

EVOLUTIONARY TUNING OF FUZZY RULE BASE SYSTEMS FOR NONLINEAR SYSTEM MODELLING AND CONTROL

Pintu Chandra Shill[†], Bishnu Sarker[‡] and Kazuyuki Murase[†]

[†]Department of System Design Engineering, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan

[‡] Department of Computer Science and Engineering, KUET, Khulna 9203, Bangladesh
{pintu, murase}@synapse.his.u-fukui.ac.jp, bishnu_43@yahoo.com

ABSTRACT

Fuzzy systems generally works based on expert knowledge base. Fuzzy Expert knowledge base derived from the heuristic knowledge of experts or experience operators in the form of fuzzy control rules and membership functions (MFs). The major difficulties for designing a fuzzy models and controllers are identify the optimized fuzzy rules and their corresponding shape, type and distribution of MFs. Moreover, the numbers of fuzzy control rules increases exponentially with the number of input output variables related to the control system. For this reason it is very difficult and time consuming for an expert to identify the complete rule set and shape of MFs for a complex control system having large number of input and output variables. In this paper, we propose a method called evolutionary fuzzy system for tuning the parameters of fuzzy rules and adjust the shape of MFs through evolutionary algorithms in order to design a suitable and flexible fuzzy models and controller for complex systems. This paper also presents new flexible encoding method methods for evolutionary algorithms. In evolutionary fuzzy system, the evolutionary algorithms is adapted in two different ways Firstly, generating the optimal fuzzy rule sets including the number of rules inside it and secondly, selecting the optimum shape and distribution of MFs for the fuzzy control rules. In order to evaluate the validity and performance of the proposed approach we have designed a test strategy for the modeling and control of nonlinear systems. The simulation results show the effectiveness of our method and give better performance than existing fuzzy expert systems.

KEYWORDS

Fuzzy Expert System, Optimization, Evolutionary Algorithms (EAs), Evolutionary Fuzzy System and Nonlinear System.

1. INTRODUCTION

Fuzzy systems have been widely used in industrial control [1], data analysis and knowledge based systems [2]-[3]. Fuzzy rule base system use linguistic form of if-then rules to describe the system. The fuzzy rule base systems are effective to handle the class of nonlinear systems, vastly more difficult to analyze and model. In designing a fuzzy rule base system, the most important factors are selecting the appropriate fuzzy rules, the optimum number of rules, the shape and distribution of MFs and the scaling factors of input output variables. There has been a sustainable amount of research works done in the field of fuzzy systems. But most approaches use the expert knowledge base for defining the fuzzy rules as well as shape and distribution of MFs and scaling factors [4].

Existing rules can be replaced with new rules generated by evolutionary algorithms. There has been no systematic procedure for the design of nonlinear systems. The learning procedure is mainly based on an evolutionary strategy.

Identification of mathematical models of nonlinear systems is vital in many fields of engineering. The designed systems are used to study the behavior of prediction, process supervision, optimization and control purposes.

Recently, there has been sustainable research about adaptive control schemes for nonlinear systems via fuzzy-neural networks have been proposed [5] and [6]. Lin et al. has been proposed a linear matrix inequalities (LMI) method for achieving the output tracking control for nonlinear time-delay systems using Takagi-Sugeno (T-S) fuzzy model approach. Recently, artificial neural networks have been successfully applied to the control of nonlinear systems [7]-[8]. But a given set of input–output patterns (called the training set), the connection weights of the neural networks are adjusted in order to approximate the input–output patterns provided in the training set according to some predefined criterion.

In this paper, evolutionary algorithms based method is represented to evolve an evolutionary fuzzy system. It not only selects optimum antecedent and consequent parameters for rule set but also selects the optimum number of rules. In evolutionary fuzzy system, the evolutionary algorithms is adapted in two different ways Firstly, generating the optimal fuzzy rule sets including the number of rules inside it and secondly, selecting the optimum shape and distribution of MFs for the fuzzy control rules. In this way make the system more adaptive to solve the large number of complex system. Evolutionary fuzzy system used unique fitness function and various evolutionary operators to make control systems convergent. The simulation results show the effectiveness, accurate and scalability of our method and give better performance than existing fuzzy expert systems.

The rest of this paper is organized as follows: Section II describes the literature relevant to the fuzzy systems. Section III presents the system description and the problem under consideration to illustrate the effectiveness of the proposed design methods. The evolutionary based design process is presented in section IV. In section V we present the simulation results of algorithms using different configurations and section V concludes this paper.

2. FUZZY SYSTEMS

In 1960s, Professor Lotfi Zadeh, for the first time introduced the idea of fuzzy logic [9]. In 1960s, Professor Lotfi Zadeh, for the first time introduced the idea of fuzzy logic [9]. Fuzzy systems are a rule base system, constructed from human expert knowledge. Fuzzy logic controllers contain four main components as shown in fig. 1. They are fuzzifier, a rule/knowledge base, an inference engine and defuzzifier [27].

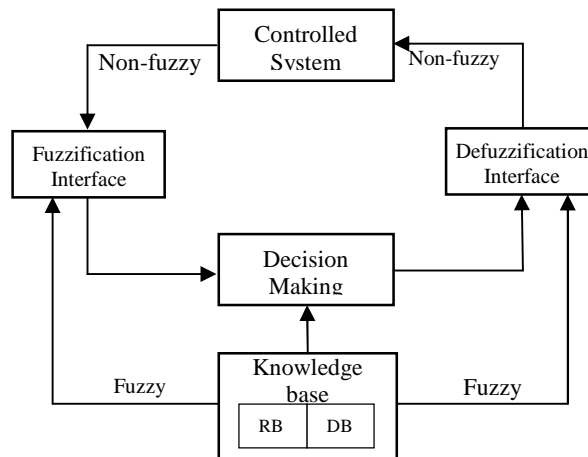


Figure 1. General Structure of Fuzzy Logic System (FLS)

Fuzzy inference engine is the central decision point of a fuzzy logic system (FLS). This part of a FLS harnesses fuzzy knowledge or rule base to map a fuzzy set in input space to a set in output space. To communicate with environment fuzzy inference engine is accompanied with fuzzification and defuzzification interface which are responsible for mapping real valued input to fuzzy input and fuzzy output to real valued output respectively. The brief description is as follows:

2.1. Fuzzifier

Membership function (μ_A) that associates with each element $x \in X$ where X represents universe of discourse, and the membership grade $\mu_A(x) \in [0,1]$ that represents the degree of association are two important considerations when defining fuzzy sets (Fig. 2). Fuzzifier maps the input value to fuzzified value that means the essential task of fuzzification is to transfer crisp input value $x \in X$ into a fuzzified value in $A \in U(\text{universe})$.

