International Journal of Innovative Computing, Information and Control
Volume **9**, Number **2**, February **2013**
 ISSN 1349-4198
pp. 451–464

EVOLUTIONARY OPTIMIZATION OF THE FUZZY CONTROLLERS IN A NAVIGATION SYSTEM FOR A MOBILE ROBOT

Abraham Melendez, Oscar Castillo and Patricia Melin

Division of Research and Graduate Studies
Tijuana Institute of Technology
Tijuana, Baja California, México
abraham.ms@gmail.com; { ocastillo; pmelin }@tectijuana.mx

Received November 2011; revised April 2012

Abstract. *This paper describes an evolutionary algorithm approach for the optimization of two control blocks within a proposed navigation control system for a mobile robot, and the control blocks are two Fuzzy Inference Systems (FIS), which are in charge of two main behaviors applied on a mobile robot that are for tracking and reaction. The evolutionary algorithm optimizes the number of membership functions and fuzzy rules for each of the FIS controllers.*
Keywords: Fuzzy logic, Mobile robot, Genetic algorithm, Control, Optimization

1. **Introduction.** The use of mobile robots has increased over the last decades in many areas from industrial work to research and household and one reason for this is that they have proved useful in each of these areas from doing very specific task to ongoing monotonous shores, and they help their human counterpart be more productive and efficient. Also as hardware technology is moving forward and developing more capable robots at lower cost, this is another reason for this increase and why we are seeing them in more common places.

The mobile robot needs to move around its environment and this is why a great deal of research has been invested on testing them with control systems that allow the robots to navigate on their own, and different methodologies have been applied from traditional control such as PD, PID [1] to soft computing methods like Fuzzy Logic [3-15], Neural Networks [10] and hybrid [16,17] ones also.

In this paper, the navigation control system has been designed to combine two key behaviors that are considered to be required for any navigation control system of a mobile robot. The first one is a tracking controller, and this is an obvious one since there is no point of having a navigation system on a robot, which cannot go to a desired location; the second one is a reactive controller and here we considered this one to be off great importance also, since the tracking controller can get the robot to the destination, but that will be on an ideal situation where there are no obstacles present on the robots path.

The reactive controller is for those cases where an obstacle free path cannot be guaranteed; this is where the reactive controller will do its work providing a behavior that will make the robot react to any type of obstacle so that the robot can continue on its journey. In this paper we describe the integration method for these two controls as part of the complete Navigation Control System, the control blocks are fuzzy inference systems of type-1 and type-2, and a general GA (Genetic Algorithm) is applied to the optimization of each of the controller blocks with a specific fitness function for each part that will evaluate the corresponding individual performance.

As related work, we can find that Cupertino et al. [18] developed a Fuzzy control of a mobile robot, based on 3 FLCs (Fuzzy Logic Controller) and one Fuzzy Supervisor that was in charge of determining which FLC behavior will be active; there the FLCs are of Type-1 and the Fuzzy Supervisor mainly acts as a switch. In our proposed method the fuzzy integrator acts more as a fusion block. Coupland [19] proposed a Type-2 Fuzzy Control of a Mobile Robot, which is based on Payton et al. [20] Command Fusion, where the idea is that a behavior should work with others to find a mutually beneficent solutions, where each behavior takes into consideration every possible output with its corresponding activation value (positive or negative), and a winner takes all network which is used to select the winning responses for each behavior. Coupland suggests using two FISs, one for goal seeking and the other for obstacle avoidance. The activation value for each of the FIS output will be a Fuzzy set that will be passed to the command fusion block to later be defuzzifed and that crisp value pass to the Actuator block, being a difference with our proposed control the integration method of the two behaviors. The control navigation of a mobile robot is a topic that has been extensively investigated over the years, and the method proposed in this paper is based on the idea that separation and the cooperation between key behaviors produce a better result than the use of a single behavior and it differs from previous approaches from the integration perspective done by a FIS that is in charge of the weighted system, which will assign a weight to each response from each controller by each control step that is combined to obtain a unified single response to the robot.

This paper is organized as follows. In Section 2, we describe the mobile robot used in these experiments. Section 3 talks about the navigation control system. Section 4 describes the development of the evolutionary method. Section 5 shows the simulation results. Finally, Section 6 shows the conclusions.

2. **Mobile Robot.** The particular mobile robot is considered in this work. The robot is based on the description of the Simulation toolbox for mobile robots [21], which assumes a wheeled mobile robot consisting of one conventional, steered, unactuated and not-sensed wheel, and two conventional, actuated, and sensed wheels (conventional wheel chair model). This type of chassis provides two DOF (degrees of freedom) locomotion by two actuated conventional non-steered wheels and one unactuated steered wheels. The Robot has two degrees of freedom (DOFs): y-translation and either x-translation or z-rotation [21]. Figure 1 shows the robot's configuration; it has 2 independent motors located on each side of the robot and one castor wheel for support located at the front of the robot.

The kinematic equations of the mobile robot are as follows:

Equation (1): The sensed forward velocity solution [21]

$$\begin{pmatrix} V_{B_x} \\ V_{B_y} \\ \omega_{B_z} \end{pmatrix} = \frac{R}{2l_a} \begin{bmatrix} -l_b & l_b \\ -l_a & -l_a \\ -1 & -1 \end{bmatrix} \begin{pmatrix} \omega_{W_1} \\ \omega_{W_2} \end{pmatrix} \tag{1}$$

Equation (2): The Actuated Inverse Velocity Solution [21]

$$\begin{pmatrix} \omega_{W_1} \\ \omega_{W_2} \end{pmatrix} = \frac{1}{R(l_b^2+1)} \begin{bmatrix} -l_a l_b & -l_b^2-1 & -l_a \\ l_a l_b & -l_b^2-1 & l_a \end{bmatrix} \begin{pmatrix} V_{B_x} \\ V_{B_y} \\ \omega_{B_z} \end{pmatrix} \tag{2}$$

Under the Metric system are

V_{B_x}, V_{B_y} Translational velocities $[\frac{m}{s}]$,

ω_{B_z} Robot z-rotational velocity $[\frac{rad}{s}]$,

ω_{W_1}, ω_{W_2} Wheel rotational velocities $[\frac{rad}{s}]$,
R Actuated wheel radius [m],
l_a, l_b Distances of wheels from robot's axes [m].

