Article information:

Publication: IEEE Transactions on Robotics

Paper Title: Extending Collision Avoidance Methods to Consider the Vehicle Shape, Kinematics, and Dynamics of a Mobile Robot

Type of paper: Regular paper

Abstract:
Most collision avoidance methods do not consider the vehicle shape and its kinematic and dynamic constraints, assuming the robot to be point-like and omnidirectional with no acceleration constraints. The contribution of this paper is a methodology to consider the exact shape and kinematics, as well as the effects of dynamics, in the collision avoidance layer, since the existing avoidance method does not address them. This is achievable by abstracting the constraints from the avoidance methods in such a way that, when the method is applied, the constraints already have been considered. This work is a starting point to extend the domain of applicability to a wide range of collision avoidance methods.

Keywords: Reactive Collision Avoidance, Mobile Robots.

Authors information:

- *Author 1*: Javier Minguez (corresponding author)

  Address: Departamento de Informática e Ingeniería de Sistemas, Centro Politécnico Superior, Universidad de Zaragoza, María de Luna 3, 50015, Zaragoza, Spain

  Email: jminguez@unizar.es

  Telf: +34 976 762350

- *Author 2*: Luis Montano

  Address: Departamento de Informática e Ingeniería de Sistemas, Centro Politécnico Superior, Universidad de Zaragoza, María de Luna 3, 50015, Zaragoza, Spain

  Email: montano@unizar.es

  Telf: +34 976 761954

Extending Collision Avoidance Methods to Consider the Vehicle Shape, Kinematics, and Dynamics of a Mobile Robot

Javier Minguez and Luis Montano

Abstract— **Most collision avoidance methods do not consider the vehicle shape and its kinematic and dynamic constraints, assuming the robot to be point-like and omnidirectional with no acceleration constraints. The contribution of this paper is a methodology to consider the exact shape and kinematics, as well as the effects of dynamics in the collision avoidance layer, since the original avoidance method does not address them. This is achievable by abstracting the constraints from the avoidance methods in such a way that, when the method is applied, the constraints already have been considered. This work is a starting point to extend the domain of applicability to a wide range of collision avoidance methods.**

Index Terms— **Reactive Collision Avoidance, Obstacle Avoidance, Mobile Robots.**

I. Introduction

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

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

The work described here centers on the consideration of the vehicle shape, as well as kinematic and dynamic constraints, during the application of a collision avoidance method. The idea is to project distance measurements into a space in which the robot can be regarded as a holonomic point. The projection accounts for collision constraints as well as for kinematic and dynamic motion constraints (the trajectories are restricted to a family of circular arcs). In this space, many reactive collision avoidance methods can be applied to the holonomic point, as all constraints are encoded in the obstacles and space itself. The computed motion command is projected back and applied to the robot. Therefore, the proposed method encompasses a complete set of well-known obstacle avoidance approaches to consider the vehicle shape, as well as the kinematic and dynamic constraints. This method has been demonstrated in real-world experiments by *wrapping* a potential field method to perform obstacle avoidance on a differentially-driven wheelchair.

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.

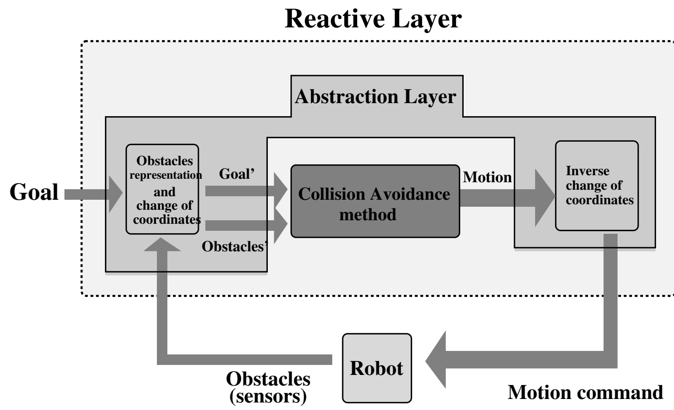


Fig. 1. The abstraction layer abstracts shape, kinematics, and dynamics of the vehicle from the avoidance method. The idea is to understand the method as a "black-box" and to modify the representation of its inputs, so that they have implicit information about these restrictions. The method is applied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

II. Related Work and Contributions

Classically, the mobility problem has been addressed by computing a geometric path that is potentially collision-free [22]. Nevertheless, when the surroundings are unknown or evolve, these techniques fail, since a pre-computed 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). Locality is the main issue when considering sensor information (the reality of the situation) during execution of a task. In such an instance, if global reasoning is required, a trap-situation can occur. Despite this limitation, collision avoidance techniques are mandatory in dealing with mobility problems in unknown and dynamic surroundings.

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

The collision avoidance problem with these constraints was considered 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 group of methods, some have been designed to solve the problem in the velocity space [14], [37]. They first compute the set of reachable commands in a short period of time, which are collision-free and allow the vehicle to stop safely. Next, they select one command 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 [32], [10], [3], [19], [6], [35]. Other methods precompute a set of collision-free circle arcs (elemental paths), that are a result of reachable commands; they then select one arc based on obstacle avoidance and convergence to the goal criteria [41], [17], [13], [16]. In general, all of these methods take into account 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 can require the use of a numerical method or a dynamic simulation (projecting vehicle positions over admissible paths) to check for collisions. That is why these methods are used on basic vehicle shapes (circular [32], [10], [19], [6], [41] or polygonal [35], [17], [13], [3]). These techniques are not generic in the sense that it is difficult to extrapolate such strategies to use them with existing methods (ad-hoc methods).

In the second group of methods, the solution of the obstacle avoidance problem is converted into 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 [24], [5]. A similar solution is proposed by breaking down the problem into subproblems (collision avoidance, kinematics and dynamics, and shape) and dealing with them sequentially [26]. Another approach is based on command filters [42]; after using the avoidance method, the commands that are not reachable or that do not avoid collisions are filtered and converted into commands that are reachable and collision-free. Other works in this direction propose a simple vehicle model and utilize control theory to compute the collision-free commands [1], [25]. The advantage of these strategies is the generality, since they can be used by many avoidance methods. However, the generated motions take into account only 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 modified to satisfy kinematics and dynamics. Thus, the final command does not guarantee avoidance with an exact shape. This leads to problems when the holonomic solution cannot be approximated or when maneuverability is a determinant factor [5].

In fact, the majority of collision avoidance methods does not consider the vehicle constraints mentioned. They assume a point-like and omnidirectional vehicle with no acceleration constraints. The **main contribution** of this work is a scheme to consider the exact shape and kinematics, as well as the effects of dynamics (reachable and admissible commands) in the collision avoidance layer. The idea is to abstract these constraints from the usage of avoidance methods (Figure 1). This technique can be applied to many vehicles with arbitrary shapes (this approach is illustrated with a differentially-driven rectangular robot).

The construction of this abstraction layer is constituted of three parts that correspond to the **three contributions** of this study:

- Firstly, the two-dimensional manifold of the three-dimensional configuration space defined by elemental circular paths is constructed, centralized on the robot. 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 in this manifold for any vehicle shape (i.e., the configurations in collision). In this manifold, a point represents the vehicle.
- Secondly, the exact calculation of the admissible configurations is described, which result from the obstacle regions computed previously (with the assumption that the braking path is a circular elemental path, typical in obstacle avoidance). Furthermore, the reachable configurations obtained by reachable commands in the manifold are represented. The effect of dynamics is represented in the manifold.
- Thirdly, a change of coordinates in the manifold is proposed, so that the circular paths become straight segments. With the manifold represented in such coordinates, the motion is free of kinematic constraints.

As a result, the three-dimensional collision avoidance problem with shape, kinematics, and dynamics is transformed into a simple problem of moving a point in a two-dimensional space with no constraints (usual approximation in collision avoidance). Thus, methods that ignore these constraints become applicable.

