

Evolutionary Optimized Neural Network (EONN) Based Motion Control of Manipulator

Neha Kapoor

National Institute of Technology, Kurukshetra, Haryana, India

E-mail: ernehakapoor@rediffmail.com

Jyoti Ohri

National Institute of Technology, Kurukshetra, Haryana, India

E-mail: ohrijyoti@rediffmail.com

Abstract— In this paper, an Evolutionary Optimized Neural Network (EONN) based control scheme is proposed. This control scheme is based on the fact that optimizing values of a few parameters of neural network can enhance its control performance. Radial Biased Neural Network (RBNN) is chosen here and PSO, one of the most emerging global optimizing techniques, is used to optimize the parameters of a RBNN. From hidden to output layer RBNN uses Gaussian function for mapping. Spread factor (s) of this intelligent RBNN is then optimized by a modified PSO to improve its performance. The proposed controller has been verified by implementing it for position control of a robotic manipulator. For comparison purpose, proposed scheme has been verified with RBNN and the classical PD controller. MATLAB environment has been chosen for simulation study carried out. Robustness of the proposed controller has been checked by applying it to the manipulator for three different paths.

Index Terms— Radial Bias Neural Network (RBNN), Particle Swarm Optimization (PSO), Evolutionary Neural Network (ENN), Hybrid Intelligent Controller, Non-Linear Control Systems

I. INTRODUCTION

Recently, two important branches i.e. Evolutionary Algorithm (EA) and the Neural Network (NN) of intelligent control schemes has been combined. Combination of these two important branches of computational intelligence (CI) is necessary and inevitable [1]. In spite of a great nonlinear learning capabilities and good prediction of Neural Network [2] a list of its disadvantages includes high processing time, prone to generalization and tuning the parameters of the network by trial and error method; which is a time consuming and frustrating task. These limitations of NN restrict its use. Auto adaptability in the system can be one of the solutions and can be made in by using EC intelligent techniques [3]. 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 [4, 5]. Most recently in [6], ENN has been called as the next generation Neural Networks.

Recently, radial basis neural networks (RBNN) have been widely used for non-linear system identification,

owing to which RBFNNs have only one hidden layer and have fast convergence speed. Besides, the RBNN are often referred to as model-free estimators since they can be used to approximate the desired outputs without requiring a mathematical description of how the outputs functionally depend on the inputs. RBNN is also known to be a good universal approximator. When utilizing RBNN, some constant parameters must be determined first. A systematic way to determine the initial structure of RBNN must be established. Generally, these parameters are determined according to the experience of the designer or are just chosen randomly. However, such kind of improper initialization usually results in slow convergence speed and poor performance of the RBNN. In [7], Radial Bias Neural Network has been used to find the equivalent torque for the manipulator control system. Similarly, in the control problem defined in this particular problem, single input single output (SISO) RBNN has been proposed. Spread factor (s) is taken as an input to the RBNN and control torque for the system to be controlled is taken as output.

Stochastic based search algorithm PSO has been widely used in recent years to get the global optimal solutions [8]. PSO, developed by Kennedy and Elbert [9] in 1995, based on the simulation of simplified animal social behavior such as fish schooling, bird flocking etc.. 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 (1) & (2).

$$v_{id} = v_{id} + c_1 \epsilon_1 (p_{id} - x_{id}) + c_2 \epsilon_2 (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

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,

ϵ_1 and ϵ_2 are the random number generators.

In [10, 11] it has been proved that PSO finds the global best solution. PSO is becoming popular due to its

simplicity in implementation and ability to converge quickly to a reasonably good solution.

Robotic manipulator is a highly non-linear, time-varying and highly coupled system. Almost all kinds of control techniques have been discussed in literature [12, 13, 14 and references there in.] for a manipulator, but because of the non availability of accurate mathematical models of manipulator systems; still the thrust for an accurate and precise controller is there. Moreover, manipulator links are driven by actuators. Actuators are connected through either direct drive or gear train. Mainly three kinds of actuators are available: 1) Pneumatic 2) Hydraulic type and 3) Electric motors. Different actuators have different dynamics. Electric motors are more famous because of their light weight and high torque. Moreover controlling of electric motors is much easier than other two actuators. Therefore, for high speed applications usually electric actuators like direct current (DC) motors, induction motors, and brushless DC motors are used. Moreover, the dynamics of most motors remains non-linear and hence popular in robotic applications.

Further, the paper is organized as follows: Section II has system dynamics and problem formulation in it. Section III has of the concept of EONN controller and other i.e. RBNN & PD controllers. Section IV has simulation example and results. In last, conclusions are given in Section V.

