Optimization of Membership Functions of a Fuzzy Logic Controller for an Autonomous Wheeled Mobile Robot Using Ant Colony Optimization

Ricardo Martínez-Marroquín, Oscar Castillo, and José Soria

Tijuana Institute of Technology, Tijuana México
ocastillo@hafsamx.org

Abstract. In this paper we describe the application of a Simple ACO (S-ACO) as a method of optimization for membership functions' parameters of a fuzzy logic controller (FLC) in order to find the optimal intelligent controller for an Autonomous Wheeled Mobile Robot. Simulation results show that ACO outperforms a GA in the optimization of FLCs for an autonomous mobile robot.

1 Introduction

Nowadays, fuzzy logic is one of the most used methods of computational intelligence and with the best future; this is possible thanks to the efficiency and simplicity of Fuzzy Systems since they use linguistic terms similar to those that human beings use.

The complexity for developing fuzzy systems can be found at the time of deciding which are the best parameters of the membership functions, the number of rules or even the best granularity that could give us the best solution for the problem that we want to solve.

A solution for the above mentioned problem is the application of evolutionary algorithms for the optimization of fuzzy systems. Evolutionary algorithms can be a useful tool since its capabilities of solving nonlinear problems, well-constrained or even NP-hard problems. Among the most used methods of evolutionary algorithms we can find: Genetic Algorithms, Ant Colony Optimization, Particle Swarm Optimization, etc.

This paper describes the application of evolutionary algorithms, such as the Ant Colony Optimization as a method of optimization of the parameters of the membership functions of the FLC in order to find the best intelligent controller for an Autonomous Wheeled Mobile Robot.

This paper is organized as follows: Section 2 shows the concept of Ant Colony Optimization and a description of S-ACO which is the technique that was applied

O. Castillo et al. (Eds.): Evolutionary Design of Intelligent Systems, SCI 257, pp. 3–16.
springerlink.com

for optimization. Section 3 presents the problem statement and the dynamic and kinematic model of the unicycle mobile robot. Section 4 shows the fuzzy logic controller proposed and in Section 5 it's described the development of the evolutionary method. In the Section 6 the simulation results are shown. Finally, Section 7 shows the Conclusions.

2 S-ACO Algorithm

Ant Colony Optimization (ACO) is a probabilistic technique that can be used for solving problems that can be reduced to finding good path along graphs. This method is inspired on the behavior presented by ants in finding paths from the nest or colony to the food source.

The S-ACO is an algorithmic implementation that adapts the behavior of real ants to solutions of minimum cost path problems on graphs [11]. A number of artificial ants build solutions for a certain optimization problem and exchange information about the quality of these solutions making allusion to the communication systems of the real ants [5].

Let us define the graph $G=(V,E)$, where V is the set of nodes and E is the matrix of the links between nodes. G has $n_G=|V|$ nodes. Let us define L^K as the number of hops in the path built by the ant k from the origin node to the destiny node. Therefore, it is necessary to find:

$$Q=\{q_a,\ldots,q_f \mid q_1 \in C\} \tag{1}$$

Where Q is the set of nodes representing a continuous path with no obstacles; $q_a,\ldots,q_f$ are former nodes of the path and C is the set of possible configurations of the free space. If $x^k(t)$ denotes a Q solution in time t, $f(x^k(t))$ expresses the quality of the solution. The general steps of S-ACO are the followings:

- Each link (i,j) is associated with a pheromone concentration denoted as τ_{ij}.
- A number $k=1,2,\ldots,n_k$ are placed in the nest.
- On each iteration all ants build a path to the food source (destiny node). For selecting the next node a probabilistic equation is used:

$$p_{ij}^k(t)\begin{cases} \dfrac{\tau_{ij}^k}{\sum_{j\in N_{ij}^k}\tau_{ij}^{\alpha}(t)} & if \quad j\in N_i^k \\ 0 & if \quad j\notin N_i^k \end{cases} \tag{2}$$

Where, N_i^k is the set of feasible nodes (in a neighborhood) connected to node *i* with respect to ant *k*, τ_{ij} is the total pheromone concentration of link *ij*, and α is a positive constant used as again for the pheromone influence.

