

Abstracting any Vehicle Shape and the Kinematics and Dynamic Constraints from Reactive Collision Avoidance Methods

J. Minguez

ISA, Dpto. de Informática e Ing. de Sistemas
Universidad de Zaragoza, Spain
jminguez@unizar.es

L. Montano

ISA, Dpto. de Informática e Ing. de Sistemas
Universidad de Zaragoza, Spain
montano@unizar.es

Abstract—This paper addresses the extension of collision avoidance methods to address any vehicle shape as well as the kinematic and dynamic constraints. The new concept is to build an *abstraction layer* between the inputs of the reactive layer and the collision avoidance method. The vehicle characteristics are incorporated in the abstraction layer, but in such a way that when the avoidance method is used the constraints have been already taken into account. The contribution of this strategy is to widen the scope of application of collision avoidance methods that do not address the vehicle constraints. The advantage with respect to existing techniques is to address all the vehicle aspects together and, in particular, to provide a closed form solution for any vehicle shape. This strategy has been validated using an standard collision avoidance method in a real robot.

I. INTRODUCTION

Autonomous motion generation is a fundamental skill required to build a complete autonomous robot. Many systems have successfully demonstrated to achieve robust motion by integrating navigational planning and reactive collision avoidance [18], [4], [12], [17], [15]. In these systems, navigational planning provides with the long term planning and is concerned with global issues such as to assure convergence to the goal. Reactive collision avoidance is the short term part, which is used as a robot protection in execution adapting the motion to the changes gathered by the sensors. In the mentioned motion systems, the reactive layer is the natural place to address local aspects of the motion execution. Some of them are the constraints imposed by the type of vehicle. In other words, the motion commands executed have to be collision-free with the vehicle shape while taking into account its kinematics and dynamics. This is the scope of the paper.

Focussing in this local aspect of the motion, on one hand, there are collision avoidance techniques that address the type of vehicle while designing the reactive layer [6], [16], [5]. The idea behind these techniques is to build from scratch the collision avoidance technique by taking into account the shape as well as the kinematic and dynamic constraints. Although they are widely used as reactive layers, it results difficult to borrow these methodologies to extend other existing collision avoidance techniques. In addition, they cannot address some complex vehicle shapes due to the lack of closed form solutions [14]. On the other hand, it is common the design

of collision avoidance methods without taking into account the vehicle characteristics [7], [3], [11], [10]. However, when dealing with real applications, this problem remains and soon or later the type of vehicle has to be addressed. This is the reason why some methods have been developed to extend collision avoidance methods to comply with the vehicle constraints [9], [2], [19]. Their advantage is that they widen the scope of application of existing techniques, since the design is independent of the reactive method. However, they address partially the vehicle aspects and the difficulty of complex shapes persists.

This paper addresses this last problematic: how to extend a given collision avoidance method to address any vehicle shape as well as the kinematic and dynamic constraints. The new concept is to build an *abstraction layer* between the inputs of the reactive layer and the collision avoidance method. The vehicle characteristics are incorporated in the abstraction layer, but in such a way that when the avoidance method is used the constraints have been already taken into account (Figure 1). From the outside, the reactive layer is the abstraction layer plus a given collision avoidance method. This reactive layer computes the motion commands that comply with all the vehicle aspects. The contribution of this strategy is to widen the scope of application of collision avoidance methods that do not address the vehicle constraints. The advantage with respect to existing techniques is to address all the aspects together and in particular to provide a closed form solution for any vehicle shape. With respect to our previous work [14], [13], this paper presents the abstraction of all the vehicle aspects in a unified framework.

