

An Adaptive Fuzzy Logic Controller based on Real Coded Quantum-Inspired Evolutionary Algorithm

Pintu Chandra Shill[†], Md. Amjad Hossain[‡], Md. Faijul Amin[†], Kazuyuki Murase[†]

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

[‡] Khulna University of Engineering and Technology, Khulna, Bangladesh

{pintu,amin,murase}@synapse.his.u-fukui.ac.jp, amjad_kuet@yahoo.com

Abstract—In this paper, fuzzy logic control systems and real coded quantum inspired evolutionary algorithm (RCQIEA) are integrated for intelligent control. Here, RCQIEAs is utilized as an adaptive method for selection and definition of fuzzy control rules and for tuning the parameters of membership function for each fuzzy control rule in two different ways. The majority of fuzzy logic controllers (FLCs) to date are working based on the expert knowledge base derived from heuristic knowledge of experienced operators. These approaches are difficult and time consuming for experts. Moreover, because manual coded FLCs use expert knowledge, there is no guarantee that the FLCs obtained will have sufficiently good performance, especially for a complex system problem with a large number of input variables. On the contrary, our proposed approach is an automatic knowledge acquisition learning method for generating or adapting FLCs using RCQIEA. In order to check the effectiveness of our proposed approach, it has been applied to solve the truck-and-trailer controller, which is well known test-bed for fuzzy control systems. The fuzzy controller obtained by the proposed approach performs better and effectively realizes the trajectory control of the truck with trailer.

Keywords—Fuzzy Logic Controller; Fuzzy Rule Base; Real Coded Quantum-Inspired Evolutionary Algorithm; Optimization; Truck-and-Trailer controller

I. INTRODUCTION

Fuzzy rule-based model is used to solve a wide range of problems providing flexible solutions. A fuzzy rule-based model is a set of fuzzy rules and the associated membership functions (MFs) that maps inputs to outputs. Fuzzy rules 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 slow as it relies on trial and error rather than a guided approach.

To surmount this shortcoming, real coded quantum inspired evolutionary algorithm (RCQIEA) can be considered as a powerful tool to find out efficient fuzzy rule sets [1], and tuning of MFs parameters. 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 MFs.

Because of this, the MFs optimization problems can be regarded as 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 genetic algorithms (GAs), neural networks (NNs), self-organizing feature map (SOFM), tabu search (TS), 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 [2] in fuzzy rule base (RB) minimization. Belarbi 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. Kim [1] presented a method to find out efficient fuzzy rule sets using quantum inspired evolutionary. They design a fuzzy logic control system for path planning of mobile robot. Meredith [3] also applies GA to the fine tuning of MFs in a FLC for a helicopter. Bagis [4] have presented an approach for the determination of the structure and parameters of fuzzy RB. He applied this approach in the modeling of the nonlinear or complex systems. Bai and Chen [5] proposed an automatic method for a 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 [6] has 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 MF according to the clusters in the first step. They applied this method to pattern recognition. Huang [7] has presented a fuzzy knowledge integration technique based on the PSO. Their proposed approach consists of two phases: Firstly, it encodes the fuzzy rule sets and fuzzy sets with its corresponding MFs. Secondly, the particle swarm algorithm is used to explore the fuzzy rule sets, fuzzy sets and MFs to its optimal or the approximately optimal extent. Koga, et. al. [8] has presented a method, the self-organizing relationship (SOR) network with fuzzy inference based evaluation for backing up a truck-and-trailer controller. They use SOR network to establish desired I/O (Rule base) relationship of a target system.

In a classical evolutionary algorithm like GA, information is stored as classical binary (0 or 1) format. So information stored in a classical register represents a single state. But in quantum inspired evolutionary algorithm, smallest unit of information is called Q-bit which may be in '0'state or '1'

state or any superposition state of them. Superposition of logical state can be expressed as “ $\alpha|0\rangle + \beta|1\rangle$ ”. Another way of writing “superposition” is as a vector:

$$\alpha|0\rangle + \beta|1\rangle \leftrightarrow \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

The complex number α and β are called the “amplitudes” of the superposition. $|\alpha_i|^2$ gives the probability that the Q-bit will be found in “0” state and $|\beta_i|^2$ gives the probability that the Q-bit will be found in “1” and they satisfy the normalization condition $|\alpha_i|^2 + |\beta_i|^2 = 1$. A Q-bit individual as a string of m Q-bits is defined as:

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix}$$

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

Quantum inspired evolutionary algorithms has a better characteristic of population diversity than others, since it can represent linear superposition of states probabilistically. But higher value of m needs higher computing time of the algorithm. For an optimization problem with fitness function $f(x_1, x_2, \dots, x_n)$, if we represent each real variable x_i by k bits then the total number of Q-bits in an Q-bit individual $m = k \times n$. So if we apply the evolutionary operators on huge number of bits then they will definitely take a significant amount of time to execute.

In this paper, FLCs and RCQIEA are integrated for the intelligent control of a truck-and-trailer controller problem. RCQIEA is employed as an adaptive method for selection and definition of fuzzy control rules of the FLC as well as shape of MFs. It is found that our proposed integrated architecture is able to generate comprehensible and reliable fuzzy rules with appropriate shape of MFs by a self-learning adaptive method.

