Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Optimization of Type-2 Fuzzy Logic Controllers for Mobile Robots Using Evolutionary Methods

Ricardo Martinez and Antonio Rodriguez
UABC, Tijuana, México
mc.ricardo.martinez@hotmail.com,
ardiaz@uabc.mx

Oscar Castillo, Patricia Melin
Instituto Tecnológico de Tijuana
Tijuana, México
{ocastillo, epmelin}@hafsamx.org

Luis T. Aguilar
CITEDI
Tijuana, México
luis.aguilar@ieee.org

***Abstract*— We describe in this paper the application of evolutionary methods for the optimization of type-2 fuzzy logic controllers. These optimal type-2 fuzzy logic controllers are used for the trajectory tracking control of autonomous mobile robots. The evolutionary method to consider is the genetic algorithm, presenting simulations results in Simulink© of Matlab.**

***Keywords*— Autonomous Mobile Robot, Evolutionary Methods, Genetic Algorithms, Type-2 Fuzzy Logic.**

I. INTRODUCTION

Evolutionary Computing is the collective name for a range of problem-solving techniques based on the principles of biological evolution, such as natural selection and genetic inheritance. These techniques are being increasingly widely applied to a variety of problems, ranging from practical applications in industry and commerce to leading-edge scientific research [1]. There are many techniques for evolutionary computing and for this study we used genetic algorithms (GA). This technique is used to optimize a type-2 fuzzy system. Once the optimized type-2 fuzzy logic controllers (FLC) are obtained, we apply them to an autonomous mobile robot for the trajectory tracking control.

This paper is organized as follows: Section II presents the theoretical basis and problem statement; Section III presents the Type-2 Fuzzy Logic Controller design. Section IV provides simulation results of the evolutionary computing techniques using the controller described in Section III. Finally, Section V presents the conclusions.

II. THEORETICAL BASIS AND PROBLEM STATEMENT

A. Type-2 Fuzzy Logic Systems

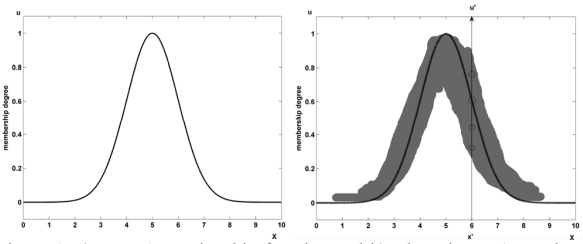


Figure 1. a) Type-1 membership function and b) Blurred type-1 membership function.

If we have a type-1 membership function, as in Figure 1 (a), and we are blurring it to the left and to the right as illustrated in Figure 1 (b), then, for a specific value x', the membership function (u'), takes on different values, which are not all weighted the same, so we can assign an amplitude distribution to all of those points. Doing this for all $x \in X$, we create a three-dimensional membership function –a type-2 membership function– that characterizes a type-2 fuzzy set [1, 11]. A type-2 fuzzy set $\widetilde{A}$, is characterized by the membership function:

$$\widetilde{A} = \left\{\left((x,u),\mu_{\widetilde{A}}(x,u)\right) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\right\} \quad (1)$$

in which $0 \le \mu_{\widetilde{A}}(x,u) \le 1$. Another expression for $\widetilde{A}$ is,

$$\widetilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\widetilde{A}}(x,u)/(x,u) \qquad J_x \subseteq [0,1] \quad (2)$$

where $\int\int$ denotes union over all admissible input variables x and u. For discrete universes of discourse $\int$ is replaced by $\sum$ [11]. In fact $J_x \subseteq [0,1]$ represents the primary membership of x, and $\mu_{\widetilde{A}}(x,u)$ is a type-1 fuzzy set known as the secondary set. Hence, a type-2 membership grade can be any subset in [0,1], the primary membership, and corresponding to each primary membership, there is a secondary membership (which can also be in [0,1]) that defines the possibilities for the primary membership [10].
This uncertainty is represented by a region called footprint of uncertainty (FOU). When $\mu_{\widetilde{A}}(x,u) = 1, \forall u \in J_x \subseteq [0,1]$ we have an interval type-2 membership function, as shown in Figure 2.

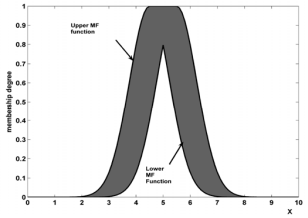


Figure 2. Interval type-2 membership function.

