Design and Implementation of an Effective Fuzzy Logic Controller based on Quantum Inspired Evolutionary Algorithm

Pintu Chandra Shill†, Md. Amjad Hossain‡, Md. Kowsar Hossain‡, Md. Faijul Amin†, 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, amin, murase}@synapse.his.u-fukui.ac.jp, {amjad_kuet, auvikuet}@yahoo.com

***Abstract*-This paper proposes a new approach based on quantum inspired evolutionary algorithm (QIEA) for effective selection and definition of fuzzy if-then control rules as well as the shapes of membership functions (MFs) to design fuzzy logic controllers (FLCs). The majority of works done on designing FLCs rely on the knowledge base derived from imprecise heuristic knowledge of experienced operators or persons. These traditional methods, however, are cumbersome to implement and very time consuming to evaluate. Our proposed approach is a self-learning adaptive method and decomposes a problem in such a way that leads to more effective knowledge acquisition and improved control performance with the FLCs. In order to verify the effectiveness of this self-learning adaptive method, a standard test-bed, the truck backer-upper problem, is considered as the test problem. During each generation, the rules are updated and the MFs' parameters are altered using a complementary double mutation operator (CDMO) and a discrete crossover (DC). This paper also demonstrates the effect of different fuzzification and defuzzification methods on the response of the FLC. The center of gravity (COG) and modified COG are used as defuzzifier to analyze the results of the fuzzy controller. The experimental results show that the proposed approach with different fuzzification and MCOG to design FLCs performs better than the traditional methods with triangular fuzzification and COG in terms of required time to backing up the truck.**

***Index Terms*-Fuzzy Logic Controller, Fuzzy Rule base, Quantum Inspired Evolutionary Algorithm, Optimization, Defuzzification, Backing up a truck**

I. INTRODUCTION

A fuzzy rule-based model consists of a set of fuzzy IF-THEN rules which maps inputs to outputs. It has numerous practical applications in control [1], prediction and inference [2, 3] and has been found to be quite successful in examining problems with uncertainty and non-linearity. To define the membership functions (MFs) and fuzzy control rules of fuzzy system, it is necessary to have expert intuition because it helps to design effective FLCs. Most of fuzzy logic controllers (FLCs) to date have been static and based upon knowledge derived from imprecise heuristic knowledge of experienced operators. The construction of FLCs based on this type of expert knowledge can be quick and effective, provided the expert knowledge is available. On the other hand, without such an expert knowledge the design of FLCs can be slow as it relies on trial and error rather than a guided approach. So we need an automated knowledge acquisition method for FLCs which will be able to improve the overall performance in fuzzy control.

For most fuzzy logic control problems, the most important issue is to determine the parameters that define the MFs and MFs optimization problems can be converted to parameter optimization problems. These parameters are generally based on the expert knowledge that is derived from heuristic knowledge of experienced control engineers and/or generated automatically. A variety of methods such as genetic algorithms (GAs), neural networks (NNs), self-organizing feature map (SOFM), tabu search (TS), and particle swarm optimization (PSO) have been used to improve the behavior of parameter optimization problem as well as selection and definition of fuzzy rules.

GA was used by Belarbi [4] in fuzzy rule base minimization. He applied GA to design FLC for the control of the pole and cart system and the control of the concentration in continuously stirred tank reactor. Arslan and Kaya [5] presented a method to adjust the shapes of MFs using GA. They designed a fuzzy logic control system having single input and output. Meredith [6] also applied GA to the fine tuning of MFs in a FLC for a helicopter. Bagis [7] presented an approach for the determination of the structure and parameters of fuzzy rule base. He applied this approach in the modeling of the nonlinear or complex systems. Bai and Chen [8] proposed an automatic method for students' evaluation task. The purpose was to automatically construct the grade MFs of lenient-type grades, strick-type grades, and normal-type grades of fuzzy rules. Yang and Bose [9] presented a method for generating fuzzy MFs with an unsupervised learning using SOFM. The SOFM approach is a two-step procedure; firstly, generate the proper clusters and secondly generate the fuzzy MFs according to the clusters in the first step. They applied this method to pattern recognition. A fuzzy knowledge integration technique based on the PSO was presented by Huang **[10]** which consists of two phases: Firstly, it encodes the fuzzy rule sets and fuzzy sets with its corresponding MFs. Secondly, the particle swarm algorithm was used to explore the fuzzy rule sets, fuzzy sets and MFs to its optimal or the approximately optimal extent.

This paper proposes a new approach based on quantum inspired evolutionary algorithm (QIEA) for the optimum design of FLCs involving large number of parameters. The QIEA employed as a self-adaptive learning strategy to automatically tune the parameters of

doi:10.4304/jcp.7.3.586-596

MFs and select the optimal set of fuzzy rules. In the second part, in order to improve the overall performance, we have used modified center of gravity method (MCOG) as a defuzzification method. On the other hand, two different fuzzifiers have been used for inputs to be interpreted to get better response from the FLCs. These fuzzifiers are the triangular and gaussian. The MCOG method works on the basis of information concerning MFs' shapes (narrow or wide). The very narrow consequent MFs indicates a very strong belief in that rule, whereas too wide indicate much less belief in that rule. As a test problem backing up the truck problem is considered.

The rest of the paper is organized as follows: In section II we illustrate literature review relevant to the FLCs. A brief description of the problem for which FLCs is to be design is given in section III. In section IV we introduce the key ideas of integrated architecture of FLCs with QIEA. This section also presents the methodology adopted for solving the problems describe in section III in fuzzy environment. Section V includes the experimental results and comparative analysis on backing up the truck problem and finally section VI presents some concluding remarks based on the present study and some future direction.

II. Fuzzy Logic Controller

The idea of fuzzy logic was first introduced in 1960s by Professor Lofti Zadeh[11]. The general configuration of a FLC can be divided into four main parts; fuzzification interface, a rule base, an inference mechanism and defuzzification interface (Fig.1).

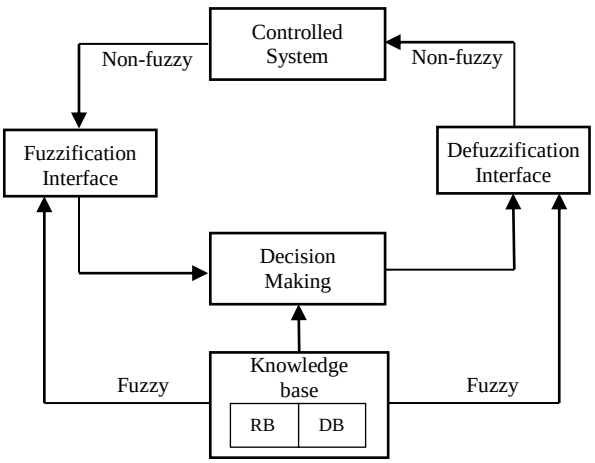


Figure 1. Fuzzy Logic Controller (FLC)

A. Fuzzifier

Fuzzy sets (Fig. 2) can be defined as μ_A, membership function that associates with each element $x \in X$ where X is the universe of discourse, a number called membership grade $\mu_A(x) \in [0,1]$. The *function* of the *fuzzifier* is to map a crisp input value $x \in X$ into a fuzzified value in $A \in U(universe)$. In this paper, we have used Non-singleton fuzzifier: $\mu_A(x_i)$ realizes maximum value 1 at $x = x_i$ and decrease from 1 to 0 while moving away from $x = x_i$.

