

Extending Reactive Collision Avoidance Methods to Consider any Vehicle Shape and the Kinematics and Dynamic Constraints

Javier Minguez and Luis Montano

Abstract—Many collision avoidance methods do not consider the vehicle shape and its kinematic and dynamic constraints, assuming a point-like and omnidirectional robot without acceleration constraints. The contribution of this paper is a methodology to consider the exact shape and kinematics as well as the effects of the dynamics in the collision avoidance layer, although the original avoidance method at hand does not address them. This is achieved by abstracting these constraints from the avoidance methods in such a way that when the method is applied, the constraints have been already taken into account. This work is a starting point to extend the domain of applicability of a wide range of collision avoidance methods.

Index Terms—Reactive Collision Avoidance, Mobile Robots.

I. INTRODUCTION

ONE fundamental skill of autonomous vehicles is to be able to execute collision-free motion tasks in unknown, unstructured and evolving environments. Under these conditions, the techniques widely used to generate motion are the collision avoidance methods. A collision avoidance method is a procedure that works within a perception-action process: sensors collect information of the state of the environment, which is then processed to compute the collision-free goal-oriented motion. The vehicle executes the motion and the process restarts (Figure 1). The result is an on-line motion sequence that drives the vehicle to the goal while avoiding collisions with the obstacles perceived with the sensors.

An essential aspect of collision avoidance methods is to consider the restrictions imposed by the vehicle used: the shape, the kinematics and the dynamics. This consideration is important, since if the shape of the robot is simply approximated, collisions will occur or the vehicle will invade prohibited zones of the space. If the kinematics are ignored, the planned movements will not correspond with the actual motions, putting the security at risk. In addition, if dynamics are ignored, the planned motions could not be feasible, thereby once again putting the motion mission at risk. These issues are thus relevant in robot collision avoidance and especially in the application at hand: a robotic wheelchair for human transportation.

The work described here centers on this problem: to consider the shape and both the kinematic and dynamic constraints during the application of a collision avoidance method. The idea is to project distance measurements into a space in which

Javier Minguez and Luis Montano are with the Instituto de Investigación en Ingeniería de Aragón, and with the Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain.

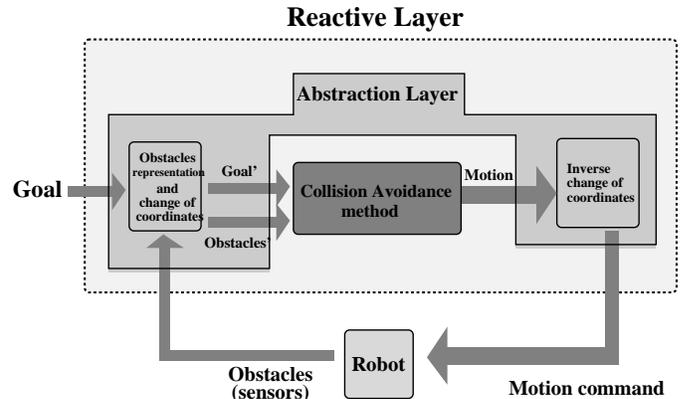


Fig. 1. The abstraction layer abstracts the shape and the kinematics and dynamics of the vehicle from the avoidance method. The idea is to understand the method as a “black-box” and modify the representation of its inputs, so that they implicitly have information about these restrictions. The method is applied naturally, however its solutions consider the restrictions (the method is “unaware” of it).

the robot can be regarded as a holonomic point. The projection accounts for collision constraints and kinematic and dynamic motion constraints (the trajectories are restricted to the family of circular arcs). In the new space, many reactive collision avoidance methods can be applied to the holonomic point since all the constraints are encoded in the obstacles and space itself. The motion command computed is projected back and applied to the robot. Therefore, the proposed approach is a general method to extend a whole set of well-known obstacle avoidance approaches to consider vehicle shape, kinematic and dynamic constraints. This technique has been demonstrated in real-world experiments by *wrapping* a potential field method to perform obstacle avoidance on a differential-drive wheelchair.

II. RELATED WORK AND OVERVIEW

Classically, the mobility problem has been addressed by computing a geometric path free of potential collisions with obstacles [21]. Nevertheless, when the surroundings are unknown or evolve, these techniques fail, since a precomputed path will almost certainly hit obstacles. Reactive collision avoidance is an alternative way to compute motion by introducing sensor information within the control loop (Figure 1). The main cost of considering the reality of the world during execution is locality. In this instance, if global reasoning is required, a trap situation could occur. Despite this limitation,

collision avoidance techniques are mandatory to deal with mobility problems in unknown and dynamic surroundings.

In collision avoidance, there is no exact procedure to take into account any shape and the kinematics and dynamics of the vehicle simultaneously. The shape and kinematics lead to a geometric problem: to compute an elemental path free of collisions (and the command that generates this path). Dynamics is a complex problem since it involves factors such as accelerations, maximum torque, inertia, slipping, etc. As usual in collision avoidance, we consider here the scope of the dynamics derived from the maximum vehicle accelerations: (*i*) motion commands reachable in a short period of time (reachable commands), and (*ii*) commands that assure that the vehicle can be always stopped before collision by applying the maximum deceleration (admissible commands).

The collision avoidance problem with these constraints has been taken into account from two perspectives: taking into account the constraints in the design of the collision avoidance method, or modifying the commands computed by a given method to comply with the constraints. In the first class of methods, some have been designed to solve the problem in the velocity space [13], [35]. They first compute the set of reachable commands in a short period of time, which are free of collisions and allow for stopping the vehicle safely. Next, one command is selected with an optimization process that favors progress, safety and convergence to the target. The elegance and simplicity of these methods have led to extensions and applications to different vehicles [30], [10], [3], [18], [6], [33]. Other techniques pre-compute a set of arcs of circles (elemental paths), free of collisions and resulting from reachable commands. Next, they select one arc based on obstacle avoidance and convergence to the goal criteria [39], [16], [12], [15]. In general, all these methods take into account the shape, kinematics and dynamics of the vehicle, but only approximately. This approximation is due to a discretization of the space of solutions (motions), or due to the fact that, depending on the vehicle shape, it could necessitate use of a numerical method or a dynamic simulation (projecting vehicle positions over admissible paths) to check collisions. That is why these methods are used on basic vehicle shapes (circular [30], [10], [18], [6], [39] or polygonal [33], [16], [12], [3]). These techniques are not generic in the sense that it seems difficult to extrapolate these strategies to be used over other existing methods (ad-hoc methods).

In the second class of methods, the solution of the obstacle avoidance technique is converted in a command that complies with the constraints. For instance, the output of the avoidance method is modified with a feedback action that aligns the vehicle with the avoidance direction in a minimum squares fashion [22], [5]. A similar solution is proposed by breaking down the problem into subproblems (collision avoidance, kinematics and dynamics, and shape) and dealt with sequentially [24]. Another approach is based on the command filters [40]; after using the avoidance method, the commands that are not reachable or do not avoid collisions are filtered out and converted into commands that are reachable and free of collisions. Other works in this direction proposed a simple model of the vehicle and use control theory to compute the collision-free commands

[1], [23]. The advantage of these strategies is the generality, since they can be used by many avoidance methods. However, the generated motions only take into account the shape of the vehicle, but only approximately. This is because, although the shape of the vehicle is addressed in the avoidance technique, the computed motion is next modified to satisfy the kinematics and dynamics. Thus, the final command does not guarantee avoidance with the exact shape. This leads to problems when the holonomic solution cannot be approximated or when maneuverability is a determinant factor [5].

A. Overview and Contributions

The majority of collision avoidance methods do not consider the vehicle constraints mentioned. They assume a point-like and omnidirectional vehicle without acceleration constraints. The **main contribution** of this work is a scheme to consider the exact shape and kinematics, and the effects of the dynamics (reachable and admissible commands) in the collision avoidance layer. The idea is to abstract these constraints from the usage of the avoidance methods (Figure 1). This technique could be applied to many vehicles with arbitrary shape (we illustrate the approach with a differential-drive and rectangular robot).

The construction of this abstraction layer has three parts that correspond with the **partial contributions** of this study:

- First, we construct – centered on the robot at each time – the two-dimensional manifold of the three-dimensional configuration space defined by elemental circular paths. This manifold contains all the configurations that can be reached at each step of the obstacle avoidance. The contribution is the exact calculation of the obstacle representation on this manifold for any vehicle shape (i.e. the configurations in collision). In this manifold, a point represents the vehicle.
- Second, we describe the exact calculation of the admissible configurations, which result from the obstacle regions computed previously (with the assumption that the path of breaking is a circular elemental path, typical in obstacle avoidance). Furthermore, we also represent the reachable configurations by reachable commands in the manifold. The effect of the dynamics is represented in the manifold.
- Finally, we propose a change of coordinates of the manifold so that the circular paths become straight segments. With the manifold represented in these coordinates, the motion is free of kinematic constraints.

