IEEE Int. Conf. on Robotics and Automation,
May 8-13, 1994. San Diego, California. Volume 1, pp.64–69

Car-Like Robots and Moving Obstacles[a]

Th. Fraichard & A. Scheuer
LIFIA, INRIA Rhône-Alpes, IMAG
46, av. Félix Viallet, 38031 Grenoble Cedex 1, France
[Thierry.Fraichard, Alexis.Scheuer]@imag.fr

Abstract — *This paper addresses motion planning for a car-like robot moving in a changing planar workspace, i.e. with moving obstacles. First, this motion planning problem is formulated in the* state-time space *framework. The state-time space of a robot is its state space plus the time dimension. In this framework part of the constraints at hand are translated into static forbidden regions of state-time space, and a trajectory maps to a state-time curve which must respect the remaining constraints. Then an approximate solution to the problem is presented.*

[a]This work was jointly supported by the European EUREKA EU-153 project "PROMETHEUS Pro-Art" and the French INRIA-INRETS Praxitèle programme.

Car-Like Robots and Moving Obstacles*

Th. Fraichard & A. Scheuer
LIFIA, INRIA Rhône-Alpes, IMAG
46, av. Félix Viallet, 38031 Grenoble Cedex 1, France
[Thierry.Fraichard, Alexis.Scheuer]@imag.fr

Abstract — *This paper addresses motion planning for a car-like robot moving in a changing planar workspace, i.e. with moving obstacles. First, this motion planning problem is formulated in the* state-time space *framework. The state-time space of a robot is its state space plus the time dimension. In this framework part of the constraints at hand are translated into static forbidden regions of state-time space, and a trajectory maps to a state-time curve which must respect the remaining constraints. Then an approximate solution to the problem is presented.*

1 Introduction

Imagine a car-like robot moving on the roadway or a shop floor. It must avoid collision with the various obstacles, moving or not. Besides such a robot has a limited steering range, this is a kinematic constraint which restricts the geometric shape of its motion. Finally it is subject to various dynamic constraints (engine power, ground/wheel interaction, etc.) which cannot be neglected especially if it moves fast. In short planning a motion for this type of robot does require to take into account all the various constraints which restrict its motion capabilities, i.e. *kinematic* and *dynamic constraints, stationary* and *moving obstacles.*

In the past 25 years, a lot of research works have addressed motion planning but most of them have focused on *path planning*, i.e. computing the geometric path to be followed[1], and very little attention has been paid to what can be referred to as *dynamic trajectory planning*, i.e. motion planning including all the aforementioned constraints.

This paper follows upon [4] which introduced the novel concept of *state-time space*[2] as a tool to address dynamic trajectory planning. In this framework part of the constraints at hand are translated into static forbidden regions of state-time space, and a trajectory maps to a state-time curve which must respect the remaining constraints. [4] addressed the case of a car-like robot moving along a given path. In this paper we address the more general case of a car-like robot moving on $\mathbb{R}^2$ and concentrate on planning forward motions only, i.e. without backing-up manœuvres. We start by formulating the problem at hand in the state-time space framework. Then we present the solution we have designed. For reasons which are to be discussed later, this method follows the paradigm of near-time-optimization [2, 7]: the search for the time-optimal solution is performed over a restricted set of *canonical trajectories*, hence the near-time-optimality of the solution. These canonical trajectories are defined as having a piecewise constant acceleration that can only change its value at given times. Besides the acceleration is selected among a finite and discrete set. Under these assumptions, it is possible to transform the problem of finding the time-optimal canonical trajectory to finding the shortest path in a directed graph embedded in state-time space.

Content of the Paper. After a short review of the works related to dynamic trajectory planning (§2), the robot and its workspace are described (§3 and 4). Then the problem at hand is formulated in the state-time space framework (§5). Finally a solution algorithm is presented (§6).

2 Related Works

To the best of our knowledge, [5] and [11] are the only references which truly address dynamic trajectory planning. However they do so with far too simplifying assumptions. [11] considers a particle with bounded acceleration moving along a given path in-between two moving particles. The solution proposed is exact and operates both in state space and configuration-time space. The algorithm presented in [5] deals with a particle with bounded velocity, acceleration and centrifugal force which moves on $\mathbb{R}^2$ among translating polygons. The solution proposed is approximate and operates in the discretized configuration-time space.