B. Fuzzy rule base

The general form of a fuzzy rule used in most FLCs is as follows:

R^k : IF x_1 is F_1^k,.....and x_n is F_n^k, THEN y is G^k.

$l = 1, 2, 3, ...M$, M = number of rules in the rule base

where $x_1, x_2, ...x_n$, and y are the input and output linguistic variables respectively. F_i^k and G^k are fuzzy sets in input sets $X \in X_1 \times X_2 \times X_n$ and output sets *Y*. Each fuzzy IF-THEN rule has an antecedents (or IF) part containing several preconditions and a consequent (THEN) part describing the output action.

C. Fuzzy Inference Engine

A fuzzy relation R^l can be defined as:

$R^l : X \times Y = \{(\vec{x}, y) : x \in X, y \in Y\}$ where $\vec{x}$ is a vector of the form $(x_1, x_2, ...x_n)^T$. This relation R^l is the actual process of mapping from fuzzy sets in *X* to fuzzy sets in *Y*. $F_1^l \times F_2^l \times ... F_n^l \rightarrow G^l$, can be called fuzzy inference process. The process involves MFs, fuzzy logic operators, and if-then control rules. Fuzzy Inference process involves application of the fuzzy operators (AND or OR) in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules.

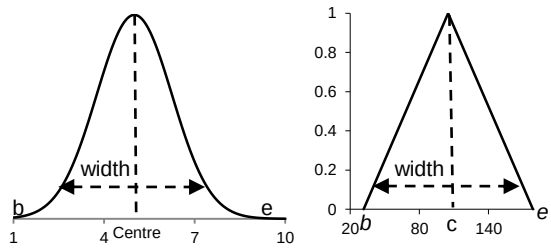


Figure 2. Fuzzy set (MFs: Gaussian and Triangular)

D. Defuzzifier

The final crisp output values are determined using a procedure known as "defuzzification process".

D.1. Center of Gravity (COG)

The "Center of gravity" method is used as defuzzification method. Defuzzification produces a numerical (point-estimate) output value for the fuzzy set. The defuzzification method is centroid defuzzification which uses the fuzzy centroid θ as output:

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

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

D.2. Modified COG

The modified height defuzzifier was used to handle the consequent uncertainty and improve the FLCs response. The expression generally used for a fuzzy set is as follows:

$$y_{mh} = \frac{\sum_{i=1}^{M} y^k \mu_{B^k}(\bar{y^k}) / \delta^{k^2}}{\sum_{i=1}^{M} \mu_{B^k}(\bar{y^k}) / \delta^{k^2}} \quad (2)$$

where δ^k is the spread measure of the kth consequent set. For Gaussian membership function δ^k is the standard deviation whereas triangular MF δ^k is the length of its base.

E. Membership functions

E.1. Triangular/Trapezoidal MF

It has the function name *trimf* defined by three points delimiting a triangle.

$$f_{triangulae}(x) = \begin{cases} 0, & \text{if } x < x_1 \\ 2\dfrac{x\text{-}x_1}{x_2 - x_1}, & \text{if } x_1 \le x \le x\dfrac{x_2 + x_1}{2} \\ 2\dfrac{x_2 - x}{x_2 - x_1}, & \text{if } \dfrac{x_2 + x_1}{2} < x \le x_2 \\ 1, & \text{if } x > x_2 \end{cases} \quad (3)$$

E.2. Gaussian MF

Gaussian MFs are built on the Gaussian distribution curve and are denoted by *gauss_mf*.

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

III. Problem Statement

Normal driving instincts can cheat us when attempting to back up a truck to a loading dock [12, 13]. The task is such a difficult one that even for a highly skilled driver needs to go forward and backward numerous times in order to position the truck at the dock successfully. If the driver is not allowed to make forward movements, successful backing becomes improbable. So, the main challenge for the truck backer-upper control problem is to design a controller to successfully back up a truck to a loading dock from any initial position. The problem, to back up a simulated truck to a loading dock in a planner parking lot is depicted in Fig.3 [12, 14]. The motion of the truck can be described by the following set of equations:

$$\begin{cases} x(t+1) = x(t) + b * \cos(\phi(t)) \\ y(t+1) = y(t) + b * \sin(\phi(t)) \\ \phi(t+1) = \phi(t) + \theta(t) \end{cases} \quad (5)$$

where ϕ is the truck angle with horizontal line, (x, y) is the coordinate of the position of rear centre of the truck and θ is the steering angle. A fuzzy controller for this problem would have three input variables (ϕ, x, y) by which the truck position could be determined exactly and an output variable θ. The controller have to make the truck reach at the loading dock at a right angle ($\phi = 90^0$) and to align the position (x, y) of the truck with the desired location (x_f, y_f). The truck should move backward by some fixed distance b at every stage. Assuming enough clearance between the truck and the loading dock we can ignore the y position of coordinate at the time of output (steering angle θ) calculation.There is no predefined path for each truck location and therefore the optimum steering angle is unknown.

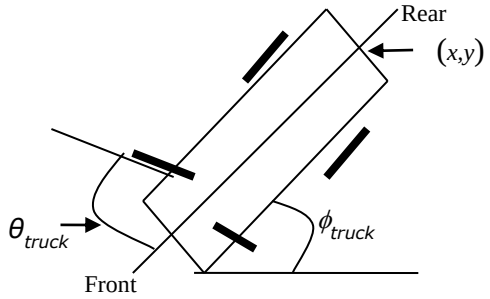


Figure 3. Diagram of Simulated Truck

So, at every stage our proposed approach should calculate the steering angle $\theta = f(x, \phi)$ to back up the truck to a loading dock from any position and initial truck angle in the loading zone. The loading zone corresponding to the plane {0, 0}×{100, 100}.

This experiment should be considered as an example of highly nonlinear complex problems. The fuzzy logic control system has to find the correct function that maps points from the three dimensional input-space to the appropriate output variable, continuously from the given initial point until the loading dock, and for all possible initial positions.

IV. Integration of FLCs and QIEA

The QIEA is a stochastic search and optimization method which combines the principles of natural biological evolution and quantum computation such as the quantum bit and the superposition of states. Superposition of logical state can be expressed as a vector [16], [15]:

$$\alpha|0\rangle + \beta|1\rangle \leftrightarrow \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (6)$$

The complex numbers α and β are called the "amplitudes" of the superposition. A quantum bit (Q-bit) is defined as the smallest unit of information in two- state computer which is defined as a pair of numbers (α, β) as

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (7)$$

And a Q-bit individual as a string of n Q-bits is defined as [16]

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \beta_1 & \beta_2 & \cdots & \beta_n \end{bmatrix} \quad (8)$$

Where $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1, 2, 3 \ldots n$. So a Q-bit individual of n bits can represent 2^n states at a time.

QIEA has a better characteristic of population diversity than others, since it can represent linear superposition of states probabilistically. But higher value of n needs higher computing time of the algorithm. For an optimization problem with fitness function $f(x_1, \cdots, x_m)$, if we represent each real variable x_i by k bits then the total number of Q-bits in an Q-bit individual $n = k \times m$. So if we apply the evolutionary operators on huge number of bits then they will definitely take a significant amount of time to execute.