With this technique, many existing or future avoidance methods can be applied 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 method [18], [21], [39], [8], Vector Field Histogram [9], [40], or Nearness Diagram Navigation [28]. To validate the technique, a potential

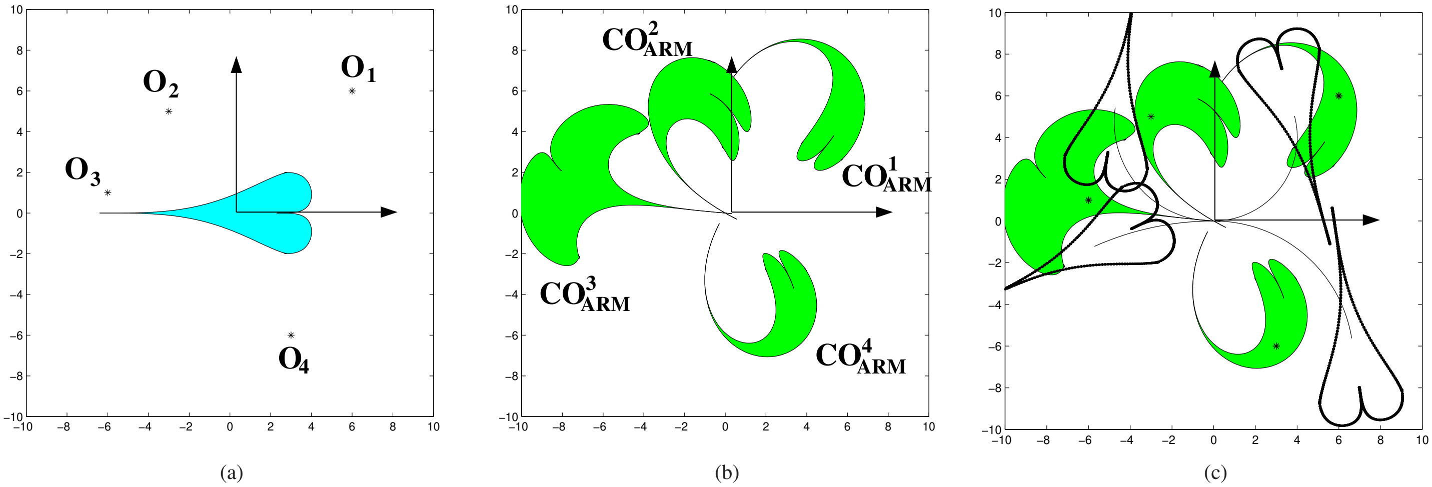


Fig. 2. This Figure shows the computation of the region of collision configurations for a heart-shaped robot that moves in circular paths. (a) Robot and obstacles O_i; (b) each obstacle point creates a region of collision locations $\mathrm{CO}^i_{\mathrm{ARM}}$ that all together are $\mathrm{CO}_{\mathrm{ARM}}$. The free space is the space outside these regions and all locations within these regions are in route of collision; and (c) superposition of workspace and ARM, as well as a few robot locations and the paths leading to them. Notice how the locations out of $\mathrm{CO}_{\mathrm{ARM}}$ are not in collision with the obstacle points.

field method [18] was utilized, integrated into a real platform (differentially-driven and rectangular).

Partial and previous results of this research were presented in [31], [30], [27]. This work describes the complete study of shape, kinematics, and dynamics in a unified framework.

The present manuscript is organized as follows: in Section III, the computation of the manifold is described. In Sections IV and V, how to abstract shape and dynamics is shown. In Section VI, the change of coordinates to abstract the kinematics is outlined. In Section VII, the abstraction layer is discussed. In Section VIII, the experimental results are summarized, and in Section IX, the conclusions of the work are given.

III. The Arc Reachable Manifold (ARM) and Configurations in Collision

Attention was focused on differentially-driven 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 location and orientation $\mathbf{q} = (x, y, \theta)$. Let $\mathcal{U}$ be the control space and $\mathbf{u} = (v, \omega)$ a control vector (where v and ω are the linear and angular velocity, respectively). It was assumed that during the execution of a constant control, the motion was constrained to a circular elemental path (see [14] for a characterization of this assumption). Subsequently it is shown 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's system of reference. An admissible circular path from the origin $(0, 0)$ to a given point (x, y) has an instantaneous turning center on the Y-axis. The radius of that circle is:

$$r = \frac{x^2 + y^2}{2y} \tag{1}$$

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

$$\theta = f(x, y) = \begin{cases} atan2(x, \frac{x^2 - y^2}{2y}) & \text{if } \; y \geq 0 \\ -atan2(x, -\frac{x^2 - y^2}{2y}) & \text{otherwise} \end{cases} \tag{2}$$

Function f is differentiable in $\mathbf{R}^2 \backslash (0, 0)$. Thus $(x, y, f(x, y))$ defines a two-dimensional manifold in $\mathbf{R}^2 \times \mathcal{S}^1$. It was called *Arc Reachable Manifold*, $\mathrm{ARM}(\mathbf{q_0}) \equiv \mathrm{ARM}$, 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).

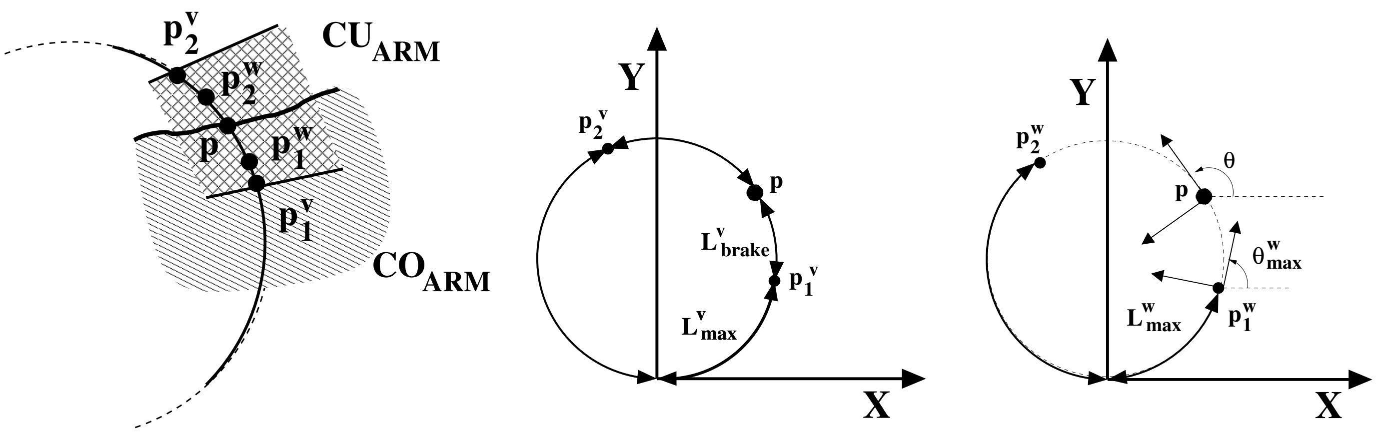
Let $g(\lambda) = (g_x(\lambda), g_y(\lambda))$, be a piecewise function that describes the robot boundary, where λ is a parameter defined in a finite interval. It was then assumed that the obstacle information was given in the form of a cloud of points (typical metric information from range sensors). For each obstacle point $\mathbf{p_f} = (x_f, y_f)$, there was a region of configurations in collision in the configuration space and part of it lied in ARM (this region is called $\mathrm{CO}^i_{\mathrm{ARM}}$). To compute it, Equation (2) was developed and some geometric properties of the problem were utilized (see [31] for details), leading to:

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

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 a piecewise function that describes the collision region boundary for a given obstacle point $\mathbf{p_i}$. The obstacle region is $\mathrm{CO}_{\mathrm{ARM}} = \bigcup_i \mathrm{CO}^i_{\mathrm{ARM}}$ for all obstacle points $\mathbf{p_i}$. The important point is that, for an arbitrary robot shape, the exact obstacle region $\mathrm{CO}_{\mathrm{ARM}}$ can be computed in the ARM (manifold of the configuration space reachable by circular paths). Figure 2 shows an illustrative example of a



(a) (b) (c)

Fig. 3. These figures show the computation of the region of unsafe configurations, $\mathrm{CU_{ARM}}$, given a point $\mathbf{p} \in \mathbf{R}^2$. (a) Of the four limit points, $\{\mathbf{p_1^v}, \mathbf{p_2^v}, \mathbf{p_1^\omega}, \mathbf{p_2^\omega}\}$, the two points farther in terms of distance over the circle in both directions leading to $\mathbf{p}$ are the border points of $\mathrm{CU_{ARM}}$. (b) Translational $\{\mathbf{p_1^v}, \mathbf{p_2^v}\}$ and (b) rotational $\{\mathbf{p_1^\omega}, \mathbf{p_2^\omega}\}$ velocity cases.

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

$$\left\{ \begin{array}{l} x_r = 2\sin^7(\lambda) \\ y_r = -4.5\cos(\lambda)(1+1.2\cos(\lambda)) + \cos^{\frac{1}{4}}(\lambda) + 2.5 \end{array} \right. \tag{4}$$

with $\lambda \in [0, \pi]$. Inserting this expression in Equation (3), the $\mathrm{CO}^i_{\mathrm{ARM}}$ for one obstacle point $\mathbf{p_i}$ and respectively the $\mathrm{CO_{ARM}}$ for all obstacles are obtained.

