Autonomous Robots 20, 43–59, 2006
© 2006 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
DOI: 10.1007/s10514-006-5363-5

Abstracting Vehicle Shape and Kinematic Constraints from Obstacle Avoidance Methods

JAVIER MINGUEZ AND LUIS MONTANO
Instituto de Investigación en Ingeniería de Aragón, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain
jminguez@unizar.es
montano@unizar.es

JOSÉ SANTOS-VICTOR
Instituto Superior Técnico, Instituto de Sistemas e Robótica, Lisboa, Portugal
jasv@isr.ist.utl.pt

Published online: 3 February 2006

Abstract. Most obstacle avoidance techniques do not take into account vehicle shape and kinematic constraints. They assume a punctual and omnidirectional vehicle and thus they are doomed to rely on approximations when used on real vehicles. Our main contribution is a framework to consider shape and kinematics together in an exact manner in the obstacle avoidance process, by abstracting these constraints from the avoidance method usage. Our approach can be applied to many non-holonomic vehicles with arbitrary shape.

For these vehicles, the configuration space is three-dimensional, while the control space is two-dimensional. The main idea is to construct (centred on the robot at any time) the two-dimensional manifold of the configuration space that is defined by elementary circular paths. This manifold contains all the configurations that can be attained at each step of the obstacle avoidance and is thus general for all methods. Another important contribution of the paper is the exact calculus of the obstacle representation in this manifold for any robot shape (i.e. the configuration regions in collision). Finally, we propose a change of coordinates of this manifold so that the elementary paths become straight lines. Therefore, the three-dimensional obstacle avoidance problem with kinematic constraints is transformed into the simple obstacle avoidance problem for a point moving in a two-dimensional space without any kinematic restriction (the usual approximation in obstacle avoidance). Thus, existing avoidance techniques become applicable.

The relevance of this proposal is to improve the domain of applicability of a wide range of obstacle avoidance methods. We validated the technique by integrating two avoidance methods in our framework and performing tests in the real robot.

Keywords: obstacle avoidance, collision avoidance, sensor-based navigation, mobile robots

1. Introduction

In order to endow vehicles with true versatility, they must execute tasks autonomously in unknown, unstructured, dynamic and unpredictable environments. Under these circumstances, motion must be generated by an obstacle avoidance method driven by sensory information. An obstacle avoidance method is a procedure that, given a sensorial measurement (obstacle description) and a final position, calculates a collision free motion towards a target. It works within a perception - action cycle where the motion is executed by the vehicle and the process restarts (Fig. 1). The result is an on-line motion sequence that drives the vehicle to the target while avoiding collisions. The avoidance task is further complicated since many robots have shape and kinematic constraints that limit motion.

Our work addresses the question of taking into account the vehicle shape and kinematics in the obstacle avoidance paradigm. It is noteworthy that we are not proposing another obstacle avoidance method. We go a step further by proposing a methodology to encompass such constraints in an exact way in the avoidance paradigm, which can be applicable to many existing methods without any redesign. This issue is important because if the vehicle shape is ignored in the avoidance problem, collisions will inevitably occur. In addition, if the kinematics are ignored, the vehicle may not

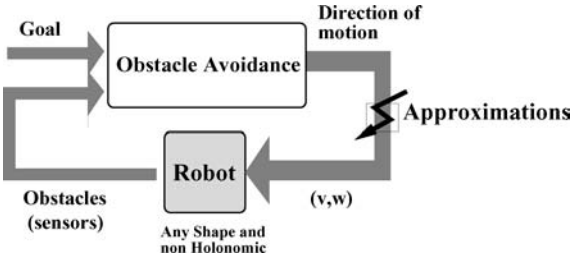

Figure 1. This figure shows the operation diagram of an obstacle avoidance method. At high frequency the process is as follows: the sensors gather information about the obstacles that is processed by the method to compute a collision free motion that drives the vehicle towards the target. The motion is executed and the process restarts.

be able to execute the computed motions. In both cases, the security of the task would be at risk, which is especially important when vehicles perform tasks in dangerous or hostile surroundings that could affect human safety.

In this work we analyze the system that avoids collisions with obstacles detected by sensors. This functionality is only a subgroup of the complete mobility problem. Other aspects involve perception, motion planning, modelling and control. They will not be addressed here, but are essential to construct a complete navigation system. Related works include motion planning (Latombe, 1991); localization and map building (Castellanos et al., 1999; Leonard and Feder, 2000; Dissanayake et al., 2001; Thrun et al., 2000); and supervision (Buhmann et al., 1995; Koenig and Simmons, 1998; Morisset and Gallab, 2002).

1.1. Path Planning versus Obstacle Avoidance

Classically, the mobility problem has been addressed by computing a geometric path, free of collisions with obstacles (Latombe, 1991). Nevertheless, when the surroundings are unknown and unpredictable, these techniques fail, since a pre-computed path would almost certainly hit obstacles. Reactive obstacle avoidance is an alternative way to compute motion by introducing sensory information within the control loop (Fig. 1). The main cost of considering the world state during execution is locality. For this reason, a trap situation may arise whenever global reasoning is required. Despite this limitation, obstacle avoidance techniques are mandatory to deal with mobility problems in unknown and dynamic surroundings.

1.2. Shape and Kinematics in Obstacle Avoidance

In obstacle avoidance, there is no procedure to consider exactly the vehicle shape and kinematics jointly. The avoidance problem with these constraints has been addressed from two points of view: (i) by computing a set of collision-free admissible motions and selecting next one of them. In a first stage, these strategies compute a set of admissible commands (or admissible paths) that are collision free and allow for stopping safely. Next, one command (or elementary path) is selected with obstacle avoidance and convergence to the target criteria. (ii) By applying an avoidance method first, and then turning the solution into an admissible motion free of collisions. These strategies use the obstacle avoidance as the heuristic to compute the motion direction, and then compute the collision-free admissible command that better aligns the robot heading with this direction.

In the first class of methods, some authors solve the problem in the control space (Fox et al., 1997; Simmons, 1996) to compute the set of admissible commands. At each step, they compute a set of collision-free commands, after which one of them is selected with an optimization process that favours progress, security and convergence towards the target. The elegance and simplicity of these methods have lead to extensions and applications in different contexts (Ogren and Leonard, 2002; Brock and Khatib, 1999; Arras et al., 2002; Ko and Simmons, 1998). In addition, some techniques pre-compute a set of arcs of circle (elementary paths), and choose one based on obstacle avoidance and convergence towards the destination (Ulrich and Borenstein, 2000; Hebert et al., 1997; Feiten et al., 1994; Hait et al., 1999). Both types of techniques compute motions that are admissible for the vehicles and consider shape. However, in these techniques there is usually a discretization of the space of solutions to compute an approximation of the set of collision-free motions (paths), and depending on the shape of vehicle it could be necessarily to use a numerical method or a dynamic simulation to check collisions. Another point is that it is difficult to extrapolate these strategies to allow classical methods for considering the constraints.

A second class of techniques converts the solution of an obstacle avoidance planner to admissible motions in every period. In Luca and Oriolo (1994) and Bemporad et al. (1996) the output of the obstacle avoidance method is modified by a feedback action that aligns the vehicle with the direction solution in a least squares sense. Although the vehicle shape is taken into account during the application of the avoidance method, the motion is modified to comply with the kinematics. The final motion is kinematically admissible but does not guarantee collision avoidance with the exact shape. In Minguez and Montano (2002) a similar solution is proposed by dividing the problem into subproblems (motion, kinematics and shape). First, the obstacle avoidance method is used. Next the direction solution is converted to admissible commands using a motion generator (Montano and Asensio, 1997; Asensio and Montano, 2002). Finally the commands are modified, if necessary, verifying collisions by dynamic simulation. The advantage of these strategies is their generality, since they can be used by many avoidance methods. However, they do not consider the vehicle constraints together during the obstacle avoidance

method stage, which would lead to difficulties in situations where the holonomic solution cannot be approximated, or when manoeuvrability is a determining factor.

1.3. Our Approach and Contributions

Most obstacle avoidance techniques do not consider the constraints mentioned previously. They assume a punctual and omnidirectional vehicle and are doomed to rely on approximations. Our main contribution is a framework to consider shape and kinematics together in an exact manner, in the obstacle avoidance process, by abstracting these constraints from the avoidance method usage. Our approach can be applied to many non holonomic vehicles with arbitrary shape (in the paper we focus on a differential-drive).

For these vehicles, the configuration space is three-dimensional (Lozano-Perez, 1983), while the control space is two-dimensional. The main idea is to construct (centred on the robot at any time) the two-dimensional manifold of the configuration space that is defined by elementary circular paths. This manifold contains all the configurations that can be attained at each step of the obstacle avoidance and is thus general for all methods. Another important contribution of the paper is the exact calculus of the obstacle representation in this manifold for any robot shape (i.e. the configuration regions in collision). Finally, we propose a change of coordinates of this manifold in such a way that the elementary paths become straight lines. Therefore, the three-dimensional obstacle avoidance problem with kinematic constraints is transformed into the simple obstacle avoidance problem of a point moving in a two-dimensional space without any kinematic restriction (the usual approximation in obstacle avoidance). Thus, many existing avoidance techniques become applicable.

With our approach, many existing or future obstacle avoidance methods can be applicable to non holonomic vehicles with arbitrary shape, without any redesign (Fig. 2). For example, our approach can be used to extend techniques such as potential field methods (Khatib, 1986; Krogh and Thorpe, 1986; Tilove, 1990; Borenstein and Koren, 1989), the family of Vector Field Histogram (Borenstein and Koren, 1991; Ulrich and Borenstein, 1998), or the Nearness Diagram (Minguez and Montano, 2004; Minguez et al., 2004). In this work we present results with a potential field method (Khatib, 1986) and the Nearness Diagram (Minguez and Montano, 2004).

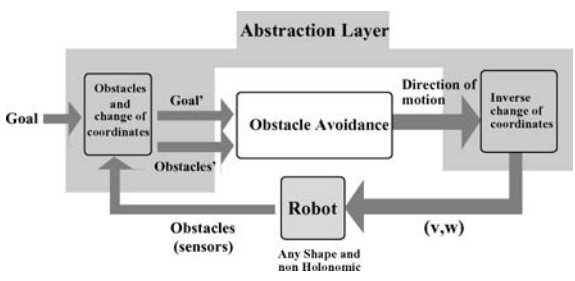

Figure 2. In this work we abstract the shape and kinematics of the vehicle from the avoidance method. The idea is to understand the method as a "black-box" and modify the representation of its inputs so that they implicitly have information about these restrictions. The method is applied naturally, however its solutions consider the restrictions (the method is "unaware" of it).