A Q-gate defined as a mutation operator is applied on the qubits to update their probability amplitudes as follows:

$$\begin{pmatrix}\alpha_i' \\ \beta_I'\end{pmatrix} = \begin{bmatrix}\cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i)\end{bmatrix}\begin{pmatrix}\alpha_I \\ \beta_I\end{pmatrix} \quad (9)$$

where $\Delta\theta_i$, i=1, 2, 3…., n, is the rotation angle. $|\alpha_i'|$ and $|\beta_i'|$ must satisfy the normalization condition $|\alpha_i'|^2 + |\beta_i'|^2=1$. The crossover operator is applied after a given interval of generations.

To design the FLCs based on the QIEA, at first we have to find out the way to represent the fuzzy rule base and MFs parameter in the form of chromosome so that the QIEA can provide the best rule base to FLCs after applying its evolutionary operators on the chromosome (rule base) for some generations. The overall integration of FLCs with QIEA is shown in Fig. 4.

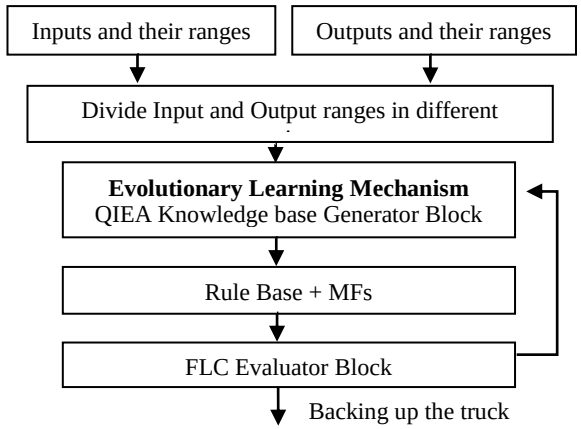


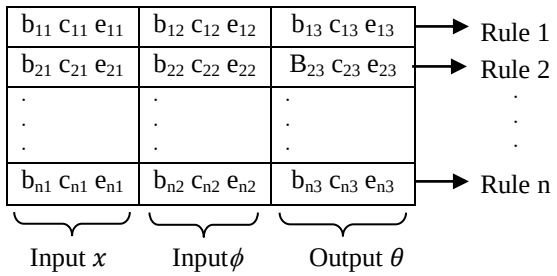
Figure 4. Integrated FLCs and QIEA Architecture

A. Division of ranges of input and output variables

For backing the truck problem we have two inputs x and ϕ and one output θ from which we have to produce fuzzy rules and MFs. As the first step, divide the domain of x, ϕ and θ into different regions. Here we divide the range of x into 5 regions and range of both ϕ and θ into 7 regions. The linguistic variables for these regions are defined of Table I.

B. Encoding and Generation of Fuzzy Rules

After defining the regions, encode the input and output spaces fuzzy regions MFs and fuzzy control rule set into string (real value).The MF for each of input and output variables are characterized by the three variables – begin (b), center (c) and end (e). The rule base containing 35 rules are generated using these variables as follows:



Where n= 35. To represent each rule actually we use center (C) and half-width (W) of the variables (input and output) then to define MF, we calculate begin = C – W/2 and end = C + W/2. That is each rule is represented as follows:

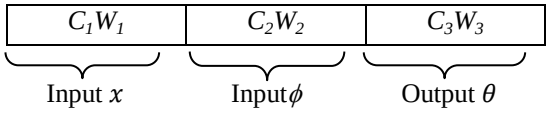


Figure 5 shows the chromosome structure that defined the fuzzy rule set and MFs parameter.

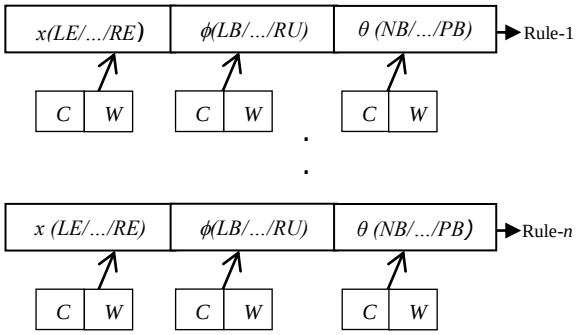


Figure 5. Chromosome structure

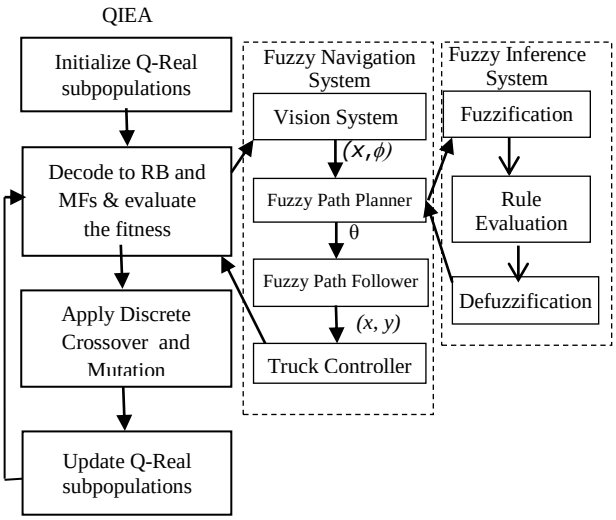


Figure 6. Data flow of proposed QIEA with fuzzy navigation system and fuzzy inference system

These rules are generated using if-then association in such a way that C and W for x, ϕ and θ lie in their particular regions. For example, if x is in LE and ϕ in RV then θ in NB. An initial rule base may contain rules as shown in following (Table II).

TABLE I.
DEFINITION OF LINGUISTIC VARIABLE

x	ϕ	θ
	LB-Left Below	NB-Negative Below
LE-Left End	LU-Left Upper	NM-Negative Medium
LC-Left Center	LV-Left Vertical	NS-Negative Small
CE-Center	ZE-Zero	ZE-Zero
RC-Right Center	RV-Right Vertical	PS-Positive Small
RE-Right End	RU-Right Upper	PM-positive Medium
	RB-Right Below	PB-Positive Below

A fuzzy rule base is a collection of fuzzy rules, which are in conventional IF-THEN form with a premise part to describe the conditions and a consequent part to state the conclusions or actions. This rule base simply indicates the possible ranges for the output values as the input values are given. The production is used to represent the premise.

TABLE II.
FUZZY CONTROL RULE SET

		ϕ						
		LB	LU	LV	VE	RV	RU	RB
x	LE	1 NM	2 PS	3 NM	4 PB	5 NL	6 NM	7 PM
	LC	8 PB	9 NL	10 NS	11 ZE	12 NM	13 PS	14 PB
	CE	15 PM	16 PS	17 NM	18 ZE	19 PB	20 PM	21 NS
	RC	22 NS	23 NM	24 PM	25 PS	26 NS	27 NL	28 PM
	RE	29 PM	30 NM	31 NS	32 PS	33 PB	34 PM	35 NS

C. Quantum-Inspired Evolutionary Procedure

The choice of input-output (IO) control signal to be set for each fuzzy rule is made by the QIEA. It initializes randomly a population of complete real valued strings. Each of these real values strings is then decoded into fuzzy rules as well as MFs and evaluated by a FLC. Each real valued string's fitness is defined as the error between the state of the system and the target set. The QIEA procedure includes some steps called selection, crossover and mutation and it is run for a defined number of iterations called generations. In each generation, QIEA proceeds according to the fitness values produced by the FLC for each real valued-string. The QIEA uses its evolutionary operators to perform a self-directed search, learning to look for better and better fuzzy rules and optimal MFs parameters. A mechanism for acceptance of a set of fuzzy control rules generated by the QIEA is needed. We only accept a set of fuzzy rules with generation ρ , where ρ is a specified number. The overall quantum inspired evolutionary procedure to adapt the FLCs is shown in Fig.6.