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

IV. Non-Admissible Configurations

Now the computation of the non-admissible configuration region $\mathrm{CNA_{ARM}}$ in the ARM is described. This region is the union of two regions:

$$\mathrm{CNA_{ARM}} = \mathrm{CO_{ARM}} \cup \mathrm{CU_{ARM}} \tag{5}$$

Region $\mathrm{CO_{ARM}}$ is the region of collision configurations (previous section); $\mathrm{CU_{ARM}}$ is the region of unsafe configurations. $\mathrm{CU_{ARM}}$ contains the configurations reached with a control after a time interval, and that cannot be cancelled by applying maximum deceleration before colliding with $\mathrm{CO_{ARM}}$. In fact, the $\mathrm{CU_{ARM}}$ region covers the $\mathrm{CO_{ARM}}$ boundary. To compute $\mathrm{CU_{ARM}}$, it was assumed that the vehicle remained on the elemental path during brakeage to reduce the complexity of all possible trajectories[1]. A point $\mathbf{p}$ from the $\mathrm{CO_{ARM}}$ boundary results in four possible points of the $\mathrm{CU_{ARM}}$ boundary: two limit points $\mathbf{p_1^v}$ and $\mathbf{p_2^v}$ for decelerating the translational velocity in both directions of the circle, and two points $\mathbf{p_1^\omega}$ and $\mathbf{p_2^\omega}$ for decelerating the rotational velocity (Figure 3a). The computation of these points is described next.

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

$$L = \left\{ \begin{array}{ll} |x|, & \text{if } y = 0 \\ |r \cdot \theta|, & \text{otherwise} \end{array} \right. \tag{6}$$

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

On one hand, regarding robot translation, the objective is to compute the two points $\mathbf{p_1^v}$ and $\mathbf{p_2^v}$, from the $\mathrm{CU_{ARM}}$ border, for a given point $\mathbf{p}$ of the $\mathrm{CO_{ARM}}$ 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:

$$\mathbf{p_1^v} = \left\{ \begin{array}{ll} (\text{sign}(x) \cdot L^v_{\max}, 0), & \text{if } y = 0 \\ (r \sin \frac{\text{sign}(x) \cdot L^v_{\max}}{r}, & \\ \quad r(1 - \cos \frac{\text{sign}(x) \cdot L^v_{\max}}{r})), & \text{otherwise} \end{array} \right. \tag{7}$$

where $L^v_{\max}$ is the maximum arc length traveled by the vehicle (during T and at v constant) that allows a deceleration of the vehicle before colliding with $\mathbf{p}$ (traveling a L^v_{brake} arc during braking), see Figure 3b. This arc is computed by:

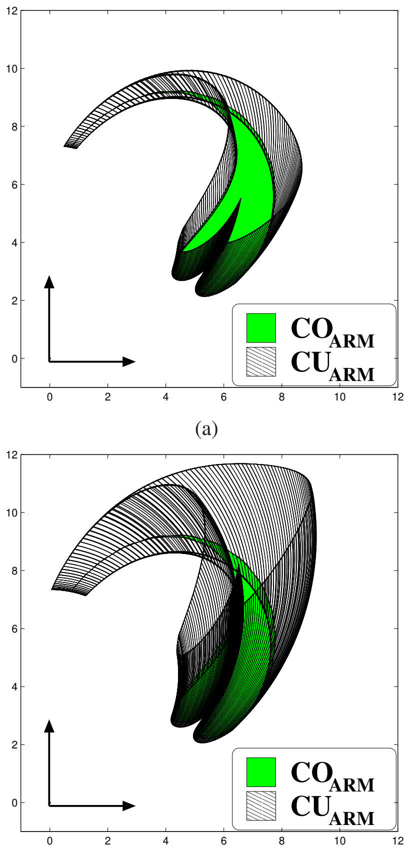
$$L^v_{\max} = L - L^v_{\text{brake}} \tag{8}$$

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

$$L^v_{\max} = a_v T^2 (\sqrt{1 + \frac{2L}{a_v T^2}} - 1) \tag{9}$$

Notice that if the distance traveled with a command v_1 in a period T is $L_1 < L^v_{\max}$, then the velocity can be cancelled before reaching $\mathbf{p}$. Location $\mathbf{p}$ can also be reached by the circle in the opposite direction (Figure 3b). Then, the other limit point $\mathbf{p_2^v}$ is computed as before but substituting L with $2\pi|r| -$

[1]With 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 considered vehicle). The implications of this assumption and the relation with prior work will be discussed in Section IX.



(a)

(b)

Fig. 4. These figures show region $\mathrm{CNA}_{\mathrm{ARM}}$ for an obstacle point in $(7, 6)$ and the "heart"-shape robot for: an acceleration a_1, in Figure (a), and for an acceleration $\frac{a_1}{4}$ in Figure (b). Region $\mathrm{CU}_{\mathrm{ARM}}$ contains the $\mathrm{CO}_{\mathrm{ARM}}$ region, acting as a security zone. The size is larger in the second case, since there is less acceleration (the vehicle needs more space to break). The region of non-admissible configurations, $\mathrm{CNA}_{\mathrm{ARM}}$, is the union of both regions.

L in Equation (8). This calculation results in the two border points $\mathbf{p_1^v}$ and $\mathbf{p_2^v}$ of $\mathrm{CU}_{\mathrm{ARM}}$. On the other hand, regarding robot rotation, the objective is to compute the two points $\mathbf{p_1^\omega}$ and $\mathbf{p_2^\omega}$, from the $\mathrm{CU}_{\mathrm{ARM}}$ border, for a given point $\mathbf{p}$ of the $\mathrm{CO}_{\mathrm{ARM}}$ boundary. The rotational velocity contributes to the orientation of the circle's tangent (angle θ), over the circle defined by $\mathbf{p}$. The point $\mathbf{p_1^\omega}$ is given by:

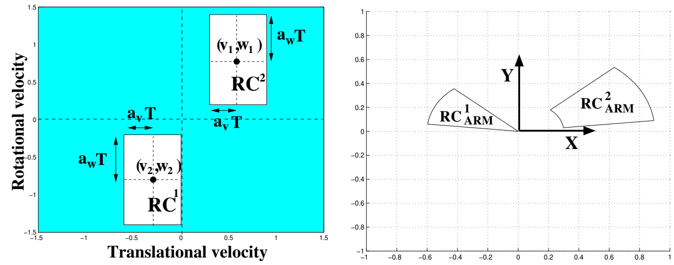
$$\mathbf{p_1^\omega} = \begin{cases} (\infty, 0), & \text{if } y = 0 \\ (r\sin(\mathrm{sign}(y)\cdot\theta^\omega_{\mathrm{max}}), & \\ \quad r(1-\cos(\mathrm{sign}(y)\cdot\theta^\omega_{\mathrm{max}}))), & \text{otherwise} \end{cases} \tag{10}$$

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

$$\theta^\omega_{\mathrm{max}} = \theta - \theta^\omega_{\mathrm{brake}} \tag{11}$$

where $\theta^\omega_{\mathrm{max}} = wT$ and $\theta^\omega_{\mathrm{brake}} = \frac{w^2}{2a_w}$. Expanding and solving yields:

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



(a) (b)

Fig. 5. (a) Velocity space of the robot and two sets RC of reachable commands given two current velocities (v_1, w_1) and (v_2, w_2). (b) The corresponding sets of reachable configurations $\mathrm{RC}^1_{\mathrm{ARM}}$ and $\mathrm{RC}^2_{\mathrm{ARM}}$ in the ARM.

The angle $\theta^\omega_{\mathrm{max}}$ is the limit angle increment. If the angle increment under command w_1 in T is $\theta_1 < \theta^\omega_{\mathrm{max}}$, the rotational velocity can be cancelled before reaching orientation θ (this is not true if $\theta_1 \geq \theta^\omega_{\mathrm{max}}$). Again, location $\mathbf{p}$ can be reached within the same circle in the opposite direction (Figure 3c). The other limit point is $\mathbf{p_2^\omega}$, computed as before, but substituting θ by $\mathrm{sign}(\theta)(2\pi - |\theta|)$ in Equation (11). The result is the two border points $\mathbf{p_1^\omega}$ and $\mathbf{p_2^\omega}$ of the $\mathrm{CU}_{\mathrm{ARM}}$.