The organization of this paper is as follows: In Section II, we review the literature relevant to the FLC. In the third section we explain the problem truck-and-trailer controller in a greater detail. In section IV, we provide the integrated FLCs and RCQIEA architecture and also show how it can be used to solve highly nonlinear complex problems. We also introduce the basic principle of RCQIEAs and discuss how they can be used to generate a satisfactory fuzzy rule base and select appropriate shape of MFs simultaneously for FLCs. Section V presents our simulation results. Finally, section VI provides some concluding remarks and future directions.

II. FUZZY LOGIC CONTROLLER

The idea of fuzzy logic was first introduced in 1960s by Professor Lofti Zadeh [9]. The general configuration of FLC, can be divided into four main parts; fuzzification interface, a rule base, an inference mechanism and defuzzification interface (Fig.1).

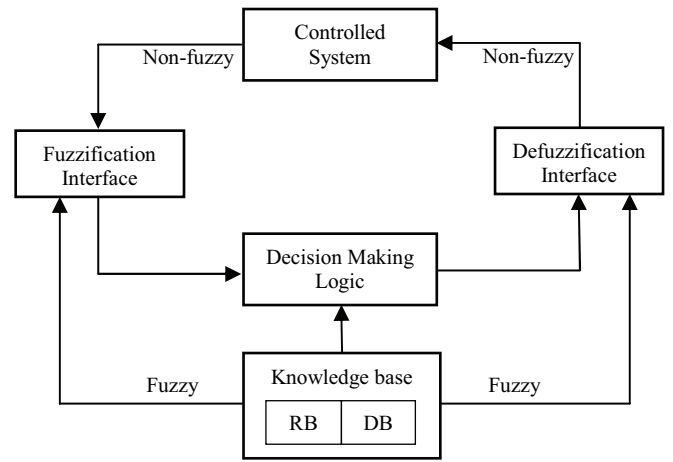


Figure 1. Fuzzy Logic Controller (FLC)

A. Fuzzifier

Fuzzy sets 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 $A \in U(\text{universe})$.

B. Fuzzy rule base

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

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

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

where $x_1, x_2, \dots, x_n$, and y are the input and output linguistic variables respectively. F_i^k and G^k are type-1 fuzzy sets in input sets $X \in X_1 \times X_2 \times \dots \times X_n$ and output sets Y .

B. 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 $(x_1, x_2, \dots, x_n)^T$. This relation R^l is the actual process of mapping from fuzzy sets in X to fuzzy sets in Y , $F_1^l \times F_2^l \times \dots \times F_n^l \rightarrow G^l$, can be called fuzzy inference process.

C. Modified Height Defuzzifier

The modified height defuzzifier was used to handle the consequent uncertainty [10, 11], The expression generally used for a fuzzy set is as follows:

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

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

III. PROBLEM STATEMENT

A. Model of Backing up a Truck-and-Trailer

The task is to back up a simulated truck-and-trailer to a loading dock in planner parking lot, depicted in Fig. 2. The four state variables ϕ_t, ϕ_c, x and y exactly determine the truck-and-trailer position. The angle ϕ_t specifies the angle of the trailer with the horizontal and ϕ_c is the relative angle of the cab with the trailer. The coordinate pair (x, y) specifies the position of the rear center of the trailer in the plane. The goal is to make the trailer arrive at the loading dock at a right angle ($\phi_t = 90^\circ$) and to align the position (x, y) of the trailer with the desired loading dock (x_f, y_f) and the truck is allowed to move backward by some fixed distance b at every stage. The loading zone corresponded to the plane $[0, 0] \times [100, 100]$ and (x_f, y_f) equaled $(50, 100)$. At every stage our proposed approach should produce the output angle β to update the θ and angle ϕ_t of the trailer with horizontal that backs up the trailer to the loading dock from any initial position. The truck-and- trailer only moves backward with fixed distance. The variable ranges were as follows:

$$\begin{aligned} -100 &\leq x \leq 100 \\ -90^\circ &\leq \phi_t \leq 270^\circ \\ -90^\circ &\leq \phi_c \leq 90^\circ \\ -30^\circ &\leq \beta \leq 30^\circ \end{aligned}$$

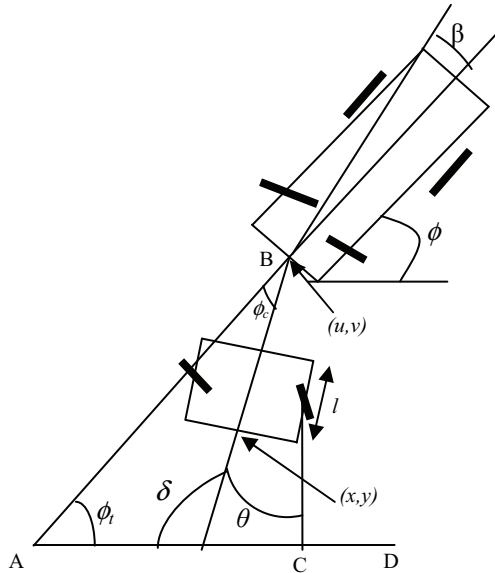


Figure 2. Diagram of truck-and-trailer

The state space of the system is four dimensional. The variable x gives the horizontal position of the midpoint of the rear of the trailer and the variable y gives the vertical position of the midpoint. Fuzzy variable ϕ_t specified the angle of the trailer with the horizontal. The fuzzy variable ϕ_c specified the relative cab angle with respect to the center line along the trailer. The extreme cab angle 90° and -90° corresponds to two “jack knife” positions of the cab with respect to the trailer. Positive ϕ_c value indicated that the cab resided on the left-hand side of the trailer. Negative value indicated that it resides on the right hand side and Fig.2 shows a positive angle value of ϕ_c .