II. OVERVIEW

For vehicles with motion constraints (whose elementary paths are circles), the construction of the abstraction layer has three parts:

- First, we construct (centered on the robot at each time) the bidimensional manifold of the tridimensional configuration space defined by elementary circular paths. This manifold contains the configurations that can be reached at each step of the collision avoidance. Furthermore, in this manifold we compute the exact collision region for

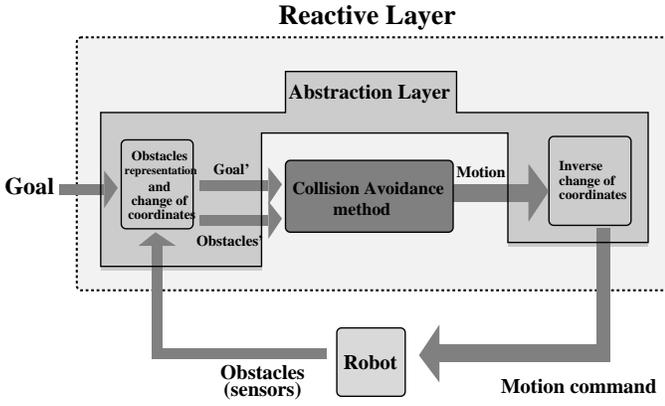


Fig. 1. The abstraction layer abstracts the shape, kinematics and dynamics of the vehicle from the avoidance method. The idea is to understand the method as a “black-box” and modify the representation of its inputs, so that they implicitly have information about these restrictions. The method is applied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

any vehicle shape (i.e. obstacle representation). In this manifold a point represents the vehicle (section III).

- Second, we compute the admissible configurations, which result from the obstacle regions computed previously¹. Furthermore, we also represent on the manifold the reachable configurations by reachable commands. The effect of the dynamics is represented in the manifold (sections IV and V).
- Finally, we propose a change of coordinates of the manifold so that the circular paths become straight segments. On the manifold in these coordinates the motion is omnidirectional (section VI).

As a result, we transform the tridimensional obstacle avoidance problem with shape, kinematics and dynamic constraints into the simple problem of moving a point in a bidimensional space without constraints. These are the assumptions made by the methods that do not address the vehicle constraints. Thus, many existing methods become applicable (Section VII). Finally, in sections VIII and IX we discuss the experimental results and draw our conclusions.

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

In this section we show how for the vehicles considered here (elementary paths are arcs of circle): (i) the vehicle configurations are constrained on a two dimensional manifold of the configuration space, and (ii) the C-obstacle regions can be exactly computed in this manifold for any vehicle shape.

We focus our attention on a syncro-drive or differential-drive robot 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} is the location and the orientation

¹We assume that the vehicle remains on the elemental path during breakage. This assumption is widely accepted in collision avoidance to reduce complexity [6], [16].

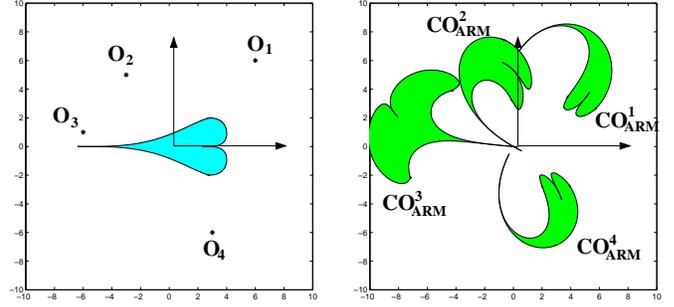


Fig. 2. This Figure shows the computation of the region of configurations in collision for a heart-shaped robot that move over circular paths. (Left) Robot and obstacles \mathbf{O}_i ; (Right) each obstacle point creates a region of collision locations CO_{ARM}^i , whose union is CO_{ARM} . The free space is the space outside these regions and all locations within these regions are in collision.

$\mathbf{q} = (x, y, \theta)$. The kinematic model of both robots is equivalent up to a variable change [8].

In the robot system of reference, an admissible circular path that leads to a point (x, y) , contains the origin $(0, 0)$ and the instantaneous turning center is 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:

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

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 call this manifold *Arc Reachable Manifold*, $ARM(\mathbf{q}_0) \equiv ARM$, since it contains all the configurations attainable by elementary circular paths from the current robot configuration \mathbf{q}_0 (i.e. all configurations attainable at each step of the obstacle avoidance).

