GUBJSE: ISSN: 2409-0476

ADAPTIVE INTERVAL TYPE-2 FUZZY LOGIC CONTROLLER FOR AUTONOMOUS MOBILE ROBOT

Pintu Chandra Shill [1], M. A. H. Akhand, Md. Saidul Islam, and M. M. Hafizur Rahman

Abstract**—**A Type-2 Fuzzy logic controller adapted with genetic algorithm, called type-2 genetic fuzzy logic controller (T2GFLC), is presented in this paper to handle uncertainty with dynamic optimal learning. Genetic algorithm is employed to simultaneous design of type-2 membership functions and rule sets for type-2 fuzzy logic controllers. Traditional fuzzy logic controllers (FLCs), often termed as type-1 fuzzy logic systems using type-1 fuzzy sets, cannot handle large amount of uncertainties present in many real environments. Therefore, recently type-2 FLC has been proposed. The type-2 FLC can be considered as a collection of different embedded type-1 FLCs. However, the current design process of type-2 FLC is not automatic and relies on human experts. The purpose of our study is to make the design process automatic. Moreover, to reduce the computation time of T2GFLC an efficient type-reduction strategy for interval type-2 fuzzy set is also introduced. The evolved type-2 FLCs can deal with large amount of uncertainties and exhibit better performance for the mobile robot. Furthermore, it has outperformed their type-1 counterparts as well as the adaptive type-1 FLCs.

Index Terms**—**Interval Type-2 FLC, Interval Type-2 Fuzzy Sets, Genetic-Algorithms, Mobile Robot, Optimization.

I. INTRODUCTION

FUZZY systems are fundamental methodologies to represent and process linguistic information, with mechanisms to deal with uncertainty and imprecision. With such remarkable attributes, fuzzy systems have been widely and successfully applied to control, classification and modeling problem and in a considerable number of applications. A fuzzy model is a set of fuzzy rules and the associated membership functions (MFs) that maps inputs to outputs. Fuzzy rules and MFs are either provided by human experts or learned from sample data. Many decision-making and problem-solving tasks are too complex to be understood quantitatively. People however succeed by using knowledge that is imprecise rather than precise. The construction of fuzzy logic controllers (FLCs) based on the appropriate expert knowledge base (KB) can be quick and effective. On the other hand, without such an expert KB the design of FLCs can be frustrating as it relies on trial and error rather than a guided approach.

To surmount this shortcoming, genetic algorithms (GAs) can be considered as a powerful tool to perform tasks such as generation of fuzzy rule base (RB), optimization of fuzzy RB, generation of MFs, and tuning of MFs types. These algorithms mimics the natural evolution and provide an effective way for searching a large and complex solution space to give close to optimal solutions in much faster times than random trial-and-error.

For most fuzzy logic control problems, the most important issue is to determine the parameters that define the type-2 MFs. Because of this, the type-2 MFs optimization problems can be converted to parameter optimization problems. These parameters are generally based on the expert KB that is derived from heuristic knowledge of experienced control engineers and/or generated automatically. A variety of methods such as GAs, neural networks (NNs) have been used to improve the behavior of parameter optimization problem as well as selection and definition of fuzzy rules.

Mendel [1] and Hagras [2] have shown that the type-1 fuzzy logic systems (FLSs) may be unable to model and minimize the effect of uncertainties that prevails' in the real world applications. One restriction is that a type-1 fuzzy set is certain in the logic where the membership grade for each input is a crisp value. On the other hand, interval type-2 FLCs (that use interval type-2 fuzzy sets, characterized by fuzzy MFs) can handle the uncertainties.

GA was used by Martínez [3] in optimization of type-2 FLC. He applied GA to design FLC for the control of the perturbed autonomous wheeled mobile robot. Melin and Castillo [4] proposed a method based on type-2 fuzzy sets and neural networks called neuro-fuzzy to learn the parameters of the fuzzy system for intelligent control of nonlinear dynamic plants. Tan [5] used GA to optimize the parameters of FLCs. His proposed approach used mixed (type-1 and type-2) fuzzy sets for real time control.

[1] This paper was received on 27 June 2014, revised on 10 July 2014 and accepted on 25 July 2014.

Pintu Chandra Shill is with the Department of Computer Science and Engineering, KUET, Khulna 9203, Bangladesh. E-mail: pintu@cse.kuet.ac.bd

M. A. H. Akhand is with the Department of Computer Science and Engineering, KUET, Khulna 9203, Bangladesh.

Md. Saidul Islam is with the Department of Electrical and Electronic Engineering, RUET, Rajshahi, Bangladesh.

M. M. Hafizur Rahman is with the Department of Computer Science, KICT, International Islamic University Malaysia (IIUM), Kuala Lumpur, Malaysia.

Type-1 fuzzy logic controllers (FLCs) are known for their ability to compensate for structured and unstructured uncertainties, to a certain degree. However, type-2 fuzzy engines have been credited to be more powerful in compensating for even higher degrees of uncertainties [6]-[8]. They are particularly suitable for time-variant systems with unknown time-varying dynamics. They also allow for more flexibility to alleviate the problems associated to the uncertainties pertaining to the choice of the system's fuzzy control rules and fuzzy membership functions.

This paper highlights a contribution to the development of type-2 fuzzy logic controller with GA.GA is employed to the simultaneous design of type-2 MFs and rule sets for type-2 fuzzy logic controllers. It is found that our proposed integrated architecture is able to generate comprehensible and reliable fuzzy rules and tuning the optimal MFs parameters by a self-learning adaptive method. We simulated type-1 and type-2 fuzzy logic controllers to perform a comparative analysis of the systems' response, in the presence of uncertainty.

The rest of the paper is organized as follows: Section II presents an introductory explanation of type-2 fuzzy sets and FLC. In Section III presents the problem statement and the kinematic and dynamic models of the mobile robot. Section IV we introduce the key ideas of our approach called the type-2 genetic fuzzy logic controller (T2GFLC), Section V provides a simulation study of the mobile robot using the controller described in Section IV. Section VI describes the sensitivity of T2GLC with respect to the type reduction strategy. Finally, Section VII presents some concluding remarks and some future directions.

II. Type-2 Fuzzy Sets and FLC

A. Interval Type-2 Fuzzy sets

The concept of a type-2 fuzzy set was introduced by Prof. Zadeh [9] in 1975. A type-2 fuzzy set is defined by membership function. The fuzzy grade of that is a fuzzy set in the closed interval [0,1] rather than a point in [0,1].