The uniform shading for the FOU represents the entire interval type-2 fuzzy set and it can be described in terms of an upper membership function $\overline{\mu}_{\widetilde{A}}(x)$ and a lower membership function $\underline{\mu}_{\widetilde{A}}(x)$. A Fuzzy Logic System (FLS) described using at least one type-2 fuzzy set is called a type-2 FLS.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain [11]. On the other hand, type-2 FLSs, are very useful in circumstances where it is difficult to determine an exact membership function, and there are measurement uncertainties [13].

It is known that type-2 fuzzy sets enable modeling and minimizing the effects of uncertainties in rule-based FLS. Unfortunately, type-2 fuzzy sets are more difficult to use and understand than type-1 fuzzy sets; hence, their use is not widespread yet. As a justification for the use of type-2 fuzzy sets, in [12] are mentioned at least four sources of uncertainties not considered in type-1 FLSs:

1. The meanings of the words that are used in the antecedents and consequents of rules can be uncertain (words mean different things to different people).
2. Consequents may have a histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not all agree.
3. Measurements that activate a type-1 FLS may be noisy and therefore uncertain.
4. The data used to tune the parameters of a type-1 FLS may also be noisy.

All of these uncertainties translate into uncertainties about fuzzy set membership functions. Type-1 fuzzy sets are not able to directly model such uncertainties because their membership functions are totally crisp. On the other hand, type-2 fuzzy sets are able to model such uncertainties because their membership functions are themselves fuzzy. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity.

A type-2 FLS is again characterized by IF-THEN rules, but its antecedent or consequent sets are now of type-2. Similar to a type-1 FLS, a type-2 FLS includes a fuzzifier, a rule base, fuzzy inference engine, and an output processor, as we can see in Figure 3.

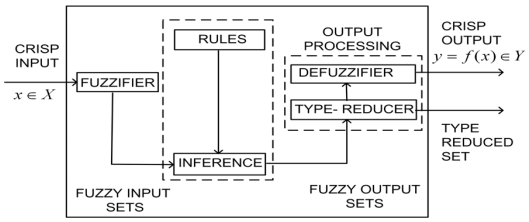


Figure 3 Type-2 Fuzzy Logic System.

The output processor includes type-reducer and defuzzifier; it generates a type-1 fuzzy set output (from the type-reducer) or a crisp number (from the defuzzifier) [4,3].

B. Evolutionary Methods Applied to Fuzzy Systems

Fuzzy systems have been successfully applied to problems in classification [4], modeling [14], control [7], and in a considerable number of applications. In most cases, the key for success was the ability of fuzzy systems to incorporate human expert knowledge. In the 1990s, despite the previous successful history, the lack of learning capabilities characterizing most of the works in the field generated a certain interest for the study of fuzzy systems with added learning capabilities.

Two of the most successful approaches have been the hybridization attempts made in the framework of soft computing, were different techniques, such as the neural and evolutionary; provided fuzzy systems with learning capabilities. Neuro-fuzzy systems are one of the most successful and visible directions of that effort. A different approach to achieve hybridization has lead to genetic fuzzy systems (GFSs). A GFS is basically a fuzzy system augmented by a learning process based on a genetic algorithm (GA) [5].

GAs are search algorithms, based on natural genetics, that provide robust search capabilities in complex spaces, and thereby offer a valid approach to problems requiring efficient and effective search processes [12,13,14]. Genetic learning processes cover different levels of complexity according to the structural changes produced by the algorithm [6], from the simplest case of parameter optimization to the highest level of complexity of learning the rule set of a rule based system. Parameter optimization has been the approach utilized to adapt a wide range of different fuzzy systems, as in genetic fuzzy clustering or genetic neuro-fuzzy systems [8].

An analysis of the literature shows that the most prominent types of GFSs are genetic fuzzy rule-based systems (GFRBSs) [5], whose genetic process learns or tunes different components of a fuzzy rule-based system (FRBS). Inside GFRBSs it is possible to distinguish between either parameter optimization or rule generation processes, that is, adaptation and learning.

There are different kinds of genetic fuzzy systems applications, but we focused on genetic tuning for parameter optimization of membership functions [1] [2] [3].

