SAFE NAVIGATION OF A CAR-LIKE ROBOT WITHIN A DYNAMIC ENVIRONMENT

Stéphane PETTI

INRIA Rocquencourt
IMARA Project
78153 Le Chesnay Cedex, France
stephane.petti@inria.fr

Thierry FRAICHARD

INRIA Rhônes-Alpes
E-motion Project
38334 St Ismier Cedex, France
thierry.fraichard@inria.fr

ABSTRACT

This paper addresses the problem of navigation of a car-like robot in dynamic environments. Such environments impose a hard real time constraint. However, computing a complete motion to the goal within a limited time is impossible to achieve in most real situations. Besides, the limited duration validity of the model used for planning requires the model and therefore the plan to be updated. In this paper, we present a Partial Motion Planning (PMP) approach as the answer to this problem. The issue of safety raised by this approach is addressed using the Inevitable Collision State formalism and effectiveness of the approach is demonstrated with several simulation examples. The quality of the generated trajectories is discussed and continuous curvature metric is integrated as a mean to improve it.

1. INTRODUCTION

A large effort has been put in major industrial countries over last decade, into developing new kinds of transportation systems as a mean to address the problems of congestion, pollution and safety raised by the increasing usage of personal cars. As a long time perspective, it is envisioned to rely on fully automated vehicles, the Cybercars [1].

In this paper we address the problem of navigation through the paradigm of deliberative approaches which consists in planning a trajectory within high dimensional spaces for which it is assumed a model exists. Actual systems like car-like robots, exhibit constraints (kinematic, dynamic and actuator constraints) that restrict its motion capabilities. It is also important to deal with moving obstacles since an actual workspace will often be dynamic. Therefore, the choice of the state space framework is the only suitable to properly express the constraints of the system that can be modelled by a differential equation adding the time dimension to express the moving obstacles present within the environment. [2]. Despite the exponentially increasing complexity of motion planning in the dimension space, probabilistic planners brought in the mid 90's a new powerful tool for rapid exploration of high dimensional state-time space [3], [4]. As for the model, the exploration in the time dimension requires a model of the future to be provided. This is admittedly a strong assumption, yet realistic considering latest results on model's prediction [5], [6].

Further, there are *real time constraints* stemming from a changing environment that have to be considered. At first, the system has the obligation to make a decision within a bounded time, otherwise it might be in danger by the sole fact of being passive. This limited available time for the system to make a decision, *ie.* plan a motion (*decision time constraint*), depends on the nature and dynamicity of the environment. Then, even though a model of the future is provided, it has a limited time validity only, estimated by the prediction algorithm of these schemes. This *validity duration constraint* requires the motion planner to periodically update its model and calculate a new plan. Even though some work addresses real time motion planning [7] a few only consider highly changing environment, performing fast replanning using probabilistic techniques, though for simple systems [8] and [9]. In our work, we believe that when more complex systems or environments are considered, the real time constraints have to be explicitly considered. In such a case a complete trajectory to the goal cannot be computed in general and partial plans only can be found. In our work we do not assume an estimate of the world's model can realistically be provided over the complete navigation planning time as assumed in [10], but only over limited time period, requiring the planner to periodically calculate completely new plans. The idea of partial planning has already been scarcely mentioned in past and we intend in this paper to settle partial motion planning as a new efficient framework for planning under real time constraints.

Then, when dealing with partial plans, it becomes of the utmost importance to consider the behaviour of the system at the end of the trajectory. What if a car ends its trajectory in front of a wall at high speed? It becomes clear that strong guarantees should be given to this trajectory in order to handle the *safety* issues raised by such a *partial planning*. Our approach towards safety resides in the use of *Inevitable Col-*

lision States (ICS) formalism recently presented in [11] suitable to establish the relation of the collision states and the dynamic constraints of a system and for which we present the first practical implementation within a motion planning scheme in a dynamic environment.
Finally, we improved the convergence of the original exploration scheme by using a continuous curvature (CC) metric, used to the authors' knowledge for the first time within a probabilistic incremental planner, providing trajectories of high quality with a strong goal orientation. After briefly presenting past related work in §2, we introduce the notations in §3 that we will carry out for the presentation of the partial motion planner in §4 and the discussion on safety issues in §5. We finally detail the exploration scheme coupled with the CC metric in §6 and provide simulation results in §7 of effective navigation of a car through different highly dynamic environments and draw our conclusion in §8 over the results and the future work to be done.