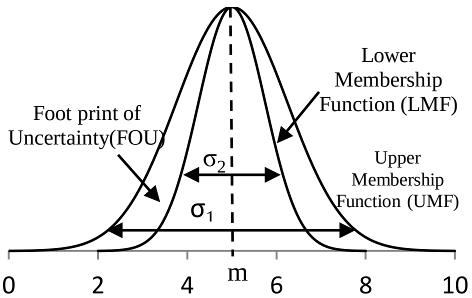


Fig. 1 An Interval type-2 fuzzy set.

A type-2 fuzzy set, denoted as $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x,u)$ [10], where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e., $\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) \mid \forall_x \in X \quad \forall_u \in J_x \subseteq [0,1]\}$ in which $0 \le \mu_{\tilde{A}}(x,u) \le 1$. A can also be expressed as follows [10]:

$$A = \int_{x \in X} \int_{u \in J_x} \mu_A(x,u)/(x,u) \quad J_x \in [0,1]$$ where

$\iint$ denotes union over all admissible x and u. J_x is called primary membership of x, where $J_x \in [0,1]$ for $\forall_x \in X$ [10]. The uncertainty in the primary memberships of a type-2 fuzzy set $\tilde{A}$, consists of a bounded region that is called the footprint of uncertainty (FOU)[10]. It is the union of all primary memberships [10].

B. Type-2 FLC

A type-2 FLC comprises five components, which are fuzzifier, knowledge base (KB) consisting of rule base (RB) and database (DB), fuzzy inference engine, type- reducer and defuzzifier as depicted in Fig. 2.

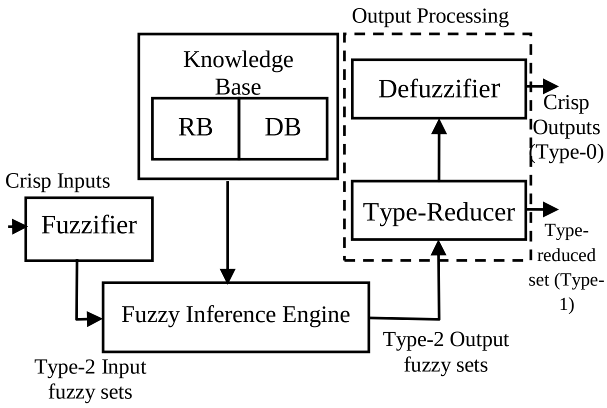


Fig. 2 A type-2 FLC.

B.1 Fuzzifier

Since the input is in crisp normalized values, a fuzzification operator $fuzz$ is used to fuzzify it in fuzzy form. The fuzzifier maps a crisp input vector with p inputs $x = (x_1, \ldots\ldots x_p)^T \in X_1 \times X_2 \times \ldots\ldots \times X_p \equiv X$ into input fuzzy sets, $\tilde{A}_x$ [11-12]. However, we will use most frequently used singleton fuzzification method as it is fast to compute and thus it is suitable for mobile real time operation. In the singleton fuzzifier, fuzzy set $\tilde{A}$ has only a single point of non-zero membership with support x_i, where $\mu_{\tilde{A}}(x,u) = 1$ for $x = x_i$ and $\mu_{\tilde{A}}(x,u) = 0$ for $x \ne x_i$ which input measurement x is perfect crisp.

B.2 Rule base

The rules will remain the same as in type-1 FLC but the antecedents and consequents will be represented by interval type-2 fuzzy sets [12]. Like most FLCs [13], the FLC discussed here applies the concepts of fuzzy implication and the compositional rules of inference for approximate reasoning. Suppose that we need to design a multiple-input-multiple-output (MIMO) mobile robot type-2 FLC having p inputs $x_1 \in X_1, \ldots\ldots, x_p \in X_p$ and c outputs $y_1 \in Y_1, \ldots.., y_c \in Y_c$ with ith fuzzy rule of the form:

$$R^i_{MIMO}: \text{ IF } x_1 \text{ is } \tilde{F}^i_1 \text{ ....and } x_p \text{ is } \tilde{F}^i_p, \quad \text{THEN}$$

$$y_1 \text{ is } \tilde{G}^i_1 \text{ .... } y_c \text{ is } \tilde{G}^i_c, \quad i = 1, \ldots, M$$

where $\tilde{F}^i_1, \ldots\ldots, \tilde{F}^i_p$ and $\tilde{G}^i_1, \ldots\ldots \tilde{G}^i_c$ are the antecedent and consequent MFs associated with the linguistic p input variables and c output variables, respectively, and M is the number of rules in the rule base.

B.3 Fuzzy Inference Engine

The fuzzy inference engine combines rules and gives a mapping from type-2 fuzzy sets in the input universe of discourse $U \in R^n$ to type-2 fuzzy sets in the output universe of discourse $V \in R$ based on the fuzzy logic principle. The structure of ith type-2 rule is having one output $y_k \in Y_k$:

$R^i: \tilde{F}^i_1 \times \tilde{F}^i_2 \times \ldots.. \times \tilde{F}^i_p \rightarrow \tilde{G}^i_k$ and the type-2 fuzzy relation can be expressed by membership function as:

$$\mu_{R^i}(\underset{\rightarrow}{x}, y) = \mu_{F^i_1 \times F^i_2 \times \ldots.. \times F^i_n \rightarrow G^i}(\underset{\rightarrow}{x}, y)$$

$$= \mu_{F^i_1}(x_1) \prod \ldots\ldots\ldots \prod \mu_{F^i_n}(x_n) \prod \mu_{G^i}(y)$$

,where $\prod$ denotes meet operation. The membership grades in the input type-2 fuzzy sets are combined with those in the output type-2 fuzzy sets using the extended sup–star composition; multiple rules are combined using the Join operation. They are defined and explained in a greater detail in [14]-[15].

