmatrix}, \quad C(q, \dot{q}) = \begin{pmatrix} C_{11} & C_{12} & 0 \\ C_{21} & C_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$G(q) = \begin{bmatrix} 0 \\ 0 \\ G_3 \end{bmatrix}$$

where

$$M_{11} = l_1^2 \left( \frac{m_1}{3} + m_2 + m_3 \right) + l_1 l_2 (m_2 + 2m_3) \cos(q_2) + l_2^2 \left( \frac{m_2}{3} + m_3 \right)$$

$$M_{13} = M_{21} = M_{31} = M_{32} = 0$$

$$M_{12} = l_1 l_2 \left( \frac{m_2}{2} + m_3 \right) \cos(q_2) - l_2^2 \left( \frac{m_2}{3} + m_3 \right) = M_{21}$$

$$M_{22} = l_2^2 \left( \frac{m_2}{3} + m_3 \right)$$

$$M_{33} = m_3$$

$$C_1 = l_1 l_2 \sin(q_2)$$

$$C_{11} = -q_2 C_1 (m_2 + 2m_3)$$

$$C_{12} = -q_2 C_1 \left( \frac{m_2}{2} + m_3 \right) = C_{21}$$

$$C_{13} = C_{22} = C_{23} = C_{31} = C_{32} = C_{33} = 0$$

$$G_3 = -m_3 g$$

In which  $q_1, q_2, q_3$  are the angle of joints 1, 2 and 3;  $m_1, m_2, m_3$  are the mass of the links 1, 2 and 3;  $l_1, l_2, l_3$  are the length of links 1, 2 and 3;  $g$  is the gravity acceleration. The system parameters of the SCARA robot are selected:

$$l_1 = 1.0m; \quad l_2 = 0.8m; \quad l_3 = 0.6m$$

$$m_1 = 1.0kg; \quad m_2 = 0.8kg; \quad m_3 = 0.5kg;$$

$$g = 9.8$$

Important parameters that affect the control performance of the robotic system are the external disturbance  $t_1(t)$ , and friction term  $f(\dot{q})$ .

External disturbances are selected as:

$$t_1(t) = \begin{bmatrix} 5 \sin(2t) \\ 5 \sin(2t) \\ 5 \sin(2t) \end{bmatrix}$$

Friction forces considered in these simulations as the following:

$$f(\dot{q}) = \begin{bmatrix} 12\dot{q}_1 + 0.2 \text{sign}(\dot{q}_1) \\ 12\dot{q}_2 + 0.2 \text{sign}(\dot{q}_2) \\ 12\dot{q}_3 + 0.2 \text{sign}(\dot{q}_3) \end{bmatrix}$$

In this simulation the robot manipulator considered to carry a load of 10kg to 20kg with no prior knowledge of the weight; using 1second to 4second of the total simulation time.

The desired trajectories for the three joint to be tracked are given:

$$q_{d1}(t) = 1 + 0.1(\sin(t) + \sin(2t))$$

$$q_{d2}(t) = 1 + 0.1(\cos(2t) + \cos(3t))$$

$$q_{d3}(t) = 1 + 0.1(\sin(3t) + \sin(4t))$$

In this simulation the model is estimated by applying a factor to the corresponding parameter matrices of the original system in each environment to count uncertainties, i.e.

$$\hat{M} = .9M, \hat{C} = .8C, \hat{G} = .85G$$

The control input  $u$  is chosen as (11) where  $A = \text{diag}[80, 80, 80]$  and  $\lambda = \text{diag}[30, 30, 30]$  in (4). The spread factor of the GRNN is chosen as .8

The simulation results are shown in Figs 1–10. As shown in Fig. 1, 3 and 5 shows the tracking of joint angles 1, 2 and 3 respectively. The joint angles track the desired trajectories and the proposed control scheme drives the robotic manipulator to its desired positions. Tracking error converges to zero as shown in Fig. 2, 4 and 6 respectively for the three joint angles. The control input generated for the three joint angles with the proposed scheme are shown in Figs. 7, 8 and 9 respectively. For comparison purpose the control effort for the second joint generated with the classical controller is shown in Fig. 10, which shows a lot of chattering in the controller. Hence it has been observed that chattering problem in the control input is eliminated with the proposed controller and it also gives enhanced robust performance under uncertainties as compared to the classical sliding mode controller.