2. RELATED WORK

Previous work on motion autonomy largely rely on reactive approaches that explore locally the velocity space of the system from which one admissible control is selected at a time. Motivated first, by a low computational cost and second, by the realistic difficulty to observe and model the environment, these schemes [12], [13] however have two strong limitations : a lack of lookahead, conducting the robot to be trapped in local minima during its trip, and a weak goal directedness keeping the robot from reaching the objective. Furthermore, kinematic or dynamic constraints inherent to car-like robots, addressed by a few specific schemes [14], [15] or [16] remain complex to handle in a general way. Global motion planning schemes have been modified as well in order to gain some reactivity toward changes within the environment. Beside early work based on dynamic programming [17], the approach of motion planning has mainly benefited from the probabilistic techniques. Only latest work on this issue, recognises the possibility of complete trajectory planning failure and attempts to provide safety guarantee. However the guarantees of τ-safety [10] or escape plans [8] do not relate the collision constraint with the dynamics of the system and do not provide necessary guarantees required for safe navigation within highly changing environments.

3. NOTATIONS

Let $\mathcal{A}$ denote the considered mobile robot placed in a workspace $\mathcal{W}$ (Fig. 1). In this paper, we consider the dynamic model

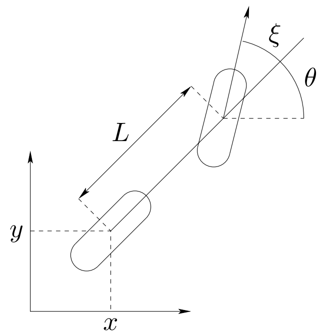


Fig. 1. The car-like vehicle $\mathcal{A}$ (bicycle model).

of a car described by the following differential equation :

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{v}\\ \dot{\xi}\end{bmatrix} = \begin{bmatrix}v\cos\theta\\ v\sin\theta\\ \frac{\tan\xi v}{L}\\ 0\\ 0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix}\alpha + \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}\gamma \qquad (1)$$

This equation is of the form $\dot{s} = f(s, u)$ where $s \in \mathcal{S}$ is the state of the system, $\dot{s}$ its time derivative and $u \in \mathcal{U}$ a control. $\mathcal{S}$ is the state space and $\mathcal{U}$ the control space of $\mathcal{A}$. A state of $\mathcal{A}$ is defined by the 5-tuple $s = (x, y, \theta, v, \xi)$ where (x, y) are the coordinates of the rear wheel, θ is the main orientation of $\mathcal{A}$, v is the linear velocity of the rear wheel, and ξ is the orientation of the front wheels. A control of $\mathcal{A}$ is defined by the couple (α, γ) where α is the rear wheel linear acceleration. and γ the steering velocity. with $\alpha \in [\alpha_{min}, \alpha_{max}]$ (acceleration bounds), $\gamma \in [\gamma_{min}, \gamma_{max}]$ (steering velocity bounds), and $|\xi| \leq \xi_{max}$ (steering angle bounds). L is the wheelbase of $\mathcal{A}$, $\mathcal{A}(s)$ is the subset of $\mathcal{W}$ occupied by $\mathcal{A}$ at a state s. Let $\phi \in \Phi$: $[t_0, t_f] \longmapsto \mathcal{U}$ denote a control input, *ie.* a time-sequence of controls. Starting from an initial state s_0, at time t_0, and under the action of a control input ϕ, the state of the system $\mathcal{A}$ at time t is denoted by $s(t) = \phi(s_0, t)$. An initial state and a control input define a trajectory for $\mathcal{A}$, *ie.* a time sequence of states.