C. The Mobile Robot

The model considered is a unicycle mobile robot (Figure 4), it consists of two driving wheels mounted of the same axis and a front free wheel.

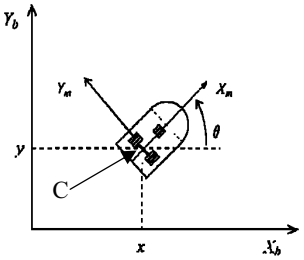


Figure 4. Wheeled Mobile Robot.

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 geometric center of the body. It has two driving wheels that are fixed to the axis that passes through center of mass "C" and one

passive wheel prevents the robot from tipping over as it moves on a plane.

In what follows, it is assumed that the motion of the passive wheel can be ignored in the dynamics of the mobile robot represented by the following set of equations [9]:

$$M(q)\dot{v} + C(q,\dot{q})v + Dv = \tau + P(t) \tag{3}$$

$$\dot{q} = \underbrace{\begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_{\upsilon} \tag{4}$$

where $q = (x, y, \theta)^T$ is the vector of the configuration coordinates; $\upsilon = (v, w)^T$ is the vector of 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 4); $P \in R^2$ is the uniformly bounded disturbance vector; $M(q) \in R^{2\times 2}$ is the positive-definite inertia matrix; $C(q,\dot{q})\vartheta$ is the vector of centripetal and Coriolis forces; and $D \in R^{2\times 2}$ is a diagonal positive-definite damping matrix. Equation (4) represents the kinematics of the system, where (x, y) is the position in the X – Y (world) reference frame; θ is the angle between the heading direction and the x -axis; v and w are the linear and angular velocities, respectively. Furthermore, the system (3)-(4) has the following non-holonomic constraint:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{5}$$

which corresponds to a no-slip wheel condition preventing the robot from moving sideways [14]. The system (4) fails to meet Brockett's necessary condition for feedback stabilization [11], which implies that no continuous static state-feedback controller exists that, stabilizes the closed-loop system around the equilibrium point.

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

$$\lim_{t\to\infty} \| q_d(t) - q(t) \| = 0, \tag{6}$$

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

III. Fuzzy Logic Control Design

This section illustrates the framework to achieve stabilization of a unicycle mobile robot around a desired path. The stabilizing control law for the system (3)-(4) can be designed using the backstepping approach [16] since the kinematics subsystem (4) is controlled indirectly through the velocity vector v [15]. The procedure to design the overall controller consists of two steps:

-Design a virtual velocity vector $\vartheta_r = \vartheta$ such that the kinematic model (4) be uniformly asymptotically stable.

-Design a velocity controller τ by using FLC that ensures

$$\| \vartheta_r(t) - \vartheta(t) \| = 0, \qquad \forall t \geq t_s \tag{7}$$

where t_s is the reachability time.

In (7), it is considered that real mobile robots have actuated wheels, so the control output is τ that must be designed to stabilize the dynamics (3), without destabilizing the system (4), by forcing $\vartheta \in R^2$ to reach the virtual velocity vector $\vartheta_r \in R^2$ in finite-time. Roughly speaking, if (7) is satisfied asymptotically $(i.e; t_s = \infty)$ then ϑ along $t < \infty$, consequently the mobile robot will be neither positioned nor oriented at desired point. Figure 5 illustrates the feedback connection which involves the fuzzy controller.

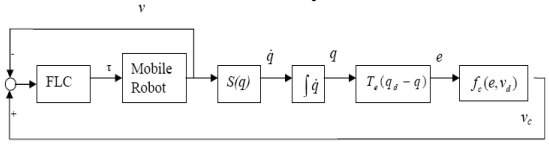


Figure 5. Tracking control structure.

A. Posture Control Design.

First, we focus on the kinematic model by designing a virtual control (ϑ_r) such that the control objective (6) is achieved. To this end, let us consider the reference trajectory $q_d(t)$ as a solution of the following differential equation:

$$\dot{q}_d = \begin{pmatrix} \cos\theta_d & 0 \\ \sin\theta_d & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_d \\ w_d \end{pmatrix} \tag{8}$$

where $\theta_d(t)$ is the desired orientation, and $v_d(t)$ and $w_d(t)$ denote the desired linear and angular velocities, respectively.