The manuscript is distributed as follows. We describe in Section 2 the computation of the manifold of the configuration space that contains all the configurations reachable by admissible paths. In Sections 3 and 4 we show how to compute the collision regions in this manifold and a change of coordinates that turns the motions omnidirectional. We discuss the complete abstraction layer in Section 5. In Section 6 we describe the experimental results and in Section 7 we discuss and draw the conclusions of our work.

2. Admissible Paths in Obstacle Avoidance

In this section we discuss how the kinematic structure of the vehicles considered here imply that all admissible paths must be arcs of circle. As a consequence, we will show that the vehicle configurations are constrained on a two dimensional manifold of the configuration space.

We focus our attention on differential-drive robots (unicycle model) moving on a flat surface, where the Workspace $\mathcal{W}$ and the Configuration space $\mathcal{C}$ are $\mathbb{R}^2$ and $\mathbb{R}^2 \times S^1$, respectively. A configuration q is the location and the orientation $q = (x, y, \theta)$. The motion of these robots is constrained by:

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \tag{1}$$

Equation (1) is a non holonomic motion constraint. The effect is to reduce the dimension of the motion space, in each configuration (Laumond et al., 1998). Therefore, a motion command can be described by two parameters only. The kinematic model of these vehicles can be expressed by:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} w \tag{2}$$

where v and w denote the linear and angular velocities.[1] During the execution of a motion, we assume that velocities remain constant in each control period. Then, the vehicle moves along a circular path or a straight line (see Fox et al. (1997) to characterize this assumption). All the methods discussed in the previous section use this motion assumption. Notice that a straight motion and a pure rotation are both circular paths with infinity and zero radii respectively.

We characterize next a feasible circular motion. In the robot system of reference, an admissible circular path contains the origin, and the instantaneous turning centre is on the Y-axis (Fig. 3(a)). Then, if (x, y) is a

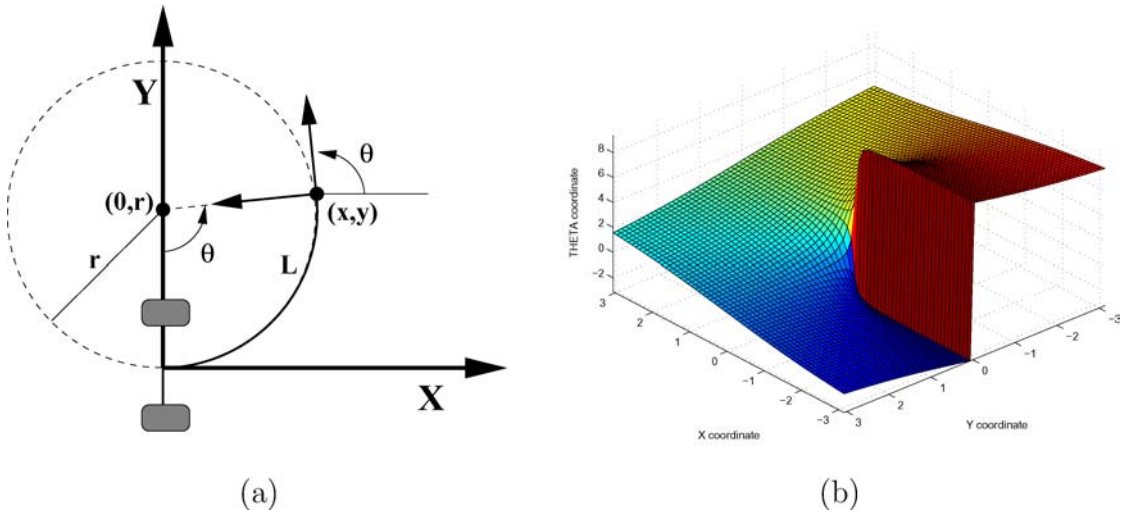

Figure 3. (a) Given a point of $\mathbb{R}^2$, there is only one circle that complies with the motion constraints. Furthermore, the robot orientation is tangent to the circle at every point; (b) the surface in the configuration space corresponds to the manifold *ARM* [where $q_o = (0, 0, 0)$]. This surface represents all the configurations reached by circular paths from q_o.

point in the workspace, there is only one circle going through $\{(x, y), (0, 0)\}$ and having its centre in the Y-axis. The radius of that circle is:

$$r = \frac{x^2 + y^2}{2y} \tag{3}$$

In other words, the instantaneous turning centre is $(0, r)$. Furthermore, the robot orientation has to be tangent to the circle at all points:

$$\theta = \begin{cases} \text{atan2}(x, r-y) & \text{if} \quad y \geq 0 \\ -\text{atan2}x, y-r & \text{otherwise} \end{cases} \tag{4}$$

From Eqs. (3) and (4) we have,

$$\theta = f(x, y) = \begin{cases} \text{atan2}\left(x, \dfrac{x^2 - y^2}{2y}\right) & \text{if} \quad y \geq 0 \\ -\text{atan2}\left(x, -\dfrac{x^2 - y^2}{2y}\right) & \text{otherwise} \end{cases} \tag{5}$$

It is easy to see that function f is differentiable in $\mathbb{R}^2 \backslash (0, 0)$. Thus $(x, y, f(x, y))$ defines a two dimensional manifold in $\mathbb{R}^2 \times \mathcal{S}^1$ when $(x, y) \in \mathbb{R}^2 \backslash (0, 0)$. We call this manifold Arc Reachable Manifold, *ARM* (q_o), since it contains all the configurations attainable by elementary circular paths from the current robot configuration q_o (Fig. 3(b)). Notice that $ARM(q_o)$ contains the configurations attainable at each step of the obstacle avoidance. For simplicity of notation we adopt $ARM \equiv ARM(q_o)$.

3. Calculus of the Robot Configurations in Collision

In the previous section we discussed how the attainable space of the configuration space $\mathcal{C}$ is constrained to a manifold *ARM* in the case of circular motion. In this section we describe an algorithm to compute the region of configurations of this manifold in collision for any vehicle shape. In the following development, obstacles are considered as a set of points (e.g. laser, see Figs. 13 and 15).

The region that contains all the configurations in collision is the intersection between the $\mathcal{C}$-Obstacle boundary (configurations in collision) and *ARM* (configurations attainable by circles). To compute it, we derive first a procedure to determine a point of the collision region boundary, given an obstacle point and a point of the robot boundary. Then, by using this procedure over a parameterization of the robot boundary, we can analytically describe the boundary of the collision region for an arbitrary vehicle shape. The calculus is derived in $\mathbb{R}^2$ since it is the domain of *ARM*.

Let be $p_f = (x_f, y_f)$ and $p_i = (x_i, y_i)$ an obstacle point and a point in the robot boundary. We want to determine the robot location $p_s = (x_s, y_s)$ over a circular path such that p_i and p_f coincide (i.e. collision occurs). The location p_s represents the contribution of p_i to the collision region boundary in *ARM*. The calculus is based on the existence of a circle $\mathcal{D}$ with radius r such that $(0, 0) \in \mathcal{D}$, the centre of $\mathcal{D}$ lies on the Y-axis, and in location $(x_s, y_s) \in \mathcal{D}$ the points p_i and p_f coincide (Fig. 4). Then:

$$(x_s, y_s) = (r \sin\theta, r(1 - \cos\theta)) \tag{6}$$

In order to solve this equation, we need to compute r and θ. We start by computing the turning radius r. The distance from the rotation center $p_r = (0, r)$ to p_i and to p_f has to be equal:

$$\|p_i - p_r\|^2 = \|p_f - p_r\|^2 \tag{7}$$

$$x_i^2 + (r - y_i)^2 = x_f^2 + (r - y_f)^2 \tag{8}$$

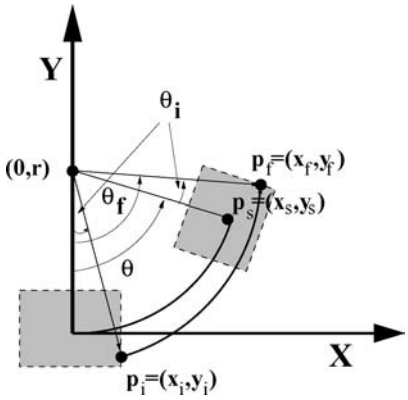

Figure 4. This figure shows how the motion over a circle leads a point of the robot boundary (x_i, y_i) to the obstacle point (x_f, y_f), in the robot system of reference. The robot location (x_s, y_s) represents the limit between collision and no collision, that is, the boundary of C-Obstacle.

to find the value of r:

$$r = \frac{\left(y_f^2 - y_i^2\right) + \left(x_f^2 - x_i^2\right)}{2(y_f - y_i)} \tag{9}$$

Secondly, we compute the rotation θ, as:

$$\theta = \theta_f - \theta_i \tag{10}$$

where θ_f denotes the angle towards the obstacle point and θ_i the angle of the robot boundary point both w.r.t. the Y-axis. To solve Eq. (6), we have to calculate $\sin\theta$ and $(1 - \cos\theta)$. From Eq. (10) we have:

$$\begin{aligned} \sin\theta &= -\frac{x_f(y_i - r) - (y_f - r)x_i}{x_f^2 + (y_f - r)^2} \\ 1 - \cos\theta &= \frac{x_f(x_f - x_i) + (y_f - r)(y_f - y_i)}{x_f^2 + (y_f - r)^2} \end{aligned} \tag{11}$$

Finally, replacing these expressions in Eq. (6) we obtain the final solution.

$$\begin{aligned} x_s &= \frac{(x_f + x_i)\cdot\left[\left(y_f^2 - y_i^2\right) + \left(x_f^2 - x_i^2\right)\right]\cdot\left[(y_f - y_i)^2 + (x_f - x_i)^2\right]}{(y_f - y_i)^4 + 2\left(x_f^2 + x_i^2\right)(y_f - y_i)^2 + \left(x_f^2 - x_i^2\right)^2} \\ y_s &= \frac{(y_f - y_i)\cdot\left[\left(y_f^2 - y_i^2\right) + \left(x_f^2 - x_i^2\right)\right]\cdot\left[(y_f - y_i)^2 + (x_f - x_i)^2\right]}{(y_f - y_i)^4 + 2\left(x_f^2 + x_i^2\right)(y_f - y_i)^2 + \left(x_f^2 - x_i^2\right)^2} \end{aligned} \tag{12}$$

For a given obstacle point p_f and a point of the robot boundary p_i, we obtain a point p_s of the boundary of the collision region in ARM.