4. PARTIAL MOTION PLANNER (PMP)

Planning in a changing environment implies to plan under real time constraints. At first, a robotic system cannot safely remain passive in a dynamic environment as it might be collided by a moving obstacle (*decision constraint*). This decision time δ_d is function of the environment's dynamicity and could be defined as the minimum time to collision with the obstacles of the environment. Secondly, in a real environment, the model of the future can be predicted over a limited time only δ_v. The planning time (or calculation time δ_c) is hence strictly limited to the minimum of these two

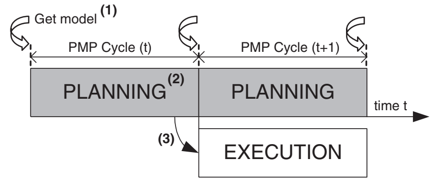


Fig. 2. Partial Motion Planning architecture

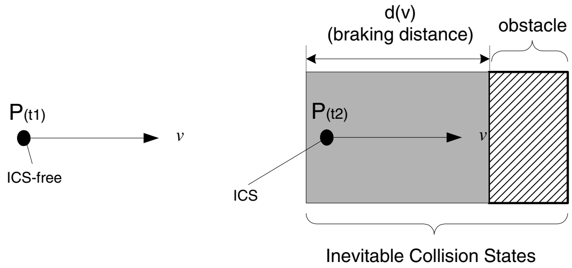


Fig. 3. Inevitable Collision State vs. Safe State

bounds. After completion of a planning cycle, it is most likely the planned trajectory of time horizon δ_h is partial. Thus, the PMP algorithm iterates over a cycle of duration δ_c as depicted in Fig. 2. We consider in this paper a constant cycle δ_c in order to be able to regularly get an update of the model, though we can note that in fact, the duration of each cycle does not have to be periodic and should be however of a length δ_c with $\delta_c = min(\delta_d, \delta_v)$ for the first cycle and $\delta_c = min(\delta_h, \delta_v)$ for the remaining cycles.

Let us focus on the planning iteration starting at time t_i:

1. An updated model of the future is acquired.
2. The state-time space of $\mathcal{A}$ is searched using an incremental exploration method that builds a tree rooted at the state $s(t_{i+1})$ with $t_{i+1} = t_i + \delta_c$.
3. At time t_{i+1}, the current iteration is over, the best partial trajectory ϕ_i in the tree is selected according to given criteria (safety, metric) and is fed to the robot that will execute it from now on. ϕ_i is defined over $[t_{i+1}, t_{i+1} + \delta_{h_i}]$ with δ_{p_i} the trajectory duration.

The algorithm operates until the last state of the planned trajectory reaches a neighbourhood of the goal state. In case the planned trajectory has a duration $\delta_h < \delta_c$, the cycle of PMP can be set to this new lower bound or the navigation (safely) stopped. In practice however, the magnitude of δ_h is much higher than δ_c.

5. SAFETY ISSUES

Like every method that computes partial motion only, PMP has to face a safety issue: since PMP has no control over the duration of the partial trajectory that is computed, what guarantee do we have that $\mathcal{A}$ will never end up in a critical situations yielding an inevitable collision? We need however to define the safety we consider. In Fig. 3 we consider a selected milestone of a point mass robot with non zero velocity moving to the right (a state of P is therefore characterised by its position (x, y) and its speed v). Depending upon its state there is a region of states (in grey) for which P, even though it is not in collision, will not have the time to brake and avoid the collision with the obstacle. As per [11], it is an Inevitable Collision State (ICS). In this paper, we refer to a safe state as an ICS-free state.

In general, computing the ICS for a given system is an intricate problem since it requires to consider the set of all the possible future trajectories. To compute in practice the ICS for a system such as $\mathcal{A}$, it is taken advantage of the approximation property established in [11]. This property shows that a conservative approximation of the ICS can be obtained by considering only a finite subset $\mathcal{I}$ of the whole set of possible future trajectories. For our application we consider the subset $\mathcal{I}$ of braking trajectories obtained by applying respectively constant controls $(\alpha_{min}, \dot{\xi}_{max})$, $(\alpha_{min}, 0)$, $(\alpha_{min}, \dot{\xi}_{min})$ until the state has stopped. Once the system is still, it is checked to be collision free (*ie.* over a trajectory obtained by applying constant (0,0) controls) until the end of the PMP cycle. With this respect, a τ safe state as introduced in [10] can be considered as an ICS-free state during τ seconds over this latter subset. In the PMP algorithm, every new state is similarly checked to be an ICS or not over $\mathcal{I}$. In case all trajectories appear to be in collision in the future, this state is an ICS and is not selected.