B.4 Type- reduction

Type-reduction is that when an interval type-2 fuzzy sets is reduced to an interval-valued type-1 fuzzy set and then these type reduced sets are defuzzified to obtain crisp outputs. In this paper we will use centroid type reduction due to its reasonable computational complexity. For the centroid type reduction process, firstly combines the output type-2 fuzzy sets using union [4] (minimum t-norm), $\tilde{B} = \bigcup_{l=1}^{M} \tilde{B}^l$, as:

$$\mu_{\tilde{B}}(y) = \coprod_{l=1}^{M} \mu_{\tilde{B}^l}(y) \qquad \forall_y \in Y$$

,where $\mu_{\tilde{B}}(y)$ is the secondary membership function for the lth rule and it depends on join and meet operation. The centroid type reduction calculates the centroid of $\tilde{B}$. The type reduced set using the centroid type reduction can be expressed as:

$$y_c(\underset{\rightarrow}{x}) = \int_{\theta_1 \in J_{y1}} \ldots\ldots\ldots\ldots \int_{\theta_N \in J_{yN}} [f_{y1}(\theta_1) * \ldots\ldots * f_{yN}(\theta_N)] / \frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i}$$

,where i=1,….,N. To compute this process, at first y domain is discretized into N points and then J_{y_i} is discretized into T_i (i=0,1,…N) points. Total number of computations is $\prod_{i=1}^{N} T_i$.

B.5 Defuzzification

After the type-reduction stage, Defuzzify the type reduced interval set $y_c(\underset{\rightarrow}{x})$, determined by its left most y_l and right most point y_k using the average of of y_l and y_k . Hence the defuzzified crisp output is

$$Y(\underset{\rightarrow}{x}) = \frac{y_l + y_r}{2}$$

III. Mathematical Formulation of the Problem

A mobile robot has to move from an initial position to the target (dock) by avoiding collisions with a single stationary obstacle in optimal path. It may have to move along a straight path or take a turn depending on the current situations in order to generate a collision-free path. The problem is taken from [16]. Figure 3 depicts the simulated geometry for the robot and loading dock schematically, in which mobile robot is moving among single stationary obstacles, in the same workspace. The control system must find incrementally a path to the loading dock, independently of the initial position of the robot.

The path planning of the mobile robot is determined by the three input variables x, y and ϕ , (considered as a point mass), where x and y are the Cartesian co-ordinates of the mobile robot and ϕ is the robot direction angle relative to the horizontal axis x. The output variable is the control steering signal θ . As a first investigation, let us assume that there exists enough clearance between the robot, the walls and the obstacle in the workspace so that we can ignore the y-position co-ordinate of the robot. The co-ordinate y

TABLE I
DEFINITION OF LINGUISTIC VARIABLE.

x	ϕ	θ
LE-Left End LC-Left Center CE-Center RC-Right Center RE-Right End	NL-Negative Large NM-Negative Medium NS-Negative Small ZE-Zero PS-Positive Small PM-positive Medium PL-Positive Large	NB-Negative Large NM-Negative Medium NS-Negative Small ZE-Zero PS-Positive Small PM-positive Medium PB-Positive Large

will be re-introduced into the discussion shortly. The state spaces of two inputs are $-115^0 \le \phi \le 295^0$ & $0 \le x \le 100$, and one output θ within [-40^0, 40^0]. At every stage, the simulated mobile robot only moved forward until it hits the border of the loading dock. The final states (x_f, ϕ_f) will be equal or close to (10, 90^0). The robot kinematics model is described by the following dynamic equations.

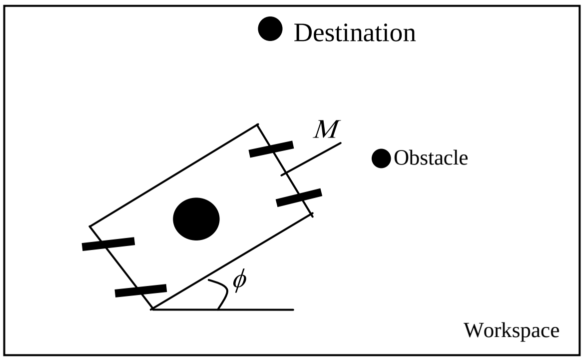


Fig. 3 Mobile Robot and loading dock illustration.

$$x_{t+1} = x_t + \cos(\phi_t + \theta_t) + \sin(\phi_t)\sin(\theta_t),$$
$$y_{t+1} = y_t + \sin(\phi_t + \theta_t) - \sin(\phi_t)\sin(\theta_t)$$
$$\phi_{t+1} = \phi_t - \sin^{-1}\left(\frac{2\sin(\theta_t)}{l}\right)$$

where l is the length of the robot, we assume l=4. Eqs. (1) will be used to derive the next state when present state and control are given. This experiment should be considered as an example of highly nonlinear complex problems. In this application we compare the performances of two differences i.e., type-1 FLCs called type-1 genetic fuzzy logic controller (T1GFLC) and T2GFLC.

IV. Hybrid Genetic-Fuzzy Optimization of the Type-2 FLCs

A genetic-type-2 fuzzy system depict in Fig.4 is a fuzzy system that uses a learning algorithm derived from genetic algorithm theory to determine its parameters (type- 2 fuzzy sets and fuzzy control rules). One of the most important factors for designing an effective type-2 FLC is to determine the optimal type-2 FLC parameters quickly and efficiently. In this paper, we employed GA to optimize the parameters of the MFs of Type-2 FLC; we consider using Gaussian Interval Type-2 MFs to each one of our three variables. At the same time, we also employed GA for the selection and definition of RB of type-2 FLC.

A. Encoding

Often one of the main challenges in designing a genetic algorithm to find a solution to a problem is finding a suitable way to encode the parameters. A chromosome is an encoded string of possible values for the parameters to be optimized. In this case, a chromosome is encodedby the defining parameters of interval type-2 MFs to each one of our three variables: x, ϕ (input variables) and control action θ(output variable) and the parameters of fuzzy control rules.

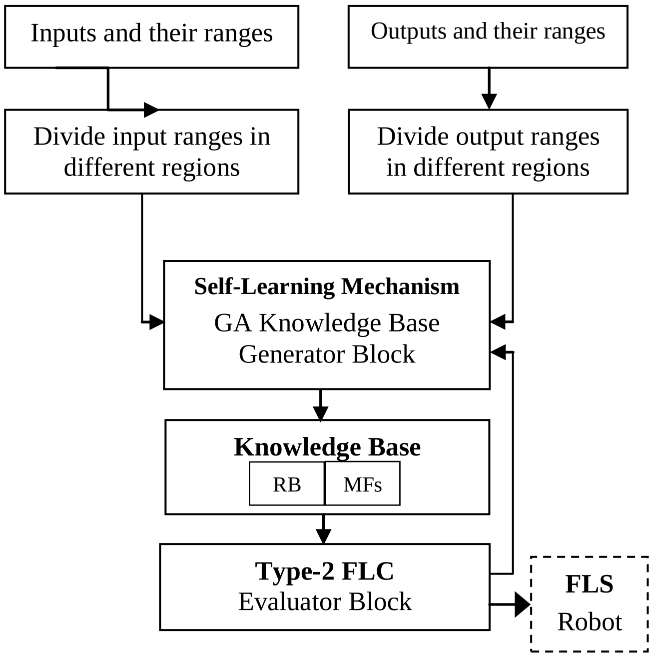


Fig. 4 Integration of type-2 FLCs and GA.

