# Intelligent control of nonholonomic mobile robots with fuzzy perception

**Article** *in* Fuzzy Sets and Systems · February 2003

DOI: 10.1016/S0165-0114(02)00229-4 · Source: DBLP



Federico Cuesta
Universidad de Sevilla


Begoña C. Arrue
Universidad de Sevilla


Reinhard Braunstingl
Graz University of Technology

Available at WWW.MATHEMATICSWEB.ORG
POWERED BY SCIENCE DIRECT

ELSEVIER

Fuzzy Sets and Systems 134 (2003) 47–64

FUZZY sets and systems

www.elsevier.com/locate/fss

# Intelligent control of nonholonomic mobile robots with fuzzy perception

F. Cuesta[a,*], A. Ollero[a], B.C. Arrue[a], R. Braunstingl[b]

[a]*Dept. Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros, Universidad de Sevilla, Camino de los Descubrimientos, E-41092 Sevilla, Spain*

[b]*Department of General Mechanics, Technical University of Graz, Kopernikusgasse 24, A-8010 Graz, Austria*

**Abstract**

This paper is focused on the intelligent control of vehicles with nonholonomic constraints. A new method to compute a fuzzy perception of the environment is presented, dealing with the uncertainties and imprecisions from the sensorial system and taking into account nonholonomic constraints of the vehicle. This fuzzy perception is used, both in the design of each reactive behavior and solving the problem of behavior combination, to implement a fuzzy behavior-based control architecture. Furthermore, teleoperation and planned behaviors, together with their combination with reactive ones, are considered; improving the capabilities of the intelligent control system and its practical application. Experimental results, of an application to control the ROMEO-3R autonomous vehicle, demonstrate the robustness of the proposed method.


*Keywords:* Fuzzy control; Fuzzy perception; Hybrid control; Nonholonomic mobile robots; Teleoperation

## 1. Introduction

In the last years, autonomous mobile robots are required to navigate in more complex domains, where the environment is uncertain and dynamic. Autonomous navigation in these environments demands adaptation and perception capabilities. Reactive control strategies [6,22] imply a strong dependency on sensed information about the robot's environment [19]. So, imprecision and uncertainties in perception from sensors have to be considered. Several researchers deal with these capabilities [3,23,21,10,25]. However, in many cases, the very maneuverable, usually small, mobile robots make possible the application of reactive control strategies which are not suitable with larger

* Corresponding author. Fax: +349-5448-7340.

*E-mail addresses:* fede@cartuja.us.es (F. Cuesta), aollero@cartuja.us.es (A. Ollero), barrue@cartuja.us.es (B.C. Arrue), braun@mech.tu-graz.ac.at (R. Braunstingl).


PII: S0165-0114(02)00229-4

size autonomous vehicles required to perform many productive tasks. So, the nonholonomic and dynamic constraints on these vehicles cannot be neglected [12].

This paper describes improvements in the perception functions used in these kinds of robots. Particularly, it deals with perception functions for the autonomous vehicle ROMEO-3R designed and built at the University of Seville as an adaptation of a conventional three-wheel vehicle used for transportation of people [18]. It should be noted that this is a nonholonomic vehicle with significant limitations in the reactive capabilities due to kinematic and dynamic constraints, and to the few number of sensors and large blind sectors in between them, making autonomous navigation a nontrivial task. The methods presented in this paper have been conceived to deal with these limitations of conventional vehicles.

On the other hand, over the last years the number of applications of fuzzy logic to mobile robot control has increased significantly [5,26,23,1,21,19,20,10,15,25], mainly due to its capabilities to cope with imprecise information and the flexibility of non linear control laws. Thus, this paper presents a new method to deal with the imprecision and uncertainty inherent to perception sensors, obtaining a *fuzzy perception* of the environment. It allows to consider different types of range sensors (ultrasonic, laser, etc.) and to take into account the nonholonomic constraints of the vehicle.

Moreover, this fuzzy perception can be used straightforward to perform the control of the mobile robot by means of fuzzy behavior-based scheme already presented in literature [26,23,1,20,25]. The main differences of the proposed approach with respect to other behavior based methods are, first, that the nonholonomic constraints are directly taken into account in the behaviors (most of the existing behavior based approaches neglect this point); and second, the fuzzy perception itself can be used both in the design of each reactive behavior and to solve the problem of blending behaviors.

Furthermore, most work in mobile robotics emphasize fully autonomous navigation. However, working in non-structured environments, as frequently encountered in real applications such as mining, exploration, transportation, and so on, cannot be performed today autonomously satisfying efficiency, reliability, and safety specifications. Thus, telerobotics concepts, combining both autonomous intelligent functions and teleoperation by humans, have played an important role in applications [2,16]. On the other hand, when high precision is required, planned navigation is usually preferred to reactive navigation, in spite of the need for an accurate representation of the environment (which demands hard sensor capabilities and high computational cost) [13]. Hence, reactive control approach should be considered as complementary instead of alternative to these techniques, and any intelligent control system for this kind of vehicles should be able to deal with all of these kinds of control strategies. Thus, it will be shown that the fuzzy behavior-based control scheme presented in this paper, allows, not only, to implement reactive behaviors but also teleoperation and planned behaviors, improving system capabilities and its practical application. Furthermore, in these behaviors, soft computing techniques play an important role to solve different problems.

The paper is organized as follows. The next section, recalls some existing techniques for environment perception and robot's control. Section 3 presents a new method to obtain a fuzzy perception of the environment, dealing with nonholonomic constraints, an heterogeneous configuration of sensors, and large blind sectors. In Section 4, the perception based fuzzy control architecture is introduced. Section 5 is devoted to present some improvements in the capabilities of the intelligent control system by considering teleoperation and planned behaviors, and their combination with reactive ones. It will be also pointed out, the role of fuzzy logic in these tasks. In Section 6, several nontrivial experimental results with the nonholonomic robot ROMEO-3R are presented. The paper ends with the conclusions and references.

## 2. Environment perception and mobile robot's control

Every autonomous robot needs some sensing devices to first get a perception of its environment and then be able to move in this environment. Generally, low cost range sensors like, for instance, infrared or ultrasonic sensors, are used. However, these sensors provide an imprecise perception of the environment, making navigation a nontrivial task. So, for example, the imprecise perception of ultrasonic sensors is a result of the fact that they provide a relatively accurate measurement of the distance to an object, but poor information about its exact location due to the angular resolution (the same perception can be obtained from an object placed at different locations) (see Fig. 1). Another source of uncertainty is a consequence of the specular reflection and well known problems like cross-talking and noise (see Fig. 1). At the same time, the problem of blind sectors between two ultrasonic sensors and large blind areas of a mobile robot with few sensors may not be neglected.

Several procedures have been developed to overcome the disadvantages of low cost sensors, like, for example, using a *grid-based representation* of the environment [17,3,21], as shown in Fig. 2b, or a *feature-based representation* [7,14,10], as shown in Fig. 2c; to first map the robot's environment and then generate the appropriate control commands. However, it should be noticed that these representations of the environment still present some inherent imprecisions and uncertainties (see Fig. 2). Moreover, for some on board sensorial layout, the computational cost of building up any of such representations could be very high or even quite difficult to obtain when considering a nonholonomic mobile robot with few sensors.

