Genetic Algorithm Based Fuzzy Logic Controller for Real Time Liquid Level Control

B Subudhi, *Associate Member*
A K Swain, *Non-member*

This paper describes the design and tuning of a fuzzy logic controller (FLC) for controlling the level of water on real time basis, in a laboratory scale pilot plant using genetic algorithm (GA). Selection of appropriate membership functions (MFs) for an FLC is iterative and a time consuming task. Here, MF of the FLC condition and action variables are selected optimally by using genetic algorithm, a probabilistic search technique inspired by mechanisms of natural selection and genetics. The optimal selection of MFs is made by minimizing integral time square error (ITSE), an indication of FLC performance. The performances of both genetic algorithm based FLC (GA-FLC) and traditional FLC are compared.

Keywords : Fuzzy Logic Controller, Genetic Algorithm, Membership Functions, Universes of Discourses, Simulation

INTRODUCTION

Despite the advancements of control theory in the last three decades, a large class of problems are solved by heuristics, developed by practicing engineers and operators over the years. Fuzzy logic is a technique which incorporates above heuristics into automatic control. Fuzzy logic is used to control highly nonlinear, complex systems or systems whose mathematical models are not known. Also in the situations where classical control methods are available, fuzzy logic is intorduced to imporve the controller performance and in some cases to simplify the control algorithm. It is verified experimentally, that the fuzzy controllers perform better than or as good as a PID controller[1-9]. For proper design of fuzzy logic controller (FLC), the membership functions (MFs) should be tuned until MFs perform acceptably for a spectrum of conditions that could exist in the controlled system. FLCs incorporate the linguistic relations between process input and output variables through linguistic terms such as positive medium (PM) or negative medium (NM). However, categorization of linguistic variables leads to some uncertainties, as fuzzy sets mean different things to different people. Fuzzy MF approximation of the confidences solves the problem of uncertainty by converting linguistic variables into precise numeric forms. There are many shapes of MFs as reported in the literature[10,11]. In this paper triangular MFs for both condition and action variables are considered. The measured plant variables are mapped into computational universes of discourses (UoDs) by scale factors, K_E, K_{CE} and K_{DU}, for computational simplicity[12]. The motivation of this work is to,

i) determine the effectiveness of the use of GA search strategy in designing FLC;

ii) select MF parameters optimally for better controller performance.

Iterative procedure for selection of MF is a very much time consuming task[7], because for the change in MF parameters, *ie*, for different shapes of the memebership functions the performance of FLC changes. Use of numerical techniques as a design tool require a lot of derivative information. Particularly, it is very difficult to produce derivative information for different fuzzy rules and membership function definitions in a FLC.

This work exploits GA search strategy to select MF optimally. GA requires only information concerning the quality of solution, produced by each set of parameters in the search process. GA search is basically a computer simulated evolution, which is used to alter the MFs of conventional FLC from one generation to next one, while optimizing integral time square error (ITSE). Here, ITSE is treated as a fitness function to link the optimization problem with GA. This GA based FLC (GA-FLC) is used to regulate the water level in a tank around a desired set point by controlling the position of a stepper motor operated controlled valve. The changing process dynamics are introduced by altering the set point for water level.

PROCESS DESCRIPTION

Fig 1 gives the schematics of the pilot plant considered for this work. A capacitance type level sensor is submerged down to the bottom of the tank. Water is being circulated through a centrifugal pump kept at constant speed. Water flow in the pipe is controlled by a stepper motor driven needle valve and manual valves. Needle valves are coupled with a separate stepper motor which is controlled by a stepper controller card. The level sensor and stepper motor are interfaced with an IBM compatible PC/XT. There is a bypass outlet between exit of the pump and needle valve position. Delay coil is an extra arrangement for introducing additional transportation lag in the system. Needle valve is the actuator for the level control system. The transfer function model of this process has been derived using two point method[12,14],