For this study, the domains for x, ϕ and θ are divided into 5, 7 and 7 interval type-2 fuzzy sets respectively. The linguistic terms (MFs) for each of the input and output variables are used to describe them, as in the following Table I. Here, each input and output type-2 MF is characterized by three parameters, one mean and two standard deviations.

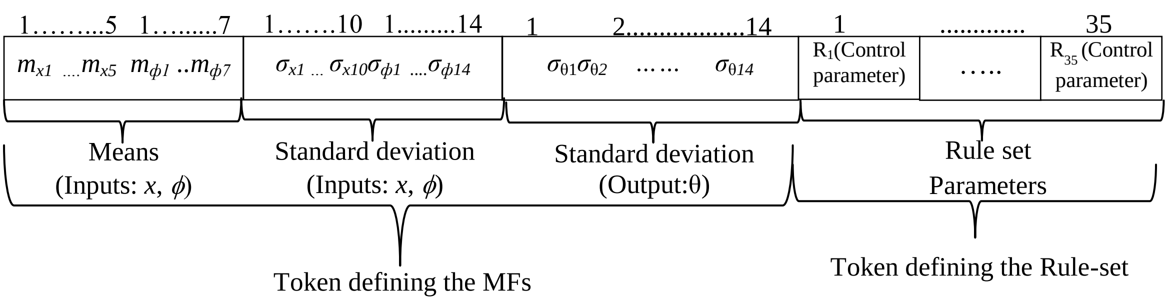


Fig. 5 GA coding schema of type-2 FLC (Chromosome).

Thus, 57 genes are used to represent the type-2 MFs FLC inputs and outputs.

An important characteristic of fuzzy models, FM, is the partitioning of the input and output space of system variables (input, output) into fuzzy regions using fuzzy sets [17]. The range of x is divided into five non-uniform intervals [0, 32.5], [32.5, 47.5], [47.5, 52.5], [52.5, 67.5], and [67.5, 100] and they are represented by five linguistic variables LE, LC, CE, RC and RE respectively. The range of ϕ is divided into seven non-uniform regions [-115, -27.5], [-27.5, 46], [46, 86.5],[86.5, 98.5] , [98.5, 146], [146, 216], [216, 295] and then they are represented by seven linguistic variables NL, NM, NS, ZE, PS, PM, and PL respectively. Similarly seven divided regions of the range of θ, [-40, -28], [-28, -12.5], [-12.5, -2.5], [-2.5, 2.5], [2.5, 12.5], [12.5, 28], [28, 40] are represented by variables NB, NM, NS, ZE, PS, PM and PB.

In this study five and seven gaussian type-2 fuzzy sets were used to partition the input spaces x and ϕ respectively and seven gaussian type-2 fuzzy sets for output spaces. The rule base, then, contains thirty-five (7 x 5) rules to account for every possible combination of input fuzzy sets. The fuzzy control if then rules are of the form: If x is ({LE, LC, CE, RC, RE}) and ϕ is ({NL, NM, NS, ZE, PS,PM, PL}) then θ is ({NB, NM, NS, ZE, PS, PM, PB}), output is one of the type-2 fuzzy sets used to partition the output space. 35 genes are used to represent the rule set. Therefore, we need to encode a total of (57+35) parameters for each individual of our population. In order to make this encoding schema we design a chromosome of 92 consecutive real genes. Figure 5 show a schematic of the chromosome structure to our genetic-type-2 FLS optimization approach.

B. Decoding Schema

In order to improving the accuracy of the type-2 FLC, we propose a genetic fuzzy model depending on the cooperation of the linguistic fuzzy rule base and type-2 fuzzy sets. Two tokens, parameters of MFs and rule base parameters, are included in the GA-chromosome, which are decoded to the knowledge base to evaluate each controller. Figure 6 shows the decoding schema of GA chromosome.

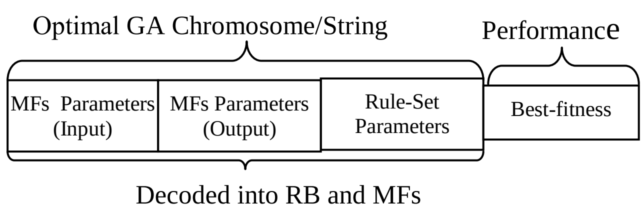


Fig. 6. GA Decoding schema of type-2 FLC

C. Measure of FIS performance-Objective function

The convergence of the GA to a feasible solution depends upon some objective measure of each potential FLC,s performance, when applied to the application. The FLC generates a fitness value according to the following evaluation function:

$$\text{Trajectory error/fitness} = \frac{\text{Length of robot trajectory}}{\text{Distance (Initail position, Desired final position)}}$$

D. GA procedure

The GA based approach uses real value encoding schema to encode each parameter (gene) in the chromosome.

For the integrated architecture, the GA based system works in the following ways (Fig.7):

Step1: **Initialization:** 60 strings (chromosomes), each string representing a potential solution, are generated by randomly chosen real numbers. Initialization of every chromosome followed by the grammatical correctness, the inner standard deviation σ_1 is less than the outer standard deviation σ_2 . Initialize the chromosome counter=1 and generation counter=1.

