A Quantum Genetic Algorithm Based Approach for Designing Membership Functions and Rule Sets for Type-2 Fuzzy Logic Controller

Ishrat Khan, Md. Shafiur Rahman Khan, Pintu Chandra Shill

Abstract— **This paper presents a type-2 quantum fuzzy logic controller (T2QFLC) for robot manipulators with unstructured dynamical uncertainty that is customized using quantum evolutionary algorithm. In order to effectively construct type-2 fuzzy logic controllers, quantum genetic algorithms are used to simultaneously design type-2 fuzzy sets and rule sets. Traditional fuzzy logic controllers (FLCs), also known as type-1 fuzzy logic systems employing type-1 fuzzy sets, struggle to describe and reduce the impact of uncertainties that are present in many real-time applications. As a result, type-2 FLC has recently been suggested. A collection of several embedded type-1 FLCs can be thought of as the type-2 FLC. The type-2 FLC design method now in use, however, is not automated and relies on the heuristic expertise of seasoned operators. Our research is aimed at automating the design process. The type-2 FLCs that have emerged can deal with a lot of uncertainty and perform better for mobile robots. Additionally, it has outperformed both the conventionally constructed type-2 FLCs and their type-1 equivalents.**

Index Terms— **Interval Type-2 FLC, Mobile Robot, Interval Type-2 fuzzy sets, Optimization, Quantum genetic Algorithms.**

I. Introduction

Fundamental representation and processing frameworks for linguistic data, fuzzy logic systems (FLS) have tools to handle uncertainty and imprecision. With such exceptional qualities, FLS has been effectively used in a wide range of applications, including control [20], classification [21], and modeling problems [22]–[24]. A set of fuzzy rules and the related membership functions (MFs) that translate inputs into outputs make up a fuzzy model.

Either human specialists give fuzzy rules and MFs, or they are learnt from sample data. It is difficult to understand many decision-making and problem-solving jobs statistically. However, people often achieve success by relying on incomplete rather than perfect knowledge.

Ishrat Khan, Department of Computer Science and Engineering, Khulna Agricultural University, Khulna, Bangladesh, (e-mail:ishrat_riya@kau.edu.bd).

Md. Shafiur Rahman Khan, Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh, (e-mail: srkrifat@gmail.com).

Pintu Chandra Shill, Department of Computer Science and Engineering Khulna University of Engineering & Technology, Khulna, Bangladesh, (e-mail: pintu@cse.kuet.ac.bd).

It may be simple and efficient to build fuzzy logic controllers (FLCs) with the right expert knowledge base (KB). However, without such a knowledgeable KB, designing FLCs can be unpleasant since it relies more on trial and error than a planned strategy.

Due to its natural biological evolution process and ability to quickly and effectively search a vast and complicated solution space, evolutionary algorithms (EA) can be utilized to get over the drawbacks of the trial-and-error approach. Therefore, EA may be used as a strong search technique to carry out activities including creating a fuzzy rule base (RB), optimizing a fuzzy RB, creating MFs, and tweaking the kinds of MFs.

The most crucial challenge, for the majority of fuzzy logic control issues, is to identify the parameters that characterize the type-2 MFs. This makes it possible to transform type-2 MFs optimization problems into parameter optimization issues. These parameters are often based on the expert KB that is developed automatically or heuristically by seasoned control engineers. Numerous techniques have been employed to enhance the behavior of parameter optimization problems as well as the choice and construction of fuzzy rules, including genetic algorithms (GAs) and neural networks (NNs).

Mendel [1] and Hagras [2] have demonstrated that type-1 fuzzy logic systems (FLSs) would not be able to simulate or limit the impact of uncertainties that are prevalent in real-world applications. One constraint is that in logic where the membership grade for each input is a crisp value, a type-1 fuzzy set is guaranteed. The uncertainties may be managed, however, by interval type-2 FLCs (which employ interval type-2 fuzzy sets, identified by fuzzy MFs).

Martnez [3] utilized GA to optimize type-2 FLC. He used GA to create FLC for the disturbed autonomous wheeled mobile robot's control. To learn the parameters of the fuzzy system for intelligent management of nonlinear dynamic plants, Melin and Castillo [4] suggested a technique based on type-2 fuzzy sets and neural networks termed neuro-fuzzy. Tan [5] optimized FLC settings using GA. He employed mixed (type-1 and type-2) fuzzy sets for real-time control in his suggested method.