Notice that this result can be used to express analytically the bounds of the collision region of ARM by substituting (x_i, y_i) in the previous equation by a parametric expression of the vehicle bounds. In other words, for an arbitrary robot shape one can compute the analytical expression of the collision regions in the configuration space reachable by circular paths. We denote the collision region CO_{ARM}. We describe next an example of this calculus for a cardioid-shaped vehicle and for a polygonal robot. Lets suppose that we have a "cardiod"-shaped robot (Fig. 5(a)), whose boundary is described by:

$$\begin{cases} x_i(\lambda) = a \cdot (1 + \cos\lambda) \\ y_i(\lambda) = \lambda \end{cases} \tag{13}$$

with $\lambda \in [0, \pi]$. Replacing this equation in Eq. (12) we obtain the CO^1_{ARM} corresponding to one obstacle point O_1. The obstacle region is $CO_{ARM} = \cup_i CO^i_{ARM}$ for all obstacle points O_i (Fig. 5).

In the case of a polygonal robot, we represent each segment by its parametric equation. Let $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ be the coordinates of one of the segments of the robot boundary. We parameterize the segment:

$$\begin{cases} x_i(\lambda) = x_1 + (x_2 - x_1)\cdot\lambda \\ y_i(\lambda). = y_1 + (y_2 - y_1)\cdot\lambda \end{cases} \tag{14}$$

where $\lambda \in [0, 1]$. Replacing this equation in Eq. (12) we obtain the transformation of a segment. By using this transformation for all the segments of the robot we get CO_{ARM} corresponding to the obstacle point. Figure 6 shows an example.

Let be g the piece-wise function that describes the robot boundary. The complexity of the calculus described is $N \times M$, where N is the number of obstacle points and M the number of pieces of function g. For instance, $M = 1$ for a circular robot or the cardioid-shaped robot, or M is equal to the number of sides for a polygonal robot. Notice that we compute the exact representation of the region in collision for any vehicle shape. Furthermore, the region of collision can always be computed as long as g exists (the robot boundary can be mathematically described by curves).

Summarizing, in this section we have described a procedure to compute exactly the boundary of the collision regions, for arbitrarily shaped robots, that move on circular paths. We have given an example of a cardioid-shaped robot and a polygonal robot. The calculus is exact, can be always applied and is not restricted to a given vehicle shape.

The avoidance problem is now transformed to a point moving in a two dimensional space ARM, but still with kinematic constraints. In the next section, we propose a change of coordinates of ARM so that the circular motions are described by straight motions (free of kinematic constraints).

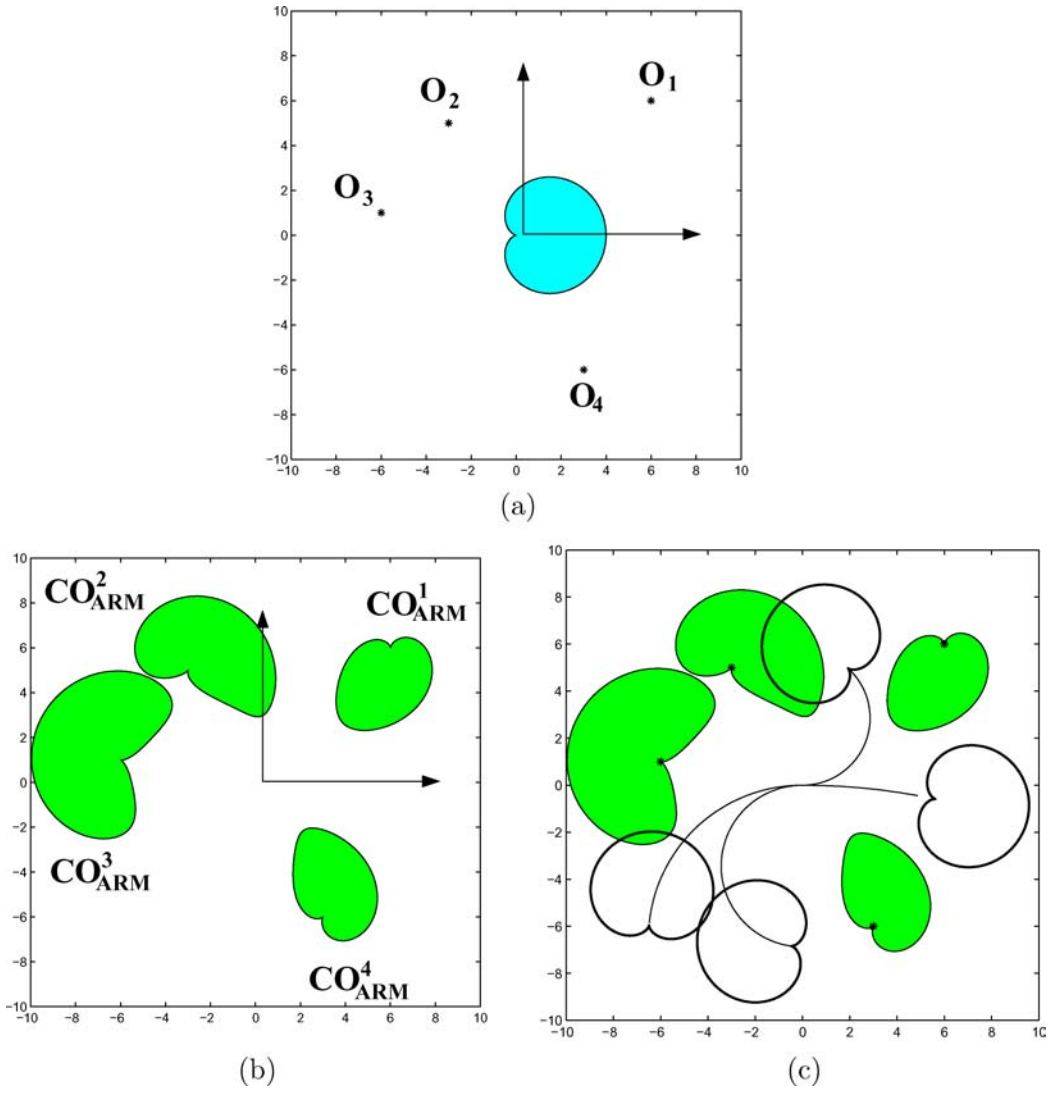

Figure 5. This figure shows the computation of the free space for a "cardioid"-shaped robot that moves over circular paths. (a) Robot and obstacles O_i; (b) each obstacle point creates a region of collision locations CO^i_{ARM} that all together are CO_{ARM}. The free space is the space outside these regions and all locations within these regions are in collision; and (c) superposition of both the workspace and the *ARM*, and some robot locations and the paths that lead to them. Notice how locations out of the CO_{ARM} are not in collision with the obstacle points.

4. The Ego-Kinematic Coordinate Transformation

We have discussed how the admissible elementary paths of the vehicles considered are circles. We identified the manifold *ARM* of the configuration space, as a function $\{\mathbb{R}^2 \backslash (0,0)\}$, which represents all the configurations reachable under circular motions. We also provided a calculus to compute the exact bounds of the collision region CO_{ARM} on this manifold. In this section we propose a change of coordinates of *ARM* so that elementary paths become straight segments with the new coordinates.

The Ego-Kinematic change of coordinates transforms the domain of the manifold $\mathbb{R}^2$ into $\mathbb{R} \times S^1$,

$$\begin{aligned} \mathbb{R}^2 &\rightarrow \mathbb{R}_0^+ \times \mathcal{S}^1 \\ (x, y) &\rightarrow (L, \alpha) \end{aligned}$$

where the distance to a point is the arc length L measured over the circle that reaches that point, and the angle univocally represents this circle (Fig. 7). Next, we discuss the computation of both coordinates.

In the robot system of reference, the radius r of the circle that goes through point (x, y) is given by Eq. (3) and the vehicle orientation θ in that point by Eq. (4). The distance to the point measured along the circle is the arc length:

$$\begin{aligned} L &= \begin{cases} |x|, \; y = 0 \\ |r \cdot \theta|, \; y \neq 0 \end{cases} \\ &= \begin{cases} |x|, & y = 0 \\ \left| \dfrac{x^2 + y^2}{2y} \operatorname{atan2}(2xy, x^2 - y^2) \right|, & y \neq 0 \end{cases} \end{aligned} \tag{16}$$

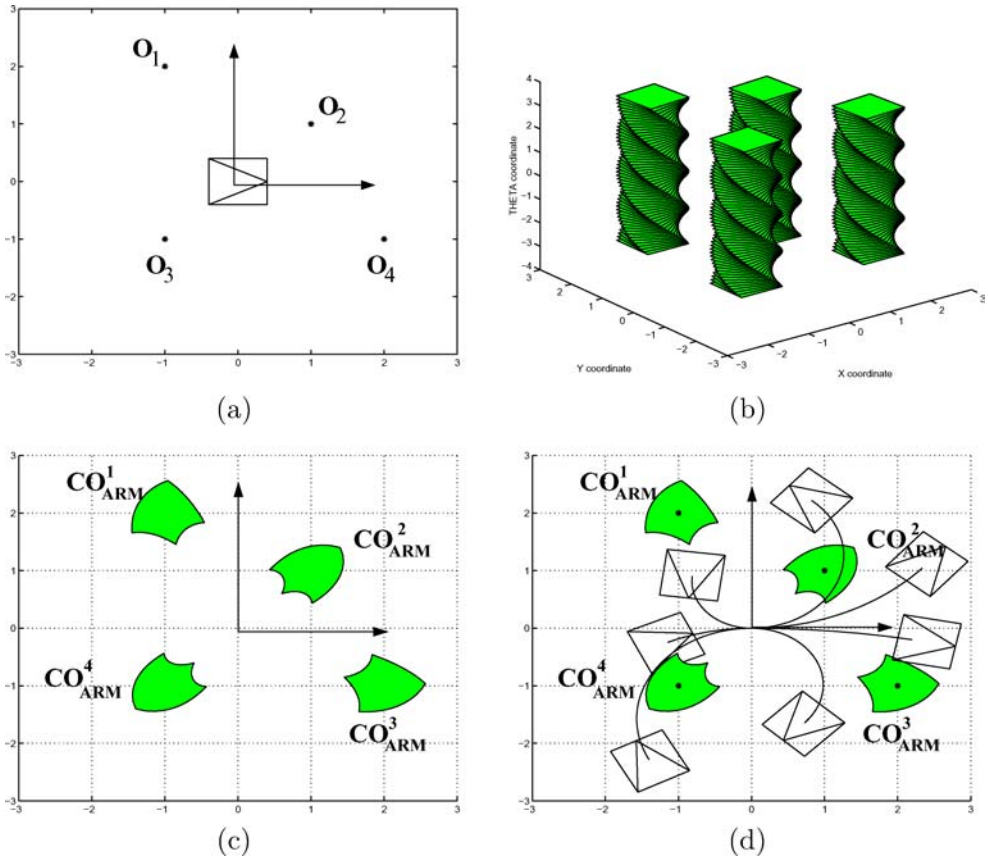

Figure 6. This figure shows the computation of the free space for a rectangular robot that moves over circular paths. (a) Robot and obstacles O_i; (b) the C-Obstacles in the configuration space that result from the four obstacle points. These volumes contain all the configurations in collision with the obstacles; (c) each obstacle point creates a region of collision locations CO^i_{ARM} that all together are CO_{ARM}. The free space is the space outside these regions. Graphically, these regions are the inter-section between the volumes in (b) and the surface of figure 3(b). (d) Superposition of both the workspace and the *ARM*, and some robot locations and the paths that lead to them.