Step2: **Decode and Evaluation**: Decode the every chromosome into RB and MFs for the construction of type-2 FLC and FLC is executed on the mobile robot until it reaches the goal position or near to the goal position. Each potential solution (FLC) is evaluated and assigned a fitness value according to its

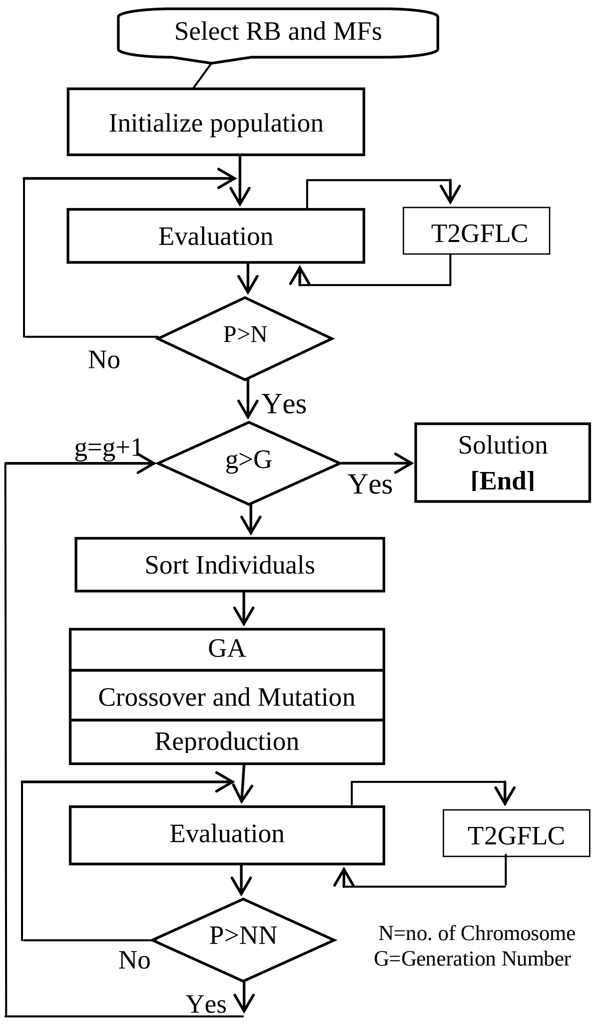


Fig. 7.Combined Type-2 FLC and GA algorithm

TABLE II (A) OPTIMAL MEANS AND STANDARD DEVIATION OF INTERVAL T2GFLC ANTECEDENTS MFS OF x AND ϕ.

	MFs of x														
	LE			LC			CE			RC			RE		
	m	σ_1	σ_2	m	σ_1	σ_2	m	σ_1	σ_2	m	σ_1	σ_2	m	σ_1	σ_2
	15.2148	32.6131	19.118	40.365	39.863	29.92	50.8752	21.8461	14.9546	63.59	43.8689	27.3946	95.51	55.38	63.63

MFs of ϕ				LE	LC	CE	RC	RE
	NL	m	-92.917	1 ZE	2 PM	3 NM	4 PB	5 PB
		σ_1	104.2758					
		σ_2	74.114					
	NM	m	4.861	6 NS	7 PS	8 PM	9 NS	10 PB
		σ_1	196.372					
		σ_2	170.334					
	NS	m	77.545	11 NM	12 NS	13 PS	14 PM	15 PB
		σ_1	134.909					
		σ_2	91.6082					
	ZE	m	89.8335	16 NM	17 ZE	18 ZE	19 PM	20 PM
		σ_1	120.334					
		σ_2	97.334					
	PS	m	143.303	21 NB	22 NM	23 NS	24 PS	25 PM
		σ_1	71.484					
		σ_2	50.114					
	PM	m	185.3798	26 PS	27 NB	28 PS	29 NS	30 PS
		σ_1	82.0035					
		σ_2	42.18					
	PL	m	256.759	31 NB	32 NB	33 NM	34 NM	35 NS
		σ_1	140.079					
		σ_2	97.44					

TABLE II (B) OPTIMAL MEANS AND STANDARD DEVIATION OF INTERVAL T2GFLC CONSEQUENTS MFS OFθ.

MFs of θ																				
NB			NM			NS			ZE			PS			PM			PB		
m	σ_1	σ_2	m	σ_1	σ_2	m	σ_1	σ_2	m	σ_1	σ_2	m	σ_1	σ_2	m	σ_1	σ_2	m	σ_1	σ_2
-35.14	20.35	14.23	-26.27	22.76	10.56	-9.15	26.53	20.75	1.45	16.47	12.44	11.18	14.45	6.76	25.96	26.74	15.79	38.97	18.63	7.96

performance to the problem. Sort the individuals according to their fitness value. A potential solution (controller) that would have caused collision with the obstacle is automatically assigned a disastrous fitness

Step3: **Recombination:** Apply crossover and mutation operator to chromosomes and generate new chromosomes as well as new generation i.e., increment the generation counter. Check the termination condition (generation counter<number of maximum generations). Reset the chromosome counter to 1 and go to step 2 otherwise go to step 4.

Step4: **Stop:** The best fitted chromosome is kept and solution has been achieved.

V. SIMULATION RESULTS AND COMPARATIVE ANALYSIS

To evaluate the accuracy of the proposed system, we have carried out a series of experiments which the controller were evolved in our simulated arena. The optimal means and standard deviations of MFs for x, ϕ, and θ are shown in table II(a-b). The generated optimal rule base (after the conversion from optimal parameter to linguistic form) also shown in Table II (a).

TABLE III. FIVE INITIAL POSITIONS (x, y, ϕ) AND THEIR STEPS OF RESULT.

Case		1		2		3		4		5	
x	y	10	15	80	50	10	30	40	50	75	25
ϕ		180		-60		45		90		-44	
T1GFLC		42		57		37		32		48	
T2GFLC		29		42		28		21		37	

Figures 8(a), (b), and (c) show the two antecedents and one consequents type-2 MFs of T2GFLC. Figure 9(b) shows the performance comparison with T1GFLC (type-1) and T2GFLC. Figure 9 shows the times for the mobile robot to reach the goal position in 5 different initial conditions and their trajectories are plotted in Fig.10. Table III shows the five initial conditions for (x, y, ϕ) with their steps. From Figure 9, 10 and table III, it is obvious that the performances of T2GFLC are better than those in T1GFLC. It not only takes less steps to arrive the goal position using interval T2FLC, but also it shows the smoother trajectories (shown in Fig.10).

It has been found that the GA based system evolves to optimal type-2 MFs and RB after some generations.