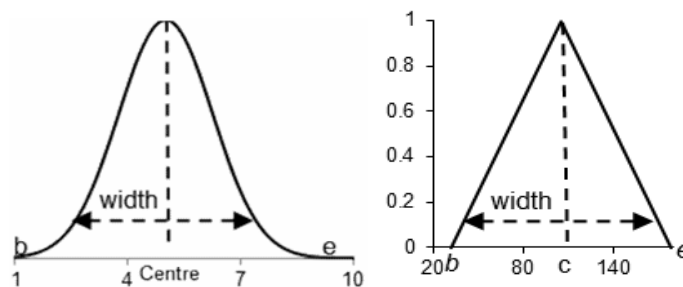


Figure 2. Fuzzy set (MFs: Gaussian and Triangular)

2.2. Fuzzy rule base

A fuzzy rule base consists of a collection of fuzzy IF-THEN rules. It is the core of fuzzy systems. Most of the FLCs use fuzzy rule base has the following form:

$$R^l : \text{IF } x_1 \text{ is } F_1^l, \dots, \text{ and } x_n \text{ is } F_n^l, \text{ THEN } y \text{ is } G^l.$$

$l = 1, 2, 3, \dots, M, M = \text{number of rules in the rule base}$

where x_1, x_2, \dots, x_n , and y represents the input and output linguistic variables respectively. F_i^l and G^l are stands for fuzzy sets in input sets $X \in X_1 \times X_2 \times \dots \times X_n$ and output sets Y . Each fuzzy IF-THEN rule can be divided into two parts: one is antecedent (or IF) and another one is consequent (THEN). Antecedent part consists of several preconditions and a consequent part defines the output action.

2.3. Fuzzy Inference Engine

Fuzzy inference engine makes a mapping from fuzzy sets to fuzzy sets. It combines the fuzzy rules in the rule base using the fuzzy logic principles. The definition of fuzzy relation R^l can take the following form:

$$R^l : X \times Y = \{(\vec{x}, y) : x \in X, y \in Y\}$$

Here \vec{x} denotes a vector of the form $(x_1, x_2, \dots, x_n)^T$. In fact the relation R^l defines the recipe to map fuzzy sets in X to fuzzy sets in Y . Fuzzy inference process can be defined as $F_1^l \times F_2^l \times \dots \times F_n^l \rightarrow G^l$. Membership Functions (MF), fuzzy logic operators, and if-then control rules are the integral part of the fuzzy inference process. Fuzzy inference process is tied together with membership functions, fuzzy operators such as AND or OR and fuzzy rule base for inferring consequent from antecedent.

2.3. Defuzzifier

Defuzzification is the process of determining the final crisp output values and defuzzifier implements defuzzification process. Defuzzification is achieved using the “Center of gravity” method. Defuzzification results in output value which is numerical for the fuzzy set. The defuzzification method is mainly the centroid defuzzification and it uses the fuzzy centroid θ as output:

$$\theta = \frac{\sum_{j=1}^p \theta_j m_o(\theta_j)}{\sum_{j=1}^p m_o(\theta_j)}$$

where O defines a fuzzy subset of the universe of discourse $\tau = \{\theta_1, \theta_2, \dots, \theta_p\}$, m is the respective MF.

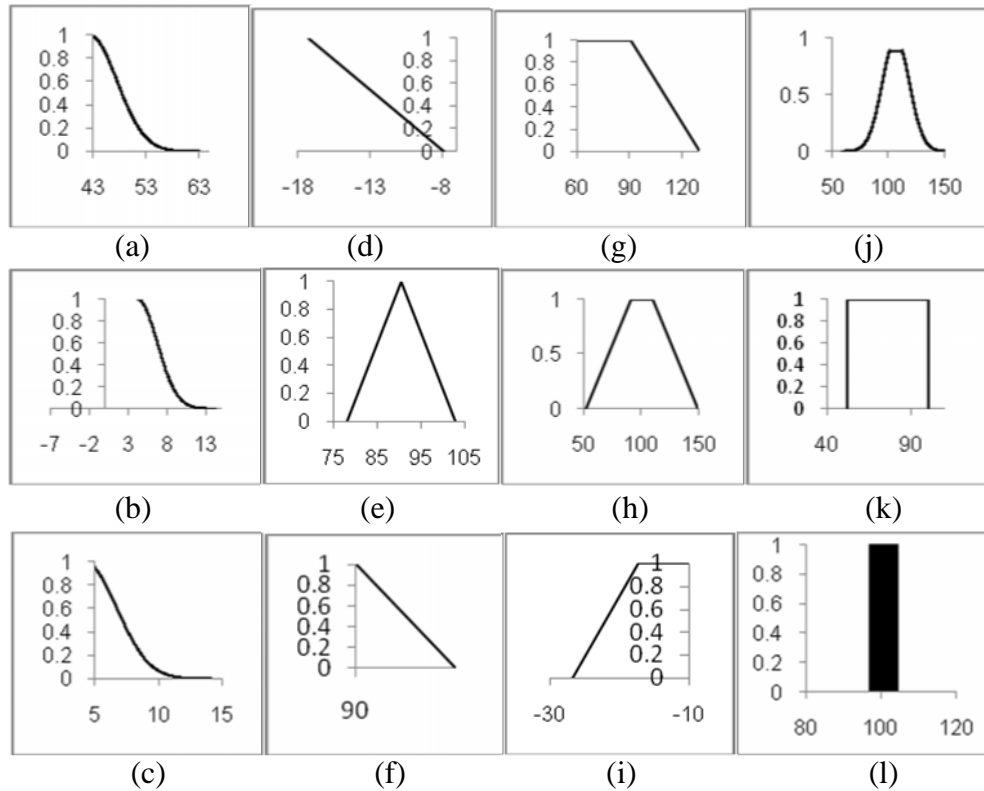


Figure 3. Membership Functions (MFs)

2.4. Membership functions(MFs)

One of the important steps of fuzzy theory applications is to generate appropriate MFs. A membership function (MFs) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1 as shown in Fig. 3.

The basic membership functions are could be piecewise linear functions, the Gaussian distribution function, the sigmoid curve and the quadratic and cubic polynomial curves. Also the membership function could be characterized to be linear and non-linear. A membership function is completely determined by three values: center (C), width (w) and the function type value (binary bit) in Fig.3. Most commonly used MFs are triangle, Gaussian, and sigmoid functions, the s-function, the z-function, - function, and bell shaped [4]. Fig. 3(a)-3(l) depicts the different type of MFs and mathematically described by equation (1)-(6).

