SELF ADJUSTING RBNN FOR TWO LINK AND THREE LINK MANIPULATOR

Neha Kapoor[1], Jyoti Ohri[2]

[1]*National Institute of Technology, Kurukshetra*
[2]*National Institute of Technology, Kurukshetra*

Abstract

This paper addresses the solution to the robust trajectory tracking problem in presence of uncertainties and disturbances. In this research, area worked upon is the development of an intelligent hybrid controller to ensure the accurate trajectory tracking of a robotic manipulator. The ONCC is a hybrid intelligent controller made with the combination of Radial Bias neural network (RBNN) and particle swarm optimization (PSO). PSO is used to get the optimized value of the spread factor. To check the robustness of the ONCC, firstly is applied to a 2 link and then to a 3 link SCARA manipulator tracking various trajectories and with different disturbances and uncertainties in the system. MATLAB platform has been used for simulation purpose. For validation purpose, results of the ONCC have been compared with the control results of original RBNN and the basic PD controller. Paper has been finished with the appropriate conclusions.

***Keywords:** Radial Bias Neural Network (RBNN), Particle Swarm Optimization (PSO), Hybrid Intelligent Controllers.*

---***---

1. INTRODUCTION

In almost all the robotic systems main causes affecting the control performance is not only the nonlinearity in the system dynamics but also the unknown dynamics in the model of the system. It can be said that because of these unmodeled and partially known dynamics these control laws cannot ensure the good control performance in a particular system. This creates a continuous need for an effective controller for the manipulator trajectory tracking problem in practical problems and hence creates a valid base for the research undertaken in the work of this paper.

Most widely used and the most classical control scheme naming, Proportional Derivative (PD) is still used in almost all the industrial applications. This is due to its enormous advantages such as simplicity, easy to implement in hardware or software, and does not require a precise process model to start up and maintain and hence has invariance to parametric uncertainties to a level [1-3]. Basic PD controller becomes unstable in the presence of external uncertainties and restricts its use. This creates a need of designing an effective control technique even without the accurate knowledge of the non linear and time varying system dynamics. One of the solutions to this is to make the classical controllers intelligent. This can be through by introduction of intelligent agents like a few of them are neural networks [4, 5] and PSO [6]. Proves have been mentioned in literature, [7-10] by proving intelligent techniques over the classical PD controller.

Neural Network has good non-linear prediction capabilities [11] and its learning capabilities have made it robust for universal applications. On the other hand, high processing time, prone to generalization in neural networks can degrade its performance. Moreover, tuning the network parameters by 'trial and error' method is a time consuming and frustrating task and hence, needs to be touched for the best output performance. This can be made possible by making the control system adaptive. Auto adaptability can be brought in the system by using one of the Evolutionary Computational (EC) intelligent techniques [6]. To get the optimal solutions stochastic based search algorithm PSO is one of the recently worldwide used EC techniques [12]. Based on the simulation of simplified animal social behavior such as fish schooling, bird flocking etc., in 1995; PSO is developed by Kennedy and Elbert [13]. Starting with random population in search space, it results in the most favorable solution. Hybrid of these NN and EC (used to optimize the parameters of the neural network) has found to be an effective solution to the control problems and hence Evolutionary Neural Networks (ENN) has been generated [14, 15].

In this paper, a RBNN control scheme having PSO optimized 'spread factor' has been proposed and proved. To check the robustness of the controller various uncertainties naming, constant disturbance, high frequency continuous disturbance, white noise and LuGre friction has been added to the two link manipulator system and compiled been proved by comparing them with RBNN and PD controllers. For validity of controller, the ONCC has been further applied to a three link SCARA manipulator.

Further the paper is summarized as; Section II has system dynamics; Brief introduction about the various controllers and the ONCC is given in Section III. Section IV has simulation example and results. In last, conclusion is in Section V.

2. SYSTEM DYNAMICS

The dynamics of revolute joint planar type of robotic manipulator can be described by following nonlinear Lagrange equation (1)[16],

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

with $q \epsilon R^n$ as the joint position variables, τ as vector of input torques, $M(q)$ is the inertia matrix which is symmetric and positive definite, $V(q,\dot{q})$ is the coriolis and centripetal matrix, $G(q)$ includes the gravitational forces, T_d includes the disturbances (fixed and continuous) and the white noise, F is the internal friction.

Tracking error vector and error velocity be defined in (2-3) as

$$e = q - q_c, \in R_n \qquad (2)$$

$$\dot{e} = \dot{q} - \dot{q}_c, \in R_n \qquad (3)$$

This controller is used to track three different paths for a two-link manipulator given in (4-6) and represented in Fig. [1 & 2].