To create an ideal interval type-2 FLC in this paper, quantum optimization is used. QGA is used to adjust type-2 fuzzy sets and rule sets for developing interval type-2 FLCs. QGA is a self-learning adaptive mechanism. It is discovered that QGA automatically adjusts the best MFs settings and generates the ideal number of fuzzy rules. To compare how the two types of fuzzy logic controllers react to uncertainty, we simulated type-1 and type-2 fuzzy logic controllers.

The remainder of the paper is structured as follows: An introduction to type-2 fuzzy sets and FLC is given in Section II. The issue description as well as the kinematic and dynamic models of the mobile robot are described in Section III. The type-2 Quantum Inspired Evolutionary Fuzzy Logic Controller (T2QIEFLC), which is our method, is introduced in Section IV. Real Coded Quantum Evolutionary Algorithm and RCQEA Procedure is discussed in Section V and VI respectively. Section VII presents a simulation analysis of the mobile robot utilizing the controller as described in Section IV. Finally, section VIII. discusses conclusion and some future works.

II. TYPE-2 FUZZY SETS AND FLC

Prof. Zadeh [6] developed the idea of a type-2 fuzzy set in 1975. Membership function defines a type-2 fuzzy set. Instead of being a point in the closed interval [0, 1], the fuzzy grade of that is a fuzzy set in that range. A type-2 membership function [7], indicated by the symbol, characterizes a type-2 fuzzy set, where and, that is,

$$\tilde{A}=\{((x,u),\mu_{\tilde{A}}(x,u))\,|\,\forall_x \in X \quad \forall_u \in J_x \subseteq [0,1]\}$$

$A=\int_{x\in X}\int_{u\in J_x}\mu_A(x,u)/(x,u)$ $\quad J_x \in[0,1]$ 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 $\forall_x \in X$ [7]. The footprint of uncertainty (FOU) [7] is a limited area that contains the uncertainty in the primary memberships of a type-2 fuzzy collection. All primary memberships are unified under it [7].

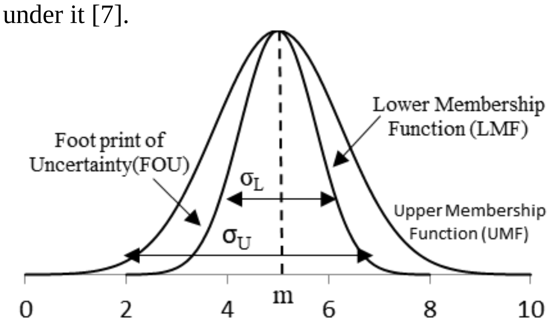


Figure 1. An Interval Gaussian type-2 fuzzy set where σ_L and σ_U are minimum and maximum resultant widths respectively.

A. Type-2 FLC

A type-2 FLC is made up of five parts, as shown in Fig. 2. These parts are fuzzifier, defuzzifier, type-reducer, knowledge base (KB) and fuzzy inference engine.

B. Fuzzifier

The input is fuzzified in fuzzy form using a fuzzification operator because it is in crisp normalized values. A crisp input vector with p inputs is translated into input fuzzy sets by the fuzzifier [8][9]. However, because the singleton fuzzification approach is quick to compute and ideal for mobile real-time operation, we have utilized it here. In the singleton fuzzifier, fuzzy set 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 $x \neq x_i$ which input measurement with x is a perfect crisp.

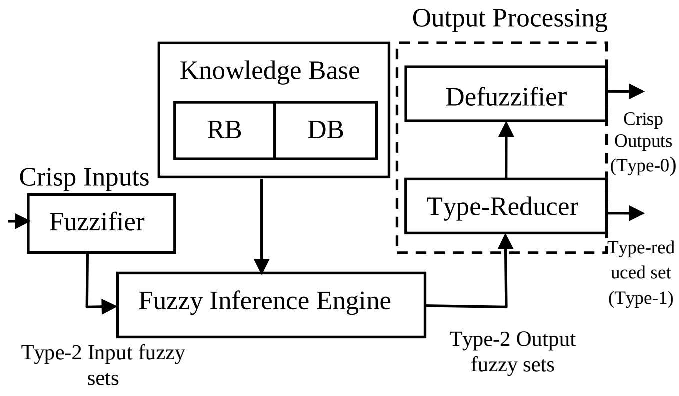


Figure 2. A type-2 FLC

C. Rule Base

The antecedents and consequents will be represented as interval type-2 fuzzy sets, but the rules will stay the same as in type-1 FLC [9]. The FLC under discussion employs fuzzy implication and compositional rules of inference for approximation reasoning, similar to the majority of FLCs [10]. Consider the case where we need to design a type-2 FLC mobile robot 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:

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

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

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.