As a result, we transform the three-dimensional collision avoidance problem with shape, kinematics and dynamic constraints into the simple problem of moving a point in a two-dimensional space without constraints (usual approximation in collision avoidance). Thus, existing methods become applicable.

With this technique, many existing or future avoidance methods could be applicable to a wide class of non-holonomic robots with arbitrary shape without any redesign. For example, this result could be used with the Potential Field methods [17], [20], [37], [8], the Vector Field Histogram [9], [38], or the Nearness Diagram Navigation [26]. To validate the technique,

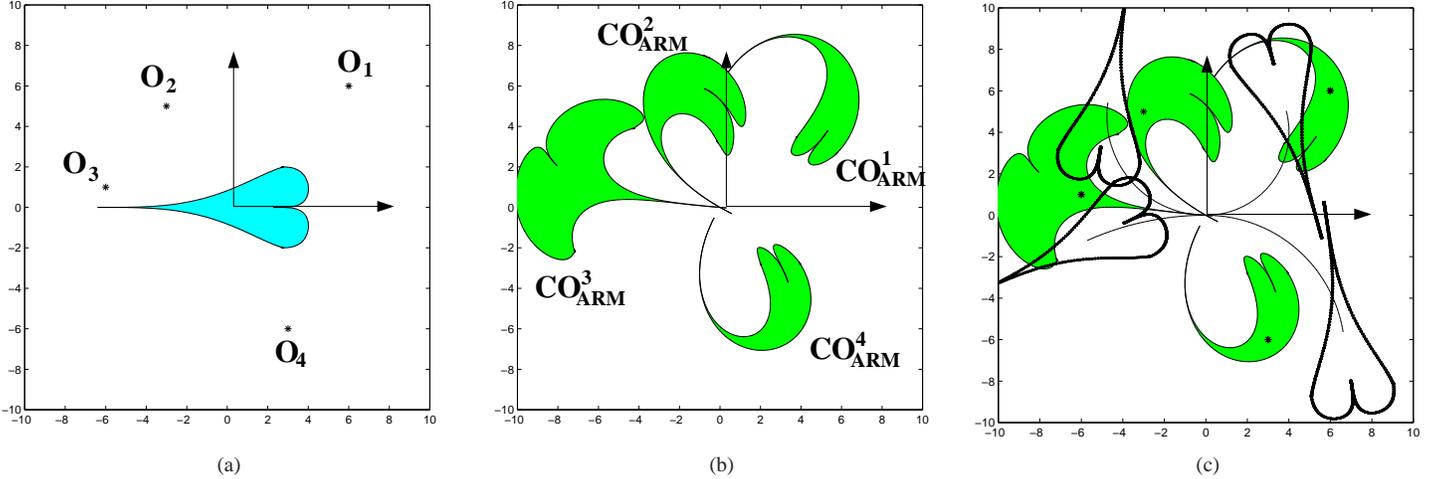


Fig. 2. This Figure shows the computation of the region of configurations in collision for a heart-shaped robot that moves in circular paths. (a) Robot and obstacles O_i ; (b) each obstacle point creates a region of collision locations CO_{ARM}^i that all together are CO_{ARM} . The free space is the 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.

we used a potential field method [17] integrated in a real platform (rectangular and differential-drive).

Partial and previous results of this research were presented in [29], [28], [25]. This work describes the complete study of the shape, kinematics and dynamics in a unified framework.

The present manuscript is organized as follows: in Section III, we describe the computation of the manifold. In Sections IV and V, we show how to abstract the shape and the dynamics. Section VI outlines the change of coordinates to abstract the kinematics. We discuss the abstraction layer in Section VII. In Section VIII, we summarize the experimental results, and in Section IX, we draw the conclusions of our work.

III. THE ARC REACHABLE MANIFOLD (ARM) AND CONFIGURATIONS IN COLLISION

We focus our attention on synchro-drive robots moving on a flat surface, where the workspace \mathcal{W} and the configuration space \mathcal{CS} are \mathbf{R}^2 and $\mathbf{R}^2 \times S^1$, respectively. A configuration \mathbf{q} contains the location and the orientation $\mathbf{q} = (x, y, \theta)$. Let \mathcal{U} be the control space and $\mathbf{u} = (v, \omega)$ a control vector (where v is the linear and ω the angular velocity). We assume that during the execution of a constant control, the motion is constrained on a circular elemental path (see [13] for a characterization of this assumption). We then show how the paths lie on a two dimensional manifold of \mathcal{CS} and how it is possible to compute the mapping of the obstacles to this manifold.

Let the reference be the robot system of reference. An admissible circular path from the origin $(0, 0)$ to a given point (x, y) has the instantaneous turning center on the Y -axis. The radius of that circle is:

$$r = \frac{x^2 + y^2}{2y} \quad (1)$$

The robot orientation θ tangent to this circle at (x, y) is:

$$\theta = f(x, y) = \begin{cases} \arctan 2(x, \frac{x^2 - y^2}{2y}) & \text{if } y \geq 0 \\ -\arctan 2(x, -\frac{x^2 - y^2}{2y}) & \text{otherwise} \end{cases} \quad (2)$$

Function f is differentiable in $\mathbf{R}^2 \setminus (0, 0)$. Thus $(x, y, f(x, y))$ defines a two dimensional manifold in $\mathbf{R}^2 \times S^1$. We called *Arc Reachable Manifold*, $ARM(\mathbf{q}_0) \equiv ARM$, since it contains all the configurations attainable by elemental circular paths from the current robot configuration \mathbf{q}_0 (i.e. all configurations attainable at each step of the obstacle avoidance).

Let $g(\lambda) = (g_x(\lambda), g_y(\lambda))$, the piece-wise function that describes the robot boundary with λ , a parameter defined in a finite interval. We then assume that the obstacle information is given in the form of a cloud of points (typical metric information from the range sensors). For each obstacle point $\mathbf{p}_f = (x_f, y_f)$, there is a region of configurations in collision in the configuration space and a part of it lies in ARM (we call this region CO_{ARM}^i). To compute it, we develop Equation (2) and use some geometric properties of the problem (see [29] for details) leading to:

$$h(\lambda) = [a \cdot (x_f + g_x(\lambda)), a \cdot (y_f - g_y(\lambda))] \quad (3)$$

where

$$a = \frac{[(y_f^2 - g_y(\lambda)^2) + (x_f^2 - g_x(\lambda)^2)] \cdot [(y_f - g_y(\lambda))^2 + (x_f - g_x(\lambda))^2]}{(y_f - g_y(\lambda))^4 + 2(x_f^2 + g_x(\lambda)^2)(y_f - g_y(\lambda))^2 + (x_f^2 - g_x(\lambda)^2)^2}$$

Function h is the piece-wise function that describes the collision region boundary for a given obstacle \mathbf{p}_i . The obstacle region is $CO_{ARM} = \bigcup_i CO_{ARM}^i$ for all obstacle points \mathbf{p}_i . The important point is that, for an arbitrary robot shape, one can compute the exact obstacle region CO_{ARM} in the ARM (manifold of the configuration space reachable by circular paths). Figure 2 shows an academic and illustrative example

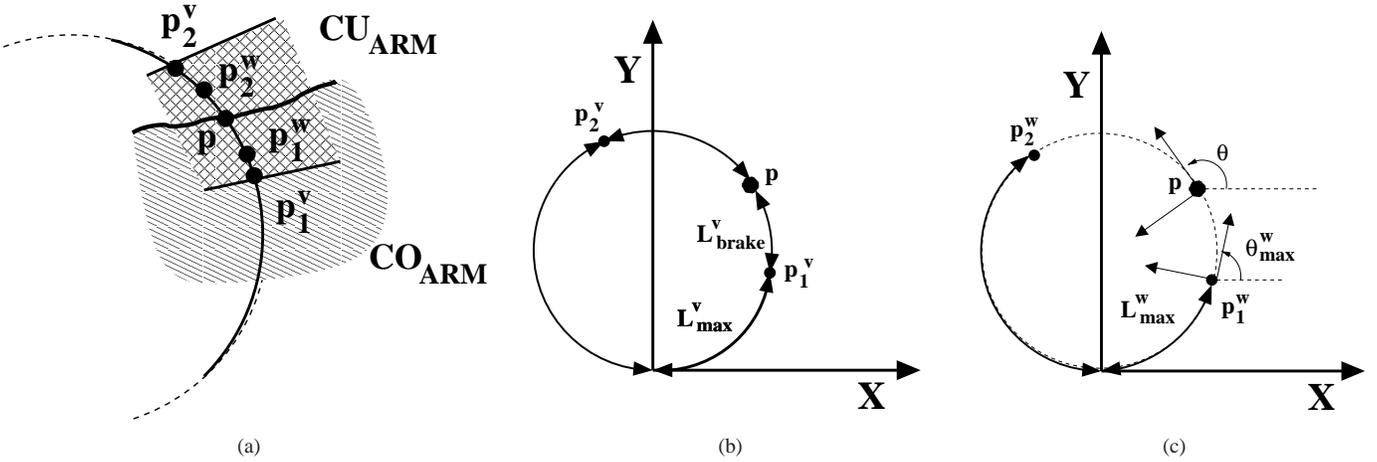


Fig. 3. These figures show the computation of the region of unsafe configurations, CU_{ARM} , given a point $\mathbf{p} \in \mathbb{R}^2$. (a) Of the four limit points, $\{\mathbf{p}_1^v, \mathbf{p}_2^v, \mathbf{p}_1^w, \mathbf{p}_2^w\}$, the two points farther in terms of distance over the circle in both directions of the circle leading to \mathbf{p} are the border points of CU_{ARM} . (b) Translational $\{\mathbf{p}_1^v, \mathbf{p}_2^v\}$ and (c) rotational $\{\mathbf{p}_1^w, \mathbf{p}_2^w\}$ velocity cases.