In the robot's local frame, the error coordinates can be defined as

$$\begin{pmatrix} \tilde{q}_1 \\ \tilde{q}_2 \\ \tilde{q}_3 \end{pmatrix} = \underbrace{\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{T_e(\theta)} \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{pmatrix} \tag{9}$$

where $(x_d(t), y_d(t))$ is the desired position in the world X – Y coordinate system, $\tilde{q}_1$ and $\tilde{q}_2$ are the coordinates of the position error vector, and $\tilde{q}_3$ is the orientation error. The associated tracking error model is

$$\begin{pmatrix} \dot{\tilde{q}}_1 \\ \dot{\tilde{q}}_2 \\ \dot{\tilde{q}}_3 \end{pmatrix} = \begin{pmatrix} w\tilde{q}_2 - v + v_d\cos\tilde{q}_3 \\ -w\tilde{q}_1 + v_d\sin\tilde{q}_3 \\ w_d - w \end{pmatrix} \tag{10}$$

which is in terms of the corresponding real and desired velocities, is then obtained by differentiating (9) with respect to time. In order to present the main result of this subsection, we need first to recall the following theorems [11].

Theorem 1 [16](Uniform stability): Let $x = 0$ be an equilibrium point for $\dot{x} = f(x,t)$ and $D \subset R^n$ be a domain containing $x = 0$. Let $V : [0,\infty) \times D \to R$ be a continuously differentiable function such that

$$W_1(x) \leq V(x,t) \leq W_2(x) \tag{11}$$

$$\frac{\partial V}{\partial t}+\frac{\partial V}{\partial x}f(x,t)\le 0 \qquad (12)$$

for all $t\ge 0$ and for all $x\in D$, where $W_1(x)$ and $W_2(x)$ are continuous positive definite functions on *D*. Then, $x=0$ is uniformly stable.

Theorem 2 [16](Uniform asymptotic stability): Suppose the assumptions of Theorem 1 are satisfied with inequality (12) strengthened to

$$\frac{\partial V}{\partial t}+\frac{\partial V}{\partial x}f(x,t)\le -W_3(x) \qquad (13)$$

for all $t\ge 0$ and for all $x\in D$, where $W_3(x)$ is a continuous positive definite function on *D*. Then, $x=0$ is uniformly asymptotically stable.

Theorem 3: Let the tracking error equations (10) be driven by the control law (virtual velocities)

$$\begin{aligned} v_r &= v_d\cos\tilde{q}_3+\gamma_1\tilde{q}_1 \\ w_r &= w_d+\gamma_2 v_d\tilde{q}_2+\gamma_3\sin\tilde{q}_3 \end{aligned} \qquad (14)$$

where γ_1,γ_2, and γ_3 are positive constants. If $v=v_r$ and $w=w_r$ for all $t\ge 0$ in (3), then the origin of the closed-loop system (10)-(14) is uniformly asymptotically stable.

Proof: Under the control (9), the closed-loop system takes the form:

$$\begin{pmatrix}\dot{\tilde{q}}_1\\ \dot{\tilde{q}}_2\\ \dot{\tilde{q}}_3\end{pmatrix}=\begin{pmatrix} w_d\tilde{q}_2+\gamma_2 v_d\tilde{q}_2^2+\gamma_3\tilde{q}_3-\gamma_1\tilde{q}_1 \\ -w_d\tilde{q}_1-\gamma_2 v_d\tilde{q}_1\tilde{q}_2-\gamma_3\tilde{q}_1\sin\tilde{q}_3+v_d\sin\tilde{q}_3 \\ -\gamma_2 v_d\tilde{q}_2-\gamma_3\sin\tilde{q}_3\end{pmatrix}. \qquad (15)$$

Note that the origin $(\tilde{q}_1,\tilde{q}_2,\tilde{q}_3)^T=0$ is an equilibrium point of the closed-loop system but not unique because $\tilde{q}_3$ can adopt several postures $(i.e.,\tilde{q}_3=0,\pi,...,n\pi)$. Genetic algorithms are applied for tuning the kinematic control gains $\gamma_i, i=1,2,3$ to ensure that the error $\tilde{q}\in R^3$ converges to the origin. The asymptotic stability theorem is invoked as a guideline to obtain bounds in the values of γ_i, which shall guarantee convergence of the error $\tilde{q}\in R^3$ to zero. For this purpose, let us introduce the Lyapunov function candidate