Furthermore, most of the above techniques, do not consider the nonholonomic constraints on the vehicle. Thus, for instance, the well known VFH method [3], neglects the dynamics and kinematics of the vehicle, assuming that the robot is able to change its direction of travel instantly. A further development on this method, the VFH+ technique [27], takes into account an approximation of the trajectory of the mobile robot, resulting in smoother trajectories and greater reliability. It shows the convenience of considering the constraints on the vehicle to compute the control action.

On the other hand, the so-called *general perception vector* technique was introduced in [5], performing reactive navigation with an omnidirectional robot (see Fig. 3), equipped with a ring of 12 homogeneous ultrasonic sensors which provide a full perception of the robot surroundings all the time. This technique, reduces the environment to a vector, computed from the data provided by all

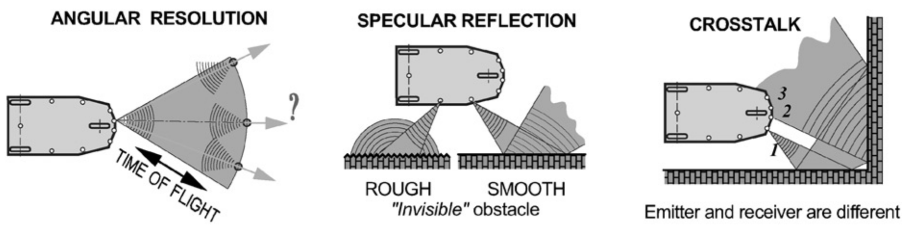



Fig. 1. Inconveniences of ultrasonic sensors.

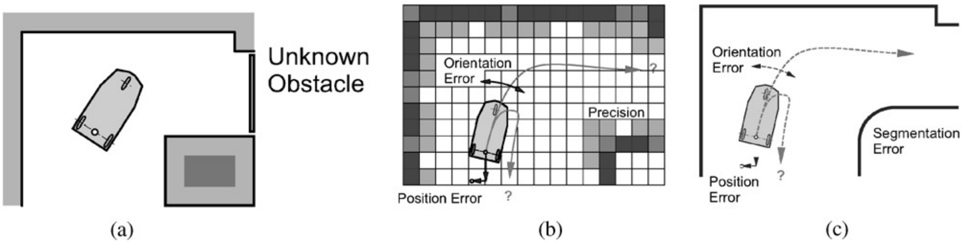



Fig. 2. (a) Vehicle in an unknown environment. (b) Grid based representation. (c) Feature-based representation.

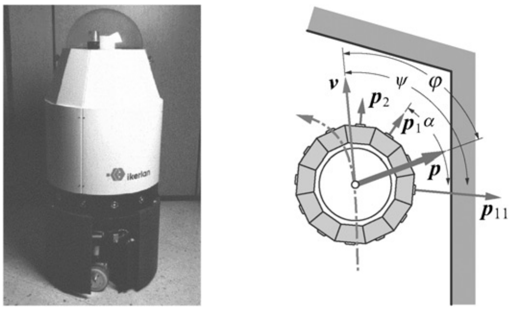



Fig. 3. Robot VEA-1 and general perception vector.

the sensors, with low computational cost but, of course, with a higher degree of uncertainty. Then, perception vector is considered in terms of fuzzy logic and the uncertainties are subsumed into the fuzzy control system. Therefore, the general perception vector characterizes situations in which the robot may find itself in rather an imprecise, but qualitatively appropriate way.

Perception of each sensor is represented by a vector $\mathbf{p}_i$, whose orientation is the direction of the sensor axis (see Fig. 3) and its length is calculated (in the range from 0 to 1) as a function of the measured distance ($d_s$) with $p_i = f(d_s) = (d_{\max} - d_s)/(d_{\max} - d_{\min})$, where $d_{\max}$ and $d_{\min}$ are the maximum and minimum distances that can be measured by the sensors, respectively. Thus, a perception length of 1 corresponds to an object detected at the minimum distance ($d_{\min}$) from the mobile robot. The *general perception vector* $\mathbf{p}$ is obtained as a combination of the calculated vectors $\mathbf{p}_i$, in such way that it has as module the maximum length of all perception vector modules $p_i$, and the direction ($\varphi$) is the same as the sum of all vectors. However, if a full perception of the mobile robot surroundings is not available at every time instant, the method fails. The next section presents improvements on the general perception vector technique to deal with nonholonomic vehicles with few sensors.

## 3. Fuzzy perception in nonholonomic vehicles

In order to apply the general perception vector method to mobile robots equipped with an heterogeneous configuration of ultrasonic sensors, with nonholonomic constraints, or without a

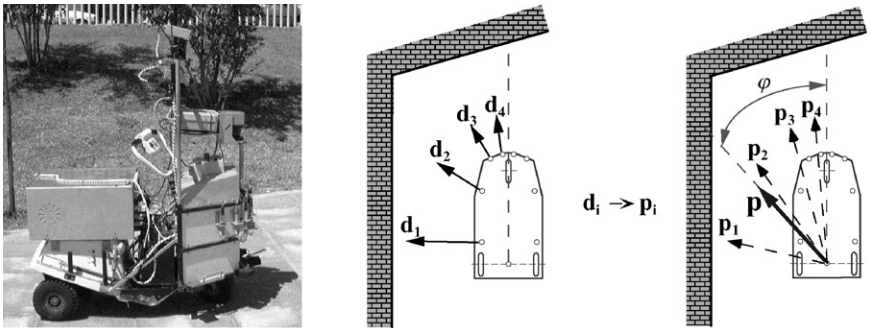



Fig. 4. Robot ROMEO-3R and general perception vector.

radial sensor layout, some modifications in its computation have to be done. So, in a vehicle like ROMEO-3R, sensor readings have to be previously translated into a common frame in order to obtain the different perceptions $\mathbf{p}_i$ as shown in Fig. 4. These perceptions will be combined giving the general perception vector $\mathbf{p}$.