of a heart-shaped robot, whose boundary $g(\lambda)$ is given by:

$$\begin{cases} x_r = 2 \sin^7(\lambda) \\ y_r = -4.5 \cos(\lambda)(1 + 1.2 \cos(\lambda)) + \cos^{\frac{1}{4}}(\lambda) + 2.5 \end{cases} \quad (4)$$

with $\lambda \in [0, \pi]$. Replacing this expression in Equation (3), we obtain the CO_{ARM}^i for one obstacle point \mathbf{p}_i and respectively the CO_{ARM} for all obstacles.

The complexity of this calculation is $N \times M$, where N is the number of obstacle points and M is the number of pieces of function g . For instance, $M = 1$ for a circular robot or the heart-shaped robot, and M is equal to the number of sides for a polygonal robot (in this case, there is one parametrization of each segment). Notice that the calculation computes the region in collision for any vehicle shape without approximations (as long as the robot boundary can be described by a piece-wise function). The collision avoidance problem is now transformed to a point moving in a two dimensional space.

IV. NON-ADMISSIBLE CONFIGURATIONS

We describe here the computation of the non-admissible configuration region CNA_{ARM} in the ARM . This region is the union of two regions:

$$CNA_{ARM} = CO_{ARM} \cup CU_{ARM} \quad (5)$$

Region CO_{ARM} is the region of configurations in collision (previous section); and CU_{ARM} is the region of unsafe configurations. This latter region contains the configurations reached with a control after a time interval that cannot be cancelled by applying maximum deceleration before collision with CO_{ARM} . In fact, the CU_{ARM} region covers the CO_{ARM} boundary. To compute CU_{ARM} , we assume that the vehicle remains on the elemental path during braking to reduce the complexity of all possible trajectories¹. A point \mathbf{p} of the CO_{ARM} boundary results in four points of the

possible CU_{ARM} boundary: two limit points \mathbf{p}_1^v and \mathbf{p}_2^v for decelerating the translation velocity in both directions of the circle, and two points for decelerating the rotational velocity \mathbf{p}_1^w and \mathbf{p}_2^w (Figure 3a). We describe the computation of these points next.

Let $\mathbf{p} = (x, y)$, a point of the CO_{ARM} boundary. Let r and θ be the radius and orientation of the tangent to the circle in \mathbf{p} [Equations (1,2)], and let L be the arc length of the circle:

$$L = \begin{cases} |x|, & \text{if } y = 0 \\ |r \cdot \theta|, & \text{otherwise} \end{cases} \quad (6)$$

Let (a_v, a_ω) be the maximum robot accelerations and T a given time interval (in practice, the sample period).

Translation

The objective is to compute the two points \mathbf{p}_1^v and \mathbf{p}_2^v of the border of CU_{ARM} for a given point \mathbf{p} of the CO_{ARM} boundary. The translational velocity contributes to the distance traveled within the circle (arc length). Thus, in one direction of the circle defined by \mathbf{p} , the point \mathbf{p}_1^v is given by:

$$\mathbf{p}_1^v = \begin{cases} (\text{sign}(x) \cdot L_{max}^v, 0), & \text{if } y = 0 \\ (r \sin \frac{\text{sign}(x) \cdot L_{max}^v}{r}, \\ r(1 - \cos \frac{\text{sign}(x) \cdot L_{max}^v}{r})), & \text{otherwise} \end{cases} \quad (7)$$

where L_{max}^v is the maximum arc length that the vehicle travels (during T and at v constant) that then allows deceleration of the vehicle before collision with \mathbf{p} (traveling an arc L_{brake}^v during braking), see Figure 3b. This arc is computed by:

$$L_{max}^v = L - L_{brake}^v \quad (8)$$

where $L_{max}^v = vT$, and $L_{brake}^v = \frac{v^2}{2a_v}$. Expanding and solving:

$$L_{max}^v = a_v T^2 \left(\sqrt{1 + \frac{2L}{a_v T^2}} - 1 \right) \quad (9)$$

Notice that if the distance traveled with a command v_1 in period T is $L_1 < L_{max}^v$, then the velocity can be cancelled before reaching \mathbf{p} .

¹With this assumption, it is possible to compute the linear and angular braking distances independently for both controls (translation v and rotation ω , which are independent for the vehicle considered). The implications of this assumption and relation to prior work will be discussed in Section IX.

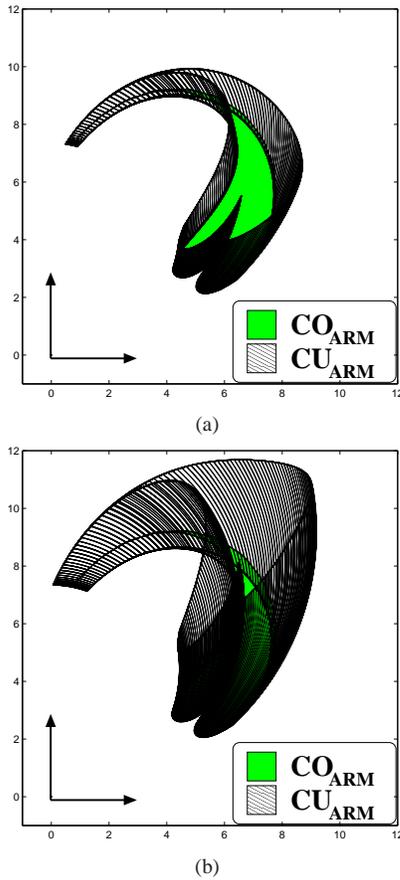


Fig. 4. These figures show the region CNA_{ARM} for an obstacle point in $(7, 6)$ and the “heart”-shape robot for two accelerations a_1 , Figure (a), and $\frac{a_1}{4}$ Figure (b). Region CU_{ARM} contains the CO_{ARM} boundary acting as a security zone. The size is larger in the second case, since the acceleration is lower (as the vehicle needs more space to brake). The region of non-admissible configurations, CNA_{ARM} , is the union of both regions.

Location \mathbf{p} can also be reached by the circle in the opposite direction (Figure 3b). Then, the other limit point \mathbf{p}_2^ω is computed as before but substituting L with $2\pi|r| - L$ in Equation (8). This calculation results in the two border points \mathbf{p}_1^ω and \mathbf{p}_2^ω of CU_{ARM} .

Rotation

The objective is to compute the two points \mathbf{p}_1^ω and \mathbf{p}_2^ω of the border of CU_{ARM} for a given point \mathbf{p} of the CO_{ARM} boundary. The rotational velocity contributes to the orientation of the tangent to the circle (angle θ) over the circle defined by \mathbf{p} . The point \mathbf{p}_1^ω is given by:

$$\mathbf{p}_1^\omega = \begin{cases} (\infty, 0), & \text{if } y = 0 \\ (r \sin(\text{sign}(y) \cdot \theta_{max}^\omega), \\ r(1 - \cos(\text{sign}(y) \cdot \theta_{max}^\omega))), & \text{otherwise} \end{cases} \quad (10)$$

where θ_{max}^ω is the maximum angular increment (obtained at constant rotational velocity ω in time T), that then allows for cancellation of ω before reaching the angle at location \mathbf{p} (the angular increment during the deceleration is θ_{brake}^ω), see Figure 3c. The angle θ_{max}^ω is:

$$\theta_{max}^\omega = \theta - \theta_{brake}^\omega \quad (11)$$

where $\theta_{max}^\omega = \omega T$ and $\theta_{brake}^\omega = \frac{w^2}{2a_\omega}$. Expanding and solving:

$$\theta_{max}^\omega = \text{sign}(\theta) \cdot a_\omega T^2 \left(\sqrt{1 + \frac{2|\theta|}{a_\omega T^2}} - 1 \right) \quad (12)$$

The angle θ_{max}^ω is the limit angle increment. If the angle increment under command w_1 in T is $\theta_1 < \theta_{max}^\omega$, the rotational velocity can be cancelled before reaching the orientation θ (this is not true if $\theta_1 \geq \theta_{max}^\omega$).

