Information Sciences 179 (2009) 2158–2174

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms

Ricardo Martínez [a], Oscar Castillo [a,*], Luis T. Aguilar [b,1]

[a] Division of Graduate Studies and Research, Tijuana Institute of Technology, Tijuana, Mexico
[b] Instituto Politécnico Nacional, Centro de Investigación y Desarrollo de Tecnología Digital, PMB 88, P.O. Box 439016, San Ysidro, CA 92143-9016, Mexico

ARTICLE INFO

Article history:
Received 7 November 2007
Received in revised form 20 December 2008
Accepted 30 December 2008

Keywords:
Interval type-2 fuzzy logic
Genetic algorithms
Mobile robots
Optimization
Intelligent control

ABSTRACT

We describe a tracking controller for the dynamic model of a unicycle mobile robot by integrating a kinematic and a torque controller based on type-2 fuzzy logic theory and genetic algorithms. Computer simulations are presented confirming the performance of the tracking controller and its application to different navigation problems.

1. Introduction

Mobile robots have attracted considerable interest in the robotics and control research community, because they have non-holonomic properties caused by non-integrable differential constrains [16,19]. The motion of non-holonomic mechanical systems [15] is constrained by its own kinematics, so the control laws are not derivable in a straightforward manner (Brockett's condition [6]).

Furthermore, most reported designs rely on intelligent control approaches such as fuzzy logic control (FLC) [4,25,34,38,39,45,46] and neural networks [15,43]. However the majority of the publications mentioned above, have concentrated on kinematics models of mobile robots, which are controlled by the velocity input, while less attention has been paid to the control problems of non-holonomic dynamic systems, where forces and torques are the true inputs: Bloch and Drakunov [5,13] and Chwa [10], used a sliding mode control to the tracking control problem.

This paper is organized as follows: Section 2.1 presents an introductory explanation of type-2 fuzzy logic, Section 2.2 presents the basics of genetic fuzzy systems, Section 2.3 presents the problem statement and the kinematic and dynamic models of the unicycle mobile robot. Section 3 introduces the posture and velocity control design where a genetic algorithm is used to select the parameters of the posture controller. Robustness properties of the closed-loop system are achieved with a type-2 fuzzy logic velocity control system using a Takagi–Sugeno model where the wheel input torques, linear velocity, and angular velocity will be considered as linguistic variables. Section 4 provides a simulation study of the unicycle mobile robot using the controller described in Section 3. Finally, Section 5 presents the conclusions.

* Corresponding author.
E-mail addresses: mc.ricardo.martinez@hotmail.com (R. Martínez), ocastillo@hafsamx.org (O. Castillo), luis.aguilar@ieee.org (L.T. Aguilar).
[1] Fax: +52 664 6231388.

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.12.028

2. Theoretical basis and problem statement

This section describes the theoretical basis of the paper as well as the problem definition. Some basics about type-2 fuzzy systems and genetic fuzzy systems are first presented [8].

2.1. Type-2 fuzzy logic systems

If we have a type-1 membership function, as in Fig. 1a, and we are blurring it to the left and to the right as illustrated in Fig. 1b, 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 [31,33]. A type-2 fuzzy set $\widetilde{A}$, is characterized by the membership function:

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

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

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

where denote union over all admissible input variables x and u. For discrete universes of discourse $\int$ is replaced by $\sum$ [32]. 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 [27].

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 Fig. 2. 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 $\bar{\mu}_{\widetilde{A}}(x)$ and a lower membership function $\underline{\mu}_{\widetilde{A}}(x)$.

A FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain [30]. 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 [35].

It is known that type-2 fuzzy sets enable modeling and minimizing the effects of uncertainties in rule-based FLS [22]. Unfortunately, type-2 fuzzy sets are more difficult to use and understand than type-1 fuzzy sets; hence, their use is not widespread yet [41]. As a justification for the use of type-2 fuzzy sets, in [32] 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 histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not all agree.

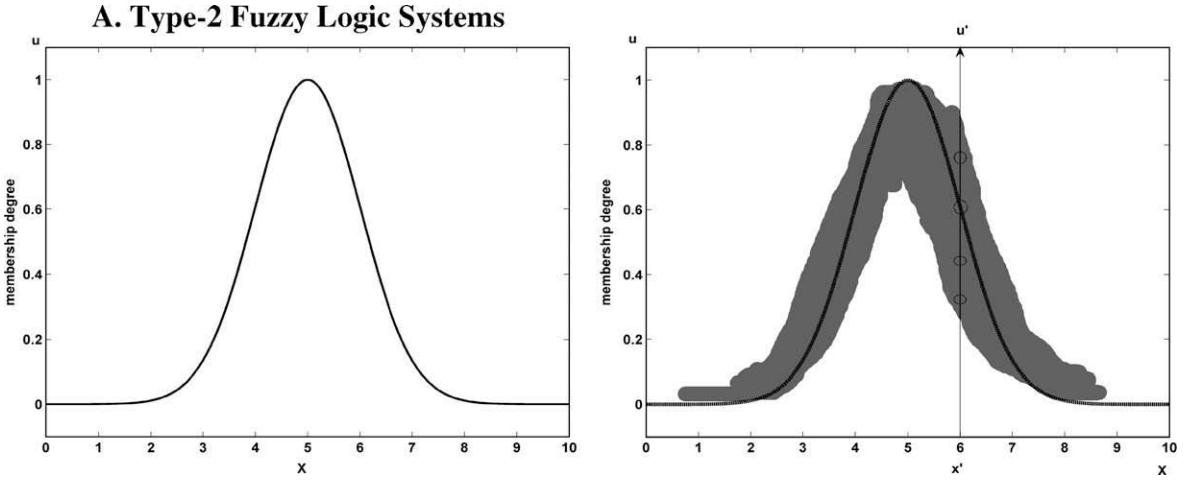

Fig. 1. (a) Type-1 membership function and (b) blurred type-1 membership function.

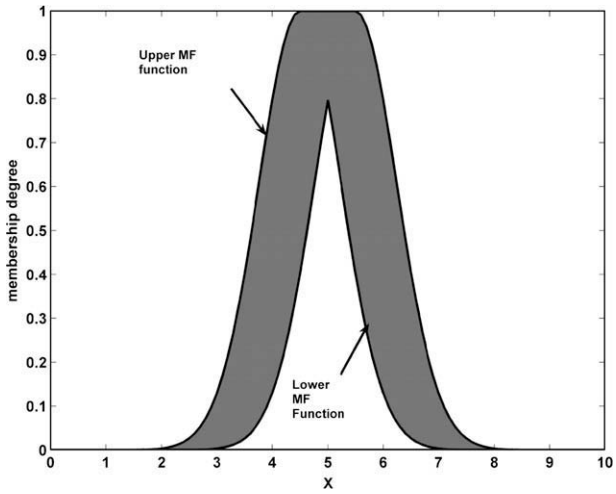

Fig. 2. Interval type-2 membership function.

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. Type-2 FLSs, can be used when the circumstances are too uncertain to determine exact membership grades such as when the training data is corrupted by noise. 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 Fig. 3. The output processor includes a type-reducer and defuzzifier; it generates a type-1 fuzzy set output (from the type-reducer) or a crisp number (from the defuzzifier) [36,37]. Next we will explain each of the blocks in Fig. 3.