## 6. Conclusion

In this paper an adaptive sliding mode controller using GRNN neural network is proposed for robotic manipulators. The discontinuous parts of the classical sliding mode controller are replaced by GRNN neural networks, which are nonlinear and continuous, to avoid the chattering. As shown in simulation the proposed GRNN neural network can compensate the system uncertainties without any prior knowledge of uncertainties. The simulation results demonstrate that the proposed adaptive sliding mode control scheme is a stable control.

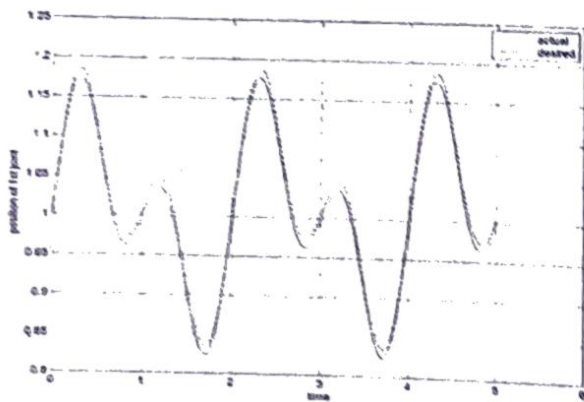


Fig. 1. Tracking of joint angle 1 with proposed method



Fig. 2. Tracking error for joint 1 with proposed method

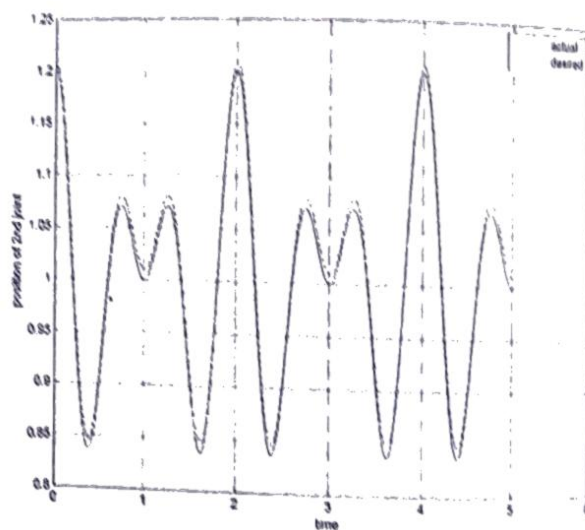


Fig. 3. Tracking of joint angle 2 with proposed method

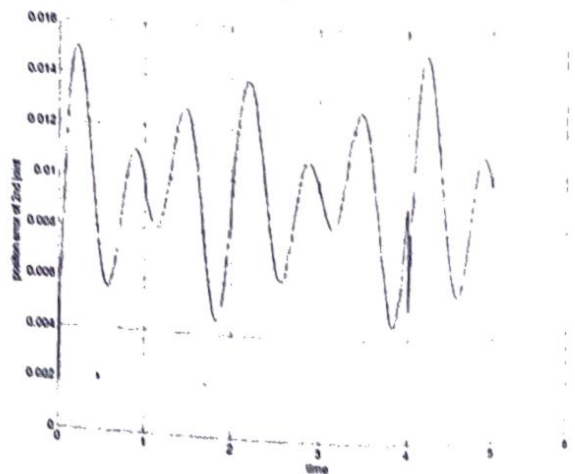


Fig. 4. Tracking error for joint 2 with proposed method

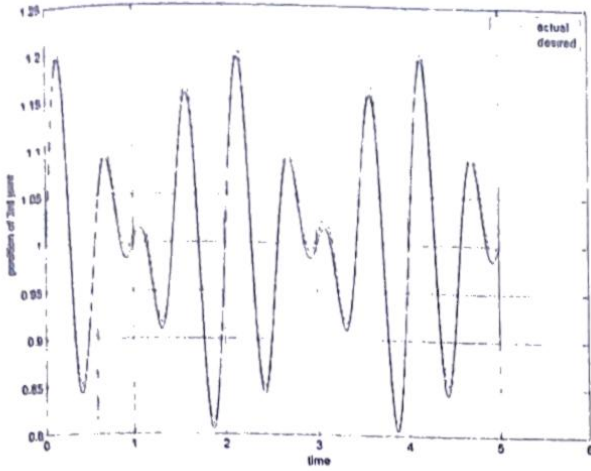


Fig. 5. Tracking of joint angle 3 with proposed method

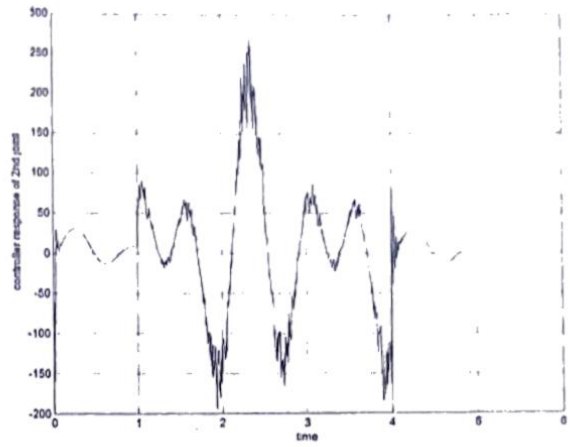


Fig. 8. control input for joint 2 with proposed method

