Chemical Reaction Algorithm for Type-2 Fuzzy Control Optimization in Mobile Robots

Submitted: 25th August 2018; accepted: 20th September 2018

David de la O, Oscar Castillo, José Soria

DOI: 10.14313/JAMRIS_3-2018/14

Abstract:
In this work the optimization process of the tracking and reactive controllers for a mobile robot are presented. The Chemical Reaction Algorithm (CRA) is used to find the optimal parameter values of the membership functions and rules for the reactive and tracking controllers. In this case, we are using five membership functions in each variable of the fuzzy controllers. The main goal of the reactive controller is aimed at providing the robot with the ability to avoid obstacles in its environment. The tests are performed on a benchmark maze problem, in which the goal is not necessarily to leave the maze, but rather that the robot avoids obstacles, in this case the walls, and penalizing for unwanted trajectories, such as cycles. The tracking controller's goal is for the robot to keep into to a certain path, this in order that the robot can learn to react to unknown environments. The optimization algorithm that was used is based on an abstraction of chemical reactions. To perform the simulation we use the "SimRobot" toolbox, the results of the tests are presented in a detailed fashion, and at the end we are presenting a comparison of results among the CRA, PSO and GA methods.

Keywords: *Chemical Reaction Algorithm, control, fuzzy logic, robotics*

1. Introduction

Lofti Zadeh (1965) proposed fuzzy logic and rules-based procedures as a means to model and capture the human knowledge and deal with uncertainty in the real word. These methods have been applied to ill defined industrial processes, since these methods are usually based on experienced people who usually obtain good results, regardless of whether they receive imprecise information [10–14, 23]. The methods have also been applied to Control of a Mobile Robot using Fuzzy Bee Colony Optimization Algorithm [2, 8], Particle Swarm [3, 16–18, 21–24, 26, 42–46], Genetic Algorithms [11, 15, 29, 47], Differential evolution [30] and Ant Colony Optimization [25, 30, 37, 41]. The origin of these impreciseness can be related to a variation of time concerning the application of a control signal and the warning of its effect [2], and nonlinearities in the dynamics of the system or sensor degradation [21]. The processes in which the fuzzy rule-based approximation have been applied include the automated process of the Operation of a Public Transport System [38], water tank control [1], [20] and sewage treatment plants [47].

We use the word fuzzy because fuzzy systems have to be precisely defined, a fuzzy controller operates as a non-linear controller that is defined with precision. Essentially what we want to emphasize is that although the phenomenon described by this theory may be fuzzy, the theory itself is accurate.

The CRA optimization algorithm was originally developed by Astudillo *et al.* [6], which is based on a metaheuristic of a population that does not change in size, in addition to applying a generalization of chemical reactions as exploration and exploitation of mechanisms. The algorithm uses chemical reactions by changing at least one of the substances (element or compound), changing their composition and sets of properties. We take as a basis the tests performed by Melendez *et al.* [32–36] and de la O [13], in which a fuzzy system is designed for the navigation of an Automonous Mobile Robot, it uses 2 controllers, a reactive controller and a tracking controller, and then optimizes the parameters and rules of the controllers using GA [32–36].

This work is organized as follows: Section 2 refers to the concepts of fuzzy logic systems, Section 3 describes the Chemical Optimization Paradigm used in the present paper. Sections 4 and 5 define the fundamental methodology of this work and the benchmark functions used. Section 6 shows the results of the simulations, comparisons and Section 7 presents the conclusion.

2. The CRA Paradigm

The algorithm of chemical reactions was developed by Astudillo *et al.* in 2011 [6], this algorithm is a new paradigm which is inspired by natural behavior of the chemical reactions, makes the population come together to find an optimal result in the search space supported by several intensifier/diversifier mechanisms.

One might think that chemical theory and it is descriptions are difficult and that have no relation with the optimization theory, but only the general scheme is considered as the basis of the chemical reaction optimization algorithm.

Astudillo *et al*. [4–7], defined the elementary terminology for characterizing and classifying artificial chemicals. Because the laws of reaction and representation of the elements/compounds are of statistical and qualitative character, then the algorithm is a representation of the general procedure of the chemical reactions. The initial description of the elements/compounds depends on the problem. These elements/compounds can be symbolized as binary, integer, floating, numbers etc.

The relationship between the elements/compounds is indirect: The interaction does not take into account the rules of interaction and molecular structure and as a consequence does not include values of temperature, pH, pressure, etc.

The Chemical reaction algorithm is a metaheuristic that explores all possible solutions that exist for a defined search space. This optimization algorithm uses an element (or compound) to represent a possible solution for a problem and the objective function measures the performance capacity of the element. The algorithm ends when the objective is achieved or the number of scheduled iterations have been reached.

The CRA does not use the external values (conservation of masses, thermodynamic characteristics, etc.), and this represents an advantage when compared to other optimization algorithms, as it is a very direct method which takes into account the main features of chemical reactions (synthesis, decomposition, substitution and double substitution) to obtain the optimal solution in a search space.

2.1. Elements or Compounds

The algorithm makes an analogy to natural chemical reactions, therefore it represents a possible solution to the problem using an element, which is initialized with values that depend on the problem to solve, and these values can be binary numbers, integers, floating, etc. These elements will interact with each other indirectly. That is, the interaction is independent of the actual molecular structure. This approach does not take into account other molecular properties, such as potential and kinetic energies, among others.

2.2. Chemical Reactions

A Chemical reaction is a chemical process in which the two substances, the so-called reactants, by the action of an energy factor, become other substances designated as compounds. Taking this process into account, chemical reactions as intensifying (substitution, double substitution reactions) and diversification (synthesis, decomposition reactions) mechanisms can be used. These four chemical reactions considered in this approach are synthesis, decomposition, single and double substitution. With these operators new solutions within a defined search space can be explored, the algorithm allows to define the percentage of the elements to be evaluated in each chemical reaction, it can be from 0% to 100%.

2.2.1. *Combination Reactions*

In this type of reactions, two of the substances that can be elements or compounds are combined to form the product. Reactions of this type are classified as combining synthesis, and are generally represented as follows:

$$B + X \rightarrow BX \tag{1}$$

2.2.2. *Decomposition Reactions*

In a decomposition reaction, a single substance decomposes or breaks, producing two or more distinct substances. The starting material must be a compound and the products can be elements or compounds. The general form of this equation is the following:

$$BX \rightarrow B + X \tag{2}$$

2.2.3. *Substitution Reactions*

In a simple substitution reaction an element reacts with a compound and takes the place of one of the elements of the compound, producing a different element and also a different compound. The general formula for this reaction is:

$$X + AB \rightarrow AX + B \tag{3}$$

2.2.4. *Double-substitution Reactions*

In a double substitution reaction, two compounds exchange pairs with each other to produce distinct compounds. The general form of these equations is [4]:

$$AB + CD \rightarrow AC + BD \tag{4}$$

The flowchart for this optimization method can be found in Figure 1, and the following list of steps is presented:

- We start by generating an initial set of elements/compounds.
- We evaluate the original set of elements, to obtain a fitness values.
- Based on the above evaluation, we select some of the elements/compounds to "induce" a reaction.
- Taking into consideration the result of the reaction, evaluations of the news element/compounds are obtained and selected elements are those of greater fitness
- Repeat the steps until the algorithm meets the terminating criteria (the desired result in the maximum number of iterations is reached) [6].