2.1.1. Fuzzifier

The fuzzifier maps a crisp point $\mathbf{x} = (x_1,\ldots,x_p)^T \in X_1 \times X_2 \times \cdots \times X_p \equiv \mathbf{X}$ into a type-2 fuzzy set $\widetilde{A}_x$ in $\mathbf{X}$ [14], which is an interval type-2 fuzzy sets in this case. We will use a type-2 singleton fuzzifier, in singleton fuzzification, the input fuzzy set has only a single point with nonzero membership [31,34]. $\widetilde{A}_x$ is a type-2 fuzzy singleton if $\mu_{\widetilde{A}_x}(x) = 1/1$ for $\mathbf{x} = \mathbf{x}'$ and $\mu_{\widetilde{A}_x}(x) = 1/0$ for all other $\mathbf{x} \neq \mathbf{x}'$ [35].

2.1.2. Rules

The structure of rules in a type-1 FLS and a type-2 FLS is the same, but in the latter the antecedents and the consequents will be represented by type-2 fuzzy sets. So for a type-2 FLS with p inputs $x_1 \in X_1,\ldots,x_p \in X_p$ and one output $y \in Y$, which is a

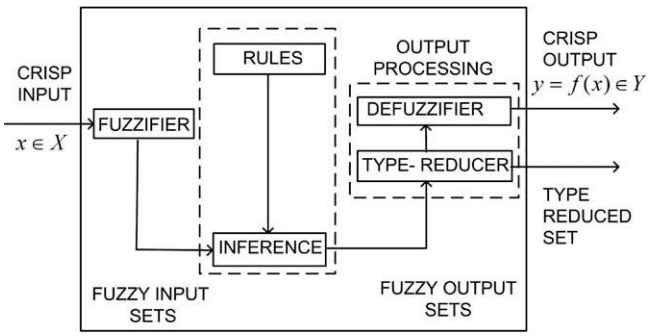

Fig. 3. Type-2 fuzzy logic system.

multiple input single output (MISO) system, if we assume there are M rules, the lth rule in the type-2 FLS can be written as follows [35]:

$$R^l: \text{ IF } x_1 \text{ is } \widetilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \widetilde{F}_p^l, \text{ THEN } y \text{ is } \widetilde{G}^l \quad l = 1, \ldots, M \tag{3}$$

2.1.3. Inference

In the type-2 FLS, the inference engine combines rules and provides a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. It is necessary to compute the join $\sqcup$ (unions), and the meet $\sqcap$ (intersections), as well as extended sup-star compositions (sup-star compositions) of type-2 relations [35]. If $\widetilde{F}_1^l \times \cdots \times \widetilde{F}_p^l = \widetilde{A}^l$, Eq. (3) can be re-written as

$$R^l: \widetilde{F}_1^l \times \cdots \times \widetilde{F}_p^l \to \widetilde{G}^l = \widetilde{A}^l \to \widetilde{G}^l \quad l = 1, \ldots, M \tag{4}$$

R^l is described by the membership function $\mu_{R^l}(\mathbf{x}, y) = \mu_{R^l}(x_1, \ldots, x_p, y)$, where

$$\mu_{R^l}(\mathbf{x}, y) = \mu_{\widetilde{A}^l \to \widetilde{G}^l}(\mathbf{x}, y) \tag{5}$$

can be written as [35]:

$$\mu_{R^l}(\mathbf{x}, y) = \mu_{\widetilde{A}^l \to \widetilde{G}^l}(\mathbf{x}, y) = \mu_{\widetilde{F}_1^l}(x_1) \sqcap \cdots \sqcap \mu_{\widetilde{F}_p^l}(x_p) \sqcap \mu_{\widetilde{G}^l}(y) = \left[\sqcap_{i=1}^p \mu_{\widetilde{F}_i^l}(x_i)\right] \sqcap \mu_{\widetilde{G}^l}(y) \tag{6}$$

In general, the p-dimensional input to R^l is given by the type-2 fuzzy set $\widetilde{A}_x$ whose membership function is

$$\mu_{\widetilde{A}_x}(\mathbf{x}) = \mu_{\widetilde{x}_1}(x_1) \sqcap \cdots \sqcap \mu_{\widetilde{x}p}(x_p) = \sqcap_{i=1}^p \mu_{\widetilde{x}i}(x_i) \tag{7}$$

where $\widetilde{X}_i (i = 1, \ldots, p)$ are the labels of the fuzzy sets describing the inputs. Each rule R^l determines a type-2 fuzzy set $\widetilde{B}^l = \widetilde{A}_x \circ R^l$ such that [35]:

$$\mu_{\widetilde{B}^l}(y) = \mu_{\widetilde{A}_x \circ R^l} = \sqcup_{x \in \mathbf{X}} [\mu_{\widetilde{A}_x}(\mathbf{x}) \sqcap \mu_{R^l}(\mathbf{x}, y)] \quad y \in Y \; l = 1, \ldots, M \tag{8}$$

This equation is the input/output relation in Fig. 3 between the type-2 fuzzy set that activates one rule in the inference engine and the type-2 fuzzy set at the output of that engine [35].

In the FLS we used interval type-2 fuzzy sets and meet under product t-norm, so the result of the input and antecedent operations, which are contained in the firing set $\sqcap_{i=1}^p \mu_{\widetilde{F}_{i_i}}(x_i' \equiv F^l(\mathbf{x}'))$, is an interval type-1 set [35],

$$F^l(\mathbf{x}') = [\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')] \equiv [\underline{f}^l, \bar{f}^l] \tag{9}$$

where

$$\underline{f}^l(\mathbf{x}') = \underline{\mu}_{\widetilde{F}_1^l}(x_1')^* \cdots {}^* \underline{\mu}_{\widetilde{F}_p^l}(x_p') \tag{10}$$

and

$$\bar{f}^l(\mathbf{x}') = \bar{\mu}_{\widetilde{F}_1^l}(x_1')^* \cdots {}^* \bar{\mu}_{\widetilde{F}_p^l}(x_p') \tag{11}$$

where $*$ is the product operation.

2.1.4. Type-reducer

The type-reducer generates a type-1 fuzzy set output which is then converted into a crisp output through the defuzzifier. This type-1 fuzzy set is also an interval set, for the case of our FLS we used center of sets (cos) type reduction, $Y_{\cos}$ which is expressed as [35]

$$Y_{\cos}(\mathbf{x}) = [y_l, y_r] = \int_{y^1 \in [y_l^1, y_r^1]} \cdots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \cdots \int_{f^M \in [\underline{f}^M, \bar{f}^M]} 1 \Big/ \frac{\sum_{i=1}^M f^i y^i}{\sum_{i=1}^M f^i} \tag{12}$$

this interval set is determined by its two end points, y_l and y_r, which corresponds to the centroid of the type-2 interval consequent set $\widetilde{G}^i$ [35],

$$C_{\widetilde{G}^i} = \int_{\theta_1 \in J_{y1}} \cdots \int_{\theta_N \in J_{yN}} 1 \Big/ \frac{\sum_{i=1}^N y_i \theta_i}{\sum_{i=1}^N \theta_i} = [y_l^i, y_r^i] \tag{13}$$

before the computation of $Y_{\cos}(\mathbf{x})$, we must evaluate Eq. (13), and its two end points, y_l and y_r. If the values of f_i and y_i that are associated with y_l are denoted f_l^i and y_l^i, respectively, and the values of f_i and y_i that are associated with y_r are denoted f_r^i and y_r^i, respectively, from Eq. (12), we have [35]

$$y_l = \frac{\sum_{i=1}^{M} f_l^i y_l^i}{\sum_{i=1}^{M} f_l^i} \tag{14}$$

$$y_r = \frac{\sum_{i=1}^{M} f_r^i y_r^i}{\sum_{i=1}^{M} f_r^i} \tag{15}$$

2.1.5. Defuzzifier

From the type-reducer we obtain an interval set $Y_{\cos}$, and then to defuzzify it we use the average of y_l and y_r, so the defuzzified output of an interval singleton type-2 FLS is [35]

$$y(\mathbf{x}) = \frac{y_l + y_r}{2} \tag{16}$$

2.2. Genetic fuzzy systems

Fuzzy systems have been successfully applied to problems in classification [47], modeling [40] control [33], and in a considerable number of applications [9]. 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 neural and evolutionary; provide 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) [11].

