

Car-Like Mobile Robot Oriented Positioning by Fuzzy Controllers

Nouredine Ouadah¹, Lamine Ourak² and Farès Boudjema³

¹ Robotics Laboratory, Centre de Développement des Technologies Avancées, BP 17, Baba hassen, Algiers 16303, Algeria

² LAAS-CNRS, 7 avenue du Colonel Roche, Toulouse 31077, France.

³ Ecole Nationale Polytechnique, BP 182, El Harrach, Algiers 16200, Algeria.
ouadah@ yahoo.fr

Abstract: In this paper, fuzzy logic controllers (FLC) are used to implement an efficient and accurate positioning of an autonomous car-like mobile robot, respecting final orientation. To accomplish this task, called "Oriented Positioning", two FLC have been developed: robot positioning controller (RPC) and robot following controller (RFC). Computer simulation results illustrate the effectiveness of the proposed technique. Finally, real-time experiments have been made on an autonomous car-like mobile robot called "Robucar", developed to perform people transportation. Obtained results from experiments demonstrate the effectiveness of the proposed control strategy.

Keywords: Fuzzy logic controller, car-like mobile robot, oriented positioning, virtual following.

1. Introduction

Human respectful way of transportation is one of the most treated projects in robotic research throughout the world. This new ways of transportation is usually based on electrical car-like mobile robots. Nevertheless, developing fully autonomous electrical car able to move in dynamic outdoor environments is not a trivial task. One of the most crucial problems is to develop an efficient and accurate algorithm allowing the car movement in an unknown environment, while respecting nonholonomic constraints. A part of this problem is to compute in real-time the inputs which imply to reach a desired destination (goal) from any initial location.

Goals are usually expressed in a 2D space by means of two coordinates plus an orientation, $[x_c, y_c, \theta_c]$. This is called an *Oriented Goal*. Most indoor robots are holonomic and usually have decoupled translation and rotation motion, thus once the 2D position has been attained, final orientation-angle can be easily reached by means of a single rotation. Unfortunately, it is not the case with nonholonomic robots where rotation is coupled to translation. In this case, positioning such robots requires a much powerful and intelligent control strategy. An accurate final orientation is generally needed in outdoor navigation for applications such as people transportation or on autonomous agriculture cars.

Many studies about autonomous navigation applied to nonholonomic robots focused on the design of a global analytic trajectory, composed of straight segments connected with either tangential circular or symmetric clothoid arcs (Scheuer & Fraichard, 1996). This approaches involves two steps: first planning a feasible

trajectory and second designing a control algorithm to follow this trajectory (Manikonda, Hendler & Krishnaprasad 1995 ; Paromtchik, Garnier & Laugier, 1997). This method has as drawback the lack of reactivity. For more reactive navigation, kinematic constraints have been introduced in the spatial representation (Santos-Victor, Minguez & Montano, 2002).

Instead of following a precise and complete trajectory, other approaches use algorithms which are based on fuzzy sets to drive the robot from an initial position to a goal (Gasós, García-Alegre & García, 1991 ; Fraichard & Garnier, 2001 ; Li, Chang & Chen, 2005). All these works has applied their approaches on nonholonomic mobile robots. Authors in (Cuesta, Gomes-Bravo & Ollero, 2004) propose a fuzzy explicit path tracking method, where a fuzzy rule based system is developed to track a pre-computed path for a holonomic small-size robot moving on complex sharp paths.

Nonholonomic vehicles have a complex and non-linear kinematic and dynamics, and therefore the interaction between wheels and ground is extremely difficult to model (Lindgren, Hague, Probert Smith & Marchant, 2002). Fuzzy controllers cope well with the incomplete and uncertain knowledge inherent to the subsystem interactions and non-linearity. In addition, fuzzy controllers are specially suited to embed the human knowledge on an autonomous mobile robot, without using a complete analytical model.

Taking into account the properties of the fuzzy systems, a fuzzy rule based controller has been designed and demonstrated to perform oriented goal navigation in an outdoor natural environment with a commercial

lawnmower automated (Garcia-Perez, Garcia-Alegre, Ribeiro & Guinea, 2003). The fuzzy rules encapsulate the expert knowledge in a minimum set, extracted from the observation of the manual operation of an expert driver in a set of field experiments. Inspired from last cited works, we propose in this work a new method based on FLC, to perform an “Oriented Positioning” on an autonomous car-like robot called Robucar.