1st path

$$q_1 = 0.3\sin(0.7t - \tfrac{\pi}{2}) + 0.3\sin(0.1t - \tfrac{\pi}{2}) + 0.7 \qquad (4a)$$

$$q_2 = 0.5\sin(0.9t - \tfrac{\pi}{2}) + 0.5\sin(0.1t - \tfrac{\pi}{2}) + 1.1 \qquad (4b)$$

2nd path

$$q_1 = \sin(0.67t) + \sin(0.3t) \qquad (5a)$$

$$q_2 = \sin(0.39t) + \sin(0.5t) \qquad (5b)$$

3rd path

$$q_1 = 1.6 - 1.6\exp(-8t) - 12.8\exp(-8t) \qquad (6a)$$

$$q_2 = 1.6 - 1.6\exp(-8t) - 12.8\exp(-8t) \qquad (6b)$$

For a 3-link manipulator path tracked is given in (7) and represented in Fig. [3-5]

$$q_1 = 1 + 0.1(\sin(t) + \sin(2t)) \qquad (7a)$$

$$q_2 = 1 + 0.1(\cos(2t) + \cos(3t)) \qquad (7b)$$

$$q_3 = 1 + 0.1(\sin(3t) + \sin(4t)) \qquad (7c)$$

3. CONTROLLERS SCHEMES

This section contains a brief overview of the various controllers worked upon in this work.

3.1 Proportional Derivative (PD)

Proportional Derivative controller is a generic control loop feedback mechanism widely used in industrial control systems. PD controller attempts to correct the error between a measured process variable and the desired set point by calculating and then outputting a corrective action that can adjust the process regularly. Mathematically PD controller can be given in the following control term (8):

$$\tau = K_p e(t) + K_d \dot{e}(t) \qquad (8)$$

where, K_p and K_d are suitable positive definite diagonal n x n matrices.

However, this widely used PD has major shortcomings, such as: it works best for process that are linear and time invariant. PD has trouble in controlling complex systems, which are usually non linear, time variant, coupled and has parameter or structure uncertainties. Secondly, to get the best output from PD it must be best tuned; this in real time applications is often a frustrating experience. Most commonly constant gain parameters to be used in the PD controller are determined by TAE (Trial And Error) method. *Data generated in this controller is stored and used further for the training of RBNN and the ONCC.*

3.2 Radial Basis Neural Network (RBNN)

It is a kind of neural networks (NN) [17]. The RBNN possesses great mapping ability and has a similar feature to the fuzzy system makes it useful to control the dynamic systems [18-20]. Recently, a lot of intelligent control studies which focus on the position control of nonlinear systems have been proposed in [21, 22]. The aim of this paper is to design a hybrid intelligent control scheme for the periodic and predefined trajectory tracking control for a robotic manipulator. For this the RBNN used in this work has two inputs and one output. Input given to the NN is error and velocity error (e *and* $\dot{e}$) and output is taken as the input torque τ to be given to the manipulator for motion control problem. This used RBNN is trained by using the data of the basic PD controller. Gaussian function is used as the activation function of each neuron from the hidden layer to the output layer.

3.3 Optimized Neural Network Controller (ONNC)

It is proposed in this problem is self adjusting using PSO and hence, named as Optimized Neural Network Controller (ONNC). This controller is obtained by mutual complementation of neural networks and particle swarm optimization. Spread factor (s) of RBNN determines the width of an area in the input space to which each neuron responds. Hence, value of s affects the output of the RBNN. To get the best value of s in space, it has been searched globally using PSO. It can be seen that, this controller uses evolutionary algorithms to optimize artificial neural networks. A flowchart representing the control scheme is given in Fig. [6].

Based on the simulation of simplified animal social behavior such as fish schooling, bird flocking etc., within a search space PSO gives the globally best solutions [23]. Starting with random population in search space, it results in the optimal solution. During each step every particle is accelerated towards its best neighboring position as well as in the direction of global best position. Calculation of new position of the swarm is given by the equations (9, 10).

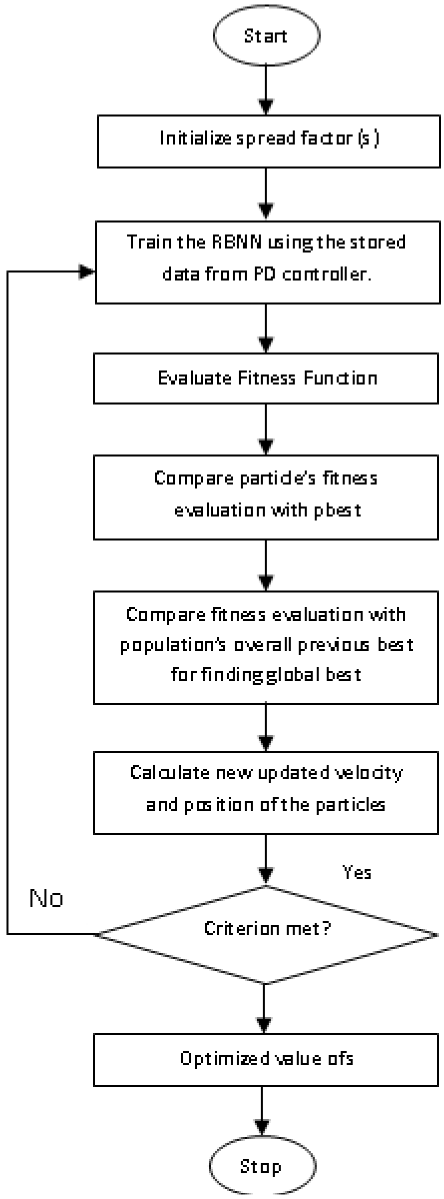

Fig. 6: Flowchart for ONCC.