The s- function:

$$s(x_1, x_r, x) = \begin{cases} 0, & \text{if } x < x_1 \\ \frac{1}{2} + \frac{1}{2} \cos \frac{x - x_r}{x_r - x_l}, & \text{if } x_1 \leq x \leq x_r \\ 0, & \text{if } x > x_r \end{cases} \tag{1}$$

The z- function:

$$z(x_1, x_r, x) = \begin{cases} 1, & \text{if } x < x_1 \\ \frac{1}{2} + \frac{1}{2} \cos \frac{x-x_r}{x_r-x_1}, & \text{if } x_1 \leq x \leq x_r \\ 0, & \text{if } x > x_r \end{cases} \quad (2)$$

- Function:

$$\Pi(x_1, x_2, x_3, x_4, x) = \min(s(x_1, x_2, x), z(x_3, x_4, x)) \quad (3)$$

Left_triangle membership function:

$$f_{Left_triangulae} = \begin{cases} 1, & \text{if } x < x_1 \\ \frac{x_2 - x}{x_2 - x_1}, & \text{if } x_1 \leq x \leq x_2 \\ 0, & \text{if } x > x_2 \end{cases} \quad (4)$$

Right_triangle membership function:

$$f_{Right_triangulae} = \begin{cases} 1, & \text{if } x < x_1 \\ \frac{x - x_1}{x_2 - x_1}, & \text{if } x_1 \leq x \leq x_2 \\ 0, & \text{if } x > x_2 \end{cases} \quad (5)$$

Triangle membership function:

$$f_{triangulae}(x) = \begin{cases} 0, & \text{if } x < x_1 \\ 2 \frac{x - x_1}{x_2 - x_1}, & \text{if } x_1 \leq x \leq \frac{x_2 + x_1}{2} \\ 2 \frac{x_2 - x}{x_2 - x_1}, & \text{if } \frac{x_2 + x_1}{2} < x \leq x_2 \\ 1, & \text{if } x > x_2 \end{cases} \quad (6)$$

Gaussian membership function:

$$f_{Gaussian}(x) = e^{-0.5y^2} \quad \text{where } y = \frac{8(x - x_1)}{x_2 - x_1} - 4 \quad (7)$$

3. PROBLEM DOMAIN

Identifying optimized fuzzy rule base is one of the major difficulties that arise when designing fuzzy models and controllers. Traditionally, optimized fuzzy rule base is achieved from the expert knowledge through rigorous trial and error process. An evolutionary technique is offered in this paper to design optimal fuzzy rule base and their corresponding MFs for modeling and control of

nonlinear systems. Evolutionary programming is well in evolving the structure and the parameter of fuzzy rule base simultaneously for a given task. Here we discuss about some nonlinear problems such as

Problem -1:

This example consider a nonlinear system having equation of

$$y = (1 + x_l^{-2} + x_w^{-1.5}), \quad 1 < x_1, x_2 < 5 \tag{8}$$

where, x_l and x_w are input variables and y is the output variable. The solution of the nonlinear system will occur when $1 < x_1$ and $x_2 < 5$

Problem -2:

Another problem is Optimum Fuel Allocation in Power Plants. This could be characterized as an optimization problem which has an objective of minimizing the allowance of the fuel oil (FO) to the power plant. Figure 4 depicts a power plant running two-boiler-turbine-generator desired to produce an output of 50 MW. Fuel oil or blast furnace gas (BFG) can be used in isolation or in any combination. The maximum available BFG is predetermined. By applying nonlinear curve-fitting Hovanessian & Stout obtained the fuel requirements for the two generators explicitly in terms of MW produced. For generator-1 the fuel requirements for fuel oil in tons per hour is given by the equation (9):

$$f_1 = 1.4609 + 0.15186 x_1 + 0.00145 x_1^2 \tag{9}$$

and for BFG in fuel units per hour

$$f_2 = 1.5742 + 0.1631 x_1 + 0.001358 x_1^2 \tag{10}$$

where x_1 is the output in MW of generator-1.

Similarly for generator-2 the fuel oil requirement is:

$$g_1 = 0.8008 + 0.2031 x_2 + 0.000916 x_2^2 \tag{11}$$

and for BFG,

$$g_2 = 0.7266 + 0.2256 x_2 + 0.000778 x_2^2 \tag{12}$$

Where x_2 is the output in MW of generator-2. Assumptions: i) Only 10.0 units of BFG are available per hour. ii) Each generator have preference to pick any combination of fuel oil or BFG iii) additive effects are found when fuel oil and BFG used in combination, for example, when

using fuel oil and BFG in 1/3 ratio in generator-1 to produce x_1 MW, then the total fuel consumption consists of $0.25 f_1$ tons of fuel oil per hour and $0.75 f_2$ fuel units of BFG per hour.

The problem can be defined as producing 50 MW from the two generators making the fuel oil consumption a minimum. Mathematically, the problem can be formulated as follows: Minimize

$$C = x_3 f_1 + x_4 g_1 \tag{13}$$

Where f_1 and g_1 are given by equations (9) and (11) respectively.

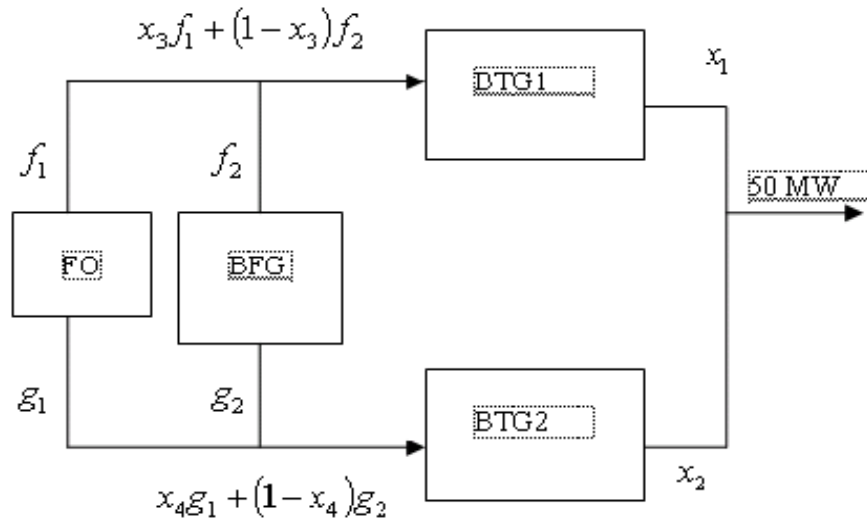


Figure 4. Two-boiler-turbine-generator Power plant

Subject to

- (a) Operating range for the generator-1: $18 \leq x_1 \leq 30$
- (b) Requirement of 50 MW of Power: $x_2 = 50 - x_1$
- (c) Operating range of generator-2 : $14 < x_2 < 25$
- (d) Fraction of fuel oil used in generator-1: $0 < x_3 < 1$
- (e) Fraction of fuel oil used in generator-2: $0 < x_4 < 1$
- (f) Availability of blast furnace gas (BFG):
 $BFG = (1 - x_3) f_2 + (1 - x_4) g_2 \leq 10.0$

where f_2 and g_2 are given by Equation (10) and (12) respectively. Hence the problem is to choose the variables x_1 , x_3 , and x_4 so that C as given by equation (13) is minimized because the variable x_2 is eliminated by using equation $x_2 = 50 - x_1$. There are five inequality constraints embodied in equation $18 \leq x_1 \leq 30$ and equation $14 < x_2 < 25$, $0 < x_3 < 1$, $0 < x_4 < 1$ and $BFG = (1 - x_3) f_2 + (1 - x_4) g_2 \leq 10.0$. Also note that there is no lower limit restriction on equation $BFG = (1 - x_3) f_2 + (1 - x_4) g_2 \leq 10.0$ since computationally BFG cannot become negative.