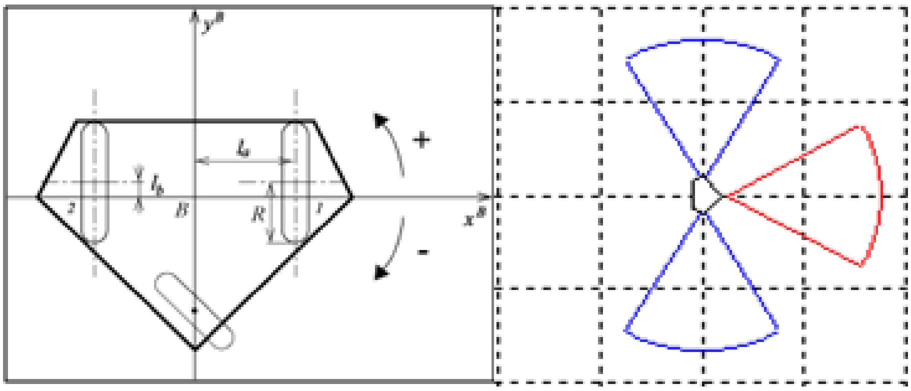


FIGURE 1. Kinematic coordinate system assignments [21]

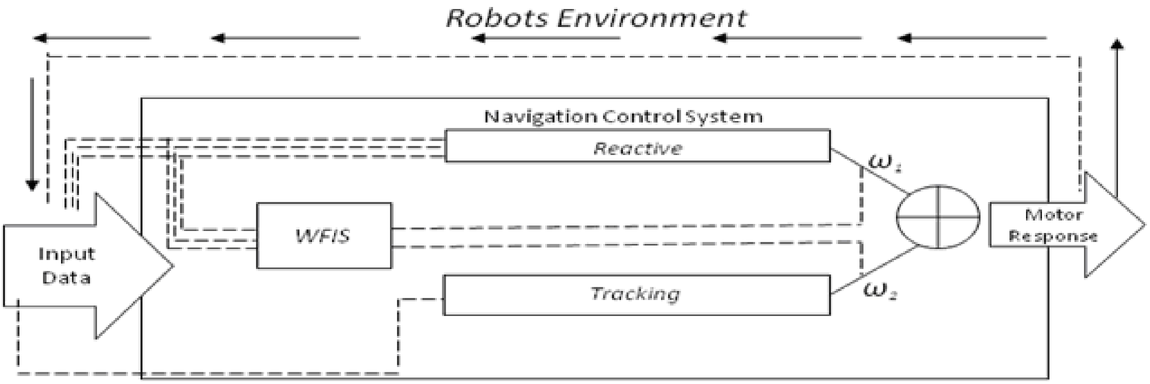
3. **Navigation Control System.** The proposed control system consists of three main fuzzy blocks: two are behavior based and the other one is in charge of the response integration, the behaviors are the reactive and tracking blocks, each one will provide its specific behavior that will be combined into one response by the integration block.

Each behavior block is in charge of its own task, the problem is that they seem to conflict with each other when an unexpected obstacle arises, because if at the time off planning the route the obstacles are present then the route can be designed to avoid them, but when there are obstacles that we were unaware, the two behaviors enter in contradiction one is designed to avoid the object and the other to keep the robot on its track.

The most common solution will be to just switch between controllers when need it; however, this approach is not very efficient due to the lack of awareness the two blocks have of each other, the reactive will effectively keep the robot from the collision but it may redirect the robot farther away from its destination to a point where the tracking controller can no longer find its way back to the reference, or the tracking controller can guide the robot straight into the obstacle if the reactive control is not activated on time. The proposed referral for control navigation is to always have both controls active and their responses are combined and generate the movement of the robot, the integration is done with another fuzzy block call WFIS [9] (Weight-Fuzzy Inference System) and what this controller does is to assign response weights to each of the controllers crisp response value.

The inputs are gathered from the information that we can collect from the robot (sensors) or the environment by other means (cameras) and from this we need to create the knowledge rule base to give higher activation values to the response we want to take the lead on the robot movement one example of the rule is the following (*If Front_Sensor_Distance is Close Then TranckingWeight is Medium and ReactiveWeight is Medium*), both off our controls provide the right and left motor speed and we combine each one with the weight given by the WFIS block. Figure 2 shows the proposed navigation control.

4. **Evolutionary Method Description.** The Genetic Algorithm (GA) was applied to each of the design problems, of finding the best fuzzy reactive and tracking controllers.



$$\mathrm{RMSR} = \frac{\left[\left(\frac{\sum_i \mu_A(y_i)\times y_i}{\sum_i \mu_A(y_i)}\right) * \omega_1\right] + \left[\left(\frac{\sum_i \mu_A(y'_i)\times y'_i}{\sum_i \mu_A(y'_i)}\right) * \omega_2\right]}{2}$$

$$\mathrm{LMSR} = \frac{\left[\left(\frac{\sum_i \mu_A(y'''_i)\times y'''_i}{\sum_i \mu_A(y'''_i)}\right) * \omega_1\right] + \left[\left(\frac{\sum_i \mu_A(y''_i)\times y''_i}{\sum_i \mu_A(y''_i)}\right) * \omega_2\right]}{2}$$

Figure 2. Navigation control system [7]

The purpose of using an evolutionary method is to find the best possible controllers of each type and this can be obtained using the GA, as it searches along the solution space, combining the attributes from the best controllers in generating new ones, this concept taken from the building blocks theory.

The idea was to optimize the parameters in the Membership Functions, but also the number of Membership functions and this means to also optimize the number of rules making this a multi objective problem. For this we will take advantage of the HGA (Hierarchical Genetic Algorithm) intrinsic characteristic to solve multi objective problems.

The work of the GA was divided in two main modules, one that handles all the operations related to the selection and chromosome manipulation, which we use for all our controllers that we work on, the other module is the one where we evaluated the performance of each chromosome and this part is different on each case. With this approach we utilize the generality of the GA and just have a specific evaluation method for each controller. Figure 3 shows the 2 main modules.

The GA module is in charge of initializing the population, selecting the chromosomes that will be used for the genetic operations and letting the Evaluation Module know which chromosomes are ready to be evaluated and reinserting them to the population pool.