$$v_{id} = v_{id} + c_1 \in_1 (p_{id} - x_{id}) + c_2 \in_2 (p_{xd} - x_{id} \quad (9)$$

$$x_{id} = x_{id} + v_{id} \quad (10)$$

where, in a D-dimensional space $\vec{x_i} = (x_{i1}, x_{i2}, \dots x_{iD})$ is a present position vector, $\vec{p_i} = (p_{i1}, p_{i2}, \dots p_{iD})$ is a best position vector, $\vec{v_i} = (v_{i1}, v_{i2}, \dots v_{iD})$ is a velocity vector, c is a constant having value 2, $\in_1$ and $\in_2$ are the random number generators. Further, improvement in the PSO is implemented by varying the accelerations coefficients (c1 and c2) in (1) with time. Varying acceleration coefficients c_1 and c_2 in (9) can be mathematically formulated as in (11) [24]

$$c_1 = (c_{1f} - c_{1i}) \frac{iter}{maxiter} + c_{1i} \quad (11a)$$

$$c_2 = (c_{2f} - c_{2i}) \frac{iter}{maxiter} + c_{2i} \quad (11b)$$

where $c_{1i,}\ c_{1f},\ c_{2i}\ and\ c_{2f}$ are constants, $iter$ is the current iteration number and $maxiter$ is the maximum number of allowable iterations. Value of w which is used in (9) is given in (12)

$$w = (w_{max} - w_{min}) \left(\frac{maxiter - iter}{maxiter}\right) + w_{min} \quad (12)$$

4. SIMULATION EXAMPLE AND RESULTS

To validate the performance of the work done in this paper, simulation study has been carried out and compiled under this section.

4.1 System Dynamics

The dynamics of a general n-link SCARA manipulator is given in (1) can be formulated as

$$M(q) = \begin{bmatrix} M_{11} & \cdots & M_{1n} \\ \vdots & \ddots & \vdots \\ M_{n1} & \cdots & M_{nn} \end{bmatrix}$$

$$\mathrm{V(q, \dot{q})} = \begin{bmatrix} V_{11} & \cdots & V_{1n} \\ \vdots & \ddots & \vdots \\ V_{n1} & \cdots & V_{nn} \end{bmatrix}$$

$$\mathrm{G(q)} = \begin{bmatrix} G_{11} \\ \vdots \\ G_{n1} \end{bmatrix}$$

Values for the manipulator are:

a. For a two link manipulator ($n = 2$)

$$\mathrm{M_{11}} = 8.77 + 1.02 \cos(\mathrm{q_2})$$

$$\mathrm{M_{12}} = \mathrm{M_{12}} = 0.76 + 0.51 \cos(\mathrm{q_2})$$

$$\mathrm{M_{22}} = 0.62$$

$$\mathrm{V_{11}} = -0.51 \sin(\mathrm{q_2})\, \dot{\mathrm{q}}_2$$

$$\mathrm{V_{12}} = -0.51 \sin(\mathrm{q_2})\, (\dot{\mathrm{q}}_1 + \dot{\mathrm{q}}_2)$$

$$\mathrm{V_{21}} = -0.51 \sin(\mathrm{q_2})\, \dot{\mathrm{q}}_1$$

$$\mathrm{V_{22}} = 0$$

$$\mathrm{G_{11}} = 74.48 \sin(\mathrm{q_1}) + 6.174 \sin(\mathrm{q_1}) + \mathrm{q_2}$$

$$\mathrm{G_{21}} = 6.174 \sin(\mathrm{q_1}) + \mathrm{q_2}$$

b. For a two link manipulator $(n = 3)$

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

$$M_{13} = M_{23} = M_{31} = M_{32} = 0$$

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

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

$$M_{33} = m_3$$

$$V_{11} = -\dot{q}_2(m_2 + 2m_3)$$

$$V_{12} = -\dot{q}_2(\frac{m_2}{3} + m_3)$$

$$V_{22} = -\dot{q}_2(\frac{m_2}{3} + m_3)$$

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

$$G_{11} = G_{21} = 0$$

$$G_{31} = -m_1 g$$

where $m_1, m_2\ and\ m_3$ are the angle of joints 1, 2 & 3; $m_1, m_2\ and\ m_3$ are the masses of the links 1, 2 & 3; $l_1, l_2\ and\ l_3$ are the lengths of the links 1, 2 & 3; g is the gravity acceleration. System parameters of three link SCARA robot are taken to be