4. EVOLUTIONARY DESIGN PROCESS

We have developed and deployed a robust method for automating the tuning and maintenance of fuzzy systems called evolutionary fuzzy system-inspired from natural evolution is widely used for optimization and learning process.

4.1. Representation

When designing a fuzzy model using an evolutionary algorithm, one of the most important considerations is the representation scheme, that is, how to encode the fuzzy system into the chromosome.

Structure Representation: The connection matrix representing the fuzzy rule base structure will be of two-dimensional (2-D) matrix form, The size of the matrix is determined by the number of rules, the number of input variables. The number of rules is also a parameter representing structure to be found during the evolutionary process. The element m_{ij} in the connection matrix has positive real value and indicates the relative importance of the j^{th} input fuzzy variable in the i^{th} rule.

α_1	α_2	
0	1.23	No of rules
0.78	0.56	
No of inputs		

Figure 5. Connection matrix representing the structure of a fuzzy rule base

Assume that there is a rule base represented by the connection matrix with two rules and two input variables x_1 and x_2 , as shown in Fig. 5. $m_{11} = 0$ means that there is no fuzzy variable of x_1 of the first rule.

Parameter Representation: The parameter matrix (Fig. 6) that contains the parameters for defining the membership functions within the fuzzy rule base defined by the connection matrix will be of 2-D matrix form. The size of the matrix is determined by the number of rules, the number of input variables x the parameter number of input fuzzy variable + the number of output variables x the parameter number of output fuzzy variable. In case the input fuzzy variable is of the trapezoidal form as in, the parameter number of input fuzzy variable is equal to four. The parameter number of output fuzzy variable is equal to one. The elements p_{ij} 's in the parameter matrix have real values.

P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	No of rules
P_{21}	P_{22}	P_{23}	P_{24}	P_{25}	P_{26}	P_{27}	P_{28}	P_{29}	
Input x_1				Input x_2				Output	

Figure 6. Parameter matrix representing the membership functions of a fuzzy rule base.

The concept of fuzzy sets and linguistic variables are tied together in expressing fuzzy knowledge. They are defined as membership functions that provide membership values for each element of their universe of discourses. The meaning of linguistic terms for the verbal

characterization of the input and output variables is fixed by appropriately defined membership functions. Although there exists no restrictions on the form of membership functions, we use piecewise linear function like trapezoids or triangles in the premise and fuzzy singletons in the consequent. The rule base consists of rules expressed by

$$R^i : \text{if } x_1 \text{ is } A_k^i(x_1) \text{ with } \alpha_{x_1} \text{ and } \dots \dots x_k \text{ is } A_k^i(x_k) \text{ with } \alpha_{x_k} \text{ then } y_1 \text{ is } B_1^i \text{ } y_q \text{ is } B_q^i \tag{14}$$

Where R^i are i^{th} rule $1 \leq i \leq r$, $x_j (1 \leq j \leq k)$ input variable, and $y_j (1 \leq p \leq q)$ output variable. $A_j^i(x_j)$ is fuzzy variable and B_i takes fuzzy singleton i.e. real number. α_{xij} (Parameters of parameter matrix) denotes a numerical positive value characterizing the relative importance of the j^{th} fuzzy variable in the i^{th} rule.

Assuming that there is a rule base with connection matrix as in Fig. 5 and parameter matrix as in Fig. 6, we can formulate a fuzzy rule base as shown in Fig. 7.

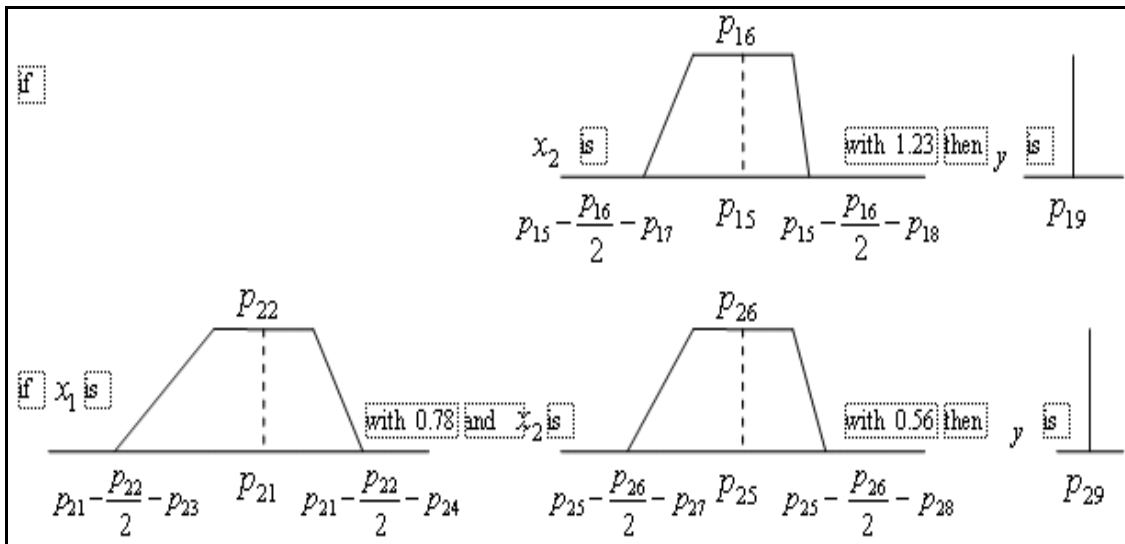


Figure 7. Fuzzy rule base represented by connection and parameter matrices.

4.2. Evaluation

In the design of an optimal fuzzy model, the first task is to minimize error of

$$E = \frac{1}{n} \left[\frac{1}{q} \sum_{p=1}^q \sum_{l=1}^n (y_{lp} - y_{lp}^*)^2 \right] \tag{15}$$

where n is the number of I/O data, y_{lp} target output, and y_{lp}^* output inferred from a fuzzy rule base. Penalty function is

$$p = \sum_{i=1}^n c_i \tag{16}$$

Where as a penalty factor, c_i is a constant value for any i ($i = 1, 2, \dots, n$), n is the number of I/O data. The fitness value of fuzzy rule base is calculated as in the following:

$$F = \frac{1}{E + P} \tag{17}$$

Maximizing F is to minimize error of (15) for modeling and of (16) for control and to force to solutions toward feasible sets by penalizing the occurrence of null sets in the universe of discourse. This fitness function provides a means for evaluating the performance of fuzzy model with the selected fuzzy rule base in the process of evolution, so that an optimized fuzzy model or fuzzy controller would be developed by the best individual.