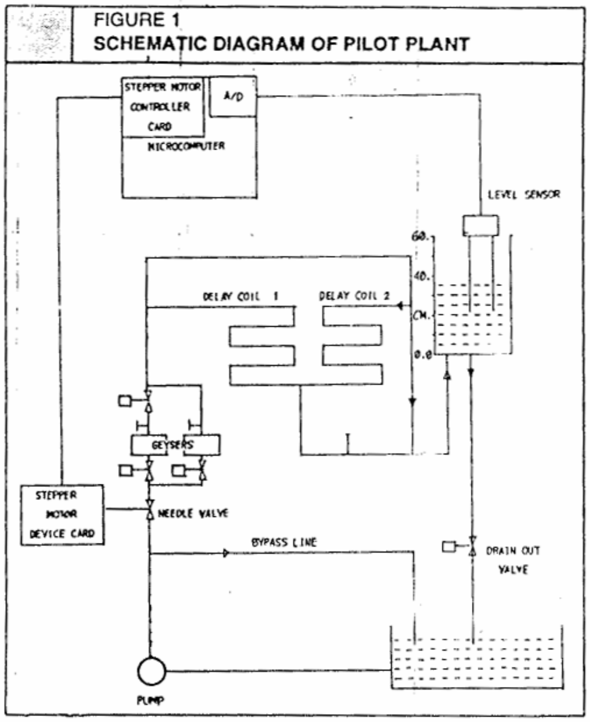
$$G(s) = \frac{Y(s)}{U(s)} = \frac{K_p\, e^{-t_d s}}{1 + Ts} \tag{1}$$

where *G (s)*, open loop transfer function of the tank; *Y (s)*, the open loop step response of water level in the tank, in cm; *U (s)*, the number of steps of stepper motor; K_p, Process gain = 4.2; t_d, time delay = 20 s; and T, time constant = 210 s.

B Subudhi and A K Swain are with Department of Electrical Engineering, Indira Gandhi Institute of Technology, Sarang 759 146, Orissa.

This paper (revised) was received on November 3, 1994. Written discussion on this paper will be received till October 31, 1995.

FIGURE 1
SCHEMATIC DIAGRAM OF PILOT PLANT



GENETIC ALGORITHM

GA is a computational model of simulated evolution used for optimization. The main idea behind GA is to start with an initial population of solutions and then attempt is made to get improvements in subsequent generations, manifesting the survival of the fittest mechanism. This survival of fittest or natural selection mechanism is used to simulate evolution of GA search to take place in a population of M numbers of chromosomes, each of length *l*. Fig 2 shows how a GA works through a simple cycle comprising of following four stages,

i) initialization of population,

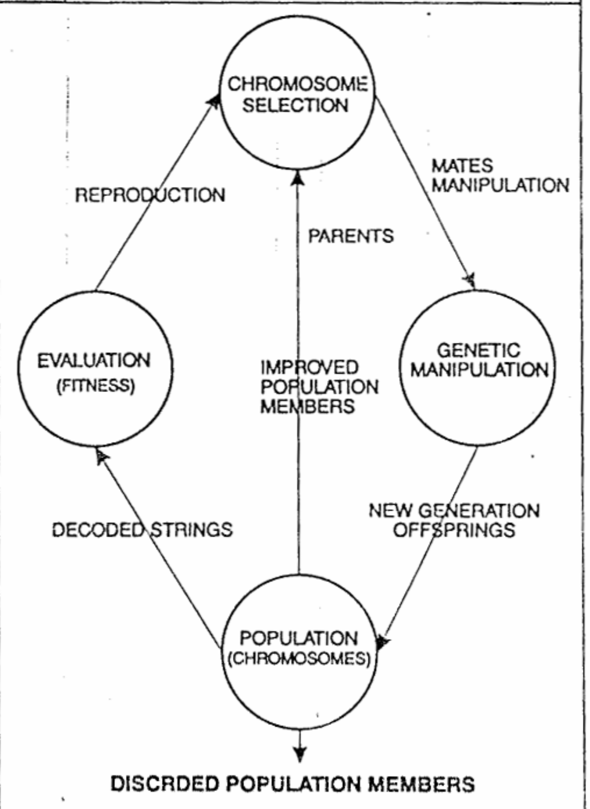
ii) evaluation of individual's fitness in a population,

iii) selection of best population members, and

iv) genetic manipulation.