On the other hand there is a large body of works which are of some interest to our problem since they address either moving obstacles, e.g. [3, 6, 8], or dynamic constraints, e.g. [13, 12, 2, 7]. Through lack of space they will not be reviewed here. Suffice it to say that they are interesting because they emphasize the role of configuration-time space as a tool to deal with moving obstacles, and state-space as a tool to deal with dynamic constraints. Accordingly it seems only natural to merge these two concepts within state-time space when it comes to address dynamic trajec-

*This work was supported by the European EUREKA EU-153 project "PROMETHEUS Pro-Art" and the French INRIA-INRETS Praxitèle programme.

[1] The reader is referred to [9] for a recent survey of this topic.

[2] The state space of a robot is the space of its configuration parameters and their derivatives. State-time space is state space plus the time dimension.

tory planning.

3 The Robot $\mathcal{A}$

In the next two sections we start by presenting the kinematic model of a car-like robot $\mathcal{A}$ so as to derive the corresponding kinematic constraints. Then we state the simplified dynamic model of $\mathcal{A}$ that we are going to use along with the corresponding dynamic constraints.

3.1 The Kinematics of $\mathcal{A}$

Let $\mathcal{A}$ be a car-like robot. It is modelled as a rigid body moving on the plane R^2. It is supported by four wheels making point contacts with the ground. $\mathcal{A}$ has two rear wheels and two directional front wheels. A configuration of $\mathcal{A}$ is defined by a triple $(x, y, \theta) \in \mathrm{R}^2 \times [0, 2\pi[$ where (x, y) are the coordinates of the rear axle midpoint R and θ the orientation of $\mathcal{A}$, i.e. the angle between the x axis and the main axis of $\mathcal{A}$ (Fig. 1).

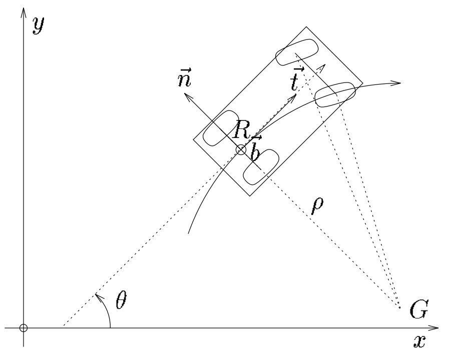

Figure 1: a car-like robot.

A body moving on the plane has only one rotation centre. Under perfect rolling assumption, a wheel must 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 rotation centre of $\mathcal{A}$. The orientation of the rear wheels being fixed, G must be 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:

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \quad (1)$$

Besides, owing 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 by a certain value ρ_{min} and the following constraint holds:

$$\dot{x}^2 + \dot{y}^2 - {\rho_{min}}^2\dot{\theta}^2 \geq 0 \quad (2)$$

Relations (1) and (2) are non-holonomic [1]. As depicted in Fig. 2, they compel $(\dot{x}, \dot{y})$ to point forward or backward along the main axis of $\mathcal{A}$ and $(\dot{x}, \dot{y}, \dot{\theta})$ to be in a two-sided cone contained in the plane perpendicular to the xy-plane that projects on this plane along the main axis of $\mathcal{A}$ [9, chapter 9].

Furthermore our main concern being in planning forward motions only, $(\dot{x}, \dot{y})$ should be not null and point forward

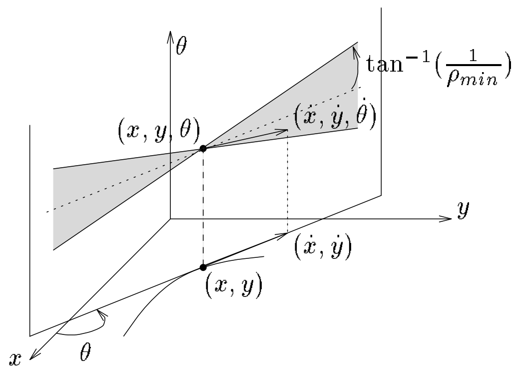

Figure 2: set of possible velocity vectors of R.

along $\mathcal{A}$'s main axis. Accordingly, it should satisfy the following constraint:

$$\dot{x}\cos\theta + \dot{y}\sin\theta > 0 \quad (3)$$

When (3) holds, θ can be derived in a unique and straightforward manner from $(\dot{x}, \dot{y})$. This property can be used to address trajectory planning for $\mathcal{A}$ in the xy-space instead of the $xy\theta$-space. More precisely $\mathcal{A}$ will be assimilated to a point moving on R^2 whose trajectory is a time-indexed curve of the xy-space.