4.3. Evolutionary Operators

4.3.1. Reproduction

At the very beginning, applying some ranking method each individual is ranked according to their fitness values. After that the upper 30% of the population is used to generate 50% of the new population.

4.3.2. Crossover

Crossover is performed using SBMAC to generate the offspring population. For each subpopulation, μ/l offspring are generated. Thus, μ numbers of offspring's are generated for the l subpopulations at the generation t .

Table 1. Connection matrix of Problem-1 & Problem-2

Rule No.	Problem-1		Problem-2	
	α_1	α_2	α_1	α_2
1	0.57	0.37	5.56	0.41
2	0.51	0.54	1.06	5.56
3	0.38	0.45	4.86	1.06
4	0.45	0.44	9.55	4.86
5	0.43	0.45	0.70	9.55
6	0.48	0.31	-3.95	0.70
7	0.56	0.69	341.10	-3.95
8	0.58	0.41	9.61	341.10
9	0.41	0.63	19.82	9.61
10	0.37	0.57	5.47	19.82
11	0.49	0.38	3.52	5.47
12	0.37	0.41	0.32	3.52
13	0.70	0.59	5.56	0.32
14	0.50	0.54	1.06	5.56
15	0.37	0.50	7.08	1.06
16	0.41	0.50	9.55	7.08
17	0.39	0.41	0.03	9.55
18	0.58	0.58	-7.27	0.03
19	0	0	1.46	-7.27
20	0	0	9.61	1.46

4.3.3. Mutation

All of the variables of an offspring are mutated using time variant mutation (TVM) operator in the mutation phase. Thus the offspring population undergoes this mutation scheme.

4.3.4. Evolutionary Procedure

The framework of evolutionary procedure used in this paper is as follows:

Step-1:The algorithm begins by generating an initial population $P(0)=[S_1(0)S_2(0).....S_{N/P}(0)]$ at random and set $i=0$. A suitable representation also decided upon for the chromosome that is the potential solution of a problem.

Table 2. Parameter matrix for Input $x_1 (P_1, P_2, P_3, P_4)$ and Input $x_2 (P_5, P_6, P_7, P_8)$ and Output P_9 of Problem -1.

Rule No.	Input x_1				Input x_2				Output P_9
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	
1	3.15	8.86	6.48	6.40	3.28	10.77	5.28	5.05	1.17
2	3.73	12.52	6.47	6.36	3.94	11.56	3.71	4.84	-0.46
3	3.46	12.60	5.07	7.13	4.11	19.99	5.16	5.29	-2.39
4	3.58	12.79	3.96	7.27	4.10	13.78	4.99	5.78	0.058
5	3.25	12.84	4.58	7.25	3.15	14.26	6.02	6.12	-1.81
6	4.18	13.74	6.59	7.02	3.41	14.12	5.65	7.07	2.14
7	4.42	14.77	6.23	6.74	3.75	14.78	5.32	7.15	-2.38
8	4.23	14.88	6.72	6.45	3.50	15.56	5.46	7.16	0.23
9	4.52	15.80	5.82	6.56	4.66	13.83	6.56	7.21	-1.85
10	5.41	15.73	5.60	6.70	2.47	15.44	5.83	7.54	1.74
11	2.97	13.13	5.02	6.69	3.86	14.44	6.21	7.55	-3.17
12	2.87	12.97	4.32	6.75	1.95	12.08	6.18	7.27	-0.30
13	3.00	11.72	4.36	7.16	1.90	12.71	6.52	7.26	-0.74
14	3.00	12.17	5.30	7.20	1.62	11.86	5.08	7.14	-0.49
15	2.19	13.84	5.10	7.69	1.80	12.34	4.82	6.58	0.66
16	2.13	13.97	5.15	7.77	1.68	12.97	4.74	7.13	1.62
17	2.39	13.41	4.63	7.73	1.05	12.54	4.57	7.00	3.26
18	3.78	13.84	3.68	7.88	0.74	12.55	4.17	7.03	0.077
19	2.32	14.38	51.58	7.78	0.54	12.50	4.09	7.00	0.03
20	3.18	12.87	3.97	7.90	0.42	12.98	4.15	7.12	0.04

Table 3. Parameter matrix for Input x_1 (P_1, P_2, P_3, P_4) and Input x_2 (P_5, P_6, P_7, P_8) and Output P_9 of Problem -2

Rule	Input x_1				Input x_2				Output P_9
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	
1	9.61	19.82	0.09	3.52	1.06	4.86	9.55	-3.95	0.70
2	19.82	0.09	3.52	0.41	4.86	9.59	0.70	341.10	-3.95
3	0.09	3.52	0.41	5.56	9.55	0.70	-3.95	9.61	341.10
4	3.52	0.41	5.56	1.06	0.70	-3.95	341.10	19.82	9.61
5	0.41	5.56	1.06	4.86	-3.95	341.10	9.61	5.47	19.82
6	5.56	1.06	4.86	9.55	341.10	9.61	19.82	3.52	5.47
7	1.06	4.86	9.55	0.70	9.61	19.82	5.47	0.32	3.52
8	4.86	9.55	0.70	-3.95	19.82	5.47	3.52	5.56	0.32
9	9.55	0.70	-3.95	341.10	5.47	3.52	0.32	1.06	5.56
10	0.70	-3.95	341.10	9.61	3.52	0.32	5.56	7.08	1.06
11	-3.95	341.10	9.61	19.82	0.32	5.56	1.06	9.55	7.08
12	341.10	9.61	19.82	5.47	5.56	1.06	7.08	0.03	9.55
13	9.61	19.82	5.47	3.52	1.06	7.08	9.55	-7.27	0.03
14	19.82	5.47	3.52	0.32	7.08	9.55	0.03	1.46	-7.27
15	5.47	3.52	3.26	5.56	9.55	0.03	-7.27	9.61	1.46
16	3.52	0.32	5.56	1.06	0.03	-7.27	1.46	19.82	9.61
17	0.32	5.56	1.06	7.08	-7.27	1.46	9.61	1.64	19.82
18	5.56	1.06	7.08	9.55	1.46	9.61	19.82	3.52	1.64
19	1.06	7.08	9.55	0.03	9.61	19.82	1.64	0.64	3.52
20	7.08	9.55	0.03	-7.27	19.82	1.64	3.52	5.56	0.64

Step-2: A fuzzy model is constructed from the parameter values of each individual and the consequent parameters are also calculated for all individual in this step.