D. Fuzzy Inference Engine

Based on the fuzzy logic principle, the fuzzy inference engine mixes rules and provides a mapping from type-2 fuzzy sets in the input universe of discourse to type-2 fuzzy sets in the output universe of discourse. The output of the ith type-2 rule is as follows:

$\tilde{R}^i: \tilde{F}_1 \times \tilde{F}^i_2 \times \ldots\ldots \times \tilde{F}^i_p \rightarrow \tilde{G}^i_k$ and the type-2 fuzzy relation can be expressed by membership function as:

$$\mu_{R^i}(\vec{x},y)=\mu_{F^i_1\times F^i_2\times\ldots\times F^i_n\rightarrow G^i}(\vec{x},y)$$
$$=\mu_{F^i_1}(x_1)\prod\ldots\ldots\prod\mu_{F^i_n}(x_n)\prod\mu_{G^i}(y)$$

Where $\prod$ denotes meet operation. The extended sup-star composition is used to mix membership grades in the input and output type-2 fuzzy sets, and the Join operation is used to combine several rules. In [11–12], they are defined and discussed in further detail.

E. Type Reduction

When an interval type-2 fuzzy set is converted to an interval-valued type-1 fuzzy set, type-reduction has taken place. These type reduced sets are then defuzzified to produce clear outputs. Due to its fair processing complexity, centroid type reduction has been used in this work. For the centroid type reduction process, firstly combines the output type-2 fuzzy sets using union [4] (minimum t-norm), $\tilde{B} = \cup_{l=1}^{M} \tilde{B}^{l}$, 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:

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

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 $\prod_{i=1}^{N} T_i$.

F. Defuzzification

After the type-reduction stage, defuzzify the type reduced interval set $y_c(\vec{x})$, 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

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

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. Depending on the circumstances, it could have to proceed in a straight line or make a curve to create a path that avoids collisions. The issue is adapted from [13]. Figure 3 shows the simulated geometry for the robot and loading dock in a schematic manner. In this workspace, a mobile robot is traveling amid a single fixed barrier. Independent of the robot's starting position, the control system has to progressively identify a route to the loading dock. 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 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.

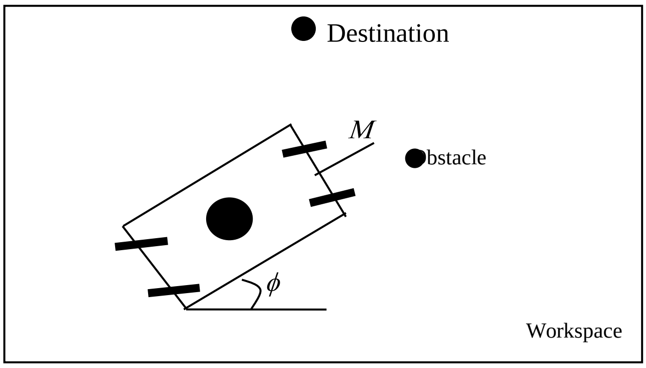


Figure 3. Mobile Robot and loading dock illustration.

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

Where l is the length of the robot, we assume l=4. Eq. (1) will be used to derive the next state when present state and control are given.

A very nonlinear difficult situation is what this experiment exemplifies. Here, type-1 FLCs known as type-1 genetic fuzzy logic controllers (T1GFLC) and type-2 genetic fuzzy logic controllers (T2GFLC) are compared for their performance.

IV. Hybrid Q-Fuzzy Optimization of the type-2 FLCs

Based on the idea of quantum computing, QIEA is a probability optimization method. The smallest piece of information in quantum computing is referred to as a Q-bit. A Q-bit can be represented as $\alpha|0\rangle + \beta|1\rangle \leftrightarrow \binom{\alpha}{\beta}$

Where α^2 +β^2=1. |α|^2 indicates the probability of finding the Q-bit in “0” state and |β|^2 indicates the probability of finding the Q-bit in “1” state. A Q-bit may be in “1” state, in “0” state or in a linear superposition of the two states.