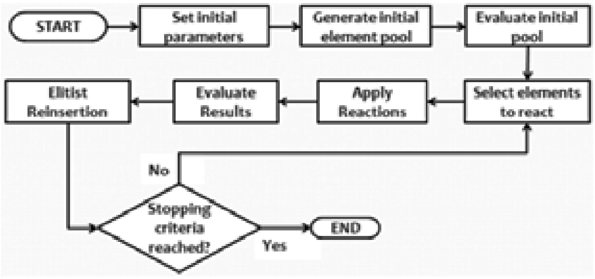


Fig. 1. Flowchart of the CRA

This algorithm consists of a metaheuristic based on a static population, and applies an abstraction of chemical reactions as intensifying mechanisms and diversification. It also uses an elitist reinsertion strategy which allows for the perpetuity of the best elements and, therefore, the average fitness of the whole set of elements increases with each iteration.

The reactions of synthesis and decomposition are used for exploration in the search space of the solutions: These procedures demonstrate to be effective and promptly lead to the results of a desired optimal value.

The single and double substitution reactions allow the algorithm to search for obtaining optimal values around a previously found solution.

We start the algorithm by randomly generating a set of elements/compounds under the uniform distribution space of possible solutions, and this is represented as follows:

$$X = \{x_1, x_2, \ldots, x_n\}, \tag{5}$$

Where x_n is used to represents the element/compound.

The total number and the representation of the original elements depend on the complexity of the problem that is solved.

In order to find the best possible controllers we use a metaheuristic strategy, which has proven to produce good results, and this is achieved by applying the CRA (see Fig. 2). In this case the algorithm will search the solution space of the problem to be solved. Combining the values of the best controllers and generating new controllers. The goal is to optimize the parameters of the membership functions and fuzzy rules.

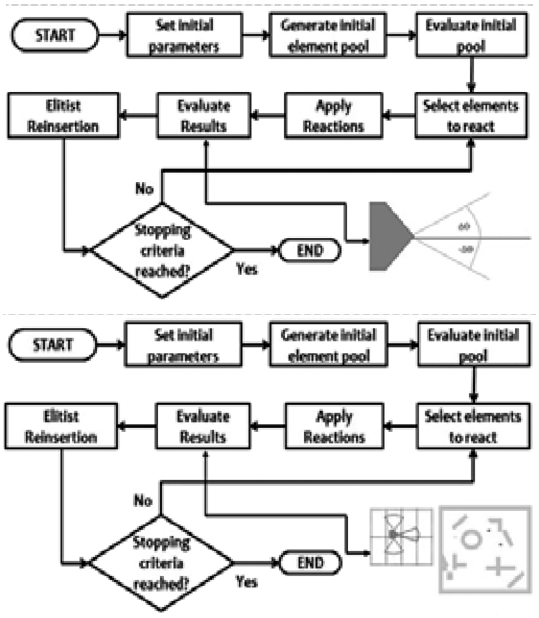


Fig. 2. General flowchart of the chemical reaction algorithm optimizing the fuzzy controllers

3. Description of the Tool for Simulation

The SIMROBOT toolbox software [30] enables performing robot simulations and is used for testing the fuzzy controllers. The mobile robot has two controlled and sensed wheels, in addition to an uncontrolled and un-sensed wheel. Figure 3 illustrates this type of robot. This robot has two degrees of freedom: y-translation and x-translation or z-rotation.

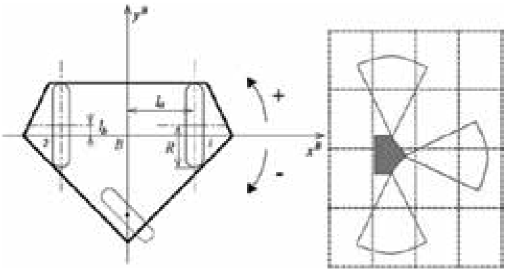


Fig. 3. Kinematic coordinate system [27]

The kinematic equations of the mobile robot are as follows:

Equation 6 shows the sensed forward velocity solution

$$\begin{pmatrix} V_{B_x} \\ V_{B_y} \\ \omega_{B_z} \end{pmatrix} = \frac{R}{2l_a} \begin{bmatrix} -l_b & l_b \\ -l_a & -l_a \\ -1 & -1 \end{bmatrix} \begin{pmatrix} \omega_{W_1} \\ \omega_{W_2} \end{pmatrix} \tag{6}$$

Equation 7 shows the Actuated Inverse Velocity Solution:

$$\begin{pmatrix} \omega_{W_1} \\ \omega_{W_2} \end{pmatrix} = \frac{1}{R\left(l_b^2+1\right)} \begin{bmatrix} -l_a l_b & -l_b^2 \\ l_a l_b & -l_b^2 \end{bmatrix} \tag{7}$$

where (in the metric system):

V_{B_x}, V_{B_y} are the translational velocities of the robot's body [m/s],

ω_{B_z} is the robot's z-rotational velocity [rad/s],

V_{B_x}, V_{B_y} are the wheels' rotational velocities [rad/s],

R the actuated wheel radius [m],

l_a, l_b are the distances between the wheels and the robot's axes [m].

4. Simulations and Tests

Two different control tests were performed to experiment with the performance of the algorithm in control problems, and each test is described as follows.

We used the Chemical Reaction Algorithm (CRA) to optimize the parameters and rules of the reactive and tracking fuzzy controllers, one test was to optimize only the parameters of the fuzzy controller, leaving the controller's fuzzy rules fixed. A second test was to optimize parameters and fuzzy controller rules, and for this test we executed two CRAs simultaneously, one that optimizes the parameters and another one that optimizes the rules, alternating in each iteration.

The CRA performs the task of initializing the parameters of each fuzzy controller, selecting the elements and the chemical reactions that will be applied, evaluating the results and, through the simulation, performing an elitist reinsertion and putting them back into the population.

4.1. Reactive Controller

The reactive control has the purpose to achieve the same capability that a person has when is driving, that is, to react to unanticipated circumstances, road traffic congestions, traffic of the signs, etc., but in a more elementary level. We use a maze to test possible solutions, in the which the objective is not to guide the robot through the maze to the exit, but rather obstacle avoidance. The objective is to optimize the robot controller to find the maze output, use the maze to optimize the reactive control due to the characteristic of the situation of the simulation, i.e. it is a confined space in which the robot can not move easily and each wall is considered as an obstacle for the robot to avoid them while moving. We use a Mamdani type FIS, which consists of 3 inputs, that are the distances obtained by the robot sensors, as mentioned in Section 2, and 2 outputs that control the speed of the servo motors in the robot, and all this information is encoded in each element.