Step-3: Evaluate every individual and the evaluation function returns a fitness value according to the solving performance of the problem.

Step-4: Apply evolutionary operators to obtain the next (population P_{i+1}). This paper presents an insight to provide the enhanced search capability through crossover and mutation operator applied together. Cross over and mutation operator altogether capable of enhancing the rate convergence.

Step-5: Set $i = i + 1$. Return to step 2) if the G_{max} is not reached or the procedure is terminated. If the procedure is terminated we get the best fitted chromosome is the best solution.

5. SIMULATION AND RESULTS

The implementation of the evolutionary fuzzy system is written in c++ and compiled using the Borland c++ 4.5 compilers. The work was developed fuzzy system for modeling a nonlinear system. We have assumed for this simulation the maximum acceptable number of fuzzy rules is 20 (One rule 11 parameters i.e. input x_1, x_2 each has four parameters p_1, p_2, p_3, p_4 and p_5, p_6, p_7, p_8 respectively and each has one importance m_j and output y has one parameter p_9). So the maximum number of parameter will be 220.

Connection Matrix: The connection matrix (Parameter of fuzzy rule) of problem-1 and problem-2 is shown in Table-1.

Parameter Matrix: The parameter matrix (Parameters of fuzzy rule) of problem-1 and problem-2 is shown in Table-2 and Table-3.

5.1. MFs Representation of Fuzzy Rule

We translate the connection and parameter matrix form of fuzzy rules into the membership form of fuzzy rules from the Table-1, Table-2, and Table-3. Fig. 6 (Problem-1) and Fig. 7 (Problem-2) show some of the different types of membership function form of fuzzy rules.

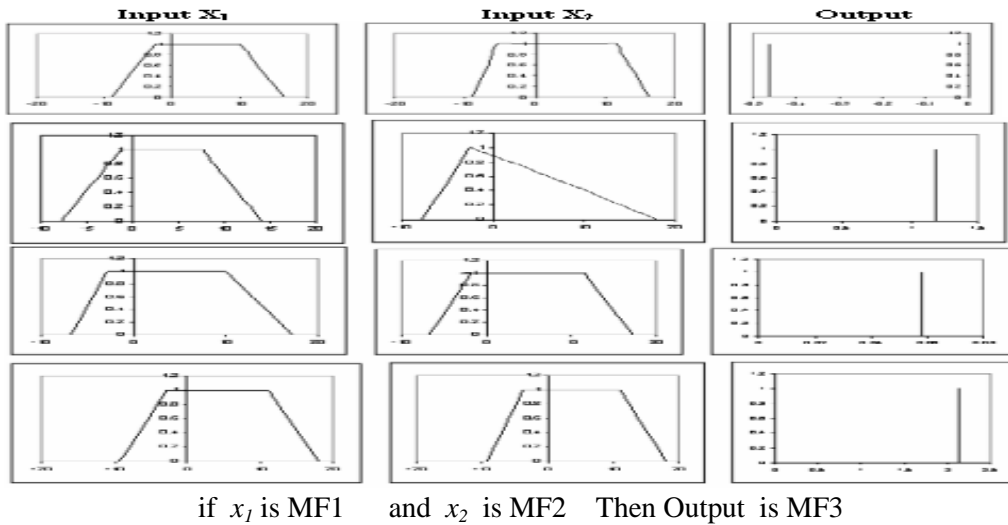


Figure 6. Fuzzy Rule Represented by Membership functions of Problem1

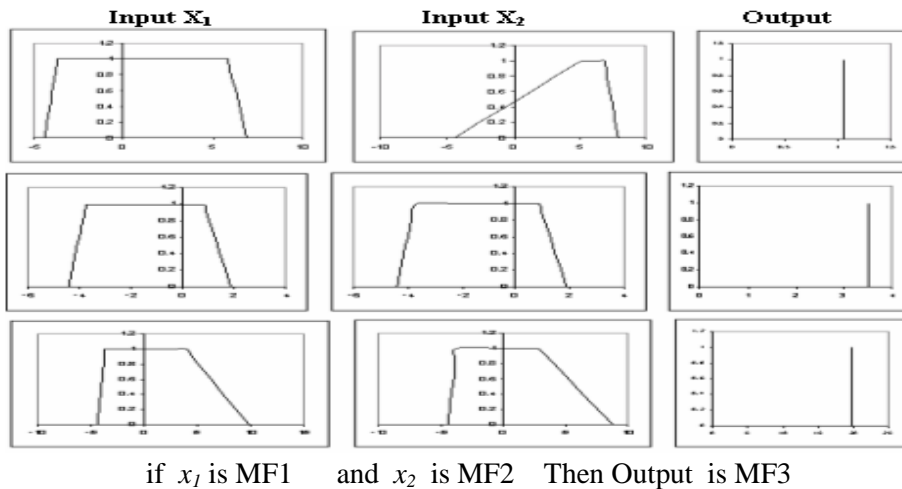


Figure 7. Fuzzy Rule Represented by Membership functions of Problem2

5.2. Evolution Histogram

During each generation the fitness of an individual is evaluated. The evolution histogram of generation versus error shows that as the number of generation increases the error decreases in

Fig. 8 and Fig. 9. These Figures also shows that after a certain generation the fitness value is fixed, the evolution tends to be convergent at an early stage.

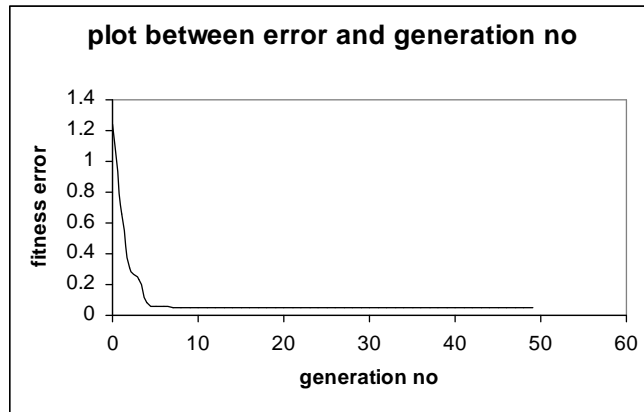


Figure 8. Evolution histogram of problem-1

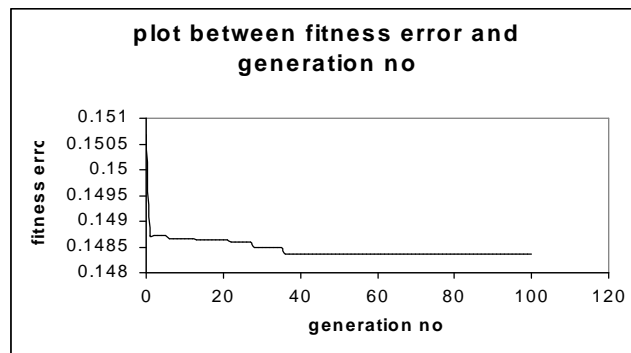


Figure 9. Evolution histogram of problem-2

5.3. Comparative analysis

The simulation study is presented to illustrate the validity of the proposed optimization method and compare the performance with the traditional fuzzy logic controller.