GAs are search algorithms, based on natural evolution, that provide robust search capabilities in complex spaces, and thereby offer a valid approach to problems requiring efficient and effective search processes [42–44]. Genetic learning processes cover different levels of complexity according to the structural changes produced by the algorithm [12], from the simplest case of parameter optimization to the highest level of complexity of learning the rule set of a rule-based system [17,18]. 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 [20,29].

Analysis of the literature shows that the most prominent types of GFSs are genetic fuzzy rule-based systems (GFRBSs) [11], 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.

It is important to distinguish between tuning (alternatively, adaptation) and learning problems:

- Tuning is concerned with optimization of an existing FRBS, whereas learning constitutes an automated design method for fuzzy rule sets that starts from scratch. Tuning processes assume a predefined RB and have the objective to find a set of optimal parameters for the membership and or the scaling functions, DB parameters.
- Learning processes perform a more elaborated search in the space of possible RBs or whole KBs and do not depend on a predefined set of rules.

These are different kind of genetic fuzzy systems applications, but we focus in this paper on genetic tuning for optimization parameters of membership functions for type-2 fuzzy systems [2,1,7].

2.3. The mobile robot

The model considered in this paper is that of a unicycle mobile robot (Fig. 4), it consists of two driving wheels mounted of the same axis and a front free wheel [3].

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 [26]:

$$M(q)\dot{v} + C(q,\dot{q})v + Dv = \tau + P(t) \tag{17}$$

$$\dot{q} = \underbrace{\begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_{v} \tag{18}$$

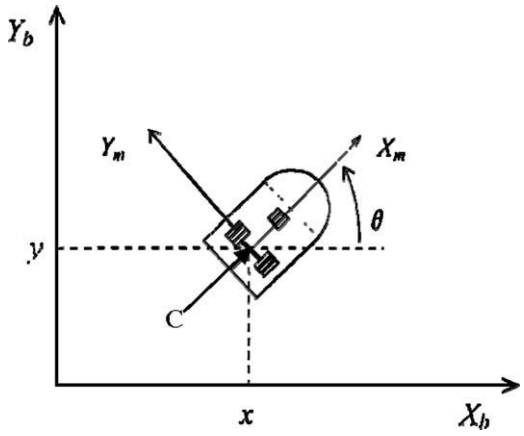

Fig. 4. Wheeled mobile robot.

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 (Fig. 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. Eq. (18) 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 (17), (18) has the following non-holonomic constraint:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{19}$$

which corresponds to a no-slip wheel condition preventing the robot from moving sideways [28]. The system (18) fails to meet Brockett's necessary condition for feedback stabilization [6], which implies that no 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

$$\lim_{t\to\infty} \|q_d(t) - q(t)\| = 0, \tag{20}$$

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

3. 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 (17), (18) can be designed using the backstepping approach [24] since the kinematics subsystem (18) is controlled indirectly through the velocity vector v. The procedure to design the overall controller consists of two steps:

(1) Design a virtual velocity vector $\vartheta_r = \vartheta$ such that the kinematic model (18) be uniformly asymptotically stable.
(2) Design a velocity controller τ by using FLC that ensures

$$\|\vartheta_r(t) - \vartheta(t)\| = 0, \quad \forall t \geqslant t_s \tag{21}$$

where t_s is the reachability time.

In (21), it is considered that real mobile robots have actuated wheels, so the control input is τ that must be designed to stabilize the dynamics (17), without destabilizing the system (18), by forcing $\vartheta \in R^2$ to reach the virtual velocity vector $\vartheta_r \in R^2$ in finite-time. Roughly speaking, if (21) 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. Fig. 5 illustrates the feedback connection which involves the fuzzy controller.

3.1. Posture control design

First, we focus on the kinematic model by designing a virtual control (ϑ_r) such that the control objective (20) is achieved. To this end, let us consider the reference trajectory $q_d(t)$ as a solution of the following differential equation:

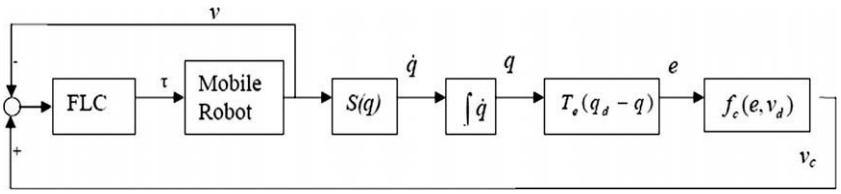

Fig. 5. Tracking control structure.

$$\dot{q}_d = \begin{pmatrix} \cos\theta_d & 0 \\ \sin\theta_d & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_d \\ w_d \end{pmatrix} \tag{22}$$

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

$$\begin{pmatrix} \tilde{q}_1 \\ \tilde{q}_2 \\ \tilde{q}_3 \end{pmatrix} = \underbrace{\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{T_e(\theta)} \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{pmatrix} \tag{23}$$

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

$$\begin{pmatrix} \dot{\tilde{q}}_1 \\ \dot{\tilde{q}}_2 \\ \dot{\tilde{q}}_3 \end{pmatrix} = \begin{pmatrix} w\tilde{q}_2 - v + v_d\cos\tilde{q}_3 \\ -w\tilde{q}_1 + v_d\sin\tilde{q}_3 \\ w_d - w \end{pmatrix} \tag{24}$$

which is in terms of the corresponding real and desired velocities, is then obtained by differentiating (23) with respect to time.

In order to present the main result of this subsection, we need first to recall the following theorems [6].

Theorem 1 (Uniform stability [23]). *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, ∞] $\times D \to R$ be a continuously differentiable function such that*

$$W_1(x) \leqslant V(x,t) \leqslant W2(x) \tag{25}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t) \leqslant 0 \tag{26}$$

for all $t \geqslant 0$ 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 (Uniform asymptotic stability [23]). *Suppose the assumptions of Theorem 1 are satisfied with inequality (26) strengthened to*

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t) \leqslant -W_3(x) \tag{27}$$

for all $t \geqslant 0$ 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 (24) be driven by the control law (virtual velocities)*

$$\begin{aligned} v_r &= v_d\cos\tilde{q}_3 + \gamma_1\tilde{q}_1 \\ w_r &= w_d + \gamma_2 v_d\tilde{q}_2 + \gamma_3\sin\tilde{q}_3 \end{aligned} \tag{28}$$

where γ_1, γ_2, and γ_3 are positive constants. If $v = v_r$ and $w = w_r$ for all $t \geqslant 0$ in (18), then the origin of the closed-loop system (24)–(28) is uniformly asymptotically stable.