4.1. **Chromosome encoding.** Each individual on the population will represent a FIS controller, each of which will be encoded on a vectorial structure that will have "n" main sections, one for each variable (input and output). Each main section will contain 2 subsections (control genes, Connection genes). The section and subsection sizes depend on the controller they represent.

4.2. **Reactive controller.** The function of the reactive control is to give the same ability that we apply when we are driving, that is to react to unexpected situations, traffic jams, stop lights, etc., but in a more basic concept and ability, to the problem that is the navigation of the robot. A forward moving behavior response out off the control is desired. The objective is to guide the robot through the maze avoiding any collision. It is not our objective to optimize the robot to find the maze exit, we use a maze to optimize the reactive control because of the characteristic it offers to the simulation, i.e., it is a closed space where the robot cannot easily wonder off and each wall is considered an obstacle to

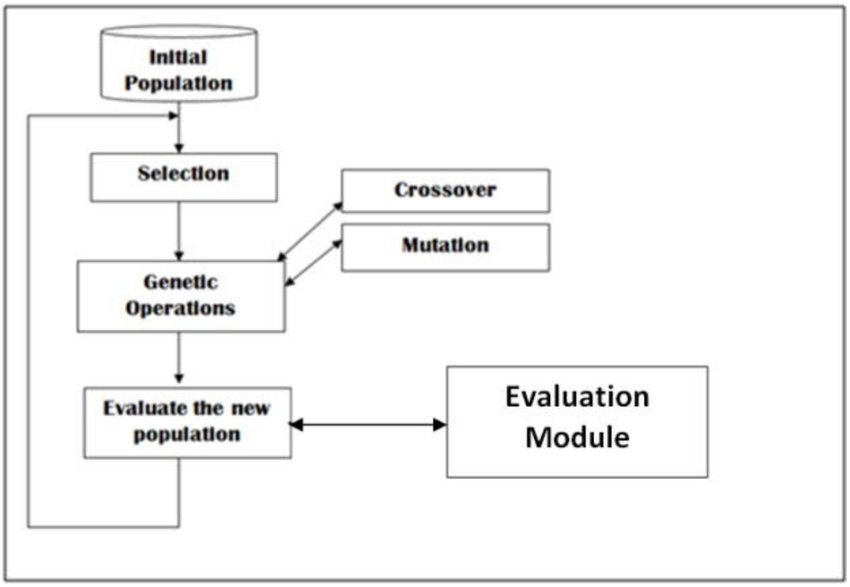


FIGURE 3. Genetic algorithm process

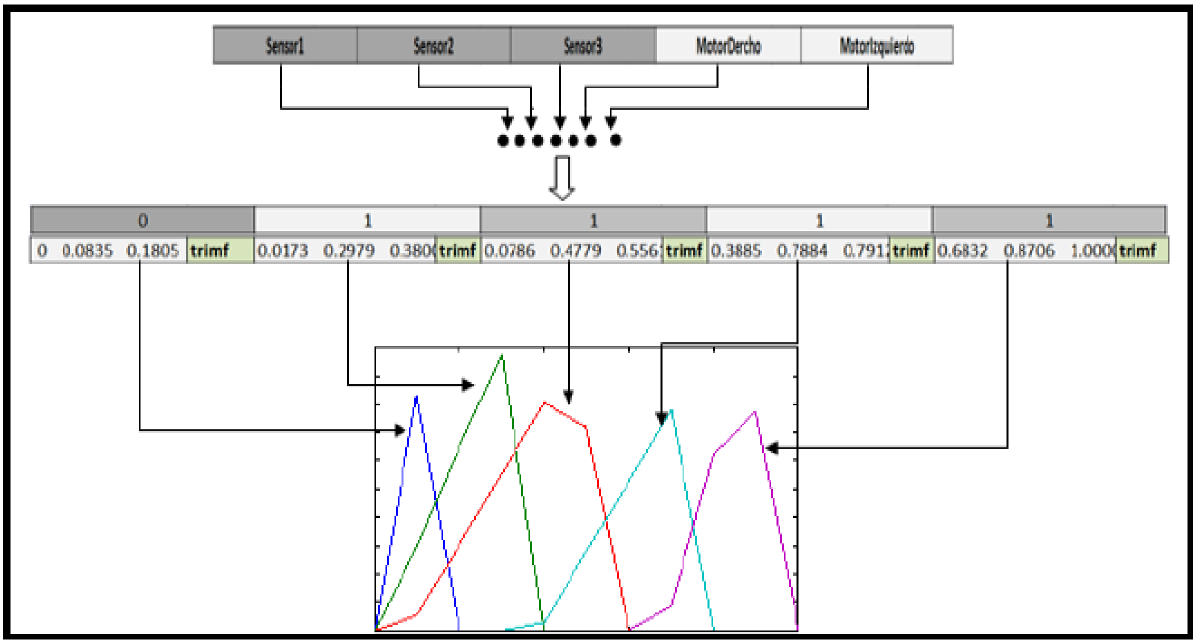


FIGURE 4. Type-1 reactive controller chromosome architecture

the robot that it must avoid while it moves around. The FISs are Mamdani interval type-2 and type-1 fuzzy system [22], each consisting of 3 inputs, which are the distances obtained by the robots sensors described on Section 2, and 2 outputs that control the velocity of the servo motors on the robot, all this information is encoded into each chromosome.

Figure 3 shows the global cycle process of the GA, under the Evaluation of each individual, where it measures the effectiveness of each of the controllers (FIS (Fuzzy Inference System)) represented by each Individual chromosome, in the test area, that will take place in a unknown environment (Maze [7]).