Problem1: Connection matrix as shown in table-2 which was affect the performance of the system.

Traditional Fuzzy System:

Table 4. Connection matrix (Parameters of fuzzy rules) of Problem 1

Rule No.	α_1	α_2
1	0.6	0.1
2	0.83	0.17
3	0.85	0.39
.	.	.
34	0.51	0.93
35	0.42	0.79

Connection Matrix: The connection matrix of a optimized fuzzy model for non-linear system in a traditional fuzzy system as shown in table-4[2].

Parameter Matrix: The parameter matrix of an optimized fuzzy model for non-linear system in a traditional fuzzy system as shown in table-5 [4].

Table 5. Parameter matrix (Parameters of fuzzy rules) of Problem1

Rule No.	Input X1				Input X2				Output Y
	$P_{r,x1}$	$P_{r,x2}$	$\frac{P_{r,x1} - P_{r,x2}}{2} - P_{r,x3}$	$\frac{P_{r,x1} + P_{r,x2}}{2} - P_{r,x4}$	$P_{r,x5}$	$P_{r,x6}$	$\frac{P_{r,x5} - P_{r,x6}}{2} - P_{r,x7}$	$\frac{P_{r,x5} + P_{r,x6}}{2} + P_{r,x8}$	
1	10 3	4.1 5	-3	585	- 0.3	3.6	-5.8	2	4.1 8
2	3.8 5	3.8 9	0.55	7.11	4.2 2	3.6 8	-4.2	6.06	0.7 9
3	4.7 2	3.5 5	-0	7.65	7.1 8	0	1.2	7.39	-3.0
.
.
34	0.0 3	0	-1.8	2.69	- 0.1	3.2 9	-4.4	6.49	4.2 9
35	2.0 1	1.2 8	1.22	2.65	3.2 5	0.9 7	0.52	7.02	0.4 4

We have compared the result of the above optimized fuzzy model in a traditional fuzzy system, our proposed evolutionary fuzzy model, and the original output value.

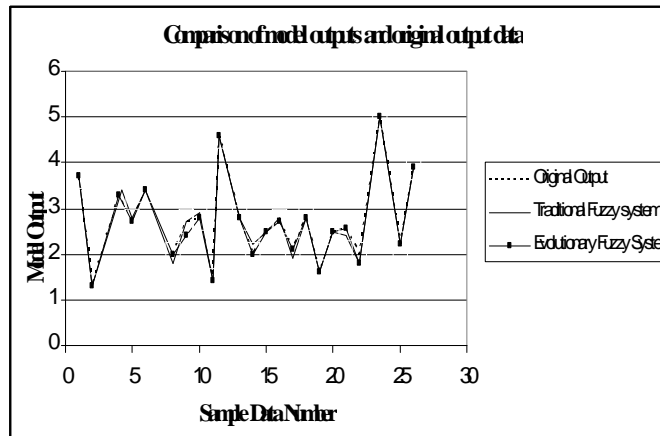


Figure 10. Comparison of model outputs and original output data

Table 4 and 5 shows the structure of fuzzy rules of the traditional fuzzy controller which has 35 fuzzy if-then rules. So the difference of rules between the traditional and evolutionary-fuzzy approach is $(35-19) = 16$ for problem-1 and $(35-20) = 15$ for problem-2. So, in evolutionary-fuzzy approach, the number of fuzzy rules is decreases with respect to the genetic- fuzzy approach.

Problem2:

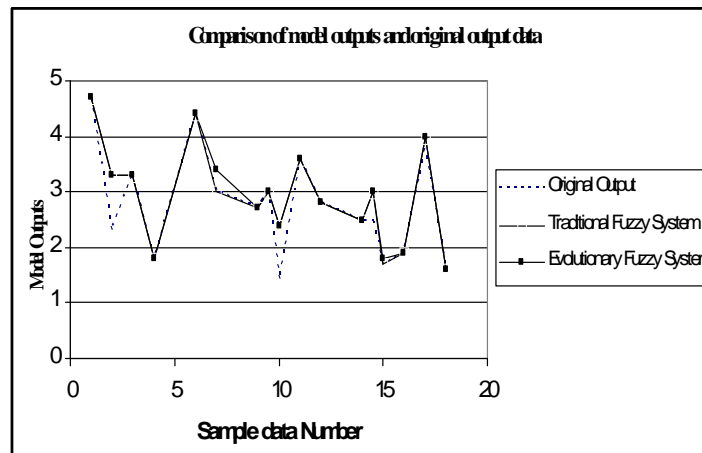


Figure 11. Comparison of model outputs and original output data

6. CONCLUSIONS

It is not possible to solve every nonlinear problem using analytical methods. Analytical methods are only able to deal with completely known systems, but they could only use an approximation of the given model. It appears frequently in nature that a priori knowledge is missing for the system. In this situation, the overall goal of this paper is to develop and analyze a new method for the optimization of fuzzy logic system for the control of nonlinear systems. The optimization method has been represented in which the membership function, shapes, types and the rule set, are evolved by an evolutionary algorithm. In addition, the parameters of the fuzzy system are adapted by using evolutionary algorithms. There are some successful results, but longer genetic search with more initial conditions and a longer control loop will certainly produce a superior result. Finally the obtained results show that better models can be obtained when using our proposed optimization method.

REFERENCES

- [1] Li-Xin Wang, "Stable and Optimal Fuzzy Control of Linear Systems," IEEE Transaction on Fuzzy systems, vol. 6, no. 1, pp. 137-143, Feb. 1998.
- [2] Y. Zhao, J. Lam, and H. Gao, "Fault Detection for Fuzzy Systems With Intermittent Measurements," IEEE Trans. Fuzzy Syst., vol. 17, no. 2, pp. 398-410, 2009.
- [3] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," IEEE Trans. Fuzzy Syst., vol. 14, no. 5, pp. 676-697, Oct. 2006.
- [4] Yuhui Shi, Russell Eberhart and Yaobin Chen, "Implementation of Evolutionary Fuzzy Systems" , IEEE Transaction on Evolutionary Computation, Vol. 7, No.2, 1999.
- [5] C. Lin, Q.-G. Wang and T.H. Lee, "H output tracking control for nonlinear systems via T-S fuzzy model approach," IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 36 , no. 2, pp. 450-457, 2006.
- [6] H.O. Wang, K. Tanaka and M.F. Griffin "An approach to fuzzy control of nonlinear systems: stability and design issues," IEEE Transactions on Fuzzy Systems, vol. 4, pp. 14-23, 1996.
- [7] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", IEEE Trans. Neural Networks, vol. 1, pp. 4-27, 1990.