3.2 The Dynamics of $\mathcal{A}$

As mentioned earlier, $\mathcal{A}$ is assimilated to a point, say R, moving on R^2. In other words the moment of inertia of $\mathcal{A}$ is neglected and its centre of mass is assumed to be R. Finally the reaction and friction forces between the wheels and the ground are transferred to R. A *configuration* of $\mathcal{A}$ is henceforth defined by a pair $q = (x, y) \in \mathrm{R}^2$ and a *state* of $\mathcal{A}$ by a pair $(q, \dot{q})$ where $\dot{q} = (\dot{x}, \dot{y}) \in \mathrm{R}^2$. Let $(\vec{t}, \vec{n}, \vec{b})$ be the inertial frame attached to $\mathcal{A}$ at point R. The $\vec{b}$ axis points towards the positive direction normal to the plane. The $\vec{n}$ axis is chosen so that $(\vec{t}, \vec{n}, \vec{b})$ is right-handed (Fig. 1). It stems from (1) and (3) that the orientation of $\vec{t}$ should be θ. The motion of $\mathcal{A}$ obeys Newtonian dynamics. The external forces acting on $\mathcal{A}$ are the gravity $\vec{G}$ and the ground reaction $\vec{R}$ which can be decomposed into their perpendicular components:

$$\vec{G} = -mg\,\vec{b} \quad (4)$$

$$\vec{R} = R_t\,\vec{t} + R_n\,\vec{n} + R_b\,\vec{b} \quad (5)$$

where m is the mass of $\mathcal{A}$ and g the gravity constant. The equation of motion of $\mathcal{A}$ can be expressed in terms of the tangent acceleration a_t and the normal acceleration a_n, namely $\vec{G} + \vec{R} = ma_t\,\vec{t} + ma_n\,\vec{n}$. Using (4) and (5), this equation can be rewritten:

$$R_t = ma_t \quad (6)$$

$$R_n = ma_n \quad (7)$$

$$R_b = mg \quad (8)$$

$\mathcal{A}$ is subject to various dynamic constraints (curvature, engine force, sliding and velocity). Each constraint can be transformed into constraints on the velocity and acceleration as shown below.

Curvature Constraint. Let κ be the signed curvature of the trajectory followed by R (κ is positive if the radial direction coincides with $\vec{n}$ and negative otherwise). The lower bound (2) on the curvature radius ρ entails that $|\kappa| \leq \kappa_{max}$ where $\kappa_{max} = 1/\rho_{min}$. Accordingly, a_n, which is equal to κv^2, where v is the tangent velocity of R, is bounded in the following way: $|a_n| \leq \kappa_{max} v^2$, or equivalently:

$$\frac{|-\dot{y}\ddot{x} + \dot{x}\ddot{y}|}{\sqrt{\dot{x}^2 + \dot{y}^2}} \leq \kappa_{max}(\dot{x}^2 + \dot{y}^2) \quad (9)$$

Engine Force Constraint. When $\mathcal{A}$ is moving, the torque applied by the brakes or the engine on the wheels is translated into a planar force of direction $\vec{t}$ whose modulus $F = ma_t$ is bounded in the following way: $F_{min} \leq F \leq F_{max}$, where F_{min} is the minimum braking force and F_{max} the maximum engine force. This relation yields the following feasible tangent acceleration range: $F_{min}/m \leq a_t \leq F_{max}/m$, which can be expressed equivalently by:

$$\frac{F_{min}}{m} \leq \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \leq \frac{F_{max}}{m} \quad (10)$$

Sliding Constraint. The component of $\vec{R}$ in the plane $\vec{t} \times \vec{n}$ represents the friction that is applied from the ground to the wheels. It is constrained by the following relation: $\sqrt{{R_t}^2 + {R_n}^2} \leq \mu R_b$ where μ is the friction coefficient between the wheels and the ground. If this constraint is violated then $\mathcal{A}$ slides. Substituting (6), (7) and (8) in this relation yields the following feasible acceleration range due to the sliding constraint: ${a_t}^2 + {a_n}^2 \leq \mu^2 g^2$, which can be expressed equivalently by:

$$\ddot{x}^2 + \ddot{y}^2 \leq \mu^2 g^2 \quad (11)$$

Note that this constraint implies that $|\ddot{x}|$ and $|\ddot{y}|$ are both upper-bounded by μg.