4.2.1. *Type-1 fuzzy reactive controller chromosome architecture.* The control genes consist of 5 bit vectors, this will indicate which fuzzy membership is or not active, the connection

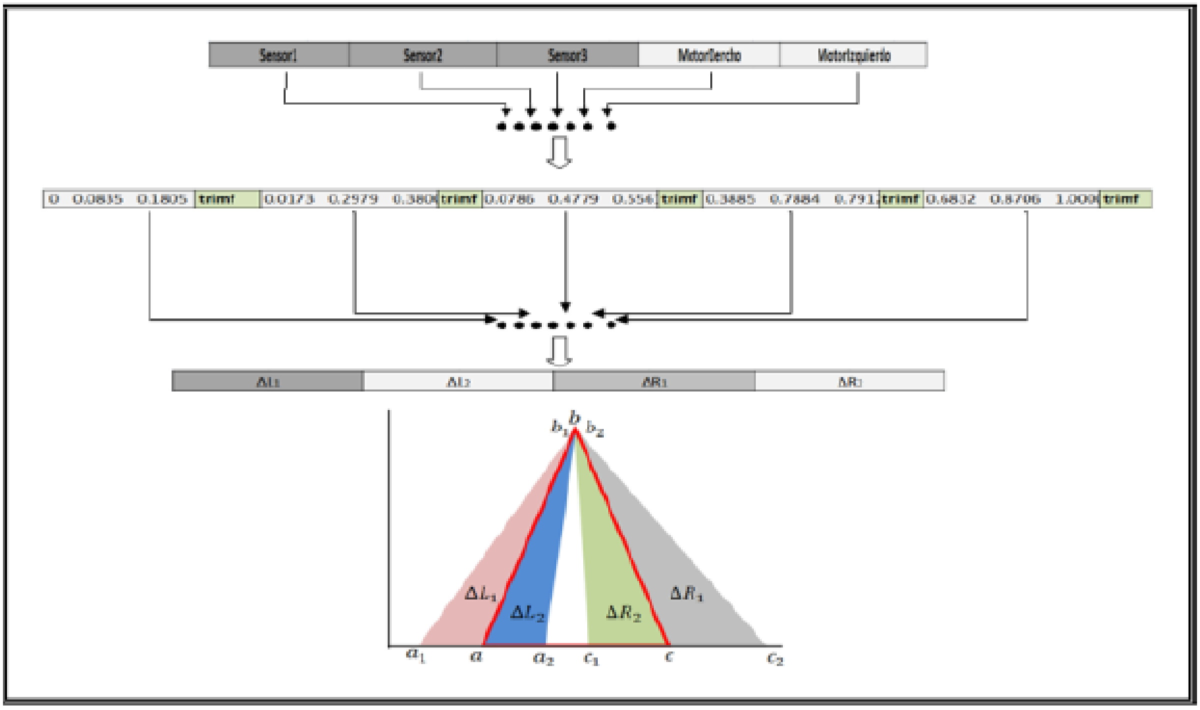


FIGURE 5. Type-2 reactive controller chromosome architecture

genes are divided in 5 subsections, 5 is the maximum number of membership functions that are allowed per variable, each of which can be trapezoidal or triangular membership function, and each of these subsection is divided into 2 sections one that indicates the type of the membership function and the other the parameters for the function; see Figure 4.

4.2.2. *Type-2 fuzzy reactive controller chromosome architecture.* The type-2 chromosome architecture is different from the type-1, here the architecture contains a type-1 FIS and 4 delta values for each membership function, keeping the type-1 FM as the base its divided into two parts Left and Right, where the start and end of the FM is considered as the origin point to apply 2 delta values to each obtaining a lower and a upper values that represent the width of the uncertainty footprint of our type-2 FM, these conversion provides a type-2 FIS from an extension of a type-1 FIS; see Figure 5.

4.3. **Tracking controller.** The tracking controller has the responsibility of keeping the robot on the correct path, this is when a reference is provided, it will move the robot to the reference and keep it on track and this is will allow the robot to move from point A to B, with in an obstacle free environment. The controller will work by keeping the error (Δep, $\Delta\theta$) to minimum values, which represents the error relative to the position and the error relative to the orientation of the front of the robot to a minimum value see Figure 7, the fuzzy system is a Mamdani type-1 FIS and consists of 2 inputs that are (Δep, $\Delta\theta$) and 2 outputs that control the velocity of the servo motors on the robot.

In Figure 6, we show the global cycle process of the GA, under the Evaluation of each individual, in which we are going to measure the effectiveness of each of the controllers (FIS (Fuzzy Inference System)) in our test area, which will take place in a closed environment with a reference given by a straight.

4.3.1. *Type-1 fuzzy tracking controller chromosome architecture.* The control genes consist of 5 bit vectors, this will indicate which fuzzy membership is or not active, the connection