Proof. Under the control (12), the closed-loop system takes the form:

$$\begin{pmatrix} \dot{\tilde{q}}_1 \\ \dot{\tilde{q}}_2 \\ \dot{\tilde{q}}_3 \end{pmatrix} = \begin{pmatrix} w_d\tilde{q}_2 + \gamma_2 v_d\tilde{q}_2^2 + \gamma_3\tilde{q}_3 - \gamma_1\tilde{q}_1 \\ -w_d\tilde{q}_1 - \gamma_2 v_d\tilde{q}_1\tilde{q}_2 - \gamma_3\tilde{q}_1\sin\tilde{q}_3 + v_d\sin\tilde{q}_3 \\ -\gamma_2 v_d\tilde{q}_2 - \gamma_3\sin\tilde{q}_3 \end{pmatrix} \tag{29}$$

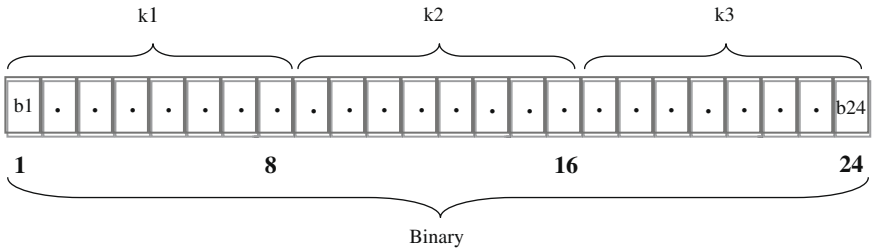

Fig. 6. Chromosome representation.

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.,$\tilde{q}_3 = 0, \pi, \ldots, n\pi$). Genetic algorithms are applied for tuning the kinematic control gains γ_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

$$V(\tilde{q}) = \frac{1}{2}\tilde{q}_1^2 + \frac{1}{2}\tilde{q}_2^2 + \frac{1}{\gamma_2}(1 - \cos\tilde{q}_3) \tag{30}$$

which is positive definite. Taking the time derivative of $V(\tilde{q})$ along the solution of the closed-loop system (29), we get

$$\dot{V}(\tilde{q}) = \tilde{q}_1\dot{\tilde{q}}_1 + \dot{q}_2\dot{\tilde{q}}_2 + \frac{1}{\gamma_2}\dot{\tilde{q}}_3 \sin\tilde{q}_3 = -\gamma_1\tilde{q}_1^2 - \frac{\gamma_3}{\gamma_2}(\sin\tilde{q}_3)^2 \leqslant 0 \tag{31}$$

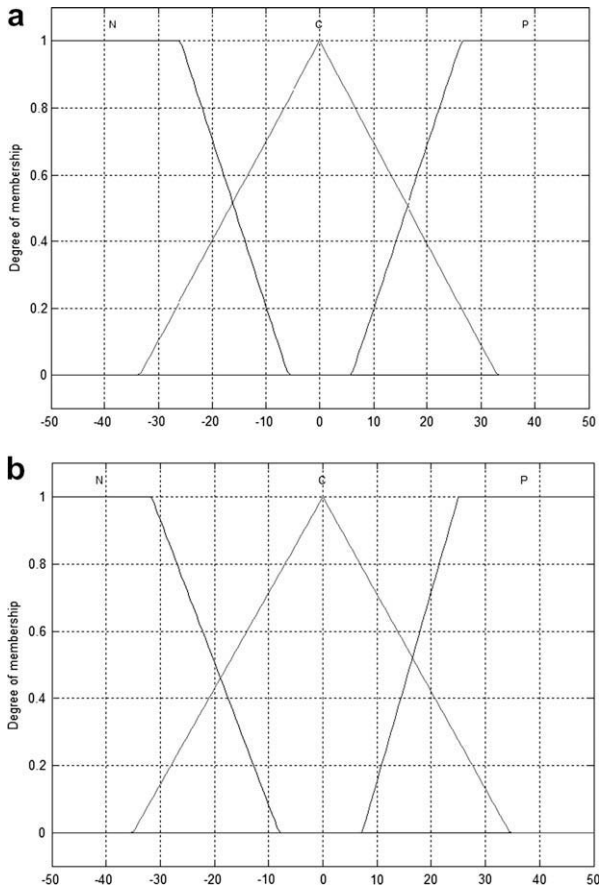

Fig. 7. (a) Linear velocity error (e_v) and (b) angular velocity error (e_w).

Table 1
Fuzzy rule 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

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

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

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

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 [37], 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. Fig. 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. Changing the crossover rate and the number of crossover points used did not affect the results. Also, changing the mutation rate did not affect the optimal results. The advantage of using the genetic algorithm to find the gains, is that time-consuming manual search of these parameters was avoided.

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

3.2. Velocity control design

In this subsection a fuzzy logic controller is designed to force the real velocities of the mobile robot (17) and (18) to match those required in Eq. (28) of Theorem 3 to satisfy the control objective (20).

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

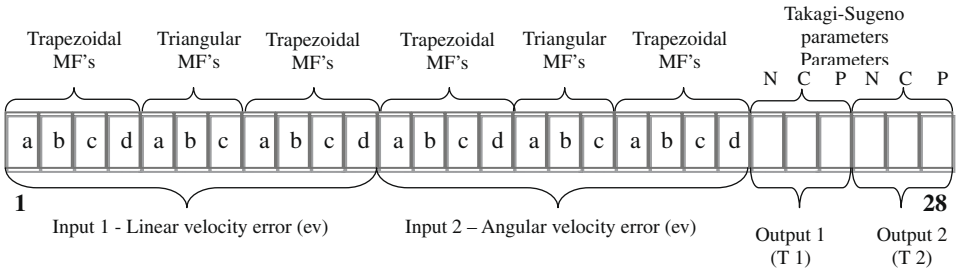

Fig. 8. Chromosome representation for the fuzzy logic controller.

Table 2
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

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]. Fig. 7 shows the input variables.

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 [21], and we use a single point in the outputs (constant values), obtained using weighted average defuzzification procedure. In Table 1, we present the rule set whose format is established as follows:

Rule i : If e_v is G_1 and e_w is G_2 then F is G_3 and N is G_4

where $G_1 \cdots G_4$ are the fuzzy set associated to each variable and $i = 1 \cdots 9$.

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

Inputs

- Linear velocity error Negative, Zero, Positive
- Angular velocity error Negative, Zero, Positive

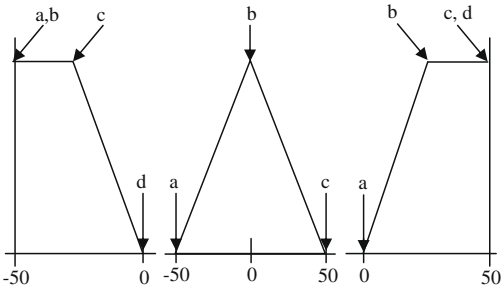

Fig. 9. Type of membership functions.

Table 3
Results of the simulation to find the constants k1, k2 and k3.

No.	Individuals	Generations	Crossover	Mutation	Average error	k1	k2	k3
1	100	70	0.8	0.3	14.1544	39	483	66
2	50	40	0.8	0.4	109.6417	451	416	80
3	20	15	0.7	0.4	41.4291	174	320	59
4	40	30	0.9	0.4	85.8849	354	311	51
5	60	80	0.9	0.4	116.6302	477	287	47
6	60	70	0.8	0.3	70.0033	293	382	59
7	70	100	0.7	0.2	31.4233	138	485	72
8	50	60	0.8	0.2	16.8501	74	402	59
9	40	20	0.5	0.2	101.0294	420	428	68
10	40	20	0.8	0.2	70.8432	299	481	78
11	55	25	0.8	0.3	15.0319	31	509	72
12	50	70	0.5	0.2	122.8667	507	406	65
13	30	50	0.8	0.3	103.7498	428	320	49
14	3	5	0.8	0.1	20.0183	27	507	98
15	2	10	0.8	0.1	31.4037	132	346	138
16	5	15	0.8	0.1	114.8469	477	219	27
17	30	50	0.8	0.1	4.1658	6	9	6
18	30	50	0.8	0.1	4.1844	6	9	5
19	30	50	0.8	0.1	4.0008	4	8	4
20	30	50	0.8	0.1	4.0293	4	5	4
21	**50**	**40**	**0.8**	**0.4**	**3.8138**	**5**	**24**	**3**
22	50	40	0.8	0.4	4.5141	10	16	6
23	50	40	0.8	0.4	6.9690	25	28	5
24	50	40	0.8	0.4	4.1202	6	15	5
25	70	80	0.8	0.3	3.9736	4	10	4
26	70	80	0.8	0.3	4.3726	7	10	4
27	30	20	0.8	0.1	4.0466	5	12	4
28	40	20	0.8	0.1	4.0621	5	11	4
29	70	80	0.8	0.3	4.7051	12	15	5
30	40	70	0.8	0.3	4.4199	8	11	5
21	40	70	0.8	0.3	4.0781	6	18	5
32	40	70	0.8	0.3	4.1669	6	11	5

Outputs

- Torque 1 Constant Negative, Zero, Positive
- Torque 2 Constant Negative, Zero, Positive

Table 2 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 and Fig. 9 shows the type of membership functions that were used.

