Co-evolutionary Genetic Algorithm for Designing and Optimizing Fuzzy Controller

Hyo-Byung Jun and Kwee-Bo Sim

Robotics and Intelligent Information System Laboratory
School of Electrical and Electronic Engineering, Chung-Ang University
221, Huksuk-Dong, Dongjak-Ku, Seoul 156-756, Korea
Tel:+82-2-820-5319, Fax:+82-2-817-0553
E-mail:kbsim@cau.ac.kr, URL:http://rics.cie.cau.ac.kr

ABSTRACT

In general, it is very difficult to find optimal fuzzy rules by experience when a system is dynamical and/or complex. Futhermore proper fuzzy partitioning is not deterministic and there is no unique solution. Therefore we propose a new design method of an optimal fuzzy logic controller, that is a co-evolutionary genetic algorithm finding optimal fuzzy rules and proper membership functions at the same time. We formalize the relation between fuzzy rules and membership functions in terms of fitness. We review the typical approaching methods to co-evolutionary genetic algorithms, and then classify them by fitness relation matrix. Applications of the proposed method to a path planning problem of autonomous mobile robots when moving objects exist are presented to demonstrate the performance and effectiveness of the method.

Keywords : Co-evolutionary Genetic Algorithm, Fuzzy Logic Controller, Fitness Relation Martix

I. Introduction

Natural evolution works on dynamic fitness landscapes that change over evolutionary time as a result of co-evolution. It is believed that co-evolution between different species or different organs results in the current state of complex natural systems. In this point, there is a growing interest in co-evolutionary systems, where two populations constantly interact and co-evolve in contrast with traditional evolutionary algorithms with single population.

Generally co-evolutionary algorithms can be classified into two categories, which are predator-prey co-evolution[1][2] and symbiotic co-evolution[3]. Predator-prey relation is the most well-known example of natural co-evolution. Hillis[1] proposed this concept with a problem of finding minimal sorting network for a given number of data. Also co-evolution between neural networks and training patterns was proposed in the concept of predator and prey[4]. A new fitness measure in co-evolution is studied in terms of dynamic fitness landscape. L. van Valen, a biologist, has suggested that the "Red Queen effect" arising from co-evolutionary arms races has been a prime source of evolutionary innovations and adaptations[5].

Symbiosis is the phenomenon in which organism of different species live together in close association, resulting in a raised level of fitness for one or more of the organisms. In contrast of predator-prey, this symbiosis has cooperative and/or positive aspects between different species. Paredis[3] proposed a symbiotic co-evolution in terms of SYMBIOT, which uses two co-evolving populations. One population contains permutations (orderings), the other one consists of solution candidates to the problem to be solved. Another approach to symbiotic co-evolution is host-parasite relation[6], which is based on the Schema Theorem and the Building Block Hypothesis. In [6], we derived an extended schema theorem associated with this host-parasite co-evolutionary algorithm.

In this paper, we propose a co-evolutionary method generating an optimal fuzzy controller, where the fitness of a population changes according to the evolution process of the other population. Basically fuzzy logic controller(FLC) is composed of fuzzifier, inference engine, rule base, and defuzzifier. A rule base is typically acquired via expert's knowledge. It is very difficult, however, to find fuzzy rules by hand when the system has a lot of input-output variables. It is even impassible when the complex and/or dynamic environment is considered. On the other hand, the proper fuzzy partitioning of input and output spaces and a correct choice of membership functions play an essential role in achieving a successful FLC design. Unfortunately, they are not deterministic and have no unique solutions.

Therefore, automatically generation of an optimal fuzzy rules and proper fuzzy partitioning is studied widely and a lot of approaches were proposed. Especially there has been a growing interest in genetic based machine learning(GBML) system, in other words classifier system[7][8]. There are two competing approaches to GBML. One is called Michigan approach[9], which uses a single set of production rules or classifiers. So each individual rule has a strength which indicates the utility of the rules to the goal of the system. The other is called Pitt approach, the individual of which consists of a set of rules[10][11]. In the context of optimal design of FLC, we should hold both of the fuzzy rules and membership functions in great account. Up to now, however, the fuzzy rules are only focused on, or the membership functions are incorporated into the fuzzy rules without considering the relation between them.