On the other hand, since a vehicle with nonholonomic constraints cannot move itself in any direction at every time instant, it is interesting to weight the different perceptions according with the direction where the obstacle was detected. In other words, an obstacle is less important if it is placed at a location that cannot be reached by the mobile robot, but it is more dangerous if it is on a reachable position. This task can be accomplished by considering the perception angle ($\varphi_i$) in the computation of the *perception function*

$$p_i = f(d_s, \varphi_i) = \mathrm{sat}_{0,1}\left(\frac{d_{\max}(\varphi_i) - d_s}{d_{\max}(\varphi_i) - d_{\min}(\varphi_i)}\right), \tag{1}$$

where $\mathrm{sat}_{0,1}(x)$ states for the saturation of $x$ in the range [0,1]. In this way, it is possible to assign different perceptions, i.e. different weights, to objects detected at the same distance relative to the mobile robot but at different directions. For example, perception function

$$p_i = f(d_s, \varphi_i) = \mathrm{sat}_{0,1}\left(\frac{nd_m(1-\varepsilon) - d_s(1-\varepsilon\cos\varphi_i)}{(n-1)d_m(1-\varepsilon)}\right) \tag{2}$$

is obtained by using the nonlinear function $d_{\min}(\varphi_i) = d_m(1-\varepsilon)/(1-\varepsilon\cos\varphi_i)$, and $d_{\max} = nd_{\min}(\varphi_i)$ (with $n > 1$), in Eq. (1). Thus, an object detected in front of the vehicle, i.e. with a perception angle $\varphi_i = 0$, at the minimum distance $d_m$, will get the highest perception length ($p = 1$), while one detected at the maximum distance ($nd_m$) will get the lowest perception. Therefore, an ellipsoidal perception area, as shown in Fig. 5b, is obtained, resulting in the curves of constant perception being ellipses of eccentricity $\varepsilon$ ($0 < \varepsilon < 1$). Furthermore, it can be shown that perception function (2) results in a better reactive navigation performance and a more robust and stable perception-based control [8].

Perception vector can be considered by means of fuzzy logic using linguistic terms to describe perception angle and length as shown in Fig. 5a and b, respectively. Thus, a fuzzy description of the environment is obtained (e.g., the obstacle is located left-front and to a certain distance, between

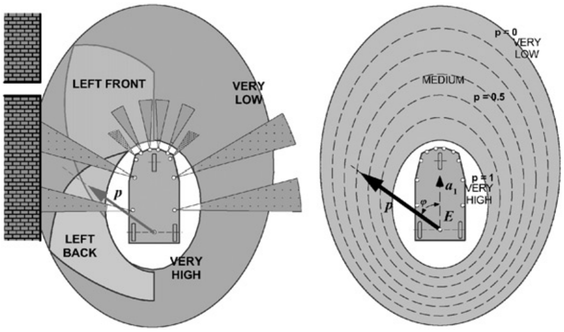



Fig. 5. Fuzzy perception.

0 and 1). It is important to recall that the perception vector does not intend to be a representation of the environment, rather than a fuzzy description of it.

Moreover, it is also possible to add a new degree of freedom by considering that the perception reference system can rotate over the vehicle reference system. This rotation is specially useful when dealing with nonholonomic vehicles, since it could make possible to adapt the perception to the trajectories described by the vehicle and to react in advance in a proper way. To do that, the perception angle is computed with respect to the so-called *direction of attention*, instead of the vehicle heading. The direction of attention corresponds to the orientation of the perception frame and it changes as the vehicles moves on. For example, due to the kinematics of the ROMEO vehicle, the direction of attention will be kept always parallel to the front wheel, which results in smoother trajectories as will be shown in Section 4.1.

Furthermore, it is interesting to stress that the perception vector implies a fuzzy high level description of the environment, being independent of the type of range sensor used. So, it is possible to use different *perception functions* from Eq. (1) for each kind of sensor (laser, ultrasonic, infrared). Thus, *sensor data fusion* can be reduced to compute different vectors from the sensor measurements (by using different perception functions, with different weights) and to combine them to obtain the perception vector.

Nevertheless, if a full perception of the surroundings is not available, the method still fails because the robot does not recognize the situation in which it is. This is illustrated in Fig. 6a by a real experiment, performing left wall following with simple general perception vector (see Section 4 for details on the controller), with the robot being unable to fulfill its mission. At the beginning, the robot tries to align itself with respect to the wall at a given distance. Later on, although the robot detects the obstacle with one sensor and starts turning right correctly, it turns left again immediately when the sensor loses contact to the obstacle. Although it rounds the obstacle successfully and starts perceiving the other wall with one of its digital front sensors, the robot loses perception this time immediately when turning right. It turns back left a little bit, gets an echo from the wall just to lose it again when turning right. These improper control actions are repeated until the robot hits the wall. Therefore, a further improvement of the method is needed, to be applied to mobile robots with few sensors or large blind sectors that cannot be neglected due to the kinematics constraints that appears in nonholonomic mobile robots. This technique, called *virtual perception memory*, is presented in the next section.

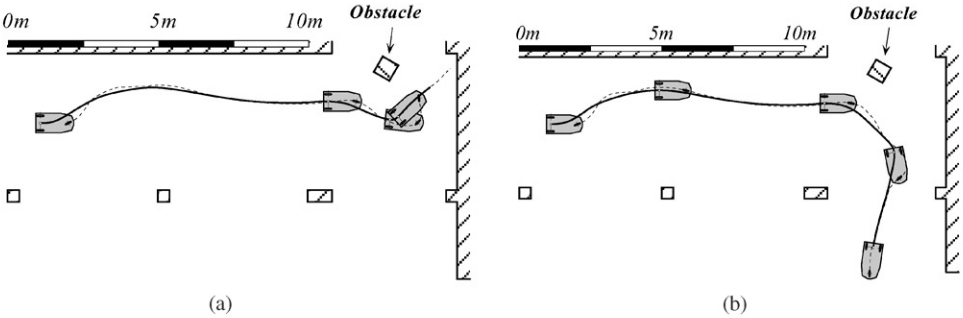



Fig. 6. (a) ROMEO-3R with loss of perception while turning right (dotted line represents the front wheel path). (b) With perception memory.

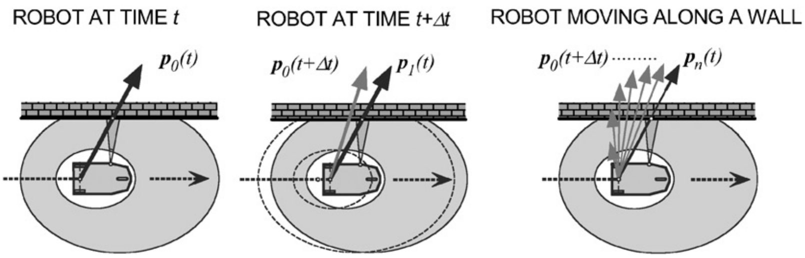



Fig. 7. Perception tracking.

### 3.1. Virtual perception memory

The main idea behind virtual perception memory is to take into account the previous perception to build the new perception vector [4]. The perceptions, are updated as the robot moves on, and used as virtual sensors (see Fig. 7).

The previous perception can be updated as follows: consider a robot of arbitrary shape equipped with proximity sensors. Any such sensor may be located at a position $\mathbf{u}$, with its axis pointing to the direction $\mathbf{s}$ (see Fig. 8). A frame $\mathbf{r}$ represents the robots position and orientation, $\mathbf{x}$ and $\theta$, respectively, with respect to the world reference system $\mathbf{w}$. The velocity $v$ of the reference point and the angular velocity $\omega_{r/w} = \dot{\theta}$ of the robot with respect to the fixed frame $\mathbf{w}$, give the state of motion. Furthermore, the virtual perception coordinate system is assumed to be located at $\mathbf{E}$, pointing to the *direction of attention* $\mathbf{a}_1$. Then, an object detected by a proximetry sensor at a distance $d_s$ could be detected by a virtual sensor placed at $\mathbf{E}$ a distance $d$, and with an orientation $\varphi$ with respect to the vehicle's direction of attention $\mathbf{a}_1$.