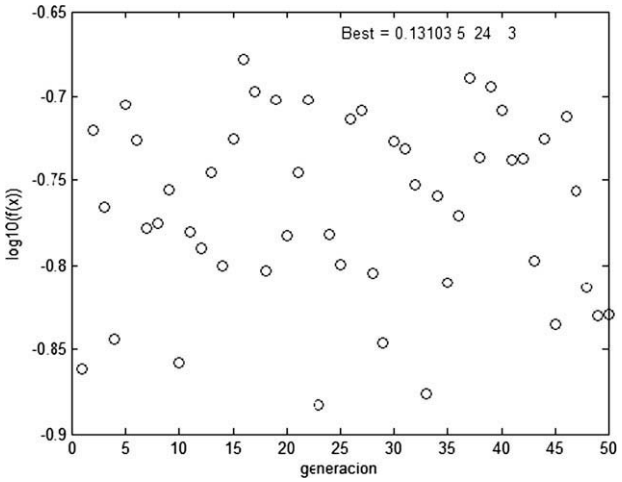

Fig. 10. Evolution of GA finding the optimal gains.

Table 4
Best simulations results.

Average error	k1	k2	k3
3.8138	5	24	3

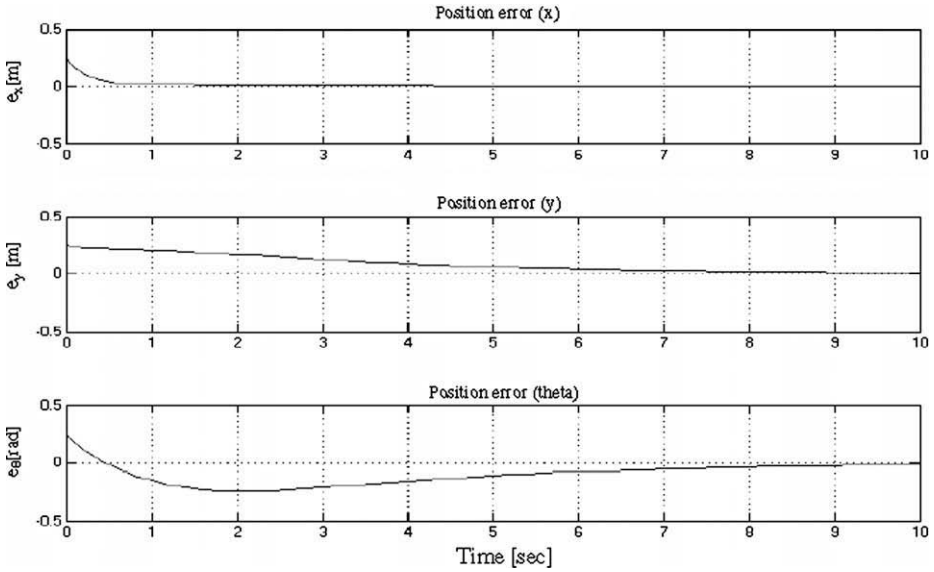

Fig. 11. Tracking errors.

4. Simulation results

In this section, we evaluate, through computer simulations performed in MATLAB® and SIMULINK®, the ability of the proposed controller to stabilize the unicycle mobile robot, defined by (17) and (18) where the matrix values

Table 5
Genetic algorithm results for type-1 FLC optimization.

No.	Individuals	Generations	% Replacement	Crossover	Mutation	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	200	100	0.7	0.4	0.1	Roulette	0.4085207	2:43:28
14	15	10	0.7	0.8	0.5	Roulette	0.5669795	1:14
15	15	25	0.7	0.9	0.2	Roulette	0.4789307	3:03
16	30	40	0.7	0.7	0.2	Roulette	0.4108032	10:28
17	50	60	0.7	0.6	0.4	Roulette	0.4111103	1:05:14
18	20	50	0.7	0.6	0.2	Roulette	0.4339689	9:13
19	80	20	0.7	0.8	0.6	Roulette	0.4490967	13:13
20	100	80	0.7	0.8	0.3	Roulette	0.4083982	6:16
21	30	60	0.7	0.8	0.5	Roulette	0.4943807	14:43
22	25	40	0.7	0.8	0.6	Roulette	0.4247892	8:10
23	70	60	0.7	0.8	0.4	Roulette	0.4084446	34:44
24	35	40	0.7	0.7	0.3	Roulette	0.4099876	11:30
25	45	50	0.7	0.7	0.3	Roulette	0.4128472	18:19
26	60	40	0.9	0.8	0.5	Roulette	0.4106830	20:38
27	80	50	0.9	0.4	0.1	Roulette	0.4095522	33:37
28	40	30	0.9	0.9	0.4	Roulette	0.4102437	10:07
29	100	35	0.9	0.7	0.4	Roulette	0.4094340	29:01
30	80	45	0.9	0.7	0.2	Roulette	0.4100034	29:50
31	5	20	0.9	0.9	0.2	Roulette	0.4377697	0:49
32	**26**	**30**	**0.9**	**0.6**	**0.3**	**Roulette**	**0.4082359**	**6:40**
33	15	15	0.9	0.7	0.4	Roulette	0.4769542	2:01
34	15	40	0.9	0.8	0.2	Roulette	0.4107797	5:12
35	5	30	0.9	0.9	0.3	Roulette	0.5089666	1:19
36	10	15	0.9	0.8	0.1	Roulette	0.5206810	1:55
37	20	25	0.9	0.6	0.2	Roulette	0.4316168	4:49

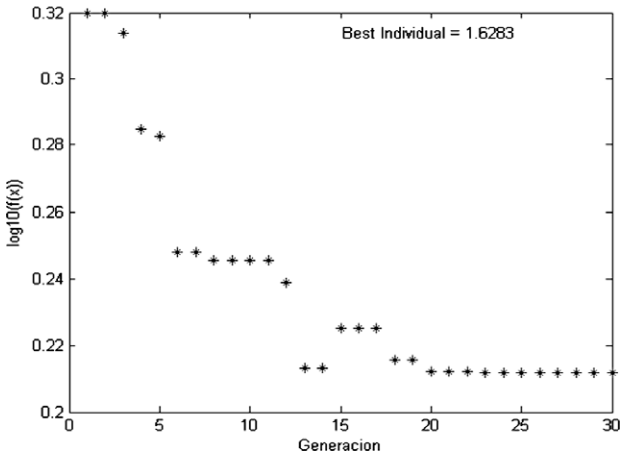

Fig. 12. Evolution of the GA for type-1 FLC optimization.

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

where taken from [14].