- Remove cycles and compute each route weight $f\left(x^k(t)\right)$. A cycle could be generated when there are no feasible candidates nodes, that is, for any *i* and any *k*, $N_i^k = \varnothing$; then the predecessor of that node is included as a former node of the path.
- Pheromone evaporation is calculated with equation (3):

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) \tag{3}$$

Where $\rho \in [0,1]$ is the evaporation rate value of the pheromone trail. The evaporation is added to the algorithm in order to force the exploration of the ants, and avoid premature convergence to sub-optimal solutions [11]. For $\rho = 1$ the search becomes completely random [11].

- The update of the pheromone concentration is realized using equation (4):

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t) \tag{4}$$

Where $\Delta\tau_{ij}^k$ is the amount of pheromone that an ant *k* deposits in a link *ij* in a time *t*.

- Finally, the algorithm can be ended in three different ways:
  - When a maximum number of epochs has been reached.
  - When it has been found an acceptable solution, with $f\left(x_k(t)\right) < \varepsilon$.
  - When all ants follow the same path

3 Problem Statement

The model of the robot considered in this paper is a unicycle mobile robot (see Figure 1), that consists of two driving wheels mounted of the same axis and a front free wheel.

A unicycle mobile robot is an autonomous, wheeled vehicle capable of performing missions in fixed or uncertain environments. The robot body is symmetrical around the perpendicular axis and the center of mass is at the geometrical center of the body. It has two driving wheels that are fixed to the axis that passes

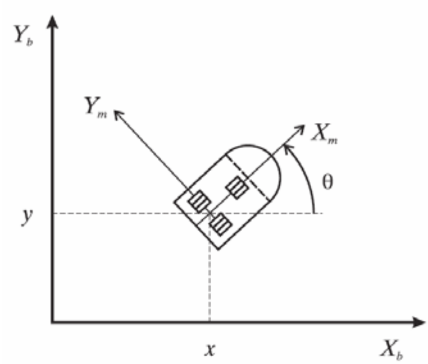


Fig. 1. Wheeled Mobile Robot [10]

through *C* and one passive wheel prevents the robot from tipping over as it moves on a plane. In what follows, it´s assumed that motion of the passive wheel can be ignored in the dynamics of the mobile robot presented by the following set of equations [8]:

$$M(q)\dot{\vartheta} + C(q,\dot{q})\vartheta + D\vartheta = \tau + F_{ext}(t) \tag{5}$$

$$\dot{q} = \underbrace{\begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_{\vartheta} \tag{6}$$

Where $q = (x, y, \theta)^T$ is the vector of the configuration coordinates; $\vartheta = (v, w)^T$ is the vector of linear and angular velocities; $\tau = (\tau_1, \tau_2)$ is the vector of torques applied to the wheels of the robot where τ_1 and τ_2 denote the torques of the right and left wheel respectively (Figure 1); $F_{ext} \in \mathbb{R}^2$ uniformly bounded disturbance vector; $M(q) \in \mathbb{R}^{2\times 2}$ is the positive-definite inertia matrix; $C(q,\dot{q})\vartheta$ is the vector of centripetal and Coriolis forces; and $D \in \mathbb{R}^{2\times 2}$ is a diagonal positive-definite damping matrix. Equation (6) represents the kinematics of the system, where (x, y) is the position of the mobile robot in the X-Y (world) reference frame, θ is the angle between heading direction and the *x*-axis *v* and w are the angular and angular velocities, respectively.

Furthermore, the system (5)-(6) has the following non-holonomic constraint:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{7}$$

which corresponds to a no-slip wheel condition preventing the robot from moving sideways[9]. The system (6) fails to meet Brockett's necessary condition for feedback stabilization [2], which implies that anon-continuous static state-feedback controller exists that stabilizes the close-loop system around the equilibrium point.