C1. Representation and Evaluation of Rule base in QIEA

In QIEA, population consists of real coded triploid chromosomes [12] where each of the chromosomes is defined based on the fuzzy rules and MFs parameters. So, the chromosome for this control system can be defined as follows:

$$\begin{pmatrix} R_1 \cdots R_i \cdots R_n \\ \alpha_1 \cdots \alpha_i \cdots \alpha_n \\ \beta_1 \cdots \beta_i \cdots \beta_n \end{pmatrix} \tag{10}$$

where$(R_i\ \alpha_i\ \beta_i)^T$, i =1, 2…. n is the ith allele of real-coded triploid chromosome,R_i is the ith rule in the fuzzy rule base defined above. A pair of probability amplitudes of one qu-bit is $(\alpha_i, \beta_i)^T$ which must satisfy the normalization condition $|\alpha_i|^2+|\beta_i|^2=1$. Here, n is the length of real-coded triploid chromosome which is 35 for this fuzzy controller. So, each allele of the chromosome can be written with eight values as$(\ C_{x,i}w_{x,i}\ C_{\phi,i}w_{\phi,i}C_{\theta,i}w_{\theta,i}\alpha_i\beta_i\ )^T$. At each generation t QIEA maintains a population of real- coded triploid chromosome $P^t=\{p_1^t,\cdots,p_j^t,\cdots,p_N^t\}$ and $\{f_1^t,\cdots,f_j^t,\cdots,f_N^t\}$ describes the corresponding fitness values of individuals where N is the size of population and p_j^t is an individual defined in (10). A rule $R_{j,i}^t$ in p_j^tcontains six values, center (C) and width for each of three variables (x, ϕ and θ). In QIEA, evolutionary operators (mutation and crossover) are applied to make change in the population having fuzzy rules which helps to adjust the FLCs to back up the truck more perfectly. The FLC evaluate a fitness value for each chromosome then update to the population is done based on these fitness values. The fitness value for each chromosome is defined as the trajectory error which is defined as follows:

Trajectory error/fitness =

$$\frac{Length\ of\ truck\ trajectory}{Distance(\ initial\ position,\ \ desired\ final\ position\ )} \tag{11}$$

C2.Mutation

Mutation operator is applied to i^{th} allele selected randomly from p_j^t. The rule of that allele is updated by using gaussian mutation where centers (C) of inputs and output variables are changed as follows:

$$C^{t+1} = C^t + (\ Max - Min)\ N(0,(\sigma_{j,i}^t)^2) \tag{12}$$

Where Max and Min are respectively upper and lower bound of the regions in whichC^t lies. The value of variance $(\sigma_{j,i}^t)^2$ is either $|\alpha_{j,i}^t|^2$ or $|\beta_{j,i}^t|^2/5$ subject to the "Fine Search" or "Coarse Search" to be implemented respectively [15]. The value of C^{t+1} may not remain in the region of C^t then it is clipped into that region as follows:

$$C^{t+1}=\begin{cases} 2*Max - C^{t+1},\ if\ C^{t+1} > Max \\ 2*Min - C^{t+1},\ if\ C^{t+1} < Min \end{cases} \tag{13}$$

The center of θ is not kept in a particular region rather its range is considered always the whole range of θ. The value of variance $(\sigma_{j,i}^t)^2$ is either $|\alpha_{j,i}^t|^2$ or $|\beta_{j,i}^t|^2/5$ subject to "Fine Search" or "Coarse Search" to be implemented [16]. Now the half-width (W^t) of each center is updated as follows:

$$W^t = \begin{cases} r*(Max - C^{t+1})\ \ if C^{t+1} > (Max + Min)/2 \\ \quad r*(C^{t+1} - Min)\ \ Otherwise \end{cases} \tag{14}$$

where r is the uniformly distributed random number in the range [0, 1]. The pair probability amplitudes of the ith allele are updated by the Rotation gate as follows:

$$\begin{pmatrix} \alpha_{j,i}^{t+1} \\ \beta_{j,i}^{t+1} \end{pmatrix} = \begin{pmatrix} \cos(\Delta\Theta_{j,i}^t) & -\sin(\Delta\Theta_{j,i}^t) \\ \sin(\Delta\Theta_{j,i}^t) & \cos(\Delta\Theta_{j,i}^t) \end{pmatrix} \begin{pmatrix} \alpha_{j,i}^t \\ \beta_{j,i}^t \end{pmatrix} \tag{15}$$

where rotation angle $\Delta\Theta_{j,i}^t$ of qubit is calculated as follows:

$$\Delta\Theta_{j,i}^t = sgn(\alpha_{j,i}^t\beta_{j,i}^t)\Theta_0\ \exp\left(\frac{|\beta_{j,i}^t|}{|\alpha_{j,i}^t| + \gamma}\right) \tag{16}$$

Where Θ_0 is the initial rotation angle, γ is the scale parameter, Θ_0 and γ control the value of the rotation angle together and have an effect on the speed of convergence, the sign *sgn* (.) determines the direction of the rotation angle.

C3. Discrete Crossover (DC) and Elitism

To backing up the truck with minimized trajectory error (fitness value) the FLC have to search for the suitable steering angle θ with respect to input variables. Discrete crossover is used here to expand the search space for the FLCs. The elitism technique is used to select the individual with minimized fitness value which also ensure that the rule base with best fitness value will not be lost.

TABLE III.
PARAMETERS OF RULE SET AND MFS (OPTIMAL SOLUTION)

	Inputs				Output	
	x		ϕ		θ	
	Center	Width	Center	Width	Center	Width
R-1	15.7297	31.1521	-38.782	51.2173	5.91529	8.0847
R-2	5.2189	20.4265	2.7657	70.766	0.103335	29.8967
R-3	28.9555	38.6555	154.72	115.28	9.03497	4.965
.	.	.	.	.	.	.
R-34	75.0726	24.9274	11.7632	101.763	-16.6113	13.3887
R-35	79.655	20.345	74.4489	164.449	-7.1172	16.6828

V. EXPERIMENTAL RESULTS AND COMPARATIVE ANALYSIS

A. Experimental Results

To evaluate the accuracy of the proposed system, we have carried out a series of experiments in which the controllers were evolved in our simulated arena. In this paper fuzzy control rules and MFs are adjusted based on QIEA which enables the FLCs to backing up the truck with minimum trajectory error (fitness value). The final control rules and MFs are obtained as shown in, Table IV and Fig. 7 respectively after adjusting the fuzzy control rules and MFs.

