Trajectory planning amidst moving obstacles: path-velocity decomposition revisited

Thierry Fraichard

Journal of the Brazilian Computer Science Society
Special issue on Robotics, Vol. 4, No. 3, April 1998

Trajectory Planning Amidst Moving Obstacles: Path-Velocity Decomposition Revisited

Th. Fraichard

Inria[a] Rhône-Alpes
Zirst. 655 av. de l'Europe, 38330 Montbonnot Saint Martin, France
thierry.fraichard@inria.fr

October 12, 1998

Abstract — This paper addresses trajectory planning for a robot subject to dynamic constraints and moving in a dynamic workspace. A car-like robot $\mathcal{A}$ with bounded velocity and acceleration, moving in a dynamic two-dimensional workspace is considered. The solution proposed is an extension of the *path-velocity decomposition* which is a practical way to address trajectory planning in dynamic workspaces. However it presents a serious drawback: it cannot find a solution if a moving obstacle stops right on the computed path. Previous answers to this problem were to consider sets of candidate paths. The answer proposed in this paper makes use of the novel concept of *adjacent paths* (like adjacent lanes of the roadway). A set of adjacent paths, one of which leads $\mathcal{A}$ to its goal, is computed. Then, assuming that $\mathcal{A}$ is able to shift from one path to an adjacent one freely, the motion of $\mathcal{A}$ along and between these paths is determined so as to avoid the moving obstacles. The fact that it is possible to switch several times between two adjacent paths makes this approach more flexible and more powerful than one considering candidate paths.

Keywords — Mobile robots, motion planning, non-holonomic constraints, dynamic constraints, moving obstacles.

Acknowledgements — This work was partially supported by the European Eureka EU-153 project "Prometheus Pro-Art".

[a]Institut National de Recherche en Informatique et en Automatique.

Trajectory Planning Amidst Moving Obstacles: Path-Velocity Decomposition Revisited

Th. Fraichard
Inria Rhône-Alpes
Zirst. 655 av. de l'Europe, 38330 Montbonnot Saint Martin, France
thierry.fraichard@inria.fr

Abstract

This paper addresses trajectory planning for a robot subject to dynamic constraints and moving in a dynamic workspace. A car-like robot $\mathcal{A}$ with bounded velocity and acceleration, moving in a dynamic two-dimensional workspace is considered. The solution proposed is an extension of the *path-velocity decomposition* which is a practical way to address trajectory planning in dynamic workspaces. However it presents a serious drawback: it cannot find a solution if a moving obstacle stops right on the computed path. Previous answers to this problem were to consider sets of candidate paths. The answer proposed in this paper makes use of the novel concept of *adjacent paths* (like adjacent lanes of the roadway). A set of adjacent paths, one of which leads $\mathcal{A}$ to its goal, is computed. Then, assuming that $\mathcal{A}$ is able to shift from one path to an adjacent one freely, the motion of $\mathcal{A}$ along and between these paths is determined so as to avoid the moving obstacles. The fact that it is possible to switch several times between two adjacent paths makes this approach more flexible and more powerful than one considering candidate paths.

Keywords: mobile robots, motion planning, non-holonomic constraints, dynamic constraints, moving obstacles.

1 Introduction

1.1 Trajectory Planning in Dynamic Workspaces

Ever since Nilsson's work in 1969 [17], motion planning has been extensively studied (the reader is referred to [15] for a recent survey of this topic). Previous works can be classified according to the type of motions which are planned. Thus it is possible to differentiate between *path planning* which is characterized by the search of a continuous sequence of configurations,[1] and *trajectory planning* which is concerned with the time history of such a sequence.

Path planning is restricted to the geometric aspects of motion planning. The only constraints that can be taken into account are time-independent constraints such as stationary obstacles and kinematic constraints, *i.e.* constraints involving the configuration parameters of the robot and their derivatives. Depending on whether it is integrable, a kinematic constraint either reduces the set of allowed configurations (like an obstacle) or restricts the geometric shape of feasible paths. On the other hand, trajectory planning with its time dimension permits to take into account time-dependent constraints such as moving obstacles and the dynamic constraints of the robot, *i.e.* the constraints imposed by the dynamics of the robot and the capabilities of its actuators.

Path planning has been extensively studied in the past twenty years. Whereas less attention has been paid to trajectory planning. And yet, when planning the motion of an actual robot, it is important to take into account the various constraints that restrict its motion capabilities and especially dynamic constraints. It is also important to deal with moving obstacles since an actual workspace will often be dynamic, *i.e.* with moving obstacles. These two points, *i.e.* moving obstacles and dynamic constraints, have been addressed in the past, but seldom simultaneously (*cf.* §2), and it is the purpose of this paper to do so. Accordingly, this paper addresses trajectory planning in dynamic workspaces, *i.e.* motion planning for a robot subject to dynamic constraints and moving in a dynamic workspace.

1.2 Contribution of the Paper

The problem of planning the two-dimensional motion of a car-like robot $\mathcal{A}$ with bounded velocity and acceleration, and moving in a dynamic workspace $\mathcal{W}$, is considered in this paper.