II. SYSTEM DYNAMICS AND PROBLEM FORMULATION

Performance of the control technique proposed in this work has been checked for position control of a robotic manipulator system with actuator dynamics. In this section, manipulator and actuator dynamics have been given below as;

Manipulator Dynamics:

The dynamics of revolute joint type of robot can be described by following nonlinear differential equation given in (3) [15],

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (3)$$

with $q \in R^n$ 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.

Actuator Dynamics:

The dynamics of the armature controller dc motor that drives the link is given by the n decoupled equation.

$$J_M \ddot{\theta}_M + B \dot{\theta}_M + F_M + R_\tau = K_\theta \quad (4)$$

where θ_M is the rotor position angle. The control input is the motor voltage vector $\theta \in R^n$ the actuator coefficients matrixes are given by:

$$J_M = \text{diag}\{J_{m_i}\}$$

$$B = \text{diag}\{B_{m_i} + K_{b_i} K_{M_i} / R_{a_i}\} \quad (5)$$

$$R = \text{diag}\{r_i\}$$

$$K_M = \text{diag}\{K_{M_i} / R_{a_i}\}$$

where the i^{th} motor has inertia J_{m_i} , rotor damping constant B_{m_i} , back emf constant K_{b_i} , torque constant K_{M_i} and armature resistance R_{a_i} . The gear ratio of the coupling from the i^{th} motor to i^{th} arm link is r_i which is,

$$q_i = r_i \theta_{M_i} \quad (6)$$

The actuator friction vector is given by

$$F_M = \text{vec}\{F_{M_i}\} \quad (7)$$

where F_{M_i} is the friction of i^{th} rotor.

Using (7) to eliminate θ_M in (4) and then substituting τ from (3) results the manipulator dynamics in terms of joint variables

$$(J_M + R^2 M)\ddot{q} + (B + R^2 C)\dot{q} + (R F_M + R^2 F) + R^2 G = R K_M \theta \quad (8)$$

or by appropriate definition of symbols,

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) + F(q, \dot{q}) = \tau \quad (9)$$

The complete dynamics (9) has the same form as the robotic manipulator dynamics (3). The complete arm-plus-actuator dynamics enjoy the same properties as the arm dynamics, all the bound assumptions holds and linearity in the parameters holds. Thus in future work, all the upcoming advance intelligent controllers is used to check their performance with and without actuator dynamics. This is to check the robustness of the controllers.

Problem Formulation:

Control in trajectory tracking means “how do we get the output of the system to follow the input?”. The basic problem in motion control of manipulator is to determine the control inputs necessary to follow, or track a desired pre-planned trajectory. Main challenges in robot motion control problem is the complexity in dynamics and presence of uncertainties especially in high speed operations or while controlling direct-drive robots, for which no gear reduction is available to mask effective inertia variations. A general design for motion control of manipulator has been represented in Figure 1.

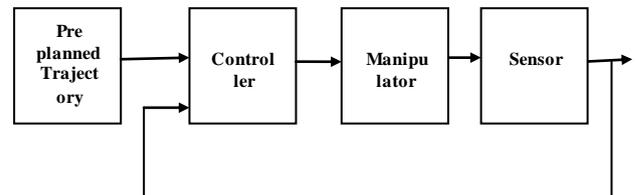


Fig. 1. General Design for Motion Control of Robotic Manipulator

III. PROBLEM SOLUTION AND CONTROL SCHEMES

Applied control schemes for the trajectory tracking problem of manipulator can be explained as below.

Acceleration coefficient of actual path covered is given in (10)

$$\ddot{q}_c = M^{-1}[\tau - V(q, \dot{q}) - G(q)] \quad (10)$$

Velocity coefficient (\dot{q}_c) and actual path (q_c) covered can be obtained as by backward integral method. Tracking error vector and error velocity be defined in (11) & (12) as

$$e = q - q_c, \in R_n \quad (11)$$

$$\dot{e} = \dot{q} - \dot{q}_c, \in R_n \quad (12)$$

A. *Proportional Derivative (PD) controller* in (13) is used for manipulator motion control.

$$\tau = K_p e(t) + K_d \dot{e}(t) \quad (13)$$

where, K_p and K_d are suitable positive definite diagonal $n \times n$ matrices. Values of the controller constants i.e. K_d and K_p are being decided by TAE (Trial And Error) method. Data generated in this controller is being stored and used further for the training of RBNN and EONN.

B. Radial Bias Neural Network (RBNN) controller

After learning, NN has a capability of generalization and then respond optimally to the unknown situations. Input torque given to the manipulator is very important. In this paper, a RBNN network is trained by using the data of a simple PD controller. Gaussian function is used as the activation function of each neuron in the hidden layer. The excitation values of this Gaussian function are distributed between the input values. The output of the network is given by (14) as

$$u = \sum_{j=1}^n w_j \exp \left[\frac{-\|s - c_j\|^2}{s^2} \right] \quad (14)$$

where j is the j^{th} neuron of the hidden layer, c_j is the central position of the neuron j and s is the spread factor of Gaussian function. This trained RBNNPD is used further to find the output torque to be given to the manipulator. Input given to the NN is spread factor (s) and output is the control torque (τ) to be given to robotic manipulator.