The control objective is to design a fuzzy logic controller of τ that ensures:

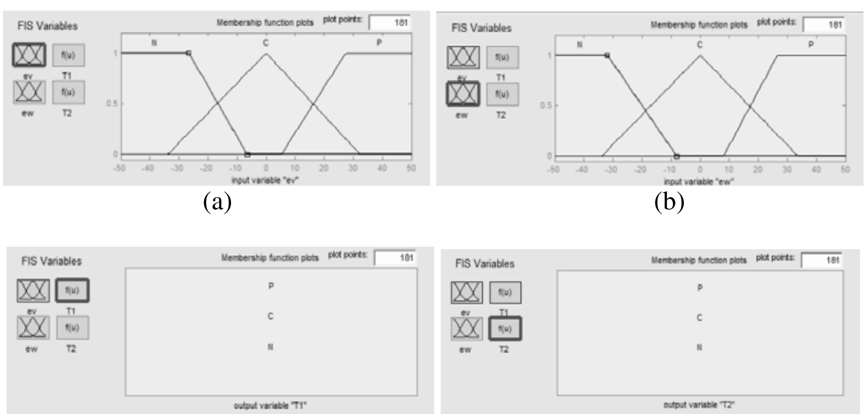
$$\lim_{t\to\infty}\left\| q_d(t) - q(t) \right\| = 0 \tag{8}$$

for any continuously, differentiable, bounded desired trajectory $q_d \in \mathbb{R}^3$ while attenuating external disturbances.

A more detailed description can be found on reference [10].

4 Fuzzy Logic Control Design

In order to satisfy the control objective it is necessary to design a fuzzy logic controller for the real velocities of the mobile robot. To do that, a Takagi-Sugeno fuzzy logic controller was designed, using linguistic variables in the input and mathematical functions in the output. The error of the linear and angular velocities (v_d, w_d respectively), were taken as inputs variables, while the right (τ_1) and left (τ_2) torques as outputs. The membership functions used on the input are trapezoidal for the negative (*N*) and positive (*P*), and a triangular was used for the zero (*C*) linguistic terms. The interval used for this fuzzy controller is [-50 50] [10].



(a) (b) (c) (d)

Fig. 2. (a) Linear velocity error. (b) Angular velocity error. (c) Right output (τ_1). (d) Left output (τ_2).

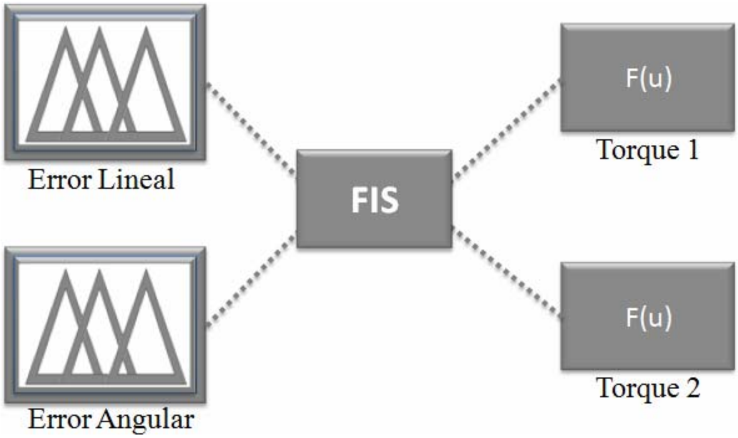


Fig. 3. Fuzzy Logic Controller Architecture.

Figure 2 shows the input and output variables, and figure 3 shows the general FLC architecture.