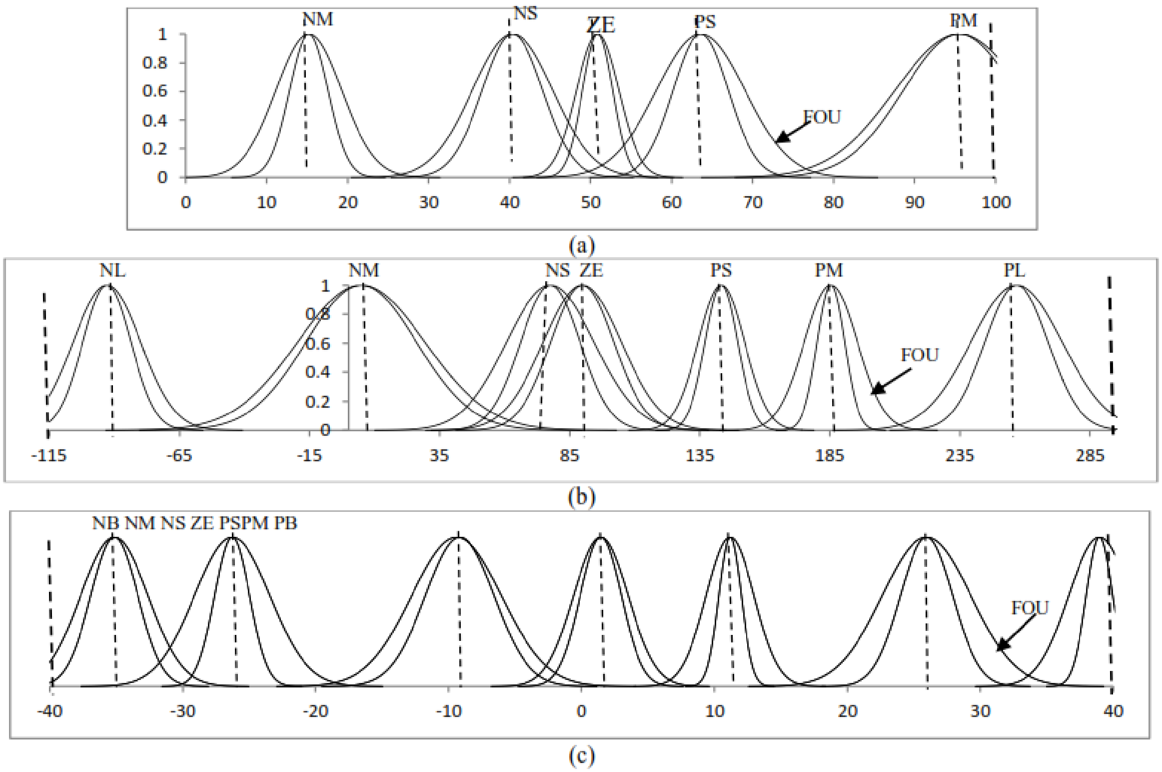


Fig. 8 Fuzzy type-2 MFs Gaussian shape for each linguistic fuzzy set value of the optimal solution (a) Input *x* (b) Input ϕ and (c) Output θ

An example of the genetic progress is presented in Fig. 9 (a) which demonstrations the performance of the best chromosome found so far against the number of generations.

VI. Sensitivity Analysis and Evaluating Our Work

In order to improve the overall performance of T2GFLC, we presented the modified type reduction strategy and used it as a type reducer. Then the simulation results are compared to the existing approach for evaluating our work.

A. *Modified Type reduction method*

The computation complexity of doing of type reduction strategy is very high. This has urged researchers to search for ways to alleviate this high computational burden if type-2 FLCs are to find their way to real-world applications. In order to reduce the computational complexity, in our study we introduced a type reduction strategy based on the vertical slice representation of interval type-2 fuzzy set. A type-2 fuzzy set $\tilde{A}$ can be represented as a collection of its vertical slices for the discrete case,

$$\tilde{A} = \sum_{x\in X}\left[\sum_{u\in J_x}\mu_{\tilde{A}}(x,u)/u\right]/x$$

$$= \sum_{i=1}^{N}\left[\sum_{u\in J_{x_i}}\mu_{\tilde{A}}(x_i,u)/u\right]/x_i = \sum_{i=1}^{N}\mu_{\tilde{A}}(x_i)/x_i$$

The vertical slice is embedded type-1 fuzzy set that can be easily reduced. The centroid of each vertical slice can be computed as follows:

$$u_j = \sum_{i=1}^{n_j} u_j^i * 1/\sum_{i=1}^{n_j} 1 = \frac{1}{n_j}\sum_{i=1}^{n_j} u_j^i$$

For an interval type-2 fuzzy set, the average of *n* point's discrete vertical slice is the mean of upper and lower MFs.

$$u_j = \frac{1}{n_j}\sum_{i=1}^{n_j} u_j^i = \frac{1}{2}(\bar{u}_j + \underline{u}_j)$$

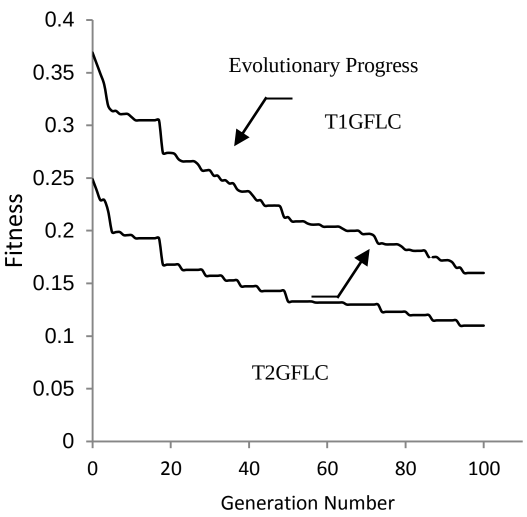
,where $\bar{u}$ and $\underline{u}$ are upper and lower grades of the type reduced set. The centroid of the interval type-2 set can be expressed as follows:

$$x_c = \sum_{j=1}^{N} x_j * u_j / \sum_{j=1}^{N} u_j$$

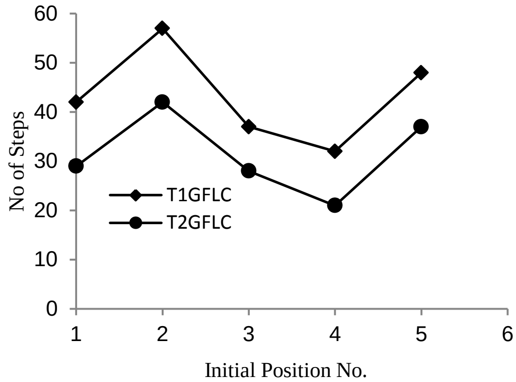
$$= \sum_{j=1}^{N} x_j * [0.5*(\bar{u}_j + \underline{u}_j)] / \sum_{j=1}^{N} 0.5*(\bar{u}_j + \underline{u}_j)$$

$$= \frac{(\sum_{j=1}^{N} x_j * \bar{u}_j + \sum_{j=1}^{N} x_j * \underline{u}_j)}{(\sum_{j=1}^{N} \bar{u}_j + \sum_{j=1}^{N} \underline{u}_j)}$$

It has been shown that the crisp value (x_c) of interval type -2 fuzzy set can be depends on lower and upper bounds on the foot print of uncertainty. The overall computation cost of our proposed approach has been reduced by using this modified type reduction strategy and compared with the other existing approach, our proposed algorithm much more efficient.