4.1.1. *Reactive Controller with Type-1 Fuzzy Logic*

We encode each membership function of the fuzzy reactive controller, represented by an element, into 25 positions of a vector of real values, which represents the values of each parameter of the triangular membership function, which has five membership functions in each of its variables (see Fig. 4).

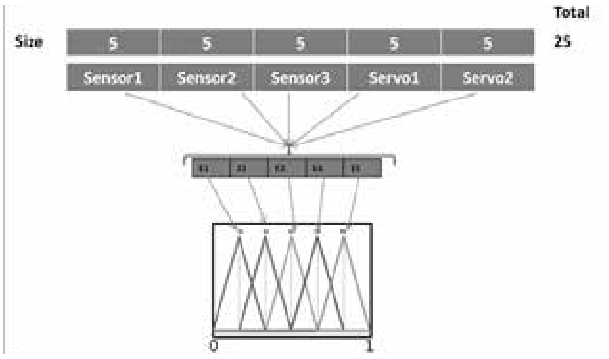


Fig. 4 Structure of the element to fuzzy parameters

We encode the values of the rules of the fuzzy reactive controller, represented by an element, into 250 positions of a vector of integer values, which represent the value of the rules set of the fuzzy controller (see Fig. 5).

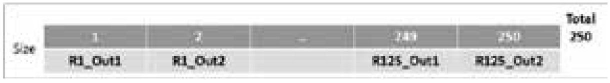


Fig. 5. Structure of the element to fuzzy rules

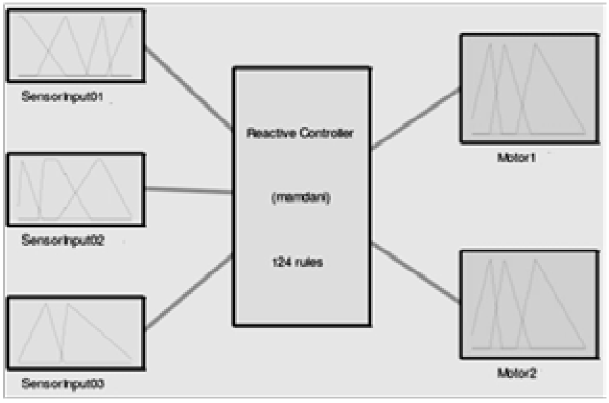


Fig. 6. Fuzzy reactive control inputs

The controller is a Mamdani fuzzy system and it has 3 inputs (the sensors of the robot) and two outputs which control the speed of each servomotor of the robot. This is illustrated in Figure 6.

4.1.2. *Reactive Controller with Type-2 Fuzzy Logic*

We encode each membership function of the fuzzy reactive controller, represented by an element, into 50 positions of a vector of real values, which represent the values of each parameter of the triangular membership function, which has five membership functions in each of its variables (see Fig. 7).

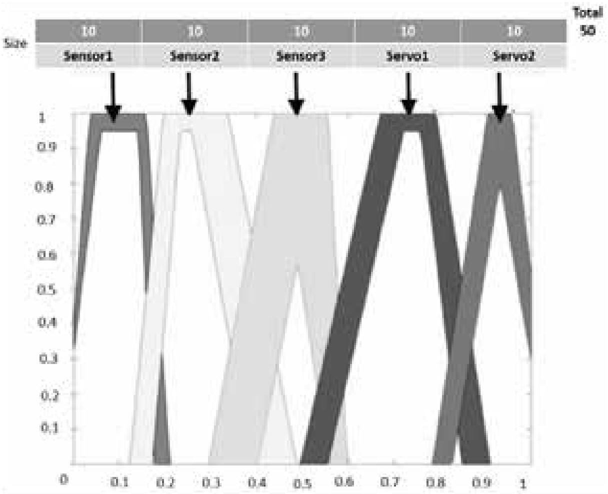


Fig. 7. Element Encoding to fuzzy parameters

We encode the values of the rules of the fuzzy reactive controller, represented by an element, into 250 positions of a vector of integer values, which represent the values of the rules set of the fuzzy controller (see Fig. 8).

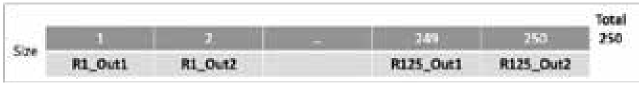


Fig. 8. Structure of the element to fuzzy rules

The controller is a Mamdani fuzzy system and it has 3 inputs (the sensors of the robot) and two outputs which control the speed of each servomotor of the robot, respectively. This is illustrated in Figure 9.

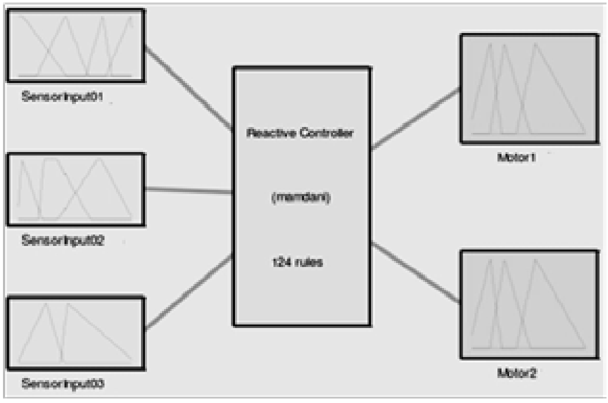


Fig. 9. Fuzzy reactive control inputs

4.2. Tracking Controller

The goal of the tracking controller is to keep the robot on the right path, in a given reference. The robot will be able to move about the reference and stay on the road, being able to move from point A to B, without obstacles present in the path.

4.2.1. *Tracking Controller with Type-1 Fuzzy Logic*

We encode each membership function of the fuzzy tracking controller, represented by an element, into 20 positions of a vector of real values, which represents the values of each parameter of the triangular membership function, which has five membership functions in each of its variables (see Fig. 10).

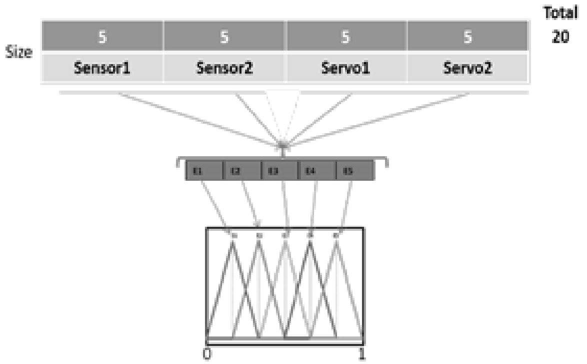
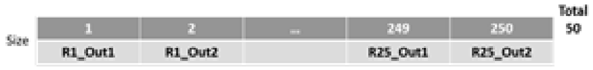


Fig. 10. Structure of the element to fuzzy parameters

We encode the values of the rules of the fuzzy reactive controller, represented by an element, into 50 positions of a vector of integer values, which represent the values of rules set of the fuzzy controller (see Fig. 11).