The solution proposed is derived from the 'path-velocity decomposition' introduced in [14], which addresses motion planning in two complementary stages: (a) planning a geometric path that avoids the stationary obstacles and (b) planning the velocity along this path so as to avoid the

[1] The *configuration* of a robot is a set of independent parametres that uniquely defines the position and orientation of every point of the robot.

moving obstacles. This approach is of a practical interest because it decomposes the original problem into two more simple sub-problems (*cf.* §2 for the complexity issues). As a matter of fact, this type of approach has been largely used before (*cf.* §2). However path-velocity decomposition presents a serious drawback: it cannot find a solution if a moving obstacle stops right on the computed path. A possible answer to this problem is to consider a set of candidate paths [19]. The answer proposed in this paper is the novel concept of *adjacent paths*. Adjacent paths are formally defined in §4.1.2. Meanwhile, an informal illustration of what adjacent paths are can be found in the roadway: roads are usually divided into several adjacent lanes and every driver knows what passing from one lane to an adjacent one means. The fact that it is possible to switch several times between two adjacent paths makes this approach more flexible and more powerful than one considering candidate paths.

Thanks to this concept, a novel motion planning scheme that also operates in two complementary stages was designed. The first stage, *paths-planning*, takes into account all the time-independent features of the problem at hand (stationary obstacles and $\mathcal{A}$'s kinematic constraints) whereas the second stage, *trajectory-planning*, deals with the time-dependent ones (moving obstacles and $\mathcal{A}$'s dynamic constraints). In the paths-planning stage, a set of adjacent paths, one of which leading $\mathcal{A}$ to its goal, are computed. These paths are collision-free with the stationary obstacles and respect $\mathcal{A}$'s kinematic constraints. In the trajectory-planning stage, given that $\mathcal{A}$ is able to shift from one path to an adjacent one freely, the motion of $\mathcal{A}$ along and between these paths is determined so as to avoid any collision with the moving obstacles while respecting $\mathcal{A}$'s dynamic constraints.

1.3 Outline of the Paper

§2 reviews the complexity issues and the works related to trajectory planning in dynamic workspaces. Then §3 formally states the problem at hand. Afterwards §4 describes an algorithm that solves this problem by using the path-velocity decomposition.

2 Complexity Issues and Related Works

There are results suggesting that trajectory planning in dynamic workspaces is generally intractable [30]. Even the two-dimensional case is computationally involved (*cf.* the NP-hard result established in [3] and the P-Space algorithm presented in [5]). On the other hand, the one-dimensional case seems less intricate (*cf.* the two polynomial algorithms presented in [18] and [21]) hence the interest of the path-velocity decomposition.

A general approach that deals with moving obstacles is the configuration-time space approach which consists in adding the time dimension to the robot's configuration space [7]. The robot maps in this configuration-time space to a point moving among stationary obstacles. Accordingly the different approaches developed in order to solve the path planning problem in the configuration space can be adapted in order to deal with the specificity of the time dimension and used (*cf.* [15]). Among the existing works are those based upon extensions of the visibility graph [7, 14, 21] and those based upon cell decomposition [12, 24].

There are several results for time-optimal trajectory planning for Cartesian robots subject to bounds on their velocity and acceleration [5, 18]. Besides optimal control theory provides some exact results in the case of robots with full dynamics and moving along a given path [2, 28]. Using these results, some authors have described methods that computes a local time-optimal trajectory [27, 25]. The key idea of these works is to formulate the problem as a two-stage optimization process: optimal motion time along a given path is used as a cost function for a local path optimization (hence local time-optimality). However the difficulty of the general problem and the need for practical algorithms led some authors to develop approximate methods. Their basic principle is to define a grid which is searched in order to find a near-time-optimal solution. Such grids are defined either in the workspace [26], the configuration space [22], or the state space of the robot [4].

Few research works take into account moving obstacles and dynamic constraints simultaneously, and they usually do so with far too simplifying assumptions, *e.g.* [12] and [18]. Ref. [10] introduced the concept of state-time space as a tool to deal with trajectory planning with moving obstacles and dynamic constraints. It was used to plan the one-dimensional motion of a mobile robot subject to velocity and acceleration bounds and moving amidst moving obstacles. Later, it was applied to the case of a car-like robot subject to dynamic constraints and moving along a given path [9], and on a planar surface [11]. More recently, [8] has presented a two-stage algorithm that computes a local time-optimal trajectory for a manipulator arm with full dynamics and moving in a dynamic workspace: the solution is computed by first generating a collision-free path using the concept of velocity obstacle, and then by optimizing it thanks to dynamic optimization.

3 Statement of the Problem

3.1 The Robot $\mathcal{A}$

Let $\mathcal{A}$ be a car-like robot with two rear wheels and two directional front wheels. It is modelled as a polygon moving on the plane $\mathbb{R}^2$. A configuration of $\mathcal{A}$ is completely defined by the 3-tuple $(x, y, \theta) \in \mathbb{R}^2 \times [0, 2\pi[$ where (x, y) are the coordinates of the rear axle midpoint R and θ is the orientation of $\mathcal{A}$ (Fig.1).