$l_1 = 1m$ $l_2 = 0.8m$ $l_3 = 0.6m$ $m_1 = 1kg$ $m_2 = 0.8kg$ $m_3 = 0.5kg$ $g = 9.8$

4.2 Controllers

a. PD Controller:
Values of the controller gains for PD controller by TAE are chosen as

a. For a *two link* manipulator

$$K_p = \begin{bmatrix}1000 & 0\\ 0 & 200\end{bmatrix}; \quad K_d = \begin{bmatrix}70 & 0\\ 0 & 50\end{bmatrix};$$

b. For a *three link* manipulator

$$K_p = \begin{bmatrix}100 & 0 & 0\\ 0 & 50 & 0\\ 0 & 0 & 450\end{bmatrix}$$

$$K_d = \begin{bmatrix}1.7 & 0 & 0\\ 0 & 0.8 & 0\\ 0 & 0 & 1.0\end{bmatrix}$$

c. RBNN Controller:

In RBNN controller, spread factor is taken as 2.

d. Optimized Neural Network Controller (ONNC):

In ONNC, for two-link and three-link manipulators, range of spread factor, is taken as 0.1 to 0.7. Size of population generated =30. Number of iterations are taken as 15 with w_{max}=0.9 and w_{min}=0.4. Values of c_1 and c_2 are changed from 2.5 to 0.5 and 0.5 to 2.5 respectively. *PSO is used to optimize (minimize) the mean square error (mse) during working.*

4.3 Disturbances

Robotic manipulator is inevitably subject to structured and unstructured uncertainties which are very difficult to model accurately and hence can hardly track the pre-defined trajectory. Motivated by this, goal of this paper is to develop a robust controller to achieve trajectory tracking of manipulator with two-link and three-link manipulator in presence of modeling uncertainties. After looking into lot many research papers, a wide range of disturbances covered in this paper are:

a. Fixed Disturbance: Fixed torque is inserted in the system as disturbance in all the controllers for joint angle 1 and joint angle 2. Value of fixed disturbance inserted in the system is

$$T_d = \begin{bmatrix}5\\ 5\end{bmatrix}$$

b. Continuous Disturbance: A high frequency continuously changing disturbance is inserted in the manipulator system to check out its effect. This continuous torque can be represented as

For two link manipulator

$$T_d = \begin{bmatrix}2\cos(5t)\\ 1.7\sin(5t)\end{bmatrix}$$

For three link manipulator

$$T_d = \begin{bmatrix}2\cos(5t)\\ 1.7\sin(5t)\\ \sin(5t)\end{bmatrix}$$

c. Uniform Random White Noise: Uniform Random White noise is a random signal with a flat (constant) power spectral density.

d. Lugre friction: The LuGre model is a dynamic friction model presented in [25] and can be modeled mathematically as in (13-15) given below:

$$\dot{z} = v - \frac{|v|}{g(v)} z \qquad (13)$$

$$F = \sigma_o z + \sigma_1 \dot{z} + \sigma_2 v \qquad (14)$$

$$g(v) = F_c + (F_s - F_c)\exp\left(\frac{v}{v_s}\right)^2 \qquad (15)$$

where z is average bristle deflection, σ_o is stiffness of bristles, σ_1 is bristle damping coefficient, σ_2 is viscous damping coefficient, v is relative velocity between moving parts, F_c is coulomb coefficient, F_s is static coefficient, v_s is striberk velocity.

4.4 Results

Mean Square Error (mse), maximum error (max) and the mean error (mean) are the criteria kept to check the accuracy of the controller. Comparative results are tabulated in tables (1-3) for two link manipulator for various disturbances and for various paths. Table (4) represents the errors for ONCC, RBNN and PD controllers for the path taken by incorporating various disturbances in the manipulator dynamics. It has been observed from all the four tables that in all the cases (without and with all the disturbances) ONCC has minimum error and hence can be declared as one of the best controller.

Trajectory tracked by the manipulator with various controllers has been given in Figs. [7-12] below. Figs. [8-10] represents the actual trajectory tracked with various controllers. These figures represents that accuracy in the path tracked is maximum with ONCC. RBNN controller has lesser accuracy than ONCC but show better results than a PD controller. Similar results have been observed in Figs. [11-13] which have the error graphs for various controllers. Tracking error is minimum for ONNC.

5. CONCLUSION

This paper is based on the fact that there is a huge rise in the area of research in intelligent and their hybrid controllers. ONNC in this paper is a hybrid controller. In ONCC, spread factor, s of RBNN is made self adjusting by optimizing s using PSO. ONCC has been compared with RBNN and PD controller. Results have been seen in the form of graphs and tables. These graphs represent the best performance of ONCC in presence of all the uncertainties. Better perform of ONCC can be seen from error comparison tables. All the calculated performance incides naming, mse, max and mean error are found to be lowest in ONCC. With this discussion it can be concluded that these intelligent hybrid ONCC is better than the RBNN and PD controller and can be used further for various complex, non linear control problems.