Again, location \mathbf{p} can be reached within the same circle in the opposite direction (Figure 3c). The other limit point is \mathbf{p}_2^ω , computed as before, but substituting θ by $\text{sign}(\theta)(2\pi - |\theta|)$ in Equation (11). The result is the two border points \mathbf{p}_1^ω and \mathbf{p}_2^ω of the CU_{ARM} .

From these four limit points $\{\mathbf{p}_1^\omega, \mathbf{p}_2^\omega, \mathbf{p}_1^\omega, \mathbf{p}_2^\omega\}$, the two points farther in terms of distance over the circle in both directions of the circle leading to \mathbf{p} are the borders points of the CU_{ARM} region (Figure 3a). Finally, by applying this procedure to all the border points of CO_{ARM} , we get the CU_{ARM} and thus the non-admissible configurations CNA_{ARM} (Equation (5)). Figure 4 depicts an example. It is easy to demonstrate that the region of unsafe configurations CU_{ARM} contains the bounds of the obstacle region CO_{ARM} . In fact, if we ignore the dynamics $a_v, a_\omega \rightarrow \infty$, then the CU_{ARM} tends to be the bounds of CO_{ARM} . In other words, there are no unsafe configurations since the braking distance approaches zero (infinite accelerations are assumed).

The CNA_{ARM} is computed without additional algorithmic complexity when CO_{ARM} is computed, and the procedure derived here is valid for any vehicle shape.

In summary, we have described a calculation to compute the non-admissible configuration region in the manifold ARM for a vehicle with arbitrary shape, a given dynamics and a fixed time interval (the sampling period T).

V. REACHABLE CONFIGURATIONS

The remaining aspect of the vehicle dynamics is the reachable commands: commands reachable in a short period of time given the system dynamics and the current velocity. The set of reachable commands is $RC = [v_o \pm a_v T, w_o \pm a_\omega T]$, where (v_o, w_o) is the current velocity, (a_v, a_ω) is the vehicle acceleration and T is the sample period. The set of reachable configurations RC_{ARM} in ARM is:

$$RC_{ARM} = \{\mathbf{q} \in ARM \mid \mathbf{q} = h(v, \omega), \forall (v, \omega) \in RC\} \quad (13)$$

where $h(v, \omega)$ is the function that computes the configuration reached after executing a command (v, ω) during time T :

$$h(v, \omega) = \begin{cases} (vT, 0), & \text{if } \omega = 0 \\ \left(\frac{v}{\omega} \sin(\omega T), \frac{v}{\omega} (1 - \cos(\omega T)) \right), & \text{otherwise.} \end{cases} \quad (14)$$

Notice that RC_{ARM} contains all the configurations reachable in ARM in a time T given the system dynamics and the current velocity.

VI. THE EGO-KINEMATIC COORDINATE TRANSFORMATION

This section deals with the vehicle kinematics. The original idea of this transformation is to present the motion problem in a parameterized space, in which the paths depend on parameters that identify the admissible paths and the distance traveled over these paths [29]. In the case at hand, we apply a change of coordinates of ARM so that the elemental paths become straight segments in the new coordinates (motion is omnidirectional). The change of coordinates transforms the domain of the manifold \mathbf{R}^2 into $\mathbf{R} \times S^1$. We call ARM^P to ARM in the new coordinates, where given a configuration $\mathbf{q} = (x, y) \in ARM$, the corresponding configuration is $\mathbf{q}^P = (L, \alpha) \in ARM^P$.

The first coordinate of \mathbf{q}^P is the arc length L over the circle that leads to \mathbf{q} (Equation (6)). The second coordinate α is a parameter² that uniquely represents the circle:

$$\alpha = \begin{cases} \arctan(\frac{1}{r}), & x \geq 0 \\ \text{sign}(y)\pi - \arctan(\frac{1}{r}), & \text{otherwise} \end{cases} \quad (15)$$

where r is the radius of the circle.

One important property of ARM is that given a configuration and a time period T , there is one command that leads the vehicle to this configuration at T . This is also valid for ARM^P , since a direction α uniquely determines a turning radius:

$$r = \begin{cases} \tan^{-1} \alpha, & \alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \tan^{-1}(\text{sign}(\sin \alpha) \cdot \pi - \alpha), & \text{otherwise} \end{cases} \quad (16)$$

Furthermore, given a time T , one can compute the command (v, ω) that preserves r and moves the vehicle a distance L over this circle:

$$(v, \omega) = (\text{sign}(\cos \alpha) \frac{L}{T}, \text{sign}(\sin \alpha) |\tan \alpha| \frac{L}{T}) \quad (17)$$

A location in ARM^P is given by a direction and a distance on this direction. The elemental paths in ARM^P are thus rectilinear (omnidirectional motion), whereas they represent circular paths in ARM (kinematic admissible paths in the workspace). That is, we represent ARM in a new coordinate system where the motion is omnidirectional. Furthermore, given a location $\mathbf{q}^P \in ARM^P$ and a time T , one can compute the kinematic admissible motion command that moves the vehicle a distance L over the circle of radius r (defined by α) in the workspace.

VII. ABSTRACTION OF THE SHAPE, KINEMATICS AND DYNAMICS FROM THE OBSTACLE AVOIDANCE METHODS

In this section we describe how to use the previous results to abstract the vehicle shape, kinematics and dynamics from the obstacle avoidance methods. These methods follow a cyclic process: given an obstacle description and a target location they compute a target-oriented collision-free motion. The motion is executed by the vehicle and the process restarts. The idea behind the abstraction is to include two steps prior

(incorporation of the shape, kinematics and dynamics) and one subsequent (motion computation) to the application of the method (Figure 1). At each iteration, given the sensor information (obstacles) and a target location, the process is:

- 1) Shape and dynamics: Computation of the non admissible configuration region CNA_{ARM} and reachable region RC_{ARM} (Sections IV and V).
- 2) Kinematics: change of coordinates of ARM , where CNA_{ARM}^P and RC_{ARM}^P are the previous regions in the new coordinates (Section VI).
- 3) Obstacle avoidance: application of the obstacle avoidance method in ARM^P to compute the most promising motion direction β_{sol} .
- 4) Motion: computation of the closest configuration \mathbf{q}_{sol}^P to β_{sol} that is reachable and admissible, i.e. $\mathbf{q}_{sol}^P \in RC_{ARM}^P$ and $\mathbf{q}_{sol}^P \notin CNA_{ARM}^P$. Once \mathbf{q}_{sol}^P is obtained, the motion command is given by Equation (17). To compute \mathbf{q}_{sol}^P , we first obtain the set of configurations S_{sol} closest to β_{sol} :

$$S_{sol} = \arg \min_{\mathbf{q}^P \in RC_{ARM}^P, \mathbf{q}^P \notin CNA_{ARM}^P} \|\mathbf{q}^P - \mathbf{p}^P\| \quad (18)$$

, where \mathbf{p}^P is the configuration projection of \mathbf{q}^P over the unitary vector in the direction of β_{sol} . When $|S_{sol}| = 1$, there is only one possible configuration that we select as solution \mathbf{q}_{sol}^P . Otherwise, we select in S_{sol} the closest to the target \mathbf{q}_{target}^P :

$$\mathbf{q}_{sol}^P = \arg \min_{\mathbf{q}^P \in S_{sol}} \|\mathbf{q}^P - \mathbf{q}_{target}^P\| \quad (19)$$

Figure 5 shows this process using a rectangular vehicle with differential traction and a generic obstacle avoidance method (assumes a robot without constraints, that is point-like and omnidirectional). At a given time, the robot collects the sensor information about the obstacles and a target location (Figure 5a). The objective is to compute a motion command free of collisions that moves the vehicle towards the target (taking into account shape, kinematic and dynamic constraints). The steps are:

- 1) Computation of the reachable and non-admissible configuration regions, RC_{ARM} and CNA_{ARM} , in ARM (Figure 5b). In this manifold, the robot is a point and the effects of the dynamics are represented in the manifold.
- 2) Change of coordinates of ARM (Figure 5c). In ARM^P , the robot is a point and the motion is omnidirectional (straight paths), which are the applicability conditions of many obstacle avoidance methods.
- 3) Application of the obstacle avoidance method to obtain the motion direction β_{sol} that avoids the non-admissible regions CNA_{ARM}^P while moving the vehicle towards the target \mathbf{q}_{target}^P (Figure 5c).
- 4) This direction is used to select a configuration $\mathbf{q}_{sol}^P \in RC_{ARM}^P$, which results in a motion command (v_{sol}, ω_{sol}) using Equation (17). Figure 5d shows this command in the vehicle velocity space.

By construction, this command is goal-oriented, free of collisions, complies with the kinematics, and is dynamically reachable and admissible. Notice that in this methodology, the

²From a physical point of view, α is the angle of a free wheel, located at a distance l from the origin on the X -axis, which aligns tangent to the circle of motion with radius r .