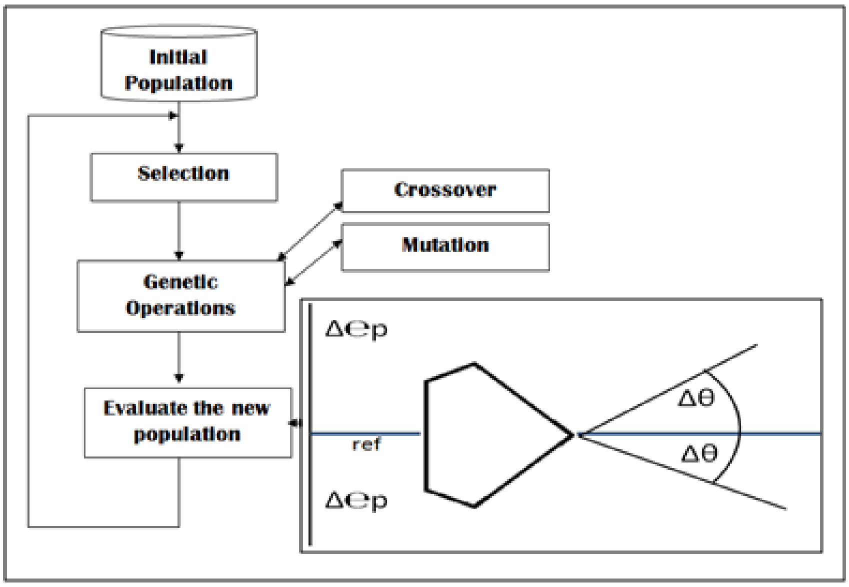


FIGURE 6. GA process for the tracking controller and FIS inputs ep, $e\theta$

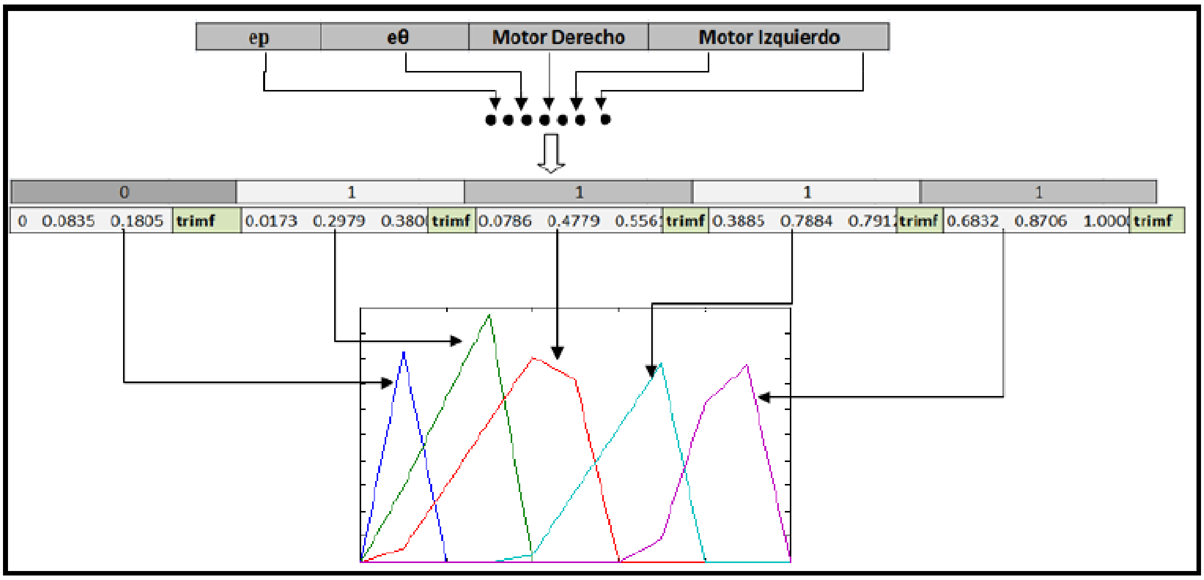


FIGURE 7. Tracking controller chromosome architecture

genes are divided in 5 subsections, 5 is the maximum number of membership functions allowed per variable, each of which can be trapezoidal or triangular, and each of these subsections is divided in to 2 sections one that indicates the type of the membership function and the other the parameters for the function. Figure 7 shows the chromosome structure.

4.4. **Fuzzy rules.** The rules population is a different and separated population for each controller with respect to the control population, this is because the optimization procedure is totally different, but they are tightly related because the number of active rules depends on the number of active membership functions. In order to optimize the fuzzy rules we have a population off all the possible subsets keeping one restriction that the number of active membership functions must be the same. With Equation (3) we obtain the size of the fuzzy rules population, where m, n, p represent the maximum number of membership functions we allow for the input variables, and k is the maximum number of membership function for the output variables, in our case ($m = n = p = k = 5$) Equation (3) gives a total of 625 fuzzy rules subsets.

$$rp = m * n * p * k \tag{3}$$

In this case, we will only have one active subset that can match the fuzzy controller that has the following membership functions active, $S_{(a,b,c,d,e)}$ where a, b, c are the number of active membership functions for the input variables and d, e for the output variable, and we use an index table for each of the fuzzy subsets, see Table 1.

Table 1. Rules index table

$S_{(1,3,2,2,2)} =$

Input 01	Input 02	Input 03	Output 01	Output 02
1	1	1	1	1
1	1	2	1	1
1	2	1	1	1
1	2	2	2	1
1	3	1	1	2
1	3	2	2	1

A special mutation operator is applied (Equation (4)) to find the optimal fuzzy rule set for the reactive controller, the shift operation that is used, changes the consequent part of the rule.

$$h_{(i,j,q,r,s)} = h_{(i,j,q,(r+\Delta r),(s+\Delta s))} \tag{4}$$

where $h_{(i,j,q,r,s)}$ is the consequent of the rule that has i, j, q, r, s. Active membership functions, Δr, Δs represent our shift operator, with a probability of 0.01.

4.5. **Objective function.** The GA will be generating individuals that will need to be evaluated and assigned a crisp value that will represent the controller performance on each of the criteria that we want to improve. For this, we need to provide the GA with a good evaluation scheme that will penalize unwanted behaviors and reward with higher fitness values those individuals that provide the performance we are looking for in our controller; if we fail to provide a proper evaluation method we can guide the population to suboptimal solutions or non solution at all.

4.6. **Reactive controller objective function.** The criteria used to measure the Reactive controller performance takes into account the following:

- Covered Distance.
- Time used to cover the distance.
- Battery life.

A Fitness FIS will provide the desired fitness value, adding very basic rules that reward the controller that provided the longer trajectories and smaller times and higher battery life. This seems like a good strategy that will guide the control population into evolving and provide the optimal control, but this strategy on its own it is not capable of doing just that, it needs to have a supervisor on the robots trajectory to make sure is a forward moving trajectory and that they do not contain looping parts. For this, a Neural Network (NN) is used to detect cycle trajectories that do not have the desired forward moving behavior by giving low activation value and higher activation values for the ones that are cycle free. The NN has two inputs and one output, and 2 hidden layers; see Figure 8.

The evaluation method for the reactive controller has integrated both parts the FIS and the NN where the fitness value for each individual is calculated with Equation (5),