We propose here a new approach designing an optimal FLC, which is based on co-evolutionary genetic algorithms. In the next section, the relation between fuzzy rules and membership functions is formulated using the fitness function of co-evolutionary genetic algorithm. In section 3, we explain how to construct FLC using co-evolutionary genetic algorithms including DNA coding method[12]. Simulation conditions and some results are described in section 4, and finally conclusions are followed.

II. Co-evolutionary Genetic Algorithms and Fitness Relation

In contrast with traditional evolutionary algorithms with single population, co-evolutionary systems have two populations which constantly interact and co-evolve. Here, we formulate the relation of those two populations in terms of fitness.

2.1 Relation matrix between two populations

Let $X=\{x_1, x_2, \cdots, x_n\}$ be a *primary* population at a certain generation, and $Y=\{y_1, y_2, \cdots, y_m\}$ be a *secondary* population at the same generation. We used the term of *primary* to indicate a population having positive aspects with the objective function. Then $f_R(x, y)$ is a normalized fitness function that has $0 \le f_R(x, y) \le 1$. Since this fitness value represents the degree of fitness, it can be considered as a membership value of the fuzzy set 'fitness'.

Now we define a fitness relation matrix R as follows:

$$R(X, Y) = \begin{bmatrix} f_R(x_1, y_1) & f_R(x_1, y_2) & \cdots & f_R(x_1, y_m) \\ f_R(x_2, y_1) & f_R(x_2, y_2) & \cdots & f_R(x_2, y_m) \\ \vdots & & \cdots & \vdots \\ f_R(x_n, y_1) & f_R(x_n, y_2) & \cdots & f_R(x_n, y_m) \end{bmatrix} \quad (1)$$

where $f_R(x_i, y_i)$ is the fitness value acquired by the individuals x_i and y_i, and n, m are the sizes of primary and secondary populations, respectively.

Several fuzzy sets are combines to produce a single set by an aggregation operation on fuzzy sets which is defined by[13]

$$h: [0,1]^k \rightarrow [0,1], \quad k \ge 2 \quad (2)$$

such that

$$\mu_A(x) = h(\mu_{A_1}(x), \mu_{A_2}(x), \cdots, \mu_{A_k}(x)), \quad \forall x \in U \quad (3)$$

where A_k is a fuzzy set in the universe of discourse U and $\mu_{A_k}(x)$ is the grade of membership of x in A_k. Generally, the aggregation operators are called averaging operators if they lie between the min operator and the max operator, such as

$$\begin{aligned} \min(a_1, a_2, \cdots, a_n) &\le h(a_1, a_2, \cdots, a_n) \\ &\le \max(a_1, a_2, \cdots, a_n) \end{aligned} \quad (4)$$

where $a_i = \mu_{A_i}(x)$, $i=1, \cdots, k$. One typical parametric averaging operator is the generalized means which is defined as

$$h_a(a_1, a_2, \cdots, a_k) \triangleq \left(\frac{a_1^a + a_2^a + \cdots + a_k^a}{k} \right)^{1/a} \qquad (5)$$

where a is a real number but $a \neq 0$. The generalized means covers the entire interval between the min and the max operators, because when a approaches $-\infty$, $h_a(a_1, a_2, \cdots, a_k)$ becomes $\min(a_1, a_2, \cdots, a_k)$, and when a approaches ∞, $h_a(a_1, a_2, \cdots, a_k)$ becomes $\max(a_1, a_2, \cdots, a_k)$.

In the next sub-section we define and classify the categories of the co-evolutionary algorithm using the fitness relation matrix and the aggregation operators. Also we extract the boundaries of the system's performance from the generalized means operator.

2.2 Classification of the co-evolutionary algorithm using relation matrix

We will now classify and define the categories of the co-evolutionary algorithms using the above fitness relation matrix. We call a co-evolutionary algorithm a *promotive(cooperative)* one if the following conditions are satisfied:

$$f_{[R\downarrow X]}(x) = \underset{y}{h_a} f_R(x, y) \qquad (6)$$

$$f_{[R\downarrow Y]}(y) = \underset{x}{h_a} f_R(x, y) \qquad (7)$$

where $f_{[R\downarrow X]}(x)$ and $f_{[R\downarrow Y]}(y)$ are the fitness functions of the primary population and the secondary population, respectively, and down arrow means the generalized projection of R onto each population. Also a co-evolutionary algorithm is called a *suppressive(competitive)* one if the following conditions are satisfied:

$$f_{[R\downarrow X]}(x) = \underset{y}{h_a} f_R(x, y) \qquad (8)$$

$$f_{[\overline{R}\downarrow Y]}(y) = \underset{x}{h_a} f_{\overline{R}}(x, y) \qquad (9)$$

where $\overline{R}$ is the complement of the relation matrix R, defined by the fitness function, such as

$$f_{\overline{R}}(x, y) \triangleq 1 - f_R(x, y). \qquad (10)$$

Easily we can see from the above equation that the fitness direction of the secondary population is opposite to that of the primary one in the suppressive co-evolutionary algorithm.

