Optimization of Type-2 and Type-1 Fuzzy Tracking Controllers for an Autonomous Mobile Robot under Perturbed Torques by Means of a Chemical Optimization Paradigm

Leslie Astudillo, Patricia Melin, and Oscar Castillo

Tijuana Institute of Technology, Tijuana México
{epmelin,ocastillo}@hafsamx.org

Abstract. This paper addresses the tracking problem for the dynamic model of a unicycle mobile robot. A novel optimization method inspired on the chemical reactions is applied to solve this motion problem by integrating a kinematic and a torque controller based on fuzzy logic theory. Computer simulations are presented confirming that this optimization paradigm is able to outperform other optimization techniques applied to this particular robot application.

1 Introduction

Optimization is an activity carried out in almost every aspect of our life, from planning the best route in our way back home from work to more sophisticated approximations at the stock market, or the parameter optimization for a wave solder process used in a printed circuit board assembly manufacturer optimization theory has gained importance over the last decades. From science to applied engineering (to name a few), there is always something to optimize and of course, more than one way to do it.

In a generic definition, we may say that optimization aims to find the "best" available solution among a set of potential solutions in a defined search space. For almost every problem exists a solution, not necessarily the best, but we can always find an approximation to the "ideal solution", and while in some cases or processes is still common to use our own experience to qualify a process, a part of the research community have dedicated a considerably amount of time and efforts to help find robust optimization methods for optima finding in a vast range of applications.

It has been stated the difficulty to solve different problems by applying the same methodology, and even the most robust optimization approaches may be outperformed by other optimization techniques depending on the problem to solve.

P. Melin, O. Castillo (Eds.): *Soft Computing Appli. in Optimization,* STUDFUZZ 294, pp. 3–26.
DOI: 10.1007/978-3-642-35323-9_1

When the complexity and the dimension of the search space make a problem unsolvable by a deterministic algorithm, probabilistic algorithms deal with this problem by going through a diverse set of possible solutions or candidate solutions. Many metaheuristic algorithms can be considered probabilistic, while they apply probability tools to solve a problem, metaheuristic algorithms seek good solutions by mimicking natural processes or paradigms. Most of these novel optimization paradigms inspired by nature were conceived by merely observation of an existing process and their main characteristics were embodied as computational algorithms.

The importance of the optimization theory and its application has grown in the past few decades, from the well known Genetic Algorithm paradigm to PSO, ACO, Harmonic Search, DNA Computing, among others, they all were introduced with the expectation of improving the results obtained with the existing strategies.

There's no doubt that there could be some optimization strategies presented at some point that were left behind due their complexity and poor performance. Novel optimization paradigms should be able to perform well in comparison with another optimization techniques and must be "easily adaptable" to different kinds of problems.

Optimization based on chemical processes is a growing field that has been satisfactorily applied to several problems. In [25] A DNA based algorithm was to solve the small hitting set problem. A catalytic search algorithm was explored in [30], where some physical laws such as mass and energy conservation were taken into account. In [19], the potential roles of energy in algorithmic chemistries were illustrated. An energy framework was introduced, which keeps the molecules within a reasonable length bounds, allowing the algorithm to behave thermodynamically and kinetically similar to real chemistry. A chemical reaction optimization was applied to the grid scheduling problem in [29], where molecules interact with each other aiming to reach the minimum state of free potential and kinetic energies. The main difference between these metaheuristics is the parameter representation, which can be explicit or implicit.

In this paper we introduce an optimization method inspired on the chemical reactions and its application for the optimization of the tracking controller for the dynamic model of the unicycle mobile robot.

The importance of applying this chemical optimization algorithm is that different methods have been applied to solve motion control problems. Kanayama et al. [13] propose a stable tracking control method for a non-holonomic vehicle using a Lyapunov function. Lee et al. [15] solved tracking control using backstepping and in [17] with saturation constraints. Furthermore, most reported designs rely on intelligent control approaches such as fuzzy logic control [3][12][16][23][27][28] and neural networks [10][26].

However the majority of the publications mentioned above, have concentrated on kinematic models of mobile robots, which are controlled by the velocity input, while less attention has been paid to the control problems of nonholonomic dynamic systems, where forces and torques are the true inputs: Bloch and Drakunov [4] and Chwa [8], used a sliding mode control to the tracking control problem. Fierro and Lewis [9] propose a dynamical extension that makes possible

the integration of kinematics and torque controller for a nonholonomic mobile robot. Fukao et al. [11], introduced an adaptive tracking controller for the dynamic model of mobile robot with unknown parameters using backstepping methodology, which has been recognized as a tool for solving several control problems [24] [31].Motivated by this, a mamdani fuzzy logic controller is introduced in order to drive the kinematic model to a desired trajectory in a finite-time, considering the torque as the real input, a chemical reaction optimization paradigm is applied and simulations are shown.

Further publications [2][18][6] have applied bio-inspired optimization techniques to find the parameters of the membership functions for the fuzzy tracking controller that solves the problem for the dynamic model of a unicycle mobile robot, using a fuzzy logic controller that provides the required torques to reach the desired velocity and trajectory inputs.

In this paper, the main contribution is the representation of the fuzzy controller in the chemical paradigm to search for the optimal parameters. Simulation results show that the proposed approach outperforms other nature inspired computing paradigms, such as genetic algorithms, particle swarm and ant colony optimization.

The rest of this paper is organized as follows. Section 2 illustrates the proposed methodology. Section 3 describes the problem formulation and control objective. Section 4 describes the proposed fuzzy logic controller of the robot. Section 5 shows some experimental results of the tracking controller and in section 6 some conclusions and future work are presented.

2 The Chemical Optimization Paradigm

The proposed chemical reaction algorithm is a metaheuristic strategy that performs a stochastic search for optimal solutions within a defined search space. In this optimization strategy, every solution is represented as an element (or compound), and the fitness or performance of the element is evaluated in accordance with the objective function. The general flowchart of the algorithm is shown in Figure 1.

The main difference with other optimization techniques [25][30][19][29] is that no external parameters are taken into account to evaluate the results, while other algorithms introduce additional parameters (kinetic/potential energies, mass conservation, thermodynamic characteristics, etc), this is a very straight forward methodology that takes the characteristics of the chemical reactions (synthesis, decomposition, substitution and double-substitution) to find for optimal solution.

This approach is a static population-based metaheuristic that applies an abstraction of the chemical reactions as intensifiers (substitution, double substitution reactions) and diversifying (synthesis, decomposition reactions) mechanisms. The elitist reinsertion strategy allows the permanence of the best elements and thus the average fitness of the entire element pool increases with every iteration. The algorithm may trigger only one reaction or all of them, depending on the nature of the problem to solve, in example; we may use only the decomposition reaction sub-routine to find the minimum value of a mathematical function.