The rule set of the FLC contains 9 rules, which governs the input-output relationship of the FLC and this adopts the Takagi-Sugeno style inference engine [10], and it is used with a single point in the outputs, this mind that the outputs are constant values, obtained using weighted average defuzzification procedure. In Table 1 we present the rule set whose format is established as follows:

$$\text{Rule } i: \text{if } e_v \text{ is } \mathrm{G}_1 \text{ and } e_w \text{ is } G_2 \text{ then } F \text{ is } G_3 \text{ and } N \text{ is } G_4$$

where $G_1..G_4$ are the fuzzy set associated to each variable i=1,2,…,9.

Table 1. Fuzzy rules set

e_v / e_w	N	C	P
N	N / N	N / C	N / P
C	C / N	C / C	C / P
P	P / N	P / C	P / P

To find the best FLC, we used a S-ACO to find the parameters of the membership functions. Table 2 shows the parameters of the membership functions, the minimal and maximum values in the search range for the S-ACO algorithm to find the best fuzzy logic controller.

It is important to remark that values shown in Table 2 are applied to both inputs and both outputs of the fuzzy logic controller.

Table 2. Parameters of the membership Functions.

MF TYPE	POINT	MINIMAL VALUE	MAXIMAL VALUE
Trapezoidal	a	-50	-50
	b	-50	-50
	c	-15	-5.1
	d	-1.5	-0.5
Triangular	a	-5	-1.8
	b	0	0
	c	1.8	5
Trapezoidal	a	0.5	1.5
	b	5.1	15
	c	50	50
	d	50	50
Constant (N)	a	-50	-50
Constant (C)	a	0	0
Constant (P)	a	50	50

5 ACO Architecture

A S-ACO algorithm was applied for the optimization of the membership functions for the fuzzy logic controller. For developing the architecture of the algorithm it was necessary to follow the next steps:

1. Marking the limits of the problem in order to eliminate unnecessary complexity.
2. Representing the architecture of the FLC as a graph that artificial ants could traverse.
3. Achieving an adequate handling of the pheromone but permitting the algorithm to evolve by itself

5.1 Limiting the Problem and Graph Representation

One of problems found on the development of the S-ACO algorithm was to make a good representation of FLC. First we reduced the number of elements that the method needed to find by deleting the elements whose minimal value and maximal values are the same (see Table 2) and therefore if they were included they will not change any way. Table 3 shows the parameters of the membership functions included in the search.

The next step was to represent those parameters shown in table 3; to that, was necessary to discretize the parameters in a range of possible values in order to represent every possible value as a node in the graph of search. The level of discretization between minimal and maximal value was of 0.1 (by example: -1.5, -1.4, -1.3,…, -0.5).

Table 3. Parameters of the Membership Functions Included in S-ACO Search

MF TYPE	POINT	MINIMAL VALUE	MAXIMAL VALUE
Trapezoidal	c	-15	-5.1
	d	-1.5	-0.5
Triangular	a	-5	-1.8
	c	1.8	5
Trapezoidal	a	0.5	1.5
	b	5.1	15

Table 4 shows the number of possible values that each parameter can take.

Table 4. Number of Possible Values of the Parameters of Membership Functions

MF TYPE	POINT	COMBINATIONS
Trapezoidal	c	100
	d	15
Triangular	a	33
	c	33
Trapezoidal	a	15
	b	100

Figure 4 shows the search graph for the proposed S-ACO algorithm, the graph can be viewed as a tree where the root is the nest and the last node is the food source.