						Total
Size	1	2	...	249	250	50
	R1_Out1	R1_Out2		R25_Out1	R25_Out2	

Fig. 11. Structure of the element to fuzzy rules

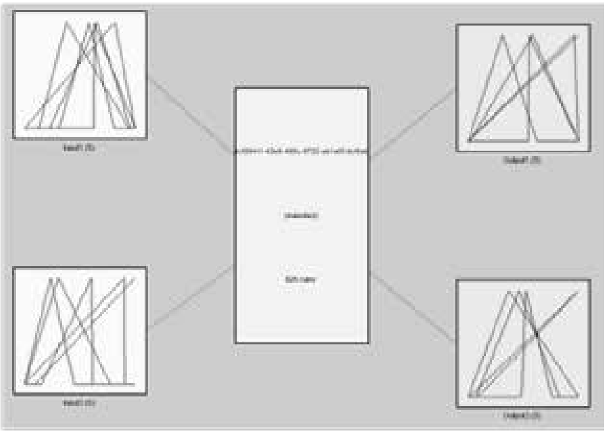


Fig. 12. Fuzzy controller inputs ⅇp, ⅇθ

The controller will take into account the errors (Δep, Δθ) in its minimum values, Figure 12, the minimum values to which are refert to the relative error of the orientation of the left front and the relative error of the position. We used a Mamdani Fuzzy system and its 2 inputs are (Δep, Δθ) and two outputs which control the speed of each servomotor of the robot and this is illustrated in Figure 12.

To calculate the performance of the controller we use the equation of the mean square error between the reference and the path of the robot.

4.2.2. *Tracking Controller with Type-2 Fuzzy Logic*

We encode each membership function of the fuzzy tracking controller, represented by an element, into 40 positions of a vector of real values, which represent the values of each parameter of the triangular membership function, which has five membership functions in each of its variables (see Fig. 13).

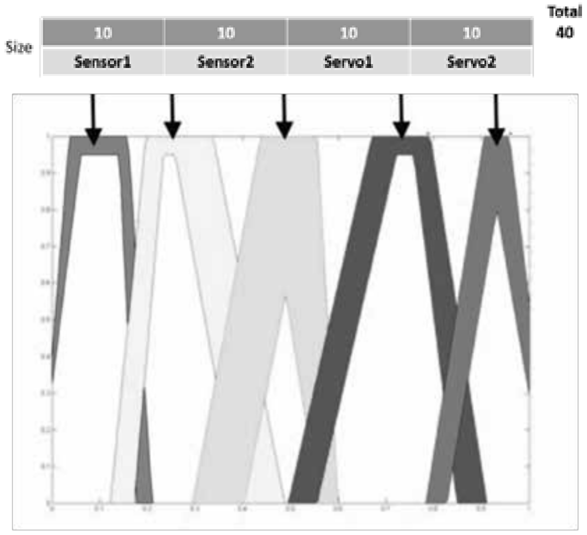
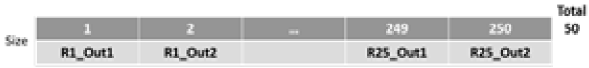


Fig. 13 Structure of the element to fuzzy parameters

We encode the values of the rules of the fuzzy reactive controller, represented by an element, into 50 positions of a vector of integer values, which represent the values of rules set of the fuzzy controller (see Fig. 14).



						Total
Size	1	2	...	249	250	50
	R1_Out1	R1_Out2		R25_Out1	R25_Out2	

Fig. 14 Structure of the element to fuzzy rules

The controller will take into account the errors (Δep, Δθ) in its minimum values, Figure 7, the minimum values to which we refer are the relative error of the orientation of the left front and the relative error of the position. We used a Mamdani Fuzzy system and its 2 inputs are (Δep, Δθ) and two outputs which control the speed of each servomotor of the robot and this is illustrated in Figure 15.

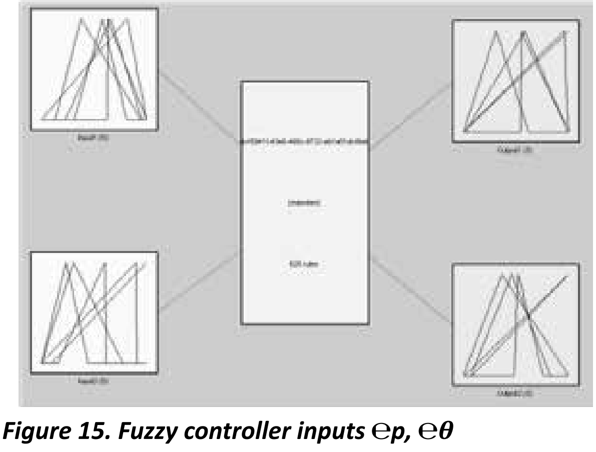


Figure 15. Fuzzy controller inputs ⅇp, ⅇθ

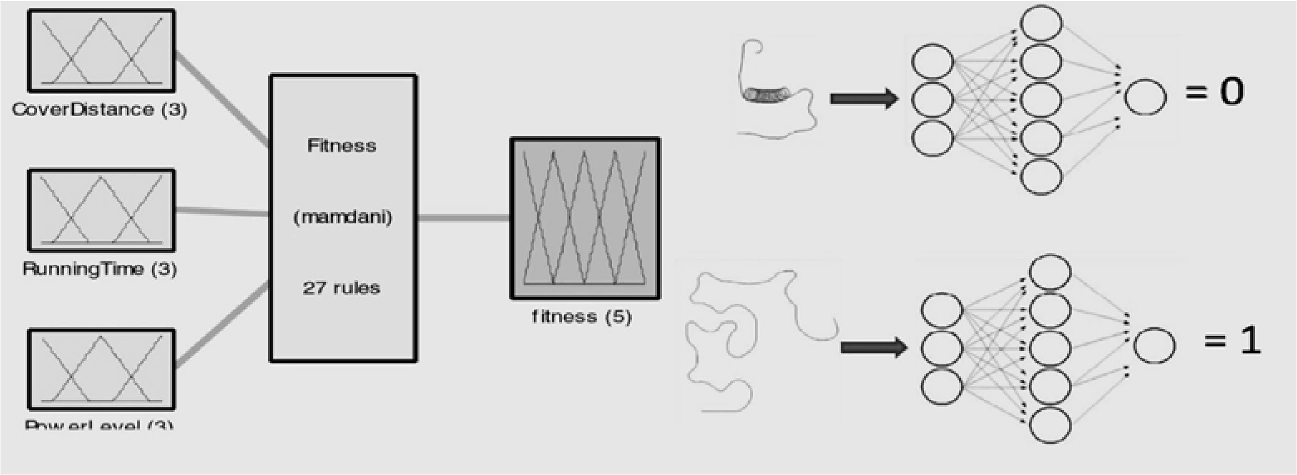


Figure 16. Fitness Function for the Reactive Controller

To calculate the performance of the controller we use the equation of the mean square error between the reference and the path of the robot.