For each obstacle point there is a region of configurations in collision in configuration space (that depend on the vehicle shape) and some of them lie in ARM . Let be $(x_i, y_i) = g(\lambda)$, where g is the piece-wise function that describes the robot boundary and λ a parameter defined in a finite interval. Then, the function $(x_s, y_s) = h(x_i, y_i, x_f, y_f)$:

$$\begin{aligned} x_s &= \frac{(x_f + x_i)[(y_f^2 - y_i^2) + (x_f^2 - x_i^2)] \cdot [(y_f - y_i)^2 + (x_f - x_i)^2]}{(y_f - y_i)^4 + 2(x_f^2 + x_i^2)(y_f - y_i)^2 + (x_f^2 - x_i^2)^2} \\ y_s &= \frac{(y_f - y_i)[(y_f^2 - y_i^2) + (x_f^2 - x_i^2)] \cdot [(y_f - y_i)^2 + (x_f - x_i)^2]}{(y_f - y_i)^4 + 2(x_f^2 + x_i^2)(y_f - y_i)^2 + (x_f^2 - x_i^2)^2} \end{aligned} \quad (3)$$

is the piece-wise function that describes the collision region boundary for a given obstacle (x_f, y_f) (see [14] for details). In other words, for an arbitrary robot shape, one can compute the obstacle region CO_{ARM} in the manifold of the configuration space reachable by circular paths in ARM . Figure 2 shows an academic but illustrative example of a heart-shaped robot, where the robot boundary $g(\lambda)$ is given by:

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

with $\lambda \in [0, \pi]$. Substituting this expression in Equation (3) we obtain the CO_{ARM}^i , corresponding to one obstacle point \mathbf{p}_i . The obstacle region is $CO_{ARM} = \bigcup_i CO_{ARM}^i$ for all obstacle points \mathbf{p}_i (Figure 2).

The complexity of this calculus is $N \times M$, where N is the number of obstacle points and M the number of pieces of function g . For instance, $M = 1$ for a circular robot or the 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 calculus computes without approximations the region in collision for any vehicle shape (as long as the robot boundary can be described by a piecewise function).

The collision avoidance problem is transformed now to a point moving in a two dimensional space.

IV. CONFIGURATIONS NON ADMISSIBLE

We describe here the computation of the non admissible configuration region CNA_{ARM} in the ARM . This region is the union of two regions: (i) region of configurations in collision CO_{ARM} (previous section); and (ii) region of configurations that once reached, the vehicle cannot stop by applying maximum deceleration before collision (configurations not safe CNS_{ARM}):

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

The not safe configuration region, CNS_{ARM} , contains the configurations reached after executing a velocity command during a time interval, whose velocity cannot be canceled by applying the maximum deceleration before traversing the bounds of CO_{ARM} (what implies collision). This region is a cover of the CO_{ARM} boundary since there are two directions of travel over the same circle.

The computation of the CNS_{ARM} region is based on computing the linear and angular braking distances independently for both controls (translation v and rotation w , which are independent for the vehicle considered). Then, the CNS_{ARM} is the union of the not safe configurations for translation CNS_{ARM}^v and rotation CNS_{ARM}^w .

$$CNS_{ARM} = CNS_{ARM}^v \cup CNS_{ARM}^w \quad (6)$$

Let (a_v, a_w) be the maximum robot accelerations and T a given time interval (in practice the sample period). Let be $\mathbf{p} = (x, y)$ a point of the piecewise function that describes the CO_{ARM} boundary (computed in the previous section). Let r and θ be the radius and orientation of the tangent to the circle in \mathbf{p} (Equations (1,2)), and L be the arc length of the circle:

$$L = \begin{cases} |x|, & y = 0 \\ |r \cdot \theta|, & y \neq 0 \end{cases} \quad (7)$$

The translation velocity contributes to the distance traveled within the circle (arc length). The maximum arc L_{max}^v that the