The following inputs and outputs are chosen to represent the system and illustrated in Fig.2.

Three Inputs:

- x , Cartesian x -coordinate of truck rear end.
- ϕ_t , Angle of the trailer with horizontal.
- ϕ_c , Relative angle of the cab with the trailer.

One Output

- β , Angle of the trailer updated at each step.

Fuzzy variable ϕ_t, ϕ_c and x defined the input variables. Fuzzy variable β defined the output variable. The angle that is needed to update the trailer is β . With the output β value computed, the updated angle of the trailer with the horizontal is ϕ_t' :

$$\phi_t' = \phi_t + \beta \quad (1)$$

The trailer position is (x, y) moved to the new position (x', y') :

$$\begin{aligned} x' &= x + r \cos(\phi_t') \\ y' &= y + r \sin(\phi_t') \end{aligned} \quad (2)$$

where r denotes a fixed backing distance. Then the joint of the truck and trailer (u, v) moved to the new position (u', v') :

$$\begin{aligned} u' &= x' - l \cos(\phi_t') \\ v' &= y' - l \sin(\phi_t') \end{aligned} \quad (3)$$

Where l denotes the trailer length.

Again, the steering angle

$$\begin{aligned} \theta &= \phi_{c,h}' - \phi_{c,h} \\ \phi_{c,h}' &= \phi_{c,h} + \theta \end{aligned} \quad (4)$$

Where $\phi_{c,h}'$ denotes the updated cab angle with horizontal and $\phi_{c,h}$ denotes the cab angle with horizontal.

From $\triangle ABC$,

$$180^0 = \phi_c + \phi_t + \delta \quad (5)$$

$$\delta = 180^0 - (\phi_c + \phi_t)$$

And from $\angle BCA$ and $\angle BCD$,

$$\angle BCA + \angle BCD = 180^0$$

$$\delta + \phi_{c,h} = 180^0$$

$$180^0 - (\phi_c + \phi_t) + \phi_{c,h} = 180^0$$

$$\phi_{c,h} = \phi_c + \phi_t \quad (6)$$

$$\text{Let, } \theta = \frac{\beta}{2}$$

From equation (4) and equation (6) the updated relative angle of the cab with the trailer:

$$\phi'_{c,h} = \phi'_c + \phi'_t$$

$$\theta + \phi_{c,h} = \phi'_c + \phi'_t$$

$$\phi'_c = \theta + \phi_{c,h} - \phi'_t$$

$$\phi'_c = -\frac{\beta}{2} + \phi_c + \phi_t - \beta - \phi_t$$

$$\therefore \phi'_c = \phi_c - \frac{3}{2}\beta \quad (7)$$

Then the steering angle value $\theta = \phi'_{c,h} - \phi_{c,h}$ where $\phi_{c,h}$ denotes the cab angle with the horizontal.

IV. INTEGRATION OF FLCs AND RCQIEA

The first step in designing an integrated FLC and RCQIEA, as shown in Fig.3, is to decide which parts of the KB are subject to be optimized by the RCQIEA. The KB of fuzzy system is made of two components: the parameters of the fuzzy RB, constituted of the collection of fuzzy rules as well as the shape of MFs and MFs type associated with the linguistic labels of each fuzzy control rule. In this case, selection and definition of fuzzy control rules as well as the shape of MFs are adapted using RCQIEA.

In RCQIEA, each Q-bit contains directly a real variable x_i along with two probability amplitudes (α_i, β_i) consisting an allele of real coded triploid chromosome. So each real coded triploid chromosome requires only n Q-bits (alleles) and thus reduces the execution time of the evolutionary operators, hence whole algorithm.

In this paper, R_i , a fuzzy rule is similar to the x_i described above. Each R_i consists of several real variables and those will need so many Q-bits if we use quantum inspired evolutionary algorithm. To reduce the execution time we use RCQIEA instead of quantum inspired evolutionary algorithm. Moreover, RCQIEA uses discrete crossover operator which expand the search space providing better exploration and exploitation characteristic of the algorithm.

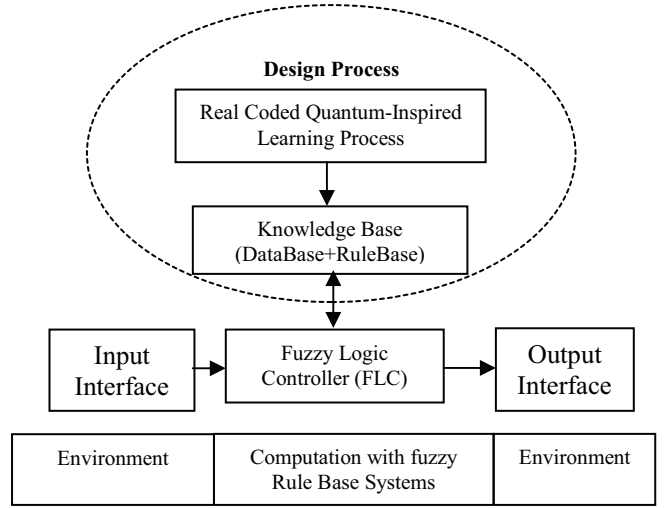


Figure 3. Learning schema of Real Coded Quantum Inspired Evolutionary Algorithm based FLCs

A. Divide the input output variables range

At first identify the variables (inputs, states and outputs) of the problem. Then partition the input and output universe of discourse or the interval spanned by each variable into a number of fuzzy regions (fuzzy sets) and assign a fuzzy MF, i.e., linguistic term of each region. For truck-and-trailer controller divide the state space input x , ϕ_t , and ϕ_c variable space and output β variable space into fuzzy regions and assign a fuzzy MFs of each region. In our running example, we divide the domain for x , ϕ_t , ϕ_c and θ into 5, 7, 3 and 7 regions respectively. The linguistic levels for these regions are defined as of Table 1.

