Available online at www.worldscientificnews.com

World Scientific News

An International Scientific Journal

WSN 121 (2019) 14-21 EISSN 2392-2192

Neural Network based control of Robot Manipulator

Arnav Ohri[1,a], Salim[2,b]

[1]General Electric, Bangalore, India

[2]KIET Group of Institutions, Ghaziabad, India

[a,b]E-mail address: arnav9999@gmail.com , salim@kiet.edu

ABSTRACT

This article proposes an RBFNN (Radial Basis Function Neural Network) and sliding mode based controller to manipulate the robot manipulator. The technique used has been based on a sliding mode control approach that can drive the system towards a sliding surface by Gaussian radial basis function neural network based tuned-controller.

***Keywords*:** Robotic Manipulator, RBF Neural Network, Sliding Mode Control

1. INTRODUCTION

Advanced control-design methodologies has been used to overcome the limitations of traditional or conventional-feedback control techniques like PID because complex systems have many dissemblance due to unknown external disturbances, parasitical, unmodelled dynamics and plant change. Advanced control system techniques do not require any model of system and the controller, they are model-free.

Artificial Neural Networks (ANN) is an advanced control scheme. In this article RBFNN special technique of ANN is used for functional approximation [1]. In RBF Neural Network neuron present in the input layer corresponds to prophesier variable. Neurons present in hidden layer consist of radial basis centered function. Output contains weighed sum of output from hidden layers. RBF Neural Networks are similar to K-Nearest Neighbor (KNN) models [3].

(Received 02 February 2019; Accepted 15 February 2019; Date of Publication 16 February 2019)

Here weight is equal to RBF (Distance). Robust Characteristics to System Disturbances and Parameter vagueness are included in Slide Mode Control which is a type of Variable Structure Controller (VSC). A VSC System has a special type of Nonlinear System characterized by a discontinuous control which changes the System Structure when the States reach the Intersection of Sets of Sliding Surfaces [2]. The System behaves independently of its general dynamical characteristics and system disturbances once the controller has driven the System into a Sliding Mode [2]. Discontinuous laws of sliding mode control have the capability to drive the trajectories towards the sliding mode surface in finite time. Proposed technique uses RBF neural network and sliding mode controller to control the position of robotic manipulator [3, 4]. The combination includes the advantages of RBFNN and SMC; simulation results show fast convergence of tracking with minimum error almost zero as compared to conventional controllers. The present article organized as follow. Generalized dynamics of rigid robotic manipulator is given in section II and section III presents the controller structure. Section IV nonce the simulation study and results obtained and final conclusion in section V.

2. MODEL OF ROBOTIC MANIPULATORS

Multi-link rigid robotic manipulator second-order nonlinear vector differential equation is shown in Equation 1 [11].

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

Position, velocity and acceleration vectors are given by $q, \dot{q}, \ddot{q} \in R^n$, $M(q) \in R^{n\times n}$ relation shows inertia matrix, $C(q,\dot{q}) \in R^{n\times n}$ relation shows centrifugal torques, $F(q,\dot{q}) \in R^{n\times n}$ relation shows unstructured vagueness of the dynamics including friction and other disturbances, $G(q,\dot{q}) \in R^{n\times n}$ is the gravity vector and $u(t) \in R^{n\times 1}$ relation shows actuator torque vector [13].

3. RBF BASED SLIDING MODE CONTROL

SMC control design objective has to move the joint position q to the desired position q_d. The tracking error has been illustrated in the following form:

$$e = q - q_d \tag{2}$$

The siding (hyper) surface is written in Equation 3 as

$$s = \dot{e} + ce \tag{3}$$

The sliding surface is given by $s(t)$, and the control law u is used for RBF neural network [1]. Stimulation results for the hidden layer distances between the input values of sliding variables and centre position are shown in Equation 4.

$$\Phi_i = ||s - \mu_i|| \tag{4}$$

where the input sliding variable is $s(t)$ and μ_i is the central position of i^{th} neuron. Weight between hidden and output layer is given by w_i. Weight adjusted is based on an adaptive rule.