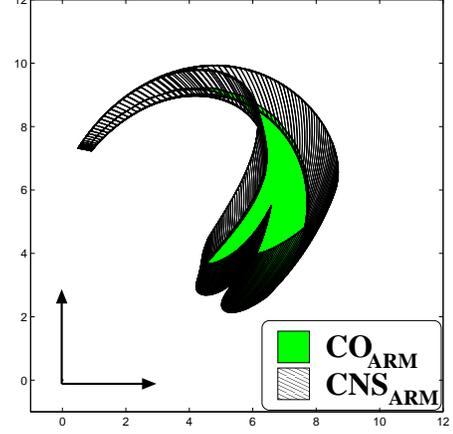


Fig. 3. This Figure depicts the region CNA_{ARM} for an obstacle point $\mathbf{r} \in \mathcal{W}$, and a heart-shaped robot. Region CO_{ARM} is the collision region in ARM . Region CNS_{ARM} is an enlargement of the CO_{ARM} boundary that depends on the braking distances. The region of non admissible configurations CNA_{ARM} is the union of both regions.

vehicle travels (during T and at v constant) that allows next to decelerate the vehicle before collision with \mathbf{p} (traveling an arc L_{brake}^v during breakage) is [13]:

$$L_{max}^v = a_v T^2 \left(\sqrt{1 + \frac{2L}{a_v T^2}} - 1 \right) \quad (8)$$

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 canceled before reaching \mathbf{p} . Location \mathbf{p} can also be reached over the circle in the opposite direction. Then, there is another limit point \mathbf{p}_2^v computed as previously but on the other side of the circle. The result is the two border points \mathbf{p}_1^v and \mathbf{p}_2^v of the CNS_{ARM}^v . The case of the rotational velocity is analogous, but it contributes to the orientation of the tangent to the circle (angle θ). The result is CNS_{ARM}^w .

The set of not admissible configurations CNA_{ARM} is the union of the configurations in collision CO_{ARM} and the set of not safe configurations CNS_{ARM} (Equation (5)), which is the union of both CNS_{ARM}^v and CNS_{ARM}^w (Equation (6)). Notice that the computation of the CNA_{ARM} does not add complexity to the calculus presented in the previous section.

In summary, we have described a calculus 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).

V. REACHABLE CONFIGURATIONS IN ARM

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 $RC = [v_o \pm a_v T, w_o \pm a_w T]$, where (v_o, w_o) is the current velocity. The set of reachable configurations RC_{ARM} in ARM is:

$$RC_{ARM} = \{\mathbf{q} \in ARM \mid \mathbf{q} = s(v, w), \quad \forall (v, w) \in RC\}$$