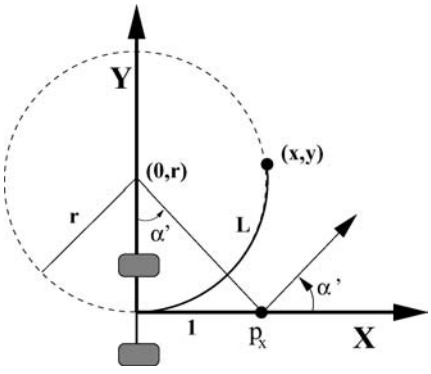

Figure 7. This figure shows how a differential-drive vehicle reaches a point of the space (x, y) by a circular path (of radius r). On a point of the X-axis, the angle α is tangent to the circle.

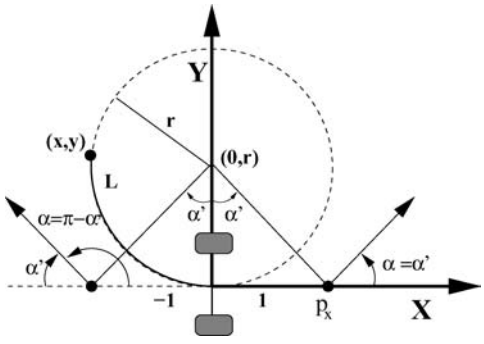

Figure 8. This figure shows how, where the point is located in the positive X-axis, the value of α is α'. If the point is in the negative axis, the problem is symmetrical with respect to the Y-axis.

This distance is the first coordinate. The second coordinate has to identify the circle univocally and give the sense of travel. The turning radius r is a unique descriptor of the circle going through a point and that complies with the motion constraints (Eq. (3)). However, this descriptor is unbounded while we search a bounded representation. This is achieved through an angular variable, constructed as follows. Let p_x be a point in the X-axis (for example the (0, 1)).[2] Let $\mathcal{T}$ be the line joining $(0, r)$ and p_x. Then, α' is the angle comprised between the perpendicular line to $\mathcal{T}$ and the X-axis:

$$\alpha' = \arctan\left(\frac{1}{r}\right) \tag{17}$$

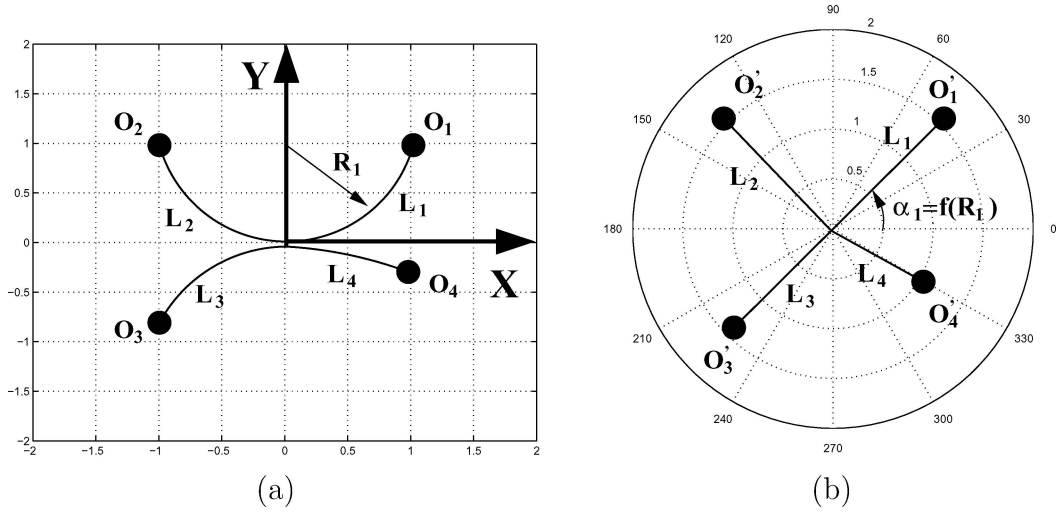

Figure 9. This figure shows how we express *ARM* in the new coordinates. (a) Four points in $\mathbb{R}^2$ and circular paths that go through them. (b) The points represented in the new coordinates. Notice that to reach a point we need a motion direction and a distance (rectilinear motion), whereas they fix a turning radius and the arc length in workspace.

Given a turning radius, we obtain a tangent direction α' (a bounded descriptor of the circle that goes through a point). By using Eq. (3), we get:

$$\alpha' = \arctan\left(\frac{2y}{x^2+y^2}\right) \tag{18}$$

The second coordinate is then:

$$\alpha = \begin{cases} \alpha', & x \geq 0 \\ \text{sign}(y)\cdot\pi - \alpha', & x < 0 \end{cases} \tag{19}$$

where α' is given by Eq. (18). This definition implies that when $x \geq 0$ the direction of travel is "forward" (the angle α is equal to α', Fig. 7), and when $x < 0$ the motion is "backward" (α is the result of the same calculus assuming the symmetrical problem with respect to the Y-axis, Fig. 8). Notice that each value of α_s univocally determines a turning radius, r_s:

$$r_s = \begin{cases} \dfrac{1}{\tan(\alpha_s)}, & \alpha_s \in \left[-\dfrac{\pi}{2}, \dfrac{\pi}{2}\right] \\ \dfrac{1}{\tan(\text{sign}(\alpha_s).\pi - \alpha_s))}, & \text{otherwise} \end{cases} \tag{20}$$

The coordinate α distinguishes the direction of travel: $\cos\alpha_s \geq 0$ is "forward" motion while the opposite is "backward" (although r and α do not differentiate the direction).

With this parameterization of the domain of *ARM*, the coordinates of a point depend on the distance measured along the admissible path (the arc length of the circle L), and on a descriptor of the circular path that reaches the point (since a describes one turning radius r). We call ARM^P to *ARM* in the new coordinates.

Figure 9 shows an example of the Ego-Kinematic coordinate transformation. On the left there are four points O_i and the corresponding circular paths. Points are represented in ARM^P as O_i' (Fig. 9(b)), so that they are reached by rectilinear motions. In other words, to reach a point of ARM^P we require a direction a (which fixes a turning radius, by Eq. (20)) and a distance L (the arc length). Thus, the admissible paths in ARM^P are rectilinear (omnidirectional motion), whereas they represent circular admissible paths in *ARM* (admissible paths in the workspace).

In summary, we represent *ARM* in a new coordinate system where the motion is omnidirectional (without kinematic constraints) whereas it represents a motion over an admissible path for the robot. In the next section we use the previous results related to robot shape and kinematics to construct the abstraction layer.

5. Abstraction of the Shape and Kinematic Constraints from the Obstacle Avoidance Method

In this section we use the previous results to abstract the shape and the kinematics of the vehicle from obstacle avoidance methods. These techniques work within a cycle, computing on-line collision free motion given a description of the obstacles and a destination. The motion is executed by the vehicle and the process restarts (Fig. 1). The idea is to build an abstraction layer so that the solutions computed consider the shape and the motion constraints of the vehicle without redesigning the method (Fig. 2). This is achieved by including two stages: (i) incorporate the shape and the kinematics before the method application and (ii) motion computation. At each iteration the procedure is:

1 *Shape:* Construction of the region in collision with the obstacles CO_{ARM} (procedure described in Section 3).

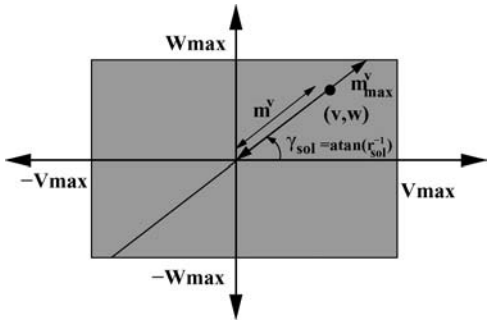

Figure 10. This figure shows how to compute the motion command (v, w) within the physical limits (maximum velocities v_{max} and w_{max}) given a turning radius r_{sol}. The module of the speed vector m^v is reduced linearly with the distance to the closest obstacle (when there are obstacles closer than a given distance to the robot).

2 *Kinematics:* Change of coordinates of the collision region CO_{ARM} to CO_{ARM^P}, (procedure described in Section 4).
3 *Obstacle avoidance:* Application of the obstacle avoidance method in ARM^P, to compute the most promising motion direction α_{sol}.
4 *Motion:* the direction solution α_{sol} is transformed into a motion command (v, w) as follows. First, we compute the radius solution r_{sol} by using Eq. (20). Then, we compute a command (v, w) that preserves the turning radius $v = w.r_{sol}$. Any command on the line with slope r_{sol} (Fig. 10) is valid. One strategy to select one command is to reduce the module of the speed vector m^v as a function of the distance to the closest obstacle:

$$m^v = \begin{cases} m^v_{max}, & d_{obs} \geq d_{min} \\ m^v_{max} \cdot \dfrac{d_{obs}}{d_{min}}, & \text{otherwise} \end{cases} \tag{21}$$

where m^v_{max} is the distance from the velocity origin to the bound of the rectangle of maximum velocities, d_{obs} is the distance to the closest obstacle in ARM^P, and d_{min} is a distance threshold to check whether the velocity is maximum. The final command is:

$$(v, w) = \begin{cases} (m^v \cdot \cos\gamma_{sol}, m^v \cdot \sin\gamma_{sol}) & |\alpha_{sol}| \in \left[-\dfrac{\pi}{2}, \dfrac{\pi}{2}\right] \\ (-m^v \cdot \cos\gamma_{sol}, m^v \cdot \sin\gamma_{sol}) & \text{otherwise} \end{cases} \tag{22}$$

where $\gamma_{sol} = \arctan(r^{-1}_{sol})$. The sign preserves the "forward" and "backward" motion. Finally, the command (v, w) is executed by the vehicle and the process restarts.