Of the four limit points $\{\mathbf{p_1^v}, \mathbf{p_2^v}, \mathbf{p_1^\omega}, \mathbf{p_2^\omega}\}$, the two points farther in terms of distance in both directions of the circle leading to $\mathbf{p}$ are border points of the $\mathrm{CU}_{\mathrm{ARM}}$ region (Figure 3a). Finally, by applying this procedure to all border points of $\mathrm{CO}_{\mathrm{ARM}}$ the $\mathrm{CU}_{\mathrm{ARM}}$ is computed, and thus the non-admissible configurations $\mathrm{CNA}_{\mathrm{ARM}}$ (Equation (5)) are obtained. Figure 4 depicts an example. It is easy to demonstrate that the region of unsafe configurations $\mathrm{CU}_{\mathrm{ARM}}$ contains the bounds of the obstacle region $\mathrm{CO}_{\mathrm{ARM}}$. In fact, if dynamics is disconsidered, $a_v, a_\omega \rightarrow \infty$, then the $\mathrm{CU}_{\mathrm{ARM}}$ tends to be the bounds of $\mathrm{CO}_{\mathrm{ARM}}$. In other words, there are no unsafe configurations when the braking distance approaches zero (infinite accelerations are assumed).

Once $\mathrm{CNA}_{\mathrm{ARM}}$ is computed, $\mathrm{CO}_{\mathrm{ARM}}$ is computed with no additional algorithmic complexity, and the procedure derived here is valid for any vehicle shape.

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

V. Reachable Configurations

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

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

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

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

Notice that $\mathrm{RC}_{\mathrm{ARM}}$ contains all the reachable configurations in ARM in a time T given the system dynamics and current velocity.

VI. The Ego-Kinematic Coordinate Transformation

This section deals with vehicle kinematics. The original idea of such a transformation is to present the motion problem in a parameterized space, in which the paths depend on the parameters that identify the admissible paths and on the distance traveled over these paths [31]. In the case considered, a change of coordinates was applied to ARM so that the elemental paths became straight segments in the new coordinates (motion was omnidirectional). The change of coordinates transformed the domain of the manifold from $\mathbf{R}^2$ to $\mathbf{R} \times S^1$. In the new coordinates, ARM is called $\mathrm{ARM}^{\mathrm{P}}$, where to a given configuration $\mathbf{q} = (x, y) \in \mathrm{ARM}$, the corresponding configuration is $\mathbf{q}^{\mathrm{P}} = (L, \alpha) \in \mathrm{ARM}^{\mathrm{P}}$. The first coordinate of $\mathbf{q}^{\mathrm{P}}$ is the arc length L over the circle that leads to $\mathbf{q}$ (Equation (6)). The second coordinate α is a parameter[2] that represents the circle:

$$\alpha = \left\{ \begin{array}{ll} \mathrm{atan}(\frac{1}{r}), & x \geq 0 \\ \mathrm{sign}(y)\pi - \mathrm{atan}(\frac{1}{r}), & \text{otherwise} \end{array} \right. \quad (15)$$

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 in T. This is also valid for $\mathrm{ARM}^{\mathrm{P}}$, since a direction α determines an unique turning radius:

$$r = \left\{ \begin{array}{ll} \cot\alpha, & \alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \cot(\mathrm{sign}(\sin\alpha)\cdot\pi - \alpha), & \text{otherwise} \end{array} \right. \quad (16)$$

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

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

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

[2]From a physical point of view, α is the angle of a free wheel, located at a distance 1 from the origin on the X-axis, which is tangently aligned to the circle of motion with radius r.

VII. Abstraction of Shape, Kinematics, and Dynamics from the Obstacle Avoidance Methods

This section describes 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 is repeated. The idea behind the abstraction is to include two steps prior (incorporation of 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 $\mathrm{CNA}_{\mathrm{ARM}}$ and reachable region $\mathrm{RC}_{\mathrm{ARM}}$ (Sections IV and V).
2) Kinematics: Change of coordinates in ARM, where $\mathrm{CNA}^{\mathrm{P}}_{\mathrm{ARM}}$ and $\mathrm{RC}^{\mathrm{P}}_{\mathrm{ARM}}$ are the previous regions in the new coordinates system (Section VI).
3) Obstacle avoidance: Application of the obstacle avoidance method to $\mathrm{ARM}^{\mathrm{P}}$ to compute the most promising motion direction β_{sol}.
4) Motion: computation of the reachable and admissible configuration $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}}$ that is closest to β_{sol} and satisfies $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}} \in \mathrm{RC}^{\mathrm{P}}_{\mathrm{ARM}}$ and $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}} \notin \mathrm{CNA}^{\mathrm{P}}_{\mathrm{ARM}}$. Once $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}}$ is obtained, the motion command is given by Equation (17).
   To compute $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}}$, the set of configurations S_{sol} closest to β_{sol} must be obtained:
   $$S_{\mathrm{sol}} = \underset{\mathbf{q}^{\mathrm{P}} \in \mathrm{RC}^{\mathrm{P}}_{\mathrm{ARM}},\ \mathbf{q}^{\mathrm{P}} \notin \mathrm{CNA}^{\mathrm{P}}_{\mathrm{ARM}}}{\arg\min} ||\mathbf{q}^{\mathrm{P}} - \mathbf{p}^{\mathrm{P}}|| \quad (18)$$
   where $\mathbf{p}^{\mathrm{P}}$ is the projection of the $\mathbf{q}^{\mathrm{P}}$ configuration onto the unit vector in the direction of β_{sol}. When $|S_{\mathrm{sol}}| = 1$, there is only one possible configuration $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}}$ to be selected as a solution. Otherwise, S_{sol} closest to the target $\mathbf{q}^{\mathrm{P}}_{\mathbf{target}}$ is selected:
   $$\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}} = \underset{\mathbf{q}^{\mathrm{P}} \in S_{\mathrm{sol}}}{\arg\min} ||\mathbf{q}^{\mathrm{P}} - \mathbf{q}^{\mathrm{P}}_{\mathbf{target}}|| \quad (19)$$

Figure 6 shows this process using a rectangular vehicle with differential traction and a generic obstacle avoidance method (assuming a robot with no constraints, point-like, and omnidirectional). At a given time, the robot collects the sensor information about the obstacles and the target location (Figure 6a). The objective is to compute a collision-free motion command 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, $\mathrm{RC}_{\mathrm{ARM}}$ and $\mathrm{CNA}_{\mathrm{ARM}}$, in ARM (Figure 6b). In this manifold, the robot is a point and the effects of dynamics are represented in the manifold.
2) Change of coordinates in ARM (Figure 6c). In $\mathrm{ARM}^{\mathrm{P}}$, the robot is a point and the motion is omnidirectional (straight paths), which are the applicability conditions of many obstacle avoidance methods.