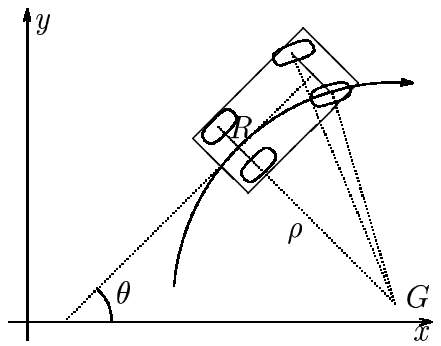

Figure 1: a car-like robot.

Kinematic Constraints. A body moving on the plane has only one centre of rotation. Let G be the center of rotation of $\mathcal{A}$. Assuming pure rolling condition, a wheel can only move in a direction which is normal to its axle. Therefore, when $\mathcal{A}$ is moving, the axles of its wheels intersect at G. The orientation of the rear wheels being fixed, G is located on the rear wheels axle (possibly at an infinite distance) and R moves in a direction which is normal to this axle. In other words, the following constraint holds:

$$\tan\theta \quad = \quad \dot{y}/\dot{x} \tag{1}$$

Besides, due to the fact that the front wheels orientation is mechanically limited, the distance ρ between R and G, *i.e.* the curvature radius at point R, is lower bounded:

$$\rho \quad \geq \quad \rho_{\min} \tag{2}$$

Relations (1) and (2) are non-holonomic [1], they restrict the geometric shape of feasible paths for $\mathcal{A}$.

Dynamic Constraints. Let v be the velocity of $\mathcal{A}$ measured along its main axis and let $a_{\tan}$ and $a_{\rm rad}$ be respectively the tangential and radial components of the acceleration applied to $\mathcal{A}$. The following dynamic constraints should be satisfied:

$$0 \leq \quad v \quad \leq v_{\max} \tag{3}$$

$$-a_{\max} \leq \quad a_{\tan} \quad \leq a_{\max} \tag{4}$$

$$-g_{\max} \leq \quad a_{\rm rad} \quad \leq g_{\max} \tag{5}$$

The meaning of these constraints is quite straightforward. Inequality (3) is a speed limit and (4) is an upper bound on the rate of change of speed. The purpose of (5) is to ensure that the radial acceleration does not exceed the counteracting centrifugal acceleration which is supplied by friction between the wheels and the ground. Note that (3) implies that $\mathcal{A}$ is not allowed to back-up. It must always move forward which is the case in normal road-driving situations.

Although simple, these dynamics constraints are rich enough to demonstrate the interest of our approach. More accurate dynamics constraints could easily be dealt with (*cf.* [9, 11]).

3.2 The Workspace $\mathcal{W}$

The workspace $\mathcal{W}$ is a subset of $\mathbb{R}^2$, it is cluttered up with stationary obstacles $\mathcal{B}_i^S, i \in \{1, \ldots, s\}$, and with moving obstacles $\mathcal{B}_j^M, j \in \{1, \ldots, m\}$. Both types of obstacles are modelled as convex polygonal regions of $\mathcal{W}$.

3.3 The Problem

In this framework, a trajectory between an initial configuration q_s and a final configuration q_g is defined by a mapping Γ taking a time $t \in [0, t_f]$ to a configuration $\Gamma(t) = (x(t), y(t), \theta(t))$ and such that $\Gamma(0)=q_s$ and $\Gamma(t_f)=q_g$. The duration of the trajectory Γ is t_f. Γ must be *collision-free*, *i.e.* it must respect:

$$\forall t \in [0, t_f],\ \forall i \in \{1, \ldots, s\},\ \forall j \in \{1, \ldots, m\},$$

$$\mathcal{A}(t) \cap \mathcal{B}_i^S = \emptyset \text{ and } \mathcal{A}(t) \cap \mathcal{B}_j^M(t) = \emptyset$$

where $X(t)$ designates the region of $\mathcal{W}$ occupied by the object X at time t. Besides Γ must be *feasible*, *i.e.* it must respect the constraints (1)-(5) presented earlier. Finally we are interested in finding a time-optimal trajectory, *i.e.* a solution Γ such that t_f should be minimal.

4 The Motion Planning Scheme

As mentioned earlier, our approach addresses the problem at hand in two complementary stages. The first stage — *paths-planning* — computes a set of adjacent paths, one of which leads $\mathcal{A}$ to its goal. These paths are collision-free with the stationary obstacles and respect the kinematic constraints of $\mathcal{A}$. Given that $\mathcal{A}$ is able to shift from one path to an adjacent one, the second stage — *trajectory-planning* — determines the motion of $\mathcal{A}$ along and between these paths so as to avoid the moving obstacles while respecting the dynamic constraints of $\mathcal{A}$.

4.1 Paths-Planning

Paths-planning is performed in three steps. To begin with, a nominal path is computed (§4.1.1). Afterwards a set of adjacent paths is automatically derived from this nominal path (§4.1.2). As we will see further down, these adjacent paths are not necessarily collision-free with the stationary obstacles and do not necessarily respect the kinematic constraint (2). In order to solve these two problems, parts of the adjacent paths have to be invalidated (§4.1.3).