Figure 11 illustrates an example by using a rectangular and differential-drive robot, and a generic obstacle avoidance method that considers neither the shape (assumes that the vehicle is a point) nor the motion constraints (assumes that the vehicle is omnidirectional). The figure shows an iteration of the process described. At a given moment, the robot gathers information about the obstacles and the target (Fig. 11(a)). The objective is to compute a motion command that avoids collisions, while leading the robot towards the destination.

Stage 1: Current sensory information is used to construct the region in collision given the shape of the robot in ARM, i.e. the CO_{ARM} (Fig. 11(b)). In this manifold the robot is a point.
Stage 2: The Ego-Kinematic coordinate transformation is applied to the domain of the manifold (Fig. 11(c)), so that in ARM^P the robot is represented by a point and the motions are over straight segments (applicability conditions of the obstacle avoidance method).
Stage 3: The avoidance method is used to obtain the motion direction that avoids the obstacles while moving the robot towards the destination.
Stage 4: That direction is used to compute the movement, which corresponds to a circular motion in the workspace (Fig. 11(a)). The motion is kinematically admissible and takes into account the exact shape by construction. The vehicle executes the motion command and the process restarts.

Notice that, with this methodology, the modifications introduced in the scheme are the calculus of the new obstacle regions and a coordinate transformation. For that reason the method performs the task "being unaware" that it is applied in a representation where the solutions consider the shape and the kinematics of the vehicle. This is the key aspect of this approach.

In the following section we applied this scheme to two obstacle avoidance methods on a real robot.

6. Experimental Results

The objective of this section is to validate our methodology with two obstacle avoidance methods working on a real vehicle with shape (square) and kinematic constraints (differential-drive). First we describe the vehicle, the sensor and the obstacle avoidance methods, and finally the experimental results.

6.1. *Vehicle, Sensors and Obstacle Avoidance Methods*

The vehicle is a *Labmate* available at the University of Zaragoza, Spain. This robot is square (0.8 × 0.8 m) differential-drive (Fig. 12). In order to collect information about the obstacles, the vehicle was equipped with a 3*D* TRC laser, with a maximum range of 6.5 m, a precision of 0.025 m, and a field of view of 240 degrees (a point by

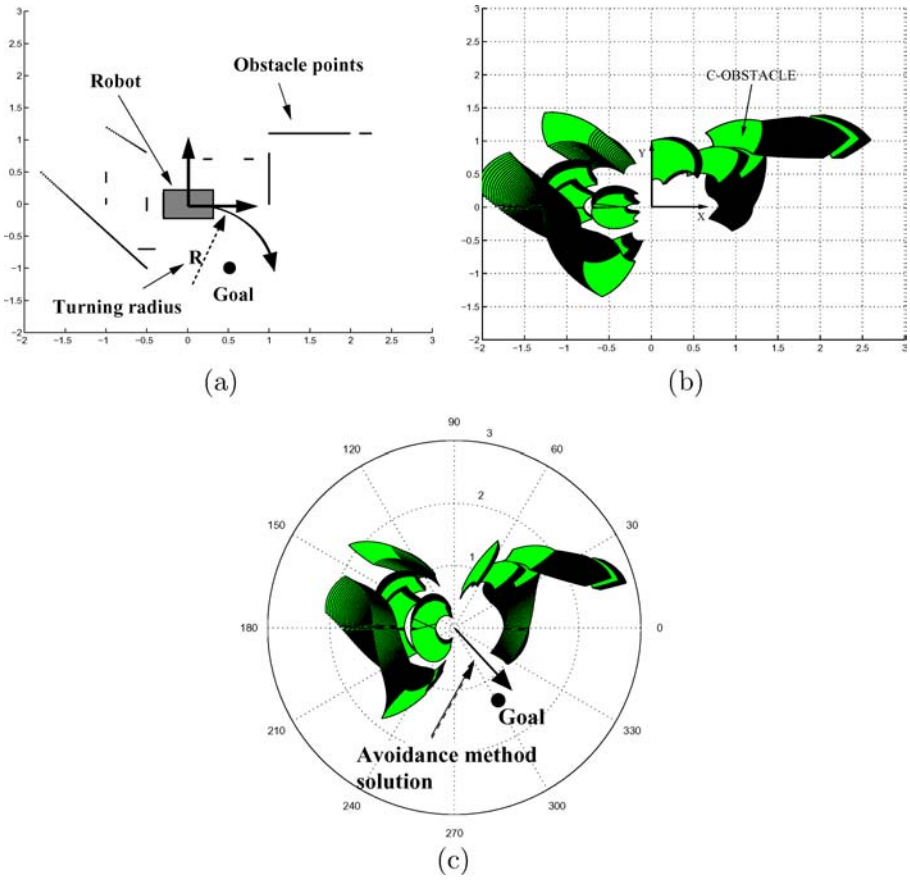

Figure 11. Iteration of the obstacle avoidance process. A sensory description of the obstacles (a) is used to compute the region of the configuration space in collision with the robot (rectangular shape) over the manifold of circular paths, (b), where the robot is represented by a point. The coordinates of the manifold are changed (c) so that the motion is free of kinematic constraints and the robot is a point. Here we use the avoidance method that computes the best motion direction "Avoidance method solution", which represents an arc of circle in the workspace ["Turning radius" in figure (a)].

degree). All the calculations were carried out on a microSun SparcII 60 MHz.

In all the experiments, the environments were completely unknown, dynamic with a priori unpredictable behaviour (people moving) and unstructured (obstacles such as cardboard, people, chairs, tables and boxes). Under these circumstances, an obstacle avoidance method is the right choice to move the vehicle.

We selected two obstacle avoidance methods, a potential field method (Khatib, 1986) (very formal and well known (Koren and Borenstein, 1991)) and the nearness diagram navigation (Minguez and Montano, 2004; Minguez et al., 2004), and heuristic method.

In the potential field method (PFM in short) the robot is modelled as a particle moving in the configuration space, affected by forces created by a potential field. The target position creates a potential that attracts the particle, whilst the obstacles create a repulsive potential. The movement is computed to follow the direction of the artificial force resulting from the sum of both potentials (most promising direction of motion). This method (in the obstacle avoidance version) cannot be applied on the differential robot without approximations, since the direction of the potential gradient does not comply with the non holonomic motion constraint (Eq. (1)). In other words, the structure of the potential does not represent the fact that in the configuration space not all motions are allowed. On the other hand, considering shape would imply constructing a representation of the obstacles in the three-dimensional configuration space, which would be difficult to carry out in real time.

The nearness diagram method (ND in short) is based on the situated-activity paradigm of behavioural design (see (Arkin, 1999) for a review). A set of situations describes the problem and how to act in each case (actions). The situations represent abstractions of all possible configurations between the robot, obstacles and destination. For each case, there is an associated action represented by a motion law. During the execution phase (knowing the location of obstacles, vehicle and destination), one of these situations is identified, and the corresponding action is executed to compute the motion. The advantage of this method is that it uses a "divide and conquer" strategy based on situations to simplify the difficulty of navigation. Therefore, good results have been obtained in difficult scenarios such as very clut-

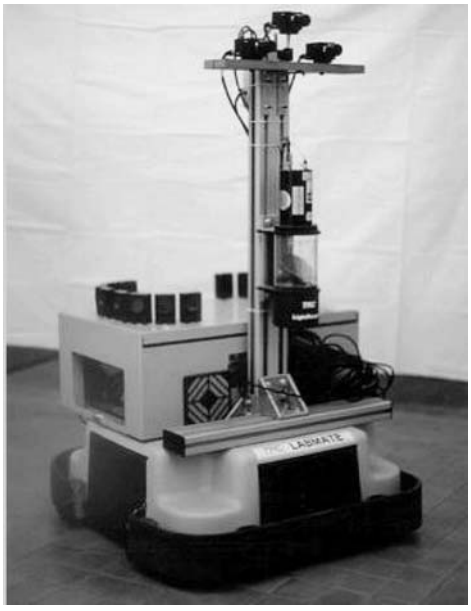

Figure 12. The vehicle is square with differential traction and equipped with a laser $3D$ TRC.

tered situations that make it hard to manoeuvre the vehicle. This method cannot be directly used on the considered robot since it assumes that the robot is a point (or circular), and calculates the most promising motion direction (assuming that the robot is holonomous, analogue to the PFM).

Both methods assume that the robot is a point (they ignore the shape restriction) and that can move in any direction (no kinematic constraint). When used on this type of vehicle, shape and motion are approximated. In the next section we show how we have applied both methods, without approximations, using the proposed methodology.

6.2. *Experiments*

In the experiments we start by discussing how our methodology computes motions that carry out the avoidance task while driving the vehicle to the destination (the main task is not penalized by including the abstraction layer). In addition, we show the improved robustness when taking into account the constraints. Finally, we discuss that, when both methods are used without the abstraction layer, the solutions cannot be executed without approximations (very strong in some cases).

For the potential field method (PFM), we fixed the sampling period to 0.250 sec, since it represents the upper bound of the time to compute both, the CO_{ARM^P} and the solution of the avoidance method, for each sensorial measurement (laser). With this cycle time, we obtained fast reactions to the sensor information. We set the maximum velocities to $(v_{\max}, w_{\max}) = (0.3\frac{\text{m}}{\text{sec}}, 0.45\frac{\text{rd}}{\text{sec}})$.

Figure 13 shows one of the experiments. The robot was driven to the destination avoiding collisions with all the unexpected obstacles. This included objects placed in random locations by a human (see the trajectory followed and the points detected by the laser in Fig. 13a, and some snapshots of the motion in Figs. 13(b)–(e)). The robot reached the destination in 99 s.

In all the experiments the shape of the vehicle was considered during the avoidance, since the potential is applied to avoid the collision region of the configuration space CO_{ARM}. As a consequence, in some places the vehicle manoeuvred in relatively dense surroundings. The kinematics of the vehicle were taken into account during the whole experiment. This is because the potential was applied in ARM^P to calculate the best motion direction, which fixes a turning radius and a command (v, w) that preserves it at every moment. The turning radii computed during the experiment are depicted in Fig. 14(a), and the velocity profiles of the experiment in Fig. 14(b).