$$V(\tilde{q})=\frac{1}{2}\tilde{q}_1^2+\frac{1}{2}\tilde{q}_2^2+\frac{1}{\gamma_2}(1-\cos\tilde{q}_3) \qquad (16)$$

which is positive definite. Taking the time derivative of $V(\tilde{q})$ along the solution of the closed-loop system (15), we get

$$\dot{V}(\tilde{q})=\tilde{q}_1\dot{\tilde{q}}_1+\tilde{q}_2\dot{\tilde{q}}_2+\frac{1}{\gamma_2}\dot{\tilde{q}}_3\sin\tilde{q}_3=-\gamma_1\tilde{q}_1^2-\frac{\gamma_3}{\gamma_2}(\sin\tilde{q}_3)^2\le 0. \qquad (17)$$

Thus concluding that for any positive constant γ_i, the closed-loop system is uniformly stable. To complete the proof it remains to note that $\tilde{q}_1,\tilde{q}_2,\tilde{q}_3\in L_\infty^n;\tilde{q}_1,\tilde{q}_3\in L_\infty^n$ and $\dot{\tilde{q}}_1,\dot{\tilde{q}}_2\in L_\infty^n$ where

$$L_2^n=\left\{x(t):R_+\mapsto R^n \,\middle|\, \|x(t)\|_\infty^2=\int_0^\infty\|x(t)\|_2^2\,dt<\infty\right\} \qquad (18)$$

$$L_2^n=\left\{x(t):R_+\mapsto R^n \,\middle|\, \|x(t)\|_\infty^2=\sup\|x(t)\|_2^2<\infty\right\} \qquad (19)$$

hence we conclude, by applying Barbalat's lemma that $\tilde{q}_1$ and $\tilde{q}_3$ converge to the origin. Finally, by invoking the Matrosov's Theorem [13], convergence of $\tilde{q}_2$ to the origin can be concluded.

The genetic algorithm was codified with a chromosome of 24 bits in total, eight bits for each of the gains. Figure 6 shows the binary chromosome representation of the individuals in the population. Different experiments were performed, changing the parameters of the genetic algorithm and the best results were obtained by comparing the corresponding simulations.

The advantage of using the genetic algorithm to find the gains, is the time-consuming manual search of these parameters was avoided.

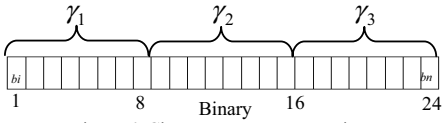


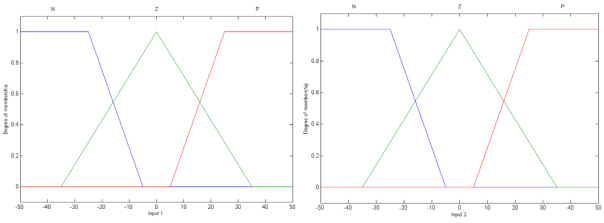
Figure 6. Chromosome representation.

where b*i*, *i* = 1, . . . 24 are binary values (0 or 1) representing the constants gains parameters.

B. Velocity Control Design

In this subsection a fuzzy logic controller is designed to force the real velocities of the mobile robot (3) and (4) to match those required in equations (14) of Theorem 3 to satisfy the control objective (6).

We design a Takagi-Sugeno fuzzy logic controller for the autonomous mobile robot, using linguistic variables in the input and mathematical functions in the output. The linear (ϑ_d) and the angular (w_d) velocity errors were taken as input variables and the right (τ_1) and left (τ_2) torques as the outputs. The membership functions used in the input are trapezoidal for the Negative (N) and Positive (P), and triangular for the Zero (C) linguistics terms. The interval used for this fuzzy controller is [-50 50]. Figure 7 shows the input variables.



(a) (b)

Figure 7. (a) Linear velocity error (e_v). (b) Angular velocity error (e_w).

The rule set of the FLC contain 9 rules, which govern the input-output relationship of the FLC and this adopts the Takagi-Sugeno style inference engine [14], and we use a single point in the outputs (constant values), obtained using weighted average defuzzification procedure. To find the best

fuzzy controller, we used a genetic algorithm to find the parameters of the membership functions. In figure 8 we show the chromosome with 28 bits (positions).