This paper is organized as follow: Section 2 is started by an introduction to the kinematic model of the Robucar, then “Simple Positioning”, and “Oriented Positioning” principals are exposed. This section ends with computer simulation results of the proposed strategy for simple positioning task. Section 3 is devoted to present real-time experimental results of oriented positioning task applied on the Robucar. It starts with a presentation of simulation results, which are compared and the end to real-time results.

2. Autonomous positioning

The experimental platform used in this work is a car-like mobile robot. Such robot is a mobile vehicle where the front wheels can turn on the left or right, but kept always parallel. The rear wheels are fixed parallel to the car body. Such wheeled robots are systems with hard nonholonomic constraints where differential expressions involving velocity and state vectors are non-integrable. Each wheel presents kinematic constraints due to non-slipping and non-tilting conditions (Cuesta, Gomes-Bravo & Ollero, 2004). Fig.1 shows a car-like robot with corresponding parameters: $[x \ y]$ are the rear point R coordinates, and θ is the robot's heading. Linear velocity of the point R is represented by v . The orientation of the front wheels is noted ϕ , and l is the distance between the rear and the front wheels axes.

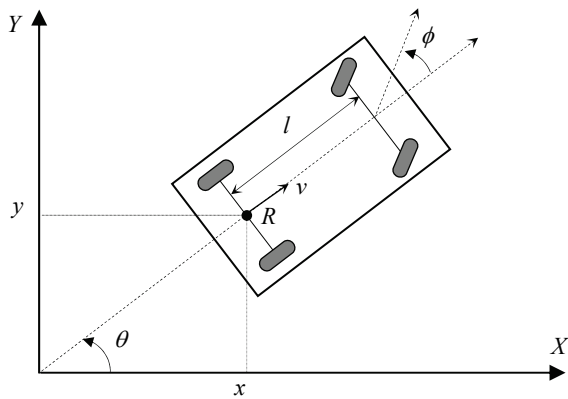


Fig. 1. Car-like mobile robot.

The kinematic equation of the robot is described by:

$$\begin{aligned}\dot{x} &= v \cos \theta \cos \phi \\ \dot{y} &= v \sin \theta \cos \phi\end{aligned}\quad (1)$$

$$\dot{\theta} = v \frac{\sin \phi}{l} \quad (2)$$

The vector $[x \ y \ \theta]^T$ is called configuration vector, and the vector $[v(t) \ \phi(t)]^T$ is the control vector. This work is divided into two steps. The first step consists on the implementation of a simple positioning. This task is done by a FLC which makes the robot moving from any initial position to a final destination, respecting nonholonomic constraints. The second task is the oriented positioning.

2.1 Simple Positioning

To accomplish a simple positioning of a car-like mobile robot, the control vector $[v(t) \ \phi(t)]^T$ must be computed in real-time, to make the robot moving from its initial position $[x_i \ y_i]$ to any desired goal $[x_G \ y_G]$, respecting its nonholonomic constraints. For that, a FLC called “Robot Positioning Controller” (RPC) has been implemented in order to insure the convergence of the robot to the desired goal. The variables used by the RPC are represented on fig. 2 :

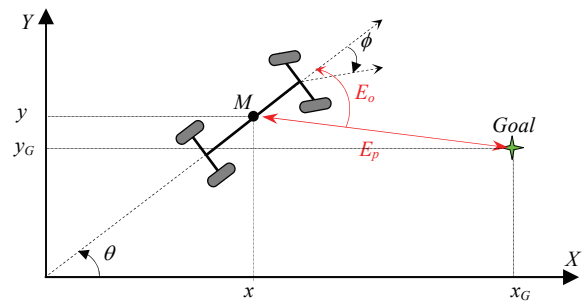


Fig. 2. Simple positioning task.

From this figure:

$$E_p = \sqrt{\Delta x^2 + \Delta y^2} \quad (3)$$

$$E_o = \theta - \Delta \theta \quad (4)$$

$$\Delta x = x_G - x \quad (5)$$

$$\Delta y = y_G - y \quad (6)$$

$$\Delta \theta = \text{Artan}(\Delta y, \Delta x) \quad (7)$$

where $[x \ y \ \theta]^T$ is the configuration vector of the robot with respect to the point M , the middle of its axis. $[x_G \ y_G]^T$ are the coordinates of the goal (desired position). E_p is called the “Position Error” and E_o is the “Orientation Error”. The implemented RPC has as inputs the two variables E_p and E_o . It provides as output the control vector of the robot $[v(t) \ \phi(t)]^T$. For the design of the RPC, membership functions are taken triangular.