For the 100×100 loading dock, the ranges of input and output variables are considered as follows [14]

$$\begin{cases} 0 \le x \le 100 \\ -90 \le \phi \le 270 \\ -30 \le \theta \le 30 \end{cases} \quad (17)$$

TABLE IV.
RULE SET FOR THE FLC

		ϕ						
		LB	LU	LV	VE	RV	RU	RB
x	LE	PS	ZE	PM	NM	NB	NB	PS
	LC	PM	PS	NS	NM	NM	ZE	NB
	CE	PM	PM	PS	ZE	NS	NM	NM
	RC	PB	PB	PM	PM	PS	NS	NM
	RE	NS	PB	PB	PM	PM	NM	NS

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 [-90, 0], [0, 66.5], [66.5, 86], [86.5, 94.5], [94.5, 113.5], [113.5, 182.5], [182.5, 270] then they are represented by seven linguistic variables LB, LU, LV, VE, RV, RU and RB respectively. Similarly seven divided regions of the range of θ, [-30, -20], [-20, -7.5], [-7.5, -2.5], [-2.5, 2.5], [2.5, 7.5], [7.5, 20], [20, 30] are represented by variables NB, NM, NS, ZE, PS, PM and PB. Considering these regions of input-output variables, a population of real-coded chromosomes containing fuzzy rules is generated and then it is updated by QIEA to provide a best rule base for the FLCs.

The QIEA runs for 5 times with number of generation ρ = 100, population size =10, scale parameter $\gamma = 5$ and initial rotation angle $\Theta_0 = 0.1\pi$. At the beginning of the simulation, the antecedent and consequent part of each fuzzy control rule set cells are empty. To get the optimal parameter of antecedent and consequent part of each rule and MFs parameters from the developed QIEA generator block, we start the simulation from twenty initial positions (x_0, y_0, ϕ_0): (25, 30, and 60^0), (80, 30, 150^0)..etc. The initial positions are randomly selected and cover the simulated backing up the truck loading zone.

For all the given initial positions the QIEA converges to some sets of fuzzy control rules. During the simulation, we generate twenty sets of values associated to fuzzy rules from twenty different initial positions. After translating the parameter values into sets of rules as shown in Table IV using fuzzy amalgamation (FA) methodology [18]. The FA method starts with an empty new rule set. Then it compares and combines two rule sets as follows. If the corresponding cells of two rule sets are same then add the cell entries into the corresponding cell of the new rule set. On the other hand, if corresponding cells of two rule sets are different, then an average of the two different entries is computed using the following formula:

$$Numeric\_Value(NV) = \frac{1}{n}\sum_{i=1}^{n} N_i \quad (18)$$

where n is the number of combining tables (here n = 2), N_i is the numeric values (center) of fuzzy rule. Find the linguistic term (fuzzy set) for NV where it has the highest membership value, and return that linguistic term as a result of two different entries into the corresponding cell of the new rule set.

TABLE V.
IF-THEN ASSOCIATION FROM INPUTS TO OUTPUT

	QIEA - Optimal Chromosome					
	Premise				Conclusions	
		Input x	Operator	Input (ϕ)		Output (θ)
R-1	If	X = LE	and	ϕ = LB	Then	θ = PS
R-2	If	X = LE	and	ϕ = LU	Then	θ =ZE
R-5	If	X = LE	and	ϕ = LV	Then	θ = PM
.	.	.	.	.	.	.
R-35	If	X = RE	and	ϕ = RB	Then	θ = NS

There are 35 fuzzy rules (fuzzy rule base) forming a 5×7 tables with rows to hold the corresponding actions (outputs) as shown in Table IV.

Table III shows the numerical values of one of the rule base generated by QIEA. The FLCs use these fuzzy rules to backing up the truck. For each rule the FLCs generate the steering angle θ from the current position of the truck represented by two input variables x and ϕ by if-then association which is shown in Table V.

The triangular MFs (Fig.7) shapes are used for simpler calculation and better description of the problem depending on the automatic knowledge acquisition.

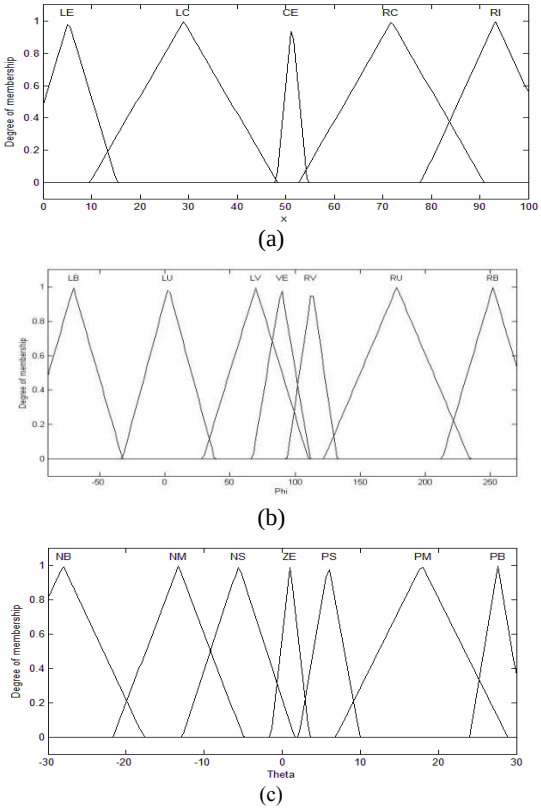


Figure 7. Fuzzy MFs Triangular shape for each linguistic fuzzy set value of the optimal solution (a) Input x (b) Input ϕ (c) Output θ

The MFs and rule set are design tools that give opportunity to model a control surface, a convenient way to examine a two input/one-output control strategy and controller properties. It is obvious that using these attributes one can more precisely fulfil a quality criterion in a full operational range. The control surface shows the input out relationship visually and it also shows the degree of nonlinearity between input and output of the controller. The control surface is defined with 35 rules as shown in Fig.8.

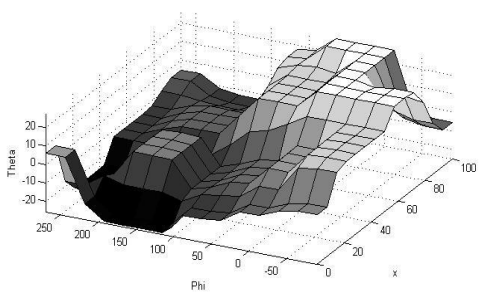


Figure 8. Control surface generated by applied FLC

With the above rule set (shown in Table IV), the QIEA-fuzzy controller produces successful backing up a truck trajectories starting from any given initial condition. Figure 9 shows a typical example of fuzzy backing up truck trajectories. In each iteration, all rules in rule base are not contributing to the control result within an inference engine. In most cases the fuzzy control system uses (fire) only one or two rules in the rule base at each iteration.