TABLE I. DEFINITION OF LINGUISTIC VARIABLE

x	ϕ_t	ϕ_c	β
LE – Left End LC – Left Center CE – Center RC – Right Center RE – Right End	LB – Left Below LU – Left Upper LV – Left Vertical VE – Vertical RV – Right Vertical RB – Right Below RU – Right Upper	NE-Negative ZR-Zero PO-Positive	NL – Negative Large NM – Negative Medium NS – Negative Small ZE – Zero PS – Positive Small PM – Positive Medium PL – Positive Large

B. Encoding Technique

After defining the regions, encode the input and output spaces fuzzy regions into strings (real value) and the fuzzy rule is written a string (real valued) as depicted in Fig.4.

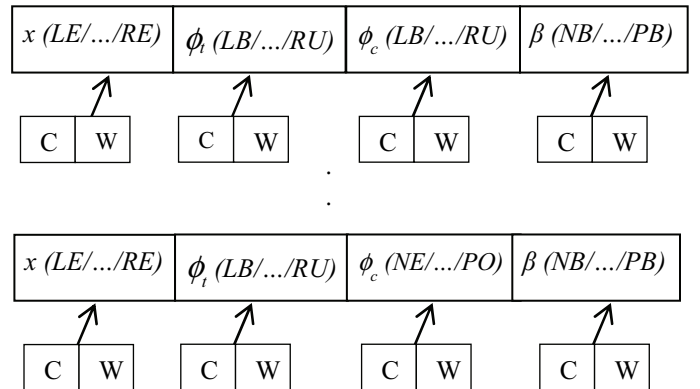


Figure 4. Chromosome structure

C. Real Coded Quantum-Inspired Evolutionary Procedure

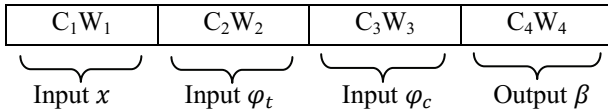
The choice of input-output (IO) control signal to be set for each fuzzy rule is made by the RCQIEA. It initializes randomly a population of complete real valued strings. Each of these real values strings is then decoded into fuzzy rules and evaluated by a FLC. Each real valued string's fitness is defined as the error between the state of the system and the target set. RCQIEA procedures, such as selection, crossover and mutation then proceed according to the fitness values produced by the FLC for each real valued-string. The RCQIEA then performs 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 RCQIEA is needed. We only accept a set of fuzzy rules with generation ρ , where ρ is a specified number. The overall real coded quantum inspired evolutionary procedure to adapt the FLCs is shown in Fig.5.

D. Representation of RB and MFs in RCQIEA

In this paper, we consider RCQIEA where population consists of real coded triploid chromosomes [12]. Each chromosome of RCQIEA population is defined based on the fuzzy rules. So, the chromosome for this control system can be defined as follows:

$$\begin{pmatrix} R_1 \dots R_i \dots R_n \\ \alpha_1 \dots \alpha_i \dots \alpha_n \\ \omega_1 \dots \omega_i \dots \omega_n \end{pmatrix} \quad (8)$$

where $(R_i \alpha_i \omega_i)^T$, $i = 1, 2, \dots, n$ is the i -th allele of real-coded triploid chromosome, R_i is the i -th rule in the fuzzy rule base and it is represented using center (c) and width (w) of input and output variables as follows:



A pair of probability amplitudes of one Q-bit is $(\alpha_i, \omega_i)^T$ which must satisfy the normalization condition $|\alpha_i|^2 + |\omega_i|^2 = 1$. Here, n is the length of real-coded triploid chromosome which is 105 for this fuzzy controller. At each generation t RCQIEA maintains a population of real-coded triploid chromosome $P^t = \{p_1^t, \dots, p_j^t, \dots, p_N^t\}$, where p_j^t is an individual defined in (8). A rule $R_{j,i}$ in p_j^t contains eight values, center (c) and width (w) for each of four variables (x, ϕ_c, ϕ_t and β).