To initiate the simulated evolution, the solutions to the problem are encoded on chromosomes. The parameter set in this study consists of entire set of anchor points, that locate the triangles defining the fuzzy sets N, Z, and P. The coding technique adopted in this study is concatenated unsigned integer coding of anchor points to take care of multiple parameter coding[11]. GA is used either to expand or shrink the base widths of fuzzy triangles *N*, *Z* and *P*. The extreme fuzzy sets *N* and *P* require the definition of only one anchor point, since one set of triangles is fixed at the limiting values of the variables. Two anchor points are required to describe interior triangle, defining fuzzy set Z. GA-FLC design for this process requires the selection of 9 parameters. The FLC is designed with 3 term sets *N* (negative), *Z* (zero) and *P* (positive). The fuzzy triangles for *N*,*Z* and *P* sets, used for both process input and output variables are shown in Fig 3.The parameters of the GA search space are the anchor points L_{ij}, C_{ij} and R_{ij} left, center and right points of the base of triangles *N*, *Z* and *P*. Let the universes of discourses for error *e*, change in error *ce* and change in control action *du* are *E*, *CE* and *DU*, respectively.

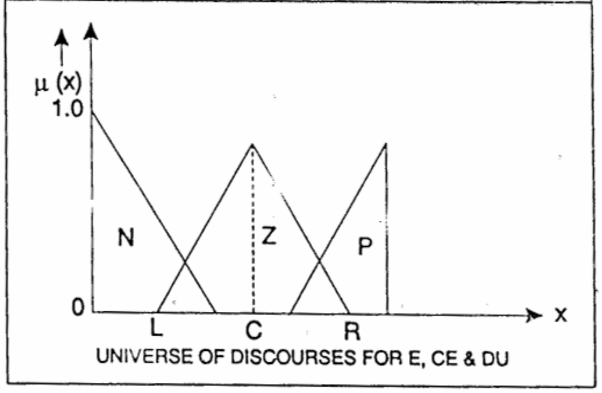
FIGURE 2
WORKING OF A GENETIC ALGORITHM



The universes are discretized into certain number of segments or quantization levels. In Fig 3, a common UoD is taken to represent UoDs *E*, *CE* and *DU*, and the variable *x* represents *e*, *ce* or *du*. Here the membership function of *x* takes different values for different fuzzy sets (i = 1,... 3 for three fuzzy sets *N*, *Z* and *P*) and quantization levels (j = 1, ..., 49). Each chromosome, *ie*, one set of nine anchor points is decoded[12] to give membership functions according to a particular input X_{ij} using equation (2).

$$\begin{aligned} \mu\,(X_{ij}) \quad & 1 - M_{L_{ij}}(C_{ij} - X_{ij}), \text{ if } L_{ij} \le X_{ij} \le C_{ij} \\ & = 1 - M_{R_{ij}}(X_{ij} - C_{ij}), \text{ if } C_{ij} \le X_{ij} \le R_{ij} \\ & = 0 \qquad , \text{ if } X_{ij} \le L_{ij} \text{ or } X_{ij} \le R_{ij} \end{aligned} \tag{2}$$

FIGURE 3
MEMBERSHIP FUNCTIONS FOR FUZZY SETS N, Z AND P



where $i = 1, \ldots, 3$, for three fuzzy sets, $j = 1, \ldots, 49$, for quantization levels of a variable and X_{ij}, a FLC condition or action variable of ith fuzzy set and jth quantization interval; $M_{L_{ij}}$, slope of MF to the left of the center point; and $M_{R_{ij}}$, slope of MF to the right of the centre point.