4.1.1 Planning a Nominal Path

A feasible path for $\mathcal{A}$ is a continuous sequence of configurations, *i.e.* a curve in the $(xy\theta)$-space that must respect the non-holonomic constraints (1) and (2). However (1)

implies that the (xy)-curve followed by R, say Π, completely defines a path for $\mathcal{A}$.

As a consequence of (1) and (3), Π must be of class C^1 (a curve is of class C^n if it is differentiable n times and if its n^{th} derivative is continuous). Besides (2) implies that the curvature of Π (wherever it is defined) must be upper-bounded by $1/\rho_{\min}$.

A path Π which is of class C^1 and whose curvature is upper-bounded by $1/\rho_{\min}$, is feasible but, it is important to note that $\mathcal{A}$ has to stop whenever a curvature discontinuity occurs (so as to change its front wheels' orientation). Our main concern being in planning 'high' speed and forward motions only, Π is furthermore defined as a planar curve of class C^2. The C^2 property insures that the path is manœuvre-free and that $\mathcal{A}$ can follow it without having to stop (no curvature discontinuity).

Path planning for car-like robots such as $\mathcal{A}$ is an issue that has been largely addressed in the past ten years and several path planners have been proposed, *e.g.* [1, 13, 16, 29]. However all these path planners generate paths made up of straight segments connected with tangential circular arcs of minimum radius (such paths are the shortest ones for car-like robots [6, 20]). Unfortunately this type of path is not C^2. Accordingly we developed the first C^2 path planner for car-like robots. This planner has already been presented in [23] so we will not detail it here. Suffice it to say that it generates a nominal path Π_N that is feasible and collision-free.

4.1.2 Computing Adjacent Paths

Recall that a path for $\mathcal{A}$ is a (xy)-curve Π of class C^2 whose curvature is upper-bounded by $1/\rho_{\min}$. Assuming that Π does not intersect itself, a point P' located at a distance from Π smaller than $\rho_{\min}$ has a unique normal projection P on Π. Let $d_\Pi(P')$ be the signed distance[2] between P' and its projection P on Π. The path *adjacent to* Π *on its left at a distance* δL is defined as: $\Pi_l = \{P' \in \mathcal{W} \mid d_\Pi(P') = \delta L\}$ (Fig.2). Similarly, the path *adjacent to* Π *on its right at a distance* δL is defined as: $\Pi_r = \{P' \in \mathcal{W} \mid d_\Pi(P') = -\delta L\}$.

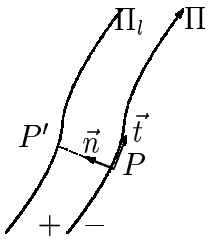

Figure 2: Π_l, a path adjacent to Π on its left.

[2] If P' is on the left (resp. right) side of Π, then $d_\Pi(P')$ is positive (resp. negative).

Thanks to this definition, it is possible to recursively compute a set of paths adjacent to the nominal path Π_N on its left and on its right. Let $\mathcal{P} = \{\Pi_k, k \in \{1, \ldots, n\}\}$ be the whole set of adjacent paths, Π_N included. Let us denote by $\mathcal{R}(\Pi)$ the set of points whose distance to Π is smaller than $\delta L/2$. The distance δL between any two adjacent paths is chosen so as to ensure that, $\forall \Pi \in \mathcal{P}$:

1. The region swept by $\mathcal{A}$ when it follows a path Π is included in $\mathcal{R}(\Pi)$.
2. The region swept by $\mathcal{A}$ on its left (resp. right) side when it leaves a path Π by making a right (resp. left) turn of radius $\rho_{\min}$, is included in $\mathcal{R}(\Pi)$ (Fig.3a).
3. The region swept by $\mathcal{A}$ on its left (resp. right) side when it reaches a path Π by making a right (resp. left) turn of radius $\rho_{\min}$ is included, in $\mathcal{R}(\Pi)$ (Fig.3b).

As we will see further down, these three properties permits to simplify collision checking both along (§4.1.3) and between the paths (§4.2).

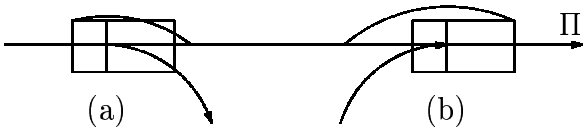

Figure 3: regions swept by $\mathcal{A}$ on its left when leaving/reaching a path Π by making a right turn.