C. Evolutionary Optimized Neural Network (EONN)

Evolutionary Neural Networks refer to a special class and advance class of a neural networks in which evolution is used for adaptation along with the learning. List of advantages of EONN over NN includes [4]

- Avoidance if trapping in local minima of error function.
- The fitness function does not have to be differentiable or even continuous.
- Better performance in certain problems.
- Independence from initial conditions.

A general procedure of EONN is presented in the form of flowchart in Fig. 2 below. The proposed controller in this work can be explained as: spread factor (s) of RBNN is optimized by the improved PSO algorithm. Improvement in the PSO is implemented by varying the

accelerations coefficients (c_1 and c_2) in (1) with time (15) [16]

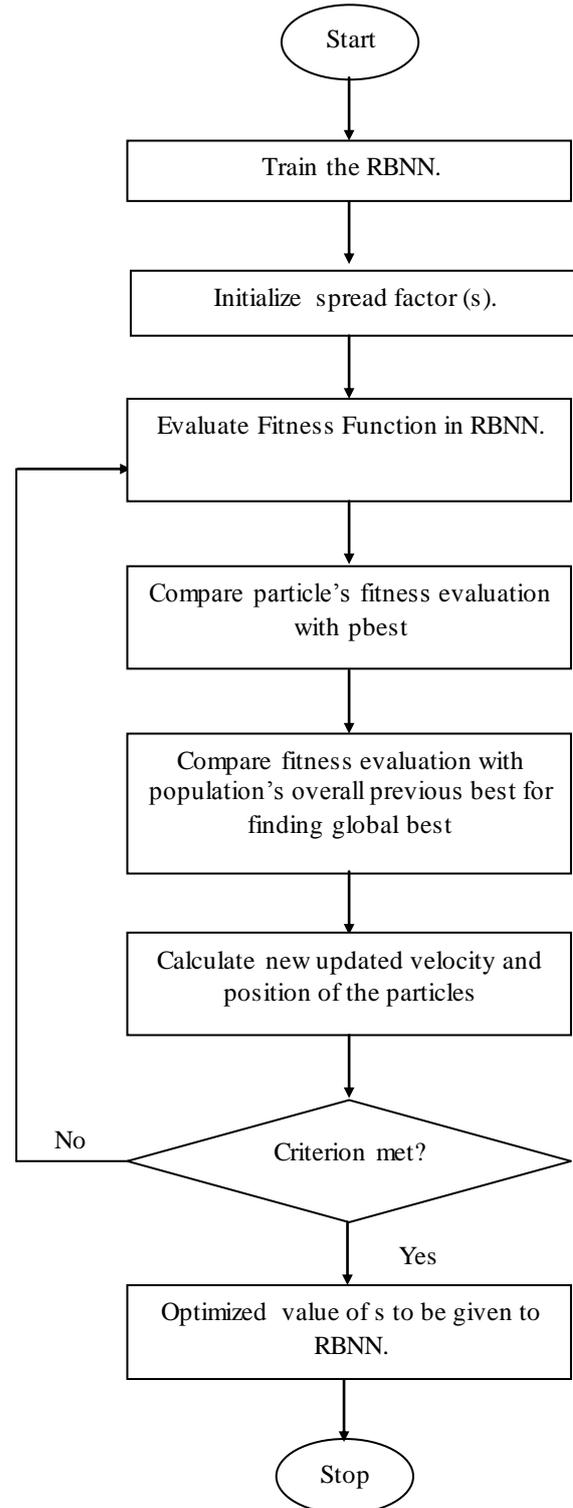


Fig. 2. Flowchart Representing Eonn Working.

$$c_1 = (c_{1f} - c_{1i}) \frac{\text{iter}}{\text{maxiter}} + c_{1i} \quad (15a)$$

$$c_2 = (c_{2f} - c_{2i}) \frac{\text{iter}}{\text{maxiter}} + c_{2i} \quad (15b)$$

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 (1) is given in (16)

$$w = (w_{\max} - w_{\min}) \left(\frac{\text{maxiter} - \text{iter}}{\text{maxiter}} \right) + w_{\min} \quad (16)$$

Fitness function used here is mean square error (MSE) to be minimized. A 2 dof manipulator with this controller is used to track three different trajectories given in (17-19) and represented in Figure 2 & 3.