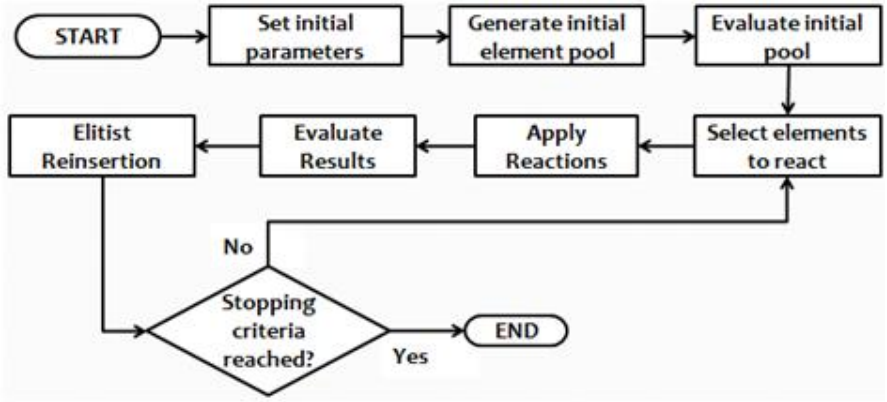

Fig. 1 General flowchart of the chemical reaction algorithm

The pseudocode for the chemical reaction algorithm is as follows:

Chemical_Reaction _Algorithm
Input: *problem_definition*, *objective_function*, *dimensions*,
1. Assign values to variables: *pool_size*, *Trials*, *upper_boundary*, *lower_boundary*, *synthesis_rate*, *decomposition_rate*, *singlesubstitution_rate*, *doublesubstitution_rate*. 2. Generate Randomly *Initial_Pool* in interval [*lower_boundary*, *upper_ boundary*] 3. Evaluate *Initial_Pool* *4.* Identify *best_solution* *5.* **while** (stopping criteria not met) **do** 6. Perform *Synthesis_Procedure*; Get *Synthesis_vector* 7. Perform *Decomposition_Procedure*; Get *Decomposition_vector* 8. Perform *SingleSubstitution_Procedure*; Get *SingleSubstitution_vector* 9. Perform *DoubleSubstitution_Procedure*; Get *DoubleSubstitution_vector* 10. Evaluate *Synthesis_vector*, *Decomposition_vector*, *SingleSubstitution_ vector*, *DoubleSubstitution_vector* 11. Apply *elitist_reinsertion*; Get *improved_pool* 12. Update *best_solution* *13.* **end while**
Output: *best_solution*

Every nature inspired paradigm has their own way to encode candidate solutions. When these parameters are defined, a set of processes or procedures are applied to lead the population to an optimal result. The main components of this chemical reaction algorithm are described below.

Elements/Compounds
These are the basic components of the algorithm. Each element or compound represents a solution within the search space. The initial definition of elements and/or compounds depends on the problem itself and can be represented as binary numbers, integer, floating, etc. They interact with each other implicitly; this is, the definition of the interaction is independent of the real molecular structure; in this approach the potential and kinetic energies and other molecular characteristics are not taken into account.

Chemical Reactions
A chemical reaction is a process in which at least one substance changes its composition and its sets of properties, in this approach, the chemical reactions behave as intensifiers (substitution, double substitution reactions) and diversifying (synthesis, decomposition reactions) mechanisms. The 4 chemical reactions considered in this approach are the *synthesis, decomposition, single and double substitution reactions.* The objective of these operators is exploring or exploiting new possible solutions within a slightly larger hypercube than the original elements/compounds, but within the previously specified range.

The *synthesis* and *decomposition* reactions are used to diversify the resulting solutions; these procedures showed to be highly effective to rapidly lead the results to a desired value. They can be described as follows.

Synthesis Reactions
Is a reaction of two reactants to produce one product. By combining two (or more) elements, this procedure allows to explore higher valued solutions within the search space. The result can be described as a compound *(B+C → BC)*. The pseudocode for the synthesis reaction procedure is as follows:

Synthesis_Procedure
Input: *selected_elements*, *synthesis_rate* 1. n = size (*selected_elements*) 2. i = floor (n / 2) 3. **for** j = 1 to i – 1 *4.* *Synthesis* = $selected\_elements_j + selected\_elements_{j+1}$ 5. $j = j + 2$ 6. **end for**
Output: *Synthesis_vector*

Decomposition Reactions
In this reaction, typically, only one reactant is given, it allows a compound to be decomposed into smaller instances *(BC → B+C)*. The pseudocode for the decomposition reaction procedure is as follows:

Decomposition_Procedure
Input: *selected_elements*, *decomposition_rate* 1. n = size (*selected_elements*) 2. Get *randval* randomly in interval [0, 1] 3. **for** i = 1 to n *4.* $Deco_1 = selected\_elements_i$ x *randval* *5.* $Deco_2 = selected\_elements_i$ x (1 – *randval*) 6. $i = i + 1$ 7. **end for**
Output: *Decomposition_vector* ($Deco_1, Deco_2$)

The *single* and *double substitution* reactions allow the algorithm to search for optima around a good previously found solution and they're described as follows.

Single-Substitution Reactions
When a free element reacts with a compound of different elements, the free element will replace one of the elements in the compound if the free element is more reactive than the element it replaces. A new compound and a new free element are produced; during the algorithm, a compound and an element are selected and a decomposition reaction is applied to the compound; two elements are generated from this operation. Then, one of the new generated elements is combined with the non-decomposed selected element *(*$C + AB \rightarrow AC + B$*)*. The pseudocode for the single-substitution reaction procedure is as follows:

SingleSubstitution_Procedure
Input: *selected_elements*, *singlesubstitution_rate* 1. n = size (*selected_elements*) 2. i = floor (n / 2) 3. $a = selected\_elements_1, selected\_elements_2, \ldots, selected\_elements_i$ 4. $b = selected\_elements_{i+1}, selected\_elements_{i+2}, \ldots, selected\_elements_{ix2}$ 5. Apply *Decomposition_Procedure* to a; Get $Deco_1, Deco_2$ 6. Apply *Synthesis_Procedure* ($b + Deco_1$); Get *Synthesis_vector*
Output: *SingleSubstitution _vector* (*Synthesis_vector*, $Deco_2$)

Double-Substitution Reactions
Double-substitution or double-replacement reactions, also called double-decomposition reactions or metathesis reactions involve two ionic compounds, most often in aqueous solution. In this type of reaction, the cations simply swap anions; during the algorithm, a similar process that in the previous reaction happens, the difference is that in this reaction both of the selected compounds are decomposed and the resulting elements are combined between each other (*$AB + CD \rightarrow CB + AD$*)*. The pseudocode for the double-substitution reaction procedure is as follows:

DoubleSubstitution_Procedure
Input: *selected_elements*, *doublesubstitution_rate* 1. n = size (*selected_elements*) 2. i = floor (n / 2) 3. a = $selected\_elements_1$, $selected\_elements_2$, …, $selected\_elements_i$ 4. b = $selected\_elements_{i+1}$, $selected\_elements_{i+2}$, …, $selected\_elements_{ix2}$ 5. Apply *Decomposition_Procedure* to a and b; Get ($Deco_1$, $Deco_2$), ($Deco_1$', $Deco_2$') 6. Apply *Synthesis_Procedure* ($Deco_1$ + $Deco_1$'), ($Deco_2$ + $Deco_2$') Get $Synthesis\_vector_1$, $Synthesis\_vector_1$'
Output: *SingleSubstitution _vector* ($Synthesis\_vector_1$, $Synthesis\_vector_1$')

In this chemical reaction algorithm we may trigger only one reaction or all of them, depending on the nature of the problem to solve, e.g., we can apply only the decomposition reaction sub-routine to find the minimum value of a mathematical function.

Throughout the execution of the algorithm, whenever a new set of elements/compounds are created, an elitist reinsertion criteria is applied, allowing the permanence of the best elements and thus the average fitness of the entire element pool increases through iterations.