A safe trajectory consists of safe states. However, a practical problem appears when safety has to be checked for the continuous sequence of states defining the trajectory. In order to solve this problem and further reduce the complexity of the PMP algorithm, we presented in [18] a property that simplifies the safety checking for a trajectory. This property is important since first, it proves a trajectory is continuously safe while the states safety is verified discretely only, and second it permits a practical computation of safe trajectories by integrating a dynamic collision detection module within existing incremental exploration algorithms, like A* or Rapidly-Exploring Random Tree (RRT) [19].

6. SPACE EXPLORATION

6.1. incremental exploration

In our work, we used the efficient RRT method and replaced the geometric collision checker with our inevitable collision state checker. The control space of our system is reduced to the set of bang bang controls $\tilde{\mathcal{U}}$=$(\alpha, \dot{\xi})$ with

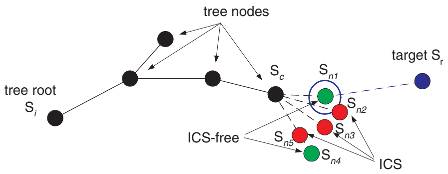
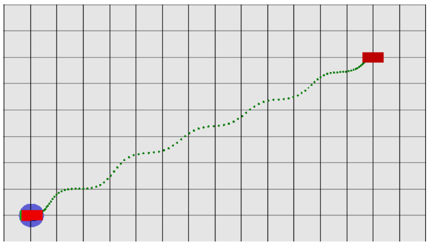
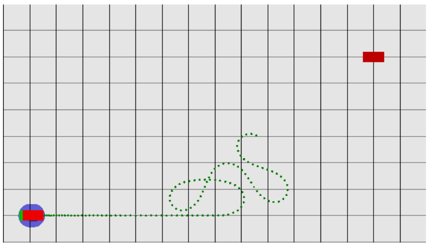


Fig. 4. Tree contruction over a PMP cycle



(a) Continuous Curvature metric



(b) L_∞ metric

Fig. 5. Navigation within a highly dynamic environment

$\alpha \in [\alpha_{min}, 0, \alpha_{max}]$ and $\dot{\xi} \in \left[\dot{\xi}_{max}, 0, \dot{\xi}_{min}\right]$ The exploration of the state-time space consists in building incrementally a tree as follows (Fig. 4): a milestone s_r is generated within $\mathcal{W}$ with a probability p to be the goal. The closest state s_c to s_r is selected. A control from $\widetilde{\mathcal{U}}$ is applied to the system during a fixed time (integration step). In case the new state s_n of the system is not an ICS, this control is valid. The operation is repeated over all control inputs and finally the new state, safe and closest to s_r, is finally selected and added to the tree. At the beginning of the first cycle it is necessary the initial state is safe over $\mathcal{I}$ during δ_vs. The roots of the tree of the subsequent cycles being the best end node of the former tree and by construction safe, it is necessary to check they remain safe over the new period δ_v of the updated model.

6.2. Metric Issues

One difficulty when performing motion planning using an incremental approach is the choice of the metric used to select and expand the nodes in order to build the tree. This parameter is recognised to have a large influence on the trajectory quality specially when dealing with non-holonomic