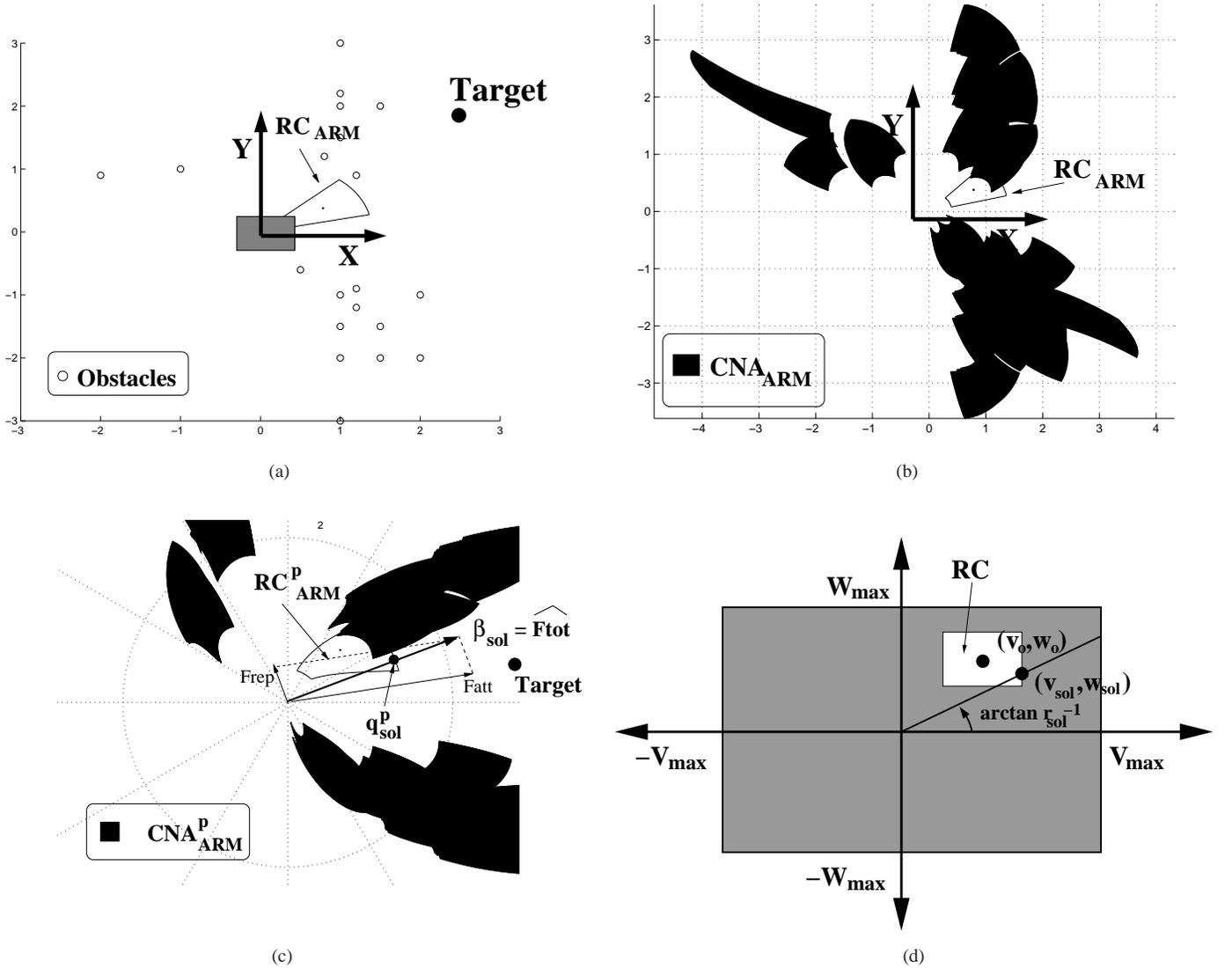


Fig. 5. These figures show the usage of the abstraction technique to take into account the shape, kinematics and dynamics of the vehicle in the application of the avoidance method. (a) Rectangular vehicle and obstacle distribution. (b) The reachable configurations, RC_{ARM} , and the non admissible region, CNA_{ARM} , in the ARM . (c) Change of coordinates of ARM to ARM^P . In ARM^P , the robot is a point and the motion is omnidirectional. Then, the avoidance method is applied to obtain the most promising direction β_{sol} (that avoids the obstacles while moving the current location towards the target). In the example a potential field method is used: the most promising motion direction $\beta_{sol} = \widehat{F_{tot}}$ is obtained by $F_{tot} = F_{rep} + F_{att}$, where the obstacles exert a repulsive force F_{rep} and the target an attractive one F_{att} . This direction, β_{sol} , is then projected to the RC_{ARM}^P to obtain the configuration solution q_{sol}^P . (d) Finally, given q_{sol}^P , the solution command v_{sol} is computed by Equation (17), which can be shown in the velocity space.

modification introduced with respect to the direct application of the method is a change of spatial representation. However, the method solutions working in this representation take into account the vehicle constraints. In other words, the collision avoidance method has been extended to address the vehicle constraints. This is the main contribution of this work.

In the next section, we show how we have used this scheme to apply a given obstacle avoidance method in a real vehicle.

VIII. EXPERIMENTAL RESULTS

In this section, we validate the proposed methodology with a collision avoidance method working on a real vehicle with rectangular shape, kinematic and dynamic constraints (differential-drive). First, we describe the vehicle, the sensor

and the collision avoidance method, and then we discuss the experimental results.

A. Vehicle, Sensor and Collision Avoidance Method

The vehicle is a robot built from a commercial wheelchair in our laboratory (Figure 6). The vehicle is rectangular ($1.2m \times 0.8m$) and differential-drive with drive wheels in the back. We have installed two computers, Intel 800MHz, on board, one for control and the other for higher-level purposes (in which the collision avoidance technique is executed). The sensor is a planar laser that works at 5Hz with a field of view of 180° and 0.5° resolution (361 points) placed in the front. We put a weight of 60kg on the wheelchair to simulate a seated person.

In all the experiments, the scenario was unknown, dynamic with an unpredictable behavior and unstructured. Under these

conditions, a collision avoidance method is the correct choice to reactively move the vehicle. We selected a potential field method (PFM in short) [17], since it is a formal method widely known and used. In the PFM, the robot is modeled as a particle moving in the configuration space affected by a field of forces. The target location exerts a force that attracts the particle while the obstacles exert repulsive forces. The motion is computed to follow the direction of the artificial force resulting from the composition of these forces (most promising motion direction).

This method cannot be applied to a differential-drive robot without approximations. This is because the direction of the force does not satisfy the non-holonomic constraint. In other words, the structure of the potential field does not represent the fact that not all the motions are allowed in the configuration space. Furthermore, to take into account the vehicle geometry would imply construction of the obstacle representation in the three-dimensional configuration space, which would be difficult to execute in real time. Finally, although the generation of reachable commands could be done with a force control [17], [37], including the braking distance (admissible commands) in the formulation of a PFM has only been done in works related to the abstraction layer presented here [28]. Due to these facts, the usage of a PFM for obstacle avoidance usually assumes a point-like vehicle (the shape is ignored) that can move in any direction (omnidirectional without dynamics).

Notice that these assumptions are very relevant for the kind of vehicle. The approximation of a rectangular geometry by a point or a circle is not realistic, because the motor-wheels are in the back (when the vehicle turns, it sweeps a large area that must be taken into account). Due to the kinematics, the vehicle moves on arcs of circles, and therefore, to assume omnidirectional motion is a gross approximation that could put the safety at risk. The dynamics play an important role because if they are ignored: (i) the motion planned cannot be feasible, again putting the safety at risk; (ii) shaking behaviors will arise, making it uncomfortable for the end user; and (iii) the vehicle slips in detriment of the odometry and of the system in general. In other words, the wheelchair vehicle leads to work conditions where it is very important to take into account the vehicle constraints.

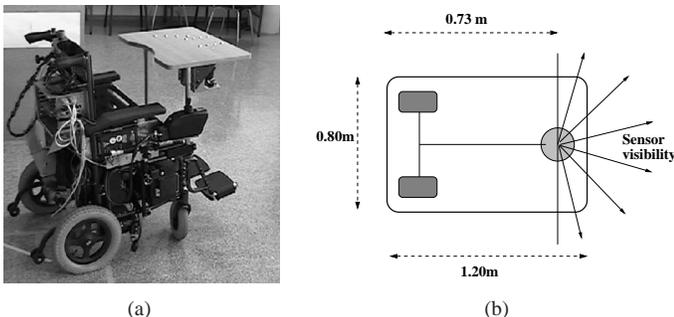


Fig. 6. (a) The robot is a rectangular wheelchair vehicle with differential-drive traction and equipped with a SICK laser. (b) Scheme of the distribution of the wheels and sensor in the vehicle.