In order to have a better picture of the general schema for this proposed chemical reaction algorithm, a comparison with other nature inspired paradigms is shown in Table 1.

Table 1 Main elements of several nature inspired paradigms

Paradigm	Parameter Representation	Basic Operations
GA	Genes	Crossover, Mutation
ACO	Ants	Pheromone
PSO	Particles	Cognitive, Social Coefficients
GP	Trees	Crossover, Mutation (In some cases)
CRM	Elements, Compounds	Reactions (Combination, Decomposition, Substitution, Double-substitution)

3 The Mobile Robot

Mobile robots are non-holonomic systems due to the constraints imposed on their kinematics. The equations describing the constraints cannot be integrated symbolically to obtain explicit relationships between robot positions in local and global coordinate's frames. Hence, control problems that involve them have attracted attention in the control community in recent years [14].

The model considered is that of a unicycle mobile robot (see Figure 2) that has two driving wheels fixed to the axis and one passive orientable wheel that are placed in front of the axis and normal to it [5].

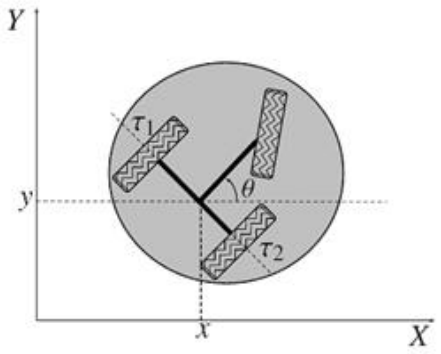

Fig. 2 Diagram of a wheeled mobile robot

The two fixed wheels are controlled independently by the motors, and the passive wheel prevents the robot from overturning when moving on a plane.

It is assumed that the motion of the passive wheel can be ignored from the dynamics of the mobile robot, which is represented by the following set of equations [9]:

$$\dot{q} = \begin{vmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} v \\ w \end{vmatrix} \tag{1}$$

$$M(q)\dot{v} + V(q,\dot{q})v + G(q) = \tau$$

Where $q = [x, y, \theta]^T$ is the vector of generalized coordinates which describes the robot position, *(x,y)* are the Cartesian coordinates, which denote the mobile center of mass and θ is the angle between the heading direction and the *x*-axis (which is taken counterclockwise form); $v = [v, w]^T$ is the vector of velocities, *v* and *w* are the linear and angular velocities respectively; $\tau \in R^r$ is the input vector, $M(q) \in R^{nxn}$ is a symmetric and positive-definite inertia matrix, $V(q,\dot{q}) \in R^{nxn}$ is the centripetal and Coriolis matrix, $G(q) \in R^n$ is the gravitational vector. Equation (1.a) represents the kinematics or steering system of a mobile robot.

Notice that the no-slip condition imposed a non holonomic constraint described by (2), that it means that the mobile robot can only move in the direction normal to the axis of the driving wheels.

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{2}$$

The control objective will be established as follows: Given a desired trajectory $q_d(t)$ and the orientation of the mobile robot we must design a controller that applies an adequate torque τ such that the measured positions $q(t)$ achieve the desired reference $q_d(t)$ represented as (3):

$$\lim_{t\to\infty}\|q_d(t)-q(t)\|=0 \tag{3}$$

To reach the control objective, the method is based on the procedure of [9], we are deriving a $\tau(t)$ of a specific vc(t) that controls the steering system (1.a) using a Fuzzy Logic Controller (FLC). A general structure of tracking control system is presented in Figure 3.

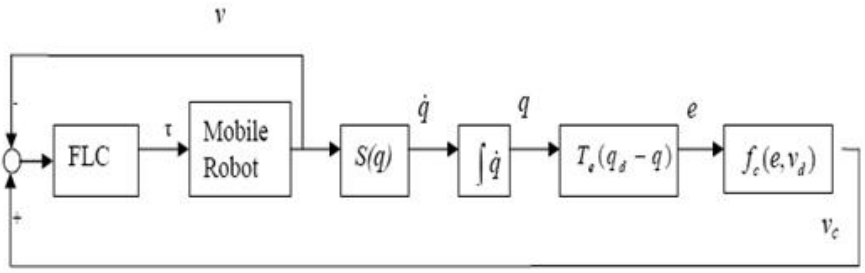

Fig. 3 Tracking control structure

The control is based on the procedure proposed by Kanayama et al. [13] and Nelson et al. [21] to solve the tracking problem for the kinematic model $v_c(t)$. Suppose that the desired trajectory q_d satisfies (4):

$$\dot{q}_d=\begin{vmatrix}\cos\theta_d & 0\\ \sin\theta_d & 0\\ 0 & 1\end{vmatrix}\begin{vmatrix}v_d\\ w_d\end{vmatrix} \tag{4}$$

Using the robot local frame (the moving coordinate system x-y in figure 1), the error coordinates can be defined as (5):

$$e=T_e(q_d-q),\begin{vmatrix}e_x\\ e_y\\ e_\theta\end{vmatrix}=\begin{vmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{vmatrix}\begin{vmatrix}x_d-x\\ y_d-y\\ \theta_d-\theta\end{vmatrix} \tag{5}$$

And the auxiliary velocity control input that achieves tracking for (1.a) is given by (6):

$$v_c = f_c(e, v_d), \begin{vmatrix} v_c \\ w_c \end{vmatrix} = \begin{vmatrix} v_d + \cos e_\theta + k_1 e_x \\ w_d + v_d k_2 e_y + v_d k_3 \sin e_\theta \end{vmatrix} \tag{6}$$

Where k_1, k_2 and k_3 are positive gain constants.

The first part for this work is to apply the proposed method to obtain the values of k_i (i = 1, 2, 3) for achieving the optimal behavior of the controller, and the second part is to optimize the fuzzy controller.

4 Fuzzy Logic Controller

The purpose of the fuzzy logic controller (FLC) is to find a control input τ such that the current velocity vector v is able to reach the velocity vector v_c and this is denoted as:

$$\lim_{t \to \infty} \| v_c - v \| = 0 \tag{7}$$

The inputs variables of the FLC correspond to the velocity errors obtained of (10) (denoted as e_v and e_w: linear and angular velocity errors respectively), and 2 outputs variables, the driving and rotational input torques τ (denoted by F and N respectively). The initial membership functions (MF) are defined by 1 triangular and 2 trapezoidal functions for each variable involved. Figure 4 depicts the MFs in which N, Z, P represent the fuzzy sets (Negative, Zero and Positive respectively) associated to each input and output variable.

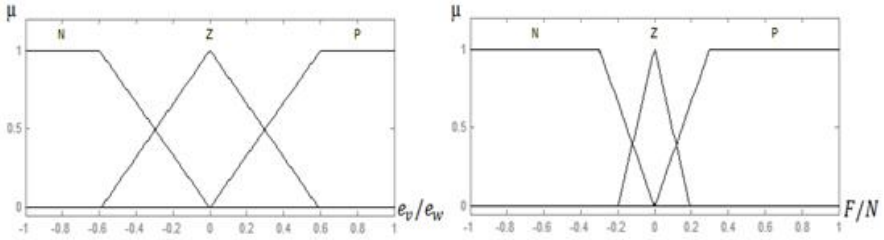

Fig. 4 Membership functions of the (a) input e_v and e_w, and (b) output variables *F* and *N*

The rule set of the FLC contain 9 rules, which govern the input-output relationship of the FLC and this adopts the Mamdani-style inference engine. We use the center of gravity method to realize defuzzification procedure. In Table 2, we present the rule set whose format is established as follows:

Rule i: If e_v is G1 and e_w is G2 then F is G3 and N is G4

Where G1…G4 are the fuzzy sets associated to each variable and i= 1 ... 9. In this case, P denotes “Positive”, N denotes “Negative”, and Z denotes “Zero”.