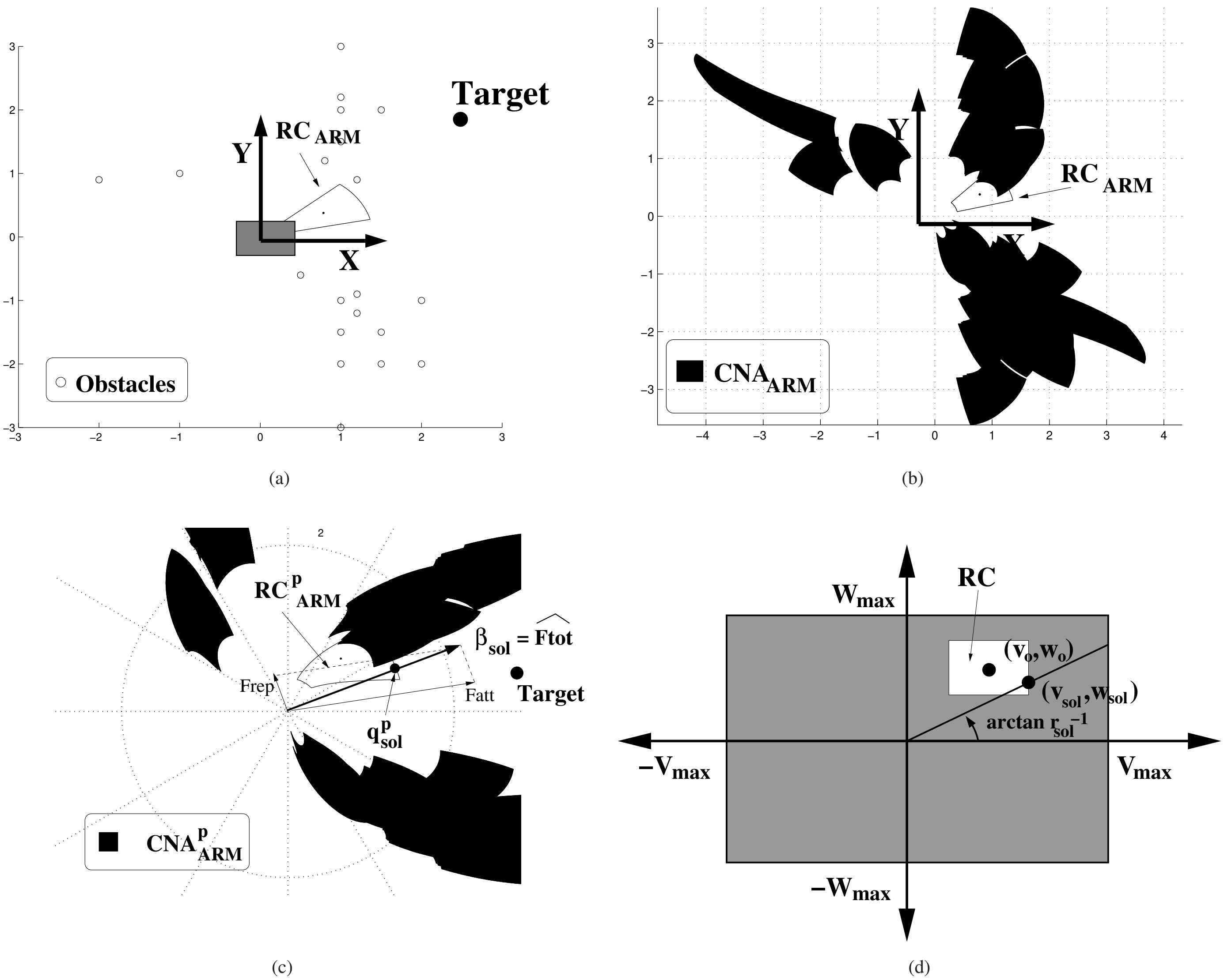


Fig. 6. These figures show the usage of the abstraction technique to take into account the shape, kinematics, and dynamics of the vehicle when applying the avoidance method. (a) Rectangular vehicle and obstacle distribution. (b) The reachable configurations, $\mathrm{RC}_{\mathrm{ARM}}$, and the non-admissible region, $\mathrm{CNA}_{\mathrm{ARM}}$, in ARM. (c) Change of coordinates, from ARM to $\mathrm{ARM}^{\mathrm{P}}$. In $\mathrm{ARM}^{\mathrm{P}}$, the robot is a point and the motion is omnidirectional. The avoidance method is then applied to obtain the most promising direction β_{sol} (that avoids the obstacles while moving from the current location towards the target). In this example a potential field method was utilized: the most promising motion direction $\beta_{\mathrm{sol}} = \widehat{\mathbf{F_{tot}}}$ is obtained by $\mathbf{F_{tot}} = \mathbf{F_{rep}} + \mathbf{F_{att}}$, where the obstacles exert a repulsive force $\mathbf{F_{rep}}$ and the target exerts an attractive force $\mathbf{F_{att}}$. The direction β_{sol} is then projected to the $\mathrm{RC}^{\mathrm{P}}_{\mathrm{ARM}}$ to obtain the configuration solution $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}}$. (d) Finally, given $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}}$, the solution command $\mathbf{v}_{\mathrm{sol}}$ is computed by Equation (17), which is shown in the velocity space.

3) Application of the obstacle avoidance method to obtain the motion direction β_{sol} that avoids the non-admissible regions $\mathrm{CNA}^{\mathrm{P}}_{\mathrm{ARM}}$ while moving the vehicle towards target $\mathbf{q}^{\mathrm{P}}_{\mathbf{target}}$ (Figure 6c).
4) This direction is used to select a configuration $\mathbf{q}^{\mathrm{P}}_{\mathrm{sol}} \in \mathrm{RC}^{\mathrm{P}}_{\mathrm{ARM}}$, which results in a motion command $(v_{\mathrm{sol}}, \omega_{\mathrm{sol}})$ using Equation (17). Figure 6d shows this command in the vehicle velocity space.

By construction, this command is goal-oriented, collision-free, in compliance with kinematics, and dynamically reachable and admissible. Notice that in this methodology, the modification introduced with respect to the direct application of the method is a change in spatial representation. However, the solutions of the method in this representation take into account the vehicle constraints. In other words, the collision avoidance method was extended to address the vehicle constraints. This is the main contribution of this work.

In the next section, it is shown how this scheme was used to apply a given obstacle avoidance method to a real vehicle.

VIII. Experimental Results

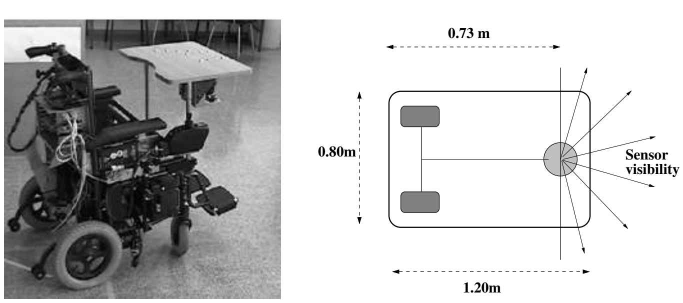
In this section the proposed methodology is validated with a collision avoidance method operating on a real vehicle, considering its rectangular shape, kinematic and dynamic constraints (differentially-driven). The vehicle, the sensor, and the collision avoidance method are described, and then the experimental results are discussed.

A. Vehicle, Sensor and Collision Avoidance Method

The vehicle is a robot built from a commercial wheelchair in our laboratory (Figure 7). The vehicle is differentially-driven and rectangular $(1.2\mathrm{m} \times 0.8\mathrm{m})$, with the drive wheels at the rear. Two Intel $800MHz$ computers were installed onboard, one for control and the other for higher-level purposes (execution of the collision avoidance technique). The sensor is a planar laser, placed at the front, operating at 5 Hz with a $180°$ field of view and $0.5°$ resolution (361 points). A weight of 60 kg was placed on the wheelchair to simulate a seated person. In all experiments, the scenario was unknown, dynamic with an unpredictable behavior, and unstructured. Under these conditions, a collision avoidance method is the correct choice to move the vehicle reactively. A potential field method (PFM in short) [18] was selected, since it is a widely known and utilized method. 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 differentially-driven robot without any 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 motions are allowed in the configuration space. Furthermore, to take into account the vehicle geometry it would imply construction of an obstacle representation in the three-dimensional configuration space, which would be difficult to execute in real-time. Finally, although the generation of reachable commands can be accomplished with a force control [18], [39], the inclusion of the braking distance (admissible commands) in the formulation of a PFM has only been done in a few works related to this abstraction layer [30]. 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).

These assumptions are very relevant for the type of vehicle used. The approximation of a rectangular geometry by a point or a circle is not realistic, since the motor-wheels are at the rear (when the vehicle turns, it sweeps a large area that must be taken into account). Due to kinematics, the vehicle moves in arcs of circles, and therefore, to assume an omnidirectional motion is a gross approximation that can put safety at risk. Dynamics play an important role and if they are ignored: (i) the planned motion may not be feasible, again putting safety at risk; (ii) oscillating behaviors will appear, making it uncomfortable for the end-user; and (iii) the vehicle skids in detriment to 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. The next section shows how to utilize the proposed technique to move the vehicle with standard PFM, taking into account all the constraints (shape, kinematics, and dynamics).



(a) (b)

Fig. 7. (a) The robot: a rectangular wheelchair vehicle with differential-drive traction and equipped with a SICK laser. (b) Distribution scheme of the wheels and sensor on the vehicle.

B. Adaptation of the Abstraction to a Differentially-driven Vehicle

The theoretical development was proposed for a unicycle or syncro-driven vehicle; however, the available vehicle is differentially-driven (Figure 7). The kinematic models are equivalent [23] and the elementary path are circles. The velocity space of the differentially-driven vehicle encompasses the velocities of the wheels (v_r, v_l) (they can be represented as (v, w) using a change of variable) [3]. There are two comments regarding the adaptation of the abstraction layer to this vehicle: (i) the set of reachable commands RC is different for (v_r, v_l). However, once the change of variable is applied, the RC of (v, w) contains (v_r, v_l) and thus is an approximation; and (ii) the computation of the non-admissible configuration region $\mathrm{CNA}_{\mathrm{ARM}}$ for both vehicles is the same.

C. Experiments

In the experiments, a sampling period of $T = 0.2$ sec was fixed (5 Hz is the frequency of the laser). This period is a maximum bound for the computation time of the algorithm[4]. The maximum accelerations of the vehicle were $(a_v, a_\omega) = (0.6\frac{\mathrm{m}}{\mathrm{sec}^2}, 0.6\frac{\mathrm{rd}}{\mathrm{sec}^2})$ and the maximum velocities were fixed to $(v_{\max}, w_{\max}) = (0.3\frac{\mathrm{m}}{\mathrm{sec}}, 0.8\frac{\mathrm{rd}}{\mathrm{sec}})$, which are not very high due to the robotic application (human transportation). In the experiments, three aspects were tested: (i) the collision avoidance task was carried out with the method using the abstraction layer: The vehicle was driven to the target whilst collisions with obstacles were avoided. (ii) The computed motion considered the shape, kinematics, and dynamics of the vehicle. (iii) When abstraction was not used, the PFM method computed solutions that could not be executed without approximations.