(a)



(b)

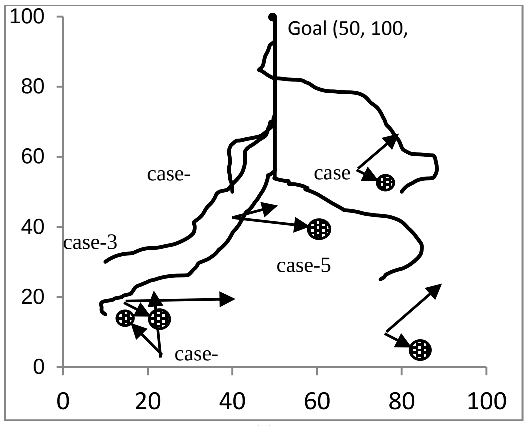
Fig. 9 (a) Shows the results of trajectory errors in T1GFLC and T2GFLC, and (b) Show the total steps to reach goal position by 5 different initial cases.

A. *Evaluating the work*

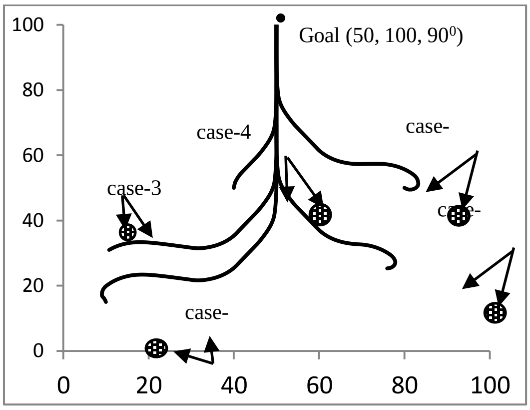
The mobile robot is a typical problem in nonlinear motion control of nonholonomic systems. It is a notable example that is generally used as benchmark problem for the evaluation of new control algorithms and as such it has been well analyzed [16].

Recently Mohammadian and Stonier in [16] have presented the fuzzy control system based on a genetic algorithm for trajectory tracking of a mobile robot in the presence of obstacle. In their work, binary coded GA used to tune the consequent of fuzzy rules. They define a mapping from the input space to the output space based on the combined fuzzy rule base using the defuzzifying procedure. According to the robot kinematics equations, the work of Mohammadian and Stonier has been used for comparison. Figure 11 shows simulated results of Mohammadian and Stonier.

Simulated results using the present proposed approach with enhanced type reduction strategy for the different initial positions are shown in Fig. 12. We compare our approach with existing approaches [16]; the comparison results show that the proposed approach outperforms in terms of shorter trajectories, trajectories smoothness and the processing time.

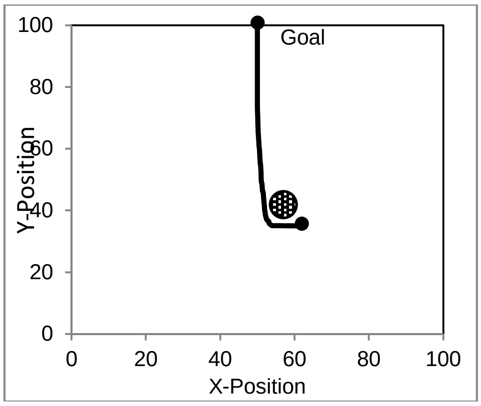


(a)

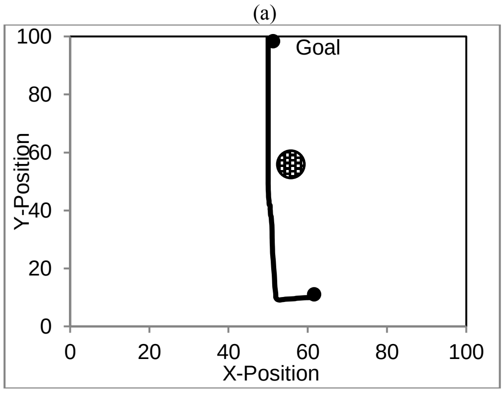


(b)

Fig. 10 (a) Show truck trajectories avoiding stationary obstacle (cross-hatched circle) via T1GFLC, and (b) Show via interval T2GFLC, all from 5 different initial conditions.



(a)



(b)

Fig. 11 Trajectories from the fuzzy controller for initial positions: (a) (62, 35, 135^0) and (b) (62, 10, 200^0).

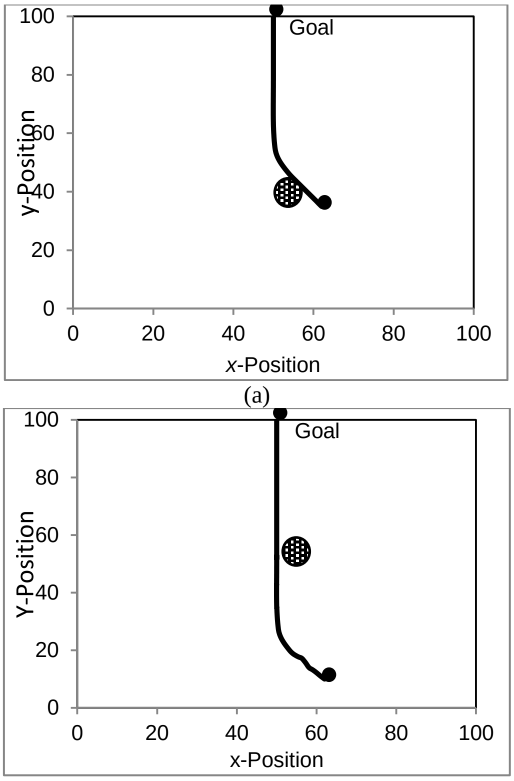


Fig. 12Trajectories from the type-2 fuzzy controller for initial positions: (a) (62, 35, 135^0) and (b) (62, 10, 200^0).