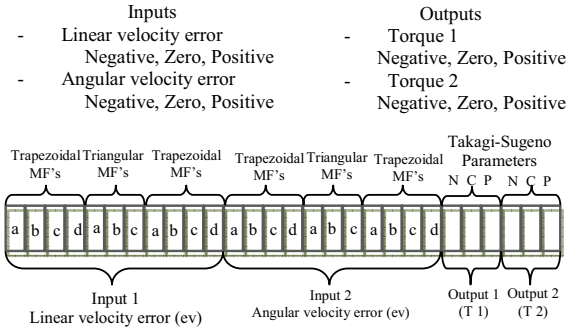


Figure 8. Chromosome representation for the fuzzy logic controller.

Table I shows the parameters of the membership functions, the minimal and the maximum values in the search range for the genetic algorithm to find the best fuzzy controller system.

TABLE I. PARAMETERS OF THE MEMBERSHIP FUNCTIONS

MF Type	Point	Minimum Value	Maximum Value
Trapezoidal	a	-50	-50
	b	-50	-50
	c	-15	-5.05
	d	-1.5	-0.5
Triangular	a	-5	-1.75
	b	0	0
	c	1.75	5
Trapezoidal	a	0.5	1.5
	b	5.05	15
	c	50	50
	d	50	50

IV. SIMULATION RESULTS

In this section, we evaluate, through computer simulation performed in MATLAB® and SIMULINK®, the ability of the proposed controller to stabilize the unicycle mobile robot, defined by (3) and (4) where the matrix values

$$M(q)=\begin{bmatrix} 0.3749 & -0.0202 \\ -0.0202 & 0.3739 \end{bmatrix},\ C(q,\dot{q})=\begin{bmatrix} 0 & 0.1350\vartheta \\ -0.1350\vartheta & 0 \end{bmatrix},\ D=\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

where taken from [11].

The desired trajectory is the following one:

$$\vartheta_d(t)=\begin{cases} v_d(t)=0.2(1-\exp(-t)) \\ w_d(t)=0.4\sin(0.5t) \end{cases} \qquad (20)$$

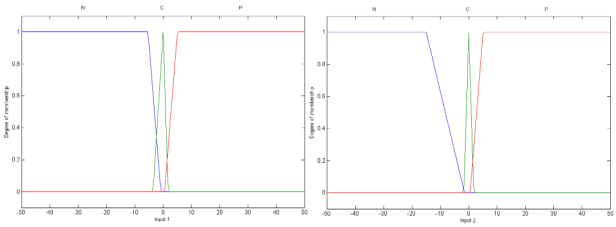
and was chosen in terms of its corresponding desired linear v_d and angular velocities w_d, subject to the initial conditions

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

The gains $\gamma_i, i=1,2,3$ of the kinematic model (14) were tuned by using genetic algorithm approach resulting in $\gamma_1=5$, $\gamma_2=24$ and $\gamma_3=3$, the best gains that were found.

A. *Genetic algorithm results for the optimization of the Type-1 fuzzy logic controller (FLC).*

Figure 9 shows the parameters optimized by the genetic algorithm for the input variables (e_v, e_w).



(a) (b)

Figure 9. (a) Linear velocity error, (b) Angular velocity error.

Table II contains the results of the FLC, obtained by varying the values of generation number, percentage of replacement, mutation and crossover.

TABLE II. GA RESULTS FOR TYPE-1 FLC OPTIMIZATION

No.	Ind	Gen	% Repl	Cross	Mut	Selection Method	Average error	G.A. time
1	50	30	0.7	0.7	0.2	Roulette	0.4122618	7:18
2	100	25	0.7	0.8	0.3	Roulette	0.4212924	12:11
3	20	15	0.7	0.8	0.2	Roulette	0.5524043	1:26
4	10	20	0.7	0.8	0.2	Roulette	0.4899811	1:21
5	80	25	0.7	0.7	0.4	Roulette	0.4126189	9:57
6	150	50	0.7	0.6	0.3	Roulette	0.4094381	43:15
7	90	60	0.7	0.9	0.4	Roulette	0.4087614	44:43
8	10	25	0.7	0.8	0.2	Roulette	0.5703853	2:09
9	65	40	0.7	0.8	0.2	Roulette	0.4099531	22:52
10	30	25	0.7	0.9	0.5	Roulette	0.4086178	6:21
11	70	50	0.7	0.8	0.3	Roulette	0.4086729	29:17
12	80	50	0.7	0.9	0.3	Roulette	0.4099137	33:32
13	**26**	**30**	**0.9**	**0.6**	**0.3**	**Roulette**	**0.4082359**	**6:40**

In Figure 10 we can observe the stability of the position errors and the trajectory tracking of the autonomous mobile robot.