1) General obstacle avoidance task with abstraction: Figure 8 depicts two experiments carried out in scenarios in which obstacles were randomly placed in order to hinder the wheelchair motion (unknown, dynamic, unpredictable and unstructured scenarios). The difference between the experiments was the settings: Experiment 1 had higher obstacle density

[3]The change of variable is $v = \frac{v_r + v_l}{2}$ and $w = \frac{v_l - v_r}{d}$, where d is the radius of the wheel. This change is also valid for accelerations (a_l, a_r).

[4]The computation time is quite variable because it depends on the number of obstacle points measured (ranging from 0 to 361). It was observed that this period was an upper bound of the computation time.

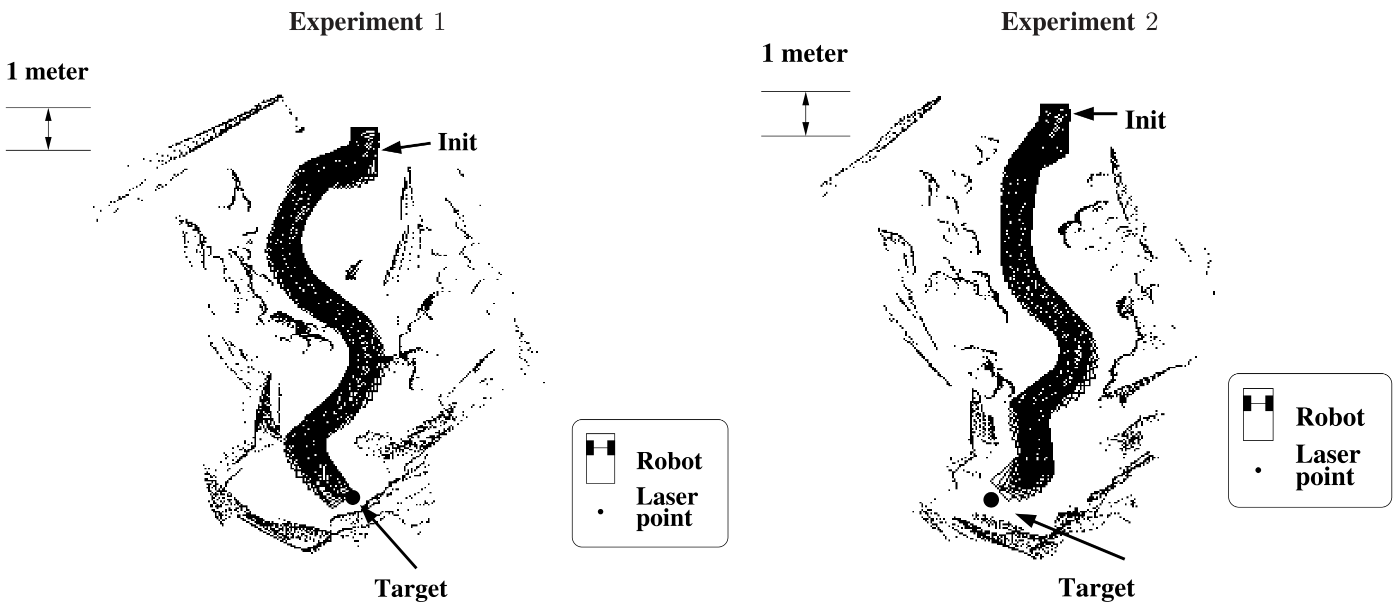


Fig. 8. Experiment 1 and Experiment 2. The path executed by the vehicle, the laser points gathered during the execution, and a snapshot of the experiment.

(more difficulty to maneuver), while Experiment 2 was more dynamic (unpredictable). In both cases, the vehicle reached the target location without collisions (see in Figure 8 the vehicle trajectory and laser points gathered). The introduction of the abstraction layer did not penalize the work of the method in avoiding obstacles. 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 if ignoring such constraints, the obstacle avoidance with this vehicle could have been heavily penalized (Subsection VIII-A), and it is doubtful that it could reach the target otherwise. The durations of the trials were 43 sec and 41 sec, the mean velocities were $0.18\frac{\mathrm{m}}{\mathrm{sec}}$, $0.24\frac{\mathrm{rad}}{\mathrm{sec}}$ and $0.12\frac{\mathrm{m}}{\mathrm{sec}}$, $0.14\frac{\mathrm{rad}}{\mathrm{sec}}$, respectively, for Experiments 1 and 2 (see in Figure 9 the velocity profiles of the reference commands and the real vehicle behavior).

2) Shape, kinematics, and dynamics in obstacle avoidance: Next is the description of how the vehicle restrictions were taken into account during the experiments. The commands computed by the method were always kinematically admissible, as they resulted from admissible circular paths. This occurred because the avoidance method was applied to the $\mathrm{ARM}^{\mathrm{P}}$ manifold, where directions corresponded to a turning radius. The motion command solution is the command that performs this turn. For example, Figure 10 depicts one step of the application of the PFM to the $\mathrm{ARM}^{\mathrm{P}}$ during one trial.

In order to address the vehicle dynamics, the method computes commands that are reachable in a short period of time, also talking into account the braking distance. The computed commands are reachable because the avoidance method computes a direction solution β_{sol} in the $\mathrm{ARM}^{\mathrm{P}}$, which is then used to select a location in $\mathrm{RC}^{\mathrm{P}}_{\mathrm{ARM}}$ (containing the configurations that can be reached in time T for the $\mathrm{ARM}^{\mathrm{P}}$, given the system dynamics). Figure 9 depicts the translational and rotational velocity profiles for the trials. Notice how the commands were reachable, since given one command, the following command was always in RC. As a consequence, the vehicle executed the planned motion strictly.

The motion commands assure that the vehicle can be stopped without collision by applying maximum deceleration (the braking distance is taken into account). This occurs because the commands are computed using admissible configurations, i.e., configurations that are not in $\mathrm{CNA}^{\mathrm{P}}_{\mathrm{ARM}}$. No emergency stops were observed in the experiments, since the PFM avoided the $\mathrm{CNA}^{\mathrm{P}}_{\mathrm{ARM}}$ regions with good safety margins. In fact, this is a desirable behavior because the configurations in the vicinities of the $\mathrm{CNA}^{\mathrm{P}}_{\mathrm{ARM}}$ are close to become unsafe. This fact is much more important in vehicles with slow dynamics, as reported in [30]. Selecting admissible commands made the method conservative, and the safety of the method was increased, because there was always a guarantee of stopping the vehicle before collision.

The last aspect to address is the vehicle shape, which was considered jointly with kinematics and dynamics. The method avoided collisions with $\mathrm{CNA}^{\mathrm{P}}_{\mathrm{ARM}}$, which was constructed taking into account the exact shape of the vehicle, as well as kinematics, and dynamics. As a result, the general effect of dealing with these three aspects simultaneously is that the vehicle executes a collision-free planned motion. This allows the robot to maneuver in scenarios with high obstacle density (Figure 8).