REFERENCES

[1]. Choi, Chung, Suh, Performance And H∞ Optimality Of PID Trajectory Tracking Control For Lagrangian System, *IEEE Transactions on Robotics and Automation,* 17(6), 2001.

[2]. S. V. Drakunov, V. I. Utkin, Sliding Mode Control in Dynamical Systems, *Int. J. Control,* 55(4), 1992, pp. 1029-1037.

[3]. G. Huang, C. Jiang, Y Wang, Adaptive Terminal Sliding Mode Control And Application For UASV Re-entry, *Control and Decision,* 22(11), 2007, pp. 1297-1301.

[4]. T. C. Kuo and Y. J. Huang, A Sliding Mode PID-Controller Design for Robot Manipulators, *IEEE International Symposium on Computational Intelligence in Robotics and Automation,* June 27-30, 2005,Finland, pp. 625-629.

[5]. N. Kapoor and J. Ohri, A Neural Network Based Novel Approach for Error Optimization in Path Tracking Control of a Robotic Manipulator, *National Conference, AEMDS-2013,* TERII, Kurukshetra, pp. 78-83.

[6]. The Berkeley Institute in Soft Computing. [Online]. Available: http://www-bisc.cs.berkeley.edu.

[7]. M. Zhihon, H. R. Wu, and M. Palaniswami, An adaptive tracking controller using neural networks for a class nonlinear systems, *IEEE Transactions on Neural Network,* vol. 9, no. 5, September 1998, pp. 947-955.

[8]. Y. Guo and P. Y. Woo, An adaptive fuzzy sliding mode controller for robotic manipulators, *IEEE Transactions on Systems, Man, and Cybernetics- Part A: Systems and Humans,* vol. 33, no. 2, March 2003, pp. 149-159.

[9]. W. Yu, A.S. Poznyak, E.N. Sanchez, Adaptive control of two-link manipulator via dynamic neural network, *Proceedings of the American Control Conference,* 1999, pp. 2516–2520.

[10]. E. Kim, Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic, *IEEE Transactions on Fuzzy Systems,* 12 (3), 2004, pp. 368–378.

[11]. F. L. Lewis, S. Jagannathan & A. Yesildirek, Neural Network Control of Robot Manipulators and Nonlinear Systems, *Taylor & Francis,* 1998.

[12]. C. Sudheer, R. Maheswaran, B.K. Panigrahi & S. Mathur, A Hybrid SVM-PSO Model for Forecasting Monthly Stream flow, *Neural Comput. & Applic.,* Feb, 2013.

[13]. J. Kennedy and R. Eberhart, Particle Swarm Optimization, *IEEE Int. Conf. Neural Networks,* 1995, pp. 1942-1948.

[14]. X. Yao, "Evolutionary Artificial Neural Networks", *Int. J. of Neural Systems,* 4(3), 1993, pp. 539-567.

[15]. H. Muhlenbein, Limitations of Multi-Layer Preceptron Networks- Steps Towards Genetic Neural Networks, *Parallel Computing,* vol. 14, 1990, pp. 249-260.

[16]. M W Spong, M Vidyasagar, Robot Dynamics and Control, Wiley-India Edition. New York.

[17]. S. Haykin, Neural Networks: A Comprehensive Foundation, *Macmillan College Publishing Company,* New York, 1994.

[18]. H. D. Patino and D. Liu, Neural network-based model reference adaptive control system, *IEEE Trans. Syst., Man, Cybem. B,* vol. 30, no. 1, Feb. 2000, pp. 198-204.

[19]. S. Seshagiri and H. K. Khail, Output feedback control of nonlinear systems using RBF neural networks, *IEEE Trans. Neural Networks.,* vol. 2, issue 1, Jan. 2000, pp. 69-79.

[20]. Y. Li, S. Qiang, X. Zhuang, and O. Kaynak, Robust and adaptive back-stepping control for nonlinear systems using RBF neural networks, *IEEE Trans. Neural Networks,* vol. 15, no. 3, May 2004, pp. 693-701.

[21]. Z. Song, J. Yi, D. Zhao, and X. Li, A computed torque controller for uncertain robotic manipulator systems: Fuzzy

approach, *Fuzzy Sets and Systems*, no. 154, 2005, pp. 208-226.

[22]. R. J. Wai and P. C. Chen, Robust neural-fuzzy-network control for robot manipulator including actuator dynamics, *IEEE Trans. Ind. Electronics*, vol. 53, no. 4, Aug. 2006, pp. 1328-1349.