The output of a RBFNN which can be achieved by [1]

$$u = \sum_{i=1}^{p} w_i \exp\left(\frac{||s-\mu_i||^2}{2\sigma_i^2}\right) \tag{5}$$

The weight function and adaptive rules has adjusted weights in order to search the optimal weighting values and to obtain the stable convergence of the controller [1]. Based on regulated weights and adaptive rules the output of RBFNN approximate the nonlinear map between the sliding input variable and the control law [1]. The objective of the controller makes $s(t)s(\dot{t})$ → 0 based on the Lyapunov theory.

Hence the weight function is adjusted by following procedures [10-18]:

Let

$$E = s(t)s(\dot{t}) \tag{6}$$

$$w_i = -\eta \frac{\partial E}{\partial w_i(t)} = -\eta \frac{\partial s(t)s(\dot{t})}{\partial w_i(t)} = -\eta \frac{\partial s(t)s(\dot{t})}{\partial u} \frac{\partial u}{\partial w_i(t)} \tag{7}$$

Since

$$\frac{\partial s(t)s(\dot{t})}{\partial u} = -\Upsilon s(t)$$

$$\frac{\partial u}{\partial w_i(t)} = exp\left(\frac{||s-\mu_i||^2}{2\sigma_i^2}\right) \tag{8}$$

Hence, learning algorithm is

$$dw_i = \Upsilon s(t) exp\left(\frac{||s-\mu_i||^2}{2\sigma_i^2}\right) = \Upsilon s(t) h_i(s) \tag{9}$$

Weightings between hidden and output layers neurons have been adjusted online to bring out learning ability of RBFNN. By trial and error method we can choose μ, σ_i of RBF neural network.

4. SIMULATION AND RESULT

Neural Network based SMC has been used for the position control of a 2-link SCARA robot. Parameter matrices for 2-link SCARA robot is given by

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

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

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

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

where

$$M_{11} = l_1^2(m_1 + m_2)$$

$$M_{12} = m_2 l_2 l_1(\sin(q_1)\sin(q_2) + \cos(q_1)\cos(q_2))$$

$$M_{12} = M_{21}$$

$$M_{22} = m_2 l_2^2$$

$$C_{11} = m_2 l_2 l_1 q_2$$

$$C_{12} = 0$$

$$C_{21} = 0$$

$$C_{22} = C_{11}$$

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

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

Angles for joint 1 and joint 2 are given by q_1, q_2 respectively. Mass for joint 1 and joint 2 are given by m_1, m_2 respectively. Length for joint 1 and joint 2 are given by l_1, l_2 respectively. Acceleration due to gravity is given by g [3].

Different parameters of the SCARA robot are as follow:

Length parameters are $l_1 = 1.0m;\ l_2 = 0.8m.$

Mass parameters are $m_1 = 1.0kg;\ m_2 = 0.8kg.$

$$g = 9.8m/s^2$$

Friction can disturb the control performance of manipulator system. $f(\dot{q})$ is known as friction and can be selected as:

$$f(\dot{q}) = \begin{bmatrix} 12\dot{q}_1 + 0.2sign(\dot{q}_1) + d_1 \\ 12\dot{q}_2 + 0.2sign(\dot{q}_2) + d_2 \end{bmatrix}$$

where d_1 and d_2 are Random disturbances associated with friction are given by d_1, d_2 respectively. Optimal trajectories for the 2 different joints are given by [7]:

$$q_{d_1}(t) = -0.3\cos(\pi t)$$

$$q_{d_2}(t) = -0.3\cos(\pi t)$$

Objective of simulation study is to track the desired joint angles path, so that the tracking error (e) is minimised.

Simulation results for the tracking path of joint angle 1 and joint angle 2 are shown in Figure 1 and Figure 2 respectively. Blue lines show the desired trajectory path and red line shows the actual trajectory path obtained.

It is observed that except the initial transient period, the robotic manipulator has able to track the desired trajectory perfectly with negligible tracking error. Steady state reached very fast. The proposed controller has been able to control the position of the robotic manipulator satisfactorily in spite the presence of the friction with minimum error in the manipulator dynamics ensuring the accuracy and stability of the system.