We show specific moments of the experiment in Figs. 13(f)–(i) with the current sensorial measurements and the circle that corresponds to the radius (these moments correspond to the snapshots of Figs. 13(b)–(e) respectively). Notice how the motion on these circles always avoids (in the short term, i.e. during the sampling period 0.25 sec) collisions (with the exact shape of the robot). The global effect of the kinematic constraints is in the whole trajectory, which is composed of arcs and straight segments (Fig. 13(a)).

When the PFM is used but without the abstraction layer, the computed solutions are directions of motion that cannot be executed by the robot, since the admissible paths are arcs of circle ("PFM direction" in the Figs. 13(f)–(i)). Although in some cases the motions could be approximated, in general this is not possible (in Figs. 13(f),(h) the directions are lateral, which cannot be executed by a differential robot).

The ND method was also integrated in the methodology. The sampling period was 0.15 sec with the same robot setup.[3] However, with this method there was an additional constraint because it only computes forward motion.[4] Due to that constraint we used a change of coordinates for vehicles that only move forwards.[5]

Figure 15 shows one of the experiments carried out where the vehicle was driven to the target in an unknown, unstructured scenario constructed dynamically (see the points gathered by the laser and the trajectory of the vehicle in Fig. 15(a), and some snapshots of the motion in Figs. 15(b)–(e)). Notice that the avoidance task and motion to the goal was successfully carried out using the technique proposed. The robot reached the destination in 44 s.

Taking the shape into account was very important since it restricts manoeuvring in confined spaces (see Fig. 15(a),(c)). With respect to the kinematics, the conclusions are similar to the PFM. We show the turning radii obtained using the ND method in the ARM^P in Fig. 16(a) (notice that the manoeuvres are more abrupt due to the fast reactions required in very confined spaces) and the velocity profiles in Fig. 16(b). Some of these turning radii are illustrated in Figs. 15(b)–(f),(c)–(g),(d)–(h),(e–f) at different moments. Notice that the motion on the circles avoids collisions (within the

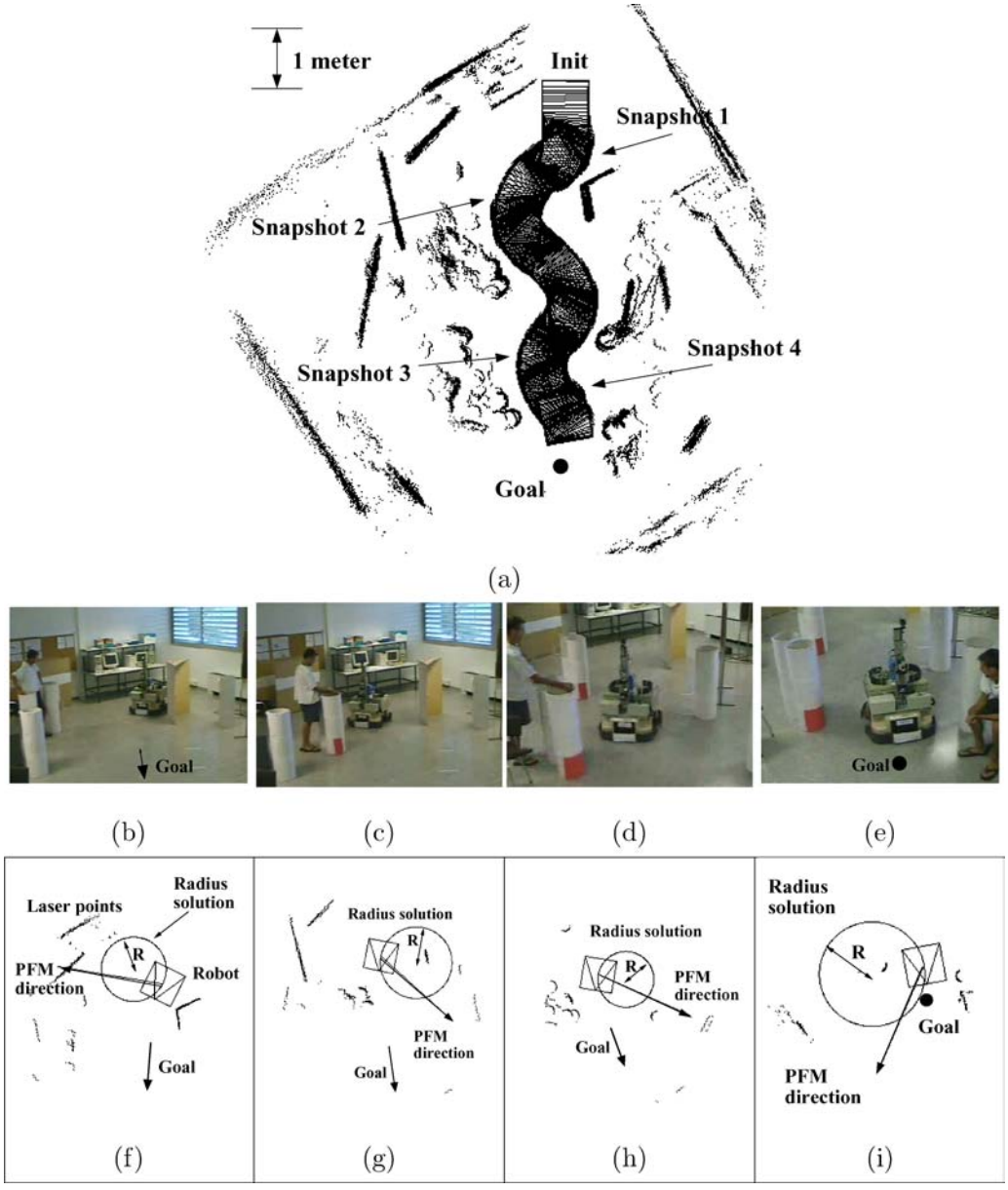

Figure 13. Experiment with the PFM method using the abstraction layer to take into account the shape and kinematics constraints. (a) Path executed by the vehicle and laser points collected in the experiment. (b)–(e) Some snapshots that correspond to Figs. (f)–(i) showing the laser measurement used, the robot location, the radius solution using PFM in the ARM^P and the direction that would be obtained by PFM without using the abstraction.

short-time application of the method, that is, the sampling period).

Finally, we show the ND solutions without using our abstraction layer in Figs. 15(f)–(i) as "ND direction". The most significant case arises in Fig. 15(h). The direction is completely perpendicular to the robot, which is physically impossible to execute. Nevertheless, in this situation, the ND method with the abstraction layer computes a radius 0, that is equivalent to turning on itself in that direction (desired behaviour).

Another conclusion of the experiments is that the abstraction layer allows for applying some methods on robots with restrictions, but does not improve the quality of a method. If a method has difficulties under some conditions, they will be present if they arise in ARM^P. For example, one difficulty of a potential method is to drive the vehicle between two very close obstacles, or the appearance of instabilities or oscillations in the motion (Koren and Borenstein, 1991). These difficulties also appeared when PFM was integrated in the abstraction layer. Nevertheless the opposite is also true. If the method works well under certain conditions, it will also perform well in ARM^P if they appear. This is the case of the ND method that is robust under conditions of extreme manoeuvrability (dense and complex scenarios). Similar results were obtained when the ND was integrated in the abstraction layer (Fig. 15(a)).

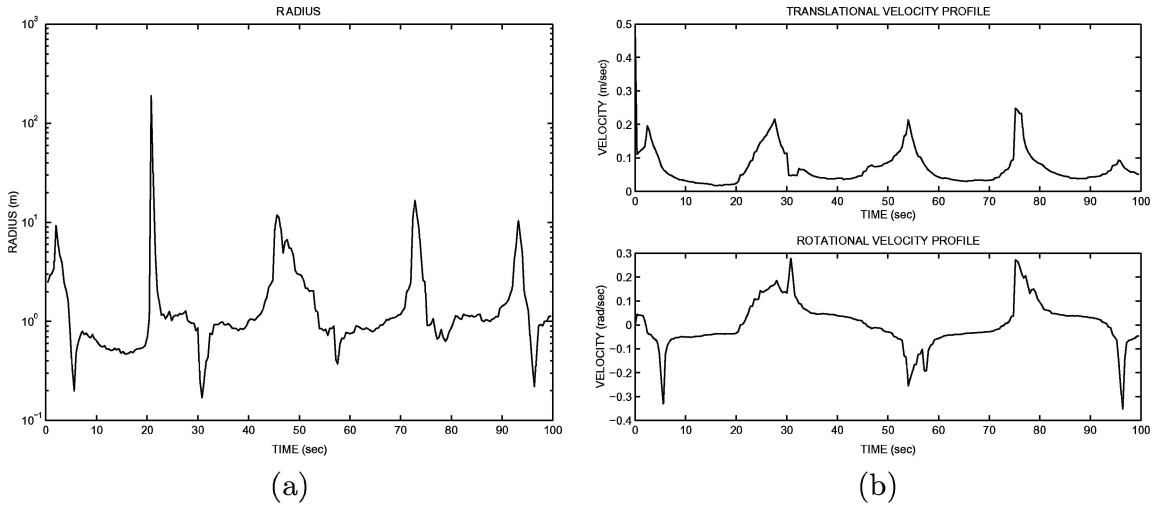

Figure 14. (a) Profile of the turning radius computed in the experiment with the potential field method and (b) linear and angular velocities profiles.

In summary, two reactive methods integrated with the abstraction carried out the avoidance task while driving the vehicle to the destination. The computed motions took into account the vehicle constraints, whereas when both methods were used alone (without the abstraction) they compute solutions that could not be executed by the vehicle without approximations (very strong in some cases).

7. Discussion and Conclusions

We have presented a framework to abstract the shape and the kinematics of a vehicle when using an obstacle avoidance method. The abstraction is based on a calculus to compute the exact boundary of the region in collision in the manifold of the configuration space defined by elementary circular paths, and in a change of coordinates where the elementary motions become omnidirectional. This construction is performed before applying the avoidance method within the perception—motion cycle.

In what follows, we discuss the main characteristics of our approach regarding obstacle avoidance methods, and next we address additional issues such as the quality and locality of the solution.

7.1. Comparison with Existing Obstacle Avoidance Methods

Next we discuss aspects related to obstacle avoidance methods and compare the approach with similar techniques that use the avoidance method before and turn the approximated solutions into admissible commands.