Now the virtual perception will be updated taking into account the robots motion as follows: considering a perception function $p = f(d, \varphi)$ and the corresponding *inverse perception function*, $d = g(p, \varphi)$, and carrying out some calculations, it can be shown (see [4] for details) that the derivatives of angle and length of the perception vector are given by (assuming $g \neq 0$ and $\partial g/\partial p \neq 0$)

$$\dot{\varphi} = \frac{1}{g}((\dot{\mathbf{x}} + \omega_{r/w} \times \mathbf{e})(\mathbf{r}_1 \sin(\alpha + \varphi) - \mathbf{r}_2 \cos(\alpha + \varphi))) - \omega_{r/w} - \omega_{a/r},$$

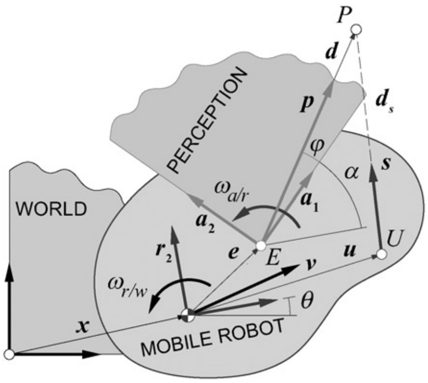



Fig. 8. Virtual perception $p$ of a mobile robot.

$$\dot{p} = -\frac{1}{\partial g/\partial p}\left((\dot{\mathbf{x}} + \omega_{r/w} \times \mathbf{e})(\mathbf{r}_1 \cos(\alpha+\varphi) + \mathbf{r}_2 \sin(\alpha+\varphi)) + \frac{\partial g}{\partial \varphi}\dot{\varphi}\right), \tag{3}$$

where $\omega_{a/r} = \dot{\alpha}$ is the angular velocity of the virtual perception coordinate system relative to the robot. Integration of Eq. (3) yields the virtual perception.

Thus, if the perception function is given by Eq. (2), then the inverse perception function is

$$g(p,\varphi) = d_{\mathrm{m}}\,\frac{1-\varepsilon}{1-\varepsilon\cos\varphi}\,(n - p(n-1)). \tag{4}$$

So, looking in the direction of attention $\mathbf{a}_1$ ($\varphi = 0$), $p = 1$ corresponds to an object at a distance $d_{\mathrm{m}}$ to the perception reference system $\mathbf{E}$, while $p = 0$ is mapped to a distance $nd_{\mathrm{m}}$. Partial derivatives of $g(p,\varphi)$, required to compute derivatives of angle and length of the perception vector in Eq. (3), are

$$\frac{\partial g}{\partial p} = d_{\mathrm{m}}\,\frac{1-\varepsilon}{1-\varepsilon\cos\varphi}\,(1-n), \tag{5}$$

$$\frac{\partial g}{\partial \varphi} = d_{\mathrm{m}}\,\frac{\varepsilon(1-\varepsilon)}{(1-\varepsilon\cos\varphi)^2}\,((n-1)p - n)\sin\varphi. \tag{6}$$

Inverse perception function (4) is a valid one since it fulfills the assumptions $g \neq 0$ and $\partial g/\partial p \neq 0$.

Fig. 6b shows the same real experiment as that of Fig. 6a, but now using the virtual perception memory. Thus, the robot also loses perception of the obstacle after starting to turn right. However, this time it tracks the perception it had before and the controller acts as if the robot continuously perceived the obstacle. Similar situations occur sometimes during the wall following operation. The robot passed the test without difficulties at an average speed of 0.5 m/s.

## 4. Perception vector based fuzzy control for nonholonomic vehicles

As it has been stated in Section 3, perception vector can be considered by means of fuzzy logic yielding a fuzzy description of the environment. This description of the environment can be easily

Table 1
Rule base controlling the robots direction of motion

| $\varphi_l \backslash p_l$ | VL | L | M | H | VH |
|---|---|---|---|---|---|
| LF | C | C | R | HR | HR |
| LC | HL | L | C | R | HR |
| LB | HL | L | C | C | R |

used as input to a fuzzy controller to perform reactive navigation as will be shown in this section. Furthermore, it is also possible to compute different perception vectors from the virtual perception memory (covering different areas around the mobile robot: left, right and front sides, for instance), and to use them to implement fuzzy controllers or behaviors which perform specific tasks taking into account nonholonomic constraints. The combination of the different behaviors, in a *cooperative scheme*, can be also easily done by means of fuzzy logic, thanks to the information provided by the perception vectors. In the following, a detailed description of the perception based fuzzy control system is performed, including implementation and combination of behaviors.

### 4.1. A behavior based control scheme based on the perception vector

Each behavior is defined by a set of fuzzy rules supplying control commands to the mobile robot. The rules have a fuzzy description of the environment obtained from the virtual memory perception as antecedents, and the control commands to be applied to the mobile robot, i.e., steering angle and linear velocity, as consequents. The implemented reactive behaviors are the following ones: *left wall following*, *right wall following*, *obstacle avoidance*, *turn around* and *corridor following*.

In case of *left wall following*, for instance, the necessary input is the perception to the left side of the mobile robot. Then, a *left hand perception vector* is built based on the perceptions stored in the virtual perception memory, yielding a left perception angle ($\varphi_l$), together with a left perception length ($p_l$). The goal of this behavior is to keep the left perception vector at a medium value and pointing to *left center*. The left wall following fuzzy logic controller is a conventional Mamdani one composed of fuzzy rules such as

IF $\varphi_l$ is LEFT_BACK and $p_l$ is HIGH then MAKE *steer* CENTER;

Table 1 shows the whole rule base controlling the robots direction of motion by means of the steering angle, where $\varphi_l$, for example, is described by the linguistic terms LF (Left_Front), LC (Left_Center), and LB (Left_Back). Fig. 9 shows the membership functions for the linguistic terms.