A simple GA uses three operators such as reproduction, crossover and mutation for its search process. These operators are implemented by generating random number, copying or exchanging some portion of strings or chromosome, and altering a particular integer of a string. The population of chromosomes is of fixed size of length l.

Population Pool

A single run of GA starts with randomly generating an initial population $P\ (g0)$, *ie*, at $g = g0$, with chromosomes. Each chromosome represents a possible solution to the problem, *ie*, one set of fuzzy MFs. $P\ (g1)$ is the population at generation $g = g1$. The main loop of GA consists of generating a new population $P\ (g1)$, applying the three GA operators to the existing population. The reproduction operator selects indi-viduls for $P\ (g1)$ from $P\ (g0)$, according to each individual's fitness in $P\ (g0)$. The members with better fitness values may receive one or more number of copies in $P\ (g0)$. The members with minimum fitness receive less, even no copies of themselves. These MFs are used for both fuzzification and defuzzification. For each simulated run of the controlled simulated plant a fitness value is computed. Selection of new strings for next generation is made according to string's fitness. Genetic manipulation is carried out on these newly reproduced strings by crossover and mutation probabilistic approach, with a view to obtain efficient MFs in next generation. The new strings are again decoded, evaluated and MFs are computed, and this cycle continues till a desired solution is achieved or the simulation length is reached.

Reproduction

Reproduction operator is implemented through an algorithm. Several methods are reported in literature[11] to select best strings for next generation. This paper uses the tournament selection method in which a string is picked up at random from the population pool. The fitness value of this string is compared with that of the adjacent string. The string with best (minimum ITSE) fitness value is selected.

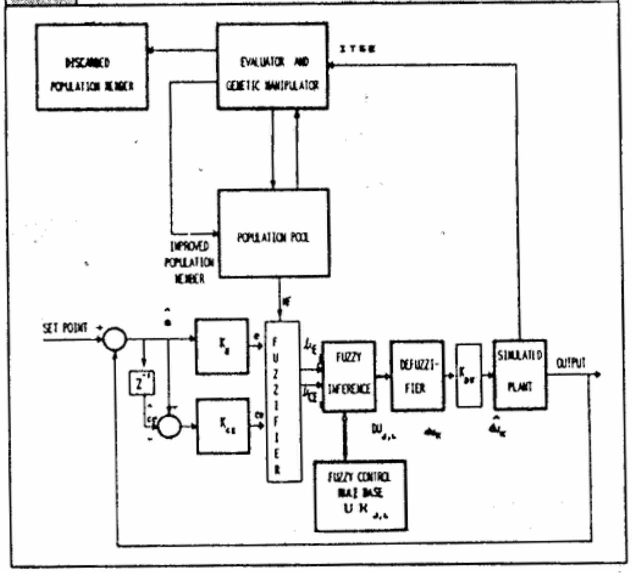
Crossover

After selection, the strigns are stored in a mating pool, where the crossover and mutation operators are applied in a probabilistic way. The crossover operator provides a mechanism for strings to exchange information through probabilistic decision. It first picks up two newly selected strings from the mating pool, produced by reproduction, and determines whether crossover is to be applied or not, from the outcome of a tossed biased coin with biasing P_{cross}, crossover probability. Then a position along the two strings is selected uniformly at random. Finally in the process of crossover the information following the crossing site are exchanged.

Mutation

Though reproduction and crossover do a lot for searching an optimal solution, mutation enhances the GA's ability in locating near optimal solution. Mutation has been implemented by replacing the value of a random integer position by a random number, if at all the mutation probability P_{mut} satisfies.