4.1.3 Checking Adjacent Paths

The nominal path is collision-free with the stationary obstacles and it respects the kinematic constraints (1) and (2). Unfortunately these two properties do not necessarily hold for an adjacent path. It may happen that such a path is no longer collision-free with the stationary obstacles or does no longer respect the curvature constraint (2). Let $\Pi \in \mathcal{P}$: Π is a mapping $[0,1] \longrightarrow \mathbb{R}^2$. Collision and curvature checking lead us to compute a set of 'forbidden' intervals in the range $[0,1]$ corresponding to parts of Π which violate either of these constraints. Let us denote by $\rho(s)$ the curvature radius of the path at the point $\Pi(s)$. The set of forbidden intervals for Π is formally defined as:

$$\begin{aligned}\mathcal{F}(\Pi) \quad = \quad & \{[a,b] \subset [0,1] \mid (\forall s \in [a,b],\ \rho(s) < \rho_{\min}) \\ & \text{or } (\exists \mathcal{B}_i^S \mid \text{Proj}(\mathcal{B}_i^S, \Pi) = [a,b])\}\end{aligned}$$

where $\text{Proj}(\mathcal{B}_i^S, \Pi)$ is the 'projection' of the stationary obstacle $\mathcal{B}_i^S$ on the path Π, *i.e.* the part of Π which entails a collision between $\mathcal{A}$ and $\mathcal{B}_i^S$:

$$\begin{aligned}\text{Proj}(\mathcal{B}_i^S, \Pi) \quad = \quad & [a,b] \subset [0,1] \mid \forall s \in [a + l_f, b - l_r], \\ & \exists P \in \mathcal{B}_i^S \cap \mathcal{R}(\Pi) \mid P' = \Pi(s)\end{aligned}$$

where P' is the normal projection of P on Π and where l_f (resp. l_r) denotes the distance between $\mathcal{A}$'s rear axle and

it front-most (resp. rear-most) point. Figure 4 depicts an example of forbidden intervals for a path Π. The interval $[a_1, b_1]$ corresponds to Proj($\mathcal{B}_i^S$,Π) while $[a_2, b_2]$ corresponds to a part of Π which does not respect the curvature constraint (2).

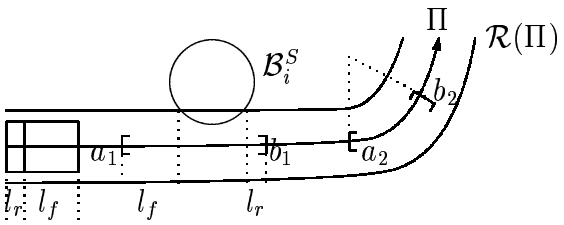

Figure 4: the 'forbidden' intervals of Π.

Note that Proj($\mathcal{B}_i^S$,Π) is defined by using $\mathcal{R}(\Pi)$ instead of the exact region swept out by $\mathcal{A}$ when it moves along Π. This choice is sound because the region swept out by $\mathcal{A}$ is included in $\mathcal{R}(\Pi)$ (*cf.* property #1 of §4.1.2). Besides it simplifies the computation of Proj($\mathcal{B}_i^S$,Π) because $\mathcal{R}(\Pi)$ has a more simple shape.

4.2 Trajectory-Planning

The output of paths-planning is a set $\mathcal{P}$ of adjacent paths and a set $\mathcal{F}$ of forbidden intervals associated with each path. The purpose of trajectory-planning is to determine $\mathcal{A}$'s motion along and between these paths so as to stay out of the forbidden intervals, avoid the moving obstacles and respect the dynamic constraints (3), (4) and (5).

Because of path-changing, *i.e.* the motion between adjacent paths, the problem at hand is two-dimensional. However it is possible to take advantage of the properties of the adjacent paths in order to reduce the problem to a one-dimensional trajectory planning problem. We have designed a trajectory planner which implements this idea. To begin with, we present a method which determines the trajectory of $\mathcal{A}$ along a given path and then we extend this method so as to incorporate path-changing.

4.2.1 Motion Along a Path

We designed a trajectory planner which is able to compute the trajectory of a given robot $\mathcal{A}$ along a given path Π so as to avoid moving obstacles and respect dynamic constraints such as (3), (4) and (5). This planner is described in [9]. It reduces the original problem to a one-dimensional problem by parameterizing Π with a single variable p representing the distance traveled along Π. Afterwards the dynamic constraints of $\mathcal{A}$ are transformed into constraints on the velocity v and the acceleration a along Π. The constraints on v are expressed by a velocity limit curve in the state space, *i.e.* the $p\times v$ plane. On the other hand, the constraints imposed by the moving obstacles can be represented by forbidden regions of the $p\times t$ space, where t represents the time dimension[14]. In order to deal simultaneously with these two types of constraints, we introduced the novel concept of **state-time space**, *i.e.* the $p\times v\times t$ space [10]. Let $\mathcal{ST}$ be this state-time space. A curve of $\mathcal{ST}$ represents a trajectory along Π. Therefore it is possible to solve the problem at hand by searching a curve in $\mathcal{ST}$. The algorithm we designed in order to find such a trajectory operates in the following way: it chooses a time-step τ and assumes that the acceleration applied to $\mathcal{A}$ during a time-step τ is either minimum, null or maximum. Accordingly a state-time has three neighbours and all the state-times that $\mathcal{A}$ can reach from a given state-time lie on a regular grid embedded in $\mathcal{ST}$ (Fig.5). This grid is then searched in order to find a solution. Accordingly trajectory planning is reduced to graph search.