4.3. Objective Function for Both Controllers

The CRA starts by creating elements to be evaluated by the Simrobot toolbox which will assign a crisp value that will represent the performance of the controller taking into account the criteria that we want to achieve. To achieve this, we must provide the CRA with a good evaluation criterion which is capable of penalizing undesirable behaviors and rewarding with higher fitness values those elements that yield the performance we desire in the controller. If we do not provide a correct evaluation method, we can guide the population of elements to suboptimal solutions or even not to a solution at all [5], [17]–[21], [31]. The algorithm has fixed parameters for the chemical reactions, for our tests and based on the proposed by Astudillo *et al.*, we use for each reaction a value of 0.2, which corresponds to take 20% of the amount of elements and react in each of the 4 reactions.

4.3.1. *Reactive Controller Objective Function*

In order to measure the performance of the controller, we will use the following criteria:

- Distance traveled,
- Time used to travel the distance,
- Battery life.

In order to measure these criteria we will use a Fitness FIS, which will provide the desired fitness value, adding very basic fuzzy rules that will give greater fitness to the controller that provided the longer trajectories in smaller times and a longer battery life. This seems to be a good strategy that will guide the algorithm to evolve and provide optimal control, but we have noticed that this strategy is not able to do just that on its own: it is also necessary to have a robot trajectory supervisor to make sure that there is a forward movement path and free of loops. For this purpose, it uses a neural network (NN) that is capable of detecting trajectories with cycles that do not have the desired forward displacement behavior, assigning a low activation value and higher activation values to those that are cycle free. The NN consists of two inputs and one output, and two hidden layers, see Figure 16.

To perform the evaluation of the reactive controller we will use the method of integrating both the FIS and the NN where the final fitness value for each element will be calculated with Equation 7. Taking into account the NN response, the activation value is set to 0.35, this means that any activation less than 0.35 will be penalized in the ability given by the FIS.

Equation 8 expresses how to calculate the fitness value of each individual

$$f(i)=\begin{cases} fv^{*}nnv, nnv<0.35 \\ fv, nnv\geq 0.35 \end{cases} \tag{8}$$

where:
fi – fitness value of the i-th individual,
fv – crisp value out of the fitness FIS,
nnv – looping trajectory activation value.

4.3.2. *Tracking Controller Objective Function*

To measure the performance of the tracking controller we use the root-mean-square error (RMSE) between the given reference and the path achieved by the robot. We will perform the test three times for each element and take the average of the three tests. The initial position of the robot with respect to the reference is random, but it ensures that a vertical position of the robots is above the reference and in another test it is below it (Fig. 17) [32–36].

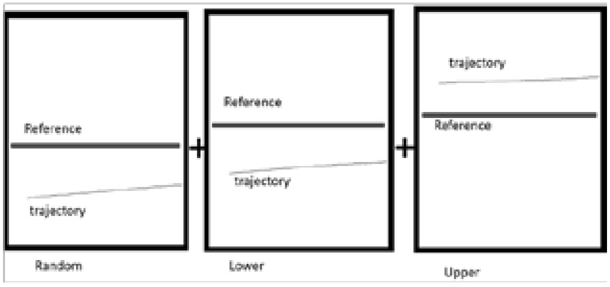


Fig. 17. Fitness Function for the Tracking Controller

5. Simulation Results

We present the results of the tests performed for each of the controllers: reactive and tracking. In order to perform these tests we have used as mentioned before the SimRobot Software and the Matlab language. To determine the suitability of each controller we use the simulation software. In the tests of the reactive controller, the robot must be able to react in a closed environment, avoiding hitting the obstacles (walls). In the tracking controller test the robot must be able to stay above the given reference.

The results will be presented in two subsections:

- Reactive Controller,
- Tracking Controller.

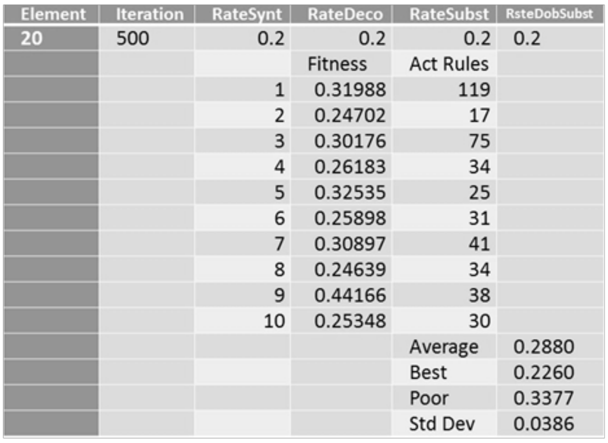
5.1. Reactive Controller

In this section, we show the results of the tests with the reactive controller, and to determine the suitability of each controller we use the simulation tool. In the tests of the reactive controller the robot must be able to react in a closed environment, avoiding hitting the obstacles (walls).

5.1.1. *Reactive Controller with Type-1 Fuzzy Logic*

We can find the configuration of the CRA and the results of the simulation tests in Table 1, where we can find the fitness value obtained in each of the experiments. We can also find statistical values which are the mean, variance, best and the worst obtained values.

Table 1. Summary of type 1 reactive controls results



Element	Iteration	RateSynt	RateDeco	RateSubst	RsteDobSubst
20	500	0.2	0.2	0.2	0.2
			Fitness	Act Rules	
		1	0.31988	119	
		2	0.24702	17	
		3	0.30176	75	
		4	0.26183	34	
		5	0.32535	25	
		6	0.25898	31	
		7	0.30897	41	
		8	0.24639	34	
		9	0.44166	38	
		10	0.25348	30	
				Average	0.2880
				Best	0.2260
				Poor	0.3377
				Std Dev	0.0386

5.1.2. *Reactive Controller with Type-2 Fuzzy Logic*

We can find the configuration of the CRA and the results of the simulation tests in Table 2, where we can find the fitness value obtained in each of the experiments. We can also find statistical values which are the mean, variance, best and the worst values obtained.