Also the performance boundaries of the system can be found from the aggregation operator h_a. If h_a is max, a fitness value of a certain individual indicates the upper boundary of that individual's capacity at the given time. If h_a is min, in the other hand, a fitness value of a certain individual indicates the lower boundary of that individual's capacity at the given time.

Ⅲ. Co-evolution of Fuzzy Rules and Membership Functions

This paper presents a new approach to automatic generation of FLC based on the concept of co-evolution algorithms. Our approach has two parallel evolution processes which are rule base (RB) population and membership function(MF) population. The overview of our approach is illustrated in Fig. 1.

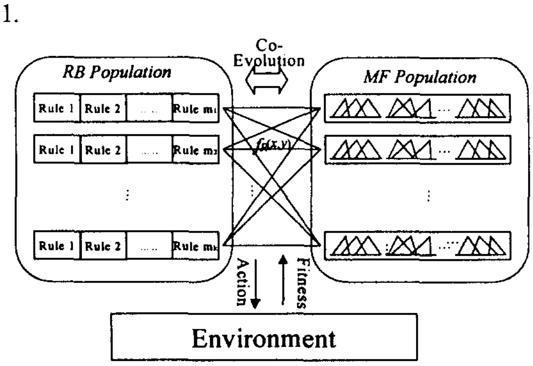


Fig. 1. A block diagram of co-evolution of rule bases and membership functions

To apply genetic algorithms to any problem, first the solution spaces should be represented by a chromosome. For our case, the encoding methods and genetic operators are explained in the following sub-sections.

3.1 Rule Base Population

The individual of the rule base population consists of a set of rules, so there are sets of rules in the rule base population. If membership functions are partitioned into T terms and there are i preconditions, then the maximum number of IF-THEN fuzzy rules is T^i. This means that the input space is divided into T^i. Therefore, unless we use all of the rules, null set problems occur when the given rule base cannot cover the current input states. We use the DNA coding method[12], therefore, which is very useful to represent a set of linguistic rules.

The biological DNA consists of nucleotides which have four bases, Aenine(A), Guanine(G), Cytosine(C), Thymine(T). A messenger RNA is synthesized from the DNA. In the RNA the base U is used instead of T. Three successive bases called codons are allocated sequentially in the messenger RNA. These codons are the codes for amino acids. 64 kinds of codons correspond to 20 different kinds of amino acids. The details of the translation into amino acid from codons are omitted here. Table 1 shows the correspondence between the amino acids and the parameters defined in this paper.

The DNA chromosome makes up sets of fuzzy rules for controlling a mobile robot. Fig. 2 shows an example of the DNA chromosome and the corresponding fuzzy rules. The DNA coding method has the features such as flexible representation of knowledge, redundant and overlapped coding, variable length of the chromosome, and no constraint on crossover points.

Table 1. Correspondence between the Amino Acids and the Parameters

Amino Acid	Phe	Leu	Ile	Val	Ser	Pro	Thr	Ala	Tyr	His	Gln	Asn	Lys	Asp	Glu	Cys	Trp	Arg	Gly
Number of Inputs	4				3		4		1				3			2	4	2	
Input Parameters	θ				S0			S1						S2		S0	S1	S2	
Position of Predicates	LT		ME	LT	ML		MR		ME				ML	MR		RT	ME	RT	

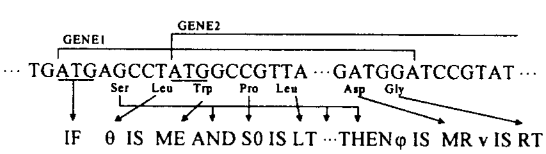
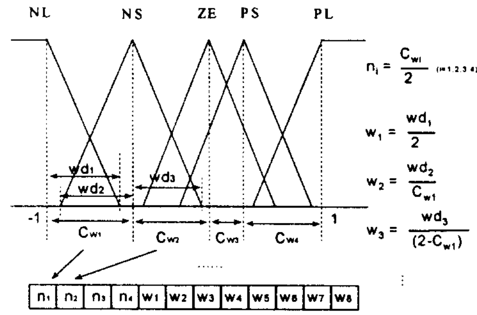