A Q-bit individual as a string of m Q-bits is defined as

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_m \\ \beta_1 & \beta_2 & \ldots & \beta_m \end{bmatrix}$$

Where $|\alpha_i|^2 + |\beta_i|^2 = 1$, i = 1, 2, 3…..m. A Q-gate which is a quantum mutation gate is used to speed up the convergence. It is defined as:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix}$$

where $\Delta\theta_i$, $i = 1, 2, 3 \ldots, m$, is the rotation angle of a Q-bit towards the "0" state or "1" state depending on its sign. After applying Q-gate, the Q-bit should satisfy the normalization condition $|\alpha'|^2 + |\beta'|^2 = 1$, where $|\alpha'|^2$ and $|\beta'|^2$ are the values of updated Q-bit.

Figure 4 illustrates a quantum-type-2 fuzzy system, which employs QIEA to establish type-2 fuzzy sets and fuzzy control rules. In this paper, we used QIEA to optimize the Type-2 FLC MFs' parameters, and we considered applying Type-2 MFs with Gaussian Intervals to each of our three variables. For the selection and definition of the RB of type-2 FLC, we simultaneously used QIEA.

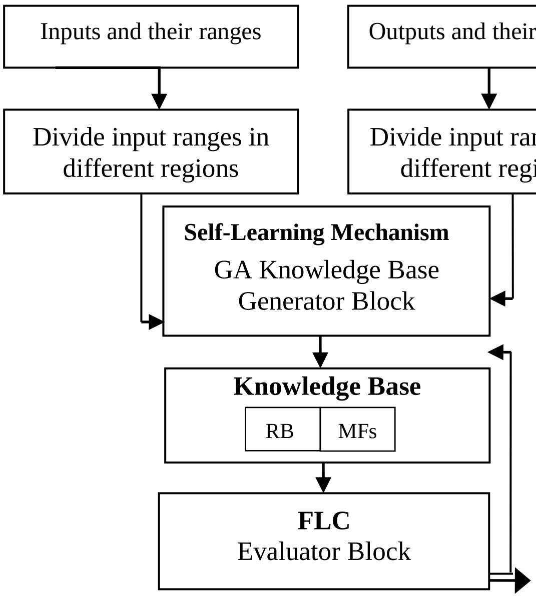


Figure 4. Integration of type-2 FLCs and QEA.

Fuzzy rules and MFs are created using two input variables x and one output variable. These variables are therefore encoded on a QEA chromosome. There are 5, 7, and 7 regions, respectively, in the domains for x, and. The following Table I lists the linguistic words (MFs) that are used to characterize each of the input and output variables.

Table I. Definition of linguistic variables.

x	LE – Left End	RC – Right Center
	LC – Left Center	RE – Right End
	CE – Center	
ϕ	LB – Left Below	RV – Right Vertical
	LU – Left Upper	RB – Right Below
	LV –Left Vertical	RU – Right Upper
	VE – Vertical	
θ	NL – Negative Large	PS – Positive Small
	NM – Negative Medium	PM – Positive Medium
	NS – Negative Small	PL – Positive Large
	ZE – Zero	

The rule base contains total 35 rules. Each rule includes the real value of *x*, Φ, and *θ*. Each variable (input and output) is divided into three parts: begin (b), center (c), and end (e). So the rule base looks like:

Input x	Input φ	Output θ	
$b_{11}\ c_{11}\ e_{11}$	$b_{12}\ c_{12}\ e_{12}$	$b_{13}\ c_{13}\ e_{13}$	→ Rule 1
$b_{21}\ c_{21}\ e_{21}$	$b_{22}\ c_{22}\ e_{22}$	$b_{23}\ c_{23}\ e_{23}$	→ Rule 2
⋮	⋮	⋮	⋮
$b_{n1}\ c_{n1}\ e_{n1}$	$b_{n2}\ c_{n2}\ e_{n2}$	$b_{n3}\ c_{n3}\ e_{n3}$	→ Rule n

To minimize the length of each rule, only the center (c) and width (w) is used. Using c and w, begin (b) and width (w) can be easily calculated:

Begin= center-width

End = center+width

So, each rule is represented as follows:

Input x	Input φ	Output θ
C_1W_1	C_2W_2	C_3W_3

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 [14]. 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 are [-40, -28], [-28, -12.5], [-12.5, -2.5], [-2.5, 2.5], [2.5, 12.5], [12.5, 28], [28, 40]. 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 *x*, 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 92consecutive real genes.

V. Real Coded Quantum Evolutionary Algorithm

There are various types of QIEA such as Quantum Evolutionary Algorithm (QEA), Real Coded Quantum Evolutionary Algorithm (RCQEA). In this paper, RCQEA is used as QIEA.