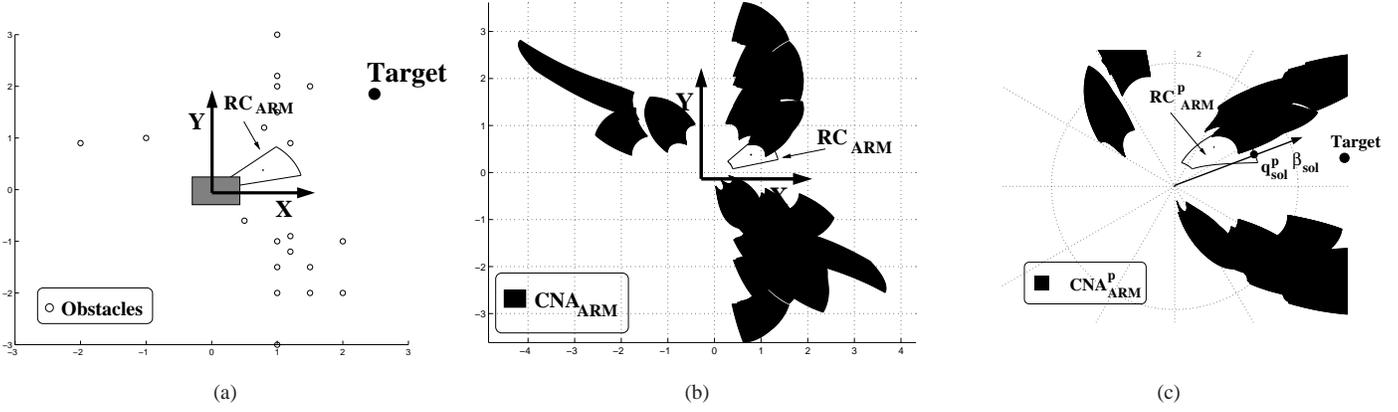


Fig. 4. These Figures show the usage of the abstraction layer to use a generic obstacle avoidance method (assumes a robot without constraints, that is punctual and omnidirectional) to work on a rectangular vehicle with differential traction. (a) Rectangular vehicle and obstacle distribution. (b) The 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 omnidirectional (applicability conditions of the obstacle avoidance method). The avoidance method is applied to obtain the most promising direction β_{sol} to avoid collisions with the CNA_{ARM}^P while approaching to the target. Direction β_{sol} is used to obtain the configuration solution \mathbf{q}_{sol}^P in the set of reachable configurations RC_{ARM}^P . Given this configuration, the motion command is then computed by Equation (12).

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

$$\mathbf{q} = s(v, w) = \begin{cases} (vT, 0), & \text{if } w = 0 \\ \left(\frac{v}{w} \sin(wT), \frac{v}{w}(1 - \cos(wT)) \right), & \text{otherwise.} \end{cases} \quad (9)$$

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 pose the motion problem in a parameterized space where the paths depend on parameters that identify the admissible paths and the distance traveled over these paths [14]. In the case here we apply a change of coordinates to ARM so that the elementary paths become straight segments with the new coordinates (motion is omnidirectional). The change of coordinates transforms the domain of the manifold \mathbf{R}^2 into $\mathbf{R} \times S^1$, where the distance to a point is the arc length L measured over the circle that reaches that point (Equation (7)), and the angle² is a parameter that univocally represents this circle:

$$\alpha = \begin{cases} \arctan\left(\frac{1}{r}\right), & x \geq 0 \\ \text{sign}(y)\pi - \arctan\left(\frac{1}{r}\right), & \text{otherwise} \end{cases} \quad (10)$$

where r is the radius of the circle. We call ARM^P to ARM in the new coordinates and $\mathbf{q}^P = (L, \alpha)$ is a configuration. In ARM^P , a direction α univocally determines a turning radius:

$$r = \begin{cases} \tan^{-1} \alpha, & \alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \\ \tan^{-1}(\text{sign}(\sin \alpha) \cdot \pi - \alpha), & \text{otherwise} \end{cases} \quad (11)$$

²From 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 aligns tangent to the circle of motion with radius r .

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

$$(v, w) = \left(\text{sign}(\cos \alpha) \frac{L}{T}, \text{sign}(\sin \alpha) |\tan \alpha| \frac{L}{T} \right) \quad (12)$$

A location in ARM^P is given by a direction and a distance on this direction. The elementary 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 build the abstraction layer between the vehicle shape, kinematics and dynamics, and the collision avoidance methods. These last 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 previous (incorporation of the shape, kinematics and dynamics) and one posterior (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).