5.2. Tracking Controller

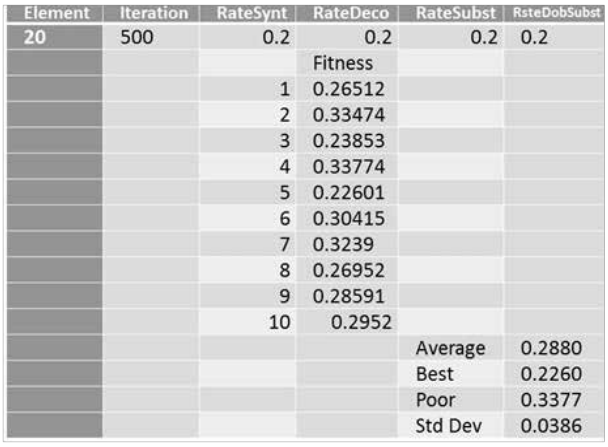
In this section, we show the results of the tests of the tracking controller, and to determine the suitability of each controller we use the simulation tool. The goal of the tracking controller is to keep the robot on the right path, with respect to a given reference.

Table 2. Summary of type 2 reactive controls results

Element	Iteration
20	1000
	Fitness
1	0.3299
2	0.3226
3	0.3143
4	0.3126
5	0.33
6	0.3607
7	0.3304
8	0.3299
9	0.3179
10	0.33
Average	0.32783
Best	0.3607
Poor	0.3126
Std Dev	0.01352

We can find the configuration of the CRA and the results of the simulation tests in Table 3, and we can also find the fitness value obtained in each of the experiments. We can also notice statistical values which are the mean, variance, best and the worst obtained values.

Table 3. Summary of Tracking Results



Element	Iteration	RateSynt	RateDeco	RateSubst	RsteDobSubst
20	500	0.2	0.2	0.2	0.2
			Fitness		
		1	0.26512		
		2	0.33474		
		3	0.23853		
		4	0.33774		
		5	0.22601		
		6	0.30415		
		7	0.3239		
		8	0.26952		
		9	0.28591		
		10	0.2952		
				Average	0.2880
				Best	0.2260
				Poor	0.3377
				Std Dev	0.0386

We can state that the results are good, because the average of the results is 0.2880, the best result is 0.2260, and when comparing with other algorithms (i.e. PSO) the result of the CRA is better.

5.3. Comparison of Results

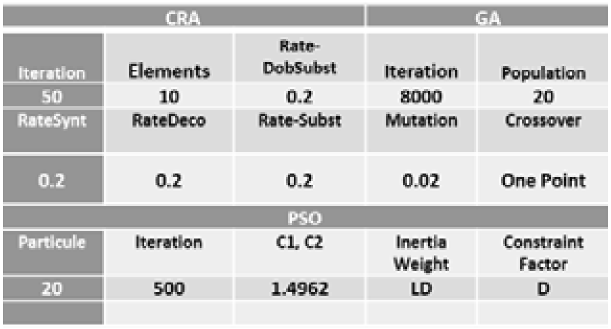
We compare against Melendez *et al.* [4], and Tables 4 and 5 summarize the results presented in [4], where we have added the results obtained using the CRA.

5.3.1. *Reactive Controller*

In this section we do a comparison of the CRA, Genetic Algorithm(GA) and Particle Swarm Optimization.

We can find the parameters used in the CRA, GA and PSO in Table 3, we note that the parameters that the CRA uses less iterations than the GA and PSO, and also a smaller population size.

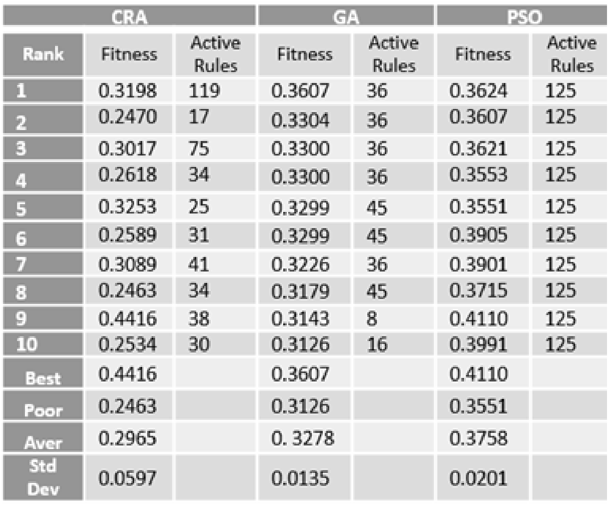
Table 4. Parameters of the CRA, GA and PSO



CRA			GA	
Iteration	Elements	Rate-DobSubst	Iteration	Population
50	10	0.2	8000	20
RateSynt	RateDeco	Rate-Subst	Mutation	Crossover
0.2	0.2	0.2	0.02	One Point
PSO				
Particule	Iteration	C1, C2	Inertia Weight	Constraint Factor
20	500	1.4962	LD	D

We can find the results of CRA, GA and PSO in Table 5, where we have the best, the worst, standard deviation and the average of each method.

Table 5. Comparison of Results of CRA and GA



	CRA		GA		PSO	
Rank	Fitness	Active Rules	Fitness	Active Rules	Fitness	Active Rules
1	0.3198	119	0.3607	36	0.3624	125
2	0.2470	17	0.3304	36	0.3607	125
3	0.3017	75	0.3300	36	0.3621	125
4	0.2618	34	0.3300	36	0.3553	125
5	0.3253	25	0.3299	45	0.3551	125
6	0.2589	31	0.3299	45	0.3905	125
7	0.3089	41	0.3226	36	0.3901	125
8	0.2463	34	0.3179	45	0.3715	125
9	0.4416	38	0.3143	8	0.4110	125
10	0.2534	30	0.3126	16	0.3991	125
Best	0.4416		0.3607		0.4110	
Poor	0.2463		0.3126		0.3551	
Aver	0.2965		0. 3278		0.3758	
Std Dev	0.0597		0.0135		0.0201	

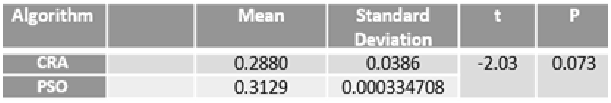
To compare the algorithms we performed an ANOVA test for the three samples with a significance level a of 0.05. Using the following parameters for the ANOVA test.
H0: All means are equal
H1: At least one mean is different
$\alpha = 0.05$.

To perform the analysis, the variances are assumed to be the same. We can conclude that the results of the ANOVA test is to reject the null hypothesis. Because of this, the Tukey test is also performed. Tukey's comparisons indicated with 95% confidence that PSO was statistically better than GA and CRA, which were assumed to be statistically the same.

5.3.2. *Tracking Controller*

We can find the parameters used for CRA, GA and PSO in Table 6, we can note that the parameters of CRA uses less iterations than the GA and PSO, and also a smaller population size.

