ICAR ‘95, 7th INTERNATIONAL CONFERENCE ON ADVANCED ROBOTICS, SANT FELIU DE GUIXOLS, SPAIN, PP.367-376, SEPT., 1995

Fuzzy Logic Wall Following of a Mobile Robot Based on the Concept of General Perception

Reinhard Braunstingl* - Pedro Sanz - Jose Manuel Ezkerra
IKERLAN Centro de Investigaciones Tecnológicas
J.M.Arizmendiarrieta, 2 - Aptdo 146, E-20500 Mondragón, Guipúzcoa / España
E-Mail: robot_movil@ikerlan.es

Abstract

This paper presents a new approach to the wall following problem of a mobile robot. Local path planning is based on a so-called concept of general perception, which means that the robot is guided by a representation of its perception only. No map of the environment is used and walls and obstacles are not modelled either. A fuzzy controller then uses the information provided by the concept of general perception to guide the robot along walls of arbitrary shape and around obstacles which are treated as part of a wall, unless the distance between obstacle and wall allows a safe passage. This paper first introduces the concept of general perception and then explains the fuzzy controller in detail. All membership functions and the complete rule base are provided. The concept of general perception together with the fuzzy controller were tested on a real robot performing wall following and obstacle avoidance missions and some of the ensuing experimental results are presented at the end of the paper.

Keywords: Mobile Robot, Wall Following, Obstacle Avoidance, Local Navigation, Perception, Fuzzy Logic.

1 Introduction

One of the basic operations of autonomous mobile robots is their moving along a wall of unknown contours. If a map of the robot´s environment and the possibility of complete localisation exist, there is of course no need for sensing along a wall. The robot could move along a preplanned path in this case simply avoiding unforeseen obstacles. However, in cases where such a map does not or not yet exists, following a wall presents a meaningful method of so-called local path planning. A mobile robot could for example start setting up a map by moving along the circumference of a hitherto unknown space. Also, if the environment is only partially known in more general terms lacking position information, for example: *this corridor leads to that particular door*, the robot can use a wall following strategy to fulfil its mission quickly without having to learn an unknown environment and without having to know where that door is exactly.

Very often wall following missions rely on ultrasonic sensors whereby the measuring data of the sensors are first used to gain a local representation of the environment in order to afterwards control the robot accordingly [7]. In this context we have to distinguish between two fundamentally different types of representation: grid-based representation [1],[2],[6], where the environment is divided into a number of cells which can be occupied or free to a certain degree, and feature-based representation [4],[5], i.e. the environment is modelled by a set of points, lines, and planes. However, both methods need sophisticated treatment of incoming sensor data, as directional resolution of ultrasonic sensors is very poor and

---

* Reinhard Braunstingl is visiting scientist of the Technical University of Graz, Kopernikusgasse 24, A-8010 Graz, Austria. E-Mail: braun@fmechds01.tu-graz.ac.at.

This project was carried out with funding from the European Community Commission for COMETT programmes.

individual measuring data does not allow conclusions to be drown as to the exact position and size of obstacles or the orientation of walls.

The wall following algorithm described in this paper does not need any representation of the environment. Instead, it is based on a representation of perception, the *concept of general perception,* which to our knowledge is something new. This concept uses the information of the ultrasonic sensors to gain a so-called general perception of the closer surroundings. This does not mean the exact position of obstacles or the orientation of walls. On the contrary, the general perception characterises situations in which the robot may find itself in rather an imprecise, but qualitatively appropriate way. It thus lends itself perfectly to descriptions using linguistic terms such as *left front*, *far away*, or *very slow*. Descriptions of this type can be processed using fuzzy logic and that is why a fuzzy logic rule base is introduced here, which - starting from the concept of general perception - realises a wall following mission of a mobile robot.

The extremely simple concept of general perception has been proved very effective in extensive tests with a mobile robot carried out in the robotic department of the Spanish research centre IKERLAN. Interpreting situations by means of general perception makes it possible for the robot to behave as desired in all situations tested (straight and curved walls, sharp and flat corners, narrow corridors, dead ends). Given a range of 0.6 m of the sensors, wall following missions at a speed of 0.45 m/s could be carried out under the following two conditions: firstly, the robot does not have any blind sectors, i.e. it must be able to perceive an obstacle or a wall with at least one sensor, and secondly, all the walls and obstacles are stationary. Obstacles are interpreted as parts of the wall and the robot drives around them or, if the gap between the wall and the obstacle is wide enough, the robot ignores the obstacle.

2 The Mobile Robot

The following considerations are based on a mobile robot with the three degrees of freedom of planar movement, x, y and φ (figure 1). It is equipped with a ring of n ultrasonic sensors which are able to perceive vertical or nearly vertical planes. The number of sensors is irrelevant as long as there are no blind sectors between them. φ refers to the orientation of this ring of sensors and not to the orientation of the robot itself, which is of no importance for the wall following algorithm. With ψ indicating the direction of movement the kinematics model of such a robot is described as follows:

$$\begin{aligned} \mathrm{d}x &= v\cos\psi \; \mathrm{d}t \\ \mathrm{d}y &= v\sin\psi \; \mathrm{d}t \\ \mathrm{d}\varphi &= \dot{\varphi} \; \mathrm{d}t \quad . \end{aligned} \tag{1}$$

Since there is no modelling of the environment the absolute position of the robot does not matter. So there is no world frame used here and the kinematics model can be expressed instead as:

$$\begin{aligned} \mathrm{d}s &= v \; \mathrm{d}t \\ \mathrm{d}\psi &= \dot{\psi} \; \mathrm{d}t \\ \mathrm{d}\varphi &= \dot{\varphi} \; \mathrm{d}t \quad . \end{aligned} \tag{2}$$

The speed v, and the angular speeds $\dot{\psi}$, and $\dot{\varphi}$ are used as control variables of the robot and generated by the fuzzy controller presented in section 4.