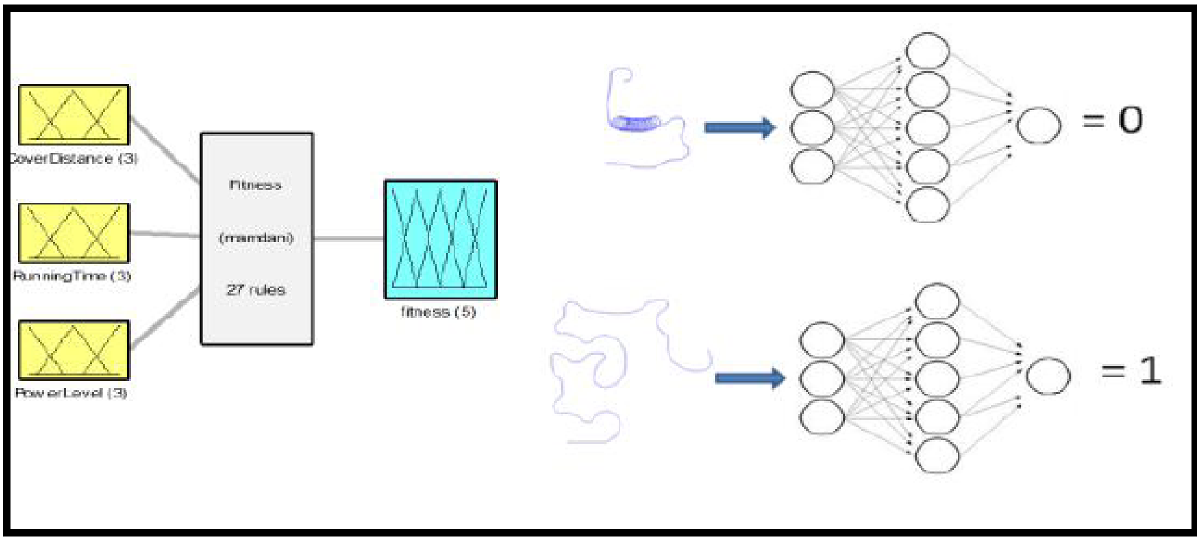


FIGURE 8. Tracking controller chromosome architecture

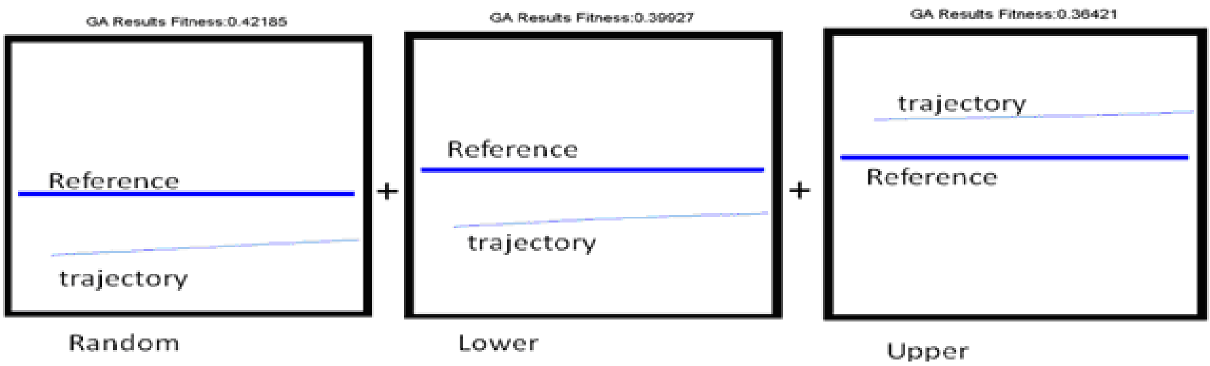


FIGURE 9. Fitness function for the reactive controller

based on the response off the NN the peak activation value is set to 0.35, this meaning that any activation lower than 0.35 will penalize the fitness given by the FIS.

$$f(i) = \begin{cases} fv * nnv, & nnv < 0.35 \\ fv, & nnv \geq 0.35 \end{cases} \tag{5}$$

where
$f(i)$ Fitness value of the i-th individual,
fv Crisp value out of the fitness FIS,
nnv Looping trajectory activation value.

4.7. **Tracking controller objective function.** The tracking controller performance is measured with the RMSE between the reference and the robots trajectory. We apply the test three times and take the average, on each of the three tests the robot and the reference vertical position is random, but we ensure that on one test the robots vertical position is above the reference and on another test is below it, we do this to ensure the controller works properly for any case the robot may need it when it is above or below (Figure 9).

5. **Simulation Results.** For the simulation experiment the GA and the evaluation process were separated into two different parts, the generic GA process was developed on the

C# language with .net 4, where a GA and Fuzzy System library was created with a GUI to setup the GA parameters, there the GA operations and cycle are run and the FIS are created. When a chromosome is ready to be evaluated it lets Matlab know and a modified version of the Simulation toolbox for mobile robots [21] is used to run each test, where the performance is measured and a Fitness value is returned to the GA process, and the communication between both process is done using a SQL server queue table.

5.1. **Type-1 reactive controller.** For the type-1 reactive controller, a GA was setup with high number of generations and a low number of population size, this because of the large solution space the reasoning behind this is that with a relative small group of individuals it will cover focused sections of the solution and can move around the space. A constrain for inputs and outputs of maximum 10 and minimum 2 FM was set, on the outputs another constrain was set and it is that the outputs had to be the same, the evaluation as described in Section 4 is based upon each individual performance on the particular maze problem.

Table 2 shows the GA configuration and the top 9 results, where we have the corresponding fitness value and the number of membership functions of each input and output, where the S represents the inputs and indicate the sensor number and M are the outputs and indicate the Motor number, and the total rules active on each controller.

Figure 10 shows the robots trajectory evolution during the execution of the GA.

5.2. **Type-2 reactive controller.** For the type-2 reactive controller a GA was setup with a smaller generation number and small population size, this because the solutions

TABLE 2. Summary of type-1 reactive controls results

	Membership Chromosome		**Fuzzy Rule Chromosome**
	Control Genes	*Connections Genes*	
Representation	Binary	Real Number	Integer
Population Size No. of Offspring	20 5		
Crossover *Crossover Rate*	One Point *1.0*	One Point *1.0*	
Mutation *Mutation Rate*	*Bit Mutation* *0.02*	*Random Mutation* *0.02*	Shift index operation
GA Parameters			
Generation	8000		
Selection	Roulette Wheel with Ranking		
Results			
Rank	Fitness	Active FM's $(S_1 + S_2 + S_3 + M_1 + M_2)$	Active Rules
1	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24
2	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24
3	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24
4	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24
5	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24
6	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24
7	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24
8	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24
9	0.4895	$(4 + 3 + 2 + 3 + 3) = 15$	24