1st trajectory

$$q_1 = 0.3 \sin(0.7t - \frac{\pi}{2}) + 0.3 \sin(0.1t - \frac{\pi}{2}) + 0.7 \quad (17a)$$

$$q_2 = 0.5 \sin(0.9t - \frac{\pi}{2}) + 0.5 \sin(0.1t - \frac{\pi}{2}) + 1.1 \quad (17b)$$

2nd trajectory

$$q_1 = \sin(0.67t) + \sin(0.3t) \quad (18a)$$

$$q_2 = \sin(0.39t) + \sin(0.5t) \quad (18b)$$

3rd trajectory

$$q_1 = 1.6 - 1.6 \exp(-8t) - 12.8 \exp(-8t) \quad (19a)$$

$$q_2 = 1.6 - 1.6 \exp(-8t) - 12.8 \exp(-8t) \quad (19b)$$

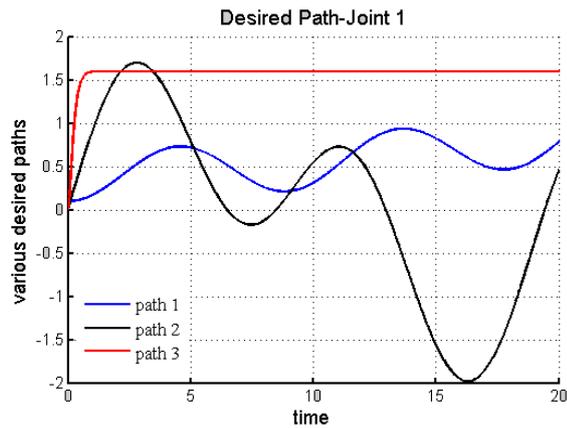


Fig. 3. Desired trajectories: Joint 1.

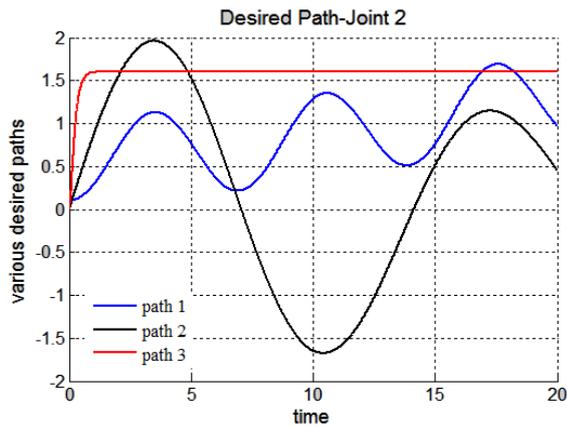


Fig. 4. Desired trajectories: Joint 2.

IV. SIMULATION EXAMPLE AND RESULTS

The dynamics of a 2 DOF robotic manipulator taken for this simulation study satisfying (3) is given as

$$M(q) = \begin{bmatrix} 8.77 + 1.02 * \cos(q_2) & 0.76 + 0.51 * \cos(q_2) \\ 0.76 + 0.51 * \cos(q_2) & 0.62 \end{bmatrix}$$

$$V(q, \dot{q}) = \begin{bmatrix} -0.51 * \sin(q_2) \dot{q}_2 & -0.51 * \sin(q_2) (\dot{q}_1 + \dot{q}_2) \\ -0.51 * \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} 74.48 * \sin(q_2) + 6.174 * \sin(q_1) + q_2 \\ 6.174 * \sin(q_1) + q_2 \end{bmatrix}$$

Parameters of the actuator dynamics discussed above in (5) are chosen as

$$\begin{aligned} J_{M_1} &= 2.85 \text{ kgm}^2, J_{M_2} = 1.4 \text{ kgm}^2, \\ K_{M_1} &= 19.44 \text{ Nm/A}, K_{M_2} = 12.28 \text{ Nm/A}, \\ R_{a_1} &= 1.2 \Omega, R_{a_2} = 1.3 \Omega, r_1 = r_2 = 1; \\ B_{M_1} &= 0.0573, B_{M_2} = 0.0253, \\ K_{b_1} &= K_{b_2} = 5 \text{ V/rad/sec}, F_{M_1} = F_{M_2} = 1.5 \end{aligned}$$

For this simulation study two cases have been taken into consideration:

Case 1: without actuator dynamics,

Case 2: with actuator dynamics.

Control gains parameters for PD controller are opted by TAE method as

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

In RBNN controller, by making about 50 trial runs of TAE methods spread factor is tuned to 2.

In proposed EONN controller, modified PSO parameters are taken as: range of spread factor as (0.1-0.7); Population size: 30; iterations: 50; $w_{\max} = 0.9$ and $w_{\min} = 0.4$; values of c_1 and c_2 are made to change from 2.5 to 0.5 and 0.5 to 2.5 respectively.

