International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 8 ǁ August 2022 ǁ PP. 437-444

Adaptive Control of RoboticManipulators with Uncertain Load Using General Regression Neural Network

Sheheeda Manakkadu[1], Sourav Dutta[2]
*[*1]Department of Computer and Information Science, Gannon University, Erie, Pennsylvania*
[2]Department of Computer Science, Ramapo College, Mahwah, New Jersey
Corresponding Author:Sheheeda Manakkadu

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.

Keywords: *General Regression Neural Network, Adaptive Control,Robotic Manipulators.*

Date of Submission: 13-08-2022 Date of acceptance: 28-08-2022

I. 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.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 asrobotic manipulators [1], 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.

The data available from measurements of an operating system is generally never enough for a backpropagation neural network [11]. Therefore, the use of a probabilistic neural network is especially advantageous due to its ability to converge to the underlying function of the data with only few training samples available. The additional knowledge needed to get the fit in a satisfying way is relatively small and can be done without additional input by the user. This makes GRNN a very useful tool to perform predictions and comparisons of system performance in practice.

1.Preliminaries

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

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

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, $F(q,\dot{q}) \in R^{n\times n}$ 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

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

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.

II. Classical sliding mode control (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 siding surface can be written as

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

Where $\lambda = diag[\lambda_1, ..., \lambda_i, ..., \lambda_n]$ and 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:

$$\frac{1}{2}\frac{ds_i^2}{dt} \le -\eta_i |s_i| \quad (5)$$

where η_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:

$$\dot{q}_r = \dot{q} - s = \dot{q}_d - \lambda e$$
$$\ddot{q}_r = \ddot{q} - \dot{s} = \ddot{q}_d - \lambda \dot{e} \quad (6)$$

The control input u can now be chosen as

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

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

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

where $\Delta f = \Delta M \ddot{q}_r + \Delta C \dot{q}_r + \Delta G + F$. It has been proven [10] that by considering the Lyapunov function candidate as

$$V = \tfrac{1}{2} s^T M s \quad (9)$$

And choosing K such that

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

where $|\Delta f_i|_{bound}$ 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) issatisfied.

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

IV. 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 $f(X,Y)$ is known, the expected value of *Y* given *x* (the regression of *Y* on *x*) can be estimated as

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

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 thefunction 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 σ 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

$$\hat{y}(x) = E[Y \mid x]$$
$$= \sum_{i=1}^{n} [y_i \exp(d_i)] / \sum_{i=1}^{n} \exp(d_i) \qquad (14)$$

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 k_i
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

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

In the above expression s is the new input and s_j is the stored input, σ is the spread factor. In equation (15) k_j is the stored output corresponding to s_j and $\hat{k}_i$ implies the estimated value of true k_i .In continuation, an adaptive law is designed to guarantee that k_i can compensate the system uncertainties. According to the property of universal approximation, there exists $\delta_i > 0$ such that

$$| \Delta f_i - \hat{k}_i | < \delta_i \qquad (17)$$

where $\hat{k}_i$ is the output of the GRNN and δ_i can be chosen as small as possible. A good estimation of k_i depends on the selection of spread factor σ .

V. Simulation

Simulation results shows that the proposed method working considerably well for different trajectories. In this section, the proposed adaptive SMC is used on a two-link scara robot, with parameter matrices given by

$$M(q) = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$$

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

$$G(q) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$$

Where

$$M_{11} = l_1^2\left(m_1 + m_2\right)$$
$$M_{12} = m_2 l_2 l_1\left(\sin(q_1)\sin(q_2) + \cos(q_1)\cos(q_2)\right)$$
$$M_{21} = M_{12}$$
$$M_{22} = m_2 l_2^2$$
$$c_{11} = m_2 l_2 l_1 q_2$$
$$c_{12} = 0$$
$$c_{21} = 0$$
$$c_{22} = c_{11}$$

$$G_1 = -1.8\cos(q_2) + 0.64\cos(q_1 + q_2)g$$
$$G_2 = -0.64\cos(q_1 + q_2)g$$

In which q_1, q_2 are the angle of joints 1,2; m_1, m_2 are the mass of the links 1,2; l_1, l_2 are the length of links 1,2 ; g is the gravity acceleration.The system parameters of the scara robot are selected:

$$l_1 = 1.0m; l_2 = 0.8m;$$
$$m_1 = 1.0kg; m_2 = 0.8kg;$$
$$g = 9.8$$

The desired trajectories for the twojoint to be tracked are given:

$$q_{d1}(t) = -0.3\cos(\pi t)$$
$$q_{d2}(t) = -0.3\cos(\pi t)$$

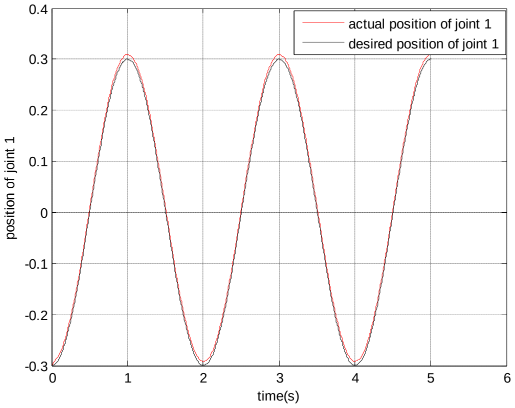
$$c_{11} = m_2 l_2 l_1 q_2$$
$$c_{12} = 0$$
$$c_{21} = 0$$
$$c_{22} = c_{11}$$
$$G_1 = -1.8\cos(q_2) + 0.64\cos(q_1 + q_2)g$$
$$G_2 = -0.64\cos(q_1 + q_2)g$$

In which q_1, q_2 are the angle of joints 1,2; m_1, m_2 are the mass of the links 1,2; l_1, l_2 are the length of links 1,2 ; g is the gravity acceleration.The system parameters of the scara robot are selected:

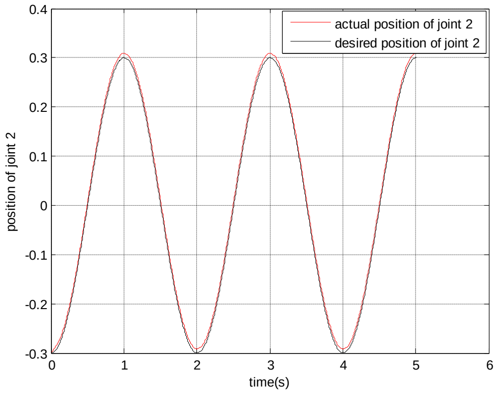
$$l_1 = 1.0m; l_2 = 0.8m;$$
$$m_1 = 1.0kg; m_2 = 0.8kg;$$
$$g = 9.8$$

The desired trajectories for the twojoint to be tracked are given:

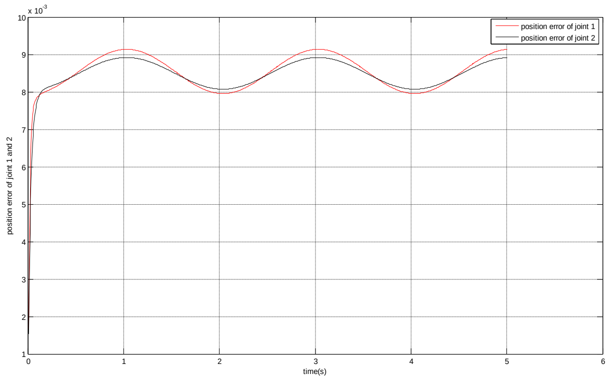
$$q_{d1}(t) = -0.3\cos(\pi t)$$
$$q_{d2}(t) = -0.3\cos(\pi t)$$



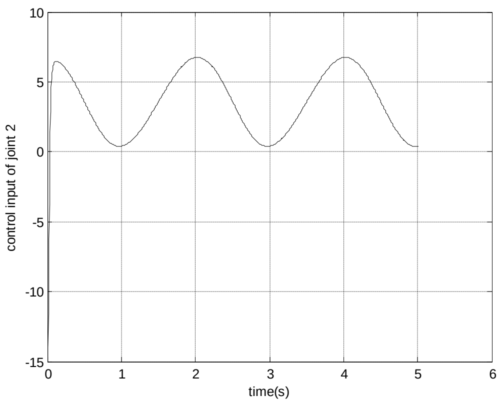
(a)



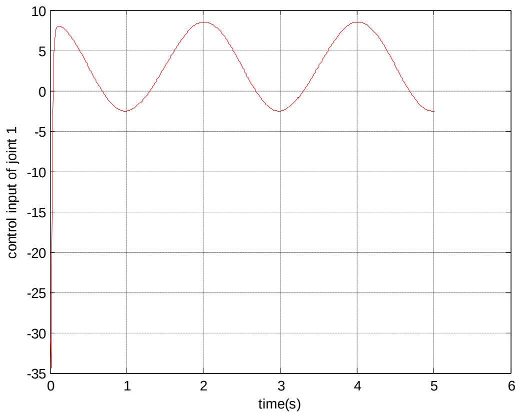
(b)