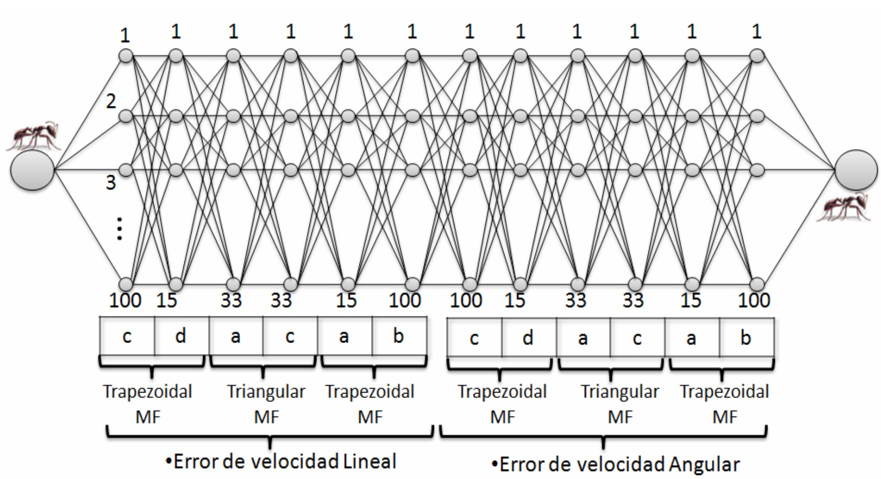


Fig. 4. S-ACO Architecture

5.2 Updating Pheromone Trail

An important issue is that the update of pheromone trail be applied in the best way possible. In this sense we need to handle the evaporation (Equation 3), and increase or deposit of pheromone (Equation 4), where the key parameter in evaporation is denoted by ρ that represents the rate of evaporation and in deposit of pheromone is denoted by $\Delta\tau$ that represents the amount of pheromone that an ant *k* deposits in a link *ij* in a time *t*. For ρ we assign a random value and Equation 9 shows the way how the increase of pheromone is calculated.

$$\Delta\tau = \frac{(e_{\max} - e_k)}{e_{\max}} \tag{9}$$

Where $e_{\max} = 10$ is the maximum error of control permitted and e_k is error of control generated by a complete path of an ant *k*. We decided to allocate $e_{\max} = 10$ in order to stand $\Delta\tau \in [0,1]$.

6 Simulation Results

In this section we present the results of the proposed controller to stabilize the unicycle mobile robot, defined by Equation (5) and Equation (6), where the matrix values

$$M(q) = \begin{bmatrix} 0.3749 & -0.0202 \\ -0.0202 & 0.3739 \end{bmatrix},$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & 0.1350\dot{\theta} \\ -0.150\dot{\theta} & 0 \end{bmatrix},$$

and $D = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$

were taken from [6]. The evaluation was made through computer simulation performed in MATLAB® and SIMULINK®.

The desired trajectory is the following one:

$$\vartheta_d(t) = \begin{cases} v_d(t) = 0.2(1 - \exp(-t)) \\ w_d(t) = 0.4\sin(0.5t) \end{cases} \tag{10}$$

Table 5. S-ACO Results of Simulations for FLC Optimization

Iterations	Ants	α	ρ	Average Error	Time
20	10	0.2	random	1.5589	00:01:30
20	10	0.2	random	1.451	00:01:34
25	10	0.2	random	1.5566	00:01:46
25	10	0.2	random	1.4767	00:01:51
25	10	0.2	random	1.4739	00:02:05
25	10	0.2	random	1.6137	00:02:08
25	10	0.2	random	1.6642	00:01:54
25	100	0.2	random	1.3484	00:20:30
25	100	0.2	random	1.3413	00:18:44
25	100	0.2	random	1.3360	00:18:31
25	100	0.2	random	1.2954	00:18:32
25	100	0.2	random	1.4877	00:18:41
25	100	0.2	random	1.2391	00:18:31
10	15	0.2	random	1.6916	00:01:14
10	15	0.2	random	1.4256	00:01:09
40	65	0.2	random	1.2783	00:19:17
40	65	0.2	random	1.4011	00:19:45
40	65	0.2	random	1.2216	00:19:33
40	65	0.2	random	1.2487	00:19:49
50	70	0.2	random	1.3782	00:26:09
50	70	0.2	random	1.0875	00:27:35
50	70	0.2	random	1.4218	00:33:45
50	70	0.2	random	1.475	01:08:48
25	80	0.2	random	1.4718	00:14:55
25	80	0.2	random	1.4212	00:15:00
25	80	0.2	random	1.3221	00:14:52
25	80	0.2	random	1.1391	00:15:41
50	80	0.2	random	1.2148	00:28:43
62	**50**	**0.2**	**random**	**1.0322**	**00:24:49**
50	80	0.2	random	1.1887	00:29:55
50	80	0.2	random	1.2158	00:29:56
60	90	0.2	random	1.3493	00:41:56
60	90	0.2	random	1.3060	00:39:48
60	90	0.2	random	1.3161	00:40:00