In the same way, the velocity command is obtained using rules like

IF $\varphi_l$ is LEFT_FRONT and $p_l$ is LOW then MAKE *velocity* MEDIUM;

The other behaviors are defined in a similar way. For example, *right wall following* behavior is equivalent to the left one, but requires the computation of a *right hand perception* vector ($\varphi_r$ and $p_r$). *Corridor tracking* behavior is straightforward obtained from the combination of left and right wall following behaviors (this also shows how primitive behaviors can be used as building blocks to produce other, even more complex, behavior by means of a hierarchical combination). In order

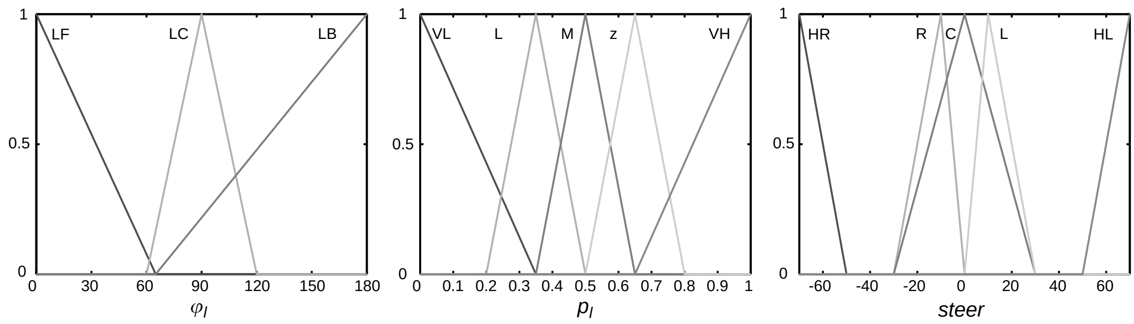



Fig. 9. Membership functions for $\varphi_1$, $p_1$ and *steer*.

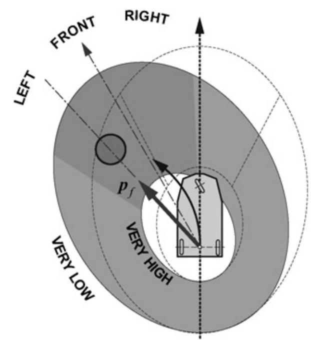



Fig. 10. Improving obstacle avoidance through direction of attention.

to implement the *obstacle avoidance* behavior in the forward motion of the robot, the left, right and frontal perception vectors are used. The frontal perception ($\varphi_f$ and $p_f$) covers a particular sector in front of the mobile robot (dark area in Fig. 10), and there are similar fuzzy rules used like:

IF $\varphi_f$ is RIGHT and $p_f$ is VERY_HIGH and $p_l$ is NOT VERY_HIGH
then MAKE *steer* HARD_LEFT;

At this point, it is convenient to remark the effect of moving the *direction of attention* (see Section 3) in the obstacle avoidance performance, making possible a faster and safer navigation. Consider, for example, the situation shown in Fig. 10, where the nonholonomic mobile robot approaches the obstacle by turning left, following the given trajectory due to its kinematics constraints. If the direction of attention is not considered, then the obstacle avoidance behavior will not practically react to the obstacle, because it is considered non-dangerous since it lays out of the frontal sector (see dashed ellipse), resulting in the sudden appearance of the obstacle in front of the vehicle after the turn. However, if the nonholonomic constraints are taken into account, i.e. direction of attention is kept parallel to the front heading wheel, then the obstacle avoidance behavior could react early, giving a smooth trajectory avoiding the collision.

*Turn around behavior* is used to perform a turn of 180° when there is no feasible way in front of the vehicle (see Fig. 11). This may also imply a corridor that is too narrow. The aim of this

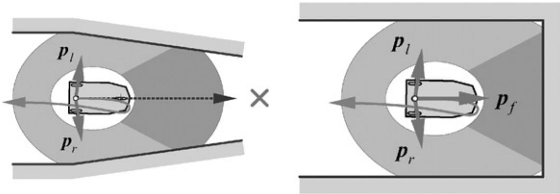



Fig. 11. Suitable situations to apply turn around behavior.

behavior is to avoid guiding the robot into a situation from which it could not recover without reverse motion due to its kinematic constraints.

### 4.2. Behavior blending

The combination of the previous behaviors is also made by means of fuzzy logic. Thus, working in parallel, each behavior produces its own control command vector, named $\mathbf{c}_i$ (steering and velocity). These vectors are fused to obtain a single command vector $\mathbf{c}$ by weighting the output of each behavior with a *behavior weight* ($BW_i$) as

$$\mathbf{c} = \frac{\sum_i (BW_i \mathbf{c}_i)}{\sum_i BW_i}, \quad i = flw, frw, oa, trn, \dots \ . \tag{7}$$

Behavior weights determine the relevance of each behavior at every time instant, and they are calculated dynamically taking into account the situation the mobile robot is in. For example, the obstacle avoidance behavior weight ($BW_{\mathrm{oa}}$) increases as the obstacle comes closer. Therefore, the influence of a behavior on the mobile robots action is situation dependent. Several approaches have been proposed in the literature which could be used to compute the behavior weights (see, for instance, [23] and the references therein, [26,20]). Some of these methods require to compute a set of specific (fuzzy) variables, or context rules, describing situations where the corresponding behavior should be activated. Nevertheless, in this paper, the behavior weights can be obtained directly from the information provided by the different perception vectors.

For example, in the case of left wall following behavior, the behavior weight has to be high if a wall is detected at the left side and at a medium distance (i.e., when the left perception vector is pointing at *LeftCenter* with a perception length around 0.5). On the contrary, if a wall is detected at the left front and very near, it would not be a good solution to perform wall following but avoiding collision. The left wall following behavior weight ($BW_{\mathrm{flw}}$) must be low in this case. This can be computed as follows

$$BW_{\mathrm{flw}} = \mathrm{sat}_{0,1}\left(1 - \left|\frac{\varphi_1 - LeftCenter}{2}\right|\right)\mathrm{sat}_{0,1}(2p_1), \tag{8}$$

where *LeftCenter* corresponds to a perception angle $\varphi_1$ with the highest membership degree to the linguistic term *Left_Center* ($\varphi_1$: $\mu_{\mathrm{LC}}(\varphi_1) = 1$). The first factor in Eq. (8) takes into account the relative orientation error of the robot with respect to the wall. This factor takes 1 when the wall is detected with a perception angle of *LeftCenter*, decreasing progressively as the error grows. The second factor computes how far away the robot is from the wall. It is 1 if the wall is detected with

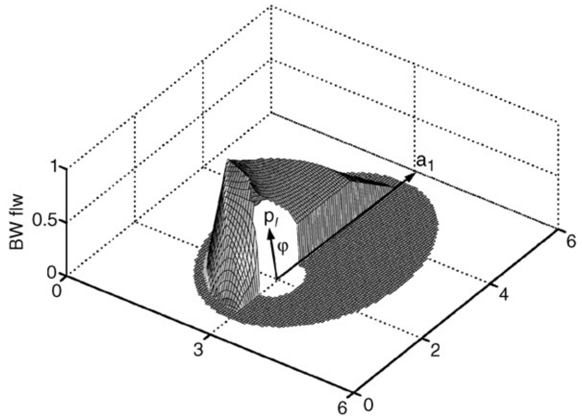



Fig. 12. Behavior weight surface for left wall following.

a perception length of 0.5 or closer. When this happens, the final $BW_{\text{flw}}$ depends fully on the relative orientation error. Fig. 12 shows $BW_{\text{flw}}$ over the perception area, i.e., it shows that $BW_{\text{flw}}$ reaches the maximum values for perception angles close to *LeftCenter* and with perception lengths larger than 0.5.