Table 2 Fuzzy rule set

e_v/e_w	N	Z	P
N	N/N	N/Z	N/P
Z	Z/N	Z/Z	Z/P
P	P/N	P/Z	P/P

5 Experimental Results

Several tests of the chemical optimization paradigm were made to test the performance of the tracking controller. First, we need to find the values of k_i (i = 1, 2, 3) showed in equation 6, which shall guarantee convergence of the error e to zero.

To evaluate the constants obtained by the algorithm, the mobile robot tracking system, which consists in equations 5 and 6 was modeled using Simulink®. Figure 5 shows the closed loop for the tracking controller.

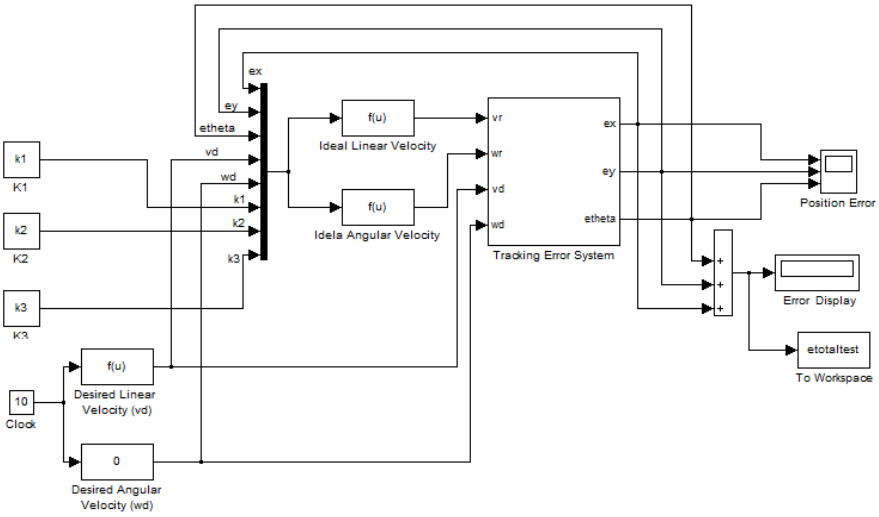

Fig. 5 Closed loop for the tracking controller system

The conditions to evaluate each result, which correspond to the final position error, are given by equation 12.

$$EP = \sum_{i=1}^{n} \frac{e_x(i) + e_y(i) + e_\theta(i)}{n} \tag{12}$$

For the first set of experiments only the decomposition reaction mechanism was triggered and the decomposition factor was varied; this factor is the quantity of resulting elements after applying a decomposition reaction to a determined "compound"; the only restriction here is that let x be the selected compound and x'_i ($i=1\ 2,\ \ldots,\ n$), the resulting elements; the sum of all values found in the decomposition must be equal to the value of the original compound. This is shown in equation 13.

$$\sum_{i=1}^{n} x'_i = x \tag{13}$$

Each experiment was executed 35 times and the test parameters for each set of experiments can be observed in Table 3.

The decomposition rate (Dec. Rate) represents the percentage of the pool to be candidate for the decomposition and the decomposition factor (Dec. Factor) is the number of elements to be decomposed into.

The selection strategy applied was the stochastic universal sampling, which uses a single random value to sample all of the solutions by choosing them at evenly spaced intervals.

In example, for a pool containing 5 initial compounds, the vector length of decomposed elements when the decomposition factor is 3 and the decomposition rate is 0.4 will be of 6 elements.

Table 3 Parameters of the Chemical Reaction Optimization

No.	Elements	Iterations	Dec. Factor	Dec. Rate
1	2	10	2	0.3
2	5	10	3	0.3
3	2	10	2	0.4
4	2	10	3	0.4
5	5	10	2	0.4
6	5	10	3	0.4
7	5	10	2	0.5
8	10	10	2	0.5

By applying this criterion, the initial pool of elements increased with every iteration; this is why the initial element pool was set to 10 elements as maximum. Table 4 shows the results after applying the chemical optimization paradigm.

Table 4 Experimental Results of the proposed method for optimizing the values of the gains k_1, k_2, k_3

No.	Best Error	Mean	k_1	k_2	k_3
1	0.0086	1.1568	519	46	8
2	4.79e-04	0.1291	205	31	31
3	0.0025	0.5809	36	328	88
4	0.0012	0.5589	2	206	0
5	0.0035	0.0480	185	29	5
6	8.13e-005	0.0299	270	53	15
7	0.0066	0.1440	29	15	0
8	0.0019	0.1625	51	3	0

As it is observed in Table 4, experiment number 6 seems to be the best result because it reached the smaller final error among all experiments.

Figure 6 shows the final position errors in *x*, *y* and *theta* for experiment no. 6.

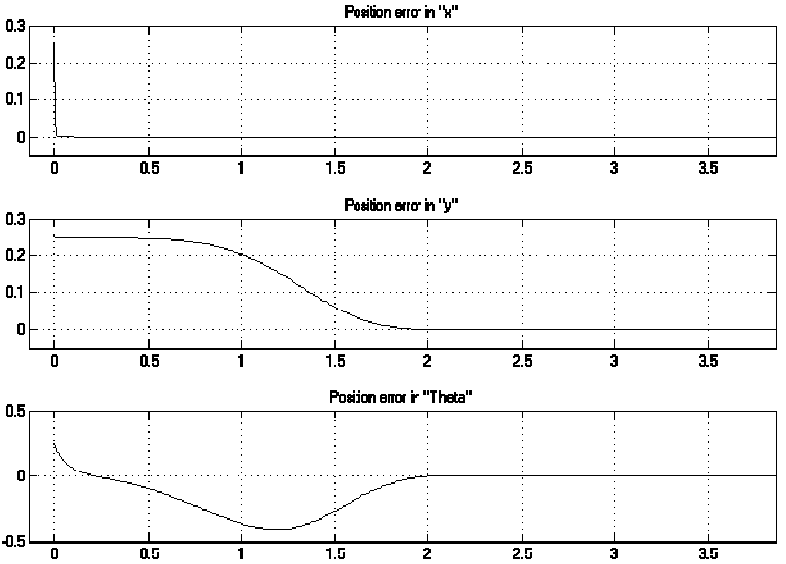

Fig. 6 Final position errors in *x*, *y* and *theta* for experiment no. 6

By analyzing the graphical results of several set of exercises, we noticed that the control obtained for some of them was "smoother" despite the average error value. This was the case for experiment no. 3, in which the final error value was significantly higher than the obtained in experiment no. 6. Figure 7 shows the final position errors in *x*, *y* and *theta* for experiment no. 3.