and was chosen in terms of its corresponding desired linear v_d and angular w_d velocities, subject to the initial conditions

$$q(0) = (0.1, 0.1, 0,)^T \text{ and } \vartheta(0) = 0 \in \mathbb{R}^2$$

The gains γ_i, i=1, 2, 3 of the kinematic model (see [10]) are $\gamma_1 = 5$, $\gamma_2 = 24$ and $\gamma_3 = 3$ were taken from [10].

6.1 S-ACO Algorithm Results for the Optimization of the FLC

Table 5 shows the results of the FLC, obtained varying the values of maximum iterations and number of artificial ants, where the highlighted row shows the best result obtained with the method. Figure 5 shows the evolving of the method.

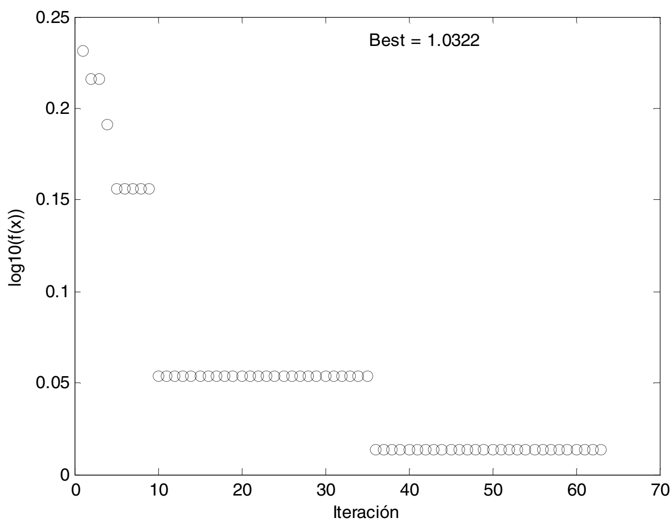


Fig. 5. Evolution of the S-ACO for FLC Optimization.

Figure 6 shows the membership functions of the FLC obtained by S-ACO algorithm.

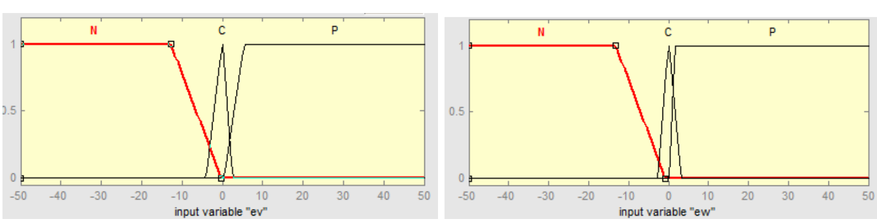


Fig. 6. (a) Linear velocity error, and (b) angular velocity error optimized by S-ACO algorithm.

Figure 7 shows the block diagram used for the FLC that obtained the best results. Figure 8 shows the results of linear and angular errors, and Figure 9 shows the output results of the fuzzy controller that represents the torque applied to the wheels of the autonomous mobile robot.