E. Fitness function

An evaluation function is used for evaluating the status of each chromosome or string and consider as an important link between the RCQIEA and the system. The FLCs generates a fitness value for each RB and MFs parameters according to the following evaluation function defined based on the trajectory error:

Trajectory error/fitness =

$$\frac{\text{Length of truck trajectory}}{\text{Distance}(\text{initial position, desired final position})} \quad (9)$$

F. Mutation

In RCQIEA, 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-and-trailer more perfectly. Mutation operator is applied to i^{th} allele selected randomly from p_j^t . The fuzzy rule of that allele is updated by using gaussian mutation where centers for the input and output variables are changed as follows:

$$C^{t+1} = C^t + (Max - Min) N(0, (\sigma_{j,i}^t)^2) \quad (10)$$

Where Max and Min are respectively upper and lower bound of the regions in which the center C^t lies. The value of

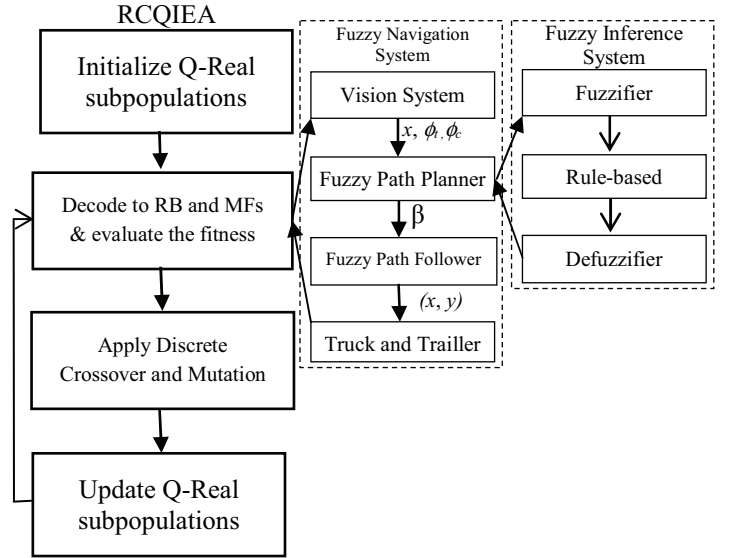


Figure 5. Data flow of proposed RCQIEA with fuzzy navigation system and fuzzy inference system

C^{t+1} may not remain in the region of C^t so it is clipped into that region. The center of β is not kept in a particular region rather its range is considered always the whole range of β . The value of variance $(\sigma_{j,i}^t)^2$ is either $|\alpha_{j,i}^t|^2$ or $|\omega_{j,i}^t|^2/5$ subject to "Fine Search" or "Coarse Search" to be implemented [12]. Now the width (W^{t+1}) of each center is calculated from the new centers of the variables as follows:

$$W^{t+1} = \begin{cases} r * (Max - C^{t+1}) & \text{if } C^{t+1} > (Max + Min)/2 \\ r * (C^{t+1} - Min) & \text{Otherwise} \end{cases} \quad (11)$$

Where r is the uniformly distributed random number in the range $[0, 1]$. The pair probability amplitudes of the i^{th} allele is updated by the rotation gate as follows:

$$\begin{pmatrix} \alpha_{j,i}^{t+1} \\ \omega_{j,i}^{t+1} \end{pmatrix} = \begin{pmatrix} \cos(\Delta\theta_{j,i}^t) & -\sin(\Delta\theta_{j,i}^t) \\ \sin(\Delta\theta_{j,i}^t) & \cos(\Delta\theta_{j,i}^t) \end{pmatrix} \begin{pmatrix} \alpha_{j,i}^t \\ \omega_{j,i}^t \end{pmatrix} \quad (12)$$

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

$$\Delta\theta_{j,i}^t = \text{sgn}(\alpha_{j,i}^t \omega_{j,i}^t) \Theta_0 \exp\left(\frac{|\omega_{j,i}^t|}{|\alpha_{j,i}^t| + \gamma}\right) \quad (13)$$

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 $sgn(\cdot)$ determines the direction of the rotation angle.

G. Discrete Crossover and Elitism

To backing up the truck-and-trailer with minimized trajectory error (fitness value) the FLCs have to search for the suitable angle of the trailer β with respect to input variables. Discrete Crossover (DC) is used here to expand the search space for the FLC. DC is performed after a fixed number of generations that is after a defined interval τ . Here DC is performed m times for each individual of the population. The process of DC can be explained as follows: First, for individual p_u^t , $u = 1, \dots, n$, select randomly another individual p_v^t , $v = 1, \dots, n$, where $u \neq v$. Let p_u^t and p_v^t be considered as parents. Then, every corresponding allele of them is exchanged with a 0.5 probability, and two new individuals (off springs) p_k^t , $k = 1, 2$, are formed. Figure 6 describes DC.

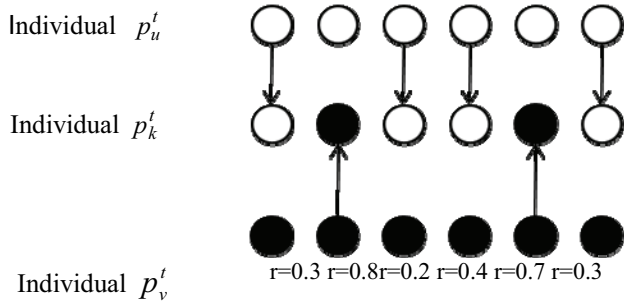


Figure 6. Discrete Crossover

The elitism technique is used to select the individuals with minimized fitness value which also ensure that the rule bases with best fitness value will not be lost.

V. EXPERIMENTAL RESULTS

This section presents the simulation results obtained from the proposed FLC simulating the truck and trailer controller problem. The RCQEA runs for 5 times with generation = 200, population size = 10, scale parameter $\gamma = 5$ and initial rotation angle $\theta_0 = 0.1\pi$. Table 2 shows the simulation parameters of RCQIEA for FLC. Figure 7(a) shows the shape of the membership functions by RCQIEA and 7 (b) shows an example of MF form of fuzzy rule.