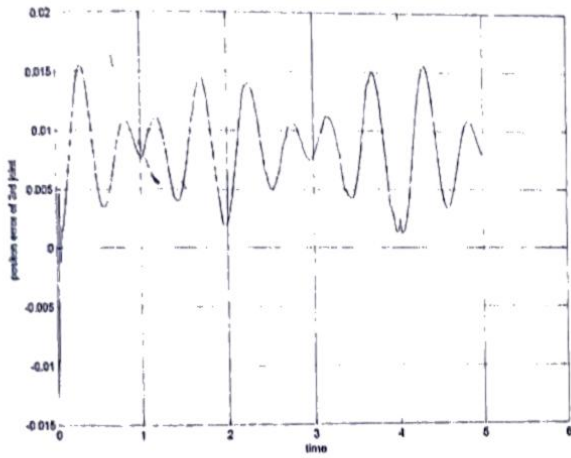


Fig. 6. Tracking error for joint 3 with proposed method

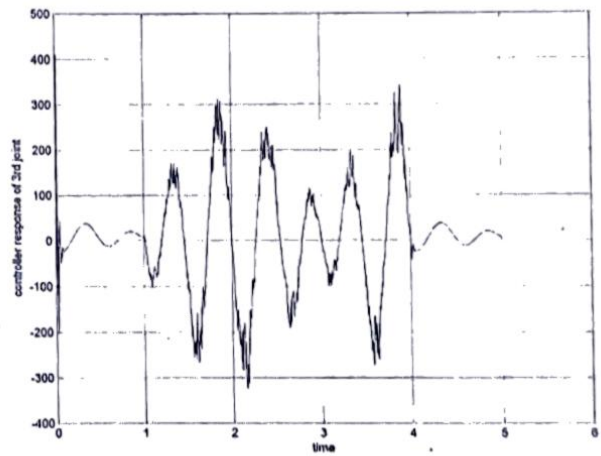


Fig. 9. Control input for joint 3 with proposed method

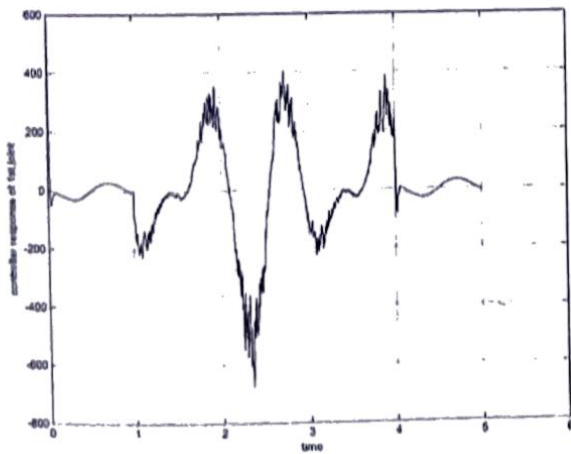


Fig. 7. control input for joint 1 with proposed method

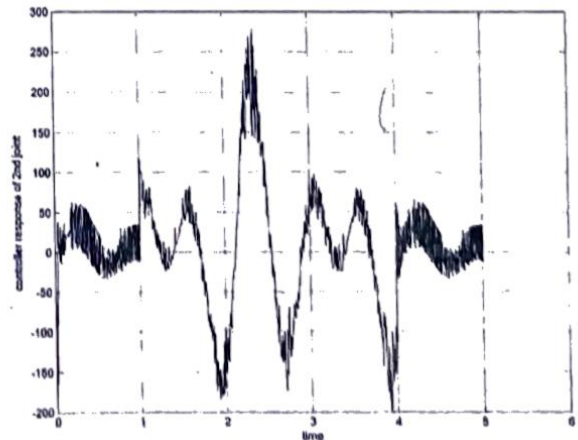


Fig. 10. Control input for joint 2 with classical sliding mode control, which is showing lots of chattering effects

## REFERENCES

- [1] J. E. Slotine, W. Li, *Applied Nonlinear Control*. Prentice-Hall Inc. New Jersey, 1991.
- [2] K. D. Yong, "Controller design for a manipulator using theory of variable structure system," *IEEE Trans. Sys. Man, Cybern.*, vol. 8, no. 2, pp. 210-218, Feb. 1978.
- [3] V. I. Utkin, "Sliding mode control design principles," *IEEE Trans. on Industrial Electronics*, vol. 40, no.1, pp. 23-36, 1993..
- [4] K. K. D. Young, V. I. Utkin, and U. Ozguner, "A control engineer's guide to sliding mode control," *IEEE Trans. on Control Systems and Technology*, Vol.7, No. 5, pp. 348-342,1999.
- [5] K S. Narendra, J. Balakrishnan, "Improving transient response of adaptive control systems using multiple models and switching," *IEEE Trans. Automatic Control*, vol. 39, pp.1861-1866, Sept. 1994.
- [6] 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..
- [7] K. Zerenik, M. Rodic, R. Safaric, B. Curk, "Neural network sliding mode robot control," *Robotica*, vol. 15, pp.23-30, 1997..
- [8] M. Ertugrul, O. Kaynak, "Neuro sliding mode control of robotic manipulators," *Mechatron*, vol. 10, no 1-2, pp.243-267, 2000.
- [9] M. Ertugrul, O. Kaynak, "Neural network adaptive sliding mode control and its application to SCARA type robot manipulator," in Proc. *IEEE ICRA'97*, pp. 2932-2937,1997.
- [10] B. Siciliano and L. Sciavicco, *Modeling and control of robot manipulators*, UK, Springer Verlag, Ssecond Edition, 2000..
- [11] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Network* , vol. 2(6) pp. 568-576, 1991.
- [12] M.Timothy, *Advanced Algorithms for Neural Networks: AC++ Sourcebook*, Wiley, Canada, 1995.
- [13] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.* , vol. 33, pp.1065-1076, 1962.

□