The desired trajectory is the following one:

$$\vartheta_d(t) = \begin{cases} v_d(t) = 0.2(1 - \exp(-t)) \\ w_d(t) = 0.4\sin(0.5t) \end{cases} \tag{32}$$

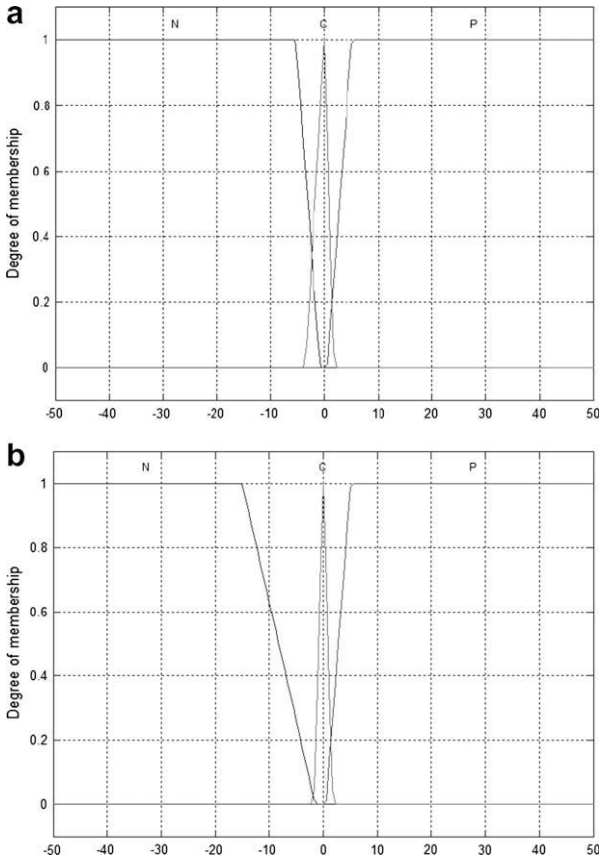

Fig. 13. (a) Linear velocity error and (b) angular velocity error.

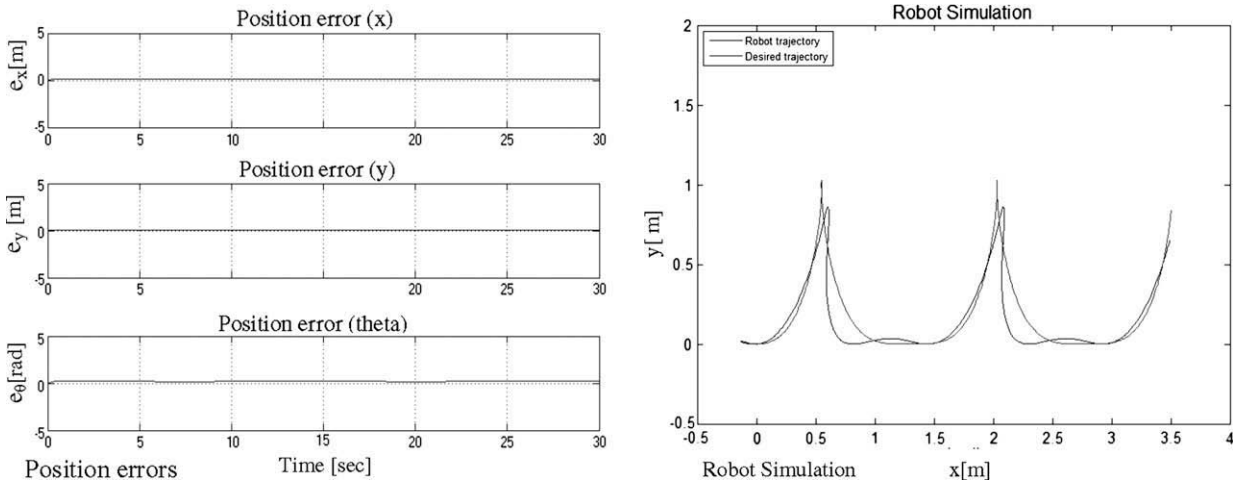

Fig. 14. Stabilization of the autonomous mobile robot with type-1 FLC.

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

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

The gains γ_i, $i = 1, 2, 3$ of the kinematic model (28) were tuned by using a genetic algorithm approach resulting in $\gamma_1 = 5$, $\gamma_2 = 24$ and $\gamma_3 = 3$, the best gains that were found.

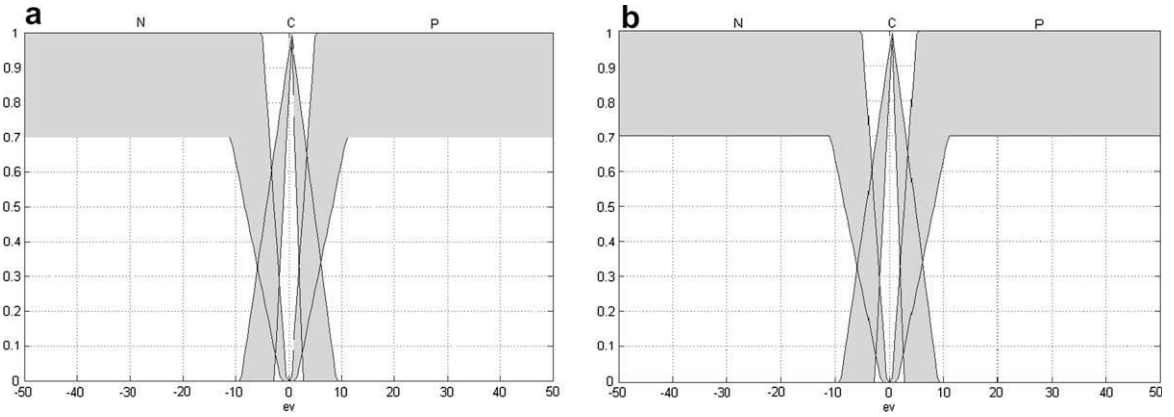

Fig. 15. (a) Linear velocity error and (b) angular velocity error.

Table 6
Genetic algorithm results for type-2 FLC optimization.

No.	Individuals	Generations	% Replacement	Crossover	Mutation	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	40	18	0.9	0.9	0.6	Roulette	0.3990670	3:29:21
14	58	45	0.9	0.8	0.6	Roulette	0.3989470	15:20:34
15	26	18	0.9	0.9	0.5	Roulette	0.4021550	3:48:35
16	10	15	0.9	0.8	0.5	Roulette	0.4028900	1:43:28
17	15	15	0.9	0.7	0.4	Roulette	0.4006630	1:45:03
18	25	22	0.9	0.8	0.5	Roulette	0.3995900	2:11:58
19	60	25	0.9	0.9	0.4	Roulette	0.4002830	9:38:31
20	30	15	0.9	0.8	0.4	Roulette	0.4110670	2:23:33
21	20	18	0.9	0.6	0.4	Roulette	0.3989810	1:28:53
22	46	28	0.9	0.6	0.4	Roulette	0.3991000	6:19:01
23	70	25	0.9	0.7	0.5	Roulette	0.3989980	10:36:57
24	54	20	0.9	0.8	0.6	Roulette	0.3992210	6:19:40
25	66	30	0.9	1	0.6	Roulette	0.3989810	12:54:21
26	**42**	**35**	**0.9**	**0.8**	**0.6**	**Roulette**	**0.3989410**	**7:11:52**
27	26	10	0.9	0.6	0.4	Roulette	0.4027990	2:29:51
28	40	20	0.9	0.6	0.4	Roulette	0.3990530	4:24:41
29	50	15	0.9	0.9	0.5	Roulette	0.4005340	3:33:39
30	80	12	0.9	0.9	0.6	Roulette	0.3997710	6:32:30
31	11	15	0.9	0.5	0.3	Roulette	0.4026380	1:55:02
32	28	18	0.9	0.8	0.3	Roulette	0.3997890	2:46:05
33	22	18	0.9	0.7	0.6	Roulette	0.4008280	2:11:57
34	15	12	0.9	0.8	0.9	Roulette	0.4109010	0:36:24
35	30	14	0.9	0.6	0.6	Roulette	0.4006320	2:32:30
36	60	18	0.9	0.7	0.5	Roulette	0.3990440	5:06:51
37	28	17	0.9	0.5	0.5	Roulette	0.3992410	3:43:54