A. Representation

Each individual is represented by real coded triploid chromosome which can be defined as follows:

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

Where $(R_i\ \alpha_i\ \beta_i)^T$, i =1, 2…. n is the i^{th} allele of real coded triploid chromosome and R_i is the i^{th} rule in fuzzy rule base. $(\alpha_i, \beta_i)^T$ is a pair of probability amplitudes of one qubit and satisfies normalization condition $|\alpha|^2+|\beta|^2 = 1$, n is the length of real-coded triploid chromosome which is 35.

B. Mutation

Gaussian Mutation operator is applied to update population at each generation. The i^{th} allele is randomly selected from p_j^t and the centers(c) of the input-output variables in the rule of the selected allele are changed as follows:

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

Where Max and Min are respectively upper and lower bound of the regions in which C^t lies. C^{t+1} may not be within the limit so it is clipped to keep it within the region of C^t. The center of θ is considered always the whole range of θ. $N(0, (\sigma_{j,i}^t)^2)$ denotes the Gaussian distribution of mean 0 and variance $(\sigma_{j,i}^t)^2$. The value of variance $(\sigma_{j,i}^t)^2$ is either $|\alpha_{j,i}^t|^2$ or $|\beta_{j,i}^t|^2$ /5 based on "Fine Search" or "Coarse Search" to be implemented []. The width (W^t) of each center is updated as follows:

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

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 Quantum Rotation Gate (QRG) as follows:

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

Here $\Delta\Theta_{j,i}^t$ is rotation angle of Q-bit and it is calculated as follows,

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

Where Θ_0 is the initial rotation angle, γ is the scale parameter. These control the rotation angle and increase the speed of convergence, the sign ***sgn*** (.) determines the direction of the rotation angle.

C. Discrete Crossover (DC) and Elitism

DC is performed repeatedly after a fixed number of generations and it expands the search space to find the suitable steering angle θ with respect to input variables to backing up the truck with minimized trajectory error (fitness value). The elitism technique is used to ensure that the rule base with best fitness value will not be lost.

VI. RCQEA Procedure

A. Initialization

A Population of N individuals $P^t = \{\ p_1^t \cdots p_j^t \cdots p_N^t\ \}$ is initialized by randomly chosen real numbers, where p_j^t is an individual defined in ().

B. Decode and evaluation

At each generation t RCQEA maintains a population of real-coded triploid chromosome. A rule $R_{j,i}^t$ in p_j^t contains six values, center (C) and width for each of three variables (x, φ and θ). Decode the every chromosome into RB and MFs for the construction of type-2 FLC and FLC is executed on the truck 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 performance to the problem. The fitness value for each chromosome is defined as the trajectory error.

C. Recombination

Apply mutation and crossover operator to chromosomes and generate new chromosomes as well as new generation. Check the termination condition and go to step 2 if the termination condition is true otherwise go to step 4.

D. Stop

The best fitted chromosome is kept and solution has been achieved.

VII. Experimental Result and Performance Comparison

We conducted a number of tests in our simulated arena where the controller was evolved in order to assess the proposed system's correctness. Table II displays the ideal MF means and standard deviations for x, and y. The produced optimum rule base is also displayed in Table II after being converted from optimal parameter to linguistic form. Fig. 8 displays fitness vs. generations required in this paper and tuned Gaussian Type-2 Fuzzy sets are also shown in Fig. 7. Table

III. Shows FAM- bank matrix for the fuzzy backing up a truck. The two antecedents and one consequent type-2 MF of T2GFLC are shown in Figures 5 and 6. The performance comparison between T1GFLC (type-1) and T2GFLC is shown in Figure 9. The timeframes it takes the mobile robot to get at the desired spot under 5 distinct beginning circumstances are shown in Figure 9, and their trajectories are depicted in Figure 5 and 6 respectively. The five starting conditions for with their stages are shown in Table IV. It is clear from Figures 5, 6, and Table IV. that the performances of T2GFLC are superior to those of T1GFLC. By adopting interval T2FLC, the objective position may be reached in less time and with smoother trajectory (shown in Fig.6).

Table IV. Starting condition with their stages.

Case		1		2		3		4		5	
x	y	15	15	15	30	30	50	65	25	75	50
ϕ		60^0		-30^0		15^0		25^0		-45^0	
T1FLC		40		56		34		31		45	
T2FLC		28		40		27		22		38	

Table II. Optimal Means and Standard Deviation of Interval T2QFLC Antecedents MFs of *x*, *y* and consequents MFs of θ.