B. Experiments

In the experiments, we fixed the sampling period T to 0.2sec (5Hz is the frequency of the laser). This period is a maximum bound for the computation time of the algorithm³. The maximum accelerations of the vehicle are $(a_v, a_\omega) = (0.6 \frac{m}{sec^2}, 0.6 \frac{rd}{sec^2})$ and we fixed the maximum velocities to $(v_{max}, w_{max}) = (0.3 \frac{m}{sec}, 0.8 \frac{rd}{sec})$, that are not very high due to the robotic application (human transportation).

In the experiments, there were three aspects to test: (i) the collision avoidance task is carried out with the method using the abstraction layer. That is, the vehicle is driven to the target whilst collisions with the obstacles are avoided. (ii) The motion computed takes into account the shape, kinematics and dynamics of the vehicle. (iii) If the abstraction is not used, the PFM method computes solutions that cannot be executed without approximations.

General obstacle avoidance task with abstraction

Figure 7 depicts two experiments carried out in scenarios in which a human was randomly placing obstacles in order to hinder the wheelchair motion (unknown, dynamic, unpredictable and unstructured scenarios). The difference between the experiments are the settings. Experiment 1 had more obstacle density (more difficulty to manoeuvre), while the second one is more dynamic (unpredictable). In both cases, the vehicle reached the target location without collisions (see the vehicle trajectory and laser points gathered). That is, the introduction of the abstraction layer did not penalize the work of the method avoiding obstacles. The shape, kinematics and dynamics of the vehicle were taken into account at all times during the experiment. As a result, the vehicle successfully achieved the avoidance task. Notice that ignoring these constraints, could heavily penalize the obstacle avoidance with this vehicle (Subsection VIII-A), and it is doubtful that it would reach the target otherwise. The durations of the trials were 43sec and 41sec, the mean velocities were $0.18 \frac{m}{sec}$ and $0.12 \frac{m}{sec}$ and $0.24 \frac{rad}{sec}$ and $0.14 \frac{rad}{sec}$ (see in Figure 8 the velocity profiles of the reference commands and the real vehicle behavior).

Shape, kinematics and dynamics in obstacle avoidance

We next describe how the vehicle restrictions were taken into account during the experiments. The commands computed by the method are always kinematically admissible, because they result from admissible circular paths. This is because the avoidance method is applied in the ARM^P manifold, where directions correspond to turning radius. The motion command solution is the command that performs this turn. For instance, Figure 9 depicts one step of the application of the PFM in the ARM^P during one trial.

In order to address the vehicle dynamics, the method computes commands that are reachable in a short period of time and that take the braking distance. On the one hand, the commands computed are reachable since the avoidance method computes a direction solution β_{sol} in the ARM^P , which is used to then select a location in RC_{ARM}^P (that contains the configurations that can be reached in time T in

³The computation time is very variable because it depends on the number of obstacle points measured (ranging from 0 to 361). Experimentally, we observed that this period was closely an upper bound of the computation time.

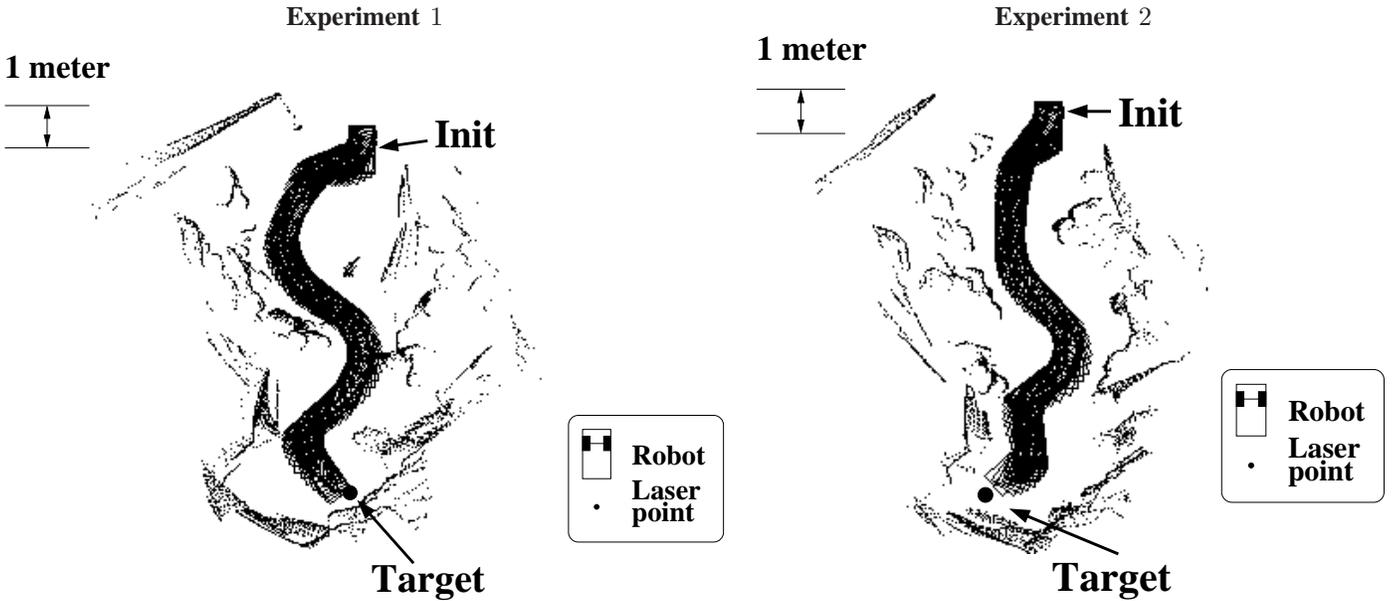


Fig. 7. Experiment 1 and Experiment 2. The path executed by the vehicle, the laser points gathered during the execution and a snapshot of the experiment.

the ARM^P , given the system dynamics). Figure 8 depicts the translational and rotational velocity profiles of the trials. Notice how the commands were reachable, because given one command, the following one is always in RC . As a consequence, the vehicle closely executes the planned motion.

On the other hand, the motion commands assured that the vehicle could be stopped without collision by applying the maximum deceleration (the braking distance is taken into account). This is because the commands are computed using admissible configurations, i.e. configurations that are not in CNA_{ARM}^P . We did not observe any emergency stop in the experiments since the PFM avoided the CNA_{ARM}^P regions as obstacles with good safety margins. In fact, this is a desirable behavior, because the configurations in the vicinity of the CNA_{ARM}^P are closed to become unsafe. This fact is much more important in vehicles with slow dynamics, as reported in [28]. The selection of admissible commands makes the method conservative. However, the method's security is increased, because there is always the guarantee of stopping the vehicle safely.

The last aspect to address is the vehicle shape, which was taken into account jointly with the kinematics and dynamics. This is because the method avoids collisions with the CNA_{ARM}^P , which is constructed taking into account the exact shape of the vehicle as well as the kinematics and dynamics. As a result, the general effect of dealing with these three aspects simultaneously is that the vehicle executes the planned motion that is collision-free. This allows the robot to maneuver in scenarios with high density of obstacles (Figure 7).

Remarks about the abstraction

Another issue is to discuss is how, without the proposed methodology, the PFM avoidance method computes solutions that approximately take into account the vehicle aspects (this was already discussed theoretically in subsection VIII-A). Figure 9 shows an example. The solution of the PFM obtained without abstraction is a motion direction in the workspace,

which cannot be executed with this vehicle without approximations.

Let us mention that the abstraction layer is a technique that allows for using some methods on vehicles, but it does not ameliorate the quality of one method in itself. If a given method has some difficulties under certain conditions, they would also be present using the abstraction. For example, a difficulty of the PFM is to drive a vehicle between very close obstacles or the instabilities and oscillations in the motion [19]. We observed these difficulties when using the PFM with the abstraction. However the opposite is also true, and if a method performs well under certain conditions, the abstraction does not penalize it (see [29], [28] for a discussion on this topic with another collision avoidance method).

In summary, in this section, we have presented the integration of a PFM method with the proposed technique working with a real vehicle. First, we have seen how the method with abstraction is able to solve the collision avoidance task. The scenarios of application were unknown, dynamic with an unpredictable behavior, and moderately dense. Second, we have seen how the shape, kinematics and dynamics are taken into account during the application of the method, although the method does not address these issues in its formulation. Third, we have discussed how the solution of the original method without the abstraction layer cannot be executed without approximations. The proposed solution has demonstrated efficiency for the wheelchair application at hand.

IX. DISCUSSION AND CONCLUSIONS

In this work, we have presented a general scheme to extend collision avoidance methods for addressing the shape, kinematics and dynamics of the vehicle. The most important aspect of this work is the generality. This is because, with this framework, existing methods could be reutilized on a wide variety of any-shape non-holonomic vehicles without an extra design or implementation effort.