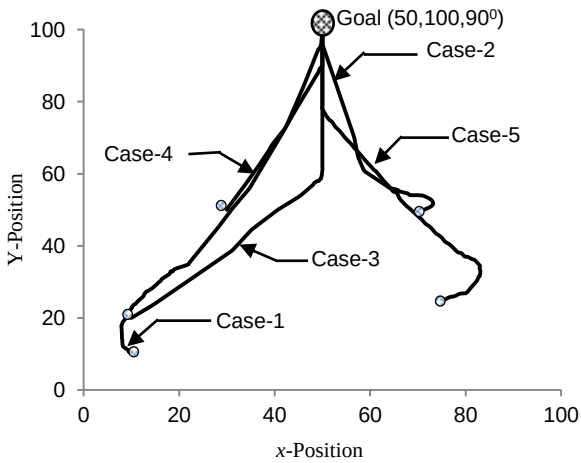


Figure 9. Truck trajectories using triangular fuzzifier and COG for different initial positions

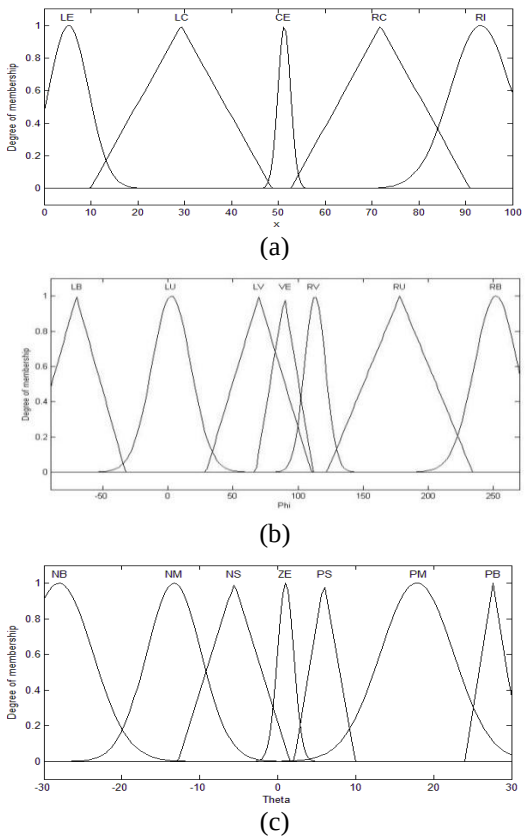


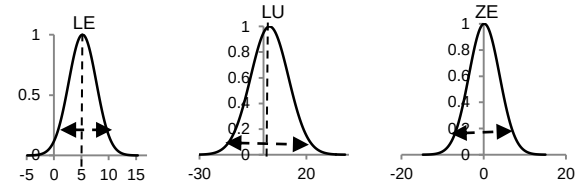
Figure 10. Fuzzy MFs Triangular and Gaussian shape for each linguistic fuzzy set value (a) Input x (b) Input ϕ (c) Output θ

For any initial state (x_0, y_0, ϕ_0) is given to back up the truck to hit the loading dock at right angle $\phi_f = 90^0$

and final position at (x_f, y_f) = (50, 100). The desired trajectory can be that the truck moves along the target line or coincide with the target line and then goes along *y*-axis to arrive at the goal.

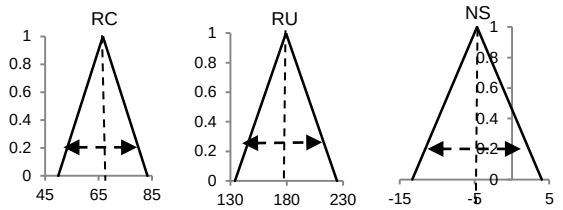
B. Comparative Analysis of FLCs response

In this section, we investigate the response of proposed FLCs based on the MCOG and different fuzzifiers methods. We have compared the performance of the evolved FLC against FLCs using different fuzzifiers as fuzzification method and MCOG as a defizzification strategy.



Rule 2: If x =LE and ϕ =LU Then θ =ZE

(a)



Rule 27: If x=RC and ϕ =RU Then θ =NS

(b)

Figure 11. MF form of fuzzy rule (a) Gaussian (b) Triangular

Figures 10 (a), (b), and (c) show the two antecedents and one consequent MFs with different fuzzifiers (triangular and gaussian). Figure 11 shows the two fuzzy control rules one is triangular and another one is gaussian.

The design of an appropriate defuzzication strategy is important as it will affect the interpretation of the FLCs response. The COG uses only the consequent MFs centers, without any information concerning their shapes (narrow/wide). But the spread of the consequent MFs influence the response of FLCs. In our proposed approach, width is used to measure the shape of consequent MFs. In order to improve the performance of FLC, modified defuzzification strategy associated with the shape of MFs is used**.**

Results obtained using MCOG and different fuzzifiers show that parking durations are shorter than those obtained using COG defuzzification strategy under the same initial conditions.

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

Case		1		2		3		4		5	
x	y	10	10	70	50	10	20	30	50	75	25
ϕ		180^0		-45^0		60^0		90^0		-30^0	
COG and single fuzzifier		57		47		37		28		52	
MCOG and two fuzzifier		51		42		31		24		48	

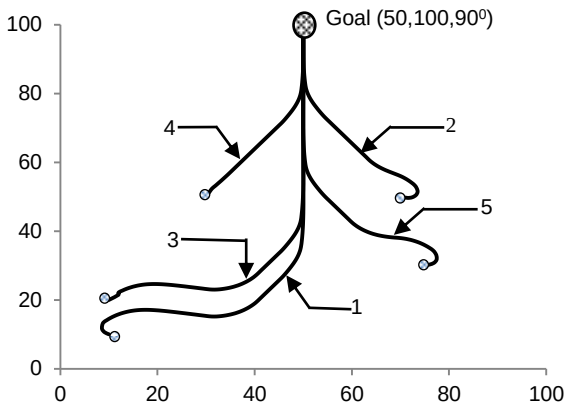
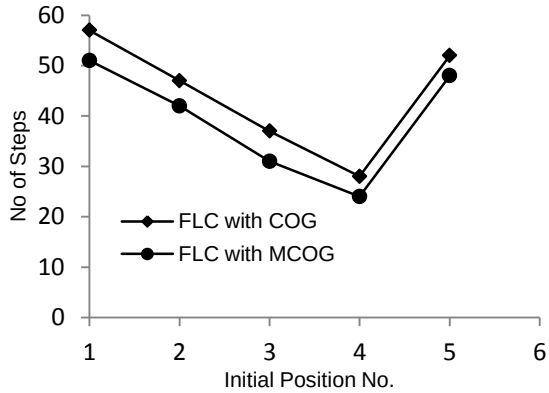
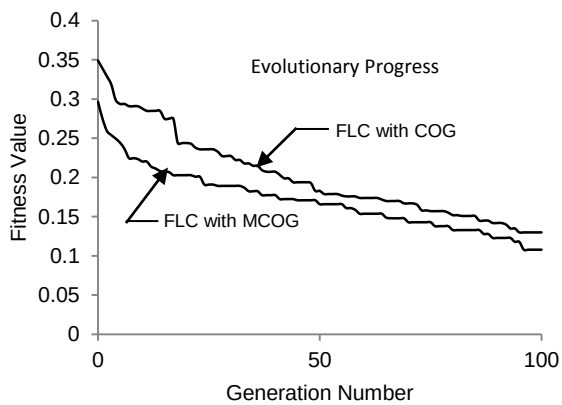


Figure 12. Truck trajectories using different fuzzifier and MCOG for different initial positions

Figure 13 (a) shows the required times for the backing up a truck to reach the goal position in 5 different initial conditions and their trajectories are plotted in Fig.12 & 9. Table VI shows the five initial conditions for (x, y, ϕ) with their steps. From Figure 9, 12, 13 and table VI, it is obvious that the performances of FLC with MCOG and different fuzzifiers are better than those in FLC with COG and single fuzzifier. It not only takes less steps to arrive the goal position using MCOG, but also it shows the smoother trajectories (shown in Fig.12 & 9).