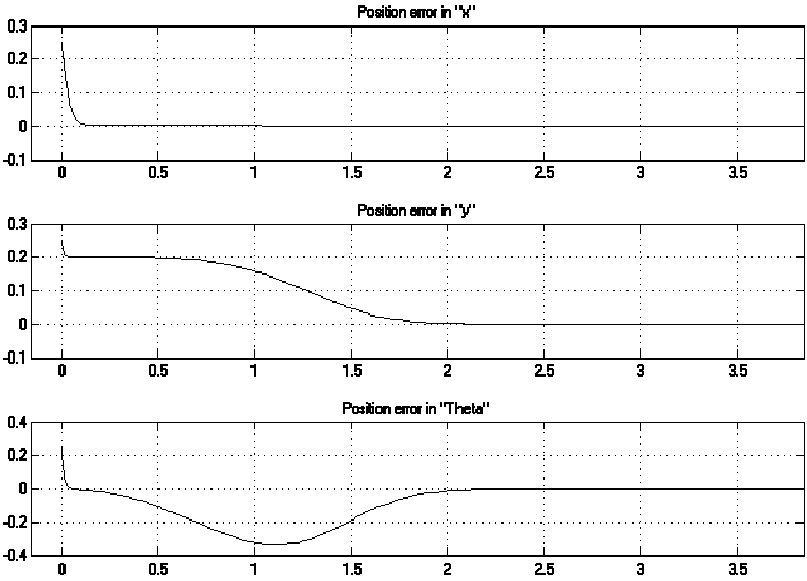

Fig. 7 Final position errors in *x*, *y* and *theta* for experiment no. 6

Making a comparison between both graphics, we can observe that the average error obtained for *theta* is 0.0338 for experiment no. 6 and 0.0315 for experiment no. 3.

This smoother control of the tracking system could make a big difference in the complete dynamic system of the mobile robot.

In previous work [22], the gain constant values were found by means of genetic algorithms. In Table 5 we have a comparison of the best results obtained with both algorithms, we can observe that the result with the chemical optimization outperforms the GA in finding the best gain values.

Table 5 Comparison of the Best Results

Parameters	Genetic Algorithm	Chemical Optimization Algorithm
Individuals	5	2
Iterations	15	10
Crossover Rate	0.8	N/A
Mutation Rate	0.1	N/A
Synthesis Rate	N/A	0.2
Decomposition Rate	N/A	0.8
Substitution Rate	N/A	0.6
Double Substitution Rate	N/A	0.6
k1, k2, k3	43, 493, 195	36, 328, 88
Final Error	0.006734	0.0025

Figure 8 shows the result in Simulink for the experiment with the best overall result, applying GAs as optimization method.

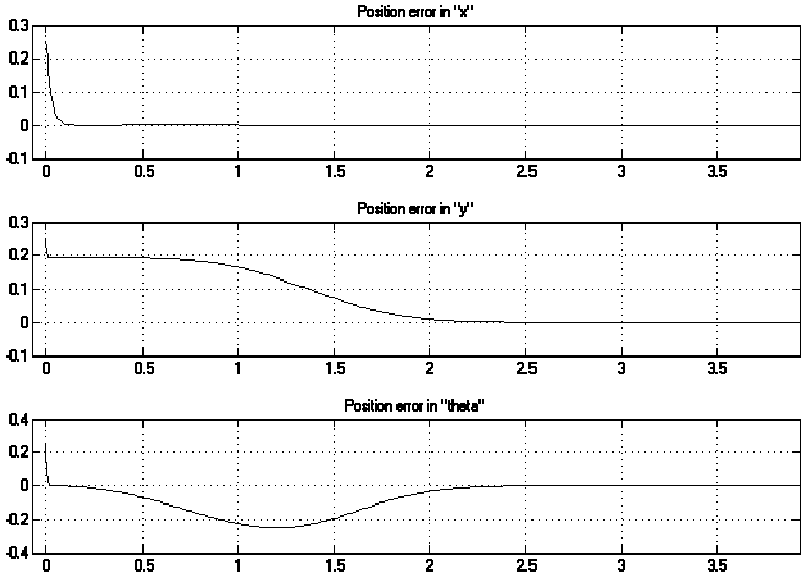

Fig. 8 Position errors in *x*, *y* and *theta* of best result applying GAs

Once we have found optimal values for the gain constants, the next step is to find the optimal values for the input/output membership functions of the fuzzy controller. Our goal is that in the simulations, the lineal and angular velocities reach zero. Table 6 shows the parameters of the simulateons for typ-1 FLC.

Table 6 Parameters of the simulations for Type-1 FLC

Parameters	Value
Elements	10
Trials	15
Selection Method	Stochastic Universal Sampling
k_1	117
k_2	226
k_3	137
Error	0.077178

Figure 9 shows the behavior of the chemical optimization algorithm throughout the experiment.

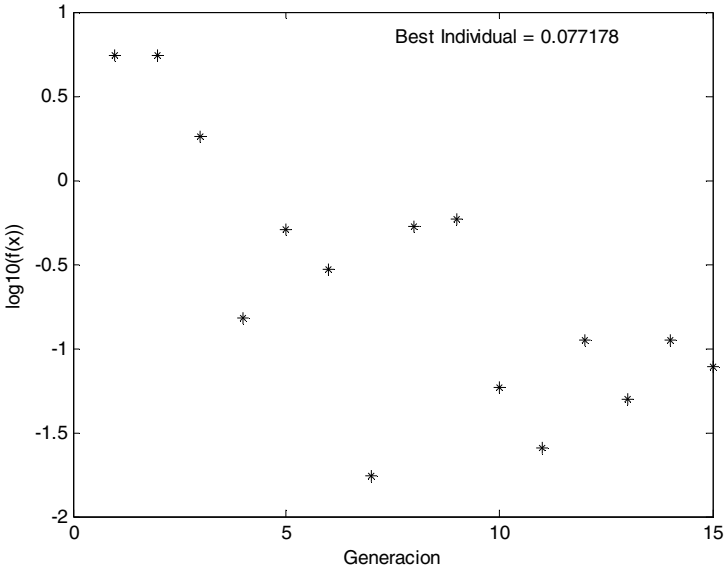

Fig. 9 Best simulation of experiments with the chemical optimization method

Figure 10 shows the resulted input and output membership functions found by the proposed optimization algorithm.

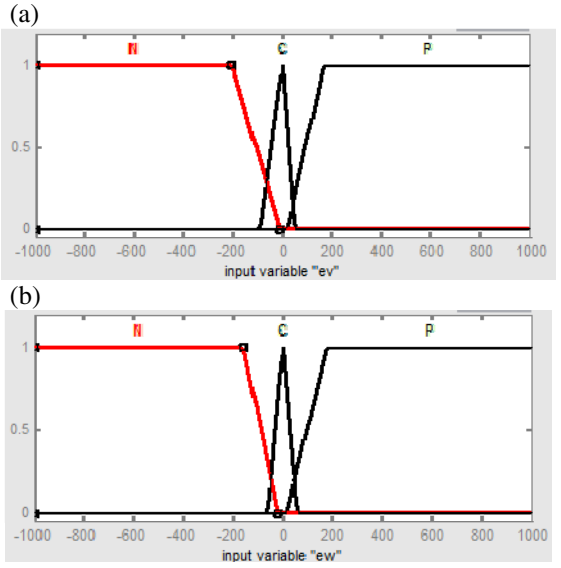

Fig. 10 Resulting input membership functions: (a) linear and (b) angular velocities and output (c) right and (d) left torque