The obstacle avoidance methods that consider these restrictions compute collisions over a set of elementary circular paths. In some cases the set is composed of circular arcs (Hebert et al., 1997; Feiten et al., 1994; Ulrich and Borenstein, 1998; Hait et al., 1999), and in others is a set of commands (where each one implies a circular path) (Fox et al., 1997; Simmons 1996; Schlegel, 1998). The complexity of this process is $N \times M \times C$, where N is the number of obstacle points, M is the number of pieces of the piece-wise function that describes the robot boundary and C is the number of pre-defined paths. The important point is that when the shape is circular or polygonal, the intersection between the robot outline and the obstacle over a circular path can be calculated (Fox et al., 1997; Arras et al., 2002). Then, $M = 1$ in the case of circular robots and M is the number of sides in the case of polygonal robots. However, this is true as long as the intersection between the robot outline and the circular paths to obstacles can be calculated. For instance, in the case of the cardioid-shape vehicle, it has to be possible to solve the system formed by Eq. (13) and $x^2 + (y - R)^2 = (R - c)^2$ (where c depends on the obstacle point and R is the radius of the path inspected). If there is no exact solution, one can try to solve the system with a numerical method or by projecting the robot position over the path checking collisions (dynamic simulation). Both strategies increase the complexity and could lead to an approximate solution.

In this work, the procedure to compute the exact region of the configuration space in collision over the manifold of circular paths has a $N \times M$ complexity. The complexity of the calculus proposed is less than existing methods, but more important the solution is exact and can always be computed (as long as the boundary of the vehicle can be described by a piece-wise function). Another important consequence is that this calculus allows for maintaining a continuous representation of the space of solutions of the method (that is why the factor $\times C$ does not appear in complexity). Existing methods could benefit from this procedure to reduce complexity, to straight forward consider any vehicle shape or to avoid a discretization of the space of solutions of the method.

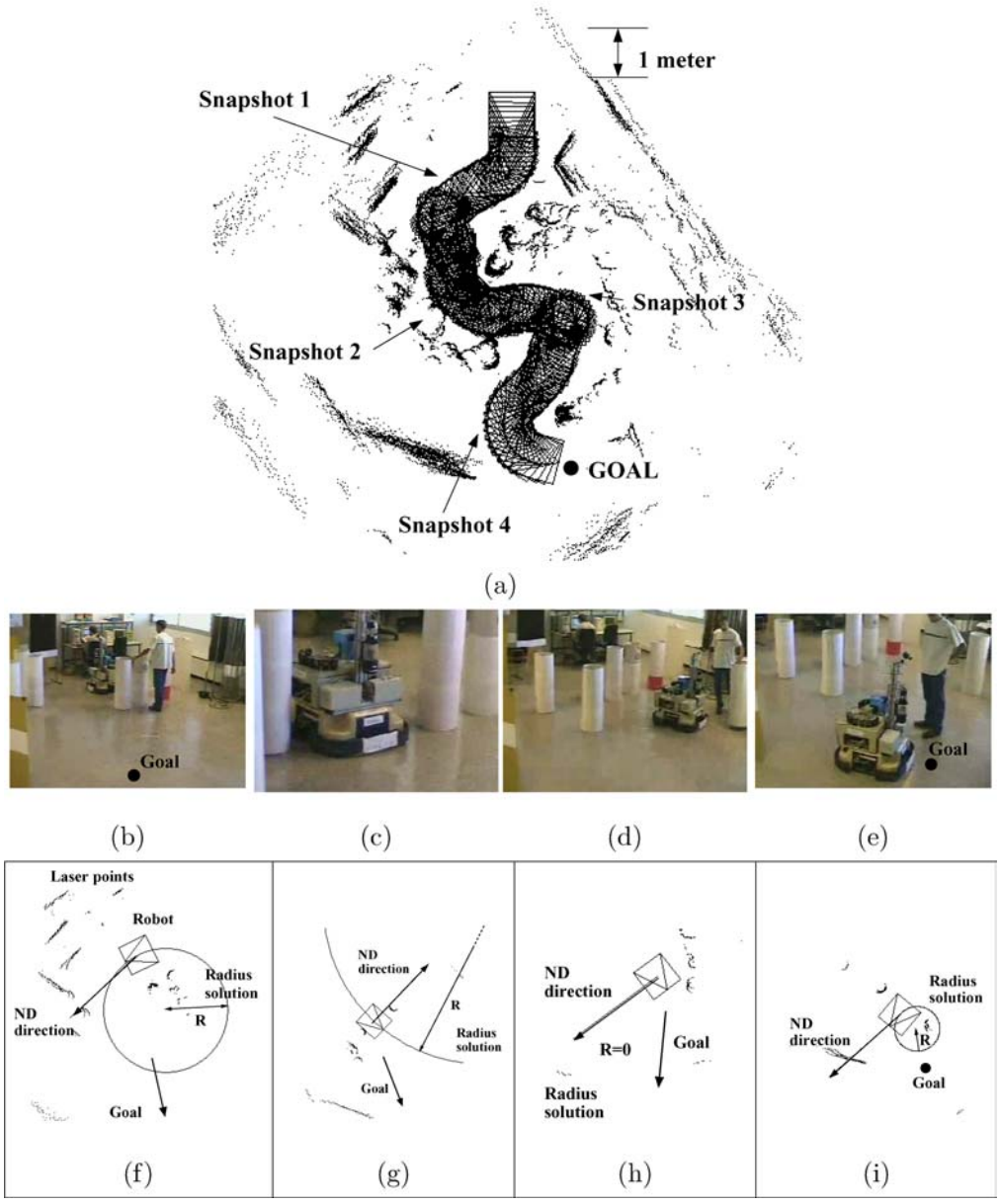

Figure 15. Experiment with the ND method using the abstraction layer to take into account the shape and kinematic constraints. (a) Path executed by the vehicle and laser points collected in the experiment. (b)–(e) Some snapshots that correspond to Figs. (f)–(i) showing the laser measurement used at that moment, the robot location, the radius solution using the ND in the ARM^P and the direction obtained by the ND without using the abstraction.

There are other techniques similar to ours but used after the avoidance technique (not before). For example (Luca and Oriolo, 1994; Bemporad et al., 1996) modify the output of the avoidance method by a feedback action and (Montano and Asensio, 1997); (Asensio and Montano, 2002) use a motion generator to align the vehicle with the direction solution. These methods run into problems in situations requiring high manoeuvrability (Bemporad et al., 1996) (due to the approximation of the non holonomicity by a motion direction). The vehicle shape could be considered after this stage by dynamic simulation (Minguez and Montano, 2002). In our technique the shape and the kinematics are taken into account in the methodology before using the method (when the method is used, the constraints have already been considered). In particular, situations that required great manoeuvrability were overcome (Fig. 15(a)).

7.2. *Quality and Locality of the Solution*

In the obstacle avoidance context, two key questions remain: the quality of the motion regarding current techniques and the locality of the solution. As discussed in the experiments, our method is a tool to allow an

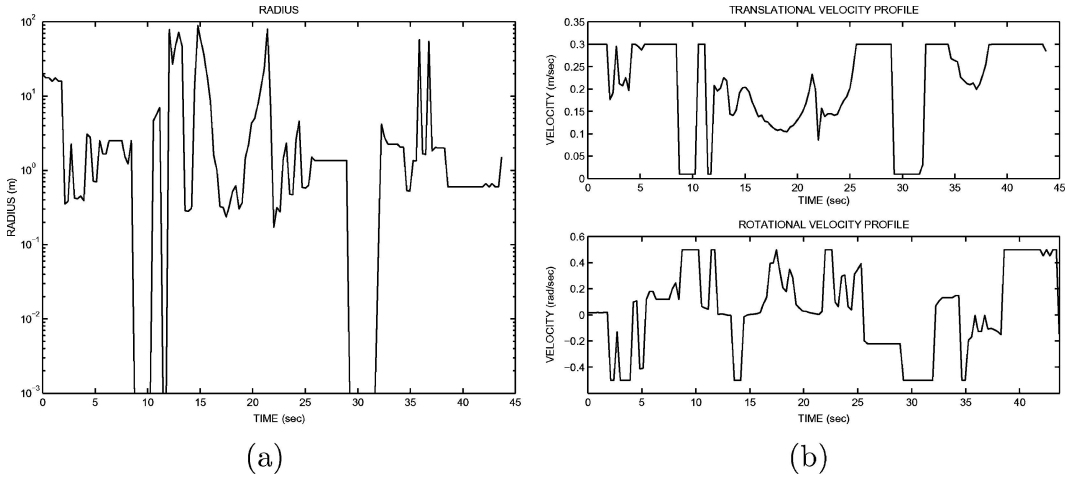

Figure 16. (a) Profile of the turning radius computed in the experiment with the nearness diagram navigation and (b) linear angular velocity and velocity profiles.

avoidance technique to consider certain restrictions. The performance of an obstacle avoidance method with this tool depends on the method and not on the tool. Therefore, to draw conclusions about the motion quality of the reactive methods, they should be compared individually (see Minguez and Montano, 2004 for a comparison of some methods). Nevertheless, the advantage of the abstraction with respect to existing techniques is generality, because many existing or future methods could use the same methodology to address the vehicle constraints.

The obstacle avoidance methods are local techniques to solve the motion problem, so cyclical motions and trap situations persist. This is a common characteristic of these methods. Nevertheless, movement is improved in terms of flexibility, adaptation and robustness in unknown, unstructured and dynamic surroundings with an a-priori unpredictable behaviour (the sensory information is included at a high frequency in the motion control loop, around 4 and 6.3 Hz in the experiments). The role of the technique presented here is to consider the vehicle restrictions in the application of the method. Therefore, this technique does not change the local nature of the method. In order to deal with the locality of obstacle avoidance methods, hybrid systems should be developed (see Arkin, 1999 for a discussion on different architectures and (Minguez and Montano, 2005) for a similar discussion in the motion context). These systems are made up of a module of global deliberation (planning) and an obstacle avoidance module (avoidance of collisions) whose synergy generates motion that avoids trap situations in (Ulrich and Borenstein, 2000; Brock and Khatib, 1999; Minguez et al., 2001; Stachniss and Burgard, 2002; Philipsen and Siegwart, 2003).

7.3. *Final Remarks*

Our belief is that this technique can be very useful to many researchers since it provides a framework to improve the domain of applicability of a wide range of obstacle avoidance methods (without significant modifications). To integrate this framework in existing vehicles is straightforward. In our work, we used it to extend two obstacle avoidance methods available in our laboratory. The results demonstrate that the avoidance task is carried out successfully, while the motions take into account the shape and kinematics. This was the objective of this work.