Results: Results of all the simulation studies for all the controllers applied on a non linear control problem of motion control of robotic manipulator has been given in Figures 5 & 6. Figure 5 (a-f) contains the trajectory tracked by three controllers for both the joints in Case 2 i.e. with actuator dynamics. It can be observed from the figure 5 that for joints 1 & 2 best trajectory is tracked by EONN. It can also be visualized that RBNN performance is proven to be better than classical PD controller. With PD controller tracking accuracy is least. Same type of results can be observed in Figure 6, which gives the continuous tracking errors. For non linear control problems performance evaluating criteria naming, transient error, steady state error, overshoot time etc. are not sufficient. Hence, for performance evaluations indices like mean error, mean square error (mse) and mean of input control torque have been chosen here. These performance evaluating factors have been tabulated in table 1. It can be observed from the results in the table that in both the cases (with and without actuator dynamics) the errors and control input torques have reduced when EONN controller is used. With actuator dynamics (case 2) errors in all the three trajectories have increased. Moreover, input control torque has also increased in Case 2.

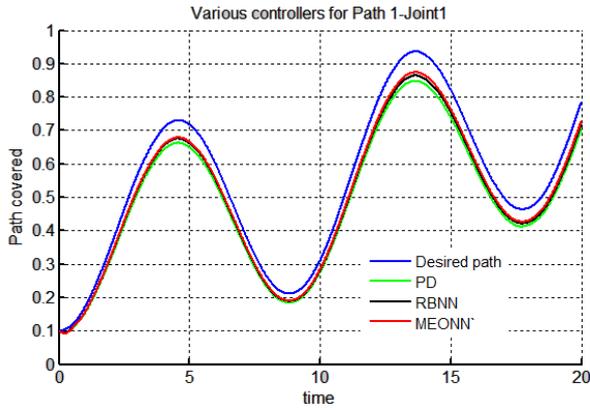


Fig. 5 (a) Trajectory 1: Joint 1.

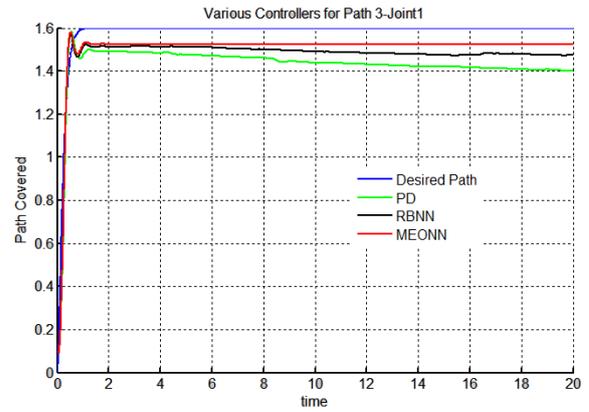


Fig. 5 (e) Trajectory 3: Joint 1.

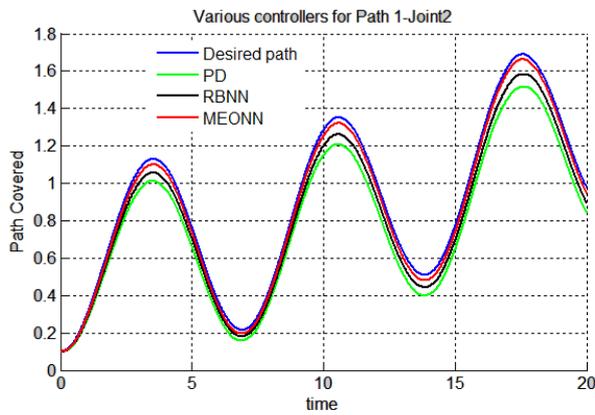


Fig. 5 (b) Trajectory 1: Joint 2.

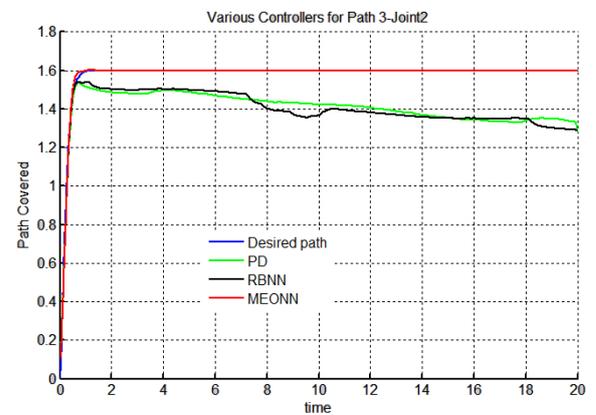


Fig. 5 (f) Trajectory 3: Joint 2.