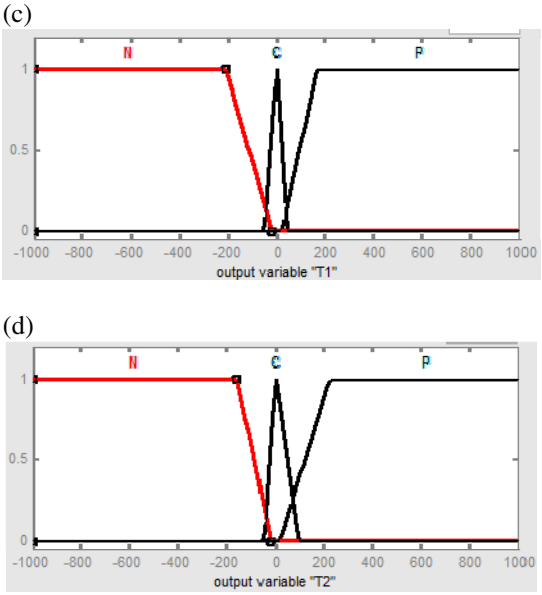

Fig. 10 (*continued*)

Figure 11 shows the obtained trajectory when simulating the mobile control system including the obtained input and output membership functions.

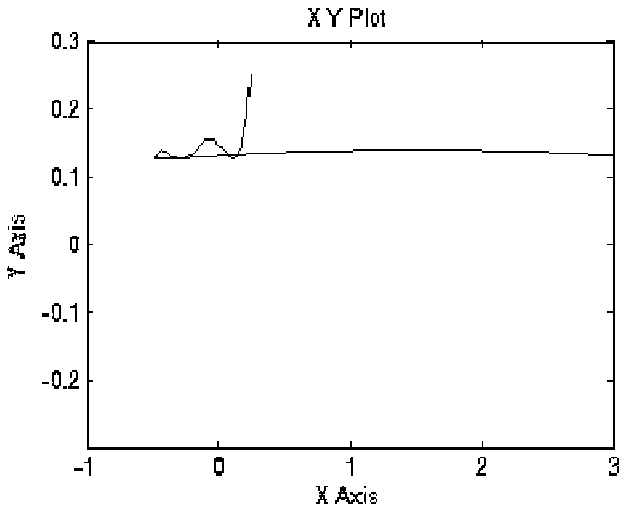

Fig. 11 Obtained trajectory when applying the chemical reaction algorithm

Figure 12 shows the best trajectory reached by the mobile when optimizing the input and output membership functions using genetic algorithms.

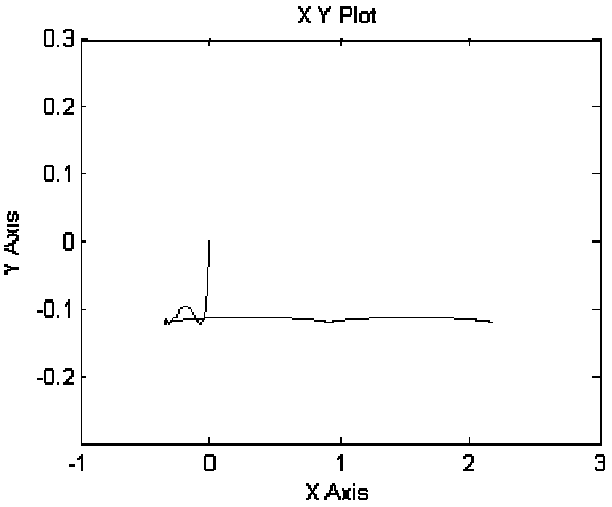

Fig. 12 Obtained trajectory using genetic algorithms

A Type-2 fuzzy logic controller was developed using the parameters of the membership functions found for the FLC type-1. The parameters searched with the chemical reaction algorithm were for the footprint of uncertainty (FOU).

Table 7 shows the parameters used in the simulations and Figure 13 shows the behavior of the chemical optimization algorithm throughout the experiment.

Table 7 Parameters of the simulations for Type-2 FLC

Parameters	Value
Elements	10
Trials	10
Selection Method	Stochastic Universal Sampling
k_1	117
k_2	226
k_3	137
Error	2.7736

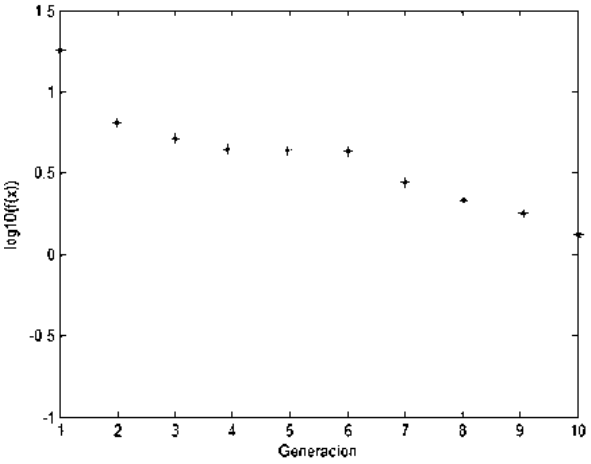

Fig. 13 Behavior of the algorithm when optimizing the type-2 FLC

Figure 14 shows the resulting type-2 input and output membership functions found by the proposed optimization algorithm and Figure 15 shows the obtained trajectory reached by the mobile robot.

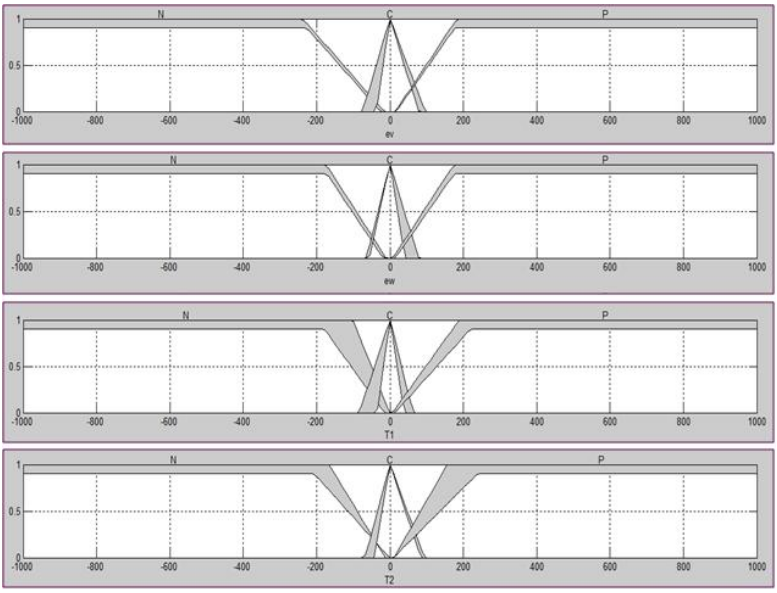

Fig. 14 Resulting type-2 input membership functions, from top to bottom: (a) linear and (b) angular velocities and output (c) right and (d) left torque