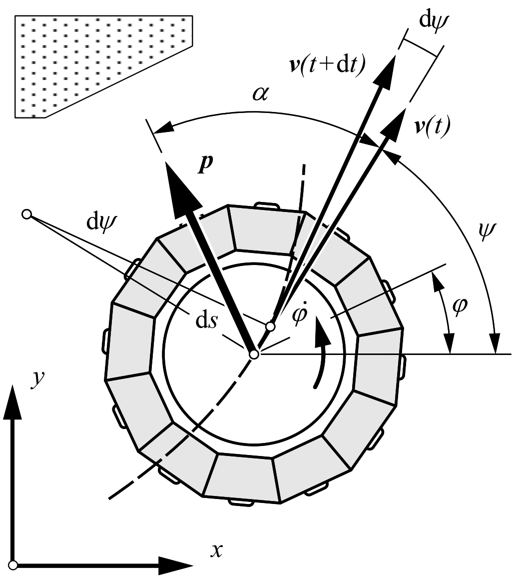

Figure 1: A mobile robot seen from above is moving in the plane. Its perception of any obstacles is represented by the vector of general perception *p* explained in section 3.

3 The Concept of General Perception

It is a well known fact, that ultrasonic sensors have very poor directional resolution. Although these sensors very accurately determine the distance to the nearest object giving back an echo, this object can be anywhere under a certain angle to the sensor´s axis. Moreover, this angle depends on the nature of the object´s surface, the distance and the tilt of the surface with regard to the sensor´s axis. That is why it would be difficult to try to first gain a representation of the immediate surroundings from the sensor data, i.e. to try to model objects or to determine the exact contours of a wall in order to control the robot accordingly. The concept of general perception avoids these difficulties because it does not undertake any kind of modelling of the environment. Instead, it aims at constructing a so-called general perception of the surroundings from the measuring data provided by all the sensors and representing it as a vector, called general perception vector.

For this purpose every ultrasonic sensor i of the mobile robot is assigned a perception vector $\boldsymbol{p}_i$. Its direction equals the orientation of the sensor´s axis and its length is a function of the distance d_i measured by this sensor:

$$p_i = \frac{d_{\max} - d_i}{d_{\max} - d_{\min}} \quad , \tag{3}$$

whereby $d_{\min}$ and $d_{\max}$ designate the shortest and longest distance respectively at which an object may be positioned to be reliably detected. p_i is limited to 0 and 1 respectively so that

$$p_i = \begin{cases} 0 & \text{for } d_i > d_{\max} \\ 1 & \text{for } d_i < d_{\min} \quad . \end{cases} \tag{4}$$

This perception vector is comparable to the obstacle vector of the Vector Field Histogram [1] but is linked to the sensor and not to a cell of a grid. The general perception vector $\boldsymbol{p}$ is composed of all individual perceptions $\boldsymbol{p}_i$. Its direction equals the sum of the perceptions of all the sensors and its length equals the strongest individual perception:

$$\boldsymbol{p} = p_{i,\max} \frac{\sum \boldsymbol{p}_i}{\left|\sum \boldsymbol{p}_i\right|} \quad . \tag{5}$$

The general perception´s change in time is represented by $\dot{p}^*$ and expressed by a scalar. For this purpose the perception´s change in time of a sensor i

$$\dot{p}_i = \frac{\mathrm{d}p_i}{\mathrm{d}t} \approx -\frac{\Delta d_i}{\Delta t\,(d_{\max} - d_{\min})} \tag{6}$$

is related to $\dot{p}_{\max}$, whereby $\dot{p}_{\max} = v_{\max} / (d_{\max} - d_{\min})$ is the perception´s change in time at head-on approach towards an obstacle at maximum speed $v_{\max}$. Moreover, only positive values are to be considered for $\dot{p}_i^*$, thus resulting in the relative perception´s change in time of a sensor i as follows:

$$\dot{p}_i^* = \begin{cases} \dfrac{\dot{p}_i}{\dot{p}_{\max}} = -\dfrac{\Delta d_i}{\Delta t\, v_{\max}} \,, & \text{if } \Delta d_i < 0 \\ 0 & \text{otherwise} \,. \end{cases} \tag{7}$$

The maximum value of all sensors i thus arrived at, is the general perception´s change in time $\dot{p}^*$:

$$\dot{p}^* = \dot{p}^*_{i,\max} \quad . \tag{8}$$

Figure 2 illustrates the concept of general perception. It shows a robot in three typical situations of a wall following mission. The general perception of the corner (figure 2a), for example, is made up of the perceptions of three sensors. The wall to the right of and at a short distance from the robot is perceived by one sensor. Two further sensors are aware of another wall. This perception, however, is less strong because of the big distance. The description of this situation in linguistic terms reads as follows: The general perception is very strong and relative to the path tangent it is to the right and somewhat ahead. The change of the perception is strongly positive if the robot moves at high speed. Another example of a standard situation is the dead end (figure 2b), which the robot enters moving along the right-hand wall. When the robot reaches the end of the dead end, the general perception starts moving further to the front. Its change in time can vary from small to very big depending on the robot´s speed. In the case of the receding corner (figure 2c), for example, the general perception is strong and to the right back.