Fig. 2. An Example of DNA coding method

3.2 Membership Function Population

As shown in Fig. 3, we use the normalized membership function partitioned with five terms. The shape of each term is triangular except the two marginal terms. For our case, the encoding method is illustrated in Fig. 3. The triangular membership function's shape is determined by the three points that are a center point and left/right width points. We assume that the NL and PL terms have fixed center points and the other three center points could be placed any position from -1 to 1 and all the left/right width of each terms could be from 0 to the maximum value from its center point to the margin.



NL : Negative Large, NS : Negative Smal, ZE : Zero, PS : Positive Small, PL : Positive Large

Fig. 3. A membership function and encoding scheme

For a variable the chromosome is consist of (number of terms - 1)×3 bits real-valued string, where the first 4 bits represent the width proportion between the neighbor center points and the last 8 bits represent the width ratio of each term's left and right margin from its center point. For example, w_3 representing NS term's right width ratio is current right width(wd_3) over its possible maximum width($2-C_{w1}$). If there are N terms, N_i input variables, and N_o output variables, then the whole length of one chromosome becomes $3\times(N-1)\times(N_i+N_o)$ bits. This encoding method guarantees the completeness, soundness, and non-redundancy between the solution and the genotype spaces. Fitness proportionate reproduction method and both of mutation and crossover are used as genetic operators.

Ⅳ. Path planning of AMR

We verify the effectiveness of the proposed algorithm by applying it to optimal path planning of autonomous mobile robot. The objective of this problem is to find a optimal path when static and moving obstacles exist. For the moving obstacle we assumed that there are two robots with the same FLC at the counterpart coner. Each robot's goal position is set to the other robot's starting point and perceives the other robot as an obstacle. A robot has three sensors(S0,S1,S2) covering $\pm 15^\circ$ to detect the distance to an obstacle. The direction of its goal(θ) is given, so there are four input variables. For the outputs, FLC gives the directional changes(φ) and speed(υ) of AMR. The simulation environmental conditions are set as follow:

- Working area : 1500×1500mm
- Robot Size : radius 25mm
- Number of robots : 2 units
- Maximum speed : 30mm/step
- Sensing Radius : 200mm
- Maximum steering angle : 90°/step

The input/output variables' ranges are restricted as shown in table 1. Fig. 4 shows the AMR's sensor configuration and situations of detecting an obstacle.

Table 1. Ranges of input/output variables

INPUT				OUTPUT	
θ	S0	S1	S2	φ	υ
-180° ~180°	0~200 mm	0~200 mm	0~200 mm	-90° ~ 90°	0~30 mm

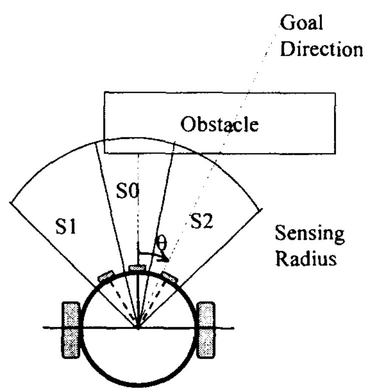


Fig. 4. Sensor Configuration

The raw fitness measure is formulated by,

$$f_R = \left(1-\frac{D_r}{D_G}\right)\cdot\frac{T_{\min}}{T}\cdot\frac{(N_N - N_n)}{N_N} \qquad (4)$$

where T is consuming time, N_n is the number of null set, T_{min} is minimum time required to reach the goal, and N_N is maximum number of null set. The fitness functions of membership function and rule base are set by,

$$f_{[R\downarrow X]}(x) = \underset{y}{h_a}\, f_R(x,y) \qquad (5)$$

$$f_{[R\downarrow Y]}(y) = \underset{x}{h_a}\, f_R(x,y) \qquad (6)$$

where $f_{[R\downarrow X]}(x)$ is the RB individual's fitness and $f_{[R\downarrow Y]}(y)$ is the MF individual's fitness. In our case, the number of rule and membership function populations is set for 50. The mutation probability of rule is 0.2, the crossover and mutation probability of membership function are set for 0.5 and 0.02, respectively.