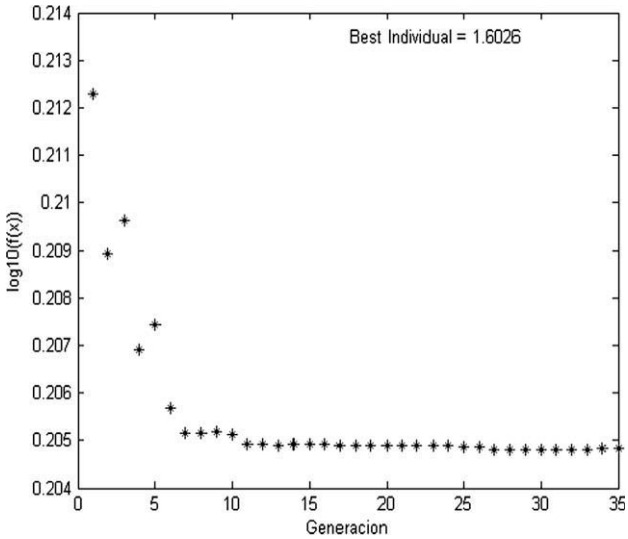

Fig. 16. Evolution of the GA for type-2 FLC optimization.

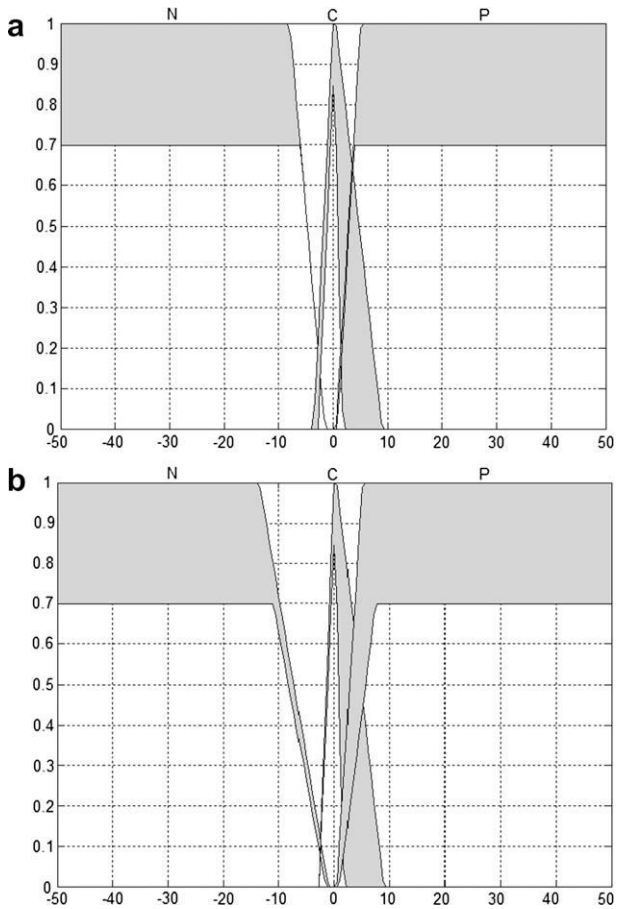

Fig. 17. (a) Linear velocity error and (b) angular velocity error.

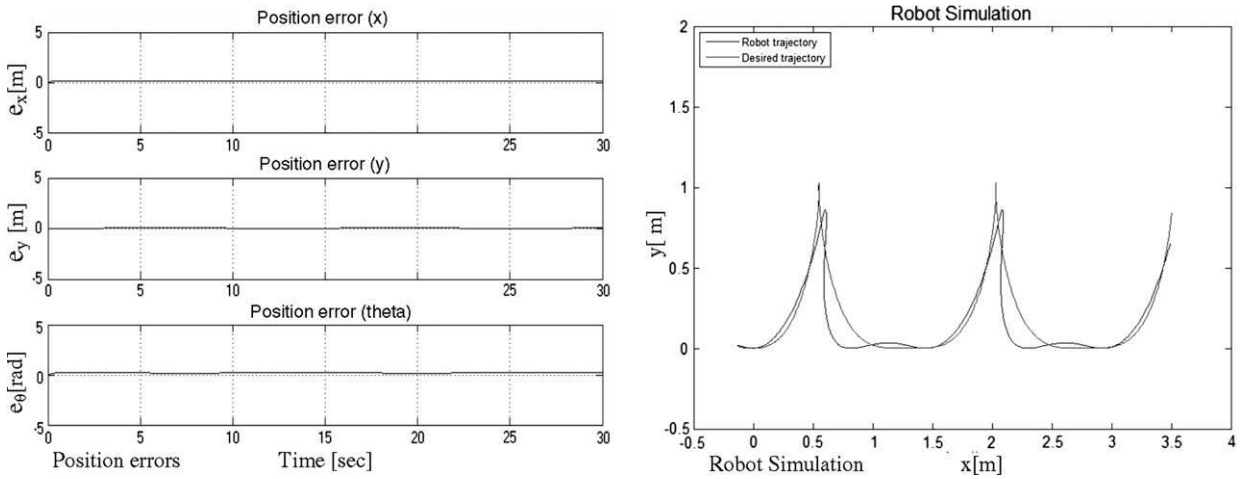

Fig. 18. Stabilization of the autonomous mobile robot with type-2 FLC.

4.1. Genetic algorithm results for the gains k1, k2 and k3

To find these gains we changed the number of generations, the mutation and the crossover operators of the genetic algorithm represented in Table 3. Fig. 10 shows the plot of the evolution of the population on the genetic algorithm finding the best gains.

The best error that was found and the values of the gains k1, k2 and k3 are shown in Table 4.

Fig. 11 shows the plot of the best simulation results for these particular gains.

4.2. Genetic algorithms results for the optimization of the type-1 fuzzy logic controller (FLC)

Table 5 contains the results of the FLC, obtained by varying the values of the generation number, percentage of replacement, mutation and crossover and Fig. 12 shows the evolution of the GA.

Continuing with the results, the best simulation for control of tracking is shown in Fig. 13, which shows the parameters optimized by the genetic algorithm for the input variables (e_v, e_w).

Fig. 14 shows the results of the linear and angular velocity errors, the input torques, the position errors and the obtained trajectory under desired trajectory of the autonomous mobile robot and we can observe the stability on the position error and how the mobile robot follows the trajectory.

4.3. Simulation results of the type-2 FLC

For the type-2 FLC we used a genetic algorithm as for the type-1 FLC and we show in Fig. 15 the membership functions based for the linear velocity error and the angular velocity error.

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

Table 6 contains the results of the type-2 FLC, obtained by varying the values of the generation number, percentage of replacement, mutation and crossover and Fig. 16 shows the evolution of the GA.

Fig. 17 shows the membership functions of the inputs for the type-2 FLC obtained with the genetic algorithm.

Fig. 18 shows the results of the linear and angular velocity errors, the input torques, the position errors and the obtained trajectory under desired trajectory of the autonomous mobile robot and we can observe the stability on the position error and how the mobile robot follows the trajectory.

5. 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 constants for the trajectory tracking and also for the optimization of the parameters of membership functions for fuzzy logic control.