The concept of general perception is perfectly suited for such a linguistic description of a multitude of situations in which a mobile robot might find itself. A description of

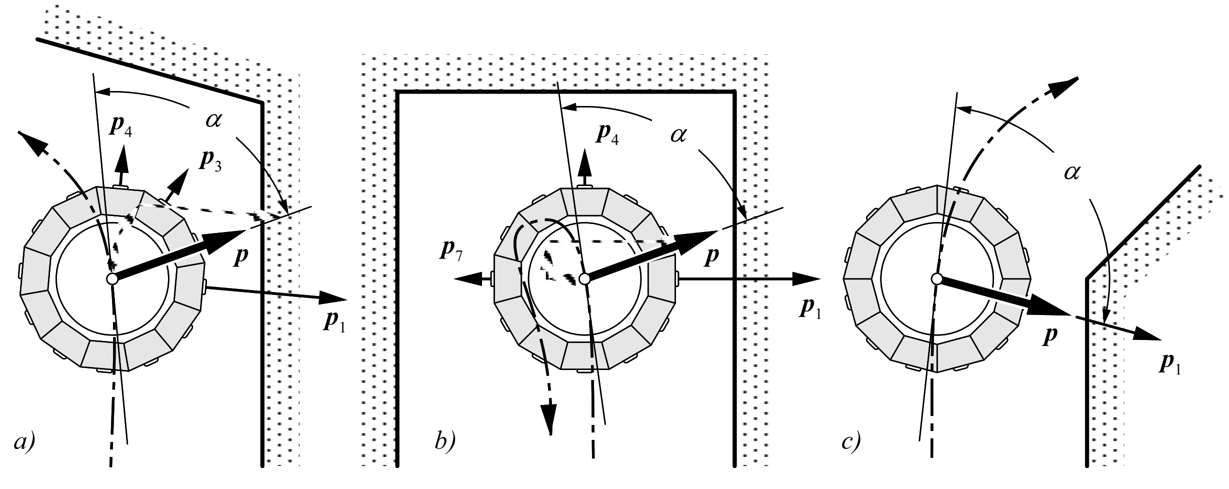

Figure 2: A mobile robot in three typical situations of a wall following mission. The general perception *p* of the corner in figure 2a is constantly moving further to the front the closer the robot is getting to the second wall. The rule base of the fuzzy logic algorithm makes use of this rotation of the vector of general perception in order to turn the robot´s speed vector to the left, too. The same happens at the end of the dead end in figure 2b. The general perception, and therefore the robot, turns to the left until it leaves the dead end along the second wall. In figure 2c one sensor loses contact with the wall at the receding corner. The remaining perception $\boldsymbol{p}_1$ of sensor 1 turns the general perception, which previously was at right angle to the wall , to the right. As a consequence, the robot starts changing its direction of movement to the right, too.

this kind is very simple, at the same time to the point, and is used in exactly this form as input for the rule base of the fuzzy logic wall following algorithm. In this context the orientation of the sensor ring is of subordinate importance. Neither is it necessary that several sensors perceive one and the same wall as was supposed in figure 2c for reasons of clarity. The concept of general perception enables a robot to follow a wall without difficulties, even if the wall is perceived by only one sensor.

Note that the concept of general perception combined with a fuzzy controller resembles, for example, the way a human being would intuitively deal with the situation shown in figure 2a. He would not care too much for the shape or the exact location of the walls in a world model. Just looking at this particular situation he would judge it first by saying that there is something to the right and in front of the robot. This corresponds to the concept of general perception. Then he would advise the robot to turn more or less to the left, perhaps to brake, depending on the robot´s speed and on how near that "something" is. All that is done by the fuzzy controller.

One main aspect of the concept of general perception is the fact that it is composed of the perception of all sensors, hence the name *general perception*. Even those sensors, which from the point of view of the direction of movement are facing backwards and whose perception might at first sight seem irrelevant, contribute to the general perception to the same degree. Using the example of a narrow corridor (figure 3) it can be shown that leaving out the backward-facing sensors would lead to a general perception characterising the situation in a completely wrong way. The robot is moving down a very narrow corridor, the walls of which are perceived by two sensors each. If the data of the backward-facing sensors 1 and 8 are not taken into consideration the angle α' between the general perception vector $\boldsymbol{p}'$ and the speed $\boldsymbol{v}$ is less than 90°. Or expressed in linguistic terms again: the general perception is located somewhat left to the front. As a consequence the robot would move to the right and distance itself from the wall. However, the closer it gets to the opposite wall the more $\boldsymbol{p}'$ turns to the right because $\boldsymbol{p}_7$ is getting smaller and $\boldsymbol{p}_2$ bigger. As a result the robot starts swinging from wall to wall. Actually, the wall which the robot is to follow is located somewhat to the left behind seen from the direction of movement, which is expressed in just this way by the general perception $\boldsymbol{p}$, which also takes into account the perceptions of sensors 1 and 8.