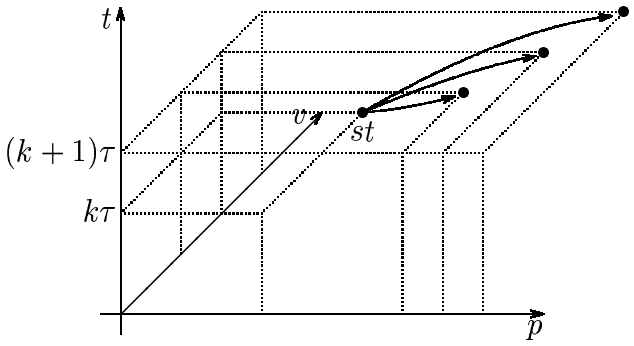

Figure 5: the grid embedded in $\mathcal{ST}$.

4.2.2 Motion Between the Paths

Let us consider a path-changing motion as depicted in Fig.6a. At time t_1, $\mathcal{A}$ shifts smoothly from its current path Π_k to an adjacent path Π_{k+1}. $\mathcal{A}$ follows a nominal trajectory Ω and reaches Π_{k+1} at a certain time t_2.

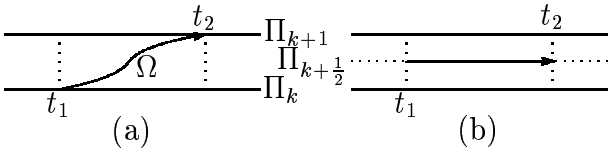

Figure 6: path-changing.

Properties #2 and #3 of §4.1.2 ensure that, when $\mathcal{A}$ moves along Ω, it always remains included in the domain $\mathcal{R}(\Pi_k)\cup\mathcal{R}(\Pi_{k+1})$. Consequently, evaluating whether Ω is collision-free can be done very simply by checking out potential collision in both paths Π_k and Π_{k+1} during the time interval $[t_1, t_2]$. Note that, in this case, the obstacles on Π_k

include both the forbidden intervals $\mathcal{F}(\Pi_k)$ and the projections of the moving obstacles $\mathrm{Proj}(\mathcal{B}_j^M(t),\Pi_k)$ — such projections being time-dependent.

Accordingly it is possible to model path-changing as a simultaneous motion along Π_k and Π_{k+1} during $[t_1, t_2]$, or equally, as a three-step process: (a) at time t_1, $\mathcal{A}$ instantaneously shifts from Π_k to a fictitious intermediate path $\Pi_{k+\frac{1}{2}}$, (b) $\mathcal{A}$ moves along $\Pi_{k+\frac{1}{2}}$ during $[t_1, t_2]$ (the obstacles of both Π_k and Π_{k+1} are assumed to be projected on $\Pi_{k+\frac{1}{2}}$ and (c) at time t_2, $\mathcal{A}$ instantaneously shifts from $\Pi_{k+\frac{1}{2}}$ to Π_{k+1} (Fig.6b). Accordingly this modelling reduces path-changing to a one-dimensional motion along a fictitious path.

In this framework. a state-time of $\mathcal{A}$ is a 4-tuple (L, p, v, t) where L is the index of the current path of $\mathcal{A}$. Let us choose a time-step τ and assume that the acceleration applied to $\mathcal{A}$ is either minimum, null or maximum. Given a method which determines the path-changing trajectory Ω[3], it is possible to determine the state-time reached by $\mathcal{A}$ at the end of a path-changing. A given state-time has now five neighbours: three on the same path and two on each adjacent paths. All the state-times reachable from a given state-time still lie on a regular grid embedded in $\mathcal{ST}$ and it is still possible to search this grid in order to find a solution (Fig.7).

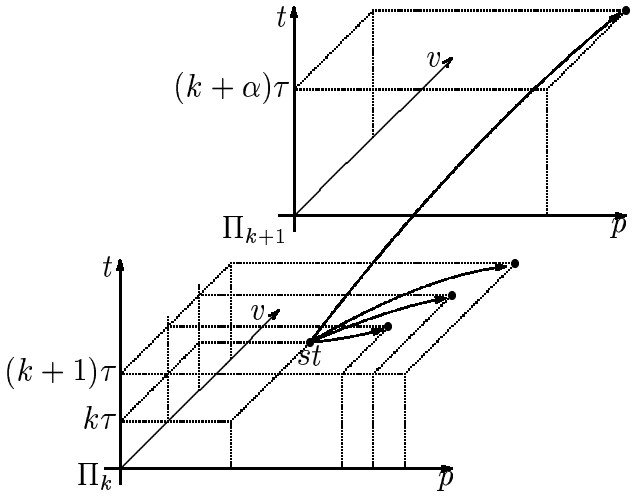

Figure 7: the grid embedded in the extended $\mathcal{ST}$. $\alpha\tau$ is the time necessary to perform the path-changing.