In Fig. 3(a), the input variable E_p is expressed in meter whereas E_o is expressed in degree in Fig. 3(b). The input variable E_p and the output variable v are decomposed into five fuzzy partitions with triangular membership functions. The same type of membership functions is used for the input variable E_o and the output variable ϕ , but these two variables have been decomposed into seven fuzzy partitions, in order to make the control more accurate (Fig3(c) and Fig3(d)).

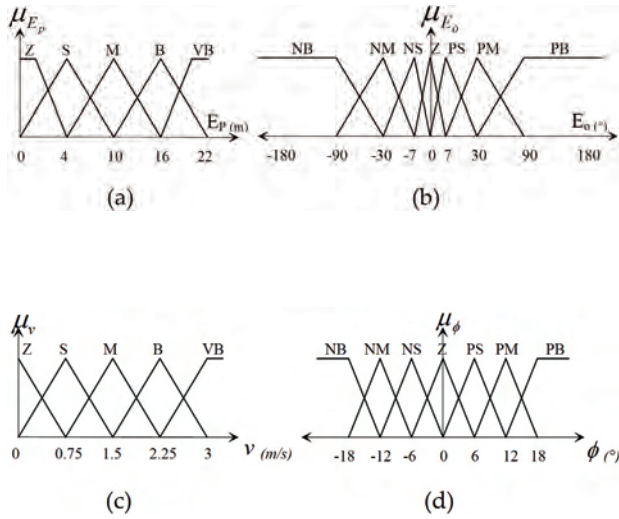


Fig. 3. (a) Membership functions of the input variable E_p . (b) Membership functions of the input variable E_o . (c) Membership functions of the output variable v . (d) Membership functions of the output variable ϕ .

The fuzzy linguistic variables employed to express the membership functions are denoted: Z for “Zero”, S for “Small”, M for “Medium”, B for “Big”, VB for “Very Big”, NB for “Negative Big”, NM for “Negative Medium”, NS for “Negative Small”, PS for “Positive Small”, PM for “Positive Medium”, PB for “Positive Big”. Setting the values for linguistic variables is not a trivial task in the design of a good and efficient FLC. These values have been set after several tests, and taking into account both dynamic characteristics of the robot and user experiences. The fuzzy reasoning rules used for the design of the RPC are summarized in table 1.

			E _o						
			NB	NM	NS	Z	PS	PM	PB
E _p	Z	ϕ	PS	PB	PB	Z	NB	NB	NS
		v	S	Z	Z	Z	Z	Z	S
	S	ϕ	PB	PB	PB	Z	NB	NB	NB
		v	S	S	S	S	S	S	S
	M	ϕ	PB	PM	PS	Z	NS	NM	NB
		v	S	S	M	M	M	S	S
	B	ϕ	PM	PS	PS	Z	NS	NS	NM
		v	S	M	B	B	B	M	S
	VB	ϕ	PM	PM	PS	Z	NS	NM	NM
		v	S	M	B	VB	B	M	S

Table 1. Fuzzy rules table for the RPC.

2.2 Oriented Positioning

Oriented positioning task consists on the accomplishment of a robot positioning, with a desired final orientation. In theory, to accomplish this task, the control vector $[v(t) \ \phi(t)]^T$ can be computed in real-time by a FLC, in order to make the robot moving from any initial configuration $[x_i \ y_i \ \theta_i]^T$ to any final desired configuration $[x_G \ y_G \ \theta_G]^T$, respecting the nonholonomic constraints, as it has been done for simple positioning. Fig. 4 shows the used variables in this case:

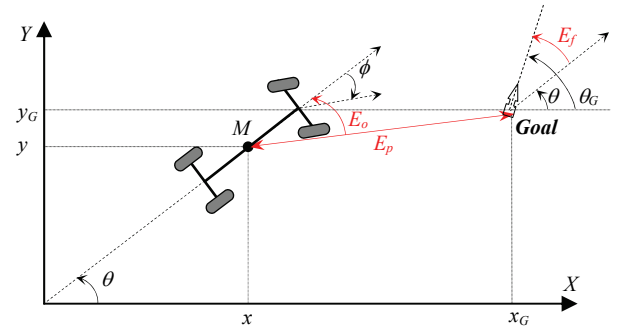


Fig. 4. Oriented positioning task.