Acknowledgments

We would like to thank the Computer and Robot Vision Lab at the Instituto de Sistemas e Robótica that hosted Javier Minguez during his visit to the Instituto Superior Técnico, Lisbon, Portugal. Thanks is also given to the Robotics and Artificial intelligence group directed by R. Chatila that hosted Javier Minguez during his visit to the LAAS-CNRS, France. In particular we thank J.P Laumond and F. Lamiraux for their fruitful comments and discussions during the preparation of this manuscript. This work was partially supported by MCYT DPI2003-7986, DGA2004T04 and the *Caja de Ahorros de la Inmaculada de Aragón*.

Notes

1. In the syncro-drive robot these are also the controls. Other robots follow this model up to a variable change. This distinction becomes more important when dynamics are taken into account (actuator limits translate different into constraints (v, w)).
2. From a physical point of view, this point is equivalent to having a free wheel at distance 1 from the origin on the X-axis, which aligns tangent to the circle of motion with angle α'.
3. As opposed to the PFM, the ND is a method that works in the workspace. In the ND, the EKT transformation is applied directly to the obstacle points and not to the C-obstacle region (that is not computed). This is why the ND version is less time consuming than the PFM.

4. This is because its original implementation was for a holonomic robot with a sensor of 180° of visibility. Under these conditions the instantaneous backward motion was eliminated.
5. In this case, to reach a point of the space, the vehicle moves over a circle but always forwards. In other words, to reach the point (x, y) in the Fig. 8 the distance to the point is $2\pi r - L$ instead of L (Eq. (16)). In addition, the angle α is not required (since there is no backward motion) and α' completely describes the circle (Eq. (18)).

References

Arkin, R. 1999. *Behavior-Based Robotics*. The MIT Press.

Arras, K., Persson, J., Tomatis, N., and Siegwart, R. 2002. Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. In *IEEE Int. Conf. on Robotics and Automation*, Washington, USA, pp. 3050–3055.

Asensio, J. and Montano, L. 2002. A kinematic and dynamic model-based motion controller for mobile robots. In *15th IFAC World Congress*, Barcelona, Spain.

Bemporad, A., Luca, A.D., and Oriolo, G. 1996. Local incremental planning for car-like robot navigating among obstacles. In *IEEE International Conference on Robotics and Automation*. Minneapolis, USA, pp. 1205–1211.

Borenstein, J. and Koren, Y. 1989. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):1179–1187.

Borenstein, J. and Koren, Y. 1991. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7:278–288.

Brock, O. and Khatib, O. 1999. High-speed navigation using the global dynamic window approach. In *IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, pp. 341–346.

Buhmann, J.M., Burgard, W., Cremers, A.B., Fox, D., Hofmann, T., Schneider, F.E., Strikos, J., and Thrun, S. 1995. The mobile robot RHINO. *AI Magazine*, 16(2):31–38.

Castellanos, J.A., Montiel, J., Neira, J., and Tardós, J.D. 1999. The SPmap: A probabilistic framework for simultaneous localization and map building. *IEEE Trans. Robotics and Automation*, 15(5):948–952.

Dissanayake, M.W.M.G., Newman, P., Durrant-Whyte, H.F., Clark, S., and Csorba, M. 2001. A solution to the simultaneous localization and map building (slam) problem. *IEEE Trans. Robotics and Automation*, 17(3):229–241.

Feiten, W., Bauer, R., and Lawitzky, G. 1994. Robust obstacle avoidance in unknown and cramped environments. In *IEEE Int. Conf. on Robotics and Automation*, San Diego, USA, pp. 2412–2417.

Fox, D., Burgard, W., and Thrun, S. 1997. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33.

Hait, A., Simeon, T., and Taix, M. 1999. Robust motion planning for rough terrain navigation. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Kyongju, Korea, pp. 11–16.

Hebert, M., Thorpe, C., and Stentz, A. 1997. *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*, Kluwer Academic Publishers.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5:90–98.

Ko, N. and Simmons, R. 1998. The lane curvature velocity method for local obstacle avoidance. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria, Canada, pp. 1615–1621.

Koenig, S. and Simmons, R. 1998. Xavier: A robot navigation architecture based on partially observable markov decision process models. In R.B.D. Kortenkamp and R. Murphy (eds.), *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, MIT Press, pp. 91–122.

Koren, Y. and Borenstein, J. 1991. Potential field methods and their inherent limitations for mobile robot navigation. In *IEEE Int. Conf. on Robotics and Automation*, Vol. 2, Sacramento, CA, pp. 1398–1404.

Krogh, B.H. and Thorpe, C.E. 1986. Integrated path planning and dynamic steering control for autonomous vehicles. In *IEEE Int. Conf. on Robotics and Automation*, San Francisco, USA, pp. 1664–1669.

Latombe, J.C. 1991. *Robot Motion Planning*. Kluwer Academic.

Laumond, J., Sekhavat, S., and Lamiraux, F. 1998. Guidelines in nonholonomic motion planning for mobile robots. *Robot Motion Planning and Control*, 229.

Leonard, J.J. and Feder, H.J.S. 2000. A computationally efficient method for large scale concurrent mapping and localization. In D. Koditschek and J. Hollerbach (eds.) *Robotics Research: The Ninth International Symposium*, Springer Verlag, Snowbird, Utah, pp. 169–176.

Lozano-Perez, T. 1983. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, 32(2):108–120.

Luca, A.D. and Oriolo, G. 1994. Local incremental planning for nonholonomic mobile robots. In *IEEE International Conference on Robotics and Automation*, San Diego, USA, pp. 104–110.

Minguez, J. and Montano, L. 2002. Robot navigation in very complex dense and cluttered indoor/outdoor environments. In *15th IFAC World Congress*, Barcelona, Spain.

Minguez, J. and Montano, L. 2004. Nearness Diagram (ND) navigation: Collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59.

Minguez, J. and Montano, L. 2005. Autonomous sensor-based motion control in unknown, dynamic and troublesome scenarios. *Journal of Robotics and Autonomous Systems*, 52(4):290–311.

Minguez, J., Montano, L., Simeon, N., and Alami, R. 2001, Global Nearness Diagram Navigation (GND). In *IEEE International Conf. on Robotics and Automation*, Seoul, Korea, pp. 33–39.

Minguez, J., Osuna, J., and Montano, L. 2004. A divide and conquer strategy to achieve reactive collision avoidance in troublesome scenarios. In *IEEE International Conference on Robotics and Automation*, Minessota, USA, pp. 3855–3862.

Montano, L. and Asensio, J. 1997. Real-time robot navigation in unstructured environments using a 3D laser rangefinder. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Vol. 2, Grenoble, France, pp. 526–532.

Morisset, B. and Gallab, M. 2002. Learning how to combine sensory-motor modalities for a robust behavior. *Advances in Plan-Based Control of Robotic Agents*, Lecture Notes in Artificial Intelligence 2466, Springer, pp. 157–178.

Ogren, P. and Leonard, N. 2002. A tractable convergent dynamic window approach to obstacle avoidance. In *IEEE International Conference on Robotics and Automation*, Laussane, Switzerland, pp. 446–451.

Philipsen, R. and Siegwart, R. 2003. Smooth and efficient obstacle avoidance for a tour guide robot. In *IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan.

Schlegel, C. 1998. Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Canada, pp. 594–599.

Simmons, R. 1996. The curvature-velocity method for local obstacle avoidance. In *IEEE Int. Conf. on Robotics and Automation*, Minneapolis, USA, pp. 3375–3382.

Stachniss, C. and Burgard, W. 2002. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Switzerland, pp. 508–513.

Thrun, S., Burgard, W., and Fox, D. 2000. A real-time algorithm for robot mapping with applications to multirobot and 3d mapping. In *IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, pp. 321–328.

Tilove, R.B. 1990. Local obstacle avoidance for mobile robots based on the method of artificial potentials. In *IEEE Int. Conf. on Robotics and Automation*, Cincinatti, OH, Vol. 2, pp. 566–571.

Ulrich, I. and Borenstein, J. 1998. VFH+: Reliable obstacle avoidance for fast mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, pp. 1572–1577.

Ulrich, I. and Borenstein, J. 2000. VFH∗: Local obstacle avoidance with look-ahead verification. In *IEEE Int. Conf. on Robotics and Automation*, San Francisco, USA, pp. 2505–2511.

Javier Minguez received the physics science degree in 1996 from the Universidad Complutense de Madrid, Madrid, Spain, and the Ph.D. degree in computer science and systems engineering in 2002 from the University of Zaragoza, Zaragoza, Spain. During his student period, in 1999 he was a research visitor in the Robotics and Artificial Intelligence Group, LAASC-NRS, Toulouse, France. In 2000, he visited the Robot and ComputerVision Laboratory (ISR-IST), Technical University of Lisbon, Lisbon, Portugal. In 2001, he was with the Robotics Laboratory, Stanford University, Stanford, USA. He is currently a fulltime Researcher in the Robot, Vision, and Real Time Group, in the University of Zaragoza. His research interests are obstacle avoidance, motion estimation and sensor-based motion systems for mobile robots.

Luis Montano was born on September 6, 1958 in Huesca, Spain. He received the industrial engineering degree in 1981 and the PhD degree in 1987 from the University of Zaragoza, Spain. He is an Associate Professor of Systems Engineering and Automatic Control at the University of Zaragoza (Spain). He has been Head of the Computer Science and Systems Engineering Department of the University of Zaragoza. Currently he is the coordinator of the Production Technologies Research in the Aragon Institute of Engineering Research and of the Robotics, Perception and Real Time group of the University of Zaragoza. He is principal researcher in robotic projects and his major research interests are mobile robot navigation and cooperative robots.

José Santos-Victor received the PhD degree in Electrical and Computer Engineering in 1995 from Instituto Superior Técnico (IST - Lisbon, Portugal), in the area of Computer Vision and Robotics. He is an Associate Professor at the Department of Electrical and Computer Engineering of IST and a researcher of the Institute of Systems and Robotics (ISR), at the Computer and Robot Vision Lab - VisLab. (http://vislab.isr.ist.utl.pt) He is the scientific responsible for the participation of IST in various European and National research projects in the areas of Computer Vision and Robotics. His research interests are in the areas of Computer and Robot Vision, particularly in the relationship between visual perception and the control of action, biologically inspired vision and robotics, cognitive vision and visual controlled (land, air and underwater) mobile robots.

Prof. Santos-Victor is an IEEE member and an Associated Editor of the IEEE Transactions on Robotics.