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 [8]. 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 $[67.5, 100]$. The range of ϕ_t is divided into seven non-uniform regions $[-90, 0]$, $[0, 66.5]$, $[66.5, 86]$, $[86.5, 94.5]$, $[94.5, 113.5]$, $[113.5, 182.5]$, $[182.5, 270]$. For another input ϕ_c , the linguistic variables are NE (Negative), ZR (Zero) and PO (Positive) and the space $-90 \leq \phi_c \leq 90$ has been divided into three intervals $[-90, -12.5]$, $[-12.5, 12.5]$, $[12.5, 90]$. Similarly seven divided regions of the range of β $[-30, -20]$, $[-20, -7.5]$, $[-7.5, -2.5]$, $[-$

$2.5, 2.5]$, $[2.5, 7.5]$, $[7.5, 20]$, $[20, 30]$ are represented by variables NL, NM, NS, ZE, PS, PM and PL.

TABLE II. MFs AND RULE SET PARAMETERS (OPTIMAL SOLUTION)

Rule No.	RCQIEA(Optimal Solution)							
	Input						Output	
	x		ϕ_t		ϕ_c		β	
	Center	Width	Center	Width	Center	Width	Center	Width
1	16.981	17.1579	-77.340	79.0289	-19.8562	54.732	4.41199	17.4324
2	13.9245	29.9475	-67.792	107.416	-1.15573	121.796	5.94693	18.7478
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
104	80.3251	16.3588	195.069	148.051	2.2023	39.6526	-3.4602	17.0375
105	96.1698	49.7159	252.462	186.979	77.97409	51.664	-1.7170	20.9209

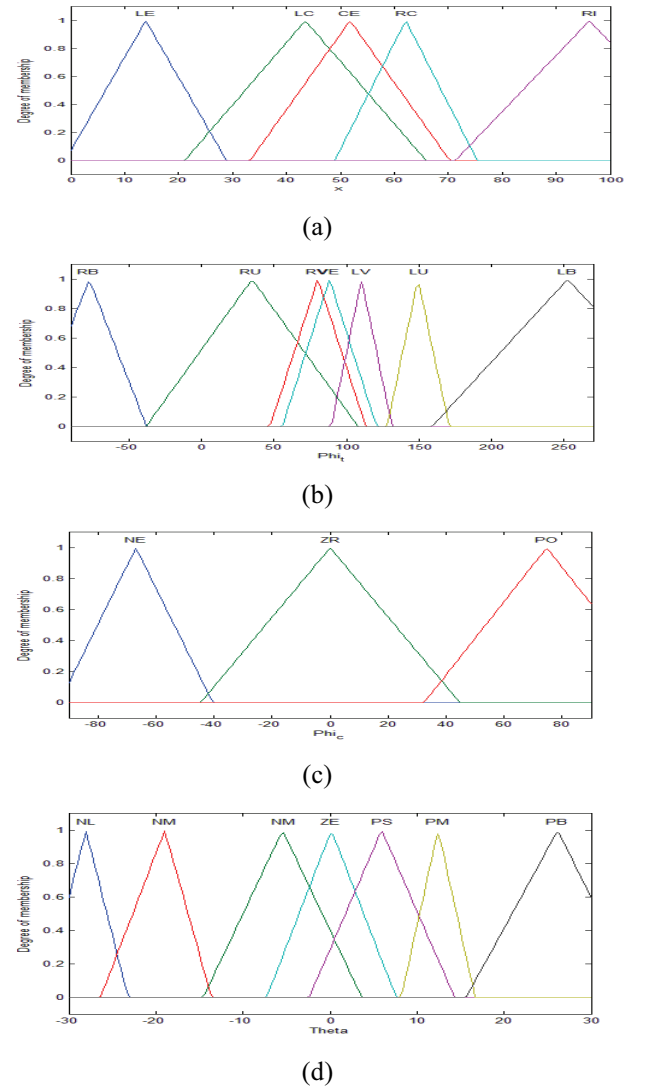
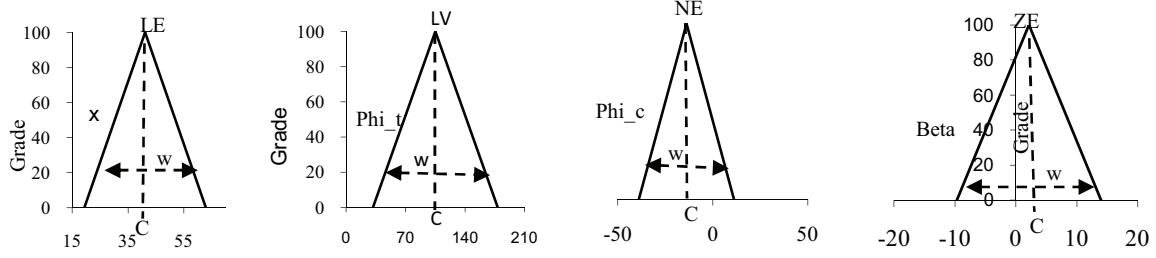


Figure 7(a). Tuned MFs using RCQIEA (a) Input x (b) ϕ_t (c) Input ϕ_c (d) Output β