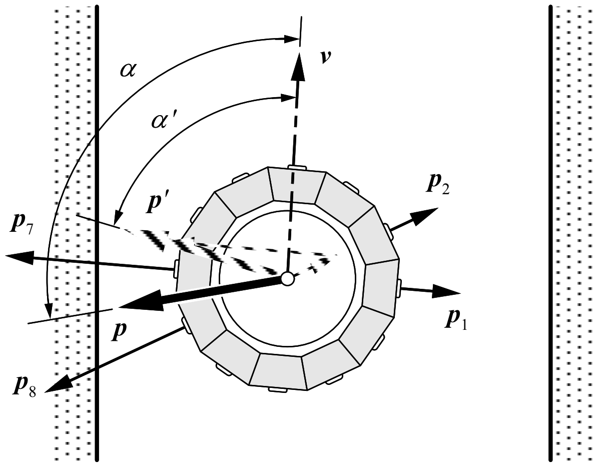

Figure 3: The backward-facing sensors have to be taken into consideration as well. In a narrow corridor neglecting sensors 1 and 8 would lead to a general perception *p´* and the robot would start swinging to and from between the two walls.

4 The Fuzzy Controller

The fuzzy controller was designed under the two assumptions that there are, firstly, no moving walls or obstacles and, secondly, no blind sectors between the sensors. This means that a wall is always perceived by at least one of the sensors no matter what the orientation of the robot is like. The first assumption does not represent a limitation caused by the concept of general perception. It was made in order to test this concept in combination with a fuzzy controller in principle. The second assumption can be met by a sufficiently large number of sensors.

4.1 Inputs

The fuzzy controller consists of a rule base of altogether 33 rules, which represent instructions to the robot regarding its behaviour in certain situations. Input values (figure 1) are the angle α between the general perception vector and the robot´s speed, the intensity of the general perception p (equation 5) and its change in time $\dot{p}^*$ (equation 8).

The robot classifies a situation using the general perception described in the previous section. For this purpose α, p, and $\dot{p}^*$ are named perception_angle, perception, and perception_change and regarded as fuzzy variables. Such variables are described using linguistic terms called adjectives. If perception_angle, for example, is 45°, a possible linguistic description reads LEFT_FRONT. At 60° perception_angle will still be LEFT_FRONT but to a lesser extent. The degree to which the adjective LEFT_FRONT applies to an angle represents its membership function. Figures 4a, 4b, and 4c show the membership functions for all the adjectives of the three input values of the rule base.

Perception_angle can be situated in four sectors which overlap between FRONT and BACK but not between LEFT and RIGHT. When approaching a wall or an obstacle almost head-on, a clear reaction either to the right or to the left is desirable, therefore the strict division. Five adjectives, which in part strongly overlap, are used to describe the intensity of the general perception. Limiting perception_change to positive values proves effective when rounding a corner because the biggest change in perception at leaving the corner is caused by the wall the robot is moving away from. At that instant this particular wall and as a consequence the perception thereof is less important than the wall to the side of the robot or, for example, an obstacle the robot is heading towards right behind the corner.

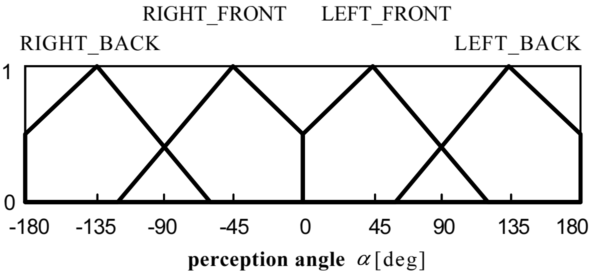

Figure 4a: The general perception can be located in four sectors, which partly overlap. There is a marked division between left and right, however, in order to achieve a clear reaction at head-on approach to a wall.

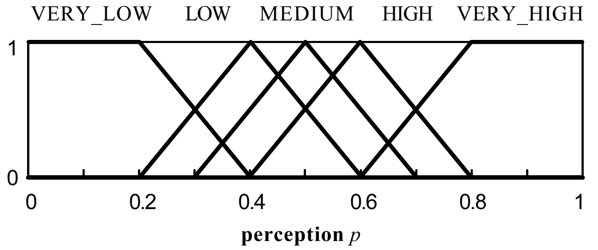

Figure 4b: Five adjectives, which strongly overlap, describe the intensity of the general perception.

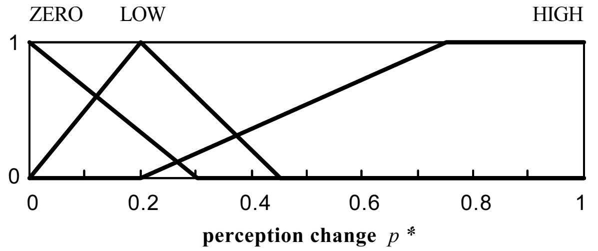

Figure 4c: Only the positive change in time of the general perception (approaching a wall) is taken into consideration .