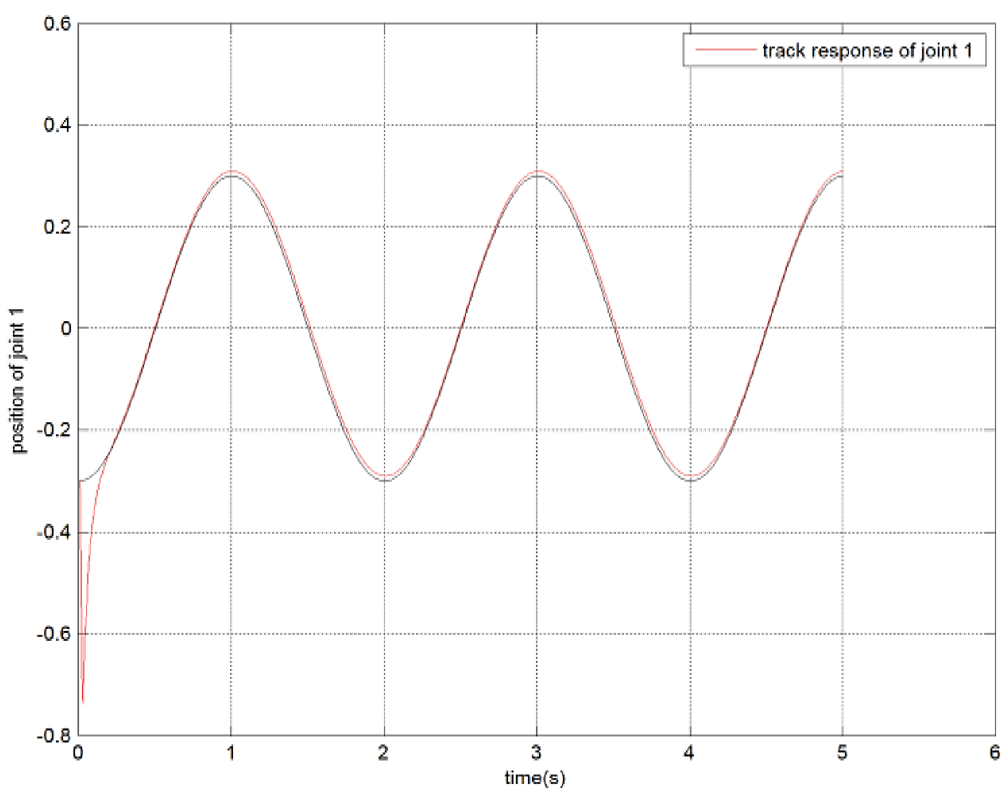

Figure 1. Tracked path response of joint position 1

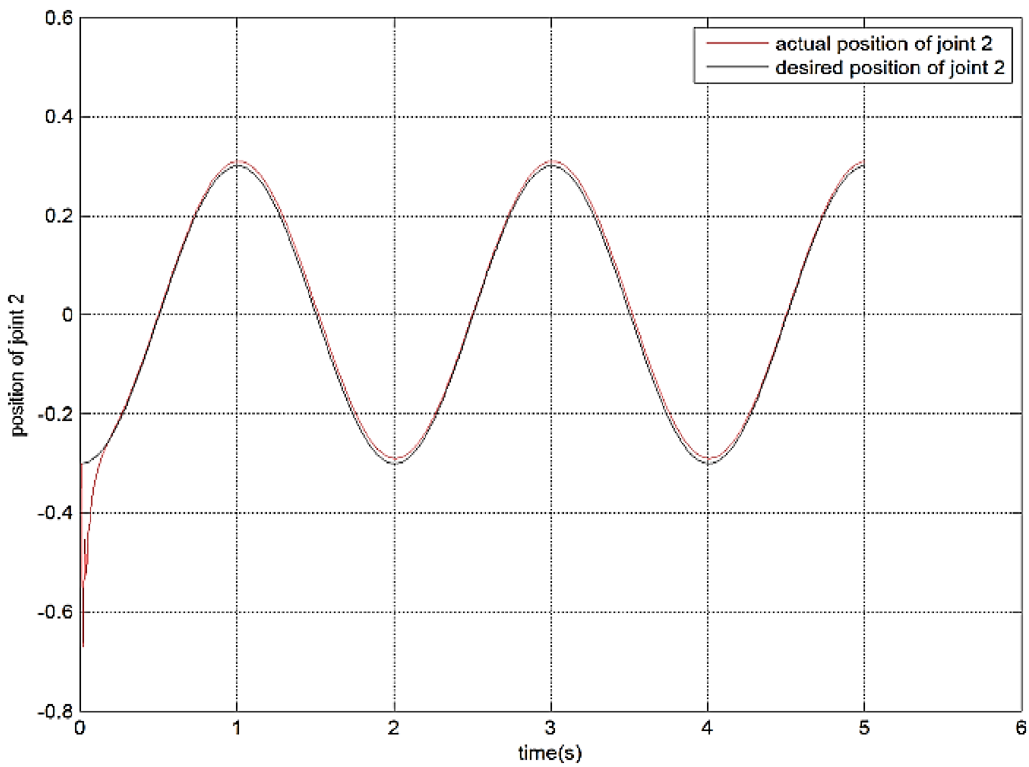

Figure 2. Tracked path response of joint position 2

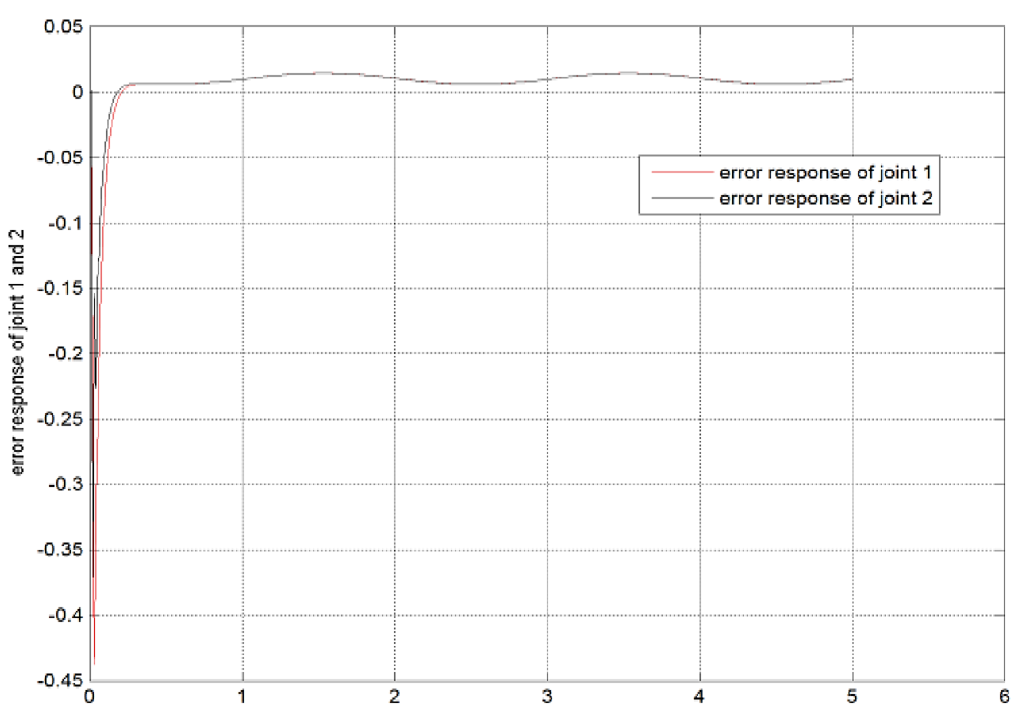

Figure 3. Tracked path error in joint position 1 and 2 (e_1 and e_2)

5. CONCLUSIONS

SMC based on RBF Neural network is applied to control a nonlinear 2 DOF Robot manipulator under friction and uncertain disturbances. The approach is based on methodology that the system has been deriving the manipulator towards a sliding surface. To control the robotic manipulator we do not require model for controller as well as system. Hence simulation results into robust performance. The proposed controller scheme successfully drive the position of robotic manipulator to the desired position with negligible tracking error as shown in the figures for tracking response for different angles.