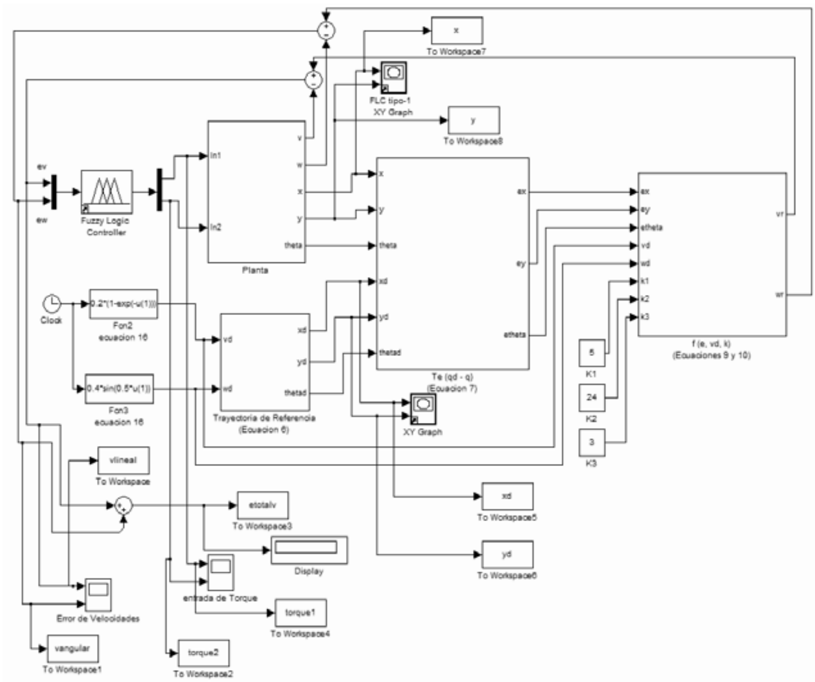


Fig. 7. Block diagram for simulation of the FLC

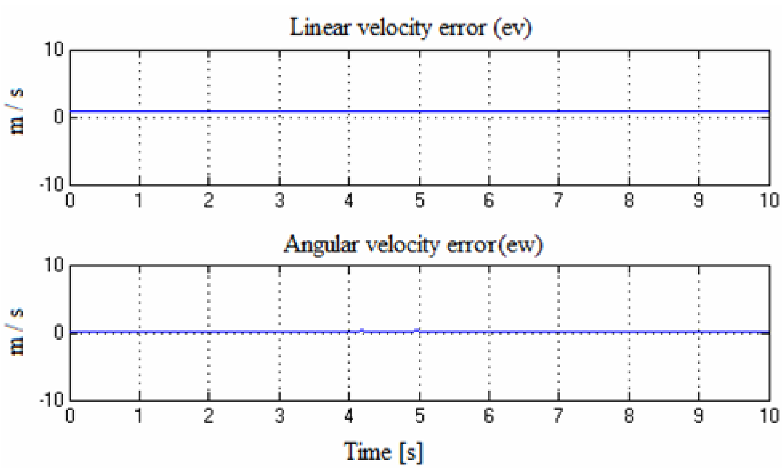


Fig. 8. Linear and angular velocity errors

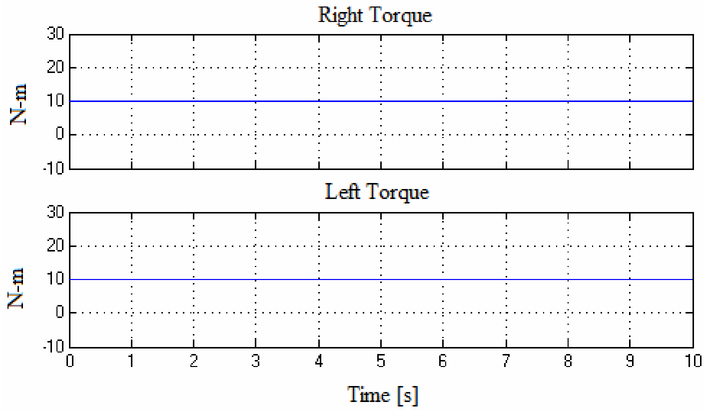


Fig. 9. Right and left torques

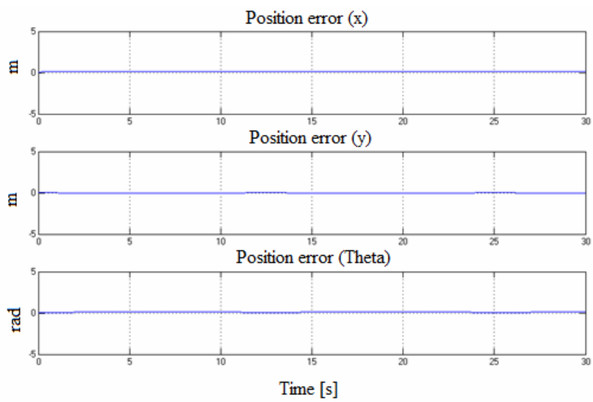


Fig. 10. Position errors in x, y, θ .

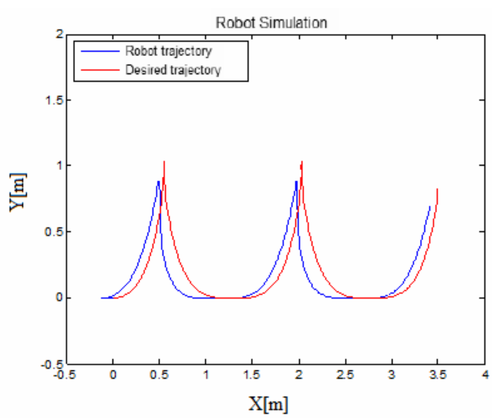


Fig. 11. Obtained trajectory

The positions errors of the autonomous mobile robot can be observed in Figure 10. Figure 11 shows the desired trajectory and obtained trajectory.

7 Conclusions

A trajectory tracking controller has been designed based on the dynamics and kinematics of the autonomous mobile robot through the application of ACO for the optimization of membership functions for the fuzzy logic controller with good results obtained after simulations.

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

References

[1] Bloch, A.M.: Nonholonomic mechanics and control. Springer, New York (2003)
[2] Brockett, R.W.: Asymptotic stability and feedback stabilization. In: Millman, R.S., Susman, H.J. (eds.) Diferential Geometric Control Theory, p. 181. Birkhauser, Boston (1983)
[3] Castillo, O., Melin, P.: Soft Computing for Control of Non-Linear Dynamical Systems. Springer, Helderberg (2001)