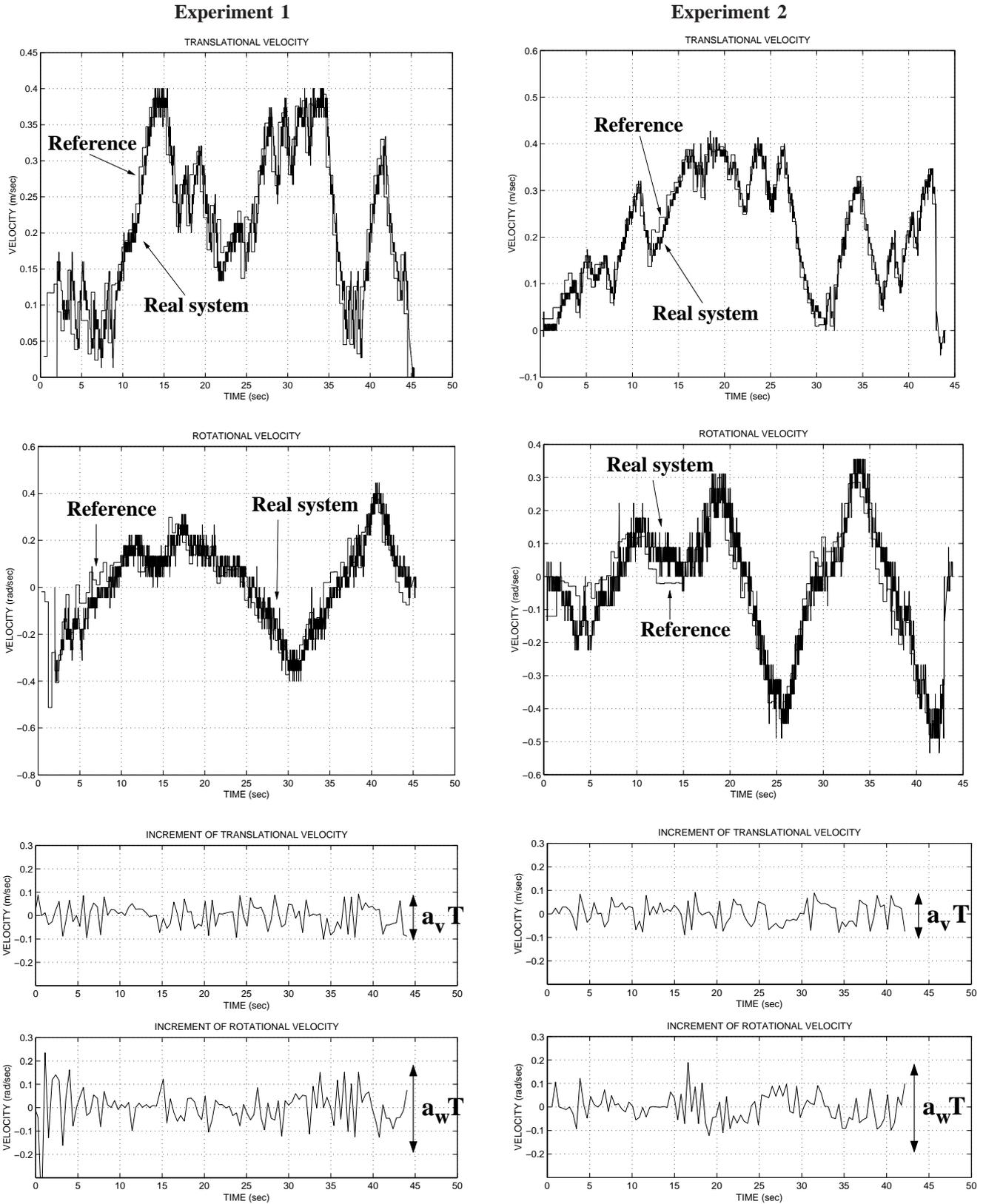


Fig. 8. This Figure shows for each experiment the translational and rotational velocity profiles: (first two rows) the commands computed and real behaviour of the system; (last row) the translational and rotational velocity increment profile.

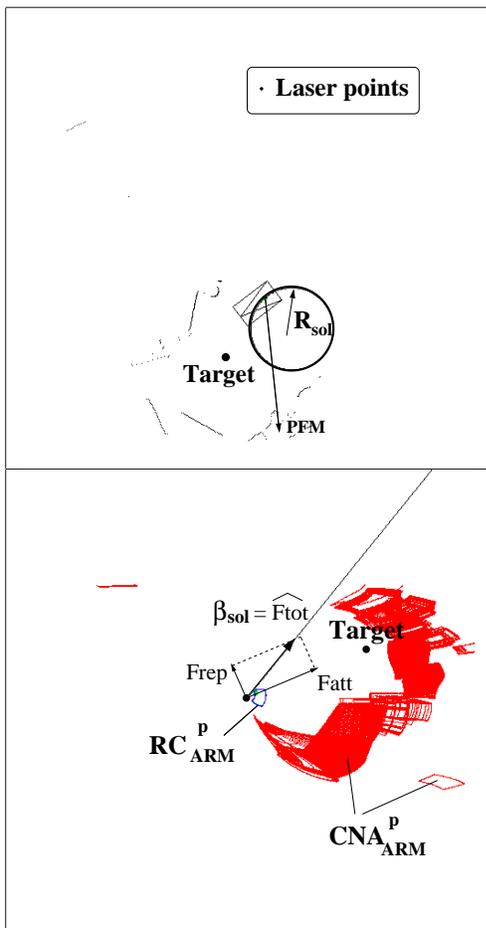


Fig. 9. This Figure shows a time instant of the execution of the method. (Top) the workspace: robot and laser sensor measurements; and (bottom) the ARM^P . The PFM is applied in the ARM^P where $\beta_{sol} = \widehat{F}_{tot}$, which represents a turning radius R_{sol} in the workspace. Notice that if the PFM would be applied in the ARM^P , the direction solution could not be directly executed by the vehicle (PFM in the figure).

A. Comparison with other Methods

This generality is the advantage with respect to existing techniques, because: (i) some have been constructed ad-hoc to take into these constraints [13], [35], [12], [38], [15]. Thus it appears difficult to reutilize these strategies to extend other methods. And (ii) some other techniques have been developed with the same objective here, but they take into account the constraints only after the method application [22], [5], [4], [24]. Thus, although the scope of application is broad, the solution is an approximation.

Let us now discuss the benefits of this approach with respect to some widely known collision avoidance techniques in theoretical terms. The techniques that consider these restrictions compute collisions either over a set of elemental circular paths [16], [12], [38], [15], or over a set of commands (where each one corresponds to a circular path) [13], [35], [33]. 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 has a closed form solution [13], [3]. However, these techniques do not generalize for arbitrary shapes. For instance, in the example of the heart-shape vehicle, the aforementioned techniques must solve the system formed by Equation (4) and $x^2 + (y - R)^2 = (R - c)^2$ (where c depends on the obstacle point and R is the radius of the inspected path); this system has no closed-form solution. Although one could solve the system by a numerical method or by projecting the robot position onto the path checking collisions (dynamic simulation), both strategies increase complexity (computation time) and would lead to an approximate solution. To address the complexity and efficiency issues, some researchers precompute the collisions with a look-up table [33] (the complexity factor becomes $N \times C$ since the M is computed off-line). However, the continuous obstacle space is discretized and the problem of the exact calculation for any arbitrary vehicle shape persists.

In this work, the procedure to compute the region of the configuration space in collision over the manifold of circular paths has a $N \times M$ complexity. This complexity factor is lower than existing methods, but more importantly, 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 calculation allows for maintaining a continuous representation of the space of solutions (that is why the term C does not appear in the complexity factor). Existing methods could benefit from this procedure to reduce complexity, to consider any vehicle shape in a straight-forward way and to avoid discretization of the space of solutions.

An assumption made in this work (and in all works that take into account the braking distance [13], [35], [10], [6], [3]) is that the braking is carried out on an elemental path. This assumption reduces the complexity of taking into account all the trajectories derived from braking. Previous methods compute an approximation of the bounds of the non admissible configuration region and have been used only over circular or polygonal vehicle shapes [13], [35], [10], [6], [3]. However, the calculation presented here computes the exact bounds of this region (with the same assumptions) and is valid for any vehicle shape.

B. Final Remarks

This work, as all works that compute admissible commands, is conservative [13], [35], [33], [10], [6], [3]. This is because only commands that allow the vehicle to stop safely are selected. As a result, the motions obtained are smooth and slow (since a subset of the control space is used). However, the motion gains security, because the possibility to safely stop the vehicle is always present (which is especially relevant in applications such as human or dangerous material transportation, motion at high speeds or systems with slow dynamic capabilities).

