A STUDY ON PARAMETER TUNING OF WEIGHTED FUZZY RULEBASE USING GENETIC ALGORITHM FOR TRAJECTORY CONTROL OF PUMA560*

MONA SUBRAMANIAM A[a †], MANJU A[b] AND NIGAM M J[c]

Department of electronics and computer engineering
Indian Institute of Technology Roorkee, Roorkee,.India-247667
Email: [a] monasubramaniama@gmail.com, [b] manju.senthil@gmail.com , [c] mkndnfec@iitr.ernet.in

Abstract:
In this paper, a study on tuning of parameters in weighted-rule fuzzy system using genetic algorithm for trajectory control of PUMA560 is performed. There are different ways in which a fuzzy system can be tuned, like tuning the scaling factors, rules, rule weights, parameters of membership functions, and changing the type of the membership function itself. The various parameters tuned in this paper are rules, rule-weights and the membership functions. The tuning of these parameters is done using Genetic Algorithm. Though research work relating to individual parameters tuning are available, the comparative analysis of these three tuning methods are yet to be done. Hence the Performances of these three parameter tuning is evaluated and the best method of tuning the rule base is decided.

Keywords: Rulebase tuning, Fuzzy controller, Genetic algorithm, PUMA560 trajectory control, Robotics.

1. Introduction

PUMA 560 is a six degree of freedom (DOF) industrial robot. It has a wide variety of applications including material handling, welding, assembling, painting, grinding and other industrial applications [Sufian and Surendra (2008a)]. Fuzzy PD+I controllers are the most general use fuzzy controller as it has the following advantages of being simple , having less overshoot, removes steady state error, and smoothens control signal [Jan Jantzen (2007)]. The most popular technique in evolutionary computation research has been the genetic algorithm which can be applied to any problem that can be formulated as function optimization problem [Sivanandam and Deepa (2008).]. By tuning the gains of the fuzzy PID controller using genetic algorithm better results are obtained [Sufian and Surendra (2008b)]. Fuzzy controllers can be tuned by various methods, like changing the scaling factor, modifying the support and spread of membership functions, modifying the rules of the Rulebase and changing the type of a membership function itself, doing so will result in change of the control surface and hence the output of the fuzzy controller [Drainkov et all (1993)]. The usefulness of Rule tuning is demonstrated by *F. Herrera et all* [Herrera et all (1995)]. Membership function tuning using genetic algorithm is studied by *Rafael Alcala et all* [Rafael et all (2005).], where it was seen how the performance would be improved by tuning the lateral position and support of the membership function. In addition to these the rule-weights can also be changed to perform a local tuning of linguistic rules, which enables the linguistic fuzzy models to cope with inefficient and/or redundant rules thereby enhancing the robustness, flexibility and system modeling capability [Rafael et all (2003a)]. By assigning a rule weight to each of the fuzzy rules, complexity is increased while its accuracy is improved which suggests a trade-off relation between the accuracy and complexity [Hisao et all (2009).]. If a rule weight is applied to the consequent part of the rule, it modifies the size of the rule's output value [Nauck (2000)].

The main motivation for writing this paper is to do a study of various parameters tuning for weighted fuzzy controller by genetic algorithm. Parameters like rules, membership functions and rule-weights play an important role in any fuzzy controller, and optimizing them is a necessary task, since these parameters are always built by designers with trial and error along with their experience or experiments. This study is done on fuzzy PD+I controller of PUMA560 robot. After performing the tuning of individual parameters, an inference is drawn as to which procedure is better than the other with reference to ISE criterion.

This paper is organized as follows. Section 2 introduces PUMA560 robot arm and its dynamics. Section 3 provides discussion on fuzzy PD+I controller scheme. Section 4 describes genetic algorithm. Section 5 briefly talks about genetic fuzzy systems. Section 6 provides an overview of weighted knowledge base tuning, Section 7 gives results and discussions, and in section 8 conclusions and future scope are presented.

2. Dynamics of Puma560

Dynamics of a serial n-link rigid robot can be written as:

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau \quad (1)$$

where q is the n×1 vector of joint displacements, $\dot{q}$ is the n×1 vector of joint velocities, τ is the n×1 vector of actuators applied torques, $M(q)$ is the n×n symmetric positive definite manipulator inertia matrix, $c(q,\dot{q})$ is the n×1 vector of centripetal and Coriolis torques and $g(q)$ is the n×1 vector of gravitational torques due to gravity. We assume that the robot joints are joined together with revolute joints.

Let the desired joint position q_d be a twice differentiable vector function. We define a control problem to determine the actuator torques in such a way that the following control aim be achieved:

$$\lim_{t\to\infty} q(t) = q_d(t) \quad (2)$$

A six DOF PUMA-560 robot is considered for the simulation, the Kinematical and dynamical parameters of the arm and the torque limitations are adopted from the work of *Srinivasan and Nigam, and references therein* [Srinivasan and Nigam(2008)].

3. Fuzzy PD+I Controller

In this structure, the fuzzy system is applied only to the proportional and derivative signal of the linear PID controller [Bong and Chung (2002)]. The integral signal uses conventional linear method. The major roll of the integral signal is to eliminate the steady state error. The transient response is affected mostly by the proportional signal and the derivative signal. For the enhancement of the transient response, the varying gains are implemented on the proportional and derivative parts using two-input fuzzy system. The nonlinearities that make the varying gains possible are added by the fuzzy control rules and the membership functions.

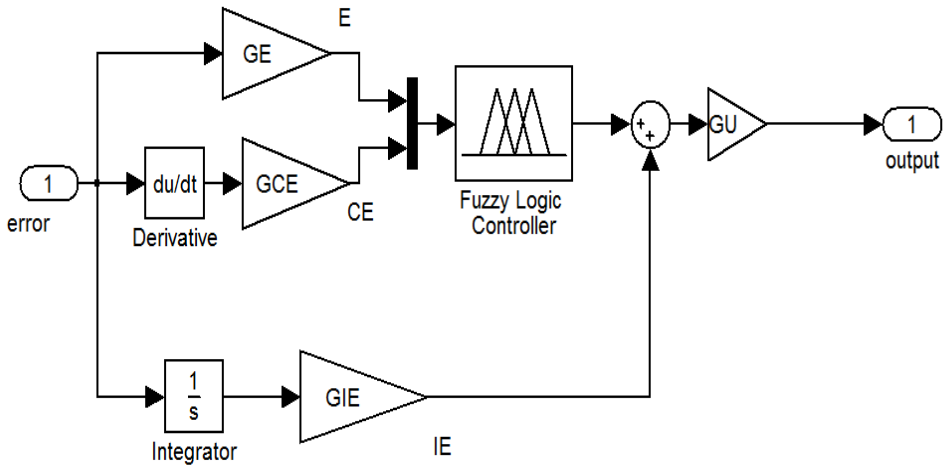
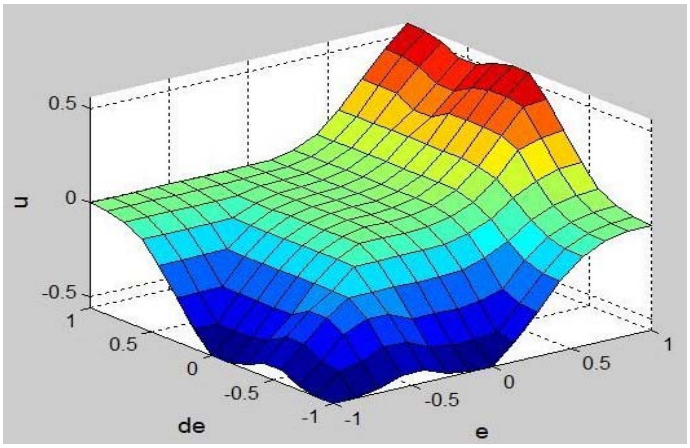
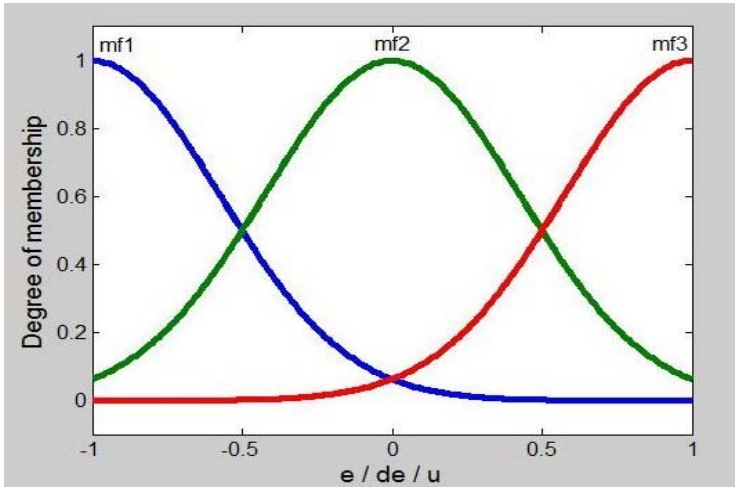


Figure 1. Structure of the Fuzzy PD+I controller

Figure 1. shows the structure of the Fuzzy PD+I controller., where GCE, GE, GIE and GU are the gains of Fuzzy PD+I controller and more often called scaling factors which can be varied to tune the controller. Genetic algorithm is used in this paper to coarsely tune these gains in order to produce base or reference system. Figure 2 (a). shows the surface view of the fuzzy PD controller and Figure 2 (b). shows the membership function and Figure 2 (c). shows the rulebase of the Fuzzy PD controller used.



(a)



(b)

de / e	mf1	mf2	mf3
mf1	mf1	mf1	mf2
mf2	mf1	mf2	mf2
mf3	mf2	mf3	mf3

(c)

Figure 2. (a) Surface view, (b) Membership function (Mamdani type) and (c) Rulebase of the Fuzzy PD controller

4. Genetic Algorithm

Genetic algorithms are general purpose search algorithms which use principles inspired by natural genetics to evolve solutions to problems. The basic idea is to maintain a population of *chromosomes* which represents candidate solutions to the concrete problem being solved, that evolves over time through a process of competition and controlled variation. Each chromosome in the population has an associated *fitness* to determine (*selection*) which chromosomes are used to form new ones in the competition process. The new ones are created using genetic operators such as crossover and mutation.

A GA starts off with a population of randomly generated chromosomes, and advances toward better chromosomes by applying genetic operators modeled on genetic processes occurring in nature. The population undergoes evolution in a form of natural selection. During successive iterations, called *generation*,

chromosomes in the population are rated for their adaptation as solutions, and on the basis of these evaluations, a new population of chromosome is formed using selection mechanism and specific genetic operators such as *crossover* and *mutation*. An evaluation or fitness function*(f)* must be devised for each problem to be solved. Given a particular chromosome, a possible solution, the fitness function returns a single numerical fitness, which is supposed to be proportional to the utility or adaptation of the solution represented by that chromosome.

Although there are many possible variants of the basic GA, the fundamental underlying mechanism consists of three operations:

1. evaluations of individual fitness,
2. formation of a gene pool (intermediate population) through selection mechanism, and
3. recombination through crossover and mutation operators

The below pseudo-code shows the structure of a Basic GA [Herrera and Magdalena (June 1997)], where *P(t)* denotes the population at generation t.

Procedure Genetic Algorithm
Begin(1)
t=0;
initialize P(t);
evaluate P(t);
While(Not *termination-condition*)do
Begin(2)
t=t+1;
select P(t) from P(t-1);
P(t)= recombine P(t);
evaluate P(t);
End (2)
End(1)

In this paper default genetic algorithm toolbox settings are used which are as follows

Population size	20
Creation function	uniform
Scaling function	Rank
Selection function	stochastic uniform
Elite count	2
Crossover fraction	0.8
Mutation function	Gaussian
Crossover	scattered
Migration direction	forward
Migration fraction	0.2
Migration interval	20

The main advantage of using GA for tuning of fuzzy PD+I controller is its ability to adapt to any constraints.

5. Genetic Fuzzy Systems

The genetic fuzzy systems are primarily used to automate the knowledge acquisition step in fuzzy system design, a task that is usually accomplished through an interview or observation of a human expert controlling the system [Hoffmann (2001)]. An evolutionary algorithm adapts either part or all of the components of the fuzzy knowledge base. Fuzzy knowledge base is not a monolithic structure but is composed of the data base and the rule base where each plays a specific role in the fuzzy reasoning process. Genetic tuning processes are targeted at optimizing the performance of an already existing fuzzy system. Designing a fuzzy rule based system is equivalent to finding the optimal configuration of fuzzy sets and/or rules, and in that sense can be regarded as an optimization problem. The optimization criterion is the problem to be solved at hand and the

search space is the set of parameters that code the membership functions, fuzzy rules and fuzzy rule-weights. The Figure 3 represents a genetic fuzzy system.

The performance is aggregated into a scalar fitness value on which basis the evolutionary (Genetic) algorithm selects better adapted chromosomes. A chromosome either codes parameters of membership functions, fuzzy rules and fuzzy rule-weights or a combination thereof. By means of crossover and mutation, the evolutionary algorithm generates new parameters for the database and/or rule base whose usefulness is tested in the fuzzy system.

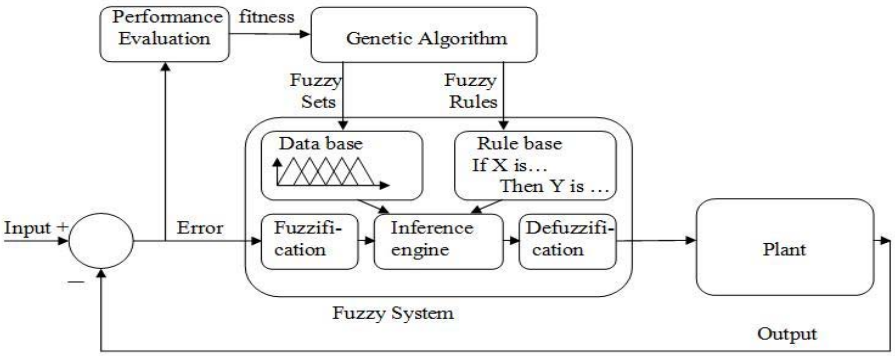


Figure 3. Genetic Fuzzy System

The objective functions considered here is based on the error criterion. In this paper performance of membership functions, rules and weight tuning are evaluated in terms of Integral square Error (ISE) error criteria. The error criterion is given as a measure of performance index. The ISEs of individual joints are added together to obtain an overall ISE. This is done to simplify the task of Genetic Algorithm. The objective of Genetic Algorithm is to minimize this overall ISE. The overall ISE is given by Equation 3.

$$ISE = \sum_{i=1}^{6} \int e_i^2(t)dt \qquad (3)$$

Where $e_i(t)$ is the error signal for the i^{th} joint. Here i can take values from 1 to 6 corresponding to 6 joints.

6. Weighted Knowledge Base Tuning

Rule weights suppose an effective extension of the conventional fuzzy reasoning process that allows tuning of the system to be developed at the rule level [Rafael et all (2003b)]. This approach improves the accuracy of the learned model since they induce a good cooperation among rules. However, they come with the drawback of a small interpret-ability loss which lies in the difficulty to interpret the actual action performed by each rule in the interpolative reasoning process. From other point of view (rule level), when weights are applied to complete rules, the corresponding weight is used to modulate the firing strength of a rule in the process of computing the defuzzified value. For human beings, it is very close to consider this weight as an importance degree associated to the rule, determining how this rule interacts with its neighboring ones. In addition, only weight values in [0,1] are considered, since this preserves the model readability. In this way, the use of rule weights represents an ideal framework for extended Linguistic Fuzzy Modeling while searching for a trade-off between accuracy and interpret-ability.

In order to do so, the weighted rule structure and the inference system extended for multiple output variables is followed which is taken from [Mucientes et all (2009)] and given by the statement below:

IF X_1 *is* A_1 *and . . . and* X_n *is* A_n

THEN Y_1 *is* B_1 *and . . . and* Y_m *is* B_m *with [w],*

Where, X_i *and* Y_j are the linguistic input and output variables respectively, A_i *and* B_j are the linguistic labels used in the input and output variables respectively, w is the real-valued rule weight, and *with* is the operator modeling the weighting of a rule.

With this structure, the fuzzy reasoning must be extended. The classical approach is to infer with the FITA (First Infer, Then Aggregate) scheme and compute the defuzzified output of the *j-th* variable as the following weighted sum:

$$y(j) = \frac{\sum m_h.w_h.P_h(j)}{\sum m_h.w_h} \qquad (4)$$

With m_h being the matching degree of the *h-th* rule, w_h being the weight associated to the *h-th* rule, and $P_h(j)$ being the characteristic value of the output fuzzy set corresponding to that rule in the *j-th* variable. In this contribution, center of gravity will be considered as characteristic value and the minimum t-norm will play the role of the implication and conjunctive operators.

If a rule weight is applied to the consequent part of a rule, it modifies the size of a rule's output value [Nauck (2000)]. By assigning a rule weight to each fuzzy rule, the complexity is increased while its accuracy is improved. This suggests a tradeoff relation between the accuracy and the complexity [Hisao et all(2009)].

Changing the center and spread of a membership function will also affect the performance of the controller [Drainkov et all (1993)]. Here Gaussian membership function is used, which is characterized by the Equation 5, where μ is the center and σ denotes the spread.

$$f(x,\sigma,c) = e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{5}$$

In the following subsections the implementation of the rule tuning, rule-weight tuning and membership function tuning are discussed.

6.1. *Rule tuning*

In this subsection the chromosomes are encoded with the values of consequent part of the if-then rule of the fuzzy rule base, in other words, contains the parameters of the consequent. Genetic algorithm is run until the terminating condition. Genetic algorithm varies these parameters stochastically and on convergence produces a rule base which will have optimal rules in it. Since we are using a 3x3 Rulebase we will require chromosomes to consist of 9 variables per fuzzy system corresponding to the 9 rules.

Structure of chromosome: $C_1, C_2, \ldots, C_n$.

6.2. *Rule-Weight tuning*

In this subsection the chromosomes are encoded with the values of weighting for consequent part of the if-then rule of the fuzzy rulebase. Genetic algorithm is run until the terminating condition. Genetic algorithm varies these parameters stochastically and on convergence produces a rulebase which will have rules with optimal rule-weights in it. Since we are using a 3x3 rulebase we will require chromosomes to consist of 9 variables per fuzzy system corresponding to the 9 rule weights.

Structure of chromosome: $W_1, W_2, \ldots, W_n$.

Where W_i is the rules-weight, *i* varies from 1 to n. In this case n=54 (i.e. 9rule-weights*6joints).

6.3. *Membership function tuning*

In this subsection the chromosomes are encoded with the values of parameters of the membership function. Gaussian membership function is used. A Gaussian membership function is characterized by a mean and a spread. The centers of the extreme end membership functions are kept as it is, since changing them will affect the universe of discourse and it is desired that the universe of discourse is kept a constant. Genetic algorithm is run until the terminating condition. Genetic algorithm varies these parameters stochastically and on convergence produces a fuzzy system which will have membership functions with mean and spread. Since we are using a 3x3 rule fuzzy system there will be 9 membership functions altogether, 6 for input and 3 for output. Considering this there will be 9 means and 9 spreads, of these 6 extreme centers are kept untouched hence the total number of parameters tuned is 12. We will require chromosomes to consist of 12 variables per fuzzy system

Structure of chromosome: $MF_1, MF_2, \ldots, MF_n$.

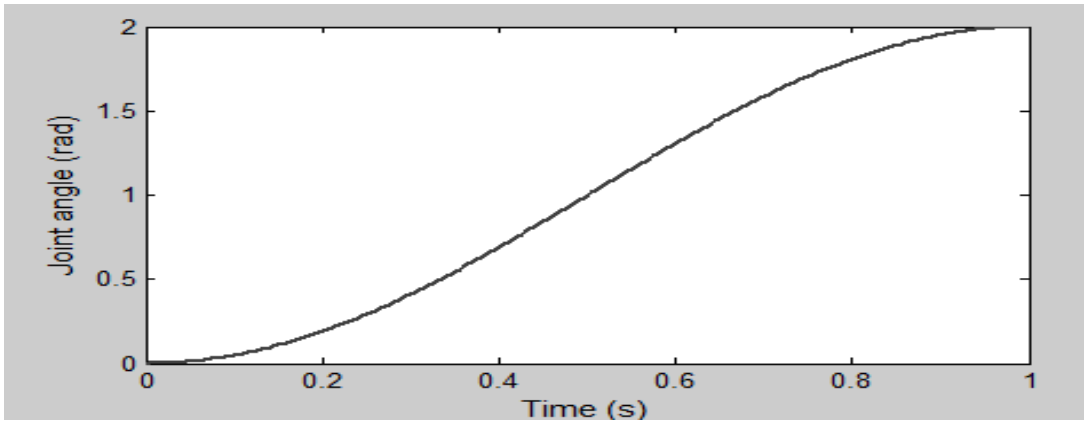
Where MF_i is the membership function parameter, *i* varies from 1 to n. In this case n=72 (i.e. 12Mf parameters*6joints).

7. Results and Discussions

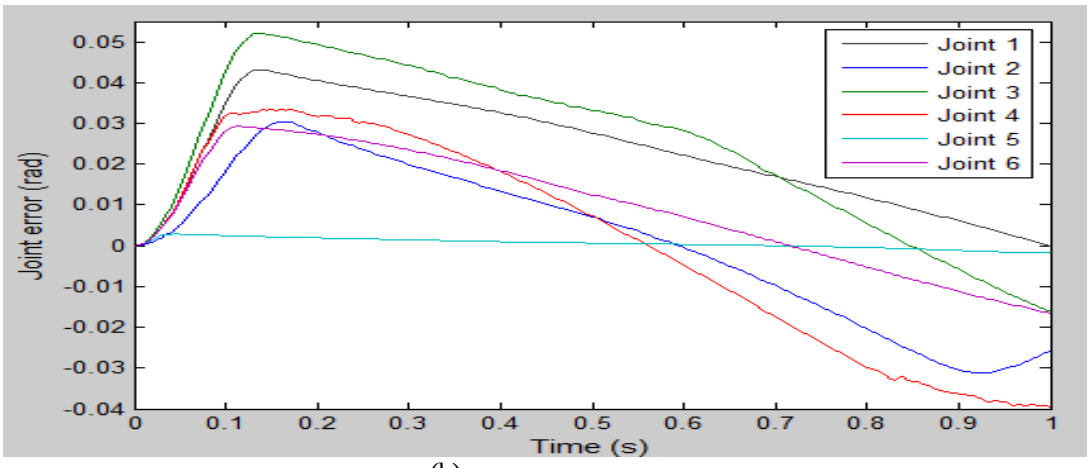
Simulations are carried out using MATLAB Version 7.0.1.24704 (R14 with Genetic algorithm & direct search toolbox and Fuzzy Logic Toolbox), 2 GHz computer with 2 GB RAM. Population size used is 20. While providing the input to the robot, care has to be taken that the trajectory used is continuous and smooth. The input trajectory used in this paper is 1-cos (pi*t) since it is a continuous and smooth function which can be double differentiated, t is time in seconds. This trajectory is used for all the joints.

Tuning of rules, weights and membership functions has been carried out genetically for a minimum of 200 generations and there after a stall limit of 50 is used until a maximum of 500 generations. Some of the details that are to be taken care of are, the weights need to be within the range [0 1], the rules generated must be valid, the universe of discourse should be kept same as the base system, the centers of the membership functions mf1 and mf3 are kept at -1 and 1 respectively and the center of the membership function mf2 is varied by the genetic algorithm to obtain an optimized location. The spread of all the membership functions are changed by the genetic algorithm. Both the spread and center are optimized in parallel while MF tuning.

Figure 10. shows Input trajectory signal given to the individual joints and Figure 11. shows the joint error generated by the given input trajectory for reference Fuzzy PD+I controller.



(a)



(b)

Figure 4. (a) Input trajectory signal given to the individual joints, (b) The joint error generated by the robot arm with reference Fuzzy PD+I controller

7.1. *Rule Tuning*

The result of rule tuning is presented here. The number of parameters tuned is 54 (9 rules × 6 joints). Figure 5 shows the Rule base after tuning of the rules and Figure 6 shows the control surface view of this rule base. Figure 7 represents the joint error generated by the robot arm with Rule tuned Fuzzy PD+I controller.

Joint 1 rule base

de / e	mf1	mf2	mf3
mf1	mf1	mf2	mf3
mf2	mf1	mf2	mf3
mf3	mf3	mf3	mf3

Joint 2 rule base

de / e	mf1	mf2	mf3
mf1	mf2	mf1	mf3
mf2	mf1	mf2	mf3
mf3	mf2	mf3	mf2

Joint 3 rule base

de / e	mf1	mf2	mf3
mf1	mf3	mf3	mf1
mf2	mf1	mf2	mf3
mf3	mf2	mf1	mf2

Joint 4 rule base

de / e	mf1	mf2	mf3
mf1	mf2	mf2	mf1
mf2	mf1	mf2	mf3
mf3	mf3	mf2	mf2

Joint 5 rule base

de / e	mf1	mf2	mf3
mf1	mf2	mf1	mf1
mf2	mf2	mf2	mf3
mf3	mf2	mf2	mf2

Joint 6 rule base

de / e	mf1	mf2	mf3
mf1	mf1	mf2	mf2
mf2	mf1	mf2	mf3
mf3	mf1	mf2	mf3

Figure 5. Rule base after Rule tuning using Genetic Algorithm

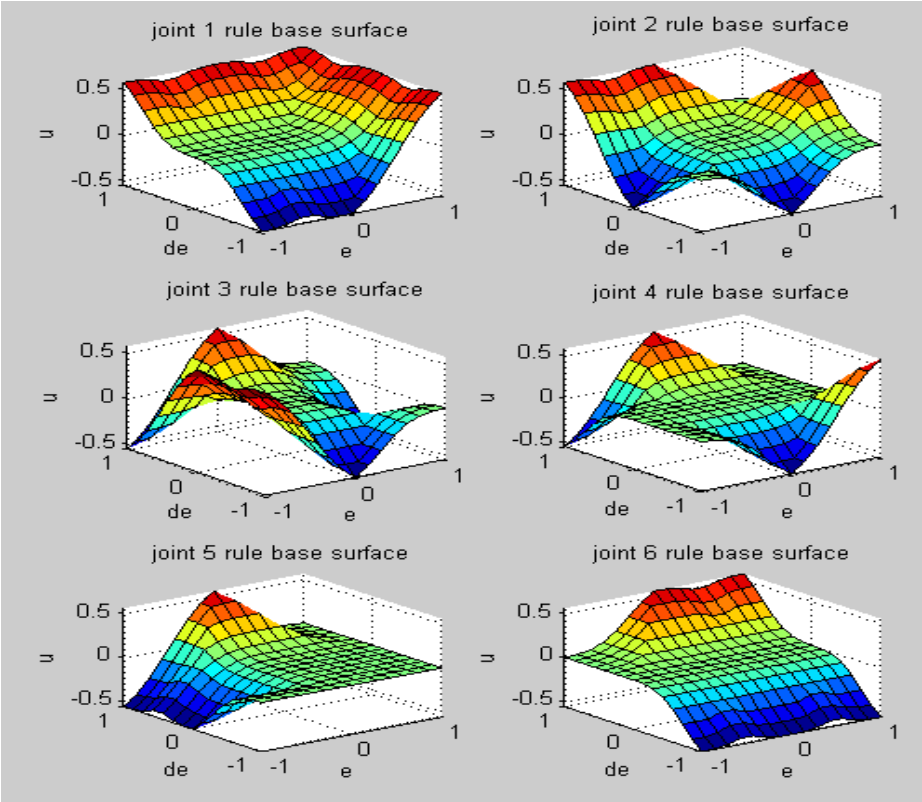


Figure 6. Control surfaces of the rulebase after rule tuning

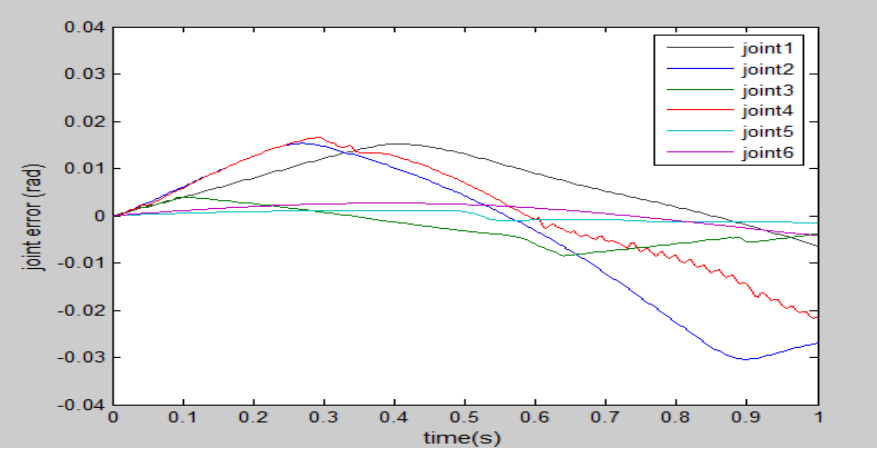


Figure 7. Joint error generated by the robot arm with Rule-tuned Fuzzy PD+I controller

7.2. *Rule-Weight Tuning*

The result of Rule weight tuning is presented here. The weights of the rules are tuned and the number of parameters tuned is 54 (9 weights × 6 joints). Figure 8 shows the weight tuned rulebase, with weights written within brackets. Figure 9 shows the control surface view of this rule base. Figure 10 represents the joint error generated by the robot arm with rule tuned Fuzzy PD+I controller.

Joint 1 weighted Rulebase

de / e	mf1	mf2	mf3
mf1	mf1 (0.0643)	mf1 (0.9544)	mf2 (0.1449)
mf2	mf1 (0.9162)	mf2 (0.0788)	mf2 (0.006)
mf3	mf2 (0.9295)	mf3 (0.9741)	mf3 (0.9884)

Joint 2 weighted Rulebase

de / e	mf1	mf2	mf3
mf1	mf1 (0.9902)	mf1 (0.2041)	mf2 (0.0231)
mf2	mf1 (0.2558)	mf2 (0.0666)	mf2 (0.7322)
mf3	mf2 (0.1519)	mf3 (0.4503)	mf3 (0.9818)

Joint 3 weighted Rulebase

de / e	mf1	mf2	mf3
mf1	mf1 (0.7259)	mf1 (0.9823)	mf2 (0.2038)
mf2	mf1 (0.7317)	mf2 (0.0133)	mf2 (0.3173)
mf3	mf2 (0.9548)	mf3 (0.0624)	mf3 (0.9994)

Joint 4 weighted Rulebase

de / e	mf1	mf2	mf3
mf1	mf1 (0.3509)	mf1 (0.3401)	mf2 (0.2646)
mf2	mf1 (0.7820)	mf2 (0.0468)	mf2 (0.00004)
mf3	mf2 (0.9622)	mf3 (0.6189)	mf3 (0.8178)

Joint 5 weighted Rulebase

de / e	mf1	mf2	mf3
mf1	mf1 (0.9643)	mf1 (0.1898)	mf2 (0.7628)
mf2	mf1 (0.6462)	mf2 (0.0799)	mf2 (0.5111)
mf3	mf2 (0.8092)	mf3 (0.2067)	mf3 (0.9495)

Joint 6 weighted Rulebase

de / e	mf1	mf2	mf3
mf1	mf1 (0.703)	mf1 (0.9276)	mf2 (0.039)
mf2	mf1 (0.9975)	mf2 (0.0454)	mf2 (0.0486)
mf3	mf2 (0.902)	mf3 (0.9922)	mf3 (0.9340)

Figure 8. Weight tuned rulebase, with weights written within brackets

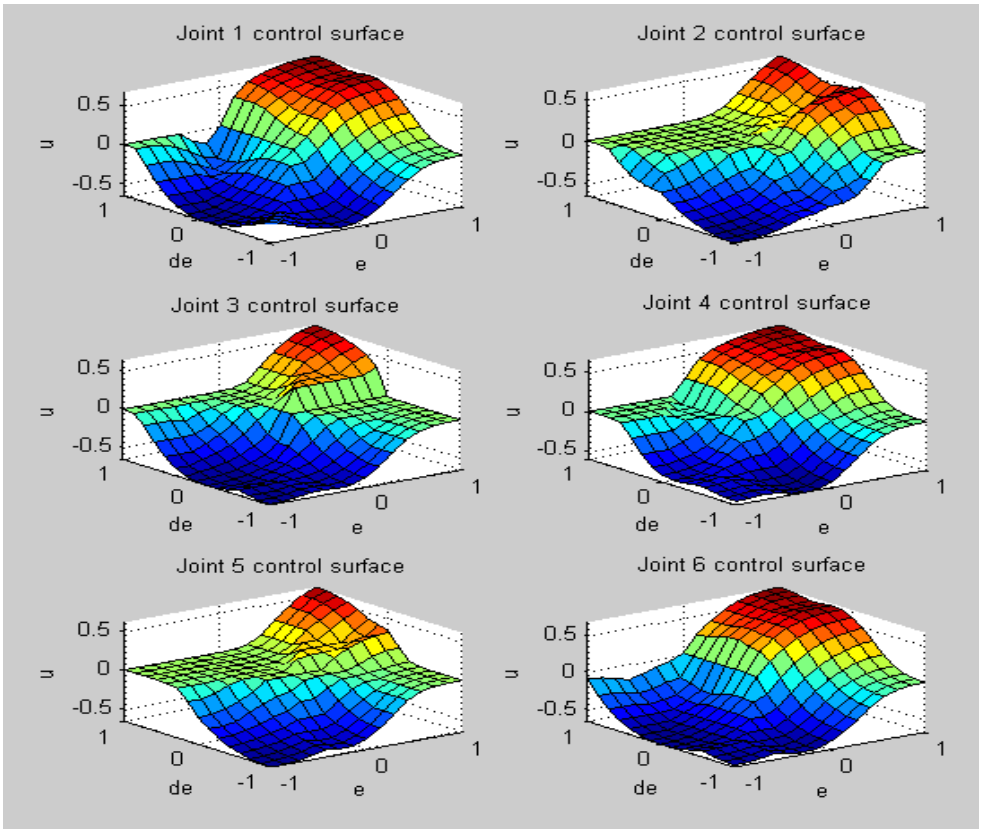


Figure 9. Control surface of weight tuned rulebase

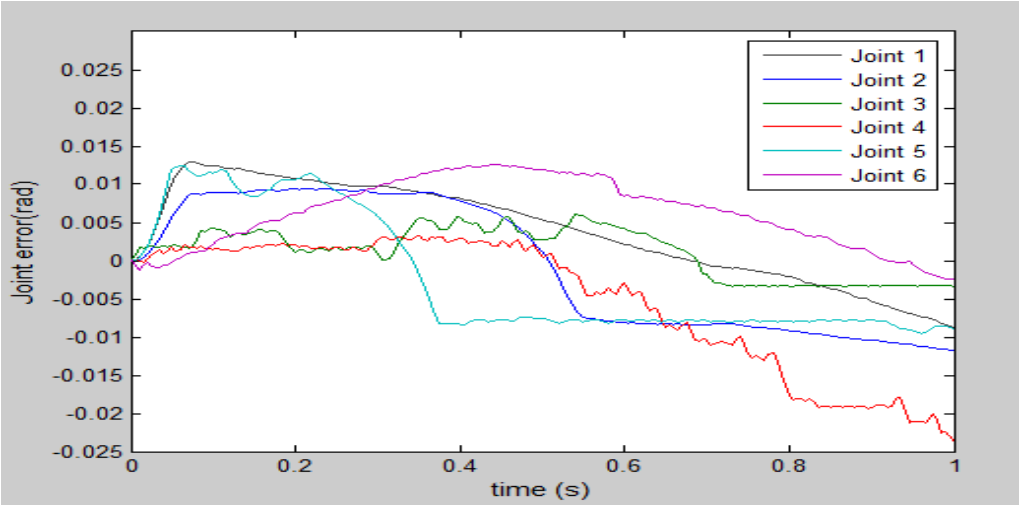


Figure 10. joint error generated by the robot arm with Rule-weight tuned Fuzzy PD+I controller

7.3. *Membership function tuning*

The result of membership function tuning is presented here. The number of parameters tuned are 72 ((9 for 'σ' + 3 for 'μ') × 6 joints). Figure 11 represent the joint error generated by the robot arm with membership functions tuned fuzzy PD+I controller. Figure 12 shows the membership functions (MF) of the fuzzy PD+I controller after MF tuning and Figure 13 shows the control surface after MF tuning.

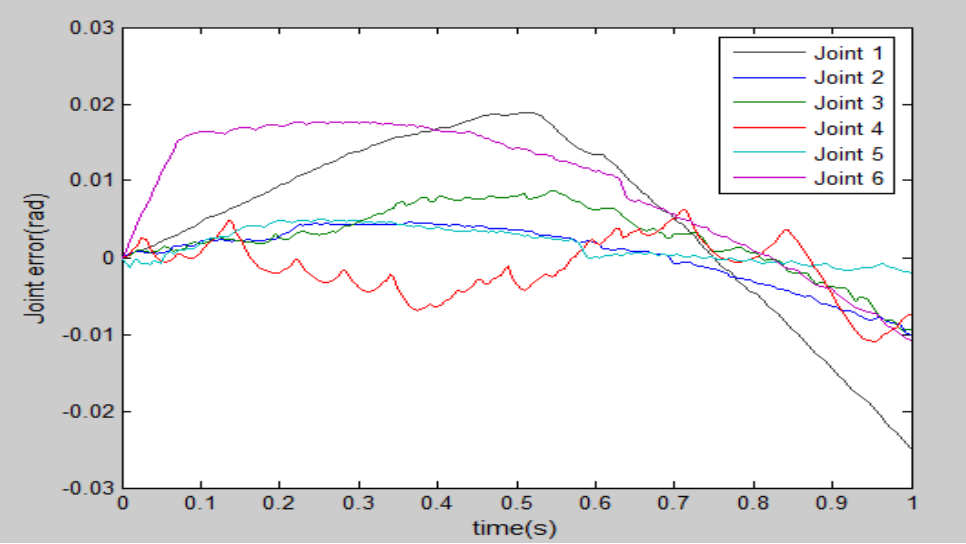


Figure 11. Joint error generated by the robot arm with Membership functions tuned Fuzzy PD+I controller

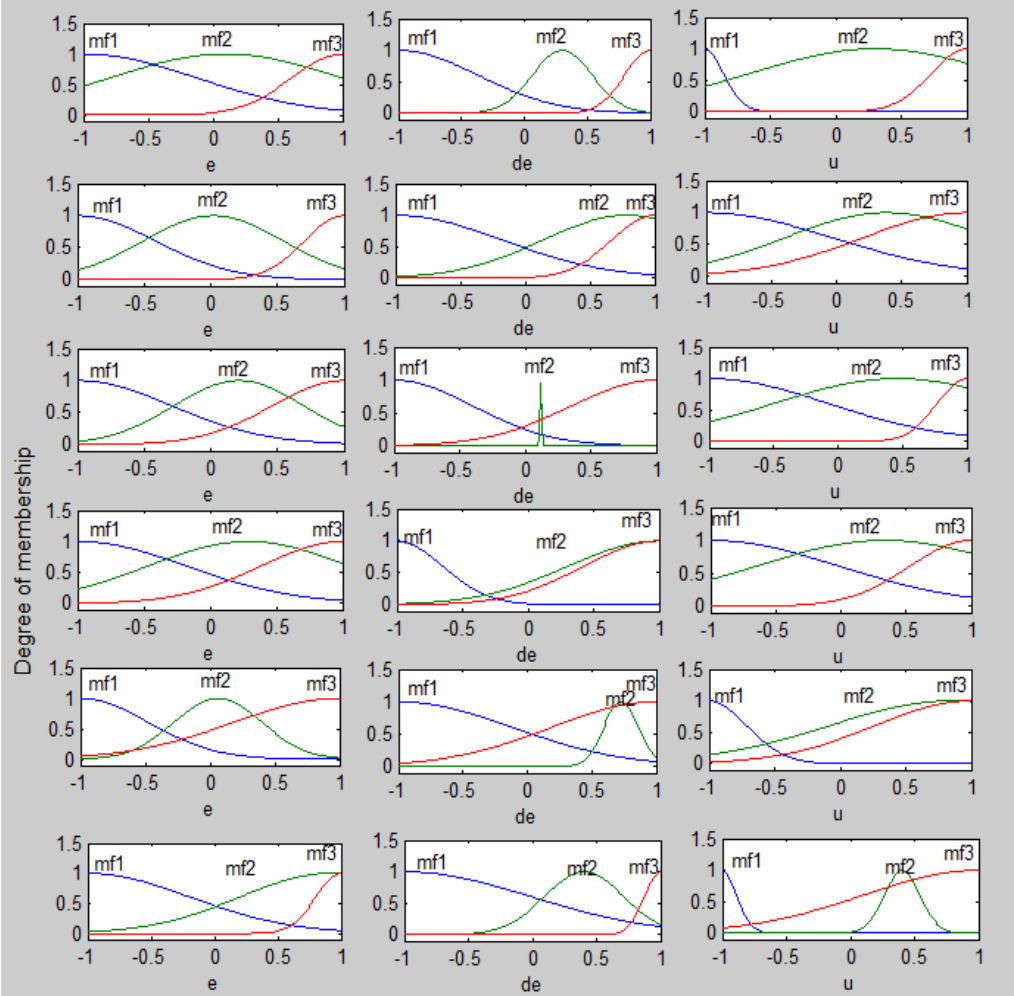


Figure 12. Membership functions of the Fuzzy PD+I controller after MF tuning

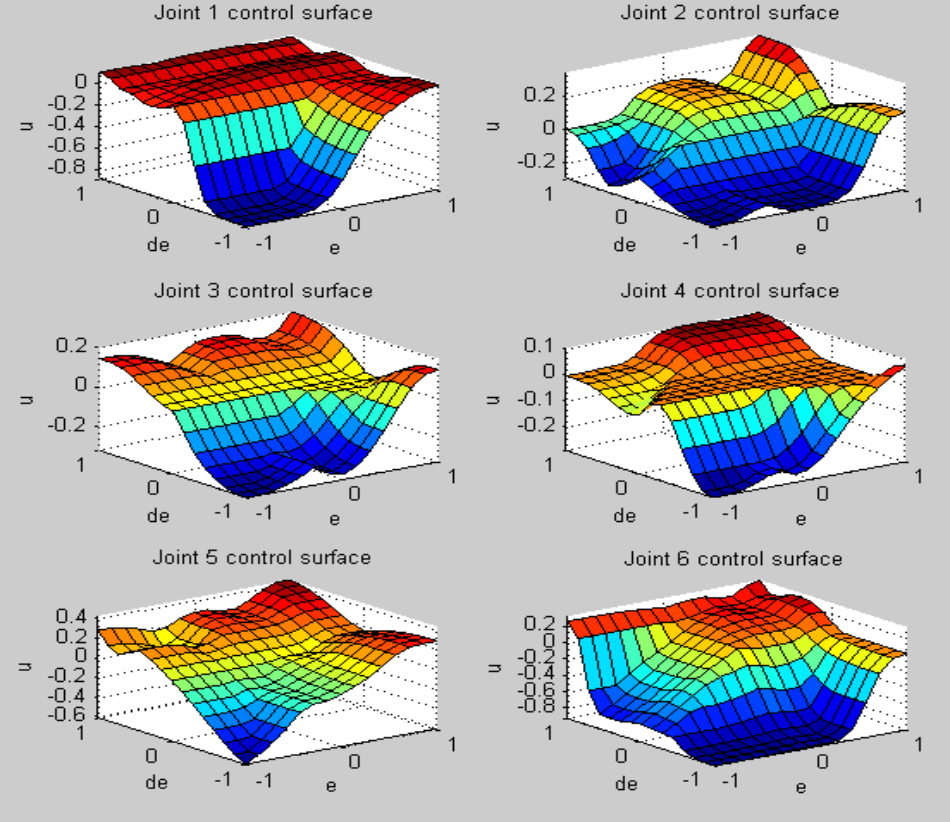


Figure 13. control surfaces of the Fuzzy PD+I controller after MF tuning

Table 1 gives the number of parameters tuned; break up of the ISEs of the individual joints and the overall ISE that is minimized by genetic algorithm. It provides results of the tuning methods in a tabular form. From this table, it is observed that the weights tuning provides better results (overall ISE) than others for this system. This is inferred from the last column of the table which provides the overall ISE.

Table 1 Tabulated results of the different tuning methods

Type	Base system	Rule tuning	Weight tuning	MF tuning
Parameters tuned	-	54	54	72
Joint 1 ISE $\int e_1^2(t)\,dt$	0.000691	7.51E-05	5.33E-05	0.000163
Joint 2 ISE $\int e_2^2(t)\,dt$	0.000361	0.00025	7.14E-05	1.53E-05
Joint 3 ISE $\int e_3^2(t)\,dt$	0.00099	1.88E-05	8.55E-06	2.46E-05
Joint 4 ISE $\int e_4^2(t)\,dt$	0.000624	0.000115	9.53E-05	1.62E-05
Joint 5 ISE $\int e_5^2(t)\,dt$	1.88E-06	1.01E-06	6.85E-05	3.11E-06
Joint 6 ISE $\int e_6^2(t)\,dt$	0.000276	4.25E-06	4.75E-05	0.000147
Overall ISE $\sum_{i=1}^{6}\int e_i^2(t)\,dt$	2.94E-03	4.64E-04	3.45E-04	3.69E-04

8. Conclusion and future scope

In this paper a study of different parameter tuning methods of the fuzzy controller for trajectory control of PUMA 560 is carried out. The different parameters tuned genetically are rules, rule-weights, and membership function of a Mamdani type fuzzy system. Table 1. provides a comparison of the methods mentioned in terms of the performance criterion. The criterion used for evaluating the performance is according to the Equation 3 which is presented in the last row of the Table 1. From the empirical data in the Table 1, it is clear that the tuning of rule-weights provides better overall ISE. In case of weight tuning the number of parameters tuned is 54 (9 weights $\times$ 6 joints) and in case of membership function tuning a close competitor of weight tuning in terms of performance, the number of parameters tuned are 72 ((9 for 'σ' + 3 for 'μ') $\times$ 6 joints). This again leverages the weight tuning in terms of lesser computational time since the number of parameters are less. As part of the future work a combination of these tuning methods can be studied.

References

[1] Bong Joo Kim, Chung Choo Chung (2002). *Design of Fuzzy PD + I Controller for Tracking Control*, Proceedings of the American Control Conference Anchorage, AK May 8-1, AACC p:2124-2129.

[2] Drainkov .D, Hellendoorn .H, Rienfrank M. (1993). *An Introduction to fuzzy control*, ISBN 81-7319-069-0, Springer-Verlag Berlin Heidelberg.

[3] Jan Jantzen.(2007). *Foundations of Fuzzy Control*, ISBN 978-0-470-02963-3, John Wiley & Sons Ltd.

[4] Herrera .F, Lozano .M, and Verdegay J. L. (1995). *Tuning Fuzzy Logic Controllers by Genetic Algorithm*, International Journal of Approximate Reasoning; vol.12 p: 299-315.

[5] Herrera .F, Magdalena .L (June 1997*). Genetic Fuzzy Systems: A Tutorial*. Tatra Mountains Mathematical Publications Vol. 13, p: 93-121, 1997. R.Mesiar,B.Riecan(Eds) Fuzzy structures, Current trends. Lecture Notes of the Tutorial: Genetic Fuzzy Systems. Seventh IFSA world congress (IFSA97), Prage,

[6] Hisao Ishibuchi, Yutaka Kaisho, and Yusuke Nojima,(2009). *Complexity, Interpretability and Explanation Capability of Fuzzy Rule-Based Classifiers*, FUZZ-IEEE 2009, Korea, August 20-24, 978-1-4244-3597-5.

[7] Hoffmann .F(2001). *Evolutionary algorithms for fuzzy control system design*, Proc. IEEE, vol. 89, p: 1318–1333.

[8] Mucientes .M, Alcala .R, Alcala-Fdez .J, and Casillas .J (2009*). Learning Weighted Linguistic Rules to Control an Autonomous Robot*, International Journal of Intelligent Systems, Vol. 24, Issue 3 p: 226-251.

[9] Nauck .D (2000). *Adaptive Rule Weights in Neuro-Fuzzy Systems*, Neural Computing & Application, Springer-Verlag London Limited, vol.9 p: 60-70.

[10] Rafael Alcal´a, Jes´us Alcal´a-Fdez, Mar´ıa Jos´e Gacto and Francisco Herrera. (2005). *Genetic Lateral and Amplitude Tuning of Membership Functions for Fuzzy Systems*, International Conference on Machine Intelligence, p: 589-595.

[11] Rafael Alcala, Oscar Cordon, and Francisco Herrera.(2003). *Combining Rule Weight Learning and Rule Selection to Obtain Simpler and More Accurate Linguistic Fuzzy Models*, Modelling with Words, LNAI 2873, Springer-Verlag Berlin Heidelberg, p: 44–64.

[12] Rafael Alcala, Jose Ramon Cano, Oscar Cordon ,Francisco Herrera, Pedro Villar , Igor Zwir (2003).*Linguistic modeling with hierarchical systems of weighted linguistic rules*, International Journal of Approximate Reasoning 32, p:187–215.

[13] Sivanandam S.N. and Deepa S.N (2008). *Introduction to Genetic Algorithms*, ISBN 978-3-540-73189-4 Springer-Verlag Berlin Heidelberg.

[14] Srinivasan Alavandar and Nigam M.J. (2008). *Fuzzy PD + I control of a six DOF robot manipulator*, Industrial Robot: An International Journal, Volume 35, Number 2, p: 125–132.

[15] Sufian Ashraf Mazhari, Surendra Kumar.(2008) . *PUMA 560 Optimal Trajectory Control using Genetic Algorithm, Simulated Annealing and Generalized Pattern Search Techniques*, International journal of electrical, computer and systems engineering 2;1, p:71-80.

[16] Sufian Ashraf Mazhari, Surendra Kumar. (2008). *Heuristic Search Algorithms for Tuning PUMA 560 Fuzzy PID Controller*, International journal of computer science 3;4. p: 277-286