4.2 Outputs

The output values of the rule base, turn, steer, and acceleration are also fuzzy variables. The numerical values arrived at for control instructions leading to changes in the orientation $\dot{\varphi}$, as well as the direction of and value for speed $\dot{\psi}$ and $\dot{v}$ respectively, depend on the inference and defuzzyfication strategies. The changing of orientation (turn) is described using two adjectives for each direction. Using the slow turning motion LITTLE the robot will orientate itself gradually in such a way that the vector of general perception is at right angle to the wall. Fast turning was introduced as additional adjective as practical tests had shown that the mobile robot had difficulties perceiving sharp corners due to the small number of sensors and lost contact to the corner at larger distance. This turning serves the purpose of directing one sensor as quickly as possible to the assumed position of the corner in case of total loss of perception at a receding corner. Changes of direction (steer) can also be done slowly or fast (HARD). An extra adjective, CENTER, is used for straight ahead movement. Finally only a few - all together four - possibilities were chosen for the changing of speed.

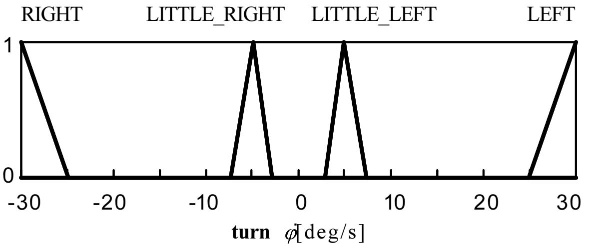

Figure 5a: The robot is able to change the sensor ring´s orientation slowly (to direct the vector of general perception) or fast (to recover lost perception).

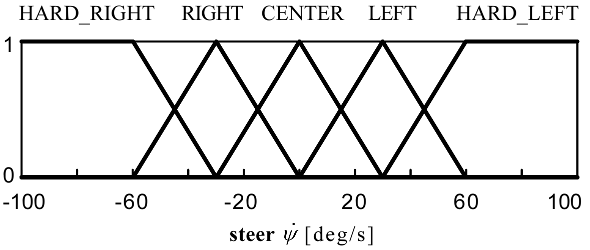

Figure 5b: There are five adjectives available to describe a change in direction. Two for speed for each side and one for straight movement.

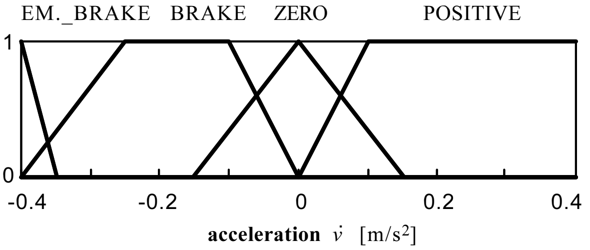

Figure 5c: Four adjectives were chosen to describe the robot´s acceleration.

4.3 Rule Base

The rule base is composed of 33 rules and is subdivided into three groups: rules for change of orientation $\dot{\varphi}$, change of direction $\dot{\psi}$, and change of speed $\dot{v}$, which use two state variables each as input. The rules are formulated as in classical logics:

IF <condition> THEN <consequence>

In contrast to Boolean logics, however, the condition may also be only partially fulfilled. Therefore the consequence of this rule will also be applied only partially. An ‘AND’ operator combines more than one antecedent in a rule via minimum, an ‘OR’ via maximum. In case that several rules are valid simultaneously a defined control command has to be generated using contradictory rule outputs. There are several possibilities to achieve this. In our case so-called correlation-product encoding with sum combination and centroid defuzzification [3] is used, which is to be briefly illustrated by means of the following simplified example from the rule base under 4.3.2 taking into account perception_angle only. The assumption reads: perception_angle -70°, and two rules to be applied are:

1) `IF perception_angle IS RIGHT_FRONT THEN MAKE steer LEFT;`
2) `IF perception_angle IS RIGHT_BACK THEN MAKE steer HARD_RIGHT;`

Under *correlation-product encoding* (figure 6) every consequence of a rule is scaled with the degree of activation. In our example the angle -70° is seen as 67% RIGHT_FRONT and therefore the fuzzy set or the membership function of the rule output LEFT is multiplied by the factor 0.67. The same is done with the output of the second rule HARD_RIGHT, which gets multiplied by 0.1 as the value -70° at the same time belongs to 10% to the fuzzy-set RIGHT_BACK. An unequivocal control command is generated from the two contradictory commands steer LEFT and steer HARD_RIGHT by first adding the two scaled fuzzy-sets (sum combination). Then the coordinate for the centre of gravity of the resulting area defines the so-called defuzzyfied value for steer (centroid defuzzification).