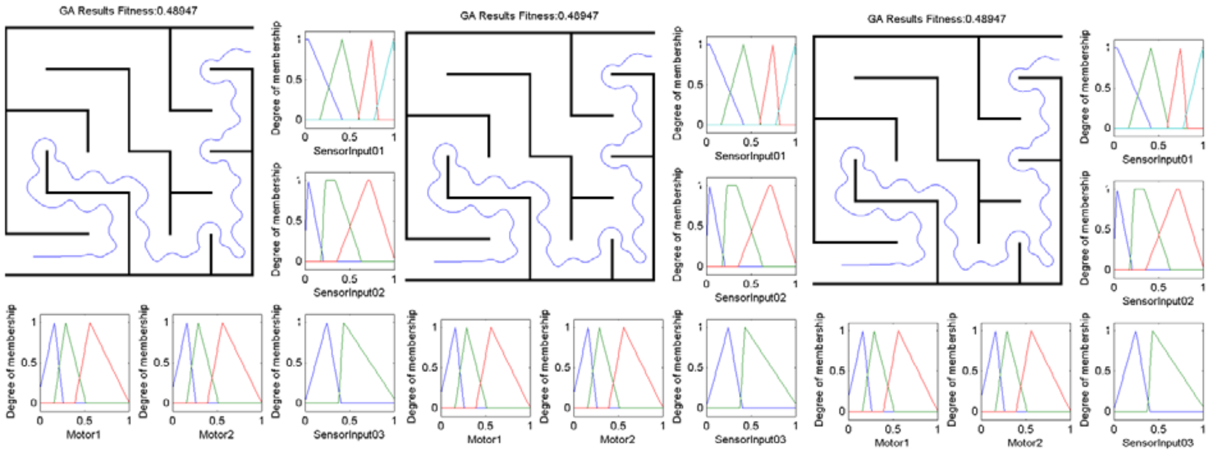


FIGURE 10. Type-1 fuzzy reactive controllers results

TABLE 3. Summary of type-2 reactive controls results

	Membership Chromosome *Connections Genes*			
Representation	Real Number			
Population Size No. of Offspring	10 5			
Crossover *Crossover Rate*	One Point *1.0*			
Mutation *Mutation Rate*	*Random Mutation* *0.02*			
Generation	200			
Selection	Roulette Wheel with Ranking			
Results				
Rank	Fitness	Active FM's $(S_1 + S_2 + S_3 + M_1 + M_2)$	Active Rules	
1	0.1701	$(4 + 3 + 2 + 3 + 3) = 15$	24	
2	0.1700	$(4 + 3 + 2 + 3 + 3) = 15$	24	
3	0.1683	$(4 + 3 + 2 + 3 + 3) = 15$	24	
4	0.1677	$(4 + 3 + 2 + 3 + 3) = 15$	24	
5	0.1669	$(4 + 3 + 2 + 3 + 3) = 15$	24	
6	0.1659	$(4 + 3 + 2 + 3 + 3) = 15$	24	
7	0.1651	$(4 + 3 + 2 + 3 + 3) = 15$	24	
8	0.1649	$(4 + 3 + 2 + 3 + 3) = 15$	24	
9	0.1649	$(4 + 3 + 2 + 3 + 3) = 15$	24	

space with this optimization approach is much smaller since we are only looking for the footprint of uncertainty on the Fuzzy membership functions, the same evaluation process as the type-1 controller is apply.

Table 3 shows the GA configuration for the reactive controller and the top 9 results, where we have the fitness value and the number of membership functions and the active rules.

Figure 11 shows the robots trajectory evolution during the execution of the GA.

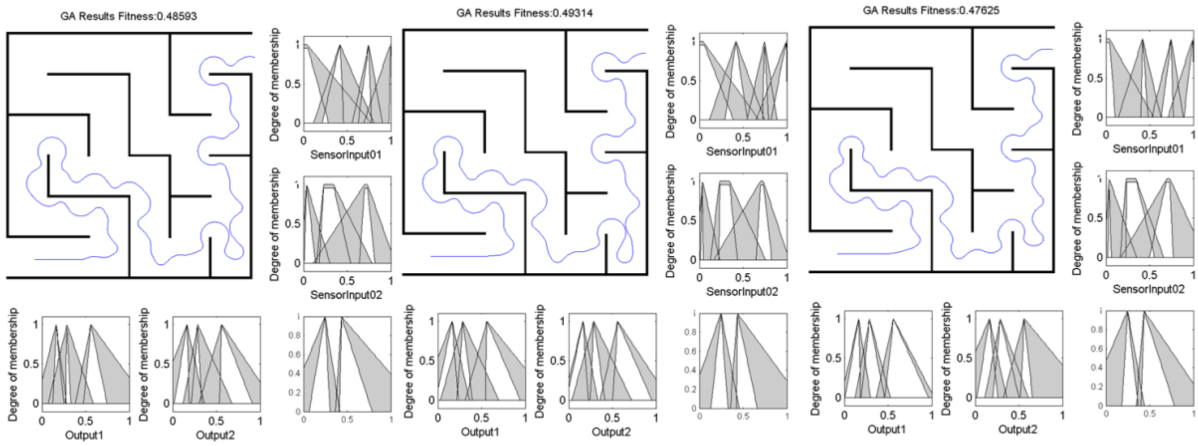


FIGURE 11. Type-2 fuzzy reactive controllers results

FIGURE 12. Type-1 fuzzy tracking controller results

5.3. **Tracking controller.** For the type-1 tracking controller, a GA was setup the same as it was for the reactive controller with high number of generations and a low number of population size, because both FIS manage the relative same number of inputs and outputs. Also a constrain for inputs and outputs of maximum 10 and minimum 2 FM was set, the evaluation as described in Section 4 is based upon a set of 3 test where the RMSE error is gather from each one and average to get the individual performance. Table 4 shows the GA configuration and the top 9 results, where we have the fitness value and the number of membership functions of each input and output, where the ep and $\Delta\theta$ represent the inputs and indicate the error on the position and orientation respectively, and M are the outputs and indicate the Motor number, and the total rules active on each controller.

Figure 12 shows the robots trajectory evolution during the execution of the GA.

6. **Conclusions.** The problem of optimizing a type-2 fuzzy reactive controller with a GA is of much greater complexity then for type-1 fuzzy logic, one reason for this is that we have a bigger solution space to handle, for this reason a two step process was used, first finding the best type-1 controller we can find and use it as our base FIS with these reducing the solution space to only find the footprint of uncertainty for that FIS.