Rule 7: If $x=LE$ and $\phi_t=LV$ and $\phi_c=NE$ Then $\beta=ZE$

Figure 7(b). MF (triangular) form of fuzzy rule

	ϕ_t							
	β	LB	LU	LV	VE	RV	RU	RB
ϕ_c	NE	¹ PS	⁴ PB	⁷ ZE	PS	NS	PS	PS
	ZR	² PS	NM	NM	NM	PS	NS	PS
	PO	³ ZE	PM	NB	NS	NM	NM	PB

$x=LE$

	ϕ_t							
	β	LB	LU	LV	VE	RV	RU	RB
ϕ_c	NE	NM	ZE	NS	PS	PS	PS	NM
	ZR	NM	NM	NM	NS	PS	ZE	PM
	PO	PS	NM	NM	NS	PB	ZE	NS

$x=LC$

	ϕ_c							
	β	LB	LU	LV	VE	RV	RU	RB
ϕ_c	NE	PS	PB	PS	ZE	PS	PB	PM
	ZR	NM	NS	ZE	ZE	PS	ZE	PM
	PO	NM	NS	ZE	NS	NS	NS	NS

$x=CE$

	ϕ_t							
	β	LB	LU	LV	VE	RV	RU	RB
ϕ_c	NE	NM	PB	PS	PM	PB	PB	NS
	ZR	NM	ZE	PS	NS	PM	NS	PB
	PO	PS	NS	NS	ZE	PM	NS	ZE

$x=RC$

	ϕ_t							
	β	LB	LU	LV	VE	RV	RU	RB
ϕ_c	NE	NS	PS	PM	PM	PB	PS	PM
	ZR	NS	PS	PS	NS	PM	PM	NS
	PO	ZE	NM	NS	PS	NS	NB	ZE

$x=RE$

Figure 8. FAM bank of the truck-and-trailer control System

At the beginning of the simulation, the antecedent and consequent part of each rule, i.e., fuzzy associative memories (FAM) cells are empty. To get the antecedent and consequent

part of each rule in terms of linguistic language form and shape of MFs from the developed RCQIEA generator block, we start the simulation from twenty initial positions $(x_o, y_o, \phi_o, \phi_{c0})$: (25, 30, 60°, 30°), (80, 30, 150°, and 60°)..etc. The initial positions are randomly selected and cover the simulated truck-and-trailer loading zone.

For all the given initial positions the RCQIEA converges to some sets of fuzzy IF-THEN 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, these RB are combined together to form the final FAM as shown in Fig.8 using Fuzzy Amalgamation (FA) methodology [13]. The FA method starts with an empty new FAM. Then it compares and combines two FAMs as follows. If the corresponding cells of two FAMs are same then add the cell entries into the corresponding cell of the new FAM. On the other hand, if corresponding cells of two FAMs are different, then an average of the two different entries is computed using the following formula:

$$Numeric - Value(NV) = \frac{1}{n} \sum_{i=1}^n N_i$$

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 FAM. There are 105 fuzzy rules (fuzzy RB or FAM can be formed as a 5×(3×7) tables with rows to hold the corresponding actions (outputs) as shown in fig.8.

With the above FAM rules (shown in Fig.8), the fuzzy truck- and-trailer controller produced successful truck-and-trailer trajectories starting from any given initial condition. Figure 9 shows the typical example of fuzzy truck-and-trailer trajectories.

In each generation, all chromosomes are evaluated by the FLC to generate the fitness error. Chromosomes are sorted according to the fitness error and the best one among all chromosomes is selected in each generation. Figure 10 shows a graph of fitness generation by generation.

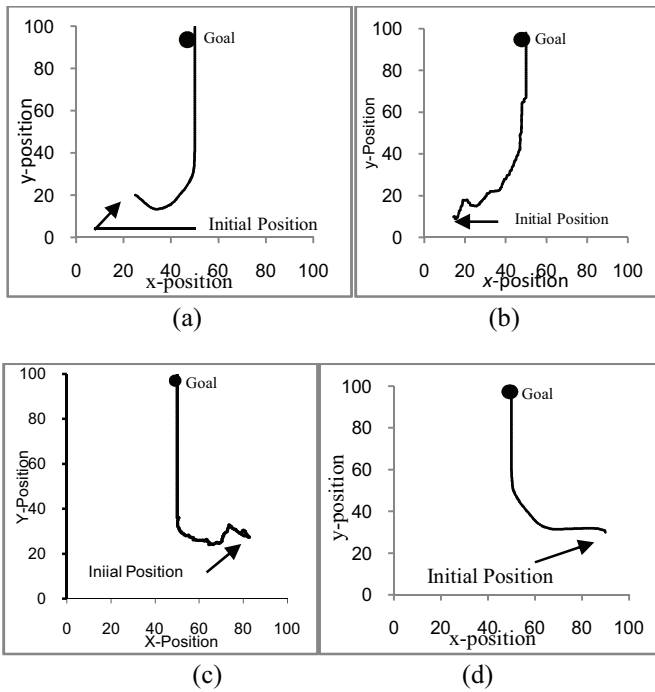


Figure 9. Sample truck-and-trailer trajectories of the fuzzy controller for the initial conditions (x, y, ϕ, ϕ_c) : (a) (25, 20, -45°, 60) (b) (80, 30, 10 and 20) (c) (15, 10, -20°, 45°) (d) (90, 30, 145°, 30°)