Fig. 5. Position tracking performance of Joint 1 & 2 of various trajectories

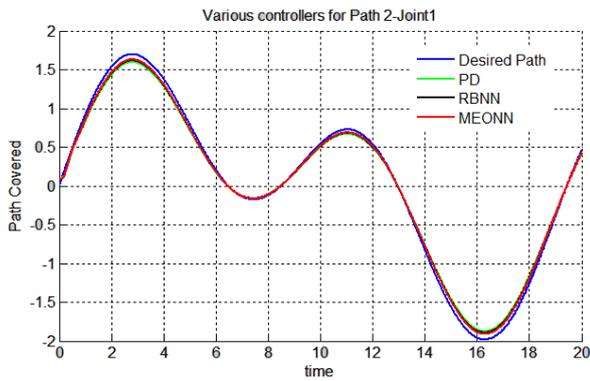


Fig. 5 (c) Trajectory 2: Joint 1.

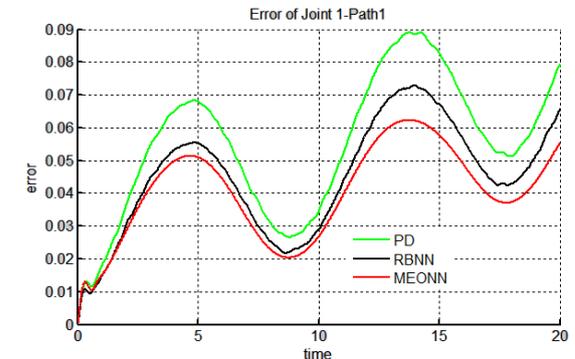


Fig. 6 (a) Trajectory 1: Joint 1.

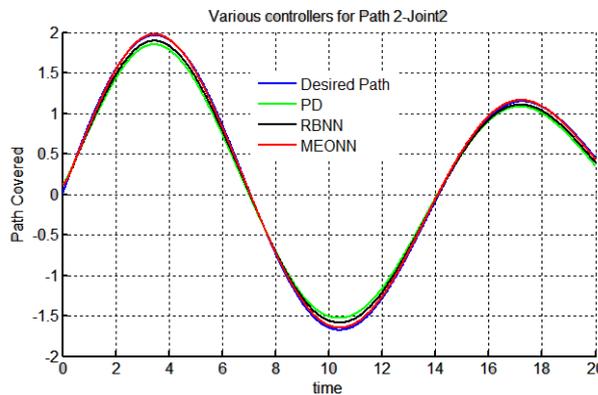


Fig. 5 (d) Trajectory 2: Joint 2.

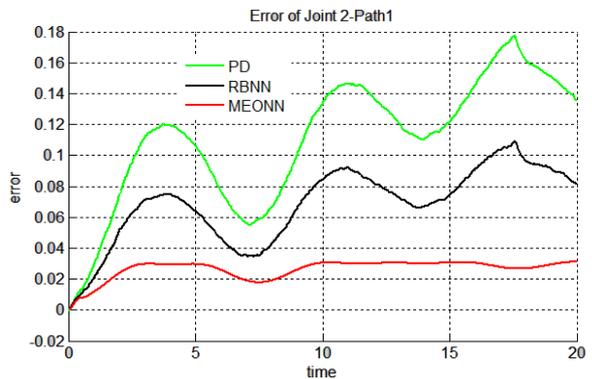


Fig. 6 (b) Trajectory 1: Joint 2.

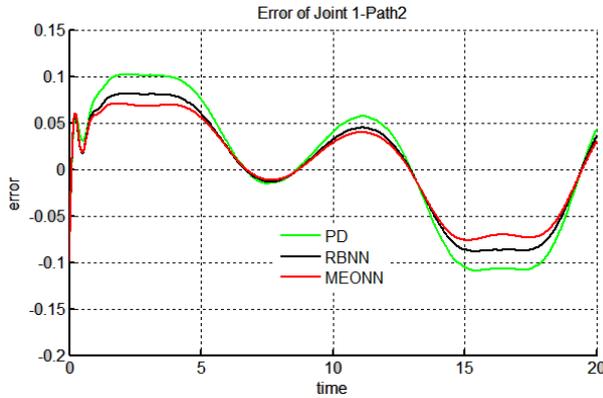


Fig. 6 (c) Trajectory 2: Joint 1.

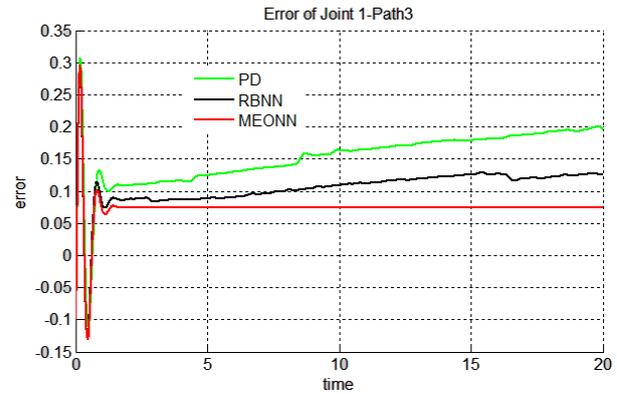


Fig. 6 (e) Trajectory 3: Joint 1.