[23]. J. Kennedy and R. Eberhart, Particle Swarm Optimization, *IEEE Int. Conf. Neural Networks*, 1995, pp. 1942-1948.

[24]. A. Ratnaweera, S. K. Halgamuge & H. C. Watson, Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients, *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, June 2004, pp: 240-255.

[25]. H. Olsson, K. J. Åström, C. Canudas de Wit, M. Gäfvert and P. Lischinsky, Friction Models and Friction Compensation, *European Journal of Control*, Dec. 1998, no.4, pp.176-195.

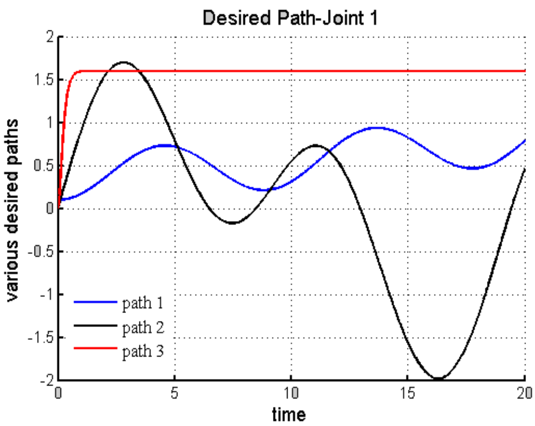

Fig. 1: Various Trajectories for Joint 1

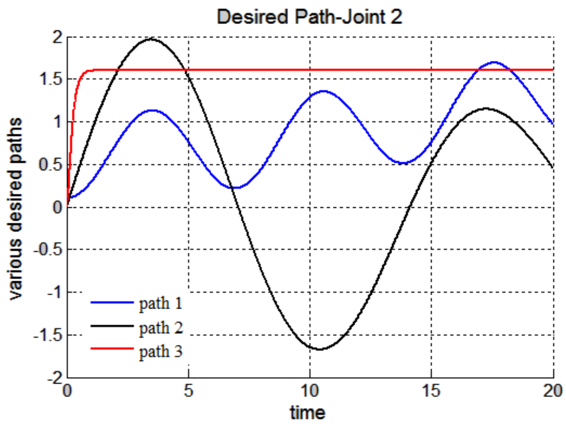

Fig. 2: Various Trajectories for Joint 2

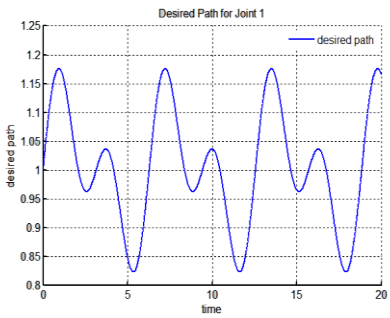

Fig. 3: Desired Trajectory for Joint 1.

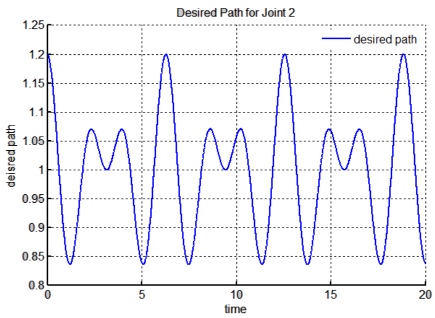

Fig. 4: Desired Trajectory for Joint 2.

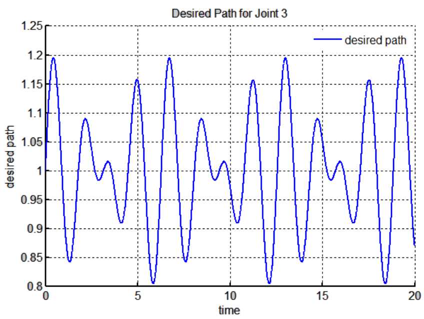

Fig. 5: Desired Trajectory for Joint 3.

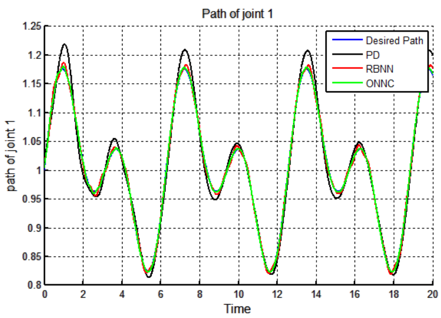

Fig. 7: Trajectory tracked by Joint 1.

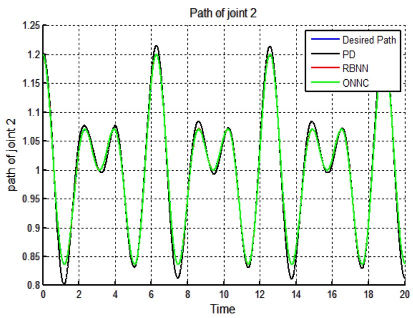

Fig. 8: Trajectory tracked by Joint 2.