			MFs of *x*																	
			LE			**LC**			**CE**			**RC**			**RE**					
			m	**σ_U**	**σ_L**	**m**	**σ_U**	**σ_L**	**m**	**σ_U**	**σ_L**	**M**	**σ_U**	**σ_L**	**m**	**σ_U**	**σ_L**			
			10.31	**34.12**	**21.16**	**30.7**	**31.86**	**26.81**	**51.2**	**20.14**	**10.25**	**61.69**	**41.81**	**29.3**	**94.21**	**57.31**	**64.13**			
MFs of *y*																				
NL			**NM**			**NS**			**ZE**			**PS**			**PM**			**PL**		
m	**σ_U**	**σ_L**	**m**	**σ_U**	**σ_L**	**m**	**σ_U**	**σ_L**	**m**	**σ_U**	**σ_L**	**M**	**σ_U**	**σ_L**	**m**	**σ_U**	**σ_L**	**m**	**σ_U**	**σ_L**
-90.81	**94.17**	**84.21**	**2.16**	**189.5**	**171.3**	**75.5**	**135.9**	**92.60**	**90.8**	**122.1**	**98.12**	**145.1**	**74.28**	**51.2**	**182.3**	**83.00**	**44.28**	**252.4**	**135.4**	**96.4**
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**
-38.16	**22.15**	**16.13**	**-25.22**	**22.86**	**11.16**	**-8.59**	**25.13**	**20.71**	**2.15**	**17.17**	**13.14**	**12.28**	**16.15**	**8.16**	**24.16**	**27.14**	**16.39**	**36.17**	**19.13**	**8.16**

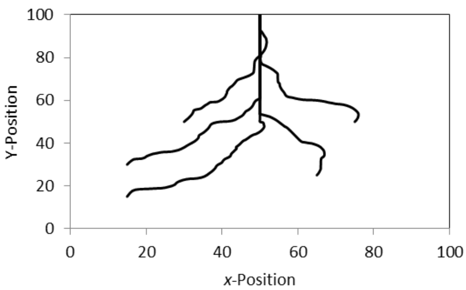


Figure. 5 Show robot trajectories avoiding stationary obstacle (cross-hatched circle) via T1QFLC.

A. *Fuzzy Control Rules*

We have obtained the fuzzy control rules (shown in table II) from the best chromosomes of QGA after 100 generations. Fig. 8 depicts a visual representation of fitness vs. generations

Table III. FAM-bank matrix for the fuzzy backing up a truck.

θ		ϕ						
		NL	*NM*	*NS*	*ZE*	*PS*	*PM*	*PL*
x	*LE*	^{1}*ZE*	^{2}*PB*	^{3}*PM*	^{4}*NM*	^{5}*NB*	^{6}*NB*	^{7}*NM*
	LC	^{8}***PM***	^{9}*PS*	^{10}*NS*	^{11}*NM*	^{12}*NB*	^{13}*PB*	^{14}*NB*
	CE	^{15}*PS*	^{16}*PM*	^{17}*PM*	^{18}*ZE*	^{19}*NS*	^{20}*NM*	^{21}*NS*
	RC	^{22}*PB*	^{23}*PB*	^{24}***NS***	^{25}*PM*	^{26}*ZE*	^{27}*NS*	^{28}*NM*
	RI	^{29}*PB*	^{30}*PB*	^{31}*PM*	^{32}*PB*	^{33}*PS*	^{34}*ZE*	^{35}*ZE*