Table 6. Comparison of Results of CRA and PSO



Algorithm		Mean	Standard Deviation	t	P
CRA		0.2880	0.0386	-2.03	0.073
PSO		0.3129	0.000334708		

6. Conclusion

In the present work, we used the CRA to optimize the parameters and rules of the fuzzy controllers, both for reactive and tracking behaviors. The reactive controller aims at giving the robot the ability to avoid obstacles in its environment. The tests were performed in a maze, in which the goal is not to leave the maze, but that the robot avoids obstacles, in this case the walls, and penalizing the unwanted trajectories as cycles. The tracking controller's goal is for the robot to be able to stay on a certain path, this test was performed 3 times for each element, this in order that the robot can react to unknown environments. After performing the tests, analyzing and comparing the results, we can notice that the algorithm is not statistically better GA, and its performance is similar to the PSO, because it is a newly created algorithm. We propose the following tasks that could be performed to improve the performance of the algorithm: to use a fuzzy controller that controls the parameters of the chemical reactions, since these values are fixed during the execution of the test, with this we will be able to do at the beginning an exploration and at the end focus on the exploitation. It is important to mention that the advantage of the algorithm is that we use it for minimizing time, less population (elements) and less iterations.

AUTHORS

David de la O, Oscar Castillo*, José Soria –
Tijuana Institute of Technology, 22379, Tijuana, Mexico.
Emails: ddsh@live.com, ocastillo@tectijuana.mx, jsoria57@gmail.com

*Corresponding author.

REFERENCES

1. Amador-Angulo, L., Castillo, O., *Comparison of the optimal design of fuzzy controllers for the water tank using ant colony optimization.* Series: Recent Advances on Hybrid Approaches for Designing Intelligent Systems. Springer International Publishing, 2014, 255–273. DOI: 10.1007/978-3-319-05170-3_18.
2. Amador-Angulo, L., Castillo, O., "A Fuzzy Bee Colony Optimization Algorithm Using an Interval Type-2 Fuzzy Logic System for Trajectory Control of a Mobile Robot". In: *Mexican International Conference on Artificial Intelligence*, Springer International Publishing, October 2015. DOI: 10.1007/978-3-319-27060-9_38.
3. Amin S., Adriansyah A., "Particle Swarm Fuzzy Controller for Behavior-based Mobile Robot". In: ICARCV'06. *9th International Conference on Control, Automation, Robotics and Vision*, Dec. 2006, pp. 1, 6, 5–8. DOI: 10.1109/ICARCV.2006.345293.
4. Astudillo L., Castillo O., Aguilar L., R. Martínez, "Hybrid Control for an Autonomous Wheeled Mobile Robot Under Perturbed Torques". In: IFSA (1) 2007, chapter 59, 594–603. DOI: 10.1007/978-3-540-72950-1_59.

5. Astudillo L., Castillo O., Aguilar L., "Intelligent Control of an Autonomous Mobile Robot using Type-2 Fuzzy Logic". In: IC-AI 2006, 565–570.
6. Astudillo, L., Melin, P., Castillo, O., "A new optimization method based on a paradigm inspired by nature". In: *Soft Computing for Recognition Based on Biometrics,* Springer Berlin-Heidelberg, 2010, 277–283.
7. Astudillo, L., Melin, P., Castillo, O., "Nature inspired chemical optimization to design a type-2 fuzzy controller for a mobile robot". In: *IFSA World Congress and NAFIPS Annual Meeting* (IFSA/NAFIPS), 2013, 1423–1428. DOI: 10.1109/IFSA-NAFIPS.2013.6608610.
8. Caraveo, C., Valdez, F., Castillo, O., "Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation", *Applied Soft Computing*, vol. 43, 2016, 131–142. DOI: 10.1016/j.asoc.2016.02.033.
9. Cardenas S., Garibaldi J., Aguilar L., Castillo O, "Intelligent Planning and Control of Robots Using Genetic Algorithms and Fuzzy Logic". IC-AI, 2005, 412–418.
10. Castillo O., Martinez R., Melin P., Valdez F., Soria J., "Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot", *Information Sciences*, vol. 192, 2012, 19–38. DOI: 10.1016/j.ins.2010.02.022.
11. Cervantes L., Castillo O., "Design of a Fuzzy System for the Longitudinal Control of an F-14 Airplane". In: *Soft Computing for Intelligent Control and Mobile Robotics*, 2011, 213–224. DOI: 10.1007/978-3-642-15534-5_13.
12. de la O D., Castillo O., Soria J., "Optimization of Reactive Control for Mobile Robots Based on the CRA Using Type-2 Fuzzy Logic". In: de la O., *Nature-Inspired Design of Hybrid Intelligent Systems*, chapter 33, 2017, 505–518. DOI: 10.1007/978-3-319-47054-2_33.
13. de la O D., Castillo O., Melendez A., Astudillo L., "Optimization of a reactive controller for mobile robots based on CRA". In: *Fuzzy Information Processing Society (NAFIPS)* held jointly with *2015 5th World Conference on Soft Computing (WConSC), 2015 Annual Conference of the North American IEEE*, 1–6.
14. de la O D., Castillo O., Astudillo L., Soria J., "Fuzzy chemical reaction algorithm with dynamic adaptation of parameters". In: *Fuzzy Logic in Intelligent System Design*, Patricia Melin, Oscar Castillo, Janusz Kacprzyk, Marek Reformat, William Melek (eds.), Springer International Publishing, 2018, 122–130. DOI: 10.1007/978-2-319-67137-6_13.
15. De Santis E., Rizzi A., Sadeghiany A, Mascioli F.M.F., "Genetic optimization of a fuzzy control system for energy flow management in micro-grids". In: *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, June 2013 pp. 418, 423, 24–28 June 2013. DOI: 10.1109/IFSA-NAFIPS.2013.6608437
16. Dongyun Wang, Guan Wang, Rong Hu, "Parameters optimization of fuzzy controller based on PSO". In: *ISKE 2008. 3rd International Conference on Intelligent System and Knowledge Engineering*, 17th–19th Nov. 2008 vol. 1, 599, 603.
17. Esmin A. A A, Aoki A. R., Lambert-Torres G., "Particle swarm optimization for fuzzy membership functions optimization". In: *2002 IEEE International Conference on Systems, Man and Cybernetics*, 6th–9th Oct. 2002, vol. 3, p. 6. DOI: 10.1109/ICSMC.2002.1176020.
18. Fierro R., Castillo O., "Design of Fuzzy Control Systems with Different PSO Variants". In: *Recent Advances on Hybrid Intelligent Systems*, chapter 6, 2013, 81–88. DOI: 10.1007/978-3-642-33021-6_6.
19. Fierro R., Castillo O., Valdez F., "Optimization of fuzzy control systems with different variants of particle swarm optimization". In: *2013 IEEE Workshop on Hybrid Intelligent Models and Applications (HIMA)*, 51–56. DOI: 10.1109/HIMA.2013.6615022.
20. Fierro R., Castillo O., Valdez F., Cervantes L., "Design of optimal membership functions for fuzzy controllers of the water tank and inverted pendulum with PSO variants". In: *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, 2013 Joint (pp. 1068–1073). IEEE.
21. Gu Fang, Ngai Ming Kwok, Quang Ha, "Automatic fuzzy membership function tuning using the particle swarm optimization". In: *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008, vol. 2, 324–328, 2008. DOI: 10.1109/PACIIA.2008.105.
22. H. X. Li, H. B. Gatland, "A New Methodology for Designing a Fuzzy Logic Controller*", IEEE Trans. On Sys., Man, and Cybernetics*, vol.25, no. 3, March 1995, 505–512.
23. Kamejima T., Phimmasone V., Kondo Y., Miyatake M., "The optimization of control parameters of PSO based MPPT for photovoltaics". In: *2011 IEEE 9th International Conference on Power Electronics and Drive Systems (PEDS)*, 5th–8th Dec. 2011, 881–883. DOI: 10.1109/PEDS.2011.6147358.
24. Lizarraga E., Castillo O., Soria J., Valdez F., "A Fuzzy Control Design for an Autonomous Mobile Robot Using Ant Colony Optimization". In: *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, chapter 20, 2014, 289–304. DOI: 10.1007/978-3-319-05170-3_20 .
25. Martínez R., Castillo O., Soria J., "Particle Swarm Optimization Applied to the Design of Type-1 and Type-2 Fuzzy Controllers for an Autonomous Mobile Robot". In: *Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition*, 2009, 247–262.
26. Martínez R., Castillo O., Aguilar L., "Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms", *Information Sciences*, 2009, vol. 179, vol. 13, 2158–2174. DOI: 10.1016/j.ins.2008.12.028.
27. Martinez-Soto R., Castillo O., Aguilar L., Baruch I., "Bio-inspired optimization of fuzzy logic controllers for autonomous mobile robots. In: *2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, 2012, 1–6. DOI: 10.1109/NAFIPS.2012.6291053.
28. Martínez-Soto R., Castillo O., Aguilar L., Melin P., "Fuzzy Logic Controllers Optimization Using Ge-