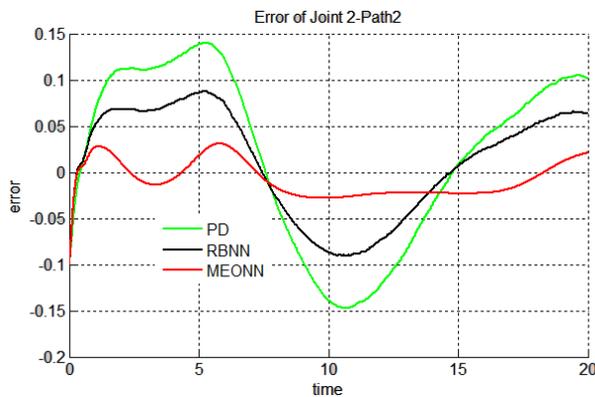


Fig. 6 (d) Trajectory 2: Joint 2.

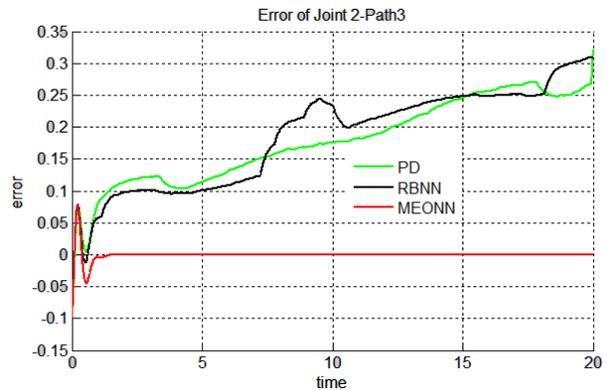


Fig. 6 (f) Trajectory 3: Joint 2.

Fig. 6. Tracking error Joint 1 & 2 of various trajectories

Table 1. Performance indices of Joint 1 & 2 for three different trajectories

| | | Case 1 | | | | | | Case 2 | | | | | |
|--------------|------|----------|-----------|----------|-----------|--------|---------|----------|-----------|----------|-----------|---------|---------|
| | | mse (e1) | mean (e1) | mse (e2) | mean (e2) | tau 1 | tau 2 | mse (e1) | mean (e1) | mse (e2) | mean (e2) | tau 1 | tau 2 |
| Trajectory 1 | PD | 0.0034 | 0.0548 | 0.0139 | 0.1107 | 19600 | 946 | 0.021 | 0.099 | 0.033 | 0.34 | 28078 | 1289 |
| | RBNN | 0.0023 | 0.045 | 0.0053 | 0.0685 | 19000 | 923.89 | 0.011 | 0.65 | 0.021 | 0.099 | 20875 | 1007.98 |
| | EONN | 0.0018 | 0.0403 | 7.33e-4 | 0.0264 | 192000 | 924 | 0.011 | 0.521 | 0.0023 | 0.0432 | 17087 | 980.99 |
| Trajectory 2 | PD | 0.0051 | 0.0046 | 0.0092 | 0.0241 | 25700 | 244.9 | 0.0087 | 0.016 | 0.023 | 0.043 | 35767 | 319.86 |
| | RBNN | 0.0033 | 0.0037 | 0.0036 | 0.0109 | 373.02 | 241.9 | 0.006 | 0.009 | 0.02 | 0.032 | 30876 | 278.9 |
| | EONN | 0.0025 | 4.40E-4 | 0.0037 | -0.0078 | 334.75 | 222.99 | 0.003 | 0.00066 | 0.008 | 0.01 | 31098 | 243.7 |
| Trajectory 3 | PD | 0.0248 | 0.1514 | 0.0357 | 0.177 | 466.77 | 196.7 | 0.055 | 0.034 | 0.088 | 0.54 | 564.32 | 876.88 |
| | RBNN | 0.0121 | 0.105 | 0.04 | 0.1841 | 410 | 134.7 | 0.0345 | 0.23 | 0.1 | 0.34 | 410 | 576 |
| | EONN | 0.0061 | 0.0735 | 8.6E-5 | 9.09E-5 | 4.4E+2 | 1.66E+2 | 0.01 | 0.001 | 0.0045 | 0.0009 | 2.45E+2 | 3.57E+2 |

V. CONCLUSION

In this work, intelligent hybrid controller, having a modified PSO to optimize RBNN parameters is proposed. This special class of hybrid controller is named as evolutionary optimized neural network (EONN). Validity of the controllers is checked by implementing it on a 2 dof manipulator. Robustness of the control scheme is checked by taking actuator dynamics into consideration. It has been concluded from the observation made by the results of the simulation study that the proposed EONN controller is giving the best results in terms of minimum

errors and reduced required control torque when compared with the classical PD and RBNN controller. Robustness of the controller is checked by implementing the same controller on three different paths. Hence, one can conclude that the proposed EONN control technique has good performance for non-linear control systems. Further improvements in the work can be made by using improvised versions of PSO [17].