Fig. 5 shows the resulting fitness changes versus generations. Also Fig. 6 shows the membership functions obtained after 500 generations. The obtained rules after 500 generations are stated in table 2. This rule base can cover all possible states and means that *'Turn to the goal direction, and if an obstacle exist in the direction of moving then turn left or right although opposite direction to the goal position'*. Fig. 7 shows the dynamic fitness landscapes, where figure (a) illustrates elite rule's fitness changes according to the changes of membership functions' generations, and figure (b) illustrates fixed elite membership function's fitness changes versus the changes of rules' generation. Using above rule base and membership functions both AMRs find their goal positions in relatively short time and avoid obstacles successfully.

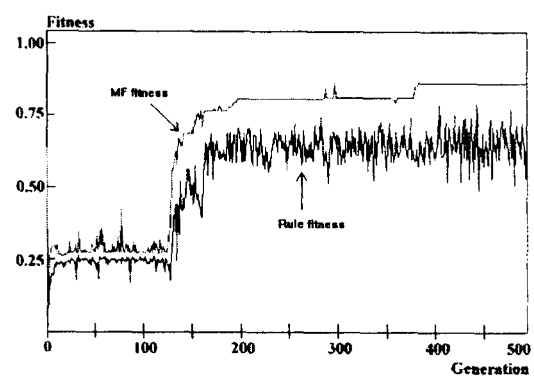
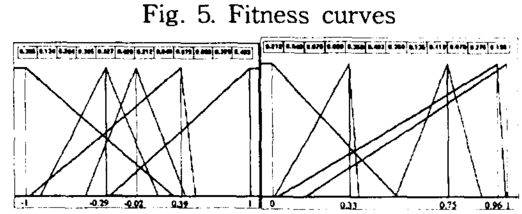
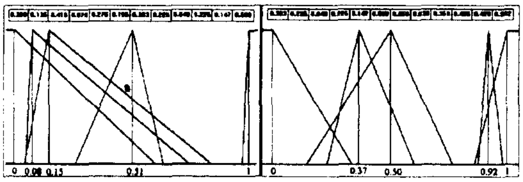