FIGURE 4
STRUCTURE OF GA-FLC



GA—FLC Design

Fig 4 shows the structure of a GA-FLC. Scale parameters such as $K_{E,}K_{CE}$ and K_{DU} are used to transform the values of *e* , *ce* and *du* in actual U_oDs to respective computational U_oDs for simplified calculation. To accommodate the variables *e*, *ce* and *du* in one U_oD with 49 quantization levels, appropriate values of $K_{E,}K_{CE}$ and K_{DU} are selected. In the initial prototype FLC design the MFs for linguistic variables *N*, *Z* and *P* are selected heuristically as shown in Fig 3. Fuzzifier receives the error and change in error at each sampling instant, and fuzzifies them using MFs as selected by GA.

Fuzzy Reasoning

The fuzzy control rule base consists of a set of statements, relating input variables and the corresponding output variables, to control the process. These statements are in the form of production rules, and let the *i*th rule is R_i which can be expressed as,

$$R_i : \text{If error } (e) \text{ is } E_i \text{ and change in error } (ce) \text{ is } CE_i \text{ then change in control action } (du) \text{ is } DU_i \quad (3)$$

where E_i, CE_i and DU_i are the fuzzy sets for *e*, *ce* and *du*, respectively.

R_i can be expressed as a fuzzy implication given by,

$$R_i : E_i \rightarrow CE_i \rightarrow DU_i \quad (4)$$

Let the universes of discourses for error, change in error and change in control are *E*, *CE* and *DU*, respectively.

R_i can be expressed as a fuzzy relation on *E*, *CE* and *DU* spaces, and is given by the cartesian product,

$$R_i = E_i \times CE_i \times DU_i \quad (5)$$

The MF of R_i, *ie*, μ_{R_i} is defined as,

FIGURE 5
FUZZY CONTROL RULE BASE

E_j ↓ CE_l →	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

$$\mu_{R_i} = \min (\mu_{E_i}(e), \mu_{CE_i}(ce), \mu_{DU_i}(du)) \qquad (6)$$

Using compositional rule of inference[15,16] for fuzzy reasoning, the consequence of each rule for measured values E' and CE' is inferred by the formula[16]

$$DU'_i = (E' \times CE')\ o\ R_i \qquad (7)$$

o is composition operator.

The fuzzy control algorithm contains 'r' numbers of fuzzy rules. Therefore, the overall relation matrix R, can be obtained OR-ing the individual rules, and has been given in equation (8).

$$\begin{aligned} R &= R_1\ U\ R_2\ U\ R_r \\ &= U\ R_i = U\ (E_i \times CE_i \times DU_i) \end{aligned} \qquad (8)$$

Where $i = 1, \ldots, r$.

Hence from equations (6) and (8) it can be noted that R can be represented as a matrix of membership functions. Therefore, the fuzzy rule base has been stored in a two dimensional array as shown in Fig 5. Consequent DU'_i for ith rule R_i can be accessed using the row index j for error and column index l for change in error of the rule base matrix. Hence, rule R_i, consequent DU'_i, error E_i, and change in error CE_i can be redenoted as $R_{j\,l}$, $DU_{j\,l}$, E_j and CE_l respectively. Hence equation (7) can be rewritten as,

$$DU'_{j,\,l} = (E'_j \times CE'_l)\ o\ R_{E_j} \to CE_l \to DU_{j,\,l} \qquad (9)$$

where $R_{j,\,l}$ is redenoted as, $R_{E_j} \to CE_l \to DU_{j,\,l}$

If sup-min operation is used for composition, the expression reduces as,

$$\mu_{DU'_{j,\,l}}(du) = \underset{e,\,ce}{Sup - min}\ (min\ \mu_{E'}(e_K), \mu_{CE'}(ce_k)), \mu_{R_{j,\,1}}(e, ce, du) \qquad (10)$$

In the implementations of FLC, as the measured values e_k and ce_k at kth sampling instant are known exactly, and are used to derive DU_{jl} for each fuzzy control rule. Hence the measured values E' and CE' are represented as nonfuzzy subsets, in which all elements have the membership value zero except at element e_k and ce_k, where the measurements occurred. Under such conditions equation 9 reduces to the following equation.[16]

$$\mu_{DU'_{j,\,l}}(du) = min\ [min\ \{\mu_{E_j}(e_k), \mu_{CE_1}(ce_k)\}, \mu_{DU'_{j,\,1}}(du)] \qquad (11)$$