One important choice in the paper is the focus on circular elemental paths. This is done to reduce the search space of all possible trajectories as in [13], [35], [10], [6], [3]. However, extensions of this research have explored the usage

of combinations of different elemental paths (manoeuvres) [7]. In this work 5 families of paths were used. A related issue is the assumption that the breaking is carry out on an elemental path. This approximation allows for avoiding the consideration of all the possible trajectories derived from breaking, as in [13], [35], [10], [6], [3]. However, some extensions for more complex breaking paths are found in [14], [34].

Let us mention that the collision avoidance methods are local techniques to solve the motion problem, so cyclical motions and trap situations persist. This is a common disadvantage of these methods. Nevertheless, movement is improved in terms of flexibility, adaptation and robustness in unknown, unstructured and dynamic surroundings with a priori unpredictable behavior (the sensory information is included at a high frequency in the motion control loop). 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 its local nature. In order to deal with the locality of collision avoidance methods, hybrid systems should be developed (see [2] for a discussion on different architectures and [27] for a similar discussion in the motion context). These systems are made up of a global deliberation module (planning) and an obstacle avoidance module (avoidance of collisions), whose synergy generates motion while avoiding the trap situations [39], [10], [36], [31], [3], [32], [11].

A limitation of the approach is that it generates suboptimal paths that sometimes are very contra-intuitive. This is because the solutions are computed over elemental circular paths and sometimes to turn in place and move straight is a much better solution. This effect becomes more significant as the target locations are farther from the robot location. Although this is common for many obstacle avoidance techniques that deal with circular paths, we suggest to place target locations close to the vehicle to mitigate local minima and these suboptimality. An extension of our technique has dealt with this issue [7].

Our belief is that this technique could be very useful to many researchers since it provides a framework to improve the robustness of the collision avoidance methods without significant modifications. In this work, we have used this method to extend a standard potential field method to work on a wheelchair vehicle. The results confirm that the avoidance task is successfully carried out while jointly taking into account the shape, kinematics and dynamics of the robot.

X. 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 also to the Robotics and Manipulation group directed by O. Khatib that hosted Javier Minguez visit to the Stanford University, USA. 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. Lamiroux for their fruitful commentaries and discussions during the preparation of this manuscript. We also thanks the comments of J.D. Tardós of the Robotic, Perception and Real Time group at the Universidad of Zaragoza, Spain. This work was partially supported

by spanish project DPI2006-15630-C02-02, DPI2006-07928 and EU project IST-1-045062.

REFERENCES

- [1] J. Alvarez, A. Shkel, and V. Lumelsky. Building topological models for navigation in large scale environments. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.
- [2] R. Arkin. *Behavior-Based Robotics*. The MIT Press, 1999.
- [3] K. Arras, J. Persson, N. Tomatis, and R. Siegwart. Real-time Obstacle Avoidance for Polygonal Robots with a Reduced Dynamic Window. In *IEEE Int. Conf. on Robotics and Automation*, pages 3050–3055, Washington, USA, 2002.
- [4] J. Asensio and L. Montano. A Kinematic and Dynamic Model-Based Motion Controller for Mobile Robots. In *15th IFAC World Congress*, Barcelona, Spain, 2002.
- [5] J. A. Bemporad, A. D. Luca, and G. Oriolo. Local incremental planning for car-like robot navigating among obstacles. In *IEEE International Conference on Robotics and Automation*, pages 1205–1211, Minneapolis, USA, 1996.
- [6] J. A. Beyanas, J. Fernandez, R. Sanz, and A. Dieguez. The Beam-Curvature Method: a New Approach for Improving Local Real-Time Obstacle Avoidance. In *15th IFAC World Congress*, Barcelona, Spain, 2002.
- [7] J. Blanco, J. Gonzalez, and J. Fernandez-Madrugal. The Trajectory Parameter Space (TP-Space): A New Space Representation for Non-Holonomic Mobile Robot Reactive Navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1195–1200, 2006.
- [8] J. Borenstein and Y. Koren. Real-Time Obstacle Avoidance for Fast Mobile Robots. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):1179–1187, 1989.
- [9] J. Borenstein and Y. Koren. The Vector Field Histogram—Fast Obstacle Avoidance for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 7:278–288, 1991.
- [10] O. Brock and O. Khatib. High-Speed Navigation Using the Global Dynamic Window Approach. In *IEEE Int. Conf. on Robotics and Automation*, pages 341–346, Detroit, MI, 1999.
- [11] O. Brock and O. Khatib. Real-Time Replanning in High-Dimensional Configuration Spaces using Sets of Homotopic Paths. In *IEEE Int. Conf. on Robotics and Automation*, pages 550–555, San Francisco, USA, 2000.
- [12] W. Feiten, R. Bauer, and G. Lawitzky. Robust Obstacle Avoidance in Unknown and Cramped Environments. In *IEEE Int. Conf. on Robotics and Automation*, pages 2412–2417, San Diego, USA, 1994.
- [13] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics and Automation Magazine*, 4(1), 1997.
- [14] T. Fraichard and H. Asama. Inevitable collision states. a step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004.
- [15] A. Hait, T. Simeon, and M. Taix. Robust motion planning for rough terrain navigation. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 11–16, Kyongju, Korea, 1999.
- [16] M. Hebert, C. Thorpe, and A. Stentz. *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*. Kluwer Academic Publishers, 1997.
- [17] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. Journal of Robotics Research*, 5:90–98, 1986.
- [18] N. Ko and R. Simmons. The lane curvature velocity method for local obstacle avoidance. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages –, Victoria, Canada, 1998.
- [19] Y. Koren and J. Borenstein. Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1398–1404, Sacramento, CA, 1991.
- [20] B. H. Krogh and C. E. Thorpe. Integrated Path Planning and Dynamic Steering control for Autonomous Vehicles. In *IEEE Int. Conf. on Robotics and Automation*, pages 1664–1669, San Francisco, USA, 1986.
- [21] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic, 1991.
- [22] A. D. Luca and G. Oriolo. Local incremental planning for nonholonomic mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 104–110, San Diego, USA, 1994.
- [23] V. Lumelsky and A. Shkel. Incorporating body dynamics into the sensor-based motion planning paradigm. the maximum turn strategy. In *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 1995.
- [24] J. Minguez and L. Montano. Robot navigation in very complex and cluttered indoor/outdoor scenarios. In *15th IFAC World Congress*, Barcelona, Spain, 2002.

- [25] J. Minguez and L. Montano. The Ego-KinoDynamic Space: Collision Avoidance for any Shape Mobile Robots with Kinematic and Dynamic Constraints. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 637–643, Las Vegas, USA, 2003.
- [26] J. Minguez and L. Montano. Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, 2004.
- [27] J. Minguez and L. Montano. Sensor-based robot motion generation in unknown, dynamic and troublesome scenarios. *Robotics and Autonomous Systems*, 52(4):290–311, 2005.
- [28] J. Minguez, L. Montano, and O. Khatib. Reactive Collision Avoidance for Navigation at High Speeds or Systems with Slow Dynamics. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 588–594, Lausanne, Switzerland, 2002.
- [29] J. Minguez, L. Montano, and J. Santos-Victor. Abstracting the Vehicle Shape and Kinematic Constraints from the Obstacle Avoidance Methods. *Autonomous Robots*, 20(1):43–59, 2006.
- [30] P. Ogren and N. Leonard. A tractable convergent dynamic window approach to obstacle avoidance. In *IEEE International Conference on Robotics and Automation*, pages –, Laussane, Switzerland, 2002.
- [31] R. Philippsen and R. Siegwart. Smooth and efficient obstacle avoidance for a tour guide robot. In *IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003.
- [32] S. Quinlan and O. Khatib. Elastic Bands: Connecting Path Planning and Control. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 802–807, Atlanta, USA, 1993.
- [33] C. Schlegel. Fast Local Obstacle Avoidance under Kinematic and Dynamic Constraints for a Mobile Robot. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Canada, 1998.
- [34] C. Schlegel. *Navigation and Execution for Mobile Robots in Dynamic Environments: An Integrated Approach*. PhD thesis, Universität Ulm, 2004.
- [35] R. Simmons. The Curvature-Velocity Method for Local Obstacle Avoidance. In *IEEE Int. Conf. on Robotics and Automation*, pages 3375–3382, Minneapolis, USA, 1996.
- [36] C. Stachniss and W. Burgard. An Integrated Approach to Goal-directed Obstacle Avoidance under Dynamic Constraints for Dynamic Environments. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 508–513, Switzerland, 2002.
- [37] R. B. Tilove. Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 566–571, Cincinnati, OH, 1990.
- [38] I. Ulrich and J. Borenstein. VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 1572–1577, 1998.
- [39] I. Ulrich and J. Borenstein. VFH*: Local Obstacle Avoidance with Look-Ahead Verification. In *IEEE Int. Conf. on Robotics and Automation*, pages 2505–2511, San Francisco, USA, 2000.
- [40] T. Wilkman, M. Branicky, and W. Newman. Reflexive Collision Avoidance: A Generalized Approach. In *IEEE Int. Conf. on Robotics and Automation*, Atlanta, USA, 1993.