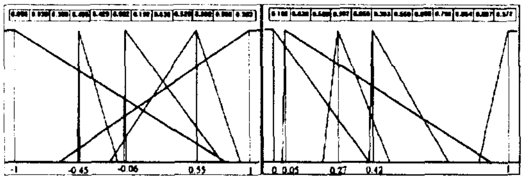
Fig. 5. Fitness curves



(a) Input variable θ (b) Input variable *S0*



(c) Input variable *S1* (d) Input variable *S2*

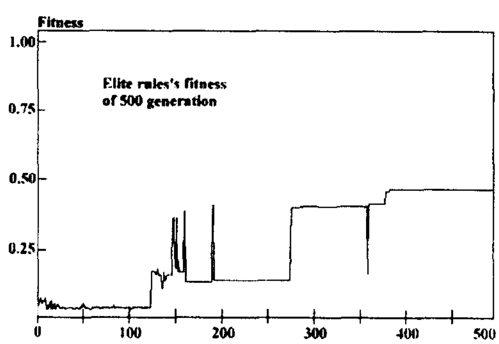


(e) Output variable φ (f) Output variable υ

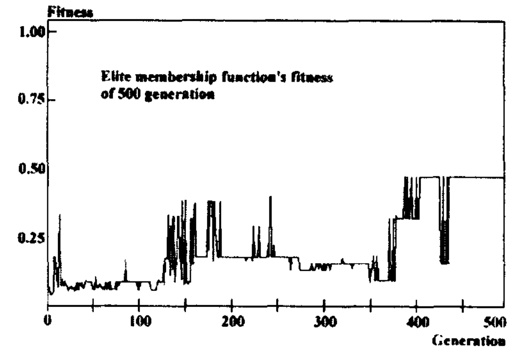
Fig. 6. Evolved membership functions

Table 2. Rule base after 500 generations

R1 : *IF S0 is VS, S1 is ME, and S2 is ME, THEN* φ *is NL and* υ *is VL.*
R2 : *IF* θ *is PS, S0 is MS, S1 is ME, and S2 is MS, THEN* φ *is PS and* υ *is MS.*
R3 : *IF S0 is MS, S1 is ME, and S2 is ME, THEN* φ *is PS and* υ *is ML.*
R4 : *IF S0 is ME THEN* φ *is PL and* υ *is ME.*
R5 : *IF S2 is ME, THEN* φ *is ZE and* υ *is ME.*
R6 : *IF S0 is VL and S2 is VS, THEN* φ *is NL and* υ *is VL.*
R7 : *IF S1 is VS, and S2 is MS, THEN* φ *is PS and* υ *is VS.*
R8 : *IF S0 is MS, S1 is ML, and S2 is ML, THEN* φ *is PS and* υ *is ME.*
R9 : *IF S2 is ML, THEN* φ *is ZE and* υ *is ML.*
R10 : *IF S1 is ML and S2 is ML, THEN* φ *is NL and* υ *is MS.*
R11 : *IF* θ *is NL, S0 is VL, and S2 is MS, THEN* φ *is NS and* υ *is ME.*
R12 : *IF* θ *is NL and S1 is ML, THEN* φ *is NL and* υ *is VS.*



(a) Fitness landscape of 500 generation elite rule



(b) Fitness landscape of 500 generation elite MF

Fig. 7. Dynamic fitness landscapes

V. Conclusions

This paper has proposed a new approach to automatically fuzzy logic controller generation using co-evolution concept. By applying the proposed method to a optimal path planning problem where moving obstacle exit, the effectiveness of the proposed method was shown. The concept of co-evolution is reviewed on the points of artificial life computation model. Two main process in co-evolution are optimal rule base generation and proper fuzzy membership function partitioning. Each population evolves according to the other's evolution process. This evolution model is considered as more analogous to natural system.

However, the relation between fitness functions should be extended more generally when more than one population evolve. That remains the future work.

Acknowledgement

This paper is supported by Institute of Information Technology Assessment, Korea.

References

[1] W. Daniel Hillis, "Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure," *Artificial Life II*, Vol. X, pp.313-324, 1991.

[2] Seth G. Bullock, "Co-evolutionary Design: Implications for Evolutionary Robotics," *The 3rd European Conference on Artificial Life*, 1995.

[3] JanParedis,"Co-evolutionary Computation," *Artificial Life*, Vol. 2, No. 4, pp.355-375,

1995.
[4] D.W. Lee, H.B. Jun, K.B. Sim, "A Co-Evolutionary Approach for Learning and Structure Search of Neural Networks," *Proc. of KFIS Fall Conference '97*, Vol. 7, No. 2, pp. 111-114, 1997.
[5] D. Cliff, G. F. Miller, "Tracking The Red Queen: Measurements of adaptive progress in co-evolutionary simulations," *COGS Technical Report CSRP363*, University of Sussex, 1995.
[6] H.B. Jun, D.J. Kim, K.B. Sim, "Structure Optimization of Neural Network using Co-evolution," *Journal of the KITE*, Vol. 35, No. 4, 1997.
[7] J.H. Holland, *Adaptation in Natural and Artificial Systems : An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, A Bradford Book, The MIT Press, 1994.
[8] M. Dorigo, U. Schnepf, "Genetics-based Machine Learning and Behaviour Based Robotics: A New Synthesis," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 23, No. 1, pp.141-154, 1993.
[9] J. H. Holland, J. S. Reitman, "Cognitive Sytems Based on Adaptive Algorithms," in *Pattern Directed Inference Systems*, Academic Press, New York, pp. 313-329, 1978.
[10] S.F. Smith, "A Learning System Based on Genetic Adaptive Algorithms", Ph. D. Thesis, Univ. of Pittsburgh, 1980.
[11] J. Kinzel, F. Klawonn, R. Kruse, " Modifications of Genetic Algorithm for Designing and Optimizing Fuzzy Controllers," *ICEC '94*, Vol. 1, pp. 28-33, 1994.
[12] R. Deaton et. al, "A DNA Based Implementation of an Evolutionary Search for Good Encodings for DNA Computation," *Proc. of '97 Magnetics Conference*, pp.267-271, 1997.
[13] Chin-Teng Lin and C.S. George Lee, *Neural Fuzzy Systems : A Neuro-Fuzy Synergism to Intelligent Systems*, Prentice Hall PTR, 1996.