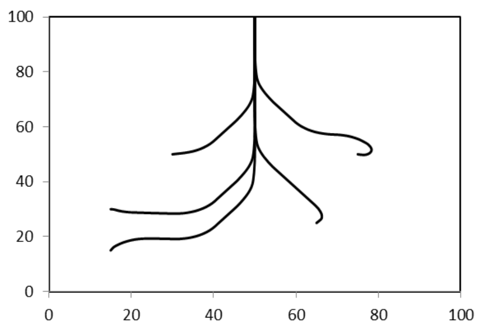
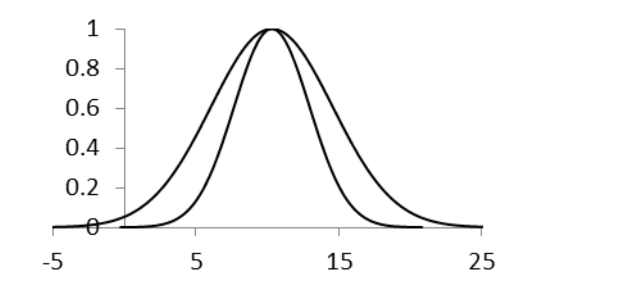
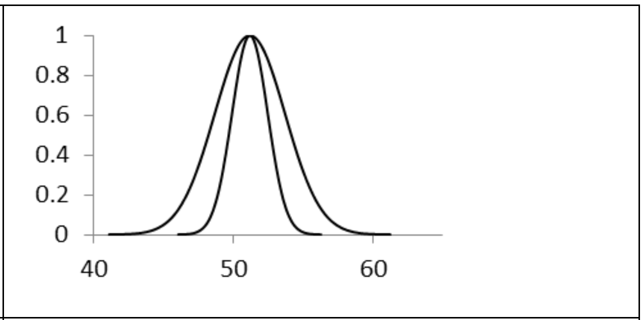


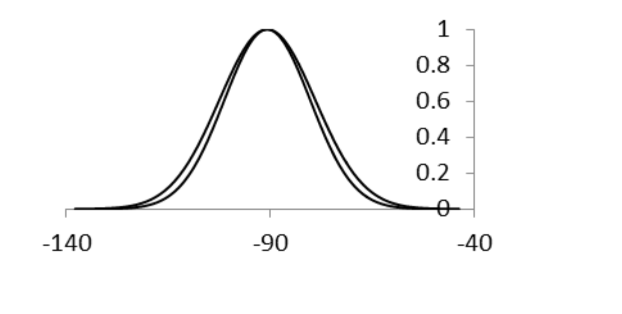
Figure 6. Show via interval T2QFLC, all from 5 different initial conditions.