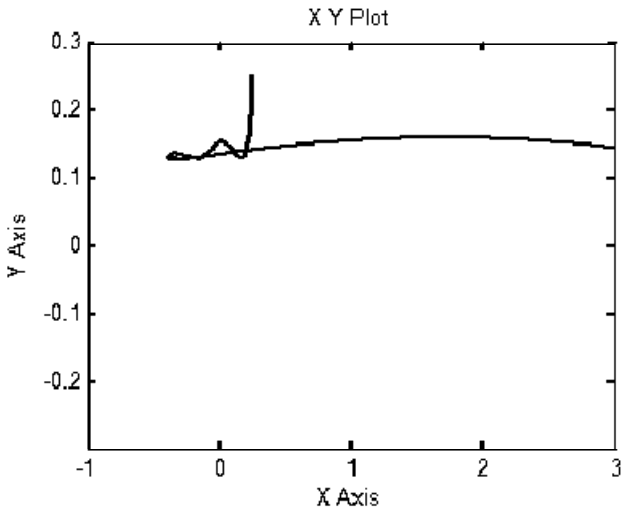

Fig. 15 Obtained trajectory for the mobile robot when applying the chemical reaction algorithm to the type-2 FLC

As observed in Table 7, the final error obtained is not smaller that the final error found for the type-1 FLC. Despite this, the trajectory obtained and showed in Figure 15 is acceptable taking into account that the reference trajectory is a straight line. In Figure 16 we can observe an "unacceptable" trajectory that was found in the early attempts of optimization for the type-1 FLC applying this chemical reaction algorithm. Here, we can observe that the parameters found were not the adequate to make the FLC follow the desired trajectory.

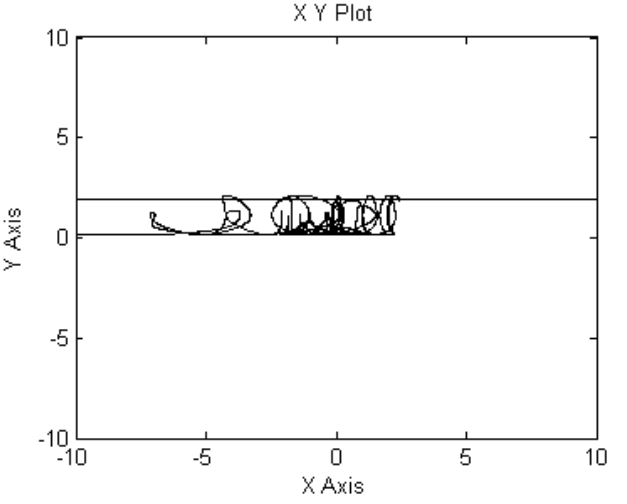

Fig. 16 Unaccepted resulting trajectory in early optimization trials

In order to test the robustness of the type-1 and type-2 FLC, we added an external signal given by equation (13).

$$F_{ext}(t) = \varepsilon \times \sin \omega \times t \tag{13}$$

This represents an external force applied in a period of 10 seconds to the obtained trajectory that will make the mobile robot to be out of its path. The idea of adding this disturbance is to measure the errors obtained with the FLC and to test the behavior of the mobile robot under perturbed torques. Table 8 shows the parameters for the simulations and the errors obtained during the run of the simulation.

Table 8 Simulation parameters and errors obtained under disturbed torques

ε	Velocity errors	Type-1 (GA)	Type-1 (CRA)	Type-2 (CRA)
0.05	Final error	4.0997	0.9815	29.5115
	Average error	4.1209	1.5823	26.6408
5	Final error	4.1059	0.9729	29.52
	Average error	3.1695	1.8679	26.1646
10	Final error	4.1045	0.9745	29.51
	Average error	3.0985	1.7438	24.9467
30	Final error	4.0912	0.9783	29.51
	Average error	2.2632	1.9481	24.6032
32	Final error	3273	0.9748	29.52
	Average error	3.4667e+003	2.8180	24.6465
34	Final error	1.5705e+004	566.8	29.51
	Average error	1.1180e+004	215.8198	24.9211
40	Final error	2.534e+004	3.5417e+04	29.51
	Average error	186.0611	5.7492e+003	23.8938
41	Final error	8839	3168	685.1
	Average error	2.0268e+004	0.0503e+003	16.5257

Figure 17 show the obtained trajectories for the type-1 FLC optimized with Genetic Algorithms.

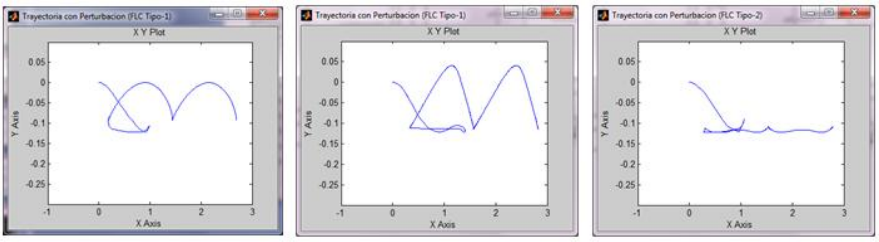

Fig. 17 From left to right, trajectory obtained with the type-1 FLC optimized with GA's. (a) ε = 30, (b) ε = 32, (c) ε = 34.

Figure 18 shows the obtained trajectories for the type-1 FLC optimized with the chemical reaction algorithm.

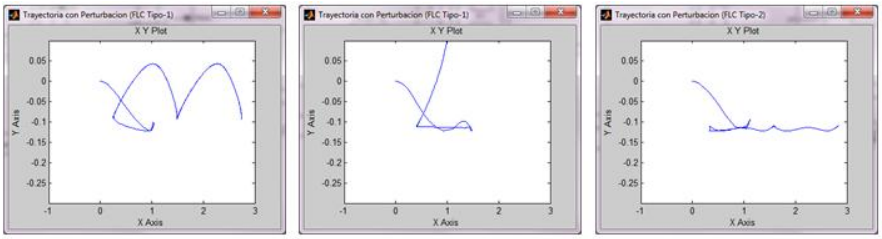

Fig. 18 From left to right, trajectory obtained with the type-1 FLC optimized with CRA. (a) ε = 30, (b) ε = 32, (c) ε = 34.

Figure 19 shows the obtained trajectories for the type-2 FLC optimized with the CRA method.

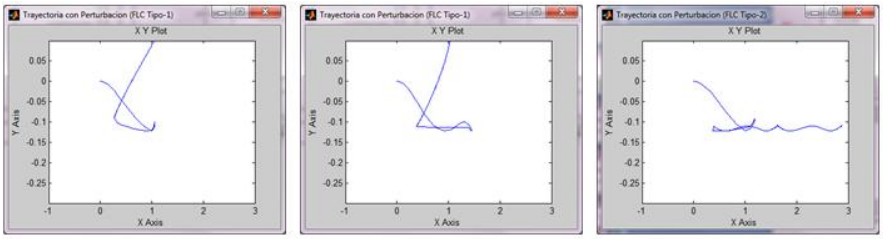

Fig. 19 From left to right, trajectory obtained with the type-2 FLC optimized with CRA. (a) ε = 30, (b) ε = 32, (c) ε = 34.

When observing Table 8 and Figures 17 to 19 we can observe that the type-2 FLC was able to maintain a more controlled trajectory in despite of the "large" error found by the algorithm (e=2.7736). For larger epsilon (ε) values, it was difficult for the type-1 FLC's to keep in the path and in a determined time, the controller was not able to return to the reference trajectory.