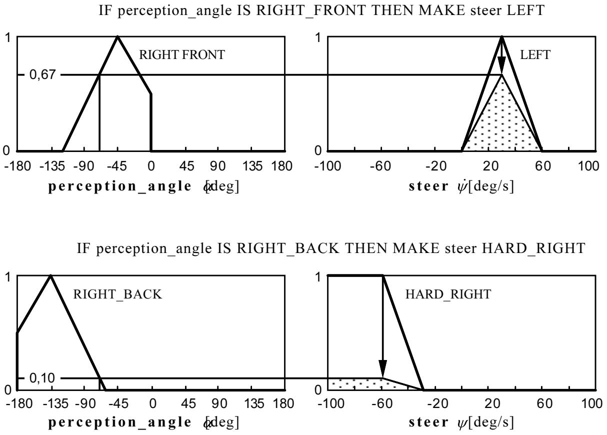

Figure 6: *Correlation product encoding*. The degree of the membership of the rule input shows the value with which the fuzzy-set of the output is scaled. The x-coordinate of the common centre of gravity of the two hatched areas defines the value of the output variable steer (*centroid defuzzyfication*).

4.3.1 Control of Orientation

This rule base consisting of only six rules has two input values, α and p, and $\dot{\varphi}$ as output. It enables the robot to keep the desired distance MEDIUM to the wall more precisely and to preserve perception more securely. With a small number of sensors the concept of general perception describes only very imprecisely such a simple situation as moving along a straight wall. The direction of the robot´s movement remaining unchanged, a slight change in the orientation of the sensor ring would result in a change to the same amount of the perception angle α. This would lead to undesired control instructions as the situation the robot is in has not changed. For example, if α is not exactly 90°, the rule base described under 4.3.2 will change the direction of speed and therefore the distance to the wall, even if the robot is moving at the exactly correct distance parallel to the wall. That is why the main aim of the rule base is to slowly turn the sensor ring so that the general perception $\boldsymbol{p}$ is at right angle to the speed. If the robot moves along a straight wall, this means that $\boldsymbol{p}$ is normal to the wall and the robot is able to precisely keep its distance. The total loss of perception already mentioned, which was observed when passing round a sharp corner, is to be kept as short as possible by a fast turning of the sensor ring so that one sensor is quickly directed at the presumed position of the corner. The six rules involved are shown in table 1a.

α p	**RB**	**RF**	**LF**	**LB**	**RB** *or* **RF**	**LB** *or* **LF**
---	LL	LR	LL	LR	---	---
VL	---	---	---	---	L	R

Table 1a: The rule base controlling the orientation of the robot via turning speed $\dot{\varphi}$.

4.3.2 Directional Control

This part of the rule base also uses α and p as input values and yields $\dot{\psi}$ as output. Starting from the concept that the speed of moving along the wall is parallel to the wall if the length of the vector of general perception remains constant, the rules try to keep the speed vector normal to the vector of general perception and the general perception at the value MEDIUM. This is achieved with the following twenty rules:

p α	**VL**	**L**	**M**	**H**	**VH**
RB	HR	HR	R	R	C
RF	C	L	L	HL	HL
LF	C	R	R	HR	HR
LB	HL	HL	L	L	C

Table 1b: The rule base controlling the robot´s direction of movement via steering speed $\dot{\psi}$.

Although total loss of perception must not happen according to the conditions laid down, it was supposed that the robot was to react with an orbital movement in such a case, in order to be able to pass sharp corners. However, it is a lot easier to achieve this reaction without fuzzy logic $(\dot{\psi} = \pm v/r)$ and it, therefore, is not provided by the rule base.

4.3.3 Speed Control

This rule base of seven rules, shown in table 1c, with the inputs p and $\dot{p}^*$ is guided by the idea that the robot needs minimum braking and maximum acceleration the farther it is away from the wall and the smaller the change of general perception in time. Fast changes of general perception on the other hand inevitably result in emergency braking. Speed is limited upward by $v_{\max} = 0.45\,\mathrm{m/s}$ and downward by $v_{\min}$, which was set as follows:

$$v_{\min} = \mathrm{Min}(0.05\,, 1 - p) \quad .$$

p $\dot{p}^*$	**VL** *or* **VH**	**L** *or* **H**	**ME**	---
ZE	ZE	P	P	---
L	EB	B	Z	---
H	---	---	---	EB

Table 1c: The rule base controlling the speed of the robot via acceleration $\dot{v}$.

5 Experimental Results

5.1 Wall Following

The mobile robot VEA-1 (Vehículo Experimental Autónomo, figure 7) developed by the Spanish research centre IKERLAN was used to test the concept of general perception under realistic conditions. This robot is of cylindre-shaped build, has a radius of 0.3 m, a height of 1.5 m, and is equipped with 12 ultrasonic sensors with a range of 0.6 m, which are arranged on its circumference. Its theoretical maximum speed given the possibility of omnidirectional movement is 1 m/s. So far it has not been possible to go at this speed due to the short range of the sensors. The 12 sensors of VEA-1 represent the minimum number because at sharp corners and with very smooth walls (glass) blind sectors of an approximate angle of aperture of 5° appeared at a larger distance which only diminished at very short distance to the wall. Therefore, maximum speed had to be limited to 0.25 m/s for safety reasons in an environment with such walls. An environment with sufficiently coarse walls, however, permitted a maximum speed of 0.45 m/s.

Figure 7: The mobile robot VEA-1, developed by the Spanish research center IKERLAN.