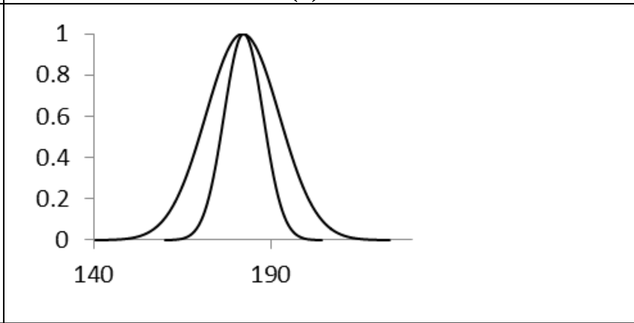
(a) LE of *x*



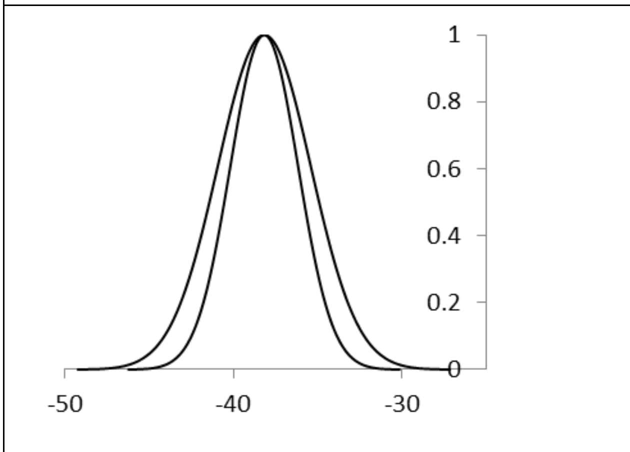
(b) CE of *x*



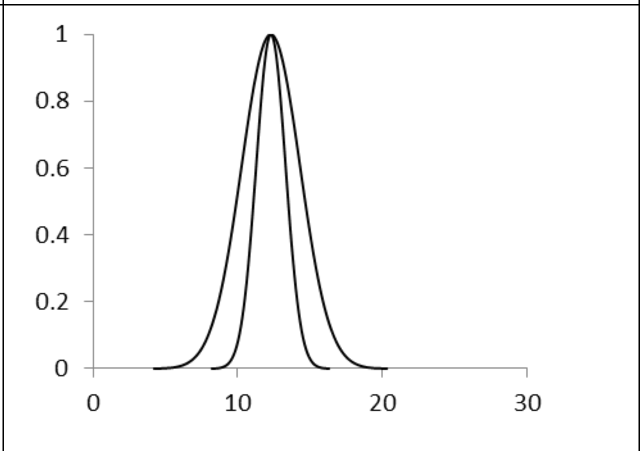
(c) NL of *y*



(d) PM of *y*



(e) NB of θ



(f) PS of θ

Figure 7: Tuned Gaussian Type-2 Fuzzy sets

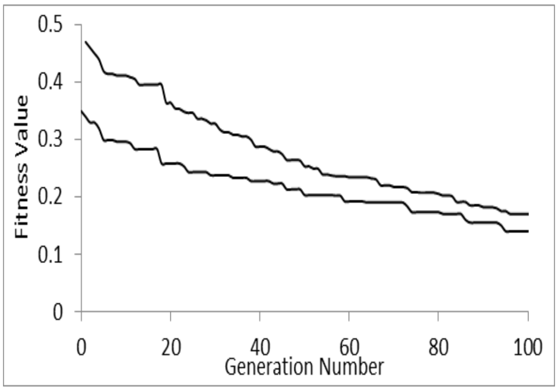


Figure 8. Fitness vs. Generations.

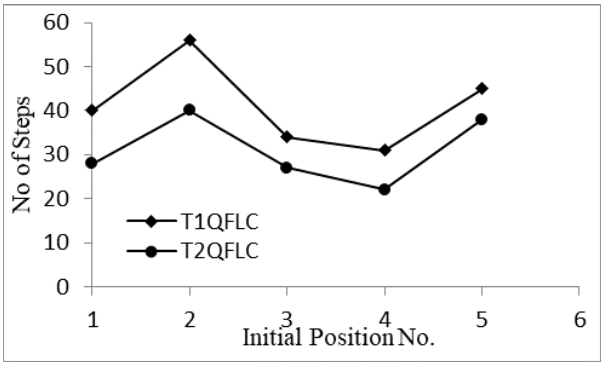


Figure 9. Shows the results of trajectory errors in T1QFLC and T2QFLC.

VIII. CONCLUSION

This paper has demonstrated the potential of evolving the type-2 MFs and rule set parameters of interval type-2 FLCs utilizing a QGA-based architecture. We have demonstrated that an integrated FLCs and hybrid GAs architecture is a self-learning adaptive approach which is able to develop a reliable fuzzy control rules and produce optimal MFs parameters without any priori information intended for mobile robot control in real-world scenarios. Evolutionary type-2 FLCs may solve control issues in domains where there is no previous information, such as in the case of mobile robots. In compared to type-1 FLCs, the genetically developed type-2 FLCs has better control performance (T1GFLC).

The following are the ideas for possible follow-up for this paper: This research project will be expanded to include intelligent control of mobile robots, robotic arm control in the presence of moving obstacles, and path planning for multiple mobile robots in the presence of numerous impediments that are either stationary or moving in the workspace.

REFERENCES

[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," Proc. of NAFIPS, pp. 22–27, 2002.

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

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

[8] 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.

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

[10] 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.

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

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

[13] M. Mohammadian andR. 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.

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

[15] R. I. John, " Embeded interval valued type-2 fuzzy sets," in Proc. FUZZ-IEEE Int. Conf., pp. 1316-1319, May 2002.

[16] J.M. Mendel, "Uncertain rule-based fuzzy logic systems: introduction and new directions," Prentice Hall, NJ. 2001.

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

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

[19] 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.

[20] Pierre Guillemin, "Fuzzy Logic Applied to Motor Control," IEEE Transaction on Industry application, vol. 31, no.1, 1996.

[21] Shang-Ming Zhou and John Q. Gan, "Constructing L2-SVM-Based Fuzzy Classifiers in High Dimensional Space with Automatic Model Selection and Fuzzy Rule Ranking," IEEE Transaction on Fuzzy Syst., vol. 15, no. 3, June 2007.

[22] Z. Liu and H.X. Li, "A probabilistic fuzzy logic system for modelling and control," IEEE Trans. Fuzzy Syst., vol. 13, pp. 848–859, 2005.

[23] L.X. Wang, "Stable adaptive fuzzy control of nonlinear systems," IEEE Trans. Fuzzy Syst., vol. 1, pp. 146–155, 1993.

[24] I-Hsum Li, Lian-Wang Lee, "A hierarchical structure of observer-based adaptive fuzzy-neural controller for MIMO systems," Fuzzy Sets and Systems, vol.185, issue 1, pp. 52–82, 2011.