- [8] K.J. Hunt, H.D. Sbarbaro, R. Zbikowski, and P.J. Gawthrop, "Neural networks for control systems—a survey," *Automatica*, vol. 28, pp. 1083–1112, 1992.
- [9] L.A. Zadeh, "Fuzzy sets," *Information and Control*, Vol. 8, No. 3, pp. 338–353, 1965.
- [10] Norberto Eiji Nawa and Takeshi Furuhashi, "Fuzzy System parameters Discovery By Bacterial Evolutionary algorithm," *IEEE Transaction On Fuzzy Systems*, Vol-7, No.5
- [11] Bing Chen and Xiaoping Liu., "Fuzzy guaranteed cost control for nonlinear systems with time-varying delay," *IEEE Transactions of Fuzzy Systems*, Volume 13, April, 2005.
- [12] Kang in-Jun, Woo Chun-Hee, Hwang Hee-Soo, and Woo Kwang B., "Evolutionary Design of Fuzzy Rule Base for Nonlinear System modeling and control," *IEEE Transaction On Fuzzy Systems*, Vol-8, No.1 February,2000.
- [13] W.R. Hwang and W.E Thompson, "Design of intelligent fuzzy logic controllers using genetic algorithms" In Proc. IEEE Int. Conf. Fuzzy Syst., Orlando, FL, pp. 1383-1388, 1994.
- [14] Dongwon Kim and Gwi-Tae Park, "A hybrid fuzzy model in nonlinear system modeling," *IEEE Transactions of Fuzzy System*, volume.17, Number-5 pp. 417-430, 2006.
- [15] Yonggon Lee and Stanislaw H. Zak, "Designing of Genetic Neural Fuzzy Antilock-Brake System Controller," *IEEE Transaction on Evolutionary Computation*, Vol-6, No.2, 2002.
- [16] Marco Russo, "Genetic Fuzzy Learning," *IEEE Transaction on Evolutionary Computation*, Vol-4, No.3, 2000.
- [17] F. Zheng, Q.-G. Wang and T.H. Lee, "Output tracking control of MIMO fuzzy nonlinear systems using variable structure control approach," *IEEE Transactions on Fuzzy Systems*, vol. 10, pp. 686–697, 2002.
- [18] A. Muhammad, "Fuzzy and evolutionary modelling of nonlinear control systems," *Mathematical and Computer Modelling*, Volume 33, Issues 4–5, pp. 533–551, February–March 2001.
- [19] Nohe R. Cazarez-Castro, Luis T. Aguilar, and Oscar Castillo, "Fuzzy logic control with genetic membership function parameters optimization for the output regulation of a servomechanism with nonlinear backlash," *Expert Systems with Applications*, Volume 37, Issue 6, pp. 4368–4378, June 2010.
- [20] M.S. Osman, Mahmoud A. Abo-Sinna, and A.A. Mousa, "A combined genetic algorithm-fuzzy logic controller (GA–FLC) in nonlinear programming", *Applied Mathematics and Computation*, Volume 170, Issue 2, pp. 821–840, November 2005.
- [21] Amir Hossein Gandomi, Amir Hossein Alavi, "Multi-stage genetic programming: A new strategy to nonlinear system modeling," *Information Sciences*, Volume 181, Issue 23, pp. 5227–5239, December 2011.
- [22] Haiping Du, and Nong Zhang, "Application of evolving Takagi–Sugeno fuzzy model to nonlinear system identification," *Applied Soft Computing*, Volume 8, Issue 1, Pages 676–686, January 2008.
- [23] Rajni Jain, N. Sivakumaran, and T.K. Radhakrishnan, "Design of self tuning fuzzy controllers for nonlinear systems," *Expert Systems with Applications*, Volume 38, Issue 4, pp. 4466–4476, April 2011.
- [24] Cheng-Jian Lin, Yong-Ji Xu, "A novel genetic reinforcement learning for nonlinear fuzzy control problems," *Neurocomputing*, Volume 69, Issues 16–18, pp. 2078–2089, October 2006.
- [25] Aydogan Savran, "An adaptive recurrent fuzzy system for nonlinear identification," *Applied Soft Computing*, Volume 7, Issue 2, pp. 593–600, March 2007.
- [26] Aytekin Bagis, "Fuzzy rule base design using tabu search algorithm for nonlinear system modeling," *ISA Transactions*, Volume 47, Issue 1, , Pages 32–44, January 2008.
- [27] How Sing Sii, Tom Ruxton and Jin Wang "A fuzzy Logic Based Approach to Qualitative Safety Modeling for Marine System", *Reliability Engineering and Systems Safety* 73(2001) pp. 19-34, 2001.

Authors

Pintu Chandra Shill received the B.Sc. degree in Computer Science Engineering (CSE) from Khulna University of Engineering and Technology (KUET), Bangladesh in 2003 and M.Sc. degree in Computer Engineering from Politecnico di Milano, Italy in 2008. He joined as a lecturer at the Department of CSE, KUET in 2004 and currently he is serving as an Assistant Professor. He has published several research papers in some reputed Journal and Conference. His research interest includes evolutionary computation, fuzzy logic and artificial neural networks.



Bishnu Sarker received the B.Sc. degree in Computer Science Engineering (CSE) from Khulna University of Engineering and Technology (KUET), Bangladesh in 2011. He joined as a lecturer at the Department of CSE, KUET in 2011 and currently he is serving as a Lecturer. His research interest includes Evolutionary Computation, Fuzzy Logic, Artificial Neural Networks, Distributed Computing, and Fault Tolerant Computing.



Kazuyuki Murase is a Professor at the Department of Human and Artificial Intelligence Systems, Graduate School of Engineering, University of Fukui, Fukui, Japan, since 1999. He received ME in Electrical Engineering from Nagoya University in 1978, PhD in Biomedical Engineering from Iowa State University in 1983. He Joined as a Research Associate at Department of Information Science of Toyohashi University of Technology in 1984, as an Associate Professor at the Department of Information Science of Fukui University in 1988, and became the professor in 1992. He is a member of The Institute of Electronics, Information and Communication Engineers (IEICE), The Japanese Society for Medical and Biological Engineering (JSMBE), The Japan Neuroscience Society (JSN), The International Neural Network Society (INNS), and The Society for Neuroscience (SFN). He serves as a Board of Directors in Japan Neural Network Society (JNNS), a Councilor of Physiological Society of Japan (PSJ) and a Councilor of Japanese Association for the Study of Pain (JASP).