On the other hand, the weight of the obstacle avoidance must be higher as the obstacle is closer. Therefore, it follows $BW_{\text{oa}} = \text{sat}_{0,1}(1-|\varphi_{\text{f}}/2|)\text{sat}_{0,1}(2p_{\text{f}})$. Furthermore, it is possible to define priorities between behaviors. This can be done by taking into account the weight of one behavior to compute the weight of the remaining. For example, if $BW_{\text{flw}}$ and $BW_{\text{oa}}$ get a high value at the same time, the priority of the left wall following behavior will decrease as $BW_{\text{flw}} = BW_{\text{flw}}(1 - BW_{\text{oa}})$. In Section 6, the application of this behavior fusion to control a real nonholonomic mobile robot is presented.

## 5. Improving reactive capabilities

The fuzzy behavior based control scheme presented in this paper, allows not only to implement reactive behaviors, but also teleoperation and planned behaviors, improving the system capabilities. Furthermore, in these behaviors, soft computing techniques play an important role to solve different problems. In the following, a detailed description of such behaviors is performed.

### 5.1. Teleoperation behavior

Several previous works have shown the applicability of telerobotic concepts in the field of autonomous mobile robots. In [2] the robot follows the general direction prescribed by the remote operator and, at the same time, autonomously avoids the collision with the obstacles while trying to match the prescribed direction as closely as possible. A more complex system is presented in [16], where the AURORA mobile robot performs autonomously greenhouse tasks and the teleoperator acts as a supervisor taking control if needed. So, a new teleoperation station has been implemented to show the behavior based system capabilities [9]. Intelligent functions including autonomous path tracking, collision detection and path planning have been also implemented.

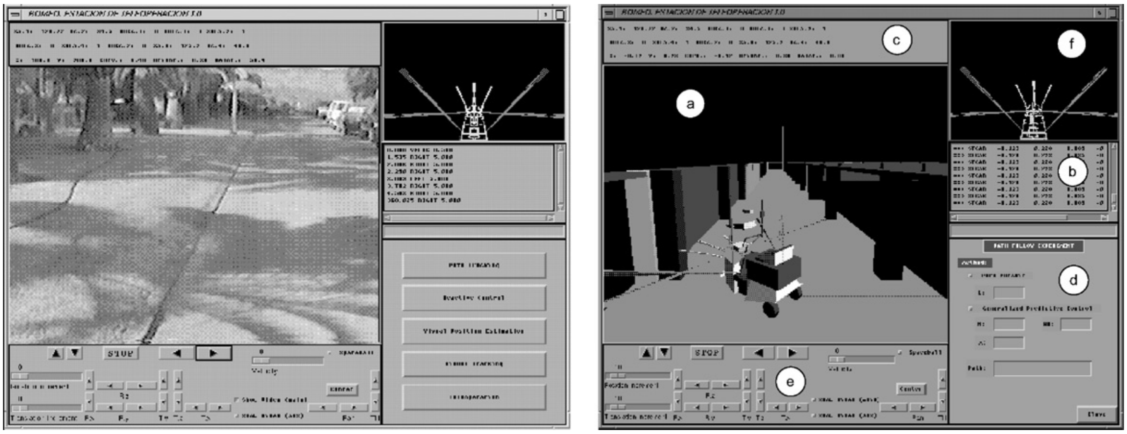



Fig. 13. Screen layout: (a) 3D window, (b) text dialog window, (c) status window, (d) experiment window, (e) auxiliary menu, (f) sensor information.

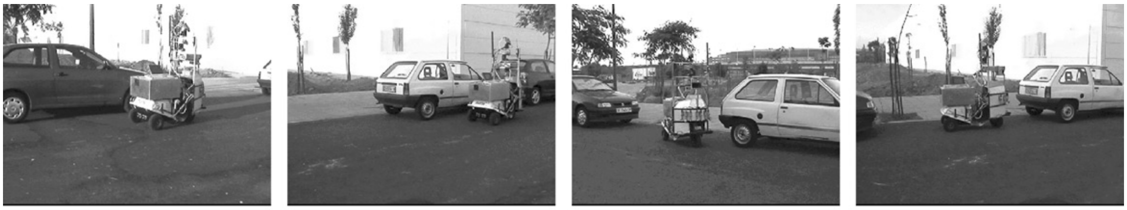
Moreover, for previously modeled environment, there are graphical simulations of the robotic platform travelling in the environment (virtual views) and predictive graphical simulation (wire-frame vehicle acting as an undelayed cursor) overlapped onto real video or virtual view (Fig. 13).

By changing the teleoperation behavior weight, the teleoperation system allows for different control methods, namely:

- Direct teleoperation of the vehicle by means of steering device.
- Supervision of the autonomous mobile robot navigation.
- Combination of teleoperation and on-board autonomous functions.

The main intelligent functions where fuzzy logic could play a significant role are the following ones:

- *Teleprogramming and qualitative command interpretation*: interpretation of commands expressed in qualitative linguistic terms (e.g. sharp turning to the right, large velocity increase) by using fuzzy logic.
- *Planning and tracking an explicit path*: several alternative path tracking algorithms are available including fuzzy path tracking [19].
- *Reactive navigation*: the teleoperation system allows several kinds of reactive navigation with different levels of autonomy, ranging from: (1) *Single behavior activation*: the vehicle is commanded to follow a left wall, a corridor, etc. (2) *Behaviors sequence*: the operator defines a *route* to be tracked by means of a sequence of behaviors [24]. For example, a fuzzy path could be ⟨*follow a left wall, then follow the corridor and after turn right*⟩ corresponding to the sequence: *Left Wall Following*, *Corridor Following*, *Turn Right*. (3) *Goal-point behavior*: the operator just only selects a goal point on the screen and the mobile robot tries to reach it avoiding the obstacles it found. (4) *Multiple behaviors*: a pure autonomous reactive navigation is performed by means of multiple simultaneous behaviors. Note that this also includes the possibility of using teleoperation as another behavior.



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

Fig. 14. ROMEO-3R performing a lateral parking.

### 5.2. Hybrid control: combination of reactive and planned behaviors

The behavior based control scheme also allows an approximation to the problem of hybrid control, i.e., the combination of reactive and planned issues. Thus, for instance, autonomous parking has been implemented combining planning and reactivity. The architecture takes advantage of the reactive navigation robustness, and relatively low precision required to navigate parallel to a line of vehicles looking for a parking place. At the same time planned navigation allows to perform an accurate navigation where high precision is required due to the characteristics of the maneuver.

So, when the vehicle is looking for a parallel parking (Fig. 14a), three behaviors cooperate: *wall* (*or line of vehicles*) *following*, *obstacle avoidance* and *lateral parking maneuver*. In this phase, the lateral parking behavior is only searching for a place large enough to park into, and does not generate commands to the vehicle (indeed, its behavior weight is zero). When a parking place is found (Fig. 14b), length and depth of the parking place are estimated. If the place is not large enough, the parking place is rejected and the vehicle continues navigating looking for a new place. In other case, lateral parking behavior performs the following steps [11]: (1) a fuzzy system is used to decide the best starting location to initiate the parking maneuver, in the same way that a human driver selects the starting point depending on the characteristics of the parking place; (2) stops the vehicle on the selected location (wall following behavior weight is set to zero); (3) computes a collision-free trajectory satisfying nonholonomic constraints; (4) activates the tracking of the planned path in reverse (Fig. 14c–d) (to do that, several path tracking techniques including fuzzy logic can be applied [19]).