The type-1 and type-2 fuzzy controllers both provide good results as they are able to guide the robot through the maze without hitting any wall and keep the robot on track,

TABLE 4. Summary of type-1 tracking results

	Membership Chromosome *Control Genes*	*Connections Genes*	**Fuzzy Rule Chromosome**
Representation	Binary	Real Number	Integer
Population Size No. of Offspring	20 2		
Crossover *Crossover Rate*	One Point *1.0*	One Point *1.0*	
Mutation *Mutation Rate*	*Bit Mutation* *0.02*	*Random Mutation* *0.02*	Shift index operation
GA Parameters			
Generation	4000		
Selection	Roulette Wheel with Ranking		
Results			
Rank	Fitness	Active FM's ($\text{ep} + \text{e}\theta + \text{M}_1 + \text{M}_2$)	Active Rules
1	0.1907	$5 + 4 + 4 + 3 = 16$	20
2	0.2021	$5 + 4 + 4 + 3 = 16$	20
3	0.2023	$5 + 4 + 4 + 3 = 16$	20
4	0.2091	$5 + 4 + 4 + 3 = 16$	20
5	0.2124	$7 + 2 + 4 + 5 = 18$	14
6	0.2125	$5 + 4 + 4 + 3 = 16$	20
7	0.2182	$6 + 2 + 5 + 2 = 15$	12
8	0.2199	$5 + 4 + 4 + 3 = 16$	20
9	0.225	$5 + 4 + 4 + 3 = 16$	20

however we expect that when we introduce noise into the tests, the type-2 will have better performance.

Acknowledgment. We would like to express our gratitude to CONACYT, and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

REFERENCES

[1] M. Bell, T. Toriu and S. Nakajima, Image-based robot map building and path planning with an omnidirectional camera using self-organising maps, *ICIC Express Letters*, vol.5, no.10, pp.3845-3852, 2011.
[2] L. Astudillo, O. Castillo and L. T. Aguilar, Intelligent control of an autonomous mobile robot using type-2 fuzzy logic, *Engineering Letters*, vol.13, no.2, pp.93-97, 2006.
[3] S. Ishikawa, A method of indoor mobile robot navigation by fuzzy control, *Proc. of Int. Conf. Intell Robot. Syst.*, Osaka, Japan, pp.1013-1018, 1991.
[4] A. Thomson and J. Baltes, A path following system for autonomous robots with minimal computing power, *Technical Report*, University of Auckland, Private Bag 92019, Auckland, New Zealand, 2001.
[5] M. Leyden, D. Toal and C. Flanagan, A fuzzy logic based navigation system for a mobile robot, *Proc. of Automatisierungs Symposium*, 1999.
[6] A. Meléndez, O. Castillo and J. Soria, Reactive control of a mobile robot in a distributed environment using fuzzy logic, *Annual Meeting of the North American Fuzzy Information Processing Society*, no.19-22, pp.1-5, 2008.
[7] A. Melendez, O. Castillo and J. Soria, Reactive and tracking control of a mobile robot in a distributed environment using fuzzy logic, *FUZZ, IEEE International Conference*, pp.1-5, 2010.

[8] L. Aguilar, P. Melin and O. Castillo, Intelligent control of a stepping motor drive using a hybrid neuro-fuzzy ANFIS approach, *Applied Soft Computing*, vol.3, no.3, pp.209-219, 2003.

[9] P. Melin and O. Castillo, Intelligent systems with interval type-2 fuzzy logic, *International Journal of Innovative Computing, Information and Control*, vol.4, no.4, pp.771-784, 2008.

[10] G. J. Klir and B. Yuan, Fuzzy sets, fuzzy logic, and fuzzy systems: Selected papers by Lotfi A. Zadeh, *Advances in Fuzzy Systems: Application and Theory*, vol.6, 1996.

[11] A. Aceves and J. Aguilar, A simplified version of Mamdani's fuzzy controller the natural logic controller, *IEEE Transactions on Fuzzy Systems*, vol.14, no.1, pp.16-30, 2006.

[12] H. N. Pishkenari, S. H. Mahboobi and A. Meghdari, On the optimum design of fuzzy logic controller for trajectory tracking using evolutionary algorithms, *IEEE Conference on Publication Cybernetics and Intelligent Systems*, vol.1, pp.660-665, 2004.

[13] O. Castillo and P. Melin, New fuzzy-fractal-genetic method for automated mathematical modeling and simulation of robotic dynamic systems, *IEEE International Conference on Fuzzy Systems*, vol.2, pp.1182-118, 1998.

[14] P. Melin and O. Castillo, Intelligent control of aircraft dynamic systems with a new hybrid neuro-fuzzy fractal approach, *Information Sciences*, vol.142, no.1-4, pp.161-175, 2002.

[15] P. Melin and O. Castillo, Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory, *Applied Soft Computing*, vol.3, no.4, pp.353-262, 2003.

[16] O. Castillo and P. Melin, New fuzzy-fractal-genetic method for automated mathematical modeling and simulation of robotic dynamic systems, *IEEE International Conference on Fuzzy Systems*, vol.2, pp.1182-118, 1998.

[17] M. M. Fateh and M. R. Soltanpour, Robust task-space control of robot manipulators under imperfect transformation of control space, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3949-3960, 2009.

[18] F. Cupertino, V. Giordano, D. Naso and L. Delfine, Fuzzy control of a mobile robot, *IEEE Robotics & Automation Magazine*, pp.74-81, 2006.

[19] S. Coupland, Type-2 fuzzy control of a mobile robot, *Ph.D. Transfer Report*, De Montfort University, UK, 2003.

[20] D. W. Payton, J. K. Rosenblatt and D. M. Keirsey, Plan guided reaction, *IEEE Transactions on Systems, Man and Cybernetics*, vol.20, no.6, pp.1370-1382, 1990.

[21] *Mobile Robotics Toolbox for Matlab 5*, http://www.uamt.feec.vutbr.cz/robotics/simulations/amrt/simrobot_en.html, 2001.

[22] E. Mamdani, Applications of fuzzy logic to approximate reasoning using linguistic synthesis, *Proc. of the 6th Int. Symp. on Multiple Value Logic*, Utah State University, pp.196-202, 1976.