Currently, the design of a type-2 fuzzy logic controller has been tested under a perturbed autonomous wheeled mobile robot, but more tests are in progress.

References

[1] R. Alcalá, J. Alcalá-Fdez, F. Herrera, A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 616–635.
[2] R. Alcalá, M.J. Gacto, F. Herrera, J. Alcalá-Fdez, A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems 15 (5) (2007) 539–557.
[3] L. Astudillo, O. Castillo, L. Aguilar, Intelligent control of an autonomous mobile robot using type-2 fuzzy logic, in: Proceedings of ICAI'06, 2006, pp. 565–570.
[4] S. Bentalba, A. El Hajjaji, A. Rachid, Fuzzy control of a mobile robot: a new approach, in: Proceedings of the IEEE International Conference on Control Applications, Hartford, CT, October 1997, pp. 69–72.
[5] A.M. Bloch, Nonholonomic Mechanics and Control, Springer Verlag, New York, 2003.
[6] R.W. Brockett, Asymptotic stability and feedback stabilization, in: R.S. Millman, H.J. Sussman (Eds.), Differential Geometric Control Theory, Birkhauser, Boston, 1983, pp. 181–191.
[7] J. Casillas, O. Cordon, M.J. del Jesús, F. Herrera, Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction, IEEE Transaction on Fuzzy Systems 13 (1) (2005) 13–29.
[8] O. Castillo, P. Melin, Soft Computing for Control of Non-Linear Dynamical Systems, Springer-Verlag, Heidelberg, Germany, 2001.
[9] Z. Chi, H. Yan, T. Pham, Fuzzy Algorithms: With Applications to Image Processing and Pattern recognition, World Scientific, Singapore, 1996.
[10] D. Chwa, Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates, IEEE Transactions on Control Systems Technology 12 (4) (2004) 633–644.
[11] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (1) (2004) 5–31.
[12] K. DeJong, Learning with genetic algorithms: an overview, Machine Learning 3 (3) (1988) 121–138.
[13] D. Driankov, H. Hellendoorn, M. Reinfrank, An Introduction to Fuzzy Control, Springer, Berlin, 1993.
[14] K. Duc Do, J. Zhong-Ping, J. Pan, A global output-feedback controller for simultaneous tracking and stabilizations of unicycle-type mobile robots, IEEE Transactions on Automatic Control 30 (2004) 589–594.
[15] R. Fierro, F.L. Lewis, Control of a nonholonomic mobile robot using neural networks, IEEE Transactions on Neural Networks 9 (4) (1998) 589–600.
[16] T. Fukao, H. Nakagawa, N. Adachi, Adaptive tracking control of a nonholonomic mobile robot, IEEE Transactions on Robotics and Automation 16 (5) (2000) 609–615.
[17] D.E. Goldberg, Genetic Algorithms in Search, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[18] D.E. Goldberg, The Design of Competent Genetic Algorithms: Steps toward a Computational Theory of Innovation, Kluwer Academic Publishers, Dordrecht, 2002.
[19] H.A. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 524–539.
[20] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
[21] S. Ishikawa, A method of indoor mobile robot navigation by fuzzy control, in: Proc. Int. Conf. Intell. Robot. Syst., Osaka, Japan, 1991, pp. 1013–1018.
[22] N.N. Karnik, J. Mendel, Centroid of a type-2 fuzzy set, Information Sciences 132 (1–4) (2001) 195–220.
[23] H. Khalil, Nonlinear Systems, 3rd ed., Prentice Hall, New York, 2002.
[24] M. Kristic, I. Kanellakopoulos, P. Kokotovic, Nonlinear and adaptive control design, Wiley-Interscience, 1995.
[25] T.H. Lee, F.H.F. Leung, P.K.S. Tam, Position control for wheeled mobile robot using a fuzzy controller, IEEE (1999) 525–528.
[26] T.-C. Lee, K.-T. Song, C.-H. Lee, C.-C. Teng, Tracking control of unicycle-modeled mobile robot using a saturation feedback controller, IEEE Transactions on Control Systems Technology 9 (2001) 305–318.
[27] Q. Liang, J.M. Mendel, Interval type-2 fuzzy logic systems: theory and design, IEEE Transactions on Fuzzy Systems 8 (5) (2000) 535–550.
[28] D. Liberzon, Switching in Systems and Control, Bikhauser, 2003.
[29] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms, Concepts and Designs, Springer, 2000. pp. 5–10.
[30] J. Mendel, George C. Mouzouris, Type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 7 (December) (1999) 643–658.
[31] J. Mendel, R. John, Type-2 fuzzy sets made simple, IEEE Transactions on Fuzzy Systems 10 (April) (2002) 117–127.
[32] J. Mendel, Robert I. Bob John, Type-2 fuzzy sets made simple, IEEE Transactions on Fuzzy Systems 10 (2) (2002).
[33] J. Mendel, Uncertain Rule-based Fuzzy Logic Systems, Prentice Hall, 2001.
[34] J. Mendel, On a 50% savings in the computation of the centroid of a symmetrical interval type-2 fuzzy set, Information Sciences 172 (3–4) (2005) 417–430.
[35] J. Mendel (Ed.), Uncertain Rule-based Fuzzy Logic Systems: Introduction and new directions, Prentice Hall, USA, 2000.
[36] W. Nelson, I. Cox, Local path control for an autonomous vehicle, in: Proceedings of the IEEE Conference on Robotics and Automation, 1988, pp. 1504–1510
[37] B. Paden, R. Panja, Globally asymptotically stable PD+ controller for robot manipulator, International Journal of Control 47 (6) (1988) 1697–1712.
[38] K.M. Passino, S. Yurkovich, Fuzzy Control, Addison Wesley Longman, USA, 1998.
[39] S. Pawlowski, P. Dutkiewicz, K. Kozlowski, W. Wroblewski, Fuzzy logic implementation in mobile robot control, in: Second Workshop on Robot Motion and Control, October 2001, pp. 65–70.
[40] W. Pedrycz (Ed.), Fuzzy Modelling: Paradigms and Practice, Kluwer Academic Press, Dordrecht, 1996.
[41] R. Sepulveda, O. Castillo, P. Melin, O. Montiel, An efficient computational method to implement type-2 fuzzy logic in control applications, in: Analysis and Design of Intelligent Systems using Soft Computing Techniques, Advances in Soft Computing, vol. 41, June 2007, pp. 45–52
[42] R. Sepúlveda, O. Castillo, P. Melin, A. Rodríguez-Díaz, O. Montiel, Experimental study of intelligent controllers under uncertainty using type-1, and type-2 fuzzy logic, Information Sciences, Informatics and Computer Science Intelligent Systems Applications 177 (10) (2007) 2023–2048.
[43] K.T. Song, L.H. Sheen, Heuristic fuzzy-neural network and its application to reactive navigation of a mobile robot, Fuzzy Sets Systems 110 (3) (2000) 331–340.
[44] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics 15 (1) (1985).
[45] C.-C. Tsai, H.-H. Lin, C.-C. Lin, Trajectory tracking control of a laser-guided wheeled mobile robot, in: Proceedings of the IEEE International Conference on Control Applications, Taipei, Taiwan, September 2004, pp. 1055–1059.
[46] S.V. Ulyanov, S. Watanabe, K. Yamafuji, L.V. Litvintseva, G.G. Rizzotto, Soft computing for the intelligent robust control of a robotic unicycle with a new physical measure for mechanical controllability, Soft Computing, vol. 2, Springer-Verlag, 1998. pp. 73–88.
[47] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man, and Cybernetics 3 (1) (1973) 28–44.