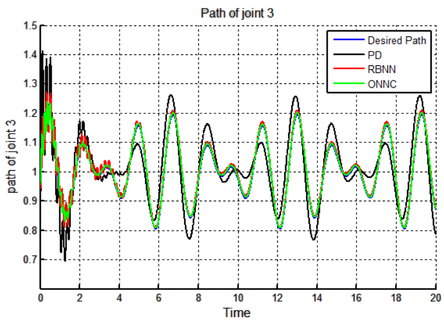

Fig. 9: Trajectory tracked by Joint 3.

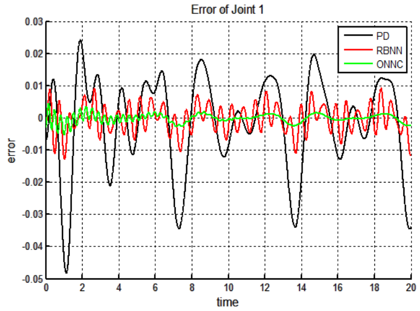

Fig. 10: Error in Joint 1(no disturbance).

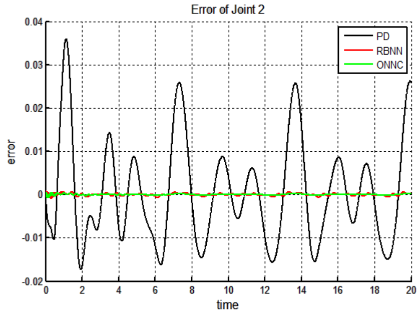

Fig. 11: Error in Joint 2(no disturbance).

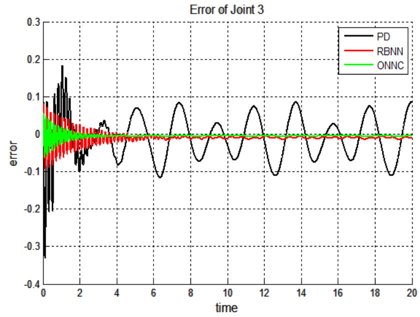

Fig. 12: Error in Joint 3 (no disturbance).

Table 1: Tracking errors for a two link manipulator for path 1.

path 1		Joint 1			Joint 2		
		mse	Max	mean	Mse	max	mean
no disturbance	PD	0.0039	0.0896	0.0045	0.0011	0.0472	0.0126
	RBNN	0.0025	0.0715	0.0037	0.0004	0.0317	0.0077
	ONCC	0.0025	0.0705	0.0037	0.0004	0.0308	0.0078
fixed disturbance	PD	0.0065	0.0249	0.0534	0.172	0.1002	0.4121
	RBNN	0.0043	0.0206	0.0435	0.0636	0.1002	0.2511
	ONCC	0.0043	0.0196	0.0435	0.0637	0.1002	0.2513
cont. Disturbance	PD	0.0039	0.0921	0.0046	0.0011	0.0519	0.0128
	RBNN	0.0025	0.0736	0.0037	0.0005	0.0371	0.0078
	ONCC	0.0025	0.0727	0.0037	0.0005	0.0343	0.008
white noise	PD	0.0039	0.0895	0.0045	0.0011	0.0472	0.0128
	RBNN	0.0025	0.0716	0.0037	0.0004	0.0323	0.0076
	ONCC	0.0025	0.0711	0.0037	0.0004	0.0313	0.0077
LuGre friction	PD	0.0051	0.1042	0.0063	0.0091	0.1405	0.0202
	RBNN	0.0033	0.0833	0.005	0.0036	0.0861	0.013
	ONCC	0.0034	0.0824	0.0048	0.0036	0.0912	0.0128
all disturbance	PD	0.0074	0.0422	0.0506	0.1335	0.1002	0.3541
	RBNN	0.0049	0.0351	0.041	0.0511	0.1002	0.2188
	ONCC	0.0049	0.0341	0.041	0.0533	0.1002	0.2238

Table 2: Tracking errors for a two link manipulator for path 2.

path 2		Joint 1				Joint 2	
		mse	Max	mean	mse	max	mean
no disturbance	PD	0.0028	0.078	0.0496	0.002	0.0578	0.0435
	RBNN	0.0018	0.0632	0.0403	0.0007	0.0351	0.0264
	ONCC	0.0018	0.0623	0.0403	0.0007	0.0312	0.0264
fixed disturbance	PD	0.0022	0.0016	0.0437	0.1353	0.0002	0.3659
	RBNN	0.0015	0.0584	0.0356	0	0.0102	0.0019
	ONCC	0.0015	0.0576	0.0356	0	0.0061	0.0018
cont. disturba nce	PD	0.0028	0.0796	0.0496	0.002	0.0639	0.0433
	RBNN	0.0018	0.0644	0.0403	0.0007	0.0405	0.0263