3) Remarks about the abstraction: Another issue to discuss is how, without utilizing the proposed methodology, the PFM avoidance method can compute solutions that approximately take into account aspects of the vehicle (which has been theoretically discussed in subsection VIII-A). Figure 10 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. In the case of complete experiments, this results in a failure when the working conditions of the robot (dynamics coupled with kinematics and shape) impose a great difficulty. For instance,

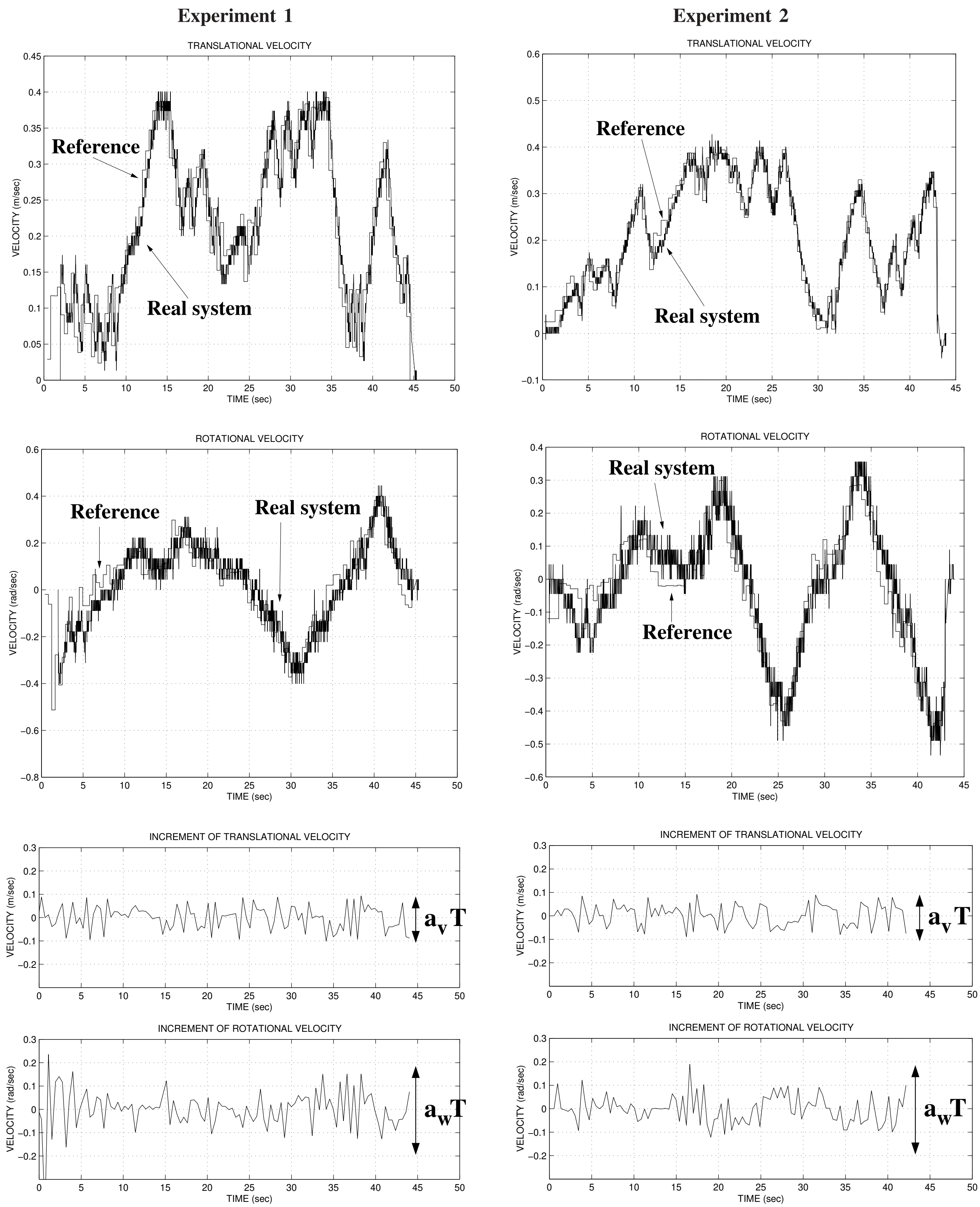


Fig. 9. This Figure shows the translational and rotational velocity profiles for each experiment: (first two rows) the computed commands and real behavior of the system; (last two rows) the translational and rotational velocity increment profiles.

Figure 11 depicts two experiments carried out with the potential field method (PFM), with and without abstraction (to convert the PFM solutions into feasible motion commands a *motion generator* was utilized [26]). In the first experiment, the vehicle had slow dynamics and faced the obstacle at maximum speed; in the second experiment, the vehicle operated at high speeds and also had high dynamic capabilities. Although both cases seem to be simple, the PFM without the abstraction layer was not able to solve them, since the robot collided with the obstacles and exhibited sudden motions. Both examples illustrate that it is important to address the constraints of the robot in obstacle avoidance, which can easily make a method fail if they are ignored.

The abstraction layer is a technique that allows the application of some methods to vehicles, but it does not ameliorate the quality of one method in itself. If a given method has some difficulties under certain conditions, the difficulties could also be present when using abstraction. For example, the PFM difficulties are to drive a vehicle between very close obstacles, and the instabilities and oscillations during motion [20]. Such difficulties were observed when using the PFM with abstraction. However, the opposite is also true and if a method performs well under certain conditions, the abstraction does not penalize the results (see [31], [30] for a discussion on this topic with another collision avoidance method). In summary, this section presented the integration of a PFM method with the proposed technique working with a real vehicle. Firstly, it was shown how the method with abstraction was able to solve the collision avoidance task. The scenarios of application were unknown, dynamic with an unpredictable behavior, and moderately dense. Secondly, it was shown how shape, kinematics, and dynamics were taken into account during the application of the method, although the method did not address these issues in its formulation. Thirdly, it was discussed how the solution of the original method without the abstraction layer cannot be executed without approximations. The proposed solution demonstrated efficiency for the wheelchair application.

IX. Discussion and Conclusions

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

A. Comparison with other Methods

Such generality is the advantage with respect to existing techniques, because: (i) some have been constructed ad-hoc to consider these constraints [14], [37], [13], [40], [16], making it difficult to reutilize these strategies with other methods; and (ii) other techniques were developed with the same objective, but take into account the constraints only after the method application [24], [5], [4], [26] (although the application scope is broad, the solution is an approximation). The benefits of this approach with respect to some widely known collision

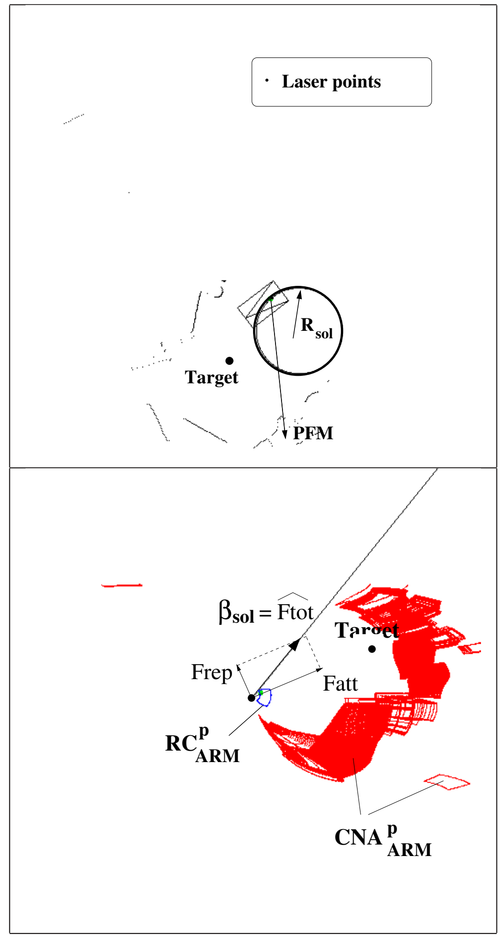


Fig. 10. This Figure shows a snapshot of the execution of the method. (a) the workspace: robot and laser sensor measurements; and (b) the $\mathrm{ARM}^{\mathrm{P}}$. The PFM is applied to the $\mathrm{ARM}^{\mathrm{P}}$ where $\beta_{\mathrm{sol}} = \widehat{\mathbf{F_{tot}}}$, which represents a turning radius R_{sol} in the workspace. Notice that if the PFM was applied to the $\mathrm{ARM}^{\mathrm{P}}$, the direction solution could not be directly executed by the vehicle (PFM in the figure).

avoidance techniques is now discussed in theoretical terms. The techniques that consider these restrictions compute collisions either over a set of elemental circular paths [17], [13], [40], [16], or over a set of commands (where each command corresponds to a circular path) [14], [37], [35]. 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 in the piecewise function describing 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 [14], [3]. However, these techniques cannot be generalized for arbitrary shapes. For instance, in the heart-shape vehicle example, 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) and this system has no closed-form solution. Although one could solve the system by utilizing a numerical method, or by projecting the robot position onto the path and checking for collisions (dynamic simulation), both strategies increase the complexity (computation time) and lead to an approximate solution. To address the complexity and