The variables E_p and E_o are those used for the case of simple positioning. E_f is computed as follow:

$$E_f = \theta_G - \theta \quad (8)$$

E_f expresses the “Final Orientation Error”.

To achieve an oriented positioning task by using one FLC, three input variables are necessary, which are E_p , E_o and E_f . Unfortunately, only two outputs variables (v and ϕ) are available. The two input variables E_o and E_f are hardly coupled, and can not be controlled in the same time by only the two variables of the robot’s control

vector. To overcome this problem, we propose to use two FLC in the same time, as it is exposed in the next section.

- Virtual Following:

The proposed solution is the use of a second FLC, called Robot Following Controller (RFC) combined to the first one (RPC used in section 2.1). Indeed, oriented positioning task is divided into two subtasks: the first task is "Simple Positioning" (SP), where the second is "Virtual Following" (VF). Each subtask is performed by one FLC. For the SP, the controller is used to make the robot approaching the desired goal by moving from the initial position $[x_i \ y_i]$ to an intermediary goal $[x_s \ y_s]$, called "Sub-goal", as it is shown in Fig.5. The RPC developed in section 2.1 is suitable for this sub-task. The Sub-goal is chosen near to the final goal $[x_G \ y_G]$, and can be computed easily offline by geometrical formulas.

In order to keep velocity variation continuous while performing oriented positioning, the RPC used for the first step must keep the robot moving either if the sub-goal is reached. For that, fuzzy rules table has been modified, especially robot's velocity linguistic variables, as it is shown on Table. 2. The modified variables are in red italic font.

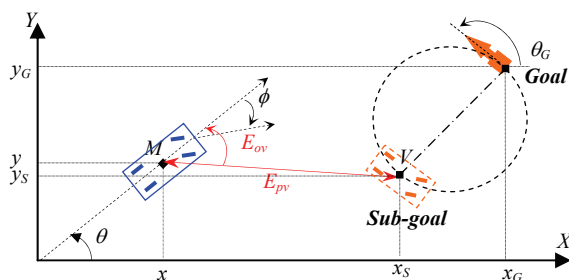


Fig. 5. Virtual Following strategy.

		E _o							
			NB	NM	NS	Z	PS	PM	PB
E _p	Z	ϕ_l	PS	PB	PB	Z	NB	NB	NS
		v_1	S	<i>S</i>	<i>S</i>	<i>S</i>	<i>S</i>	<i>S</i>	S
	S	ϕ_l	PB	PB	PB	Z	NB	NB	NB
		v_1	S	S	S	<i>M</i>	S	S	S
	M	ϕ_l	PB	PM	PS	Z	NS	NM	NB
		v_1	S	S	M	M	M	S	S
	B	ϕ_l	PM	PS	PS	Z	NS	NS	NM
		v_1	S	M	B	B	B	M	S
	VB	ϕ_l	PM	PM	PS	Z	NS	NM	NM
		v_1	S	M	B	VB	B	M	S

Table 2. Fuzzy rules table for the new RPC.

Once the robot positioned on the Sub-goal, the second FLC (RFC) has to make the robot moving on a circular trajectory by following a virtual mobile robot. The virtual robot is initially positioned on the Sub-goal. It starts moving when the real robot reaches the sub-goal, and stops on the desired goal by following a circular trajectory. This trajectory is computed to insure that the virtual robot reaches the goal with the desired orientation θ_G , as it is shown in Fig. 5. If the RFC can make the real robot following the virtual robot correctly, we are sure that at the end, the real robot will also reach the goal with the desired orientation θ_G .

The RFC used for the second sub-task has as inputs two variables: E_{vp} which is the distance between the point M (robot's center) and point V (center of virtual robot), and E_{vo} which is the angle between the segment MV and the axis of the real robot.

This controller, which is used to make the robot following the virtual robot, has been designed differently from the first one. Dynamic performances are needed to insure a good following and an accurate convergence to the desired goal. The control of velocity must be powerful to keep the robot near to the virtual robot. For that, membership functions of the input variables for the RFC has been set as it is shown in Fig. 6:

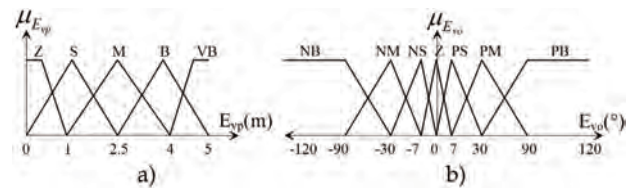


Fig. 6. (a) Membership functions of E_{vp}
(b) Membership functions of E_{vo} .