5 Experimental Results

The algorithm presented above has been implemented in C on a Sun Sparc I. We have tested the algorithm with up to four adjacent paths. In these experiments, the moving obstacles are generated at random without caring whether they collide with each other. Two examples of trajectory planning involving two paths are depicted in Figs 8 and 9. In each case, a path is associated with two windows: a trace window showing the part of the grid which has been explored and a result window displaying the final trajectory. Such a window represents the 'time×position' space of the path (the position axis is horizontal while the time axis is vertical; the frame origin is at the upper-left corner). The thick black segments represent the trails left by the moving obstacles and the little dots are points of the underlying grid. Note that the vertical spacing of the dots corresponds to the time-step τ.

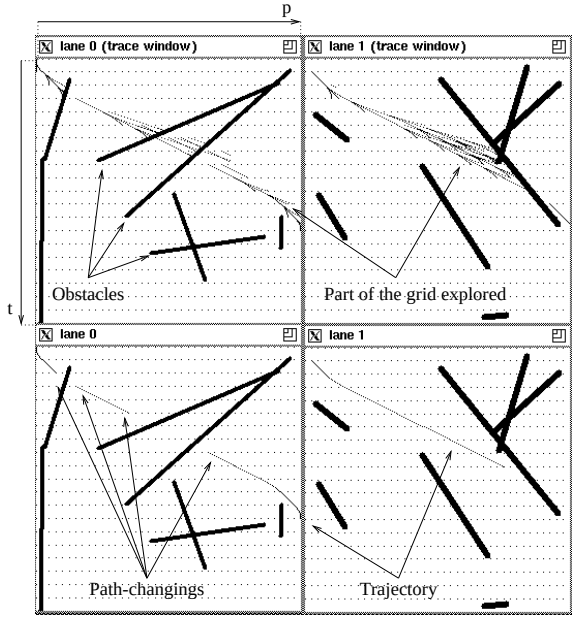

Figure 8: an example of trajectory planning with two paths: the solution has four path-changings.

In both examples, $\mathcal{A}$ starts from the first lane (`lane #0`), at position 0 (upper-left corner) and with a null velocity. It must reach the first path at position $p_{\max}$ (right border) with a null velocity. $\mathcal{A}$ can overtake by using the second path (`lane #1`). In order to simulate the behaviour of a car on the roadway, we have chosen the following values for the various variables of the problem:

$$\begin{cases} p_{\max} = 500\text{m} \\ \delta L = 4\text{m} \\ v_{\max} = 72\text{km/h} \\ a_{\max} = 1\text{m/s}^2 \end{cases}$$

It is the choice of τ that determines the size of the grid embedded in $\mathcal{ST}$, and thus the average running time of the algorithm. For a value of τ set to 5s., we have obtained running times ranging from less than a second to a few seconds.

[3]Such a method is described in [10] for a car-like robot.

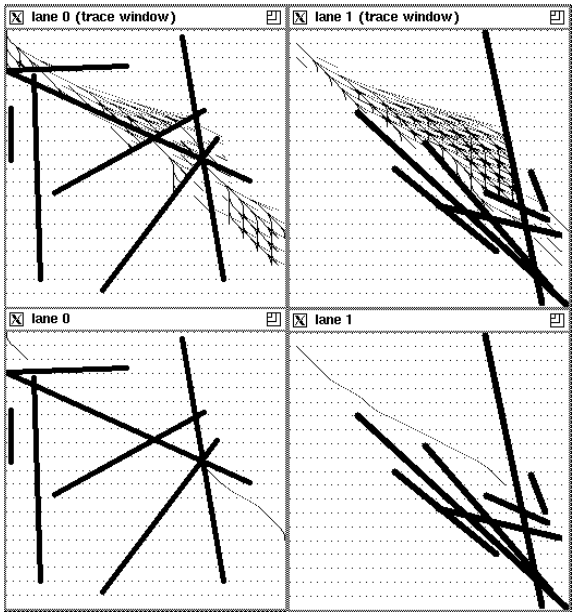

Figure 9: an example of trajectory planning with two paths: the solution trajectory has two path-changings.

6 Conclusion and Discussion

This paper addressed trajectory planning in dynamic workspaces, *i.e.* motion planning for a robot subject to dynamic constraints and moving in a dynamic workspace. The case of a car-like robot $\mathcal{A}$ with bounded velocity and acceleration, moving in a dynamic workspace $\mathcal{W} = \mathbb{R}^2$, was considered. Path-velocity decomposition is a practical way to address trajectory planning in dynamic workspaces since it decomposes the original problem into two more simple sub-problems. However it presents a serious drawback: it cannot find a solution if a moving obstacle stops right on the computed path. A possible answer to this problem is to consider a set of candidate paths. The answer proposed in this paper is the novel concept of *adjacent paths*: this concept was introduced and used within a novel planning schema that operates in two complementary stages: (a) *paths planning* and (b) *trajectory planning*. In the paths planning stage, a set of adjacent paths (like adjacent lanes of the roadway), one of which leads $\mathcal{A}$ to its goal, are computed. These paths are collision-free with the stationary obstacles and respect $\mathcal{A}$'s kinematic constraints. In the trajectory planning stage, given that $\mathcal{A}$ is able to shift from one path to an adjacent one freely, the motion of $\mathcal{A}$ along and between these paths is determined so as to avoid the moving obstacles while respecting $\mathcal{A}$'s dynamic constraints. The fact that it is possible to switch several times between two adjacent paths makes this approach more flexible and more powerful than one considering candidate paths. Although the concept of adjacent paths is particularly well suited for motion planning in two-dimensional workspaces, it could be applied to three-dimensional workspaces too.