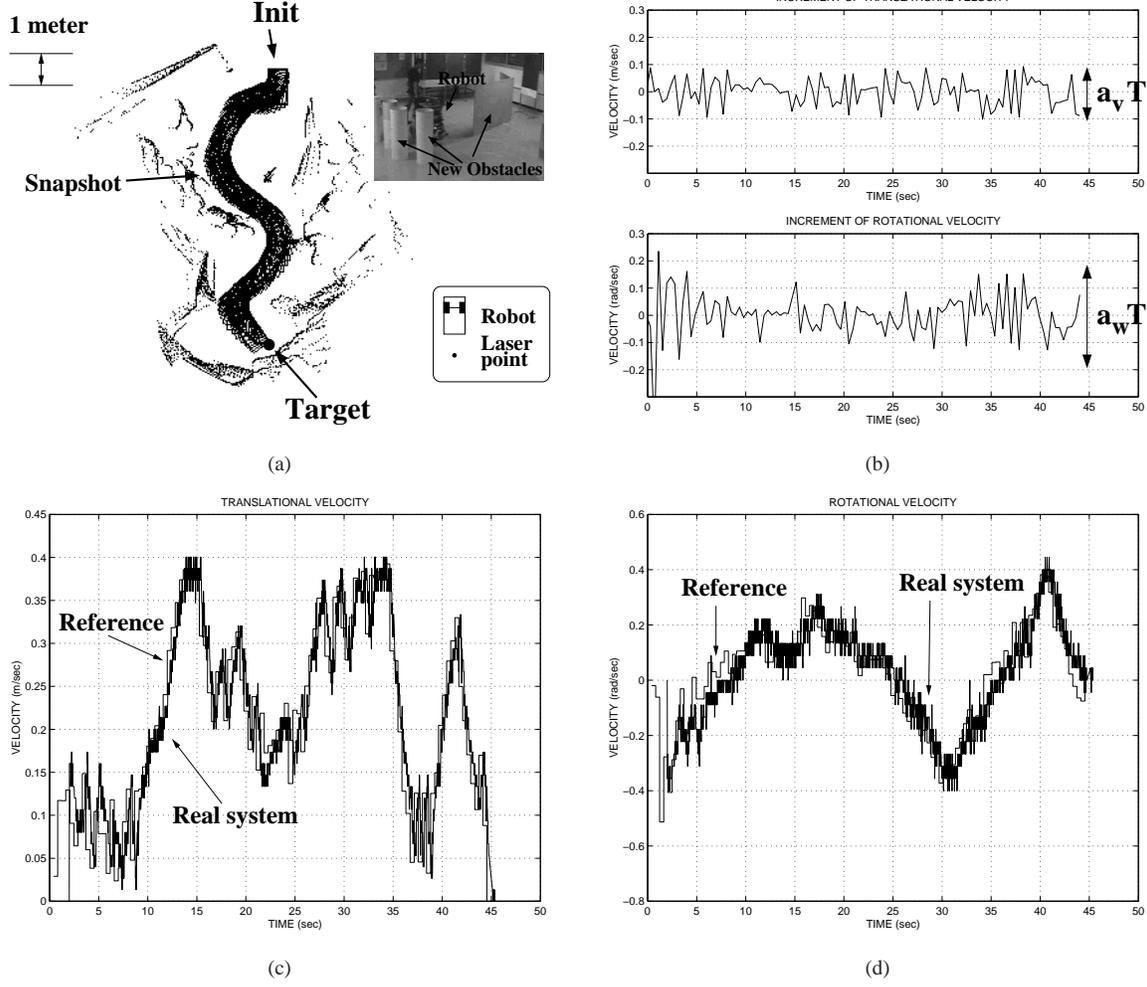


Fig. 5. (a) Path executed by the vehicle, the laser points gathered during the execution and a snapshot of the experiment. (b) Translational and rotational velocity increment profile. (c,d) Translational and rotational velocity profiles: the commands computed and real behavior of the system.

- 3) Collision avoidance: application of the collision 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, $\mathbf{q}_{sol}^P \in RC_{ARM}^P$ and $\mathbf{q}_{sol}^P \notin CNA_{ARM}^P$. Once \mathbf{q}_{sol}^P is obtained, the motion command is given by Equation (12). To compute \mathbf{q}_{sol}^P , we compute first the set of configurations S_{sol} closest to β_{sol} :

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

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 :

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

Figure 4 shows an example of this process using a rectangular vehicle with differential traction and a generic collision avoidance method (assumes a robot without constraints, that is punctual and omnidirectional).

Notice how by using the framework, the reactive layer (the abstraction layer plus the collision avoidance method) computes motion commands that comply with the vehicle constraints. In other words, using the abstraction layer, the collision avoidance method is extended to address the vehicle constraints.

VIII. EXPERIMENTAL RESULTS

In this section we use the proposed framework to extend a given obstacle avoidance method to work in a real vehicle with rectangular shape, kinematic and dynamic constraints. The vehicle is a rectangular and differential-drive wheelchair equipped with a SICK planar laser (frequency 5Hz). The maximum accelerations of the vehicle are $(a_v, a_w) = (0.6 \frac{m}{sec^2}, 0.6 \frac{rd}{sec^2})$ and we fixed the maximum velocities to $(v_{max}, w_{max}) = (0.4 \frac{m}{sec}, 0.45 \frac{rd}{sec})$, that are not very high due to the application (human transportation).