For the output variables v_2 and ϕ_2 , the membership functions are the same as for the RPC. But the rules table has been changed to make virtual robot's following more accurate. Many linguistic variables (red font size) of the rules table have been changed as shown in Table. 3:

		E _{vo}							
			NB	NM	NS	Z	PS	PM	PB
E _{vp}	Z	ϕ_2	<i>PB</i>	<i>PB</i>	<i>PB</i>	Z	NB	NB	<i>NB</i>
		v_2	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>
	S	ϕ_2	PB	PB	PB	Z	NB	NB	NB
		v_2	S	S	S	<i>S</i>	S	S	S
	M	ϕ_2	PB	PM	PS	Z	NS	NM	NB
		v_2	S	S	M	M	M	S	S
	B	ϕ_2	PM	PS	PS	Z	NS	NS	NM
		v_2	S	M	B	B	B	M	S
	VB	ϕ_2	PM	PM	PS	Z	NS	NM	NM
		v_2	S	M	B	VB	B	M	S

Table 3. Fuzzy rules table for the RFC.

The whole control strategy diagram is illustrated in Fig. 7.

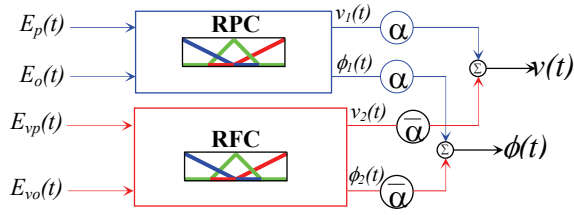


Fig. 7. Oriented positioning controllers.

The principals of the oriented positioning algorithm are shown in the figure below:

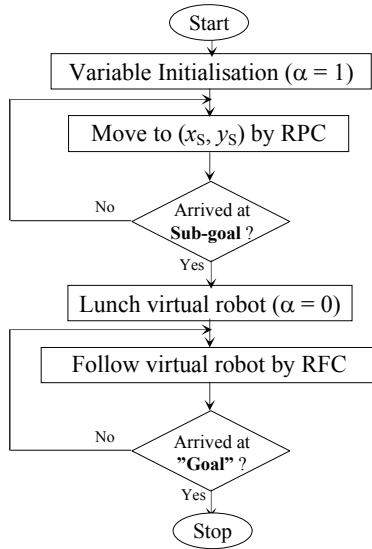


Fig. 8. Flowchart of oriented positioning.

As it is shown in Fig. 7, a Boolean variable α has been introduced to be used as a switch between the two FLC employed to achieve the oriented positioning. Initially, α is equal to "1", so that the RPC works first. Once the Sub-goal reached, α turns to "0" in order to select the RFC. By this way, the two controllers used for oriented positioning are working in serial. The controller's output variables are obtained from the following equations:

$$v(t) = \alpha v_1(t) + \bar{\alpha} v_2(t) \quad (9)$$

$$\phi(t) = \alpha \phi_1(t) + \bar{\alpha} \phi_2(t) \quad (10)$$

The used FLCs have been tested in simulation, with Simulink of Matlab 6.5, then implemented in real-time

with C++ under Linux Redhat 6. In the next section, simulation results of simple positioning are exposed and commented.

2.3 Simulation tests

To test and demonstrate the effectiveness of the proposed strategy, a computer simulation has been performed. For that, simple positioning is implemented on a car-like kinematic model. In all tests, robot's localization is computed by Incremental Odometer. Initial configuration of the robot is taken in the origin $O(0m, 0m, 0^\circ)$. This choice is not a necessary condition to get a convergence. We start testing simple positioning FLC.

In Fig. 9, four goals are chosen with the following coordinates (in meter) : $T_1(8,9)$, $T_2(-16,14)$, $T_3(-7,-13)$, and $T_4(12,-12)$. The test shown in Fig. 9(a) has been done as follow: the robot starts moving from the initial position $O(0,0)$ then goes to T_1 . Once arrived, it starts to move toward T_3 . After that, it is directed toward T_2 and finally T_4 where it stops. This test has been made to verify that the RPC controller functions correctly from any initial position to any desired goal, with both positive and negative coordinates. In the second test shown on Fig.9(b), the robot moves on a closed trajectory which starts and ends on the origin as follow:

$$(O \rightarrow T_3 \rightarrow T_2 \rightarrow T_1 \rightarrow T_4 \rightarrow T_3 \rightarrow O)$$