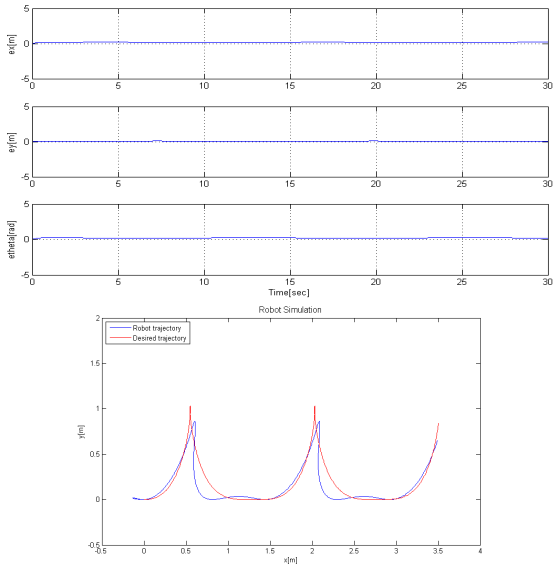
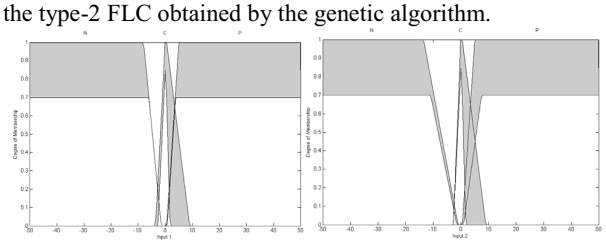


Figure 10. Stabilization of the position errors and trajectory tracking of the autonomous mobile robot with Type-1 FLC.

B. Genetic algorithms results for the optimization of the Type-2 fuzzy logic controller (FLC).

Figure 11 shows the membership functions of the inputs for the type-2 FLC obtained by the genetic algorithm.



(a) (b)

Fig 11. (a) Linear velocity error, (b) Angular velocity error.

Table III contains the results of the Type-2 FLC, obtained by varying the values of generation number, percentage of replacement, mutation and crossover.

TABLE III. GA RESULTS FOR TYPE-2 FLC OPTIMIZATION

No.	Ind	Gen	% Repl	Cross	Mut	Selection Method	Average error	G.A. time
1	50	20	0.7	0.8	0.4	Roulette	0.3993130	4:52:08
2	20	15	0.7	0.8	0.5	Roulette	0.4008340	1:13:03
3	23	20	0.7	0.8	0.4	Roulette	0.3994720	02:56:23
4	40	25	0.7	0.8	0.5	Roulette	0.3993860	6:37:16
5	30	19	0.7	0.9	0.5	Roulette	0.3994950	3:02:35
6	35	10	0.7	0.8	0.5	Roulette	0.4111980	1:15:03
7	45	25	0.7	0.9	0.5	Roulette	0.4008810	7:22:52
8	38	18	0.7	0.7	0.3	Roulette	0.3991930	3:40:29
9	60	20	0.7	0.8	0.6	Roulette	0.3989860	6:40:59
10	45	20	0.7	0.8	0.6	Roulette	0.4007900	5:56:20
11	45	15	0.7	0.7	0.5	Roulette	0.4068480	3:22:18
12	58	25	0.9	0.6	0.4	Roulette	0.3995240	7:49:24
13	**42**	**35**	**0.9**	**0.8**	**0.6**	**Roulette**	**0.3989410**	**7:11:52**

In Figure 12 we can observe the stability of the position errors and the trajectory tracking of the autonomous mobile robot.