VII. Conclusions

This paper has revealed the possibility of using GA based architecture to evolve the type-2 MFs and Rule set parameters of interval type-2 FLCs. We have shown that an integrated FLCs and hybrid GAs architecture is a self-learning adaptive method were able to generate optimal MFs parameters and establish a reliable fuzzy control rules without any priori knowledge aimed at mobile robot control in real world environments. The ability of evolved type-2 FLCs is to provide control problems where no priori knowledge is available such as in mobile robots domain. The type-2 FLCs which were genetically evolved achieved a superior control performance in comparison to the type-1 FLCs (T1GFLC) and adaptive approach [16].

Suggestions for follow-up works that may come after this paper are as follows: This research work is to be extended for intelligent control of a mobile robot, control of a robotic arm in the presence of moving obstacle, the path planning problem for multiple mobile robots with more than one obstacles either moving or fixed in the workspace.

Reference

[1] H. A. Hagras, “A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots,” *IEEE Trans. on Fuzzy Systems*, vol.12, no.4, pp.524–539, 2004.

[2] J.M. Mendel, *Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, Englewood Cliffs, NJ, 2001.

[3] R. Martínez, O. Castillo, L. T. Aguilar, “Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms,” *Information Sciences*, pp.2158–2174, 2009

[4] P. Melinand O. Castillo, “Intelligent control of nonlinear dynamic plants using type-2 fuzzy logic and neural networks,” in proc. of NAFIPS, 2002, pp. 22–27.

[5] D. W. W. W. Tan, “A simplified type-2 fuzzy logic controller for real-time control,” *ISA Trans.*, vol.45, pp.503-516, 2006.

[6] Hagras, H.A, “A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots,” *IEEE Trans. Fuzzy Syst.*, vol. 12, no.4, pp. 524–539 2004.

[7] Mendel, J.M, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*,.Prentice-Hall, 2001.

[8] Mendel, J.M. and John, R.I.B., “Type-2 fuzzy sets made simple,” *IEEE Trans. Fuzzy Syst.*,vol. 10, no. 2, pp.117–127, 2002.

[9] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning-I,” *Inform. Sci.*, vol.8, pp.199-249, 1975.

[10] J. Mendel and R. John, “Type-2 fuzzy sets made simple,” *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 117–127, Apr. 2002.

[11] Q. Liang, N. Karnik, and J. Mendel, “Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems,” *IEEE Trans. Syst., Man, Cybern. C*, vol. 30, pp. 329–339, Aug. 2000.

[12] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Upper Saddle River, NJ: Prentice-Hall, 2001

[13] J. M. Jou, P. Y. Chen, and S. F.Yang, “An Adaptive Fuzzy Logic Controller: Its VLSI Architecture and Applications,” *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol.8, no.1, Feb. 2000.

[14] N. N. Karnik and J. M. Mendel, “Operations on type-2 fuzzy sets,” *Fuzzy Sets and Systems*, vol.122, pp. 327–348, 2001.

[15] M. Mizumoto and K. Tanaka, “Some Properties of fuzzy sets of type-2,” *Infom. Control*, vol. 31 pp.312-340, 1976.

[16] M. Mohammadian and R. J. Stonier, “Fuzzy logic and genetic algorithms for intelligent control and obstacle avoidance,” in *Complex Systems: Mechanism of Adaptation*, R. J. Stonier and X. H. Yu, Eds. Amsterdam: IOS Press, 1994, pp. 149-156.

[17] L. X. Wang, *Adaptive Fuzzy systems and control: Design and Stability Analysis*, PTR Prentice-Hall, Englewood cliffs, NJ. 1994.

[18] R. I. John, “Embeded interval valued type-2 fuzzy sets,” *in Proc. FUZZ-IEEE Init. Conf.*, May 2002, pp. 1316-1319.

[20] B. Kosko .*Neural networks and fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Englewood Cliffs, NJ: Prentice-Hall, 1992.

[21] R.I. John and C. Czarnecki, “A type 2 adaptive fuzzy inferencing system,” in Proc. IEEE Systems, Man and Cybernetics, 1998, pp.2068-2073,.

[22] Q. Liang and J.M. Mendel, “Interval type-2 logic systems: theory and design,” *IEEE Trans. on Fuzzy Syst.*, vol. 8 no.5, pp.535-550, Oct. 2000.

Pintu Chandra Shill received the B.Sc. degree in Computer Science Engineering (CSE) from Khulna University of Engineering and Technology (KUET), Bangladesh in 2003, M.Sc. degree in Computer Engineering from Politecnico di Milano, Italy in 2008 and Doctoral degree in System Design Engineering in 2009 from University of Fukui, Japan. 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.

M. A. H. Akhand received the B.Sc. degree in Electrical and Electronic Engineering from Khulna University of Engineering and Technology (KUET), Bangladesh in 1999, the M.E. degree in Human and Artificial Intelligent Systems in 2006, and the Doctoral degree in System Design Engineering in 2009 from University of Fukui, Japan. He joined as a lecturer at the Department of Computer Science and Engineering at KUET in 2001, and is now a Professor. He is a member of Institution of Engineers, Bangladesh (IEB). His research interest includes artificial neural networks, evolutionary computation, bioinformatics, swarm intelligence and other bio-inspired computing techniques.

Md. Saidul Islam received the B.Sc. degree in Electrical and Electronic Engineering (EEE) from Rajshahi University of Engineering and Technology (RUET), Bangladesh in 2011. He joined as a industrial engineer at the Department of Industrial Engineering & Planning, Dird Composite textile ltd. in 2011 and currently he is serving as a senior industrial engineer. His research interest includes soft computing, power system and bio-informatics.

M. M. Hafizur Rahman received his B.Sc. degree in Electrical and Electronic Engineering from Khulna University of Engineering and Technology (KUET), Khulna, Bangladesh, in 1996. He received his M.Sc. and Ph.D. degree in Information Science from the Japan Advanced Institute of Science and Technology (JAIST) in 2003 and 2006, respectively. Dr. Rahman is now an Assistant Professor in the Dept. of Computer Science, KICT, IIUM, Malaysia. Prior to join in the IIUM, he was an Associate Professor in the Dept. of CSE, KUET, Khulna, Bangladesh. He was also a visiting researcher in the School of Information Science at JAIST and a JSPS Postdoctoral Research Fellow at Research Center for Advanced Computing Infrastructure, JAIST & Graduate School of Information Science (GSIS), Tohoku University, Japan in 2008 and 2009 & 2010-2011, respectively. His current research includes parallel and distributed computer architecture, hierarchical interconnection networks, and optical switching networks.