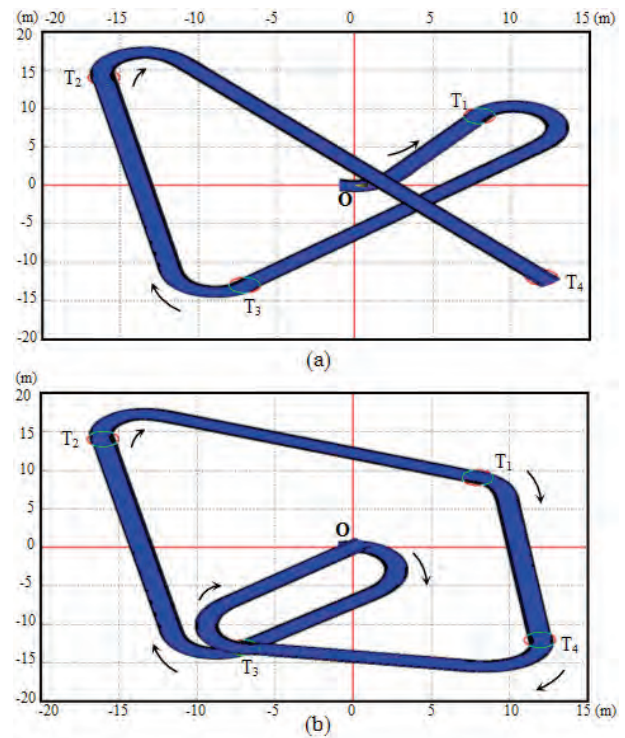


Fig. 9. Simple positioning robot trajectories.

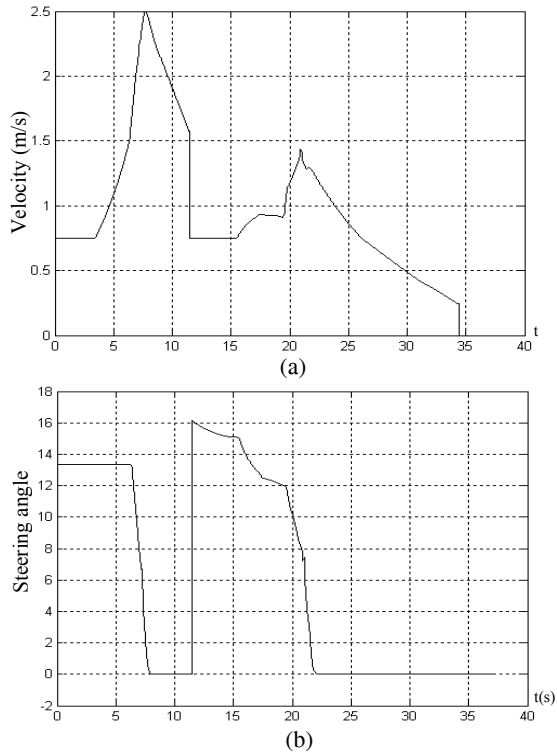


Fig. 10. Control variables in simple positioning.

The robot converges without any problem. From the first test, the control variables v and ϕ are measured when the robot was moving from the origin $O(0m,0m,0^\circ)$ to T_1 . Fig.10(a) illustrates the variations of the robot's velocity v during the simple positioning. We can see that the velocity variation is continuous and kept under limits (less than 3m/s). When the robot arrives to the goal, velocity is near to "zero".

In Fig. 10(b), the variation of the steering angle ϕ when the robot is moving to T_1 is shown. Even if the initial value is important ($\approx 14^\circ$), it does not disturb the robot because velocity v is small at this moment. In addition, the big variation of the steering angle ϕ (from 0° to 16°) between $t=10s$ and $t=15s$ does not disturb the movement of the robot, because velocity is small too. In all tests, robot stability and security was guaranteed by the used controllers.

Simulation results of oriented positioning are presented and compared to the experimental results in section 3.

3. Experiments

The proposed strategy for oriented positioning has been implemented on a car-like mobile robot called Robucar (Fig. 11).

The Robucar is a nonholonomic mobile robot designed by INRIA (France) and manufactured by Robosoft (Laugier & Parent, 1999). It is equipped by optical encoders for localization, LMS and US sensors for obstacle detection



Fig. 11. A photograph of the mobile robot Robucar and CCD camera for visual perception. Tests have been done in indoor and outdoor environments for many goals, chosen with usual orientation (0° , 45° , 90° , 180°) to make easy results verification. Fig. 12 shows trajectories for the goal $T_5(11,6)m$ with different orientations as follow:

$$(a \rightarrow -90^\circ, b \rightarrow 0^\circ, c \rightarrow 90^\circ, d \rightarrow 180^\circ).$$

The Fig. 13 shows robot trajectories for the goal $T_6(-10,8)m$ with the following orientations :

$$(a \rightarrow 45^\circ, b \rightarrow -45^\circ, c \rightarrow 135^\circ, d \rightarrow -135^\circ)$$

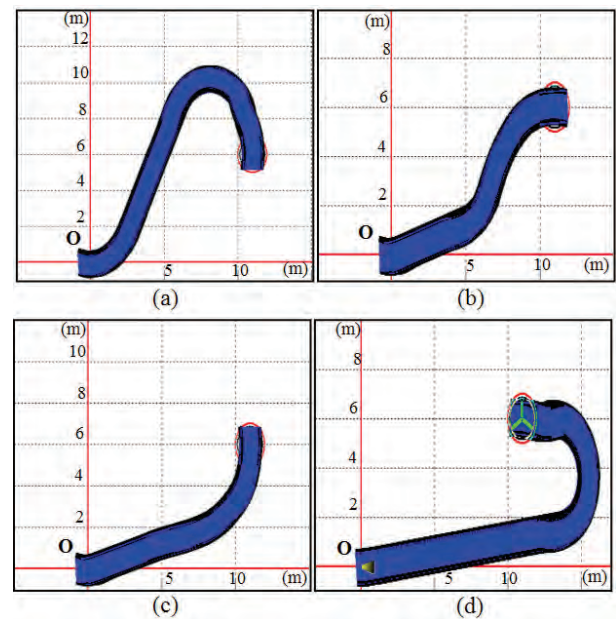


Fig. 12. Oriented positioning trajectories for goal T_5

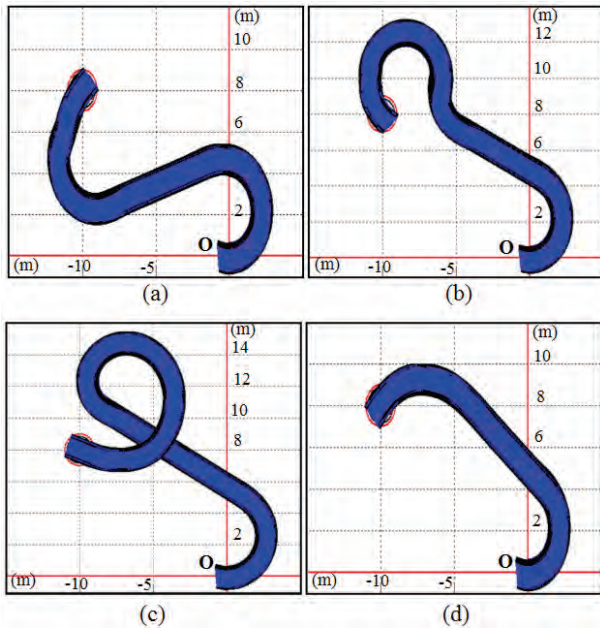


Fig. 13. Oriented positioning trajectories for goal T_6

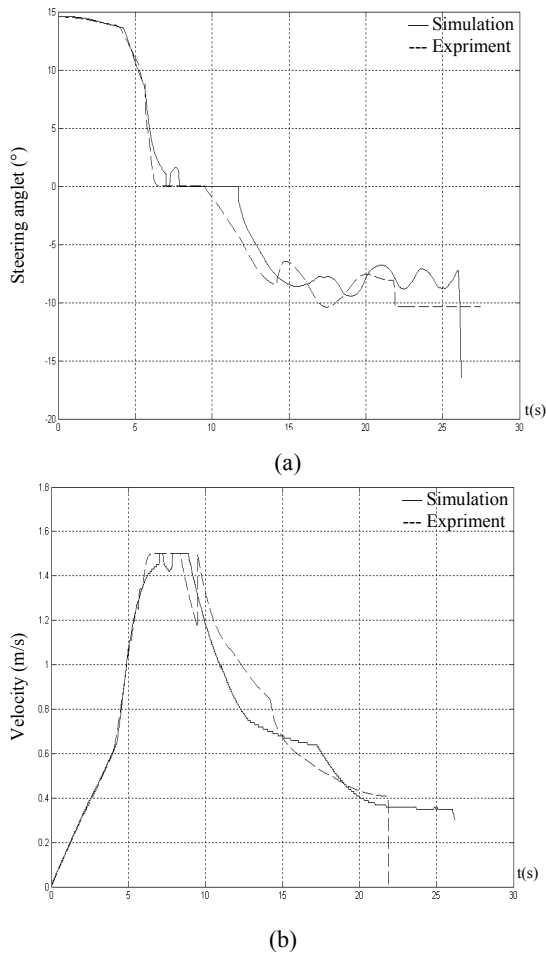


Fig. 14. Control variables in oriented positioning

A comparison between control variables variations in simulation and in real-time implementation, of the oriented positioning task, from the origin $O(0m,0m,0)$ to goal $T_7(10m,12m,-30^\circ)$ is shown on Fig. 14. Velocity variations are similar in both: differences are negligible and might be due robot-ground interactions in real tests (Fig. 14(a)). The same remarks can be done on Fig. 14(b), where the variation of the steering angle is the same in simulation and in experimental test. From all these results, we can say that the effectiveness of the proposed strategy is verified for any oriented goal $T(X_G Y_G \theta_G)$.

4. Conclusion

This paper presents a new technique used to perform the task of *Oriented Positioning* on a nonholonomic mobile robot, using two fuzzy logic controllers. The first controller used for *Simple Positioning* has been implemented and tested on simulation. Then, it is combined with a second controller, used for "Virtual Following", in order to accomplish the desired task. Real-time experimentations have been performed on the car-like mobile robot "Robucar". Obtained results show the effectiveness of the proposed techniques. It is guessed that if this technique is enhanced by introducing obstacle avoidance, it can be of interest for developing new autonomous transportation ways. So we are focusing our works on implementing an obstacle avoidance task by using fuzzy logic or neural networks.