(a)



(b)

Figure 13. (a) Show the total steps to reach goal position by 5 different initial cases and (b) Average trajectory errors or fitness values for different initial positions

We found that the QIEA based strategy evolves to optimal set of fuzzy control rules and MFs after only some generations. The evolutionary progress (Fig. 13(b)) displays the performance of the best chromosome found so far against the number of generations.

C. Comparison between Genetic-Fuzzy System and our proposed approach

In this section, the performance is compared with genetic fuzzy system in order to check the validity of our proposed quantum inspired evolutionary fuzzy approach.

A FLC using complementary MFs (C-MF) has been designed in the similar way to our approach but only centers of the MFs are adjusted through GA.

The number of steps of trajectories of the truck from given positions to back up to the loading zone controlled by the genetic fuzzy system and the quantum inspired fuzzy system are given in Table VII and Fig.14 (graphical trajectories) for the two cases. The number of required trajectory steps for genetic fuzzy system is larger than our approach. Figure 14 and table VII, it is obvious that the performances of quantum inspired fuzzy are better than the genetic fuzzy system. It not only takes fewer steps to arrive the goal position, but also it shows the smoother trajectories. If the RB and shape of MF is rich enough then the simulated truck backs up to the loading zone successfully. In this case, QIEA generates richer rule base and shape of MF than the GA.

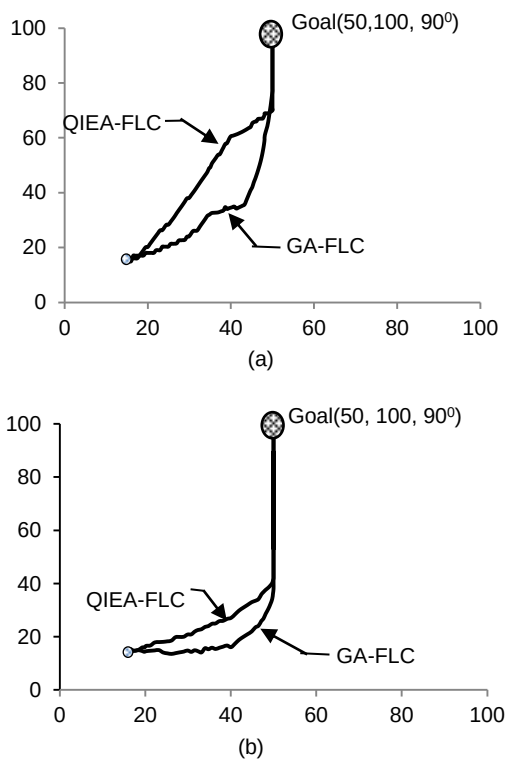


Figure 14. Comparison of sample truck trajectories of the genetic-fuzzy and QIEA-Fuzzy for two initial cases (x_0, y_0, ϕ_0):

(a) $(15.5, 15, 0^0)$ (b) $(15.5, 15, -30^0)$

TABLE VII.
TRAJECTORIES STEPS COMPARISON CONTROLLED BY THE GA-FLC AND QIEA-FLC

		Genetic-fuzzy System	Proposed Approach
Case	Initial Position	No. of Steps	
1	$(15.5, 15, 0^0)$	65	58
2	$(15.5, 15, -30^0)$	73	61

D. Sensitivity Analysis of FLC response

We have shown that the performance of the proposed fuzzy controller is varied according to the types of fuzzifier and MCOG. The question on why different type of fuzzifier gives different results, would be also interesting and to be analyzed with respect to the fuzzification and defuzzification process.

C1.Fuzzification

Fuzzification of a real-valued variable is done with intuition, experience and analysis of the set of rules and their associated MFs, i.e., fuzzy sets. The two types of MFs defined in the equation (3) and (4) produce different fuzzified value for the same real value as shown in Fig. 15.

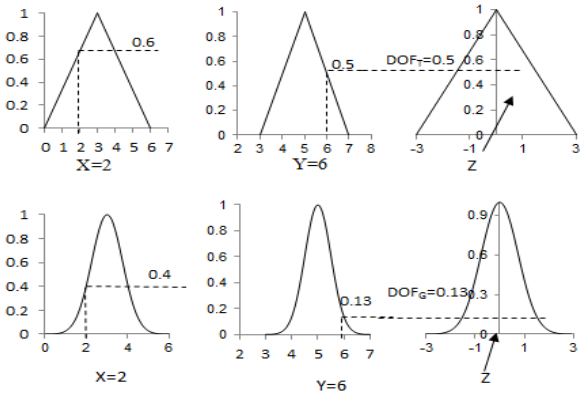


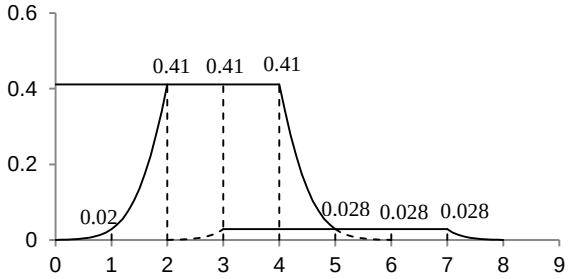
Figure 15. Two different kinds of fuzziness degrees (depends on fuzzifier) for a crisp value X=2 and Y=6

The fuzzification of the input variable should be realistic. Experience and different procedures as well as use different types of fuzzifier should be followed while designing a large fuzzy system for the realistic and accurate output. The wrong fuzzification of the input variable(s) might cause instability and error in the system.

C2. Defuzzification

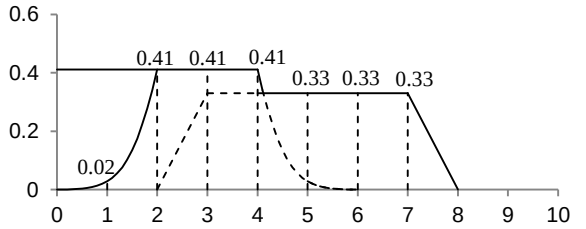
The following example shows FLC output varies according to the type of MFs.

Example: Consider a fuzzy controller with one output variable, and suppose that a given input datum excites two rules of the rule base. Now defuzzification process would yield an output value according to the MF types associated to the linguistic output labels of the two rules. In order to explicate this fact, we consider two cases; triangular and gaussian MFs. Figure 16 shows that each case results in different output values. It is easily conceivable that some value would be better than the others. Thus we need a way to define different member function types across the rule base. Some rule may prefer a particular type of MF than the others so as to make an optimal final decision.