6 Conclusions

In this paper, we presented simulation results from an optimization method that mimics chemical reactions applied to the problem of tracking control. The goal was to find the gain constants involved in the tracking controller for the dynamic model of a unicycle mobile robot. In the figures of the experiments we are able to note de behavior of the algorithm and the solutions found through all the iterations. Simulation results show that the proposed optimization method is able to outperform the results previously obtained applying a genetic algorithm optimization technique. The optimal fuzzy logic controller obtained with the

proposed chemical paradigm has been able to reach smaller error values in less time than genetic algorithms. Also, the type-2 fuzzy controller was able to perform better under the presence of disturbance for this problem in despite of the "large" error obtained (e=2.7736). The design of optimal type-2 fuzzy controllers is being performed at the time.

Acknowledgement. The authors would like to thank CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

[1] Aliev, R.A., Pedrycz, W., Guirimov, B.G., Aliev, R.R., Ilhan, U., Babagil, M., et al.: Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization. Information Sciences 181(9), 1591–1608 (2011)

[2] Astudillo, L., Castillo, O., Aguilar, L.: Intelligent Control for a Perturbed Autonomous Wheeled Mobile Robot: a Type-2 Fuzzy Logic Approach. Nonlinear Studies 14(1) (2007)

[3] Bentalba, S., El Hajjaji, A., Rachid, A.: Fuzzy Control of a Mobile Robot: A New Approach. In: Proc. IEEE Int. Conf. on Control Applications, Hartford, CT, pp. 69–72 (October 1997)

[4] Bloch, A.M., Drakunov, S.: Tracking in NonHolonomic Dynamic System Via Sliding Modes. In: Proc. IEEE Conf. on Decision & Control, Brighton, UK, pp. 1127–1132 (1991)

[5] Campion, G., Bastin, G., D'Andrea-Novel, B.: Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. IEEE Trans. on Robotics and Automation 12(1) (February 1996)

[6] Lopez, M., Melin, P., Castillo, O.: Comparative Study of Fuzzy Methods for Response Integration in Ensemble Neural Networks for Pattern Recognition. In: Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition. SCI, vol. 256, pp. 123–140. Springer, Heidelberg (2009)

[7] Cazarez-Castro, N.R., Aguilar, L.T., Castillo, O.: Fuzzy logic control with genetic membership function parameters optimization for the output regulation of a servomechanism with nonlinear backlash. Expert Systems with Applications 37(6), 4368–4378 (2010)

[8] Chwa, D.: Sliding-Mode Tracking Control of Nonholonomic Wheeled Mobile Robots in Polar coordinates. IEEE Trans. on Control Syst. Tech. 12(4), 633–644 (2004)

[9] Fierro, R., Lewis, F.L.: Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics. In: Proc. 34th Conf. on Decision & Control, New Orleans, LA (1995)

[10] Fierro, R., Lewis, F.L.: Control of a Nonholonomic Mobile Robot Using Neural Networks. IEEE Trans. on Neural Networks 9(4), 589–600 (1998)

[11] Fukao, T., Nakagawa, H., Adachi, N.: Adaptive Tracking Control of a NonHolonomic Mobile Robot. IEEE Trans. on Robotics and Automation 16(5), 609–615 (2000)

[12] Ishikawa, S.: A Method of Indoor Mobile Robot Navigation by Fuzzy Control. In: Proc. Int. Conf. Intell. Robot. Syst., Osaka, Japan, pp. 1013–1018 (1991)

[13] Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T.: A Stable Tracking Control Method For a Non-Holonomic Mobile Robot. In: Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, Osaka, Japan, pp. 1236–1241 (1991)
[14] Kolmanovsky, I., McClamroch, N.H.: Developments in NonholonomicNontrol Problems. IEEE Control Syst. Mag. 15, 20–36 (1995)
[15] Lee, T.-C., Lee, C.H., Teng, C.-C.: Tracking Control of Mobile Robots Using the Backsteeping Technique. In: Proc. 5th. Int. Conf. Contr., Automat., Robot. Vision, Singapore, pp. 1715–1719 (December 1998)
[16] Lee, T.H., Leung, F.H.F., Tam, P.K.S.: Position Control for Wheeled Mobile Robot Using a Fuzzy Controller, pp. 525–528. IEEE (1999)
[17] Lee, T.-C., Tai, K.: Tracking Control of Unicycle-Modeled Mobile robots Using a Saturation Feedback Controller. IEEE Trans. on Control Systems Technology 9(2), 305–318 (2001)
[18] Martinez, R., Castillo, O., Aguilar, L.: Optimization of type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Information Sciences 179(13), 2158–2174 (2009)
[19] Meyer, T., Yamamoto, L., Banzhaf, W., Tschudin, C.: Elongation Control in an Algorithmic Chemistry. In: Kampis, G. (ed.) ECAL 2009, Part I. LNCS, vol. 5777, pp. 273–280. Springer, Heidelberg (2011)
[20] Mohammadi, S.M.A., Gharaveisi, A.A., Mashinchi, M., Vaezi-Nejad, S.M.: An evolutionary tuning technique for type-2 fuzzy logic controller. Transactions of the Institute of Measurement and Control 33(2), 223–245 (2011)
[21] Nelson, W., Cox, I.: Local Path Control for an Autonomous Vehicle. In: Proc. IEEE Conf. on Robotics and Automation, pp. 1504–1510 (1988)
[22] Oh, S., Jang, H., Pedrycz, W.: A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization. Expert Systems with Applications 38(9), 11217–11229 (2011)
[23] Pawlowski, S., Dutkiewicz, P., Kozlowski, K., Wroblewski, W.: Fuzzy Logic Implementation in Mobile Robot Control. In: 2nd Workshop on Robot Motion and Control, pp. 65–70 (October 2001)
[24] Sahab, A.R., Moddabernia, M.R.: Backstepping method for a single-link flexible-joint manipulator using genetic algorithm. IJICIC 7(7B), 4161–4170 (2011)
[25] Shi, N.-Y., Chu, C.-P.: A molecular solution to the hitting-set problem in DNA-based supercomputing. Information Sciences 180, 1010–1019 (2010)
[26] Song, K.T., Sheen, L.H.: Heuristic fuzzy-neural Network and its application to reactive navigation of a mobile robot. Fuzzy Sets Systems 110(3), 331–340 (2000)
[27] Tsai, C.-C., Lin, H.-H., Lin, C.-C.: Trajectory Tracking Control of a Laser-Guided Wheeled Mobile Robot. In: Proc. IEEE Int. Conf. on Control Applications, Taipei, Taiwan, pp. 1055–1059 (September 2004)
[28] Ulyanov, S.V., Watanabe, S., Ulyanov, V.S., Yamafuji, K., Litvintseva, L.V., Rizzotto, G.G.: Soft Computing for the Intelligent Robust Control of a Robotic Unicycle with a New Physical Measure for Mechanical Controllability. Soft Computing 2, 73–88 (1998)
[29] Xu, J., Lam, A.Y.S., Li, V.O.K.: Chemical Reaction Optimization for the Grid Scheduling Problem. In: IEE Communication Society, ICC 2010, pp. 1–5 (2010)
[30] Yamamoto, L.: Evaluation of a Catalytic Search Algorithm. In: Proc. 4th Int. Workshop on Nature Inspired Cooperative Strategies for Optimization, NICSO 2010, pp. 75–87 (2010)
[31] Yu, J., Ma, Y., Chen, B., Yu, H., Pan, S.: Adaptive Neural Position Tracking Control for Induction Motors via Backstepping. IJICIC 7(7B), 4503–4516 (2011)