Figure 8 shows a tour along an interrupted wall, around corners and along a dead end so small that the robot simultaneously perceives the two walls on either side. The concept of general perception proved a secure means of local navigation for the robot in all situations involved and the robot passed the course without difficulties at an average speed of 0.21m/s when the maximum speed allowed was 0.25 m/s. The walls of this course are sufficiently coarse to prevent blind sectors between the sensors. In this case higher speed is possible. The dotted line shows a tour at a maximum speed of 0.45 m/s. This time the average speed is not much higher (0.23 m/s) because the robot spends more time rounding the corners where it always brakes strongly. As the robot does not take an absolute definition of its position, the paths were reconstructed using the data from the incremental decoders. As the experiment was carried out within a relatively short span of time, it was possible to neglect effects of accumulating errors. Afterwards the paths were

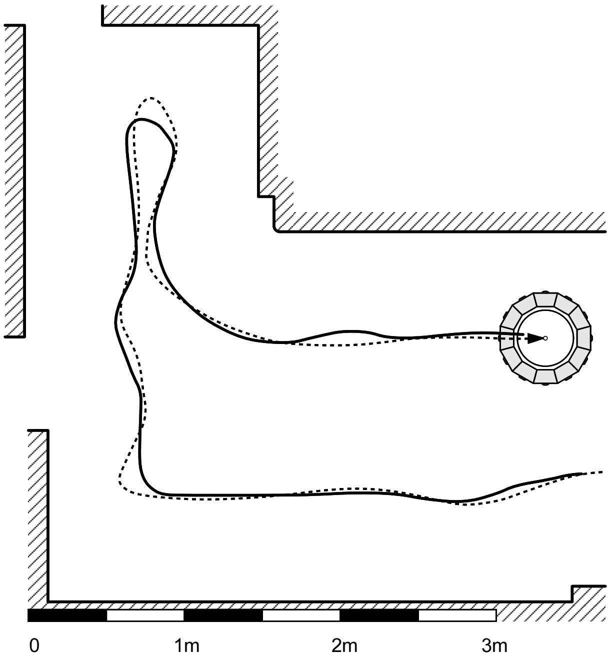

Figure 8: The robot passes this test course with corners, narrow gaps in the wall and a dead end at a maximum speed of 0.25 m/s and 0.45 m/s (dotted line). The dead end is so narrow that the robot perceives the walls on either side. The concept of general perception permits the correct interpretation of all these situations.

fitted into the layout of the course using the smallest measured distance in the turning points.

In figure 9 the robot encounters an obstacle. If the distance between the wall and the obstacle is too small for the robot to pass through, the obstacle is seen as part of the wall and circumvented (Situation A). If the distance is big enough, the robot passes the obstacle with a slight deviation from its path (Situation C). The obstacle in position B leaves a relatively large gap which, nevertheless, is too small for passing through. In this situation the robot loses orientation. As expected, it does not try to pass through the gap but starts going around the obstacle. However, it approaches the obstacle too fast so that the vector of general perception changes from LEFT_FRONT to RIGHT_FRONT. The robot therefore changes the wall it follows. The situation then is interpreted correctly as a dead end which it leaves by turning left and moving along the same path it came in. Situations like this one, however, do not represent a principal failure of the concept of general perception as the robot does not move around the obstacle for the sole reason that the vector of general perception turns around faster than the robot is able to follow, which would not happen with a controller optimised correspondingly, at lower travelling speed, or with sensors with bigger range.

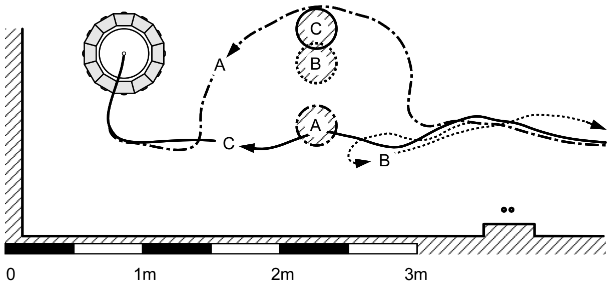

Figure 9: An obstacle at three different distances from the wall. In Position A the obstacle is seen as part of the wall and passed around. In Position C the distance between the obstacle and the wall is big enough for the robot to pass through and the robot does so, being slightly irritated by the obstacle. In Position B the gap is quite big but too small to be passed through. The situation is interpreted as a dead end and the robot drives back. The robot moves at an average speed of 0.19 m/s (A), (B), and 0.21 m/s (C).

5.2 Obstacle Avoidance

The wall following method described is not only useful to execute an explicit instruction such as “follow that wall”. It’s also used to avoid an unexpected obstacle in a predefined movement or mission. The mobile robot VEA-1 can receive by radio from a *Central Station* plans composed by elemental movements (EM). The *Central Station* is provided by a planner which generates plans from high level missions. These plans have information about the expected minimum distance to the known walls in each EM. While the robot is moving, unexpected obstacles or walls can appear and avoiding them is desired and then continue executing the rest of the plan. Taking all that into account the problem of the obstacle avoidance could be reduced to three main aspects:

1) When to start to avoid an unexpected obstacle.
2) How to avoid the obstacle.
3) When and how to finish the avoidance and continue the rest of the mission.

5.2.1 Start to avoid an unexpected obstacle.

This part has been simplified to the robot by the planner. The planner makes the calculations to obtain the minimum distance between each particular movement in the known environment. The avoidance begins when one sensor detects an object nearer than the distance given by the planner.