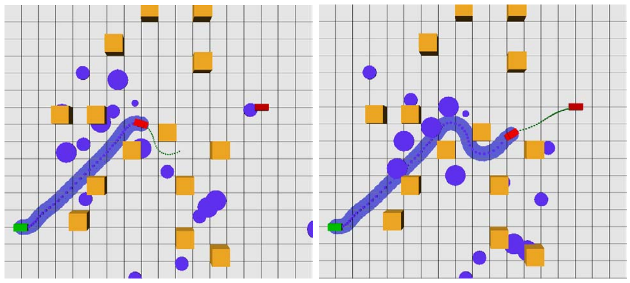


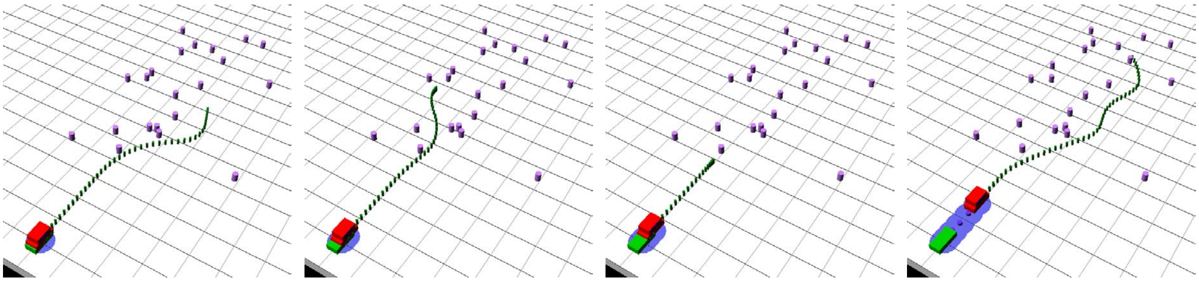
Fig. 7. Navigation within a highly dynamic environment

systems. The continuous curvature (CC) metric presented in [20] is a non-holonomic metric.Basically, this metric connects the straight path and the arc of circle of the dubbins path, where there is a discontinuity in the curvature with a clothoid. Metrics built on the L_∞ norm are oscillatory and do not provide high quality non-holonomic trajectory shapes. In Fig. 5, we illustrate the influence of the metric by comparing the CC metric we use with the metric used in [21]. The CC metric is much smoother and properly goal oriented.

7. RESULTS

The first implementation of the PMP in various dynamic environment proves the efficiency of the approach as a navigation function for a car-like robot. The software is implemented in C++ and run on a Pentium4@1.7GHz. The parameters of the PMP are a cycle δ_c of 1s, an integration step of 0.5s and for the car $v_{max} = 2.0m/s$, $\xi_{max} = \pi/3rad$, $\dot{\xi}_{max} = 0.2rad/s$, $\alpha_{max} = 0.1m/s^2$. The environment is supposed fully observable with a validity satisfying the PMP constraints (*ie.* $\delta_v > \delta_c + \delta_b$). At the initial state (left) the car is still and ICS-free, and the goal state (right) is at still as well. Fig. 6 illustrates the generated trajectory for four subsequent PMP cycles. At each cycle, given a new model of the future, a complete new plan is calculated. In this simulation, some noise is added to the nominal pedestrian's motion (small cylinders) in order to have a more realistic environment. Fig. 7 illustrates how the algorithm provides efficient navigation to the car while evolving within a highly constrained dynamic environment. The safe planned trajectory is displayed in front of the car, and the (ideally) executed trajectory built from previous PMP cycles, behind the car. Finally, Fig. 8 illustrates a case study of such an implementation within a real platform within a city. Videos can be found at `http://emotion.inrialpes.fr/fraichard/pmp-films`.

This work is currently being integrated on a real platform, a *cycab* (Fig. 10) moving in real conditions (crossing, parking lot). The observation of the environment (laser scanner/vision) will provide required information on static



(a) 1st PMP cycle plan (b) 2nd PMP cycle plan (c) 3rd PMP cycle plan (d) 4th PMP cycle plan

Fig. 6. Navigation within an environment cluttered with moving pedestrians

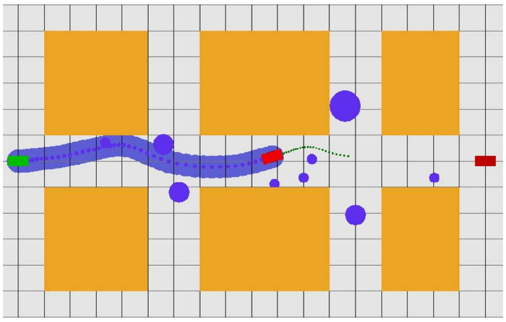


Fig. 8. Navigation on a one-way pedestrian-friendly road

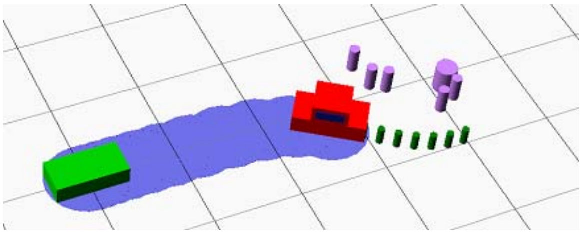


Fig. 9. trajectory avoiding pedestrians detected with a laser scanner

(parked cars) as well as moving obstacles (pedestrians and cars) (See Fig. 9), whereas the trajectories of the moving obstacles will be predicted using the work of [6].