netic Algorithms and Particle Swarm Optimization". In: *Advances in Soft Computing*, chapter 41, 2010, 475–486.
DOI: 10.1007/978-3-642-16773-7_41.

29. Martinez-Marroquin R., Castillo O., Soria J., "Parameter tuning of membership functions of a fuzzy logic controller for an autonomous wheeled mobile robot using ant colony optimization". In: *FUZZ-IEEE 2009. IEEE International Conference on Fuzzy Systems*, 2007–2012.
30. Measurement and Instrumentation, Faculty of Electrical Engineering and Computer Science, Brno University of Technology, Czech Republic Depart-ment of Control. Autonomous Mobile Robotics Toolboxfor Matlab 5. Online. http://www.uamt.feec.vutbr.cz/robotics/simulations/amrt/simrobot en.html, 2001.
31. Melendez A., Castillo O., "Optimization of type-2 fuzzy reactive control-lers for an autonomous mobile robot". In: *2012 Fourth World Congress on Nature and Biologically Inspired Computing* (NaBIC), 2012, 207–211.
32. Melendez A., Castillo O., "Evolutionary optimization of the fuzzy integrator in a navigation system for a mobile robot". In: Castillo O., Melin P., Janusz Kacprzyk (eds.), *Recent Advances on Hybrid Intelligent Systems*, 2013, vol. 451 of Studies in Computational Intelligence, 21–
–31.
DOI: 10.007/978-3-319-47054-2-43 .
33. Melendez A., Castillo O., Soria J., "Reactive control of a mobile robot in a distributed environment using fuzzy logic". In: *Fuzzy Information Processing Society, 2008. NAFIPS 2008. Annual Meeting of the North American*, 2008 1–5.
DOI: 10.1109/NAFIPS.2008.4531341.
34. Melendez A., Castillo O., Garza A., Soria J., "Reactive and tracking control of a mobile robot in a distributed environment using fuzzy logic". In: *IEEE International Conference on Fuzzy Systems*, 2010, 1–5.
DOI: 10.1109/FUZZY.2010.5583955.
35. Melendez A., Castillo O. "Hierarchical genetic optimization of the fuzzy integrator for navigation of a mobile robot". In: *Soft Computing Applications in Optimization, Control, and Recognition*, chapter 4, 2013, 77–96.
DOI: 10.1007/978-3-642-35323-9_4.
36. Porta García M. A., Montiel O., Castillo O., Sepúlveda R., *Optimal Path Planning for Autonomous Mobile Robot Navigation Using Ant Colony Optimization and a Fuzzy Cost Function Evaluation. Analysis and Design of Intelligent Systems using Soft Computing Techniques,* 2007, 790–798.
37. Milla F., Sáez D., Cortés C.E., Cipriano A., "Bus--Stop Control Strategies Based on Fuzzy Rules for the Operation of a Public Transport System", *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1394–1403.
DOI: 10.1109/TITS.2012.2188394.
38. Montiel O., Camacho J., Sepúlveda R., Castillo O., "Fuzzy System to Control the Movement of a Wheeled Mobile Robot". In: *Soft Computing for Intelligent Control and Mobile Robotics*, 2011, 445–463.
39. Ochoa P., Castillo O., Soria J., "Differential evolution with dynamic adaptation of parameters for the optimization of fuzzy controllers". In: *Recent Advances on Hybrid Approaches for designing intelligent systems*, chapter 19, 2014, 275–288.
DOI: 10.1007/978-3-319-05170-3_19.
40. Porta M., Montiel O., Castillo O., R. Sepúlveda, Melin P., "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation", *Appl. Soft Comput.*, 2009, vol. 9, no. 3, 1102–1110.
41. Vaneshani S., Jazayeri-Rad H., "Optimized fuzzy control by particle swarm optimization technique for control of SCTR", *International Journal of Electrical and Computer Engineering*, 2011, vol. 11, no. 5, 464–470.
DOI: scholar.waset.org/1307-6892/391.
42. Aguas-Marmolejo S.J., Castillo O., "Optimization of Membership Functions for Type-1 and Type 2 Fuzzy Controllers of an Autonomous Mobile Robot Using PSO", *Recent Advances on Hybrid Intelligent Systems*, vol. 451, 2013, 97–104.
DOI: 10.1007/978-3-642-33021-6_8.
43. Wong S., Hamouda A., "Optimization of fuzzy rules design using genetic algorithm", *Advances in Engineering Software*, vol. 31, issue 4, April 2000, 251–262, ISSN 0965-9978.
DOI: 10.1016/S0965-9978(99)00054-X.
44. Yen J., Langari R., *Fuzzy Logic: Intelligence, Control, and Information*, Prentice Hall, 1999.
45. Ying Bai, Hanqi Zhuang, Zvi. S. Roth, "Fuzzy Logic Control to Suppress Noises and Coupling Effects in a Laser Tracking System", *IEEE Trans on Control Systems Technology*, vol.13, no.1, January 2005, 113–121.
46. Zafer B., Oğuzhan K., "A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control", *Expert Systems with Applications*, vol. 38, issue 1, January 2011, 1017–1031, ISSN 0957-4174.
DOI: 10.1016/j.eswa.2010.07.131.