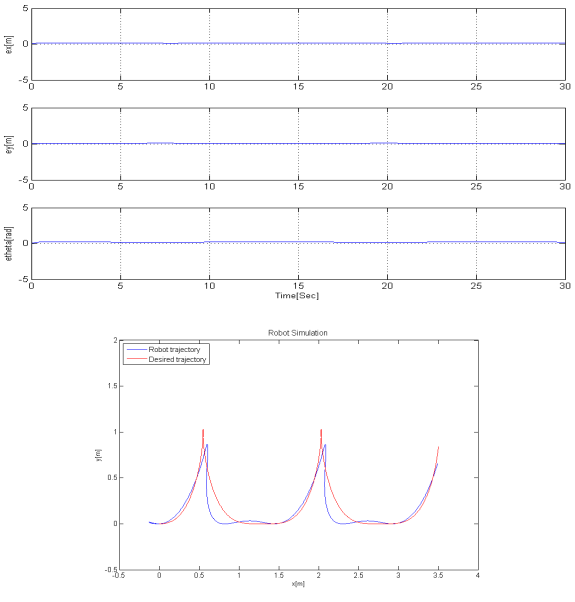


Figure 12. Stabilization of the position errors and trajectory tracking of the autonomous mobile robot with Type-2 FLC.

V. CONCLUSIONS

We have designed a trajectory tracking controller taking into account the kinematics and the dynamics of the autonomous mobile robot using type-2 fuzzy logic and genetic algorithms. Genetic algorithms are used for the optimization of the gains for the trajectory tracking and also for the optimization of the parameters of membership functions for fuzzy logic control. Until now we have good results in the control, because one of the objectives was the robot stability on a desired trajectory. The next step is to consider another optimization technique (for example PSO) for optimizing the type-2 fuzzy logic controller of the autonomous mobile robot to make the comparison between these optimization techniques.

REFERENCES

[1] A. E. Eiben, J. E. Smith, Natural Computing Series. Introduction to Evolutionary Computing, pp 2-66, Springer 2003.

[2] K. F. Man, K. S. Tang and S. Kwong., "Genetic Algorithms, Concepts and Designs", pp 5-10, Springer, 2000.

[3] R. C. Eberhart, and J. Kennedy, A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43, 1995.J.-G. Lu, "Title of paper with only the first word capitalized," *J. Name Stand. Abbrev.*, in press.

[4] D. B. Fogel, "An introduction to simulated evolutionary optimization", IEEE transactions on neural networks, vol 5, no. 1, pp. 3 – 14, jan 1994.

[5] P. J. Angeline, Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences, Evolutionary Programming VII, Lecture Notes in Computer Science 1447, Springer, :601-610. 1998.

[6] P. J. Angeline, Using Selection to Improve Particle Swarm Optimization ,In Proceedings 1998 IEEE World Congress on Computational Intelligence, Anchorage, Alaska, IEEE, 84-89. 1998.

[7] K. Veeramachaneni, L. Osadciw and W. Yan, "Improving Classifier Fusion Using Particle Swarm Optimization", IEEE Fusion Conference, Italy, July, 2006.

[8] J. Kennedy and R. Mendes, "Population structure and particle swarm performance".Proceeding of IEEE conference on Evolutionary Computation, pp. 1671-1676. 2002.

[9] T. Fukao, H. Nakagawa, N. Adachi, Adaptive Tracking Control of a NonHolonomic Mobile Robot, *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 5, pp. 609-615, October 2000.

[10] D. Liberzon, Switching in Systems and Control, Bikhauser, 2003.

[11] R. W. Brockett, *Asymptotic stability and feedback stabilization*. In Millman R.S. and Sussman H.J. (Eds.), Differential Geometric Control Theory, Birkhauser, Boston, pp. 181-191, 1983.

[12] M. Kristic, I. Kanellakopoulos and P. Kokotovic, *Nonlinear and adaptive control design*, Wiley-Interscience, 1995.

[13] B. Paden and R. Panja, "Globally asymptotically stable PD+ controller for robot manipulator," International Journal of Control, vol. 47, no. 6, pp. 1697-1712, 1988.

[14] K. M. Passino, S. Yurkovich, *"Fuzzy Control"*, Addison Wesley Longman, USA 1998.

[15] K. Duc Do, J. Zhong-Ping and J. Pan, "A global output-feedback controller for simultaneous tracking and stabilizations of unicycle-type mobile robots," *IEEE Trans. Automat. Contr.*, vol. 30, pp. 589-594, 2004.

[16] H. Khalil, *Nonlinear systems: 3rd edition.* New York: Prentice Hall, 2000.