The consequent for the complete set of rules is given by

$$\mu_{DU'_{j,\,l}} = max\,(\mu_{DU'_{j,\,l}}) \qquad (12)$$

The crisp control action for use in simulated plant is computed using center of area method as in equation (13).

$$du_k = \frac{\sum_{i=1}^{n} \mu_{DU'}(du_i)\, du_i}{\sum_{i=1}^{n} \mu_{DU'}(du_i)} \qquad (13)$$

Where n is number of quantization levels.

The simulation model is obtained after discretizing equation (1) into the form,

$$y\,(k) = d_0 \times y\,(k-1) + c_0 \times u\,(k-s) + c_1 \times u\,(k-s-1) (14)$$

where $y\,(k)$ and $u\,(k)$ are process output and control input at kth sampling instant, and d_O, c_O and c_1 are constants, depending on the sampling time, process gain and time delay, and s is the maximum possible integer period of sampling interval in t_d.

SIMULATION RESULTS

As discussed in introduction, FLC performs better than PID in most of the systems. Hence the performance of GA-FLC using the best of the different generation membership functions is compared with that of the traditional FLC and is shown in Fig 6, for the values of $K_E = 12$, $K_{CE} = 50$, $K_{DU} = 0.01$, and set point = 30cm. It is clear that GA-FLC performs better than FLC. Hence it is needless to compare its performance with PID.

Population size variation trials showed that populations greater than six and less than twelve were most effective. Smaller populations maintained less variations that converge prematurely. Selection of larger population size increases processing overhead without corresponding improvement in

FIGURE 6
RESPONSES OF FLC AND GA — FLC FOR FORTY NINE QUANTIZATION LEVELS

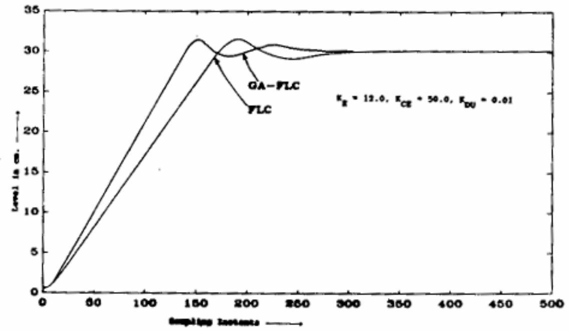


FIGURE 7
RESPONSES OF FLC AND GA — FLC FOR FORTY NINE QUANTIZATION LEVELS

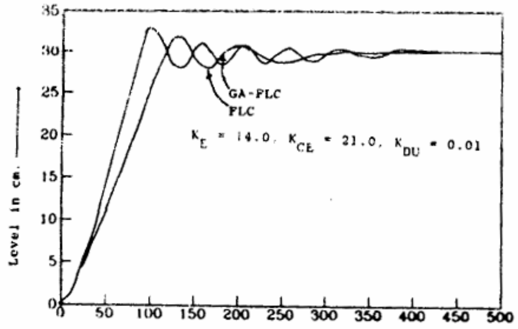


Fig 7 gives the responses of FLC and GA-FLC for 49 quantization levels, for $K_E = 14$, $K_{CE} = 21$ and $K_{DU} = 0.01$. It can be observed from Fig 6 and Fig 7 that even if the values of K_E, K_{CE} and K_{DU} are changed from 12, 50 and 0.01 to 14, 21 and 0.01, respectively, there are no appreciable changes in controller performances.

CONCLUSIONS

The approach presented in this paper is based on Darwinian evolution which rests on the idea of survival of the fittest or natural selection to produce a set of optimal MFs through repeated simulation. The implication of the results are summarized as, (a) the proposed approach significantly reduces the time and effort to find appropriate values for a large number of MFs (b) the proposed approach gives a clear understanding of the effect of MFs on the controller performance, and effect of plant parameter variations in terms of MFs.