5.2.2 How to avoid the obstacle.

The avoidance of the obstacle consists of following the contour of the obstacle in the same way that has been explained before. The maximum speed of the following process will be the speed of the EM that was in execution when the obstacle has been detected. That speed has been calculated as the maximum safe speed in the region of the environment by the planner.

5.2.3 Finish the avoidance of the obstacle.

That part of the avoidance is the most complex part because of the multiple possibilities of movements and reasons for the finishing.

The avoidance can finish:

a) When the robot gets back to one of the EMs of the plan. (Main case).
b) When a long time has elapsed from the beginning of the avoidance. (The obstacle covers all the rest of mission).
c) If the robot is very far from the point of the beginning of the avoidance. (The robot could go very far from its goal in the mission).

The cases (b) and (c) are easy to detect but the case (a) depends on the types of the movements of the robot in the mission. It’s important to know that all of the calculations to detect the end of the avoidance have to be made as fast as possible to get the maximum time free in the CPU for the rest of processes. (Position control, radio communications, avoidance, etc.). Then all of the types of movements possible are reduced to segments of lines and circumference’s arcs. In the first case the robot looks for intersection between the segments and a rectangle around the current position of the centre of robot. In the second case it looks for the distance between centres and angles. The robot looks only in some of the EMs of the plan transmitted from the *Central Station* if the plan has too many Ems.

The method has been tested in the real robot VEA-1 developed at IKERLAN and the results are represented in the figure 10 and figure 11.

In figure 10 the robot is executing one segment from A to D but when it is at point B it detects with one sensor an unexpected obstacle. Then the robot follows its contour until point C, where it detects that it is near one EM of the plan and continues executing the plan linking up with a segment from C to D with a previous turn in the direction of movement at point C. The test was repeated with 5 consecutive segment lines, the first begins at A and the last finishes at D and the result was very similar.

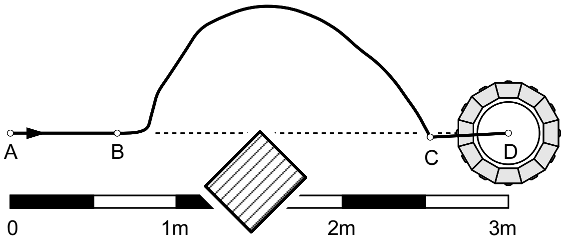

Figure 10: The predefined trajectory is the segment from A to D or various different segments beginning the first at point A and finishing the last at D. New plans are developed from the end of avoidance at C to D. From B to C the robot follows the contour of the unexpected obstacle with a maximum speed of 0.25m/s and a medium speed of 0.17m/s.

In figure 11 the robot was executing a circumference arc from A to D, it detects the obstacle at B and finds the rest of the plan at C. Then the robot generates a new arc from C to D. When the robot is at C is near to the EM but not exactly in it and the arc has to finish exactly at D, the new arc has new parameters but is generated going from C to D and with the same direction at D that it had in the previous plan to link to the next plan without interruption.

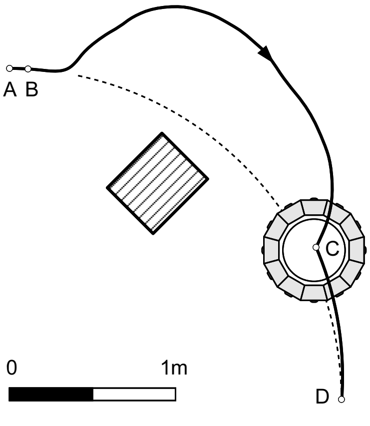

Figure 11: The predefined trajectory is the arc from A to D or various different arcs beginning the first at point A and finishing the last in D with the same radius. New plans are developed from the end of avoidance in C to D. From B to C the robot follows the contour of the unexpected obstacle with a maximum speed of 0.25m/s and a medium speed of 0.15m/s.

Then the robot is able to execute plans until it detects an unexpected obstacle, follows its contour and, if possible, comes back to one of the future EMs of the plan and continues producing link-up plans.

References

[1] Johann Borenstein, and Yoram Koren, The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots, *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, p.278-288, June, 1991.

[2] Rüdiger Dillmann, Jürgen Kreuzinger, Frank Wallner, PRIAMOS: An experimental platform for relfexive navigation, *Robotics and Autonomous Systems*, Vol. 11, Numbers 3-4, p.195-203, Dec., 1993.

[3] Bart Kosko, *Neural Networks and Fuzzy Systems*, Prentice-Hall, New Jersey, 1992.

[4] Ben J.A. Kröse, Kai M. Compagner, Franciscus C.A. Groen, Accurate estimation of environment parameters from ultrasonic data, *Robotics and Autonomous Systems*, Vol. 11, Numbers 3-4, p.221-230, Dec., 1993.

[5] John J. Leonard, Hugh F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Publishers, Boston, 1992.

[6] Hans P. Moravec, Alberto Elfes, High Resolution Maps from Wide Angle Sonar, *IEEE International Conference on Robotics and Automation*, p.116-121, March, 1985.

[7] P. van Turennout, G. Honderd, L.J. van Schelven, Wall-following control of a Mobile Robot, *IEEE International Conference on Robotics and Automation*, p.280-285, May, 1992.