Since the objective is to validate the reactive layer we conducted all the experimentation in unknown, dynamic and unstructured scenarios. As motion method, we selected a potential field method (PFM in short) [7]. This is because when it is used as a reactive collision avoidance method assumes a punctual or circular vehicle that can move in any direction (omnidirectional without dynamics) [1].

Figure 5 shows one representative experiment that we carried out in a scenario where a human was placing randomly obstacles to hinder the wheelchair motion. The reactive layer (abstraction plus PFM) correctly performed the avoidance task avoiding the unforeseen obstacles and driving the vehicle to the goal location (see the vehicle trajectory and laser points gathered during the run). The time of the experiment was 45 sec and the mean velocities were $0.18 \frac{m}{sec}$ and $0.24 \frac{rad}{sec}$.

We discuss next in more detail how the vehicle shape, kinematics and dynamics were taken into account during the experiment. Regarding the vehicle kinematics, the avoidance method was applied in the ARM^P manifold, where directions corresponded to turning radius. To address the vehicle dynamics, the avoidance method computed a direction solution β_{sol} in the ARM^P , used to choose a location in RC_{ARM}^P (that contains the reachable configurations in time T in the ARM^P given the system dynamics). Figure 5b shows the command increments profile. Notice how all the commands are reachable, since given the current command, the next command is always within RC (set of reachable commands). As a consequence, the vehicle closely executed the reference commands, i.e. the planned motion (Figure 5c, d). Furthermore, the command assured at all times 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 from admissible configurations, i.e. they are not in CNA_{ARM}^P (that is, an "enlargement" of the C -Obstacles that depends on the dynamics and computed from the exact vehicle shape, Figure 3). We did not observed any emergency stop in the experiment since the method avoided these regions as obstacles and moved the vehicle far from them. To choose admissible commands makes the method conservative. However the method security is improved since there is always the guarantee to stop the vehicle safely when required [6].

In summary, we have seen how with the technique proposed, one can extend a given obstacle avoidance method (PFM in this experimentation) to address the vehicle constraints.

IX. CONCLUSIONS

In this work we have presented a scheme to extend collision avoidance methods to address the shape, kinematics and dynamics of the vehicle. The most important aspect of this work is its generality. Notice how with the technique proposed, one can construct the reactive layer of the system independently of the collision avoidance method (the collision avoidance method is a "black-box" within this layer). Like this, the technique proposed widens the scope of application of many existing methods.

Our believe 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 the technique proposed with a collision avoidance method (that assumes a punctual and omnidirectional vehicle) to build the reactive layer that moved the wheelchair. The results confirm that the obstacle avoidance task is successfully carried out taking into account the shape, kinematics and dynamics. This was the objective of this work.

REFERENCES

- [1] R. Arkin. Towards the unification of navigational planning and reactive control. In *Working Notes of the AIII Spring Symposium on Robot Navigation*, pages 1–6, Stanford University, 1989.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics and Automation Magazine*, 4(1), 1997.
- [7] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. Journal of Robotics Research*, 5:90–98, 1986.
- [8] J. Laumond, S. Sekhavat, and F. Lamiroux. Guidelines in nonholonomic motion planning for mobile robots. *Robot Motion Planning and Control*, 229, 1998.
- [9] 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.
- [10] J. Minguez. The Obstacle Restriction Method (ORM): Obstacle Avoidance in Difficult Scenarios. In *IEEE Int. Conf. on Intelligent Robot and Systems*, Edmonton, Canada, 2005.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] R. Simmons. The Curvature-Velocity Method for Local Obstacle Avoidance. In *IEEE Int. Conf. on Robotics and Automation*, pages 3375–3382, Minneapolis, USA, 1996.
- [17] 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.
- [18] 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.
- [19] T. Wilkman, M. Branicky, and W. Newman. Reflexive Collision Avoidance: A Generalized Approach. In *IEEE Int. Conf. on Robotics and Automation*, Atlanta, USA, 1993.