References

[1] Rumelhart D, Hinton D, Williams G. Learning internal representations by error propagation. In *Parallel* Distributed Processing, Rumelhart D, McClelland F (eds), vol. 1. MIT Press, Cambridge, MA, 1986.

[2] Yu XH, Efe MO, Kaynak O. A general backpropagation algorithm for feedforward neural networks learning. *IEEE Transactions on Neural Networks* 2002; 13(1): 251–254

[3] Efe MO, Kaynak O. Stabilizing and robustifying the learning mechanisms of artificial neural networks in control engineering applications. *International Journal of Intelligent Systems* 2000; 15(5): 365–388

[4] Ge SS, Hang CC, Lee TH, Zhang T. Stable Adaptive Neural Network Control. Kluwer: Boston, 2001.

[5] Lewis FL, Ge SS. Neural Networks in Feedback Control Systems. Mechanical Engineer's Handbook. Wiley: New York, 2005.

[6] Kosmatopoulos EB, Polycarpou MM, Christodoulou MA, Ioannou PA. High-order neural network structures for identification of dynamical systems. *IEEE Transactions on Neural Networks* 1995; 6(2):431–442

[7] Suykens JAK, Vandewalle J, De Moor B. Lur'e systems with multilayer perceptron and recurrent neural networks; absolute stability and dissipativity. *IEEE Transactions on Automatic Control* 1999; 44: 770–774

[8] Yu W, Poznyak AS, Li X. Multilayer dynamic neural networks for nonlinear system on-line identification. *International Journal of Control* 2001; 74(18): 1858–1864.

[9] Sira-Ramirez H, Colina-Morles E. A sliding mode strategy for adaptive learning in Adalines. IEEE Transactions on Circuits and Systems I: *Fundamental Theory and Applications* 1995; 42(12): 1001–1012

[10] Yu X, Zhihong M, Rahman SMM. Adaptive sliding mode approach for learning in a feedforward neural network. *Neural Computing and Applications* 1998; 7: 289–294

[11] Parma GG, Menezes BR, Braga AP. Sliding mode algorithm for training multilayer artificial neural networks. *Electronics Letters* 1998; 34(1): 97–98

[12] Topalov AV, Kaynak O. Neural network modeling and control of cement mills using a variable structure systems theory based on-line learning mechanism. *Journal of Process Control* 2007; 14: 581–589

[13] Shakev NG, Topalov AV, Kaynak O. Sliding mode algorithm for on-line learning in analog multilayer feedforward neural networks. In Artificial Neural Networks and Neural Information Processing, Kaynak et al. (eds), Lecture Notes in Computer Science. Springer: Berlin, Heidelberg, New York, 2008; 1064–1072.

[14] Utkin VI. Sliding Modes in Control and Optimization. Springer: Berlin, Heidelberg, New York, 1992.

[15] Sun, T. Y.; Hsieh, S. T. & Lin, C. W. (2005). Particle Swarm Optimization Incorporated with Disturbance for Improving the Efficiency of Macrocell Overlap Removal and Placement. Proceeding of The 2005 International Conference on Artificial Intelligence, pp. 122-125

[16] Yunfei, B. & Zhang, L. (2007). Genetic algorithm based self-growing training for RBF neural Network, *IEEE Neural Networks*, Vol. 1, pp. 840-845

[17] Efe MO, Kaynak O, Yu X. Sliding mode control of three degrees of freedom anthropoid robot by driving the controller parameters to an equivalent regime. *Journal of Dynamic Systems, Measurement and Control* (ASME) 2010; 122: 632–640

[18] Topalov AV, Kaynak O. On-line learning in adaptive neurocontrol schemes with a sliding mode algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part B* 2011; 31(3): 445–450