[4] Dorigo, M., Birattari, M., Stützle, T.: Ant Colony Optimization. IEEE Computational Intelligence Magazine, 28–39 (November 2006)
[5] Dorigo, M., Stützle, T.: Ant Colony Optimization, Bradford, Cambridge, Massachusetts (2004)
[6] Duc Do, K., Zhong-Ping, Y., Pan, J.: A global output-feedback controller for simultaneous tracking and stabilizations of unicycle-type mobile robots. IEEE Trans. Automat. Contr. 20(3), 589–594 (2004)
[7] Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. Wiley, England (2005)
[8] Lee, T.-C., Song, K.-T., Lee, C.-H., Teng, C.-C.: Tracking control of unicycle-modeled mobile robot using a saturation feedback controller. IEEE trans. Contr. Syst. Technol. 9(2), 305–318 (2001)
[9] Liberzon, D.: Switching in Systems and control. Bikhauser (2003)
[10] Martínez, R., Castillo, O., Aguilar, L.: Intelligent control for a perturbed autonomous wheeled mobile robot using type-2 fuzzy logic and genetic algorithms. JAMRIS 2(1), 1–11 (2008)
[11] Porta-García, M., Montiel, O., Sepulveda, R.: An ACO path planner using a FIS for a Path Selection Adjusted with a Simple Tuning Algorithm. JAMRIS 2(1), 1–11 (2008)
[12] Sepúlveda, R., Castillo, O., Melin, P., Montiel, O.: An Efficient Computational Method to Implement Type-2 Fuzzy Logic In control Applications, Analysis and Design of Intelligent Systems Using Soft Computing Techniques. Advances in Soft Computing 41, 45–52 (2007)