Since a GA does not require any specific information, it is more flexible than any other numerical optimization technique. Furthermore, extension of this GA optimization technique to design or evaluation of rule base is currently under study. Also the authors are interested to develop hybrid systems of GA with neural networks for process control applications. The proposed GA- FLC may be used for water level control in hydro as well as thermal power plants. GA-FLC is a good candidate for efficient control of flow, temperature and level of a liquid in industrial control applications.

ACKNOWLEDGEMENT

The first author acknowledges with thanks, of the coordinator MAP laboratory, Department of Electrical Engineering, IIT Delhi, for extending the computational and laboratory facilities for carrying out this research work. Also the authors thank the reviewers for their careful reading and helpful comments.

REFERENCES

1. E H Mamdani. 'Application of Fuzzy Algorithm of a Simple Dynamic Plant.' *Proceeding of IEEE*, 1974, vol 121, p 1585.

2. E H Mamdani and S Assilan. 'A Fuzzy Logic Controller for a Dynamic Plant.' *International Journal Man-Machine Stud*, 1975, vol 7, p 1.

3. H R Van Nauta Lemke and W J Kickert. 'The Application of Fuzzy Set Theory to Control a Warm Water Plant.' *Journal AI*, 1976, vol A17, p 8.

4. W J M Kickert and H R Van Nauta Lemke. 'Application of a Fuzzy Controller in a Warm Water Plant.' *Automatica*, 1976, vol 12, p 301.

5. P J King and E H Mamdani. 'The Application of Fuzzy Control Systems to Industrial Processes.' *Automatica*, 1976, vol 13, p 235.

6. D Rutherford and G Z Carter. 'A Heuristic Adaptive Controller for a Cinter Plant.' *Proceedings of 2nd IFAC Symposium, Johannesburg*, 1976.

7. T J Procyk and E H Mamdani. 'A Linguistic Self Organizing Process Controller.' *Automatica*, 1979, p 15.

8. J J Ostergard. 'Fuzzy Logic Control of a Heat Exchange Process.' *Internal Report 7601*, Electrical Power Engineering Department Technical University of Denmark, 1976.

9. W J Kicket. 'Further Analysis and Application Fuzzy logic Control.' *Internal Report F/WK 2/75*, Queen Mary's College London, 1976.

10. C Karr. 'Genetic Algorithms for fuzzy controllers.' *AI Expert*, February 1991, p 26.

11. C L Karr and E J Gentry. 'Fuzzy Control of PH using Genetic Algorithm.' *IEEE Transaction on Fuzzy Systems*, vol 1, February 1993, p 46.

12. B Subudhi. 'A Genetic Algorithm Based Adaptive Fuzzy Logic Controller for Process Control Applications.' *M. Tech Thesis*, IIT Delhi, 1993.

13. Y F Li and C C Lau. 'Development of Fuzzy Algorithms for Servo Systems *IEEE Control Systems Magazine*, April 1989, p 7065-7114.

14. A B Corripio. 'Tuning of Industrial Control Systems.' *Instrument Society of America*, North California, 1990.

15. E H Mamdani. 'Application of Fuzzy Logic to Approximate Reasoning using Linguistic Synthesis.' *IEEE Transaction on Computers*, vol C-26, no 12, December 1977, p 1182.

16. N Ahmad. 'Real Time Intelligent Distributed Control Systems.' *Ph. D. Dissertation*, IIT Delhi, 1992.

APPENDIX

ALGORITHM FOR GA-FLC

```
Begin
  generation g = 0
  initialize p(g)
  evaluate FLC performances (ITSEs) in p(g) for each chromosome while
  (non termination-condition)
  begin
  g = g + 1
  select p(g) from p(g – 1)
  recombine p(g)
  evaluate p(g)
  end
end.
```