Acknowledgements: this work was partially supported by the European Eureka EU-153 project "Prometheus Pro-Art".

References

[1] J. Barraquand and J-C Latombe. On non-holonomic mobile robots and optimal maneuvering. *Revue d'intelligence Artificielle*, 3(2):77–103, 1989.

[2] J.E. Bobrow, S. Dubowsky, and J.S. Gibson. Time-optimal control of robotic manipulators along specified paths. *Int. Journal of Robotics Research*, 4(3):3–17, Fall 1985.

[3] J. Canny. *The complexity of robot motion planning.* MIT Press, Cambridge, MA (USA), 1988.

[4] J. Canny, B. Donald, J. Reif, and P. Xavier. On the complexity of kynodynamic planning. In *Proc. of the IEEE Symp. on the Foundations of Computer Science*, White Plains, NY (USA), November 1988.

[5] J. Canny, A. Rege, and J. Reif. An exact algorithm for kinodynamic planning in the plane. In *Proc. of the ACM Symp. on Computational Geometry*, pages 271–280, Berkeley, CA (USA), 1990.

[6] L.E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[7] M. Erdmann and T. Lozano-Perez. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.

[8] P. Fiorini and Z. Shiller. Time optimal trajectory planning in dynamic environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Minneapolis MN (US), April 1996.

[9] Th Fraichard. Dynamic trajectory planning with dynamic constraints: a 'state-time space' approach. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1394–1400, Yokohama (JP), July 1993.

[10] Th. Fraichard and C. Laugier. Kinodynamic planning in a structured and time-varying workspace. In C. Laugier, editor, *Geometric Reasoning for Perception and Action*, volume 708 of *Lecture Notes in Computer Science*, pages 19–37. Springer-Verlag, 1993.

[11] Th. Fraichard and A. Scheuer. Car-like robots and moving obstacles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 64–69, San Diego CA (US), May 1994.

[12] K. Fujimura and H. Samet. A hierarchical strategy for path planning among moving obstacles. *IEEE Trans. Robotics and Automation*, 5(1):61–69, February 1989.

[13] P. Jacobs, J-P Laumond, and A. Rege. Nonholonomic motion planning for hilare-like mobile robots. In M. Vidyasagar and M. Trivedi, editors, *Intelligent Robotics*, pages 338–347. McGraw-Hill, Bangalore (In), January 1991.

[14] K. Kant and S. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *Int. Journal of Robotics Research*, 5(3):72–89, Fall 1986.

[15] J-C. Latombe. *Robot motion planning.* Kluwer Academic Press, 1990.

[16] J-P. Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, pages 1120–1123, Milan (I), August 1987.

[17] N.J. Nilsson. A mobile automaton: an application of artificial intelligence techniques. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, pages 509–520, Washington, DC (USA), 1969.

[18] C. Ó'Dúnlaing. Motion planning with inertial constraints. *Algorithmica*, 2:431–475, 1987.

[19] T-J. Pan and R.C. Luo. Motion planning for mobile robots in a dynamic environment with moving obstacles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH (USA), May 1990.

[20] J.A. Reeds and L.A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[21] J. Reif and M. Sharir. Motion planning in the presence of moving obstacles. In *Proc. of the IEEE Symp. on the Foundations of Computer Science*, pages 144–154, Portland, OR (USA), October 1985.

[22] G. Sahar and J. H. Hollerbach. Planning of minimum-time trajectories for robot arms. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 751–758, St Louis, MI (USA), March 1985.

[23] A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble (FR), September 1997.

[24] C.L. Shih, T.T. Lee, and W.A. Gruver. Motion planning with time-varying polyhedral obstacles based on graph search and mathematical programming. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Cincinnatti, OH (USA), May 1990.

[25] Z. Shiller and J.C. Chen. Optimal motion planning of autonomous vehicles in three-dimensional terrains. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Cincinnatti, OH (USA), May 1990.

[26] Z. Shiller and S. Dubowsky. Global time optimal motions of robotic manipulators in the presence of obstacles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 370–375, Philadelphia, PA (USA), April 1988.

[27] Z. Shiller and S. Dubowsky. Robot path planning with obstacles, actuator, gripper and payload constraints. *Int. Journal of Robotics Research*, 8(6):3–18, December 1989.

[28] Z. Shiller and H-H. Lu. Robust computation of path constrained time-optimal motion. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 144–149, Cincinnatti, OH (USA), May 1990.

[29] P. Tournassoud and O. Jehl. Motion planning for a mobile robot with a kinematic constraint. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA (USA), April 1988.

[30] P.G. Xavier. *Provably-good approximation algorithms for optimal kinodynamic robot motion plans.* PhD thesis, Cornell Univ., Ithaca, NY (USA), april 1992.