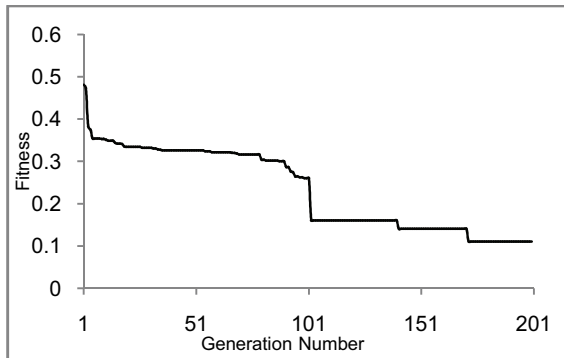


Figure 10. Fitness function value of the solution populations

VI. CONCLUSIONS

In this paper, a new RCQIEA is applied to search for optimal settings of the MFs shapes and gains of the inputs and outputs of FLC for a truck-and-trailer controller problem. The proposed RCQIEA represents a new way of self-learning adaptive FLCs. The simulation results show that it can generate the fuzzy control rules and the MFs in a consequent way automatically without any priori expert knowledge. Moreover the designed adapted fuzzy logic controller effectively realizes the trajectory control of the truck with trailer. The proposed approach may be seen as a step in the development of a better accepting of how to incorporate a fuzzy logic controller with a quantum inspired evolutionary algorithm to achieve as significant adaptive learning capability.

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 obstacle either moving or fixed in the workspace. Moreover, this work also to be extended in the analysis of the behavior of the cooperative real coded 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. H. Kim and J.H. Kim, "Multiobjective quantum-inspired evolutionary algorithm for fuzzy path planning of mobile robot," in *Proc. IEEE Congress Evol. Comput.*, May 2009.
- [2] K. Belarbi, F. Titel, W. Bourebia, and K. Benmahammed, "Design of mamdani fuzzy logic controllers with rule base minimisation using genetic algorithm," *J. Engg. Appl. Artificial Intell.*, vol. 18 pp. 875-880, Oct. 2005.
- [3] D. L. Meredith, C. L. Karr, and K. K. Kumar, "The use of genetic algorithms in the design of fuzzy logic controllers," in *Proc. third workshop on neural netw.*, vol. 1721, 1992, pp. 549-555.
- [4] A. Bagis, "Fuzzy rule base design using tabu search algorithm for nonlinear system modeling," *ISA Trans.*, vol. 47, no. 1, pp. 32-44, Jan. 2008.
- [5] S. Bai and S. Chen, "Automatically constructing grade membership functions of fuzzy rules for students' evaluation," *Expert Syst. Appl.*, vol. 35, pp.1408-1414, Oct. 2008.
- [6] C. C. Yang and N. K. Bose, "Generating fuzzy membership function with self-organizing feature map," *Pattern Recog. Letters*, vol.27, no.4, pp. 356-365, Apr. 2006.
- [7] A. F. M. Huang, S. J. H. Yang, M. Wang, and J. J.P. Tsai, "Improving fuzzy knowledge integration with particle swarm optimization," *Expert Syst. Appl.*, vol. 37, iss. 12, pp. 8770-8783, Dec. 2010.
- [8] T. Koga, K. Horio, and T. Yamakawa, "Self-organizing relationship (SOR) network with fuzzy inference based evaluation and its application to trailer-truck back-up control," in *Proc. 11th int. conf. neural inform. Processing*, 2006, pp. 368-374.
- [9] L.A. Zadeh, "Fuzzy sets," *Information and Control* 8 (3), pp. 338-353, 1965.
- [10] L. X. Wang, *Adaptive Fuzzy systems and control: Design and Stability Analysis*. Englewood cliffs, NJ: Prentice-Hall PTR, 1994.
- [11] G. C. Mouzouris and J. M. Mendel, "Nonsingleton fuzzy logic systems and application," *IEEE Trans. Fuzzy Syst.*, vol.5, no.1, Feb. 1997.
- [12] R. Zhang and H. Gao, "Real -coded quantum evolutionary algorithm for complex function with high dimension," in *Proc. IEEE Int. Conf. Mechatronics and Automation*, Aug. 2007, pp. 2974-2979.
- [13] M. Mohammadian and R. J. Stonier, "Generating fuzzy rules by genetic algorithms," in *Proc. IEEE Int. Works. Robot and Human Comm.*, 1994, pp.362-367.
- [14] Do-Hyeon Kim, Kwang-Baek Kim and Eui-Young Cha, "Fuzzy truck control scheme for obstacle avoidance," *J. Neural Comput. Appl.*, vol. 18, no. 7, pp. 801-811, Oct. 2009.
- [15] T. Yamakawa and T. Koga, "Bio-inspired self-organizing relationship network as knowledge acquisition tool and fuzzy inference engine," in *Proc. IEEE world conf. Comput. Intell.*, 2008, pp. 159-180.
- [16] Y. Li and Y. Li, "Neural-fuzzy control of truck backer-upper system using a clustering method," *Neurocomputing*, vol. 70, pp-680-688, Jan. 2007.
- [17] K. H. Han and J. H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evolutionary Comput.*, vol. 6, no. 6, pp.580-593, Dec. 2002.
- [18] K. H. Han and J. H. Kim, "Quantum-Inspired evolutionary algorithms with a new termination criterion, the gate, and two-phase scheme," *IEEE Trans. Evolutionary Comput.*, vol. 8, no.2, pp.156-169, Apr. 2004.