Velocity Constraint. The modulus of $\mathcal{A}$'s tangent velocity $\dot{q}$ cannot be null or greater than a certain value v_{max}:

$$0 < \dot{x}^2 + \dot{y}^2 \leq {v_{max}}^2 \quad (12)$$

In short a trajectory of $\mathcal{A}$ must satisfy the whole set of constraints (9), (10), (11) and (12). Note that (12) depends solely on the current state of $\mathcal{A}$. It is a *state constraint* of $\mathcal{A}$. It yields a set of forbidden states, say $\mathcal{FS}$, which is defined as such:

$$\mathcal{FS} = \{(q, \dot{q}) | \dot{x}^2 + \dot{y}^2 \leq 0 \text{ or } \dot{x}^2 + \dot{y}^2 > {v_{max}}^2\}$$

As to (9), (10) and (11), they reduce the set of accelerations $\ddot{q} = (\ddot{x}, \ddot{y})$ which can be applied to $\mathcal{A}$ at a given state. Actually they yield a set of admissible accelerations which depend solely on the current velocity $\dot{q}$ of $\mathcal{A}$ and which is formally defined as such:

$$\mathcal{AA}(\dot{q}) = \left\{ \ddot{q} \;\middle|\; \begin{array}{l} -\dot{y}\ddot{x} + \dot{x}\ddot{y} - \kappa_{max}\sqrt{\dot{x}^2 + \dot{y}^2}^3 \leq 0 \\ \dot{y}\ddot{x} - \dot{x}\ddot{y} - \kappa_{max}\sqrt{\dot{x}^2 + \dot{y}^2}^3 \leq 0 \\ \dot{x}\ddot{x} + \dot{y}\ddot{y} - F_{max}\sqrt{\dot{x}^2 + \dot{y}^2}/m \leq 0 \\ -\dot{x}\ddot{x} - \dot{y}\ddot{y} + F_{min}\sqrt{\dot{x}^2 + \dot{y}^2}/m \leq 0 \\ \ddot{x}^2 + \ddot{y}^2 - \mu^2 g^2 \leq 0 \end{array} \right\}$$

Trivially $\mathcal{AA}(\dot{q})$ is a convex region of the $\ddot{x}\ddot{y}$-space; it is the intersection of a disk of constant radius μg and four half-planes whose boundaries are pairwise parallel and thus define a rectangle. It looks as depicted in Fig. 3.

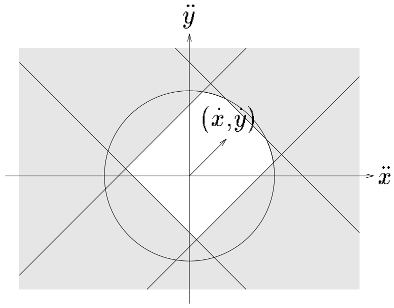

Figure 3: $\mathcal{AA}(\dot{q})$.

4 The Workspace $\mathcal{W}$

$\mathcal{A}$'s workspace, say $\mathcal{W}$, is R^2. It is cluttered up with a set of stationary and moving obstacles $\mathcal{B}_i, i \in \{1, n\}$. Let $\mathcal{B}_i(t)$ denote the region of $\mathcal{W}$ occupied by $\mathcal{B}_i$ at time t and $\mathcal{A}(q)$ the region of $\mathcal{W}$ occupied by $\mathcal{A}$ in configuration q. If, at time t, $\mathcal{A}$ is at position q and there is an obstacle $\mathcal{B}_i$ such that $\mathcal{B}_i(t)$ intersects $\mathcal{A}(q)$, then a collision occurs between $\mathcal{A}$ and $\mathcal{B}_i$. Accordingly, the constraints imposed by the moving obstacles on $\mathcal{A}$'s motion can be represented by a set of forbidden points of the $q \times t$ space. Let $\mathcal{FC}$ be this set of forbidden points, it is defined as:

$$\mathcal{FC} = \{(q, t) \mid \exists i \in \{1, n\},\ \mathcal{A}(q) \cap \mathcal{B}_i(t) \neq \emptyset\}$$

5 The State-Time Space of $\mathcal{A}$

A trajectory for $\mathcal{A}$ is a continuous time-indexed sequence of states between an initial and a final state which must verify the no-collision and dynamic constraints (9), (10), (11) and (12). However it is possible to reformulate the trajectory planning problem at hand in the *state-time space* framework [4]. A *state-time* is defined by adding the time dimension to a state; hence it is represented by a triple $(q, \dot{q}, t)$. The set of every state-time is the *state-time space* of $\mathcal{A}$, it is denoted by $\mathcal{ST}$. A state-time is *admissible* if it does not violate the no-collision and state constraint (12), i.e. if and only if:

$$s \in \mathcal{ST} \setminus (\mathcal{FC}^+ \cup \mathcal{FS}^+)$$

where $\mathcal{FC}^+$ (resp. $\mathcal{FS}^+$) is the set of state-times entailing a collision (resp. violating (12)) and $E\backslash F$ denotes the complