AN OPTIMAL RULE GENERATION SCHEME USING GENETIC ALGORITHM FOR THE DESIGN OF A FUZZY LOGIC CONTROLLER

MR Kaimal
Professor & Head

R Rajesh
Research Scholar

PP Mohanlal
Scientist, VSSC

Department of Computer Science, University of Kerala, Kariavattom
Thiruvananthapuram - 695 581, dcsku@md2.vsnl.net.in, kollamrajeshr@www.com

Abstract : *Identification of a proper set of membership function and a proper rule base poses one of the major difficulties in the design of a fuzzy logic based control system. One of the methods suggested is based on singular value decomposition, which produces appropriate linear combinations of the membership functions in order to get new membership functions for a reduced rule base. This method is easily tractable when the number of variables in the antecedent part of the rule is two. In general, when we have more number of variables in the antecedent part, better methods have to be developed for the above purpose. A method for the design of a Fuzzy Logic Controller, which generates an optimal fuzzy rule set with minimum number of rules and minimum number of membership functions, using Genetic Algorithm is presented here. We use two encoding schemes, one in which each individual represents a set of fuzzy rules for the controller and the other scheme acts as a selector of the best subset of fuzzy rules. The method has been applied to the inverted pendulum model and the results are promising.*

1. INTRODUCTION

The essential feature of a Fuzzy Logic Controller (FLC) is a set of linguistic control rules related by the dual concepts of fuzzy implication and compositional rule of inference [2, 10]. The fuzzy logic control provides an algorithm, which can convert linguistic control strategy derived on the basis of an expert's knowledge into an automatic control strategy. Design of FLCs is generally concerned with the determination of (i) input and output variables, (ii) parameters of membership functions and (iii) fuzzy rules. In order to improve the performance of FLC, parameter tuning and optimization of rule set are necessary. Of these, selection of appropriate rule set is the most important one involved in the design of an efficient FLC.

Recent studies have shown that Genetic Algorithms (GAs) are good at evolvable or adaptive systems [3, 7]. GAs are simple and powerful general-purpose stochastic optimization methods (learning mechanisms) and are derivative free techniques. The individuals of the population are expressed in a ry/real form and the GA then manipulates these ıgs by using genetic operators (reproduction, ;sover, and mutation) to obtain improved tions (where the fittest individuals survive), until ɔptimal solution is obtained. GA considers many ıts in the search space simultaneously, instead single point, thus it has a reduced chance of verging to local minima. Due to their robustness, ɔd, efficiency and flexibility, GAs have been used arious engineering and business problems.

Studies on Self Learning Fuzzy Logic Controller ɔloped using reinforcement learning models were ɔrted by Berenji and Khedkar [2]. They proposed ɔneralized approximate reasoning based control ɔitecture where learning was achieved by grating a fuzzy inference system with a feed ward neural network. And they tuned the ameters using gradient descent methods. Berenji Khedkar's work was an extension of the ıptive Heuristic Critic (AHC) algorithm of Barto al. [1] to include a priory control knowledge ɔugh fuzzy inference systems. A self-learning fuzzy c that uses the genetic algorithm to determine consequent parts of the fuzzy rules was studied Chiang, Chung and Lin [3].

Identification of a proper set of membership ction and a proper rule base poses one of the jor difficulties in the design of a fuzzy logic based ɔtrol system. One of the methods suggested is ɔed on singular value decomposition, which ɔduces appropriate linear combinations of the ɔmbership functions in order to get new mbership functions for a reduced rule base [11]. is method is easily tractable when the number of iables in the antecedent part of the rule is two. In ɔeral, when we have more number of variables in antecedent part, better methods have to be veloped for the above purpose.

In this paper we present a new method for the sign of a Fuzzy Logic Controller, which will tomatically generate an optimal fuzzy rule set with inimum number of rules and minimum number of embership functions. We make use of GA for lecting the appropriate rules using two encoding hemes, one in which each individual represents a t of fuzzy rules and the other scheme is used for lecting the best subset of fuzzy rules for the controller. The method has been applied to the inverted pendulum model and the results are promising.

This paper is organized as follows. Sec. 2 presents the rule generation scheme. In Sec. 3, inverted pendulum model is discussed. In Sec. 4, the computer simulation results are presented. Finally, conclusions are drawn in Sec. 5.

2. RULE GENERATION SCHEME

This method consists of two encoding schemes. The first or primary encoding scheme is comprised of a population called as Primary Population (PP), and each chromosome in the PP is called a Primary Chromosome (PC). Each gene corresponds to a fuzzy rule, and each rule consists of centers and spreads of 5 linguistic variables - 4 variables in the antecedent and 1 variable in the consequent. The centers and the corresponding spreads form the parameters of the triangular membership function (we report here, for convenience, only the case of symmetric triangular membership functions). Fig. 1 shows the structure of primary chromosome, where m is the number of rules (genes). PP is generated randomly during the initializing phase of GA. In each generation, mutation and crossover make changes in PP.

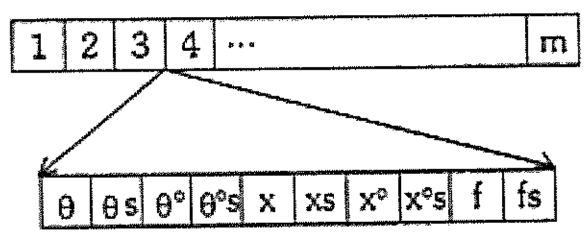


Fig. 1 – Structure of a primary chromosome

The secondary encoding scheme is comprised of a population called Secondary Population (SP) and each chromosome is called a Secondary Chromosome (SC). The number of genes in SC is equal to the number of genes in the PC. Each gene consists of a bit and corresponds to a possible rule from PC. Fig. 2 shows the structure of secondary chromosome. SP is generated randomly in each generation and mutation and crossover do not influence SP.

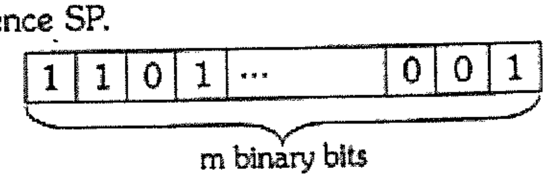


Fig. 2 – Structure of a secondary chromosome

Considering a SC and a PC, the rule selection criterion is as follows. If a '1' or '0' occur in a gene of the SC, then the associated rule in the PC is selected or discarded respectively, and thus a subset rules corresponding to a SC is selected for consideration.

The selected subset of the rules can be considered as the rule base for the fuzzy controller and the plant can be run until failure or maximum time is reached. The performance measure of the plant is taken as the fitness of the selected subset of the rules. The fitness function for measuring the performance is given in Eqn. 1, where mt is the time until which failure occurs; $\alpha_1, \alpha_2, \alpha_3$ are weights (we use $\alpha_1 = \alpha_2 = \alpha_3 = 1$) and; θ_d , θ^0_d, x^0_d are the desired values of θ, θ^0, x^0.

$$F = \frac{mt}{\sum_{t=1}^{mt} \alpha_1 [\theta(t)-\theta_d(t)]^2 + \alpha_2 [\theta^0(t)-\theta^0_d(t)]^2 + \alpha_3 [x^0(t)-x^0_d(t)]^2} \quad -> (1)$$

The algorithm for finding the fitness of PC's is also given.

Algorithm to find fitness of Primary chromosome

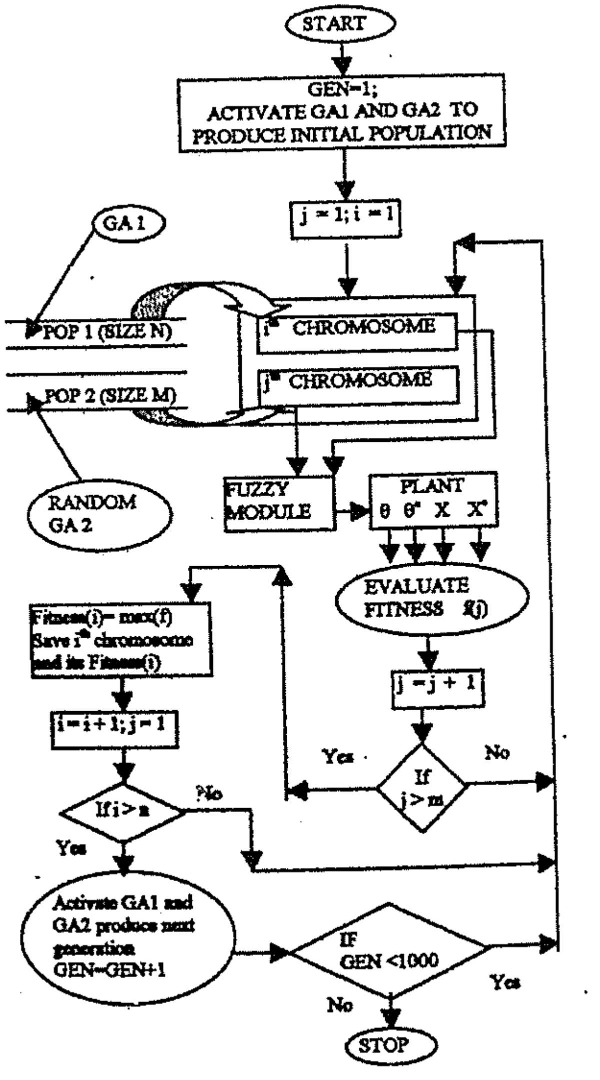
1. i = 1
2. Select the i^{th} primary chromosome
3. j = 1
4. Select the j^{th} secondary chromosome
5. Apply rule selection criteria with PC(i) and SC (j)
6. Sj ← fitness of the selected rule set
7. j = j + 1
8. If j < = k the goto 4, where k is the size of SP
9. f_i ← max ($S_1, S_2, \ldots\ldots, S_k$), where f_i is the fitness of i^{th} PC
10. i = i + 1
11. If i < = n then 2, where n is the size of PP

2.1 Selecting, Tuning and Optimizing the Membership Functions

The number of membership functions associated with a linguistic variable is not fixed. Each rule in the chromosome is coded to represent variables handling centres and spreads of five variables namely θ, θ^0, x, x^0 and f. So each rule can select its own membership functions associated with linguistic variables so as to perform in a better way.

Tuning of the parameters of MF takes place during the mutation phase of the Genetic Algorithm.

The number of membership functions associated with each input/output variable is in general, much less than the number of rules. The mutation phase of GA will make changes in the number of membership functions, its shape and position. The changes in the number of membership functions will take place when two membership functions belonging to two different rules become identical due to mutation. Moreover, during selection process if more than one chromosome occurs with same fitness, then the rule set (chromosome) with minimum number of membership functions and minimum number of rules will be selected as the best chromosome. And this will minimize the number of membership functions and the number of rules through generations



Flow chart of the proposed algorithm

;ART-POLE BALANCING MODEL

n this paper, we consider an inverted pendulum cart-pole system) as the control system. The :tives of the model are to keep the cart and the within certain given boundary values. The pole be balanced in the vertical position at $\theta = 0^0$ the cart is to be positioned in the range $\leq$ 2.4 m. The model equations are given by 2 and 3. The variables used in the model are: ›rizontal position of the cart; x^0 - velocity of the θ - angle of the pole with respect to the vertical θ^0 - angular velocity of the pole; f - force applied e cart; g - acceleration due to gravity; m_p - of the pole; m_c - mass of the cart and ℓ - half length. A failure occurs when $|\theta| > 12^0$ or > 2.4 m.

$$\frac{g \sin\theta + \cos\theta\left[\frac{-f - m\ell\,\dot{\theta}^2 \sin\theta}{m_c + m}\right]}{\ell\left[\frac{4}{3} - \frac{m\cos^2\theta}{m_c + m}\right]} \quad \text{-> (2)}$$

$$\ddot{x} = \frac{f + m\ell\left[\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta\right]}{m_c + m} \quad \text{-> (3)}$$

4. SIMULATION RESULTS

The implementation of the model is done on a Pentium II 350 MHz machine. The parameter values used in the simulations are: m = 160, ℓ = 0.5m, m_p = 0.1 kg, m_c = 1 kg, crossover rate = 0.7 and a mutation rate of 1/m. We have used the noisy roulette wheel selection scheme to increase randomness during reproduction. The shuffling crossover criteria is applied and a mutation operator of the Breeder Genetic Algorithm is used. The population sizes of primary population (n) and secondary population (k) are chosen as 20 and 10 respectively.

The values of a typical chromosome with 8 rules in the 14th generation are shown in the tables and simulation graphs in this paper. Membership function parameters of θ, θ^0 and x are given in Table 1. Table 2 shows the membership function parameters of x^0 and f. Table 3 shows the rules for the controller [The rules in the table, for example the 1st rule, is read as if θ is in T1, θ^0 is in TD1, x is in X1 and x^0 is in XD1 then force is F1]. Fig. 3 and Fig. 4 depict typical profiles of pole-angle and cart-position respectively. Several experiments were carried out by varying the intial pole-angle.

Table 1: Membership function parameters for θ, θ^0 and x in 14th generation

θ			θ			x		
lo	Center	Spread	No	Center	Spread	No	Center	Spread
1	-0.21	0.21	TD1	-1.00	1.00	X1	+1.40	2.40
2	+0.21	0.21	TD2	+1.00	1.00	X2	-2.40	1.40
3	+0.21	0.11	TD3	+0.50	0.50	X3	-1.40	2.40
4	-0.10	0.21				X4	+2.40	1.40
5	+0.11	0.21				X5	-1.40	2.16
6	-0.21	0.08						
7	+0.10	0.21						

Table 2: Membership function parameters for x^0 and f in 14^{th} generatiom

x^0			f		
No.	Center	Spread	No.	Center	Spread
XD1	-0.50	1.00	F1	-5.02	5.00
XD2	+1.00	0.50	F2	+5.00	4.99
XD3	+0.50	0.50	F3	+5.00	5.00
XD4	+1.00	1.00	F4	+0.00	5.01
XD5	-1.00	0.50	F5	+5.00	5.01
XD6	-1.00	1.00	F6	+0.02	5.00

Table 3: Rules for the controller

Rule No.	θ	θ^0	x	x^0	f
1	T1	TD1	X1	XD1	F1
2	T2	TD2	X2	XD2	F2
3	T3	TD2	X1	XD3	F3
4	T4	TD3	X3	XD4	F4
5	T5	TD3	X4	XD5	F5
6	T6	TD2	X5	XD5	F6
7	T7	TD2	X3	XD6	F3
8	T5	TD3	X4	XD5	F3

5. CONCLUSION

The studies with fuzzy models [2] and models with partial application of GA for tuning the consequent part of the fuzzy rules [3] suggests that the present model has several advantages over the other models. The application of GA in the selection and optimization of suitable rulesets for the design of FLC in two-stages is a promising step. In terms of computational complexity this method seems to be better than SVD based method.

The rule base obtained by our method is optimum as a rule set in the primary design stage. In the second stage we can make use of any one of the supervised learning schemes for fine-tuning rules so as to optimize the requirement of the FLC system.

ACKNOWLEDGEMENT

The authors are thankful to ISRO for financial support through a grant No. DOS/10/3/331/97. The authors are grateful to Dr. Das Gupta for fruitful discussions and encouragements.

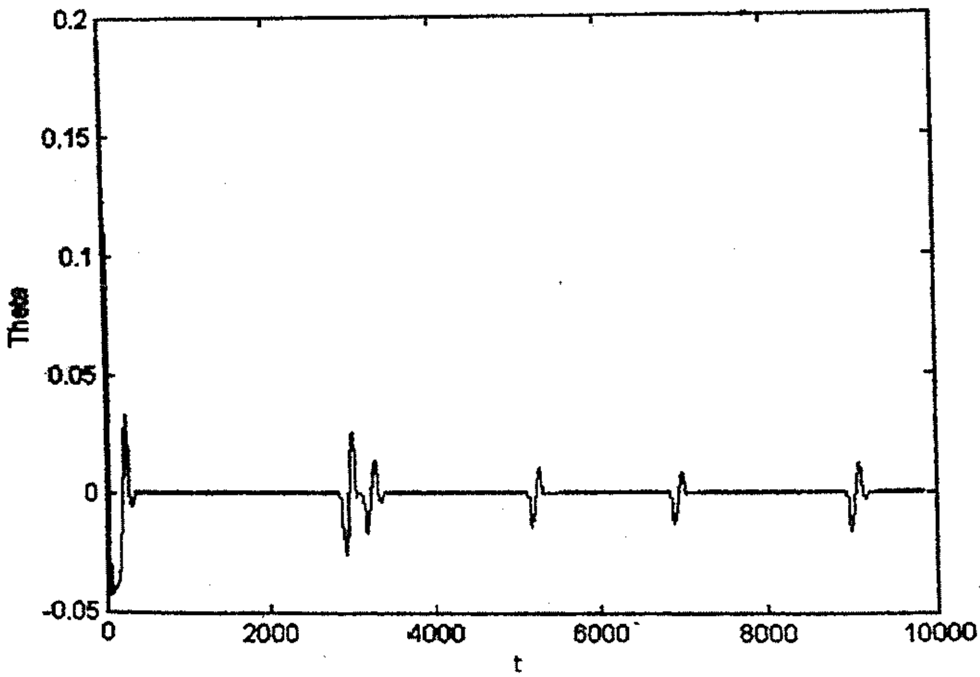


Fig.3 – The profile of pole-angle (θ). The pole attains 0 degree within 5 seconds. The oscillations in the pole angle are due to the force automatically applied by the controller so as to keep the cart position within boundaries

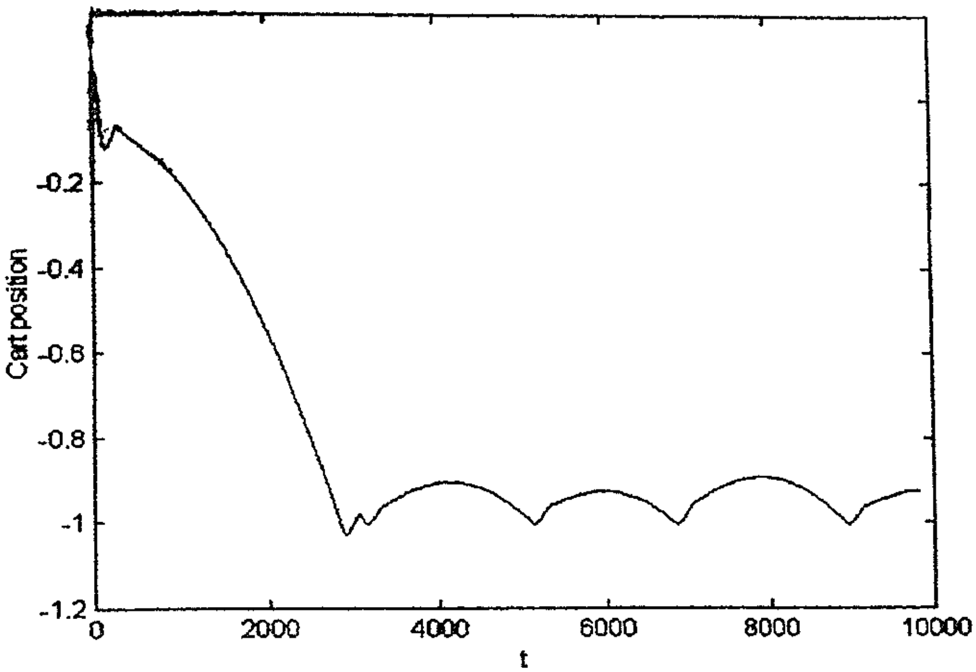


Fig.4 – Plot of Cart Position (x)

References

[1] Andrew G. Barto, Richard S. Sutton, Charles W. Anderson, Neuron like adaptive elements that can solve difficult learning control problems, IEEE Transactions on systems, Man and Cybernetics, vol. SMC-13, pp. 537-549, 1983.

[2] Hamid R. Berenji, Pratap Khedkar, Learning and Tuning Fuzzy Logic Controllers Through Reinforcements, IEEE Transactions on Neural Networks, vol. 3, No. 5, pp 724-739, September 1992.

[3] Chih. Kuan Chiang, Hung-Yuan Chung, Jin-Jye Lin, A Self-Learning Fuzzy Logic Controller using Genetic Algorithms with Reinforcements, IEEE Transactions on fuzzy systems, vol. 5, No. 3, pp. 460-467, August 1997.

[4] Jianbin Hao, Joos Vandewalle, Shahua Tan, A Rule-Based Neural Controller for Inverted Pendulum System, Int. Journal of Neural Systems, vol. 4, No. 1, pp. 55-64, 1993.

[5] T. Pal, N.R. Pal, S.D. Ray, A Self-organized Rule Generation Scheme for Fuzzy Controllers, IEEE 2000.

[6] Jyh-Shing R. Jang, Self-Learning Fuzzy Controllers Based on Tempor1. Back Propagation, IEEE Transactions on Neural Networks, vol. 3, No. 5, pp. 714-723, September 1992.

[7] M.A. Lee, H. Takagi, Integrating design stages of fuzzy systems using genetic algorithms, in Proc. IEEE Int. Conf. Fuzzy Syst., San Francisco, CA, pp. 162-617, Mar 1993.

[8] Elie Sanchez, T. Shibata, L.A. Zadeh, (eds.), Genetic Algorithms and Fuzzy Logic Systems soft computing Perspectives, World Scientific, 1997.

[9] J. Yen, R. Langari, L.A. Zadeh, Industrial applications of fuzzy logic and intelligent systems, IEEE press, 1995.

[10] L.A. Zadesh Fuzzy logic, Neural Networks, and Soft Computing, Commun. ACM, vol. 37, pp. 77-84, Mar. 1994.

[11] D. Simon, Design and rule base reduction of a fuzzy filter for the estimation of motor currents, International Journal of Approximate Reasoning, pp 145-167, 2000.