REFERENCES

[1] Yingqi C., Songyu Y. and Yi Z., "Combination of Evolutionary Computational and Artificial Neural

- Network". *Infrared and Laser Engineering*, 28(4): 6-91, 1999.
- [2] Lewis F. L., Jagannathan S. and Yesildirek A., *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor & Francis, 1998.
 - [3] The Berkeley Institute in Soft Computing. [Online]. Available: <http://www-bisc.cs.berkeley.edu>.
 - [4] Yao X., *Evolutionary Artificial Neural Networks*. *Int. J. of Neural Systems*, 4(3): 539-567, 1993.
 - [5] Muhlenbein H., *Limitations of Multi-Layer Preceptron Networks-Steps Towards Genetic Neural Networks*. *Parallel Computing*, (14): 249-260, 1990.
 - [6] Baluja S., *Evolution of Artificial Neural Network Based Autonomous Land Vehicle Controller*. *IEEE Trans. On SMC*. (26): 450-463, 1996.
 - [7] Kuo T. C. and Huang Y. J., *Global Stabilization of Robot Control with Neural Network and Sliding Mode*. *Engineering Letters*, 16(1): EL_16_1_09.
 - [8] Sudheer C., Maheswaran R., Panigrahi B.K. and Mathur S., *A Hybrid SVM-PSO Model for Forecasting Monthly Streamflow*. *Neural Comput. & Applic.*, DOI 10.1007/s00521-013-1341-y. Feb, 2013.
 - [9] Kennedy J. and Eberhart R., *Particle Swarm Optimization*. *IEEE Int. Conf. Neural Networks: 1942-1948*, 1995.
 - [10] Clerc M., *The Swarm and the Queen: Toward a Deterministic and Adaptive Particle Swarm Optimization*. *IEEE Int. Congr. Evolutionary Computation*, vol. 3: 1957, 1999.
 - [11] Clerc M. and Kennedy J., *The Particle Swarm- Explosion, Stability and Convergence in a Multi-Dimensional Complex Space*. *IEEE Trans. Evol. Comput.*, vol. 6: 58-73, Feb. 2002.
 - [12] Kapoor N. and Ohri J., *Various Methods of a Manipulator Control: Trending Towards Future*. *National Conference on Advances in Videos, Cyber Learning and Electronics (ADVCE 2012)*, at NITTTR, Chd.: 19, March 2012.
 - [13] Kapoor N. and Ohri J., *A Neural Network Based Novel Approach for Error Optimization in Path Tracking Control of a Robotic Manipulator*. *National Conference, AEMDS-2013*, at TERII, Kurukshetra: 98-103, 2013.
 - [14] Kapoor N. and Ohri J., *Fuzzy Sliding Mode Controller (FSMC) with Global Stabilization and Saturation Function for Tracking Control of a Robotic Manipulator*. *Journal of Control and Systems Engineering*. 1(2): 50-56, 2013.
 - [15] Spong M W, Vidyasagar M., *Robot Dynamics and Control*. Wiley-India Edition. New York.
 - [16] Ratnaweera A., Halgamuge S. K. and Watson H. C., *Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients*. *IEEE Trans. on Evol. Comput.*, 8(3): 240-255, June 2004.
 - [17] Khan K. and Sahai A. "A Glowworm Optimization Method for the Design of Web Services": *I.J. Intelligent Systems and Applications*, 10, 89-102, 2012.

Authors' Profile:

Neha Kapoor has received B. Tech. degree in Electrical Engineering in 2002 and masters degree from Panjab University in 2009. She is a research scholar of Ph.D from National Institute of Technology, Kurukshetra. Her research areas are non-linear control systems, artificial intelligence, robust controllers.

Jyoti Ohri is currently working as Professor in Electrical Engineering Department in National Institute of Technology, Kurukshetra. She completed her Doctorate, Masters and Bachelor degree in Electrical Engineering from National Institute of Technology, Kurukshetra, India. Her areas of interests are Control System, Adaptive and Robust Control Systems, Instrumentation and Control, Robotics and Intelligent control systems.

How to cite this paper: Neha Kapoor, Jyoti Ohri, "Evolutionary Optimized Neural Network (EONN) Based Motion Control of Manipulator", *International Journal of Intelligent Systems and Applications(IJISA)*, vol.6, no.12, pp.10-16, 2014. DOI: 10.5815/ijisa.2014.12.02