8. CONCLUSION AND FUTURE WORKS

In this paper we tackle the problem of navigation for a car-like robot within a dynamic environment and propose a Partial Motion Planning scheme (PMP) which handles the *real-time constraint* inherent to such an environment, while accounting for the kinematic and dynamic constraints of the system. The PMP algorithm consists in iteratively exploring the state-time space during a fixed limited time and building a tree using incremental techniques. During a cycle, a complete trajectory calculation to the goal can not be guaranteed in general, which raises the issue of the *safety* of our system. We use the formalism of the *Inevitable collision States* (ICS) as the theoretical answer to this safety problem. Thus, we present an original and effective algorithm navigating safely a car-like robot within highly dynamic environment. Furthermore, this paper proposes the first results of improved shape of the planned trajectories by mean of the continuous curvature metric within the incremental search algorithm. Finally, simulation results prove the overall effectiveness of the complete PMP algorithm and show a case-study for a real implementation within a pedestrian city area. Future work includes the coupling of the PMP algorithm with a closed loop control and its integration on an experimental vehicle. Our goal is to perform experimentations within a real environment, for which a model of the future obstacles' behaviour will be determined thanks to a prediction technique.

Fig. 10. the cycab: an autonomous car-like robot platform

Acknowledgment

This work was partially supported by Aisin AW Corp. and the French Predit Mobivip project.

9. REFERENCES

[1] Parent M., "Automated public vehicles : A first step towards the automated highway," in *4th World Congress on Intelligent Transport Systems*, October 1997.

[2] T. Fraichard and C. Laugier, "Kinodynamic planning in a structured and time-varying 2D workspace.," in *Int. Conf. on Robotics and Automation*, Nice, (FR), May 1992.

[3] L. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, 1996.

[4] S. LaValle, "Numerical computation of optimal navigation functions on a simplicial complex," 1998.

[5] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, September 2003.

[6] D. Vasquez and Th. Fraichard, "Motion prediction for moving objects: a statistical approach," in *Int. Conf. on Robotics and Automation*, New Orleans, LA, April 2004.

[7] O. Brock and O. Khatib, "Real time replanning in high-dimensional configuration spaces using sets of homotopic paths," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, San Francisco (US), May 2000.

[8] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, March 2002.

[9] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *Int. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland, October 2002.

[10] E. Feron, E. Frazzoli, and M. Dahleh, "Real-time motion planning for agile autonomous vehicles," in *AIAA Conference on Guidance, Navigation and Control*, Denver (US), August 2000.

[11] Thierry Fraichard and Hajime Asama, "Inevitable collision states - a step towards safer robots?," *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.

[12] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278–288, June 1991.

[13] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, July 1998.

[14] R. Simmons, "The curvature velocity method for local obstacle avoidance," in *International Conference on Robotics and Automation*, Minneapolis (USA), april 1996, pp. 3375–3382.

[15] J. Minguez, L. Montano, and J. Santos-Victor, "Reactive navigation for non-holonomic robots using the ego kinematic space," in *Int. Conf. on Robotics and Automation*, Washington (US), May 2002.

[16] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," Tech. Rep. IAI-TR-95-13, 1 1995.

[17] A. Stentz, "The focussed D* algorithm for real-time replanning," in *Int. Joint Conf. on Artificial Intelligence*, Montreal, Quebec, 1995, pp. 1652–1659.

[18] S. Petti and T. Fraichard, "Safe navigation within dynamic environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton (CA), August 2005.

[19] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," in *Int. Conf. on Robotics and Automation*, Detroit (US), May 1999, pp. 473–479.

[20] A. Scheuer and T. Fraichard, "Planning continuous-curvature paths for car-like robots," in *Int. Conf. on Intelligent Robots and Systems*, Osaka, Japan, November 1996, vol. 3, pp. 1304– 1311.

[21] E. Lamiraux, F. Ferre and Vallee E., "Kinodynamic motion planning : Connecting exploration trees using trajectory optimzation methods," in *Int. Conf. on Robotics and Automation*, New Orleans (US), April 2004.