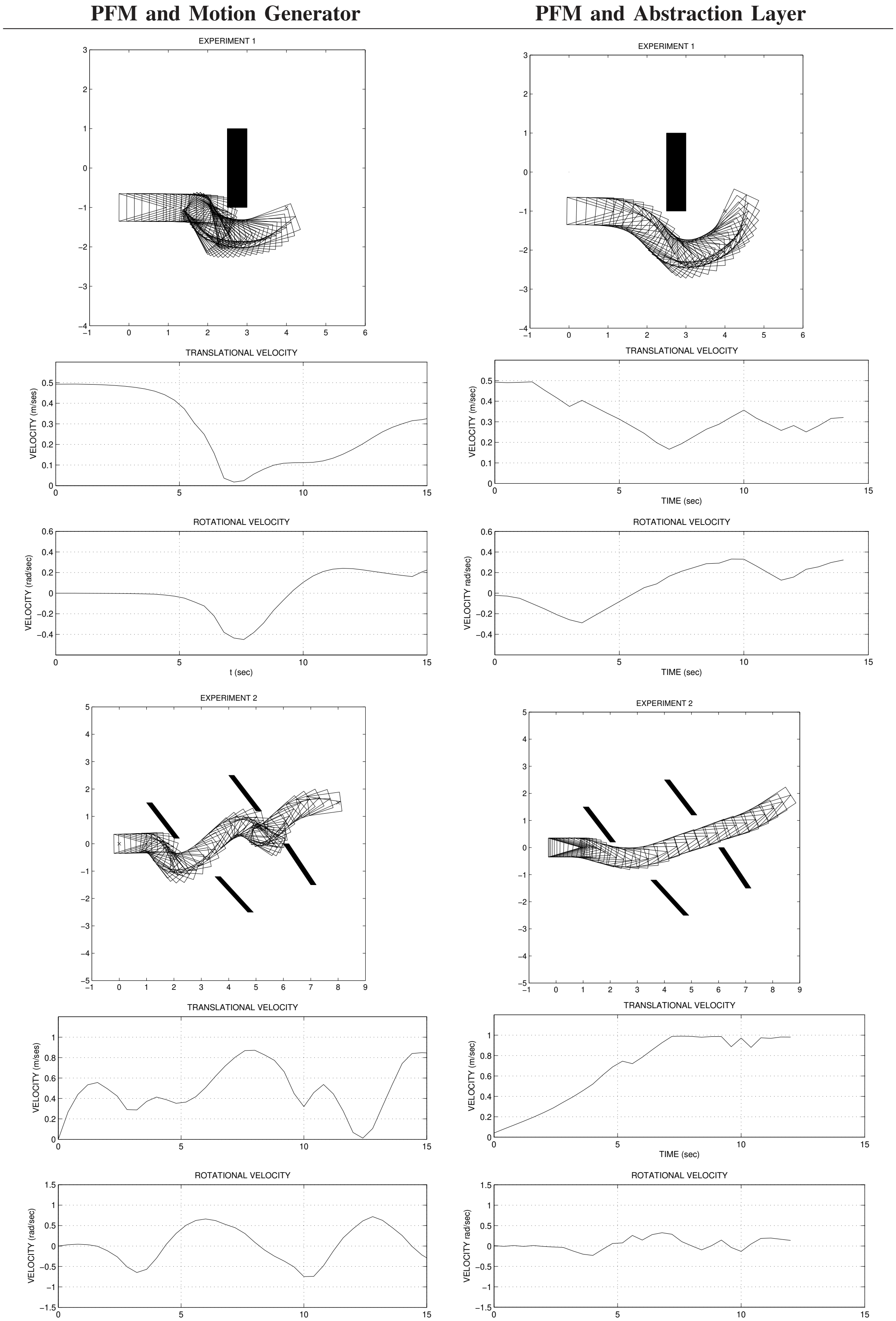


Fig. 11. This Figure depicts two experiments carried out with the potential field method (PFM) and the: motion generator (left column), abstraction layer (right column). The figures show the environment and the executed trajectory (a), and the generated translational and rotational velocities (b). In the first experiment, the vehicle had slow dynamics ($a_v = 0.3\frac{m}{sec^2}$, $a_w = 0.3\frac{rad}{sec^2}$, $v_{\max} = 0.5\frac{m}{sec}$, $w_{\max} = 0.7\frac{rad}{sec}$) and faced the obstacle at maximum speed ($v_0 = 0.5\frac{m}{sec}$). In the second experiment, the vehicle operated at high speeds and also had high dynamic capabilities ($a_v = 1\frac{m}{sec^2}$, $a_w = 1\frac{rad}{sec^2}$, $v_{\max} = 1\frac{m}{sec}$, $w_{\max} = 1.4\frac{rad}{sec}$). Although both cases seem to be simple, the working conditions of the robot (dynamics coupled with kinematics and shape) impose a great difficulty. For this reason, when considering PFM with the extraction layer, it is possible to solve such situations when otherwise it would not be possible.

efficiency issues, some researchers pre-compute the collisions with a look-up table [35] (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 collision region of the configuration space in the manifold of circular paths has a $N \times M$ complexity. This complexity factor is lower than in existing methods, but more importantly, the solution is exact and can always be computed (as long as the vehicle boundary can be described by a piecewise function). Another important consequence is that this calculation allows the maintenance of a continuous representation of the solutions space (this is why term C does not appear in the complexity factor). Existing methods can benefit from this procedure to reduce complexity, to consider any vehicle shape in a straight-forward way, and to avoid the discretization of the solutions space.

An assumption made in this work (and in all works that take into account the braking distance [14], [37], [10], [6], [3]) is that braking is carried out on an elemental path. This assumption reduces the complexity of considering all the trajectories derived from braking. Previous methods compute an approximation of the non-admissible configuration region bounds and were used only on circular or polygonal vehicle shapes [14], [37], [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 in all works that compute admissible commands, is conservative [14], [37], [35], [10], [6], [3]. This is because only the 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 safety, because the possibility of safely stoping the vehicle is always present (which is especially relevant in applications which include human or dangerous material transportation, high speed motion, or systems with slow dynamic capabilities).

Another important aspect in this paper is the focus on circular elemental paths, therefore reducing the search space of all possible trajectories as in [14], [37], [10], [6], [3]. However, extensions of this research explored the usage of combinations of different elemental paths (maneuvers) [7]. In this work 5 path families were used. A related issue is the assumption that braking is carried out on an elemental path. Such approximation allows the avoidance of the consideration of all possible trajectories derived from braking, as in [14], [37], [10], [6], [3]. Some extensions for more complex braking paths are found in [15], [36].

The collision avoidance methods are local techniques to solve the motion problem; however, the common disadvantage of these methods is that cyclical motions and trap-situations persist. 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 when applying the method, and not changing 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 [29] 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 trap-situations [41], [10], [38], [33], [3], [34], [11], [12].

A limitation of the approach is that it generates sub-optimal paths that sometimes are very counter-intuitive. This occurs because the solutions are computed over elemental circular paths, and sometimes only 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, it is suggested that the target locations be placed close to the vehicle, in order to mitigate local minima and these sub-optimal placements. An extension of our technique dealt with this issue in [7].

Our belief is that our technique can 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, this method was used to extend a standard potential field method to work on a wheelchair vehicle. The results confirm that the avoidance task was successfully carried out while jointly taking into account shape, kinematics, and dynamics of the robot.

X. Acknowledgments

The authors 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. Special thanks also to the Robotics and Manipulation group directed by O. Khatib, that hosted Javier Minguez during his stay at Stanford University, USA. Thanks are also extended to the Robotics and Artificial intelligence group directed by R. Chatila, that hosted Javier Minguez during his visit to LAAS-CNRS, France. The authors thank in particular J.P Laumond and F. Lamiraux for their fruitful commentaries and discussions during the preparation of this manuscript. The comments of J.D. Tardós of the Robotic, Perception and Real Time group at the Universidad of Zaragoza, Spain were also appreciated. This work was partially supported by the 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] 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-Madrigal. Extending Obstacle Avoidance Methods through Multiple Parameter-Space Transformations. *Autonomous Robots*, 24(1):29–48, January 2008.

[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] D. Conner, H. Choset, and A. Rizzi. Integrated planning and control for convex-bodied nonholonomic systems using local feedback control policies. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[13] 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.

[14] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics and Automation Magazine*, 4(1), 1997.

[15] T. Fraichard and H. Asama. Inevitable collision states. a step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004.

[16] 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.

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

[18] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. Journal of Robotics Research*, 5:90–98, 1986.

[19] 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.

[20] 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.

[21] 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.

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

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

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

[34] 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.

[35] 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.

[36] C. Schlegel. *Navigation and Execution for Mobile Robots in Dynamic Environments: An Integrated Approach*. PhD thesis, Universität Ulm, 2004.

[37] R. Simmons. The Curvature-Velocity Method for Local Obstacle Avoidance. In *IEEE Int. Conf. on Robotics and Automation*, pages 3375–3382, Minneapolis, USA, 1996.

[38] 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.

[39] 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, Cincinatti, OH, 1990.

[40] 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.

[41] 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.

[42] T. Wilkman, M. Branicky, and W. Newman. Reflexive Collision Avoidance: A Generalized Approach. In *IEEE Int. Conf. on Robotics and Automation*, Atlanta, USA, 1993.