	ONCC	0.0018	0.0638	0.0403	0.0007	0.0367	0.0263
white noise	PD	0.0028	0.0782	0.0496	0.002	0.0589	0.0436
	RBNN	0.0018	0.0637	0.0403	0.0007	0.0362	0.0265
	ONCC	0.0018	0.0631	0.0403	0.0007	0.0321	0.0263
LuGre friction	PD	0.0038	0.0924	0.0579	0.0182	0.2108	0.1249
	RBNN	0.0024	0.0745	0.0461	0.007	0.1217	0.079
	ONCC	0.0025	0.0753	0.0474	0.005	0.1109	0.0652
all disturbance	PD	0.0033	0.0011	0.0537	0.2537	0.0002	0.5011
	RBNN	0.0024	0.0749	0.0464	0.0027	0.0864	0.0468
	ONCC	0.0017	0.0604	0.0384	0.0003	0.0212	0.0166

Table 3: Tracking errors for a two link manipulator for path 3.

path 3		Joint 1				Joint 2	
		mse	Max	mean	mse	max	Mean
no disturbance	PD	0.0093	0.3107	0.0918	0.0001	0.0725	0.0012
	RBNN	0.0061	0.296	0.0735	0.0001	0.0771	0.0002
	ONCC	0.0061	0.3007	0.0736	0.0001	0.0764	0.0001
fixed disturbance	PD	0.0015	0.2685	0.0263	0.1876	0.0841	0.4305
	RBNN	0.0012	0.022	0.2572	0.0657	0.2553	0.0593
	ONCC	0.0062	0.2956	0.0732	0.0001	0.0767	0.0001
cont. disturbance	PD	0.0092	0.3091	0.0918	0.0001	0.069	0.0011
	RBNN	0.0061	0.2945	0.0735	0.0001	0.0743	0.0001
	ONCC	0.0061	0.2947	0.0735	0.0001	0.0739	0
white noise	PD	0.0092	0.3105	0.0917	0.0001	0.0715	0.001
	RBNN	0.0061	0.2957	0.0734	0.0001	0.0766	0.0001
	ONCC	0.0061	0.296	0.0734	0.0001	0.0764	0
LuGre friction	PD	0.0199	0.3102	0.1363	0.0324	0.2596	0.1712
	RBNN	0.0136	0.2955	0.1119	0.0161	0.1869	0.1197
	ONCC	0.0061	0.2965	0.0734	0.0001	0.077	0
all disturbance	PD	0.0071	0.2665	0.0769	0.0416	0.0113	0.1812
	RBNN	0.0034	0.2555	0.0485	0.0171	0.0604	0.1255
	ONCC	0.0037	0.2554	0.0532	0.0148	0.0432	0.1131

Table 4: Tracking errors for a three link manipulator.

		Joint 1			Joint 2			Joint 3		
		mse	max	mean	mse	max	mean	mse	max	mean
no disturbance	PD	0	0.0073	0.0001	0	0.0015	0	0.0007	0.1115	0.005
	RBNN	0	0.0045	0.0001	0	0.0064	3.6 e-06	0.0001	0.0565	0.0049
	ONCC	0	0.0045	0.0001	0	0.0064	3.6 e-06	0.0001	0.0565	0.0049
fixed disturbance	PD	0.0026	0.0064	0.0502	0.0001	0.0068	0.01	0.0315	1.8755	0.055
	RBNN	0.0026	0.0032	0.0502	0.0001	0.0048	0.01	0.031	0.0341	0.045
	ONCC	0.0016	0.0022	0.0302	0	0.0028	0	0.021	0.031	0.019
cont.	PD	0	0.006	0.0001	0	0.001	0	0.0007	0.1115	0.005

	RBNN	0	0.004	0.0001	1 e-7	6.5 e-4	3.1 e-6	0.0001	0.0564	0.0049
	ONCC	0	0.0034	0.0001	1 e-8	7.4 e-4	2.2 e-6	0.00006	0.03	0.0041
white noise	PD	0	0.0072	0.0002	0	0.0065	0	0.0218	0.4707	0.005
	RBNN	0	0.0051	0.0003	0	0.0081	0	0.0001	0.056	0.0049
	ONCC	0	0.0048	0.0002	0	0.0066	0	0.0001	0.0564	0.0048
LuGre friction	PD	0	0.0075	0.0024	0	0.0017	0.0005	0.001	0.1282	0.0021
	RBNN	0	0.0086	0.0032	0	0.0015	0.0005	0.0002	0.0571	0.0022
	ONCC	0	0.0077	0.0032	0	0.0013	0.00032	0.0002	0.045	0.002
all disturbance	PD	0	0.0159	0.0018	0	0.0141	0.0009	0.154	1.2033	0.007
	RBNN	0	0.0086	0.0024	0	0.0178	0.0009	0.0002	0.0529	0.0074
	ONCC	0	0.0074	0.0022	0	0.0151	0.0007	0.0002	0.0547	0.0072