## 6. Experimental results

This section presents some experimental results of the proposed methods to the autonomous vehicle ROMEO-3R (see Fig. 4). It has a nonholonomic tricycle locomotion and has been adapted from a conventional three-wheeled car [18]. The vehicle carries on-board an heterogeneous configuration of ultrasonic sensors. Instead of a typical ring of identical sonars, there are ten sonars of three different types, placed at different locations (see Fig. 5a). Four of them are large-range analogue sensors (0.6–3.0 m), four are Mid-Range Digital (0.2–1.0 m), and the other two are of Short-Range Digital (0.06–0.3 m). Furthermore, these ultrasonic sensors (Siemens Bero sonars) use a higher frequency and have a narrower sonar beam than the commonly used sonars in these kind of applications. The sensors are arranged in a way that six of them cover the front part of the vehicle and the other four

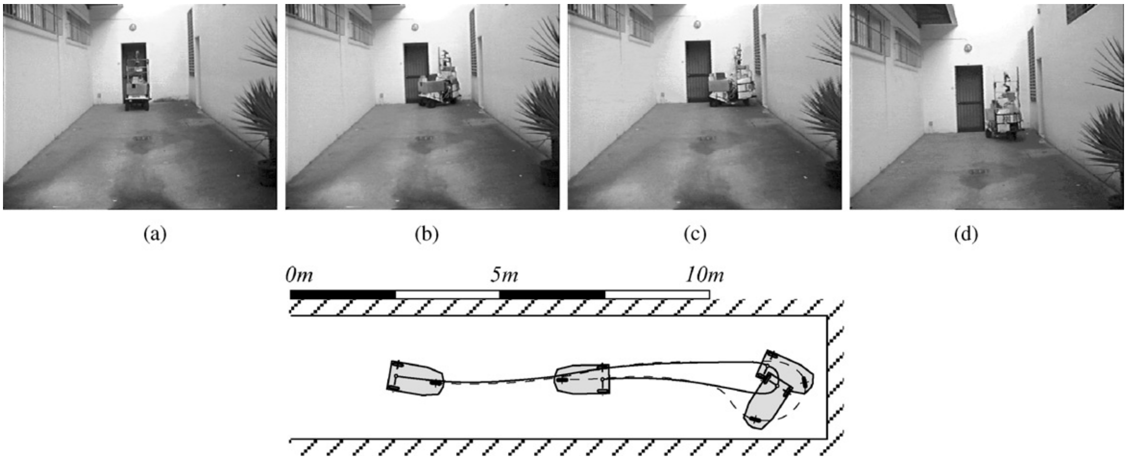



Fig. 15. Following a narrow corridor.

cover its lateral sides (two on each side). Thus, due to the relative low number of narrow sonar beam sensors, ROMEO-3R has relatively large blind sectors (see Fig. 5a).

In these experiments, the virtual perception system was placed at the center of the robot's rear axle and the direction of attention $\mathbf{a}_1$ was kept parallel to the front wheel. The perception function used is given by Eq. (2), with $n=3$, $d_{\mathrm{m}}=1.3$, and $\varepsilon=0.5$. All the experiments have been implemented in the ROMEO-3R on-board 486 DX2 66 MHz, PC computer with a real time UNIX operative system, LYNX OS v2.2, at a maximum speed of 0.9 m/s, even when the maximum speed allowed by the ROMEO-3R is about 1.3 m/s.

It is interesting to note that, although the experiments presented in this section could seem rather simple, they are quite difficult due to the few sensors available, large blind sectors in between them, the nonholonomic constraints and dimensions of the vehicle, and the way the obstacles were found.

So, in Fig. 15 the robot performs a wall following mission in a narrow dead-end corridor. In the extreme situation at the end of the corridor accurate and fast control response is required. When approaching the end wall, the frontal digital sensors detect the wall and the robot performs a fast right turn. While turning the robot loses perception of each wall temporarily due to the narrow sonar beam of the digital sensors and the incidence angles. However, perceptions of the sensors are transformed to virtual perceptions, kept in memory, and tracked as the robot moves on, thus enabling the robot to turn around smoothly. Again, the difficulty of this maneuver should be emphasized, due to the corridor width, the nonholonomic constraints in the kinematics of the vehicle, and the low number of sensors (generating large blind sectors). Thus an accurate control action is needed to avoid collision (see Fig. 15).

Another experiment is shown in Fig. 16 where the robot has to navigate through a corridor which is partially obstructed by an obstacle. The robot starts at point (x1) with corridor tracking behavior, since it has equal perception at both sides. As the robot moves on it detects free space to its right and changes its behavior smoothly to follow left wall. When entering the corridor it tries again to center itself in the corridor (x2) until it encounters the obstacle at (x3) and the obstacle avoidance behavior becomes dominant (note that the obstacle is detected just in the front of the vehicle, and

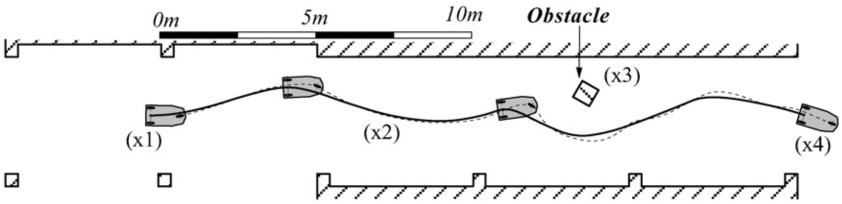



Fig. 16. ROMEO-3R with reactive behavior navigation.

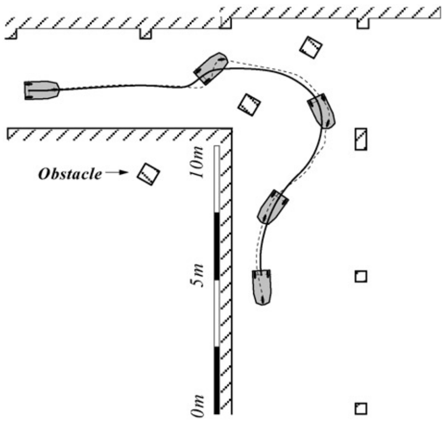



Fig. 17. Corridor following and turning right.

the collision risk is higher due to the nonholonomic constraints). The corridor is wide enough, i.e. the perception at both sides is sufficiently low that the turn around behavior is not activated, so the robot tries to round the obstacle and, indeed, detects the passage between the obstacle and the wall. From this point on the robot is again guided mainly by the corridor tracking behavior until (x4).

Finally, Fig. 17 shows the mobile robot in an autonomous behavior experiment avoiding two obstacles doing a right turn, to later on keep following a right wall. Note again the difficulties performing this experiment due to the features of the vehicle and the environment involved, making navigation a nontrivial task.

