International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:2, No:5, 2008

# Hybrid GA Tuned RBF Based Neuro-Fuzzy Controller for Robotic Manipulator

Sufian Ashraf Mazhari and Surendra Kumar Member IEEE

***Abstract*—**In this paper performance of Puma 560 manipulator is being compared for hybrid gradient descent and least square method learning based ANFIS controller with hybrid Genetic Algorithm and Generalized Pattern Search tuned radial basis function based Neuro-Fuzzy controller. ANFIS which is based on Takagi Sugeno type Fuzzy controller needs prior knowledge of rule base while in radial basis function based Neuro-Fuzzy rule base knowledge is not required. Hybrid Genetic Algorithm with generalized Pattern Search is used for tuning weights of radial basis function based Neuro- fuzzy controller. All the controllers are checked for butterfly trajectory tracking and results in the form of Cartesian and joint space errors are being compared. ANFIS based controller is showing better performance compared to Radial Basis Function based Neuro-Fuzzy Controller but rule base independency of RBF based Neuro-Fuzzy gives it an edge over ANFIS

***Keywords*—**Neuro-Fuzzy, Robotic Control, RBFNF, ANFIS, Hybrid GA.

## I. Introduction

THE Integrated Neuro-fuzzy system combines the advantages of ANN and FIS. While the learning capability is an advantage from the viewpoint of FIS, the formation of linguistic rule base will be an advantage from the viewpoint of ANN. Integrated Neuro-fuzzy systems share data structures and knowledge representations. A common way to apply a learning algorithm to a fuzzy system is to represent it in a special ANN like architecture. However the conventional ANN learning algorithms (gradient descent) cannot be applied directly to such a system as the functions used in the inference process are usually non differentiable. This problem can be tackled by using differentiable functions in the inference system or by not using the standard neural learning algorithm.

FALCON [1] uses a five-layered architecture with hybrid-learning algorithm comprising of unsupervised learning to locate initial membership functions/rule base and a gradient descent learning to optimally adjust the parameters of the membership function to produce the desired outputs. GARIC [2] implements a Neuro-Fuzzy controller by using two neural network modules, the ASN (Action Selection Network) and the AEN (Action State Evaluation Network) .ANFIS [3] implements a Takagi Sugeno FIS and has a five layered architecture. ANFIS uses hybrid backpropogation and least square method for learning. NEFCON [4] is designed to implement Mamdani type FIS and uses a mixture of reinforcement and backpropagation learning. A Neuro-Fuzzy methodology based on radial basis function and tuned with Genetic Algorithm is implemented in [5].

Robot manipulator faces uncertainties in their dynamics, such as payload mass, friction, and disturbance. Therefore, it is difficult to obtain an accurate model for manipulators. Thus, model based control systems may not be easily implemented in manipulators control. A new hybrid direct/indirect adaptive FNN controller with state observer and supervisory controller for a class of uncertain nonlinear dynamic systems is implemented in [6]. A robust adaptive fuzzy neural controller (AFNC) for identification and control of a class of uncertain multiple-input–multiple-output (MIMO) nonlinear systems is developed in [7]. A robust adaptive fuzzy neural controller (AFNC) suitable for motion control of multilink robot manipulators is implemented in [8]. A fast online structure and parameter learning algorithm, which can add or delete fuzzy control rules or neural network nodes automatically and systematically without predefinition is proposed in [9-10].

In this paper a systematic approach for designing Neuro-Fuzzy controller is developed .Starting with PID controller, Takagi-Sugeno (TS) type Fuzzy PID controller is designed. From TS type Fuzzy PID, ANFIS and Radial basis Function based Neuro-Fuzzy (RBFNF) is designed. In section II, modeling of dynamics and kinematics is discussed. Section III deals details of ANFIS based controller design. Hybrid Genetic Algorithm tuned RBF based Neuro-Fuzzy is presented in Section IV. Section V and Section VI deal with results and conclusion.

Sufian Ashraf Mazhari is working in GS E&C Gurgaon , India (Email: sufian.ashraf@gmail.com)
Surendra Kumar is Assistant Professor in the Department of Electrical Engineering, Indian Institute of Technology, Roorkee, Uttrakhand 247667, India (Email: surendra_iitr@yahoo.com)

## II. Modelling of Manipulator Dynamic and Kinematics

The dynamics of an n-link robotic manipulator is characterized by a set of highly nonlinear and strongly coupled second order differential equation.

$$D(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) + G(\theta) + F(\dot{\theta}) = \tau \quad (1)$$

where $D(\theta)$ is the nxn inertial matrix, $C(\theta,\dot{\theta})$ is the nx1 vector of centrifugal forces, $G(\theta)$ is the nx1 vector of gravity loading, $F(\dot{\theta})$ is nx1 vector of friction term. $\theta,\dot{\theta}$ and $\ddot{\theta}$ are nx1 vector for

joint angular position, velocity and acceleration, $\tau$ is nx1 joint torque vector. D, C, G, F are very complicated function of $\theta$ and $\dot{\theta}$. The dynamic parameters of Puma 560 have been taken from [11]. Puma 560 joint actuators are DC servo motors with armature voltage as control input. The motor is connected to manipulator links through gear where the Robot dynamics appears as dynamic load. The dynamics of DC motor can be represented as (2-5)

$$E_a = E_b + L\frac{dI}{dt} + RI \quad (2)$$

$$E_b = K_e N\Omega \quad (3)$$

$$I = E_a - K_e N\Omega / Ls + R \quad (4)$$

$$\tau = K_m I \quad (5)$$

Where $E_a$ is the armature voltage, $E_b$ the Back e.m.f, L and R are inductance and reactance of armature windings respectively, I is the armature current, N is gear ratio, $K_e$ is the back e.m.f constant, $K_m$ is motor constant and $\Omega$ is load angular velocity. Actuator data of puma 560 Robot is taken from [12]. The transformation between the joint space and the Cartesian space of the robot is very important since robots are controlled in the joint space, whereas tasks are defined and object manipulated in the Cartesian space. The kinematics problem deals with the analytical study of the relation between these two spaces. The direct kinematics defined as the transformation from the joint space to the Cartesian space and the inverse kinematics defined as the transformation from the Cartesian space to the joint space. While modeling the kinematics of manipulator, arm singularity and configuration must be checked. Many methods have been proposed for better and feasible solution of manipulator kinematic problems [13-20]. The Forward and Inverse kinematic equations have been modeled [16] and are given in appendix A. Control system diagram of Puma 560 is shown in Fig.1 which consists of desired Cartesian space trajectory T, inverse kinematics block I, PID controller, servo motor M, dynamics D and forward kinematics Block F. In this paper PID controller in Fig.1 is being replaced with Neuro-Fuzzy Controller.

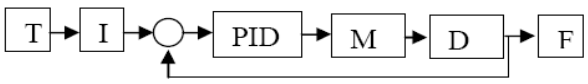


Fig. 1 Block diagram representation of Puma 560 control

Simulink model of Forward dynamics is shown in Fig.2 and simulink model of complete system is shown in Fig.3.

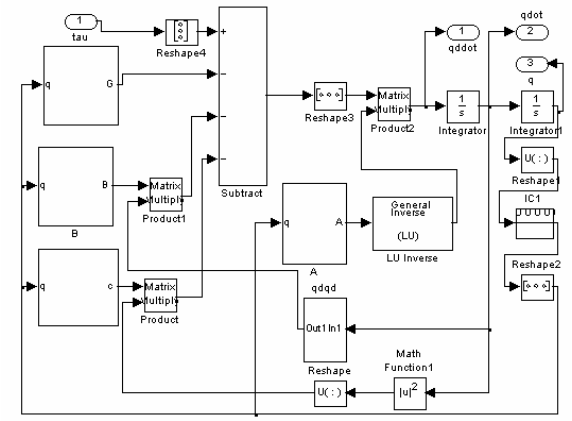


Fig. 2 Simulink diagram of PUMA560 dynamics

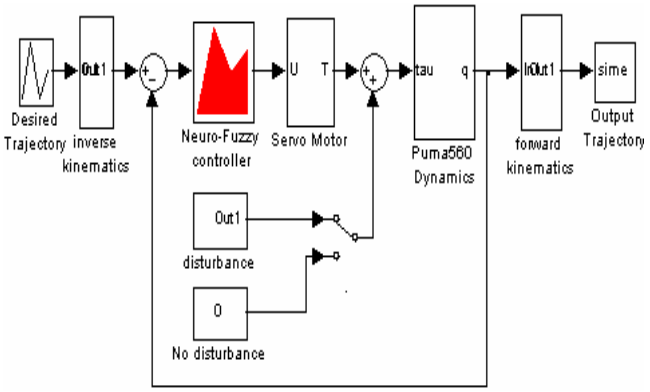


Fig. 3 Simulink model of complete system

## III. Adaptive Neuro-Fuzzy Inference System Based Controller

ANFIS implements a Takagi Sugeno FIS and has a five layered architecture. The first hidden layer is for Fuzzification of the input variables and T-norm operators are deployed in the second hidden layer to compute the rule antecedent part. The third hidden layer normalizes the rule strengths followed by the fourth hidden layer where the consequent parameters of the rules are determined. Output layer computes the overall input as the summation of all incoming signals. ANFIS uses back propagation learning to determine premise parameters (to learn the parameters related to membership functions) and least mean square estimation to determine the consequent parameters. The first step involved in designing ANFIS based controller is creation of Takagi-Sugeno FIS.A TS type Fuzzy PD+I controller is designed and data is collected from fuzzy controller for training of ANFIS.Fuzzy controller can be designed using parameters of crisp PID controller. Fuzzy PID controller is implemented as Fuzzy PD+I controller. The inputs to fuzzy controllers are error and error change. Matlab simulation diagram of fuzzy PD+I Controller is shown in Fig.4.

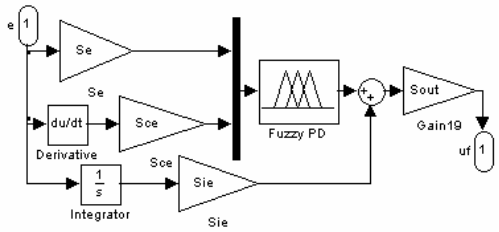


Fig. 4 Matlab simulation diagram of Fuzzy PD+I

Important steps involved in designing fuzzy controller are rule base generation and input output gains setting. If $\theta_d(k)$ is desired joint angle and $\theta(k)$ is actual output angle at any sampling instant k, error *e(k)*, change in error $\dot{e}(k)$ and integral error *ie(k)* are given as

$$e(k) = \theta_d(k) - \theta(k) \tag{6}$$

$$\dot{e}(k) = \frac{e(k) - e(k-1)}{T_s} \tag{7}$$

$$ie(k) = \sum_{k=1}^{n} \dot{e}(k) T_s \tag{8}$$

For classical PD controller the controller output is given as

$$u(k) = k_p \left( e(k) + T_d \dot{e}(k) \right) \tag{9}$$

Where $k_p$ is gain of classical PD controller, $T_d$ is derivative time constant and *u (k)* is control signal

When actuating signal *u(k)* is equal to zero

$$k_p \left( e(k) + T_d \dot{e}(k) \right) = 0 \tag{10}$$

$$\dot{e}(k) = -\frac{1}{T_d} e(k) \tag{11}$$

From (10) it is clear that $\dot{e}(t)$ directly depends upon $T_d$. If state trajectory of the closed loop controlled system with PD controller for some constant PD value is plotted, it draws a sharp boundary between positive and negative control signals. This can be used to map rule base in discrete state space by taking the diagonal element of rule base as ZE. Rule base for fuzzy controller is given in Table I.

Crisp PID controller parameters are used to initially set fuzzy input output gains. Input error scaling factor is $S_e$, error change scaling factor is $S_{ce}$ and output scaling factor is $S_{out}$

TABLE I
RULE BASE FOR FUZZY CONTROL

| $e \setminus \dot{e}$ | NB | NM | NS | ZE | PS | PM | PB |
|---|---|---|---|---|---|---|---|
| PB | ZE | PS | PM | PB | PB | PB | PB |
| PM | NS | ZE | PS | PM | PB | PB | PB |
| PS | NM | NS | ZE | PS | PM | PB | PB |
| ZE | NB | NM | NS | ZE | PS | PM | PB |
| NS | NB | NB | NM | NS | ZE | PS | PM |
| NM | NB | NB | NB | NM | NS | ZE | PS |
| NB | NB | NB | NB | NB | NM | NS | ZE |

the fuzzy controller output $u_f$ is given as

$$u_f = \left( S_e e(k) + S_{ce} \dot{e}(k) + S_{ie} ie(k) \right) S_{out}$$

TABLE II
INITIAL VALUE OF SCALING FACTORS

| Joint | $S_{e0}$ | $S_{ce0}$ | $S_{ie0}$ | $S_{out0}$ |
|---|---|---|---|---|
| 1 | 0.1 | 0.009 | 0.4 | 2109.375 |
| 2 | 0.1 | 0.0084 | 0.4267 | 2400 |
| 3 | 0.1 | 0.0084 | 0.4267 | 2400 |
| 4 | 0.1 | 0.0056 | 0.64 | 5400 |
| 5 | 0.1 | 0.0056 | 0.64 | 5400 |
| 6 | 0.1 | 0.0056 | 0.64 | 5400 |

(12)

Comparing (11) with the crisp PID controller output, values of scaling factors come out to be

$$S_e S_{out} = k_p \qquad S_{ce} / S_e = T_d \qquad S_{ie} / S_e = 1/T$$

If maximum probable error for any joint is $e_{max}$ and input /output membership function universe is taken as [-1 1], the error scaling factor $S_e$ can be set to $1/e_{max}$ .Error change and output scaling factor will be

$$S_{out} = k_p e_{max} \qquad S_{ce} = T_d / e_{max} \qquad S_{ie} = 1 / e_{max} T_i$$

Since better trajectory always starts from the current position of joint angle, initial tracking angle is zero. Taking a worst condition error of 10 radians. value of error scaling factor $S_{e0}$ is set to 0.1 and all other initial scaling factors $S_{ce0}, S_{ie0}, S_{out0}$ are calculated using values of classical PID parameters taken from [20] as given in Table II .For all six joints structure of Fuzzy controller is same as that of Fig.4 but with different scaling factors as given in Table II.

Data set collected from fuzzy controller is used for training ANFIS. Structure of ANFIS used is shown in Fig.5. Membership function after training is shown in Fig.6 and Fig.7.The designed ANFIS is substituted in place of Fuzzy block in Fig.4 keeping gains as it is. The Control system diagram with ANFIS is shown in Fig.8.

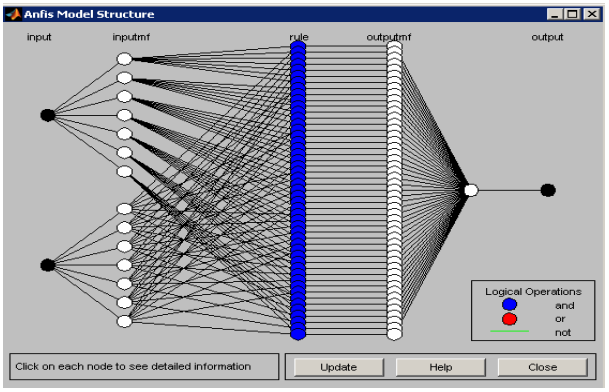


Fig. 5 structure of ANFIS

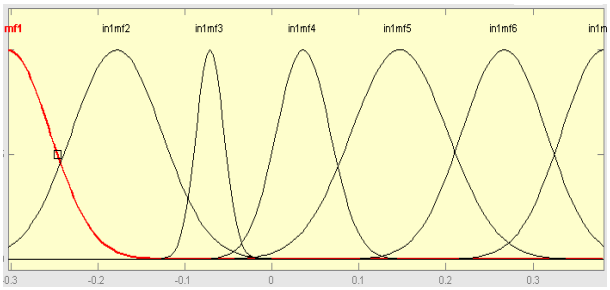

Fig .6 Learned membership function of error using ANFIS

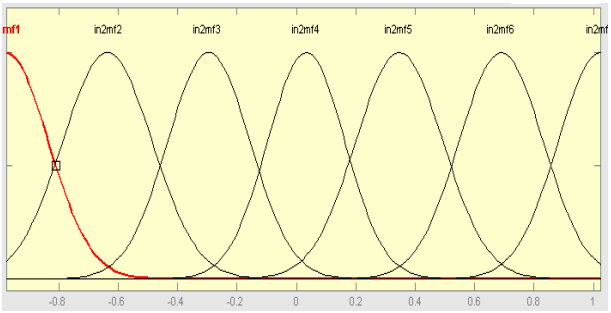

Fig. 7 Learned membership function of error change using ANFIS

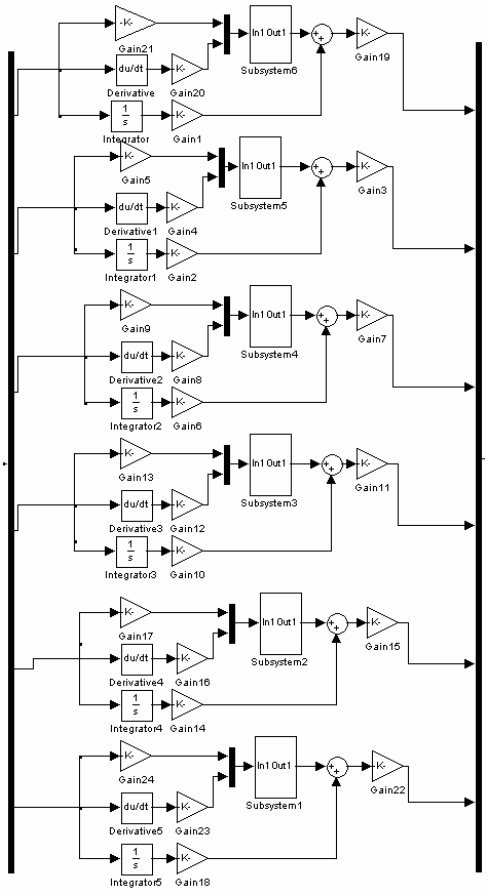

Fig. 8 Control system block diagram with ANFIS

## IV. Hybrid Genetic Algorithm Tuned Radial Basis Function Based Neuro Fuzzy Controller

The RBF neural network (RBFNF) is usually used to approximate a continuous linear or nonlinear function mapping. The structure of the two-input and single output RBFNF is shown as in Fig.9. The input layer accepts the system state feedback ($e,\dot{e}$) and the fuzzy inferencing is processed at the hidden layer. The strength of the control action for each of the fuzzy rules is given by the interconnected weights between the hidden and the output layers. The output layer implements the normalization operation to produce the control signals ($u_{nf}$).Basically, fuzzy logic control involves three main stages: Fuzzification, inferencing, and defuzzification. This fuzzy inference mechanism can be further simplified to as only pattern matching and weights averaging, thereby, eliminating the procedures of Fuzzification and defuzzification. The first operation deals with the IF part of the fuzzy control rules; it determines the matching degree of the current input to the condition of each of the fuzzy control rules. By characterizing the fuzzy input membership functions with only two parameters ($C_x$ and $D_x$), and using the Gaussian membership functions, the matching formula can be written as follows:

$$h_i = exp\left( -\left\| \frac{C^i_{x,n} - x_n}{D^i_{x,n}} \right\|^2 \right) \qquad (12)$$

For *i=1* to *T*.

Here T is the total number of fuzzy rules $C^i_{x,n}$ and $D^i_{x,n}$ denotes the center and the width of n[th] input variable's membership assigned to the ith control rules, respectively. While $\|\bullet\|$ is the norm operator presented as Euclidean distance. The matching degree process is simply an operation that returns the matching level between the inputs and the rule pattern for the ith rule. A matching degree of '1' means that a full match occurs to that rule, while a small $h_i$ indicates poor matching between the input pattern and the particular rule pattern. The weights are then averaged to obtain the control action of each output variable. Thus controller output $u_{nf}$ can be computed by normalizing the weights

$$u_{nf} = \sum_{i=1}^{p} (h_i . w_{im}) \Big/ \sum_{i=1}^{T} (h_i) \qquad (14)$$

From ANFIS block diagram shown in Fig.8 only fuzzy block is replaced with Radial basis function based Neuro-Fuzzy block keeping all the gains same .For each of 49 rules a hidden layer neuron is taken. Each of the node consists of two Gaussian functions with there centre and width as that of error and error change in antecedent part of rule base to implement (6) as shown in Fig.10.

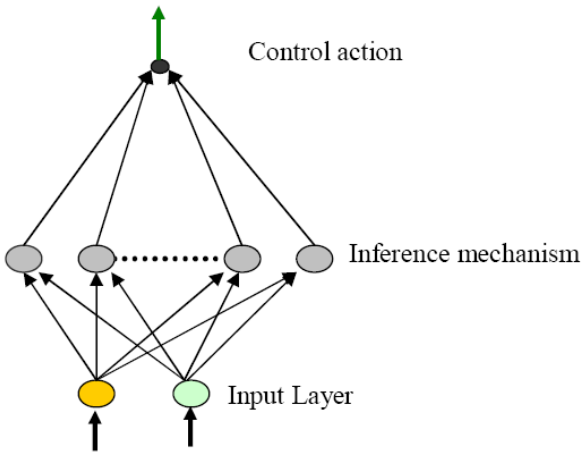


Fig. 9. Two-input and single output RBFNF

Final control output $u_{nf}$ is calculated using (7) as shown in Fig.11. For 49 hidden layer neurons there are 49 weights between hidden layer and output .Seven membership Gaussian functions for error and error change give 28 variables as each of them have there mean and variance. So total of 77 variables comes in action. GA was used initially to search the optimum search space using all these 77 variables. Neuro-Fuzzy block is attached with first joint of fuzzy controller and goal of GA was to minimize integral absolute error IAE between fuzzy output and Neuro-Fuzzy output. Fitness curve of GA shown in Fig.12. After tuning process of GA ,control using Neuro-Fuzzy started and GPS was used to search mean and variance (28 variables keeping weight same as that returned by GA) to minimize ITSE of system. After 30 iterations of GPS, variance is kept same and now only mean is varied using GPS. Tuning curve of GPS is shown in Fig.13 and Fig.14.Tuned membership error and error change membership function are shown in Fig.15 and Fig.16.

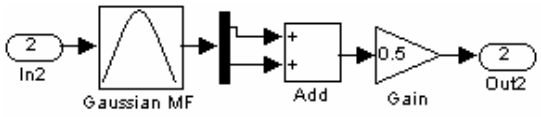


Fig. 10 Simulink model of each node

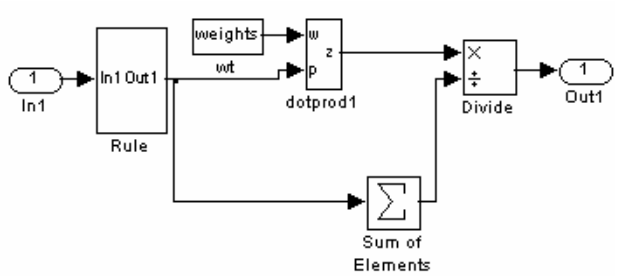


Fig. 11 Simulink model of final control output $u_{nf}$

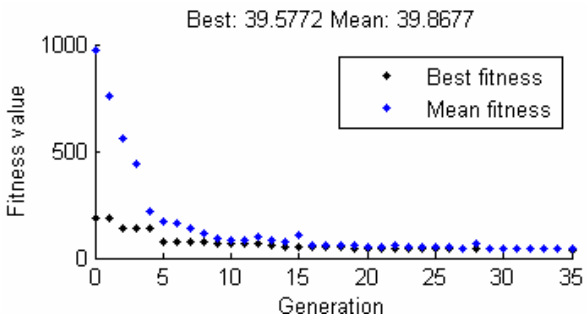


Fig. 12 Best Fitness curve of GA

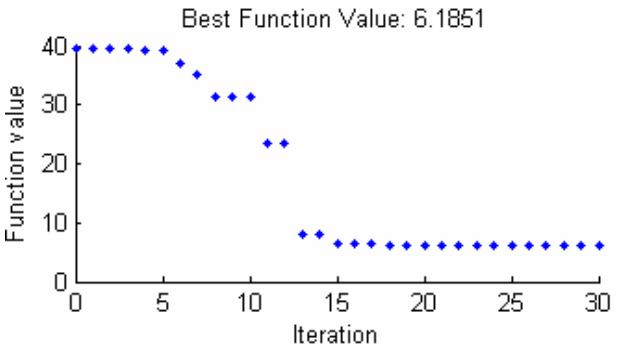


Fig. 13 Best Function value curve of GPS for mean and variance tuning

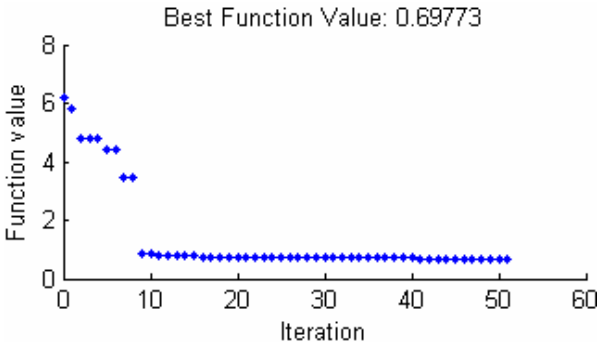


Fig. 14 Best Function value curve of GPS for mean tuning

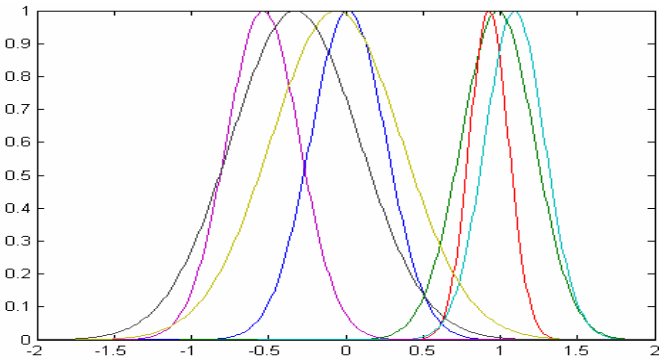

Fig. 15 Membership function of error after tuning

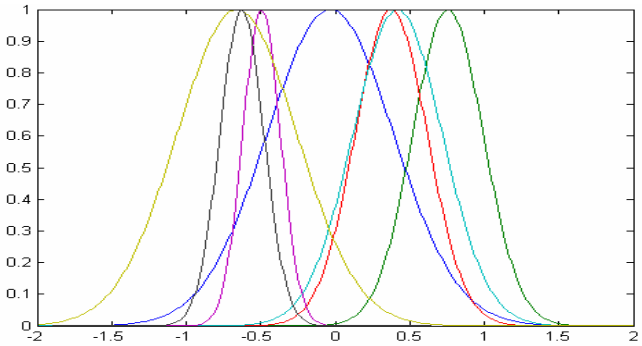

Fig.16 Membership function of error change after tuning

## V. RESULTS

Results of both types of Neuro-Fuzzy controllers are tested in terms of ITSE in Cartesian space and ISE in joint space. The desired Cartesian space trajectories taken for testing controller is butterfly. Parametric equation of Butterfly trajectory is

$$x_d = 200 * \cos(t)(e^{\cos t} - 2\cos(4t) - \sin^5(t/12)) + 350$$

$$y_d = 200 * \sin(t)(e^{\cos t} - 2\cos(4t) - \sin^5(t/12)) + 200$$

For checking the robustness of controller a disturbance torque D is applied

$$D = 1.5\sin(4.3575t) + \sin(9.825) + \sin(2.7075) + 1$$

The sampling time of system is 1ms. Fig.17 and Fig.18 show desired and actual output butterfly trajectory with RBFNF based controller with and without disturbance. Cartesian space error $dx$, $dy$ $dz$ for tracking butterfly trajectory with RBFNF based controller with and without disturbance are shown in Fig.19.a and Fig.19.b and corresponding joint space ITSE and Cartesian space Integral square errors; $ISE_x$, $ISE_y$, $ISE_z$ are shown in Table III .and Table IV. Fig.20 to Fig.22.b shows trajectory and error for ANFIS based controller. From Table III and Table IV of joint space ITSE and Cartesian space ISE, it is clear that performance of ANFIS in joint space is better than RBFNF.

TABLE III
JOINT SPACE ITSE USING NEURO FUZZY CONTROLLER

| ITSE | Butterfly | |
|---|---|---|
| | Without D | With D |
| ANFIS | 19.6763 | 76.0363 |
| RBF-NF | 42.7020 | 109.0133 |

TABLE IV
CARTESIAN SPACE ISE USING NEURO FUZZY CONTROLLER

| $ISE_x$ (mm) | Butterfly | |
|---|---|---|
| $ISE_y$ (mm) | without $D$ | with $D$ |
| ANFIS | 1205.3 | 1000.6 |
| | 1207.5 | 5154.1 |
| RBFNF | 519.6466 | 865.68 |
| | 3840.4 | 7421.5 |

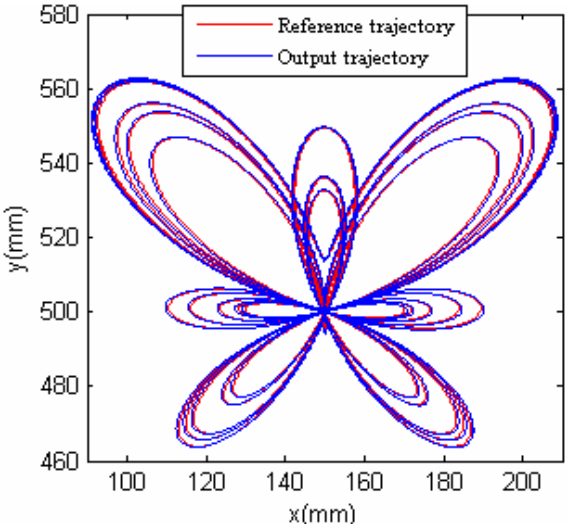


Fig. 17 Butterfly trajectory with RBFNF without disturbance

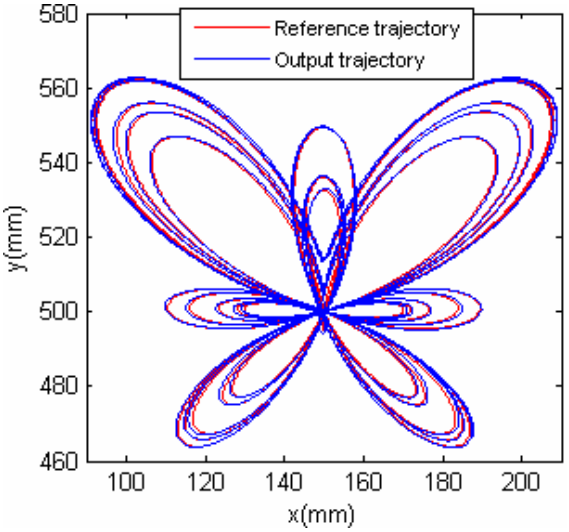


Fig. 18 Butterfly trajectory with RBFNF with disturbance

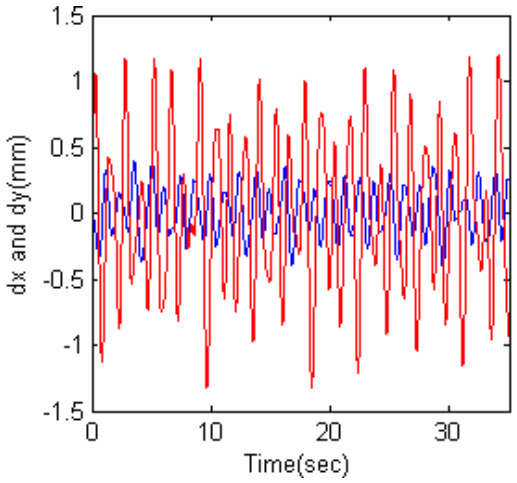


Fig. 19.a Cartesian space error for butterfly trajectory tracking using RBFNF without disturbance

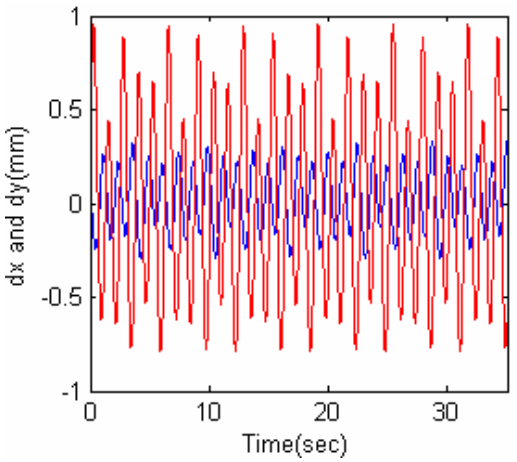


Fig. 19.b Cartesian pace error for butterfly trajectory tracking using RBFNF with disturbance

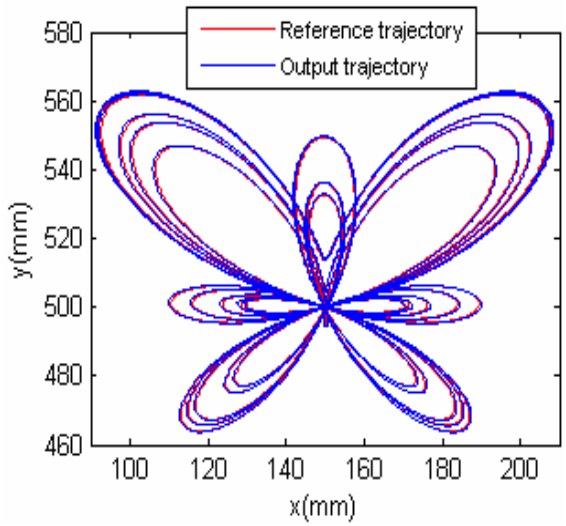


Fig. 20 Butterfly trajectory with ANFIS without disturbance

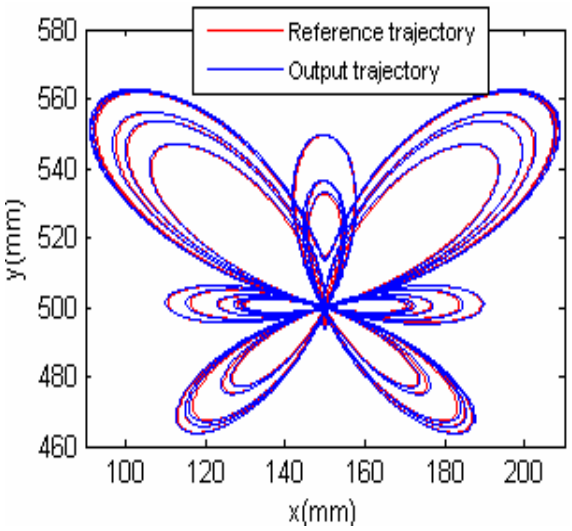


Fig. 21 Butterfly trajectory with ANFIS with disturbance

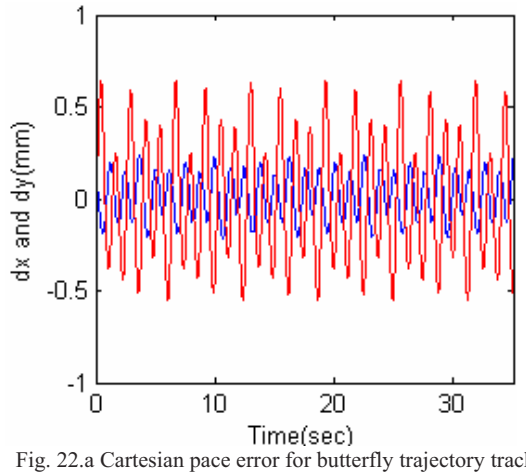


Fig. 22.a Cartesian pace error for butterfly trajectory tracking using ANFIS without disturbance

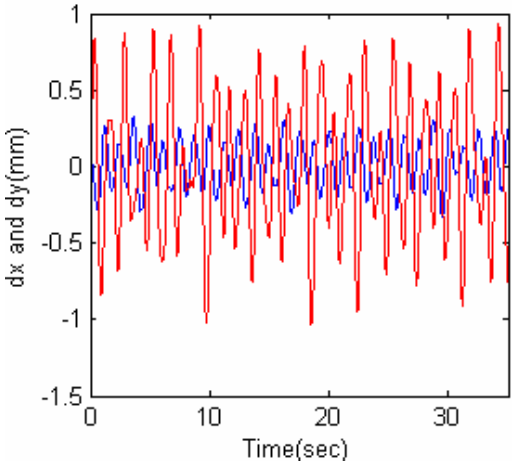


Fig. 22.b Cartesian pace error for butterfly trajectory tracking using ANFIS with disturbance

## VI. CONCLUSIONS

ANFIS and Hybrid Genetic Algorithm, Generalized Pattern Search tuned RBFNF are implemented for Puma 560 manipulator control. Both of controllers are being compared for butterfly trajectory tracking in Cartesian space .Performances in terms of joint space ITSE and Cartesian space ISE is being compared. The proposed RBFNF methodology in [5] is modified here by using hybrid GA and Generalized Pattern Search technique and successfully implemented for robotic manipulator control applications. GA tuned RBFNF is not very effective because of increase in dimensionality of search space with increase in number of antecedent in Fuzzy rule base. ANFIS is found to be slightly better than RBF-NF. But the designing of RBF-NF without rule base gives it an edge over ANFIS.

## REFERENCES

[1] Lin C T and Lee C S G, Neural Network based Fuzzy Logic Control and Decision System, IEEE Transactions on Computers,vol.40,no.12,pp. 1320-1336, 1991.
[2] Bherenji H R and Khedkar P, "Learning and Tuning Fuzzy Logic Controllers through Reinforcements," *IEEE Transactions on Neural Networks*, vol.3, pp. 724-740, 1992.
[3] Jyh-Shing Roger Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, May/ Jun 1993.
[4] A. Nurnnberger, D. Nauck, R. Kruse ,"Neuro-fuzzy control based on the NEFCON-model: recent developments," *Soft Computing - A Fusion of Foundations, Methodologies and Applications* vol.2, no. 4 , pp.168-182,Feb. 1999
[5] Teo Lian Seng, Marzuki Khalid, and Rubiyah Yusof , "Tuning of A Neuro-Fuzzy Controller by Genetic Algorithm," *IEEE Transactions on Systems, Man and Cybernetics*, Apr. 1999.
[6] C. H. Wang, T. C. Lin, T. T. Lee, and H. L. Liu, "Adaptive hybrid intelligent control for uncertain nonlinear dynamical systems," IEEE Transaction on Systems, Man and Cybernetics B, vol. 32, pp. 583–597, Oct. 2002.
[7] Y. Gao and M. Joo, "Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems," IEEE Transaction on Fuzzy System,vol. 11, pp. 462–477, Aug. 2003.
[8] Meng Joo and Yang Gao, "Robust Adaptive Control of Robot Manipulators Using Generalized Fuzzy Neural Networks,"*IEEE Transactions on Industrial Electronics*, vol. 50, no. 3, June 2003
[9] Y. Gao, M. J. Er, and S. Yang, "Adaptive fuzzy neural control of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 48, pp. 1274–1278, Dec.2001.
[10] Y. Gao and M. J. Er, "Robust adaptive fuzzy neural control of robot manipulators," in Proc. IEEE International Conference on Neural Networks,Washington, DC, 2001, pp. 2188–2193.
[11] Brian Armstrong, Oussama Khatib, Joel Burdick, "The explicit dynamic Model and Inertial Parameters of the Puma 560 Arms," *IEEE International conference on Robotics and Automation*, vol. 2 ,pp. 1608-1613, May 1994.
[12] T.J.Tarn, A.K.Bejczy, G.T.Marth, A.K.Ramadorai "Performance Comparison of Four Manipulator Servo Schemes" *IEEE control system magazine,* vol. 3 issue 1, Feb 1993.
[13] Chul-Goo Kang Online "Trajectory Planning for a PUMA Robot," *International Journal of Precision Engineering and Manufacturing, vol*. 8, vo.4, pp.16-21, Oct. 2007.
[14] Francisco Valero and Vicente Mata, Antonio Besa "Trajectory planning in workspaces with obstacles taking into account the dynamic robot behavior," *Journal of Mechanism and Machine Theory,* vol. 41, issue 5, pp. 525-536,May. 2006.
[15] Chia-Yu E. Wang, Wojciech K. Timoszyk, and James E. Bobrow "Payload Maximization for Open Chained Manipulators: Finding Weightlifting Motions for a Puma 762 Robot," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, Apr. 2001.
[16] Shadia Elgazzar "Efficient Kinematic Transformations for the PUMA 560 Robot," *IEEE Journal Of Robotics And Automation,* vol. Ra-1, no. 3, Sept. 1985.
[17] Said M. Megahed "Inverse Kinematics of Spherical Wrist Robot Arms Analysis and Simulation," *Journal of Intelligent and Robotic Systems*, vol. 5, pp. 211-227, 1992.
[18] Jean Cote, C1ement M. Gosselin and Denis Laurendeau "Generalized Inverse Kinematic Functions for the Puma Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 1 I, no. 3, Jun. 1995.
[19] Fan-Tien Cheng, Tzung-Liang Hour, York-Yin Sun, and Tsing-Hua Chen "Study and Resolution of Singularities for a 6-Dof Puma Manipulator "*IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics,* vol. 27, no. 2, Apr. 1997.
[20] Frederic Chapelle and Philippe Bidaud, "Closed form solutions for inverse kinematics approximation of general 6R manipulators," *Journal of Mechanism and Machine Theory,* vol. 39, issue. 3, pp. 323-338, Mar 2004.
[21] R.Kelly, V.santibanez A.Loria, *Control of Robot Manipulators in joint space*, Springer Advanced Textbooks in Control and Signal Processing, series 2005
[22] David A.Coley, *Introduction to Genetic Algorithm for scientist and engineer*, World scientific Publishing 1999

**Sufian Ashraf Mazhari** was born in India and received his Bachelor degree (Electrical Engineering) from Aligarh Muslim University, Aligarh India, in 2005 and M.Tech degree in Electrical Engineering from Indian Institute of Technology, Roorkee India.Currently is working with GS E & C ,Gurgaon ,India. His area of research includes Robotic Control, Optimization and Computer vision system.

**Surendra Kumar** (M'07 ) received B.E (Electrical), M.E(System Engineering and Operations Research) and Ph.D. Electrical in 1969,1971 and 1982 respectively in India. He joined the Department of Electrical Engineering, Indian Institute of Technology Roorkee, India as lecturer in 1972 .He has 36 year of teaching and research experience. Presently he is Assistant Professor .He has been on teaching assignment to University of Technology, Baghdad, IRAQ during 1987-1989.He is member IEEE, Fellow of Institution of Engineers India and member of ISTE India. His area of research interest is mainly Control, Optimization, AI application to Robotic Control and Fuzzy Reliability.

## APPENDIX.

*Abbreviation used:*

$c_i = \cos(\theta_i), s_i = \sin(\theta_i), c_{ij} = \cos(\theta_i + \theta_j)$

$s_{ij} = \sin(\theta_i + \theta_j), s_{i,-j} = \sin(\theta_i - \theta_j)$

The arm configuration parameters of Puma 560 $k_1, k_2$ and $k_3$ are defined as

$$k_1 = \begin{cases} +1, \text{ lefty} \\ -1, \text{ righty} \end{cases}$$

$$k_2 = \begin{cases} +1, \text{ elbow up} \\ -1, \text{ elbow down} \end{cases}$$

$$k_3 = \begin{cases} +1, \text{ no flip} \\ \text{-1 , flip} \end{cases}$$

The parameters are $k_1, k_2$ and $k_3$ are used to find inverse kinematics solution. However in the case of a known set of joint angles, as in the case of the direct kinematics, these parameters can be computed.

*Forward kinematics:* The problem is defined as given the joint angles vector, find the Cartesian position/orientation vector *R,* and the arm configuration parameters $k_1, k_2, k_3$

The orientation angles $r_\theta, r_\psi$ and $r_\rho$ are defined as

$$\cos(r_\vartheta) = \mathrm{c}_{23}\,\mathrm{c}_5 - \mathrm{s}_{23}\,\mathrm{s}_5\,\mathrm{c}_5$$

$$r_\psi = \theta_6 + a\tan 2[\mathrm{s}_{23}\,\mathrm{s}_4, \mathrm{s}_5\,\mathrm{c}_{23} + \mathrm{s}_{23}\,\mathrm{c}_5\,\mathrm{c}_4]$$

$$r_\rho = \theta_1 + a\tan 2[\mathrm{s}_5\,\mathrm{s}_4, \mathrm{c}_5\,\mathrm{s}_{23} + \mathrm{c}_{23}\,\mathrm{s}_5\,\mathrm{c}_4]$$

Where atan2 (x, y) is four-quadrant version $\tan^{-1}(x/y)$. As $|\cos(r_\theta)| \to 1$ the accuracy of equations deteriorates because $|\cos(r_\theta)| = 1$ is a singular point. If $\sin(r_\theta) \approx 0$, $r_\theta$ is set to zero or $\pi$ depending upon sign of $\cos(r_\theta)$. Value of $r_\psi$ is set to zero and $r_\rho$ is calculated using

$$r_\rho = \theta_1 + a\tan 2[2\,\mathrm{s}_{46}/[c_{23} + c_{r_\theta}], \mathrm{c}_{46}]$$

Position vector *R, is defined as*

$$r_x = -w_b s_1 - dc_1 - l_4 s_{r_\theta} s_{r_\rho}$$

$$r_y = w_b c_1 - ds_1 + l_4 s_{r_\theta} c_{r_\rho}$$

$$r_z = w_a + l_1 + l_4 c_{r_\theta}$$

Where

$$w_a = l_2 c_2 + l_3 c_{23'}$$

$$w_b = l_2 s_2 + l_3 s_{23'}$$

The arm configuration is determined by evaluating parameters $k_1, k_2, k_3$ .If $w_b \geq 0$ then the arm is lefty and $k_1 = +1$, but if $w_b < 0$, then the arm is righty and $k_1 = -1$. If $k_1\theta_3 \geq 0$ then the arm is elbow up and $k_2 = +1$ else $k_2 = -1$ .if $\theta_5 \geq 0$ then a no-flip solution exists and $k_3 = +1$ but if $\theta_5 < 0$ then a flip solution exist and $k_3 = -1$

*Inverse kinematics:*

The problem is defined as given Cartesian position/orientation vector *R,* and the arm configuration parameters $k_1, k_2, k_3$ find the joint angles vector.

Joint angles $\theta_1, \theta_2 ....., \theta_6$ are given as

$$\theta_1 = a\tan 2(-k_1 w_{1x}, k_1 w_{1y}) - k_1 a\tan 2(d, l)$$

Where

$$w_1 = (r_x + l_4 s_{r_\theta} s_{r_\rho})i + (r_y - l_4 s_{r_\theta} c_{r_\rho})j + (r_z - l_4 c_{r_\theta})k$$ Singular point exists if $w_{1x} = w_{1y}$. However considering the arm geometry this condition is never satisfied

$$\cos(\theta_3^{'}) = \frac{n^2 - l_2^{\,2} - l_3^{\,2}}{2 l_2 l_3}$$

$$\theta_3 = k_1 k_2 \theta_3^{'} + \delta$$

$$\mathrm{c}_5 = c_{23} c_{r_\theta} + s_{23} s_{r_\theta}\cos(r_\rho - \theta_1)$$

$$\text{if } s_5 > \xi \;\; \theta_4 = a\tan 2[s_{23} s_{r_\rho,-1}, c_{23} s_{r_\theta} c_{r_\rho,-1} - c_{r_\theta} s_{23}]$$

$$\theta_6 = r_\psi - \beta$$

Where

$$\beta = a\tan 2[s_{23} s_{r\rho,-1}, s_{r_\theta} c_{23} - c_{r_\theta} s_{23} c_{r\rho,-1}]$$

Manipulator loses a degree of freedom when two joint axes become collinear. This is case when $\sin(\theta_5) = 0$ and, consequently $\theta_4$ and $\theta_6$ become linearly dependent. The accuracy of $\theta_4$ and $\beta$ deteriorates as $\sin(\theta_5) \to 0$ and they break down completely if $\sin(\theta_5) = 0$. Therefore for some value of $\xi$ and for $\sin(\theta_5) \leq \xi$ better value of $\theta_4$ and $\theta_6$ can be obtained using the equation

$$\theta_4 - \beta = a\tan 2\left\{0.5 s_{r_\rho,-1}[c_{23} + c_{r_\theta}], c_{r_\rho,-1}\right\}$$

In the case of a no-flip condition, that is $k_3 = 1$ the wrist angles are obtained from the above equations. If, however, $k_3 = -1$, the flip solution becomes $(\theta_4 + \pi, -\theta_5, \theta_6 + \pi)$

$l_1 = .672m, l_2 = .432m, l_3 = .433m$

$l_4 = .056m, d = .149m$(offset in arm due to shoulder offset and elbow offset), $\delta = 2.72^\circ$

Actuator data of Puma 560 robot

| Motor | R | L | $K_e$ | $K_m$ | N |
|---|---|---|---|---|---|
| 1 | 1.6 | 0.0048 | 0.19 | 0.2611 | 62.55 |
| 2 | 1.6 | 0.0048 | 0.19 | 0.2611 | 107.81 |
| 3 | 1.6 | 0.0048 | 0.19 | 0.2611 | 53.15 |
| 4 | 3.9 | 0.0039 | 0.12 | 0.0988 | 76.04 |
| 5 | 3.9 | 0.0039 | 0.12 | 0.0988 | 71.92 |
| 6 | 3.9 | 0.0039 | 0.12 | 0.0988 | 76.65 |