(a) Gaussian MFs for FLC

$$x_{n+1} = \frac{1*0.028+2*0.41+3*0.41+4*0.41+5*0.028+6*0.028+7*0.028}{0.028+0.41+0.41+0.41+0.028+0.028+0.028}$$
$$= 3.14$$



$$x_{n+1} = \frac{1*0.028+2*0.41+3*0.41+4*0.41+5*0.33+6*0.33+7*0.33}{0.028+0.41+0.41+0.41+0.33+0.33+0.33}$$
$$= 4.31$$

(b)Mixed MFs for FLC

Figure 16. Effect of defuzzification process for selecting different types of fuzzifiers

The value of fuzzified and defuzzified may vary according to the fuzzifier. In this study, the effect of the different fuzzification and defuzzification process to construct the fuzzy model is investigated along with the selection and definition of fuzzy rules and the shapes of MFs.

VI. CONCLUSIONS

The main goal of this paper was to find out a way to design FLCs on the basis of QIEA. With the experimental results we have shown that our proposed approach is a self-learning adaptive method to acquire automatic fuzzy knowledge base i.e., selection and definition of fuzzy understandable and reliable fuzzy if-then control rules as well as the shapes of MFs to design FLCs without any prior knowledge. Here the FLCs works with biological evaluation of the rule bases to backing up a truck in the loading dock with minimum trajectory error. On the other hand, we have shown that different fuzzification and MCOG method has a great impact on the response of the FLCs. In this way, QIEA can build quickly an effective controller relatively. This technique may lead to an increase in the use of FLCs as the previously time-consuming design procedure can be reduced dramatically.

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 or fixed obstacle, the path planning problem for multiple mobile robots with more than one obstacles either moving or fixed in the workspace. Moreover, this works also to be extended in the analysis of the behaviour of the cooperative quantum inspired evolutionary learning proposal with high dimensional problems, where it might be necessary to include a variable feature selection component into the learning approach.

REFERENCES

[1] Y. F. Li and C. C. Lan,"Development of fuzzy algorithms for servo systems", *IEEE Control Systems Magazine*, Vol. 9, no. 3, pp. 65-73, 1989.

[2] G. W. Evans, W. Karwowski, and M. R. Wilhelm, "Application of the Fuzzy Set Methodologies in Industrial Engineering," *Elsevier*, pp. 336-342, August 1989.

[3] T. Yamakawa, and T. Koga, "Bio-inspired self-organizing relationship network as knowledge acquisition tool and fuzzy inference engine", Proc. of *IEEE world conference on Computational intelligence*, pp. 159-180, 2008.

[4] K. Belarbi, F. Titel, W. Bourebia, and K. Benmahammed, "Design of Mamdani fuzzy logic controllers with rule base minimisation using genetic algorithm," *Journal of Engineering Applications of Artificial Intelligence*, Vol. 18, pp. 875–880, 2005.

[5] A. Arslan and M. Kaya, "Determination of fuzzy logic membership functions using genetic algorithms," *Fuzzy Sets and Systems*, vol. 118, pp.297–306, 2001.

[6] D. L. Meredith, C. L. Karr, and K. Krishna Kumar, "The use of genetic algorithms in the design of fuzzy logic controllers," *In proc. Third Workshop on Neural Networks:Academic/Industrial/Defence* (WNN'92), pp. 1105-1109, 1992.

[7] A. Bagis, "Fuzzy rule base design using tabu search algorithm for nonlinear system modeling," *ISA Transactions*, vol. 47, pp. 32–44, 2008.

[8] S. Bai and S Chen, "Automatically constructing grade membership functions of fuzzy rules for students' evaluation," *Journal of Expert Systems with Applications*, vol. 35, pp. 1408–1414, 2008.

[9] C. C. Yang and N. K. Bose, "Generating fuzzy membership function with self-organizing feature map," *Pattern Recognition Letters*, vol. 27, pp. 356–365, 2006.

[10] A. F. M. Huang, S .J .H. Yang, M. Wang, and J. J.P. Tsai, "Improving fuzzy knowledge integration with particle swarm optimization," *Expert Systems with Applications*, Vol. 37, pp. 8770–8783, 2010.

[11 L.A. Zadeh, "Fuzzy sets," *Information and Control* Vol. 8 No. 3, pp. 338–353, 1965.

[12] L. Ying., And L. Yuanchun, "Neural-fuzzy control of truck backer-upper system using a clustering method", *Journal of Neurocomputing*, vol. 70, pp. 680-688, January 2007.

[13] Do-Hyeon Kim, Kwang-Baek Kim and Eui-Young Cha, "Fuzzy truck control scheme for obstacle avoidance", *Journal of Neural Computing and applications*, Vol. 18, no. 7, pp. 801-811, October 2009.

[14] J. S. Chang, J. H. Lin, and T.D. Chiueh, "Neural networks for truck backer-upper control system," *In Proc. of the international IEEE/IAS conference on industrial automation and control: emerging technologies*, Taiwan, pp. 328–334, May 1995.

[15] R. Zhang and H. Gao. "Real -coded quantum evolutionary algorithm for complex function with high dimension," *In Proc. of IEEE International Conference on Mechatronics and Automation*, China, pp. 2974-2979, August 2007.

[16] K. H. Han and J. H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 6, pp. 580-593, 2002.

[17] D. Wu, and W. W. Tan, "Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers," *Journal of Engineering Applications of Artificial Intelligence*, vol. 19, no. 18, pp. 829–841, December 2006.

[18] M. Mohammadian and R. J. Stonier, "Generating fuzzy rules by genetic algorithms", *Proc. of 3rd Int. workshop on Robot and Human Communication*, Nagoya, Japan, pp. 362-367, 1994.

[19] A. Bagis, "Fuzzy and PD controller based intelligent control of spillway gates of dams," *Journal of Intelligent Fuzzy Systems*, Vol. 14, No. 1, pp. 25–36, 2003.

[20] K. H. Han and J. H. Kim, "Quantum inspired evolutionary algorithms with a new termination criterion, He gate, and two-phase scheme," *IEEE Trans. on Evolutionary Computation*, Vol. 8, No. 2, pp. 156-169, 2004.

[21] C. K. Chiang, H. Y. Chung, and J. J. Lin, "A self-learning fuzzy logic controller using genetic algorithms with reinforcements", *IEEE Trans. fuzzy systems*, Vol. 5, No. 3, pp. 460-467, August 1997.

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.

Md. Amjad Hossain received the B.Sc. Engg. degree (with honors) in Computer Science and Engineering (CSE) from Khulna University of Engineering & Technology(KUET), Khulna - 9203, Bangladesh, 2008. He is currently serving as a Lecturer in Department of CSE, KUET. He has published several research papers in some reputed Journal and Conference. His current research interests include evolutionary computation, mobile computing and fuzzy Logic.

Md. Kowsar Hossain received the B.Sc. Engg. degree (with honors) in Computer Science and Engineering (CSE) from Khulna University of Engineering & Technology (KUET), Khulna - 9203, Bangladesh, in 2008. He is currently serving as a Lecturer in the Department of CSE of the same University. . He has published several research papers in some reputed Journal and Conference. His current research interests include evolutionary computation, mobile computing, and fuzzy logic.

Md. Faijul Amin received the B.Sc. Engg. degree (with honors) in Computer Science and Engineering (CSE) from Bangladesh University of Engineering & Technology (BUET), Dhaka, Bangladesh, in 2004. He is currently serving as an Assistant Professor in the Department of CSE at Khulna University of Engineering & Technology (KUET). He has published several research papers in some reputed Journal and Conference. His current research interests include complex valued neural network, evolutionary computation, and fuzzy logic.

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).