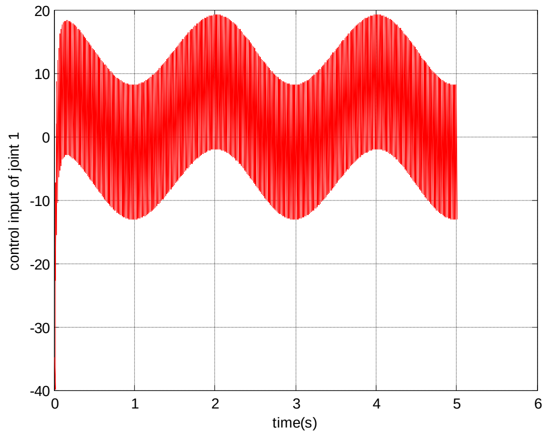
(c)



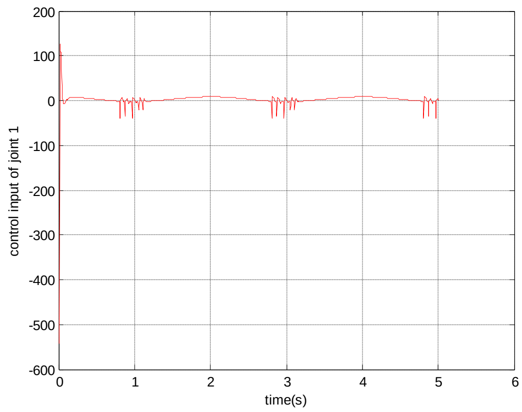
(d)



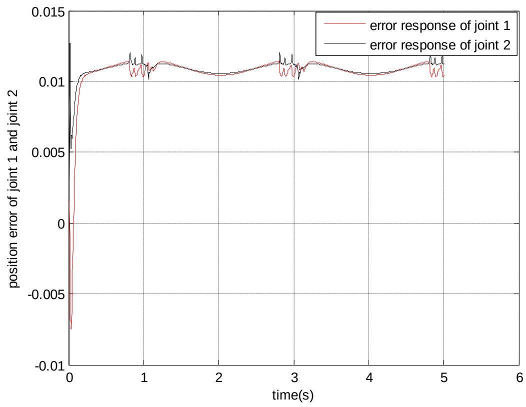
(e)



(f)



(g)



(h)

Fig 1. (a) position of joint 1 (b) position of joint 2(c) position error of joint 1 and 2 (d) control input of joint 1 (e) control input of joint 2 (f) control input of joint 1 by classical sliding mode control with sign function (h) control input of joint 1 by classical sliding mode control with saturation function (i) position error of joint 1 and 2 by classical sliding mode control with saturation function.

VI. Conclusion

The simulation results are shown in Figs 1. 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.

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,".IEEETrans.Systs, Man, Cybern., vol. SMC-8, no .2, pp. 210-218, Feb. 1978.

[3]. K. S. Narendra, J. Balakrishnan, "Adaptive control using multiple models, "IEEE Trans. Automatic Control, vol. 42,no. 2, pp. 171-187, Feb. 1997.

[4]. Kumpati. S. Narendra, Cheng xiang, "Adaptive control of discrete-time systems using multiple models," IEEE Trans.Automatic control, vol. 45, no. 9, pp. 1669-1686, sept. 2000.

[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]. K.D. Yong, Controller design for a manipulator using theory of variable structure system, IEEE Trans. Syst. Man. Cybern. SMC-8 (2), pp. 210–218, 1978.

[11]. Specht, D. F.: A general regression neural network, IEEE Trans. Neural Network 2(6) pp. 568–576, 1991.

[12]. Timothy, M.: Advanced Algorithms for Neural Networks: AC++Sourcebook, Wiley, Canada, 1995.

[13]. Parzen, E.: On estimation of a probability density function and mode, Ann. Math. Statist.33pp. 1065–1076,1962.
[14]. R.J Wai and K.Y Hsich, "Tracking Control Design for Robot Manipulator Via Fuzzy Neural Network". Proceedings of the IEEE Int. Conference on Fuzzy Systems .Volume 2,pp. 1422-142,2002.