5. References

- Cuesta, F.; Gomes-Bravo, F. & Ollero, A. (2004). Parking Maneuvers of Industrial-Like Electrical Vehicles With and Without Trailer. *IEEE Transaction on Industrial Electronics*, 51(2), 257-269.
- Fraichard, T. & Garnier, P. (2001). Fuzzy control to drive car-like vehicles. *Robotics and Autonomous Systems*, 34(1), 1-22.
- Garcia-Perez, L.; Garcia-alegre, M.; Ribeiro, A. & Guinea, D. (2003). Fuzzy control for an approaching-orienting manoeuvre with a car like vehicle in outdoor environment. *Proc. The VI Workshop de agents fisicos WAF'03*, Alicante, Spain.
- Gasós, J.; García-Alegre, M. C. & García, R. (1991) Fuzzy strategies for the navigation of autonomous mobile robots. *Proc. the 1991 International Fuzzy Engineering Symposium*, 1024-1034, Japan.
- Laugier, C. & Parent, M. (1999). Automated Vehicles for Future Urban Transport Systems. *Proc. the 1999 International Conference on Field and Service Robotics*, 29-31, Pennsylvania, USA.
- Li, T-H. S.; Chang, S-J. & Chen, Y-X. (2005). Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-

- based car-like mobile robot. *IEEE Transaction on Industrial Electronics*, 50(5), 867-880.
- Lindgren, D. R.; Hague, T.; Probert Smith, P. J. & Marchant, J. A. (2002). Relating torque and slip in an odometric model for an autonomous agricultural vehicle. *Autonomous Robots*, 13, 73-86.
- Manikonda, V.; Hendler, J. A. & Krishnaprasad, P. S. (1995). Formalizing behaviour-based planning for nonholonomic robots. *Proc. the 1995 International Joint Conference on Artificial Intelligence*, 142-149, Montreal, Canada.
- Paromtchik, I.; Garnier, P. & Laugier, C. (1997). Autonomous maneuvers of a nonholonomic vehicle. *Proc. the 1997 Fifth International Symposium on Experimental Robotics*, 213-224, Spain.
- Scheuer, A. & Fraichard, T. (1996). Planning continuous-curvature paths for car-like robots. *Proc. the 1996 International Conference on Intelligent Robots and Systems*, 1304-1311, Osaka, Japan.
- Santos-Victor, J.; Minguez, J. & Montano, L. (2002) Reactive navigation for non-holonomic robots using the ego kinematic space. *Proc. the 2002 International Conference on Robotics and Automation ICRA'2002*, 3074-3080, Washington DC, USA.