## 7. Conclusions

Several reactive mobile robot control techniques do not consider the nonholonomic constraints on the vehicle. However, the dynamics and kinematics constraints should not be neglected to obtain smoother trajectories and greater reliability.

In this paper, a new method for the intelligent control of nonholonomic vehicles has been presented. Thus, a technique to obtain a fuzzy perception of the environment, dealing with the uncertainties, imprecisions and blind sectors from the sensorial system and taking into account nonholonomic and dynamical constraints of the vehicle, has been introduced. Moreover, this fuzzy perception is directly used, both in the design of each reactive behavior and solving the problem of behavior combination, to implement a fuzzy behavior based control architecture. It should be remarked that, at difference with other behavior based approaches, in the proposed technique the nonholonomic constraints are considered in the design of each behavior.

Furthermore, in order to improve the capabilities of the intelligent control system and its practical applicability, teleoperation and planned behaviors, together with their combination with reactive ones, have been considered. Experimental results, of an application to control the ROMEO-3R autonomous vehicle, demonstrate the robustness of the proposed method.

In all the above control methods, artificial intelligence techniques have played an important role, solving a wide class of different problems (ranging from the fuzzy perception to a fuzzy supervisory path tracking method).

## Acknowledgements

The authors would like to thank the Editor and anonymous reviewers for their comments to improve this paper. This work has been supported in part by the CYCIT TAP99-0926-C04-01.

## References

[1] B.C. Arrue, F. Cuesta, R. Braunstingl, A. Ollero, Fuzzy behaviors combination to control a nonholonomic mobile robot using virtual perception memory, in: FUZZ-IEEE97, Barcelona, 1997, pp. 1239–1244.

[2] J. Borenstein, Y. Koren, Teleautonomous guidance for mobile robots, IEEE Trans. Systems Man Cybernet. 20 (6) (1990) 380–394.

[3] J. Borenstein, Y. Koren, The vector field histogram-fast obstacle avoidance for mobile robots, IEEE J. Robot. Automat. 7 (3) (1991) 278–288.

[4] R. Braunstingl, Improving reactive navigation using perception tracking, in: 2nd World Automation Congress, Montpellier, 1996.

[5] R. Braunstingl, P. Sanz, J.M. Ezkerra, Fuzzy logic wall following of a mobile robot based on the concept of general perception, in: Proc. Seventh Internat. Conf. on Advanced Robotics, vol. 1, Spain, 1995, pp. 367–376.

[6] R.A. Brooks, A robust layered control system for a mobile robot, IEEE J. Robot. Automat. R.A-2 (1) (1986) 14–23.

[7] J.L. Crowley, Navigation for an intelligent mobile robot, IEEE J. Robot. Automat. RA-1 (1) (1985) 31–41.

[8] F. Cuesta, Stability analysis of multivariable fuzzy systems. Application to reactive control of mobile robots, Ph.D. Thesis, Dept. Ingeniería de Sistemas y Automática, Univ. de Sevilla, 2000. (in Spanish).

[9] F. Cuesta, B.C. Arrue, A. Ollero, A new system for intelligent teleoperation of vehicles, in: 3rd IFAC Symp. on ICV'98, Madrid, Elsevier Science Amsterdam, 1998, pp. 297–302.

[10] J. Gasós, A. Rosetti, Uncertainty representation for mobile robots: perception, modeling and navigation in unknown environments Fuzzy Sets and Systems 107 (1999) 1–24.

[11] F. Gómez-Bravo, F. Cuesta, A. Ollero, Parallel and diagonal parking in nonholonomic autonomous vehicles, Eng. Appl. Artif. Intell. 14 (2001) 419–434.

[12] A. Hemami, Steering control problem formulation of low speed tricycle-model vehicles, Internat. J. Control 61 (4) (1995) 783–790.

[13] J.C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, Dordrecht, 1991.

[14] J. Leonard, H.F. Durrant-Whyte, Directed Sonar Sensing For Mobile Robot Navigation, Kluwer Academic Publishers, Boston, 1992.

[15] R.C. Luo, T.M. Chen, Autonomous mobile target tracking system based on grey-fuzzy control algorithm, IEEE Trans. Ind. Electron. 47 (4) (2000) 920–931.

[16] A. Mandow, J.M. Gómez-de-Gabriel, J.L. Martínez, V.F. Muñoz, A. Ollero, A. García-Cerezo, The autonomous mobile robot AURORA for greenhouse operation, IEEE Robot. Automat. Mag. 3(4) (1996) 18–28.

[17] H.P. Moravec, A. Elfes, High resolution maps from wide angle sonar, in: IEEE Internat. Conf. on Robotics and Automation, 1985, pp. 116–121.

[18] A. Ollero, B.C. Arrue, J. Ferruz, G. Heredia, F. Cuesta, F. López-Pichaco, C. Nogales, Control and perception components for autonomous vehicle guidance. Application to the romeo vehicles, Control Eng. Practice 07 (10) (1999) 1291–1299.

[19] A. Ollero, A. García-Cerezo, J.L. Martínez, A. Mandow, Fuzzy tracking methods for mobile robots, in: Applications of Fuzzy Logic: Towards High Machine Intelligence Quotient, Prentice-Hall, Englewood Cliffs, NJ, 1997. (Chap. 17).

[20] A. Ollero, G. Ulivi, F. Cuesta, Fuzzy logic applications in mobile robotics, in: H.B. Verbruggen, H.-J. Zimmermann, R. Babuska (Eds.), Fuzzy Algorithms for Control, Kluwer Academic Publishers, Boston, 1999, pp. 301–324. (Chap. 11)

[21] G. Oriolo, G. Ulivi, M. Vendittelli, Real-time map building and navigation for autonomous robots in unknown environments, IEEE Trans. Systems Man Cybernet. -Part B 28 (3) (1998) 316–333.

[22] D. Payton, An architecture for reflexive autonomous vehicle control, in: Proc. IEEE Conf. on Robotics Automation, 1986, pp. 1838–1845.

[23] A. Saffiotti, Fuzzy logic in autonomous robot navigation: a case study, Soft Comput. 1 (4) (1997) 180–197.

[24] A. Saffiotti, K. Konolige, E.H. Ruspini, A multivalued logic approach to integrating planning and control, Artif. Intell. 76 (1–2) (1995) 481–526.

[25] K.-T. Song, L.-H. Sheen, Heuristic fuzzy-neuro network and its application to reactive navigation of a mobile robot, Fuzzy Sets and Systems 110 (3) (2000) 331–340.

[26] E. Tunstel, Mobile robot autonomy via hierarchical fuzzy behaviour control, in: 2nd World Automation Congr., Montpellier, 1996.

[27] I. Ulrich, J. Borenstein, VFH+: reliable obstacle avoidance for fast mobile robots, in: Proc. 1998 IEEE Internat. Conf. on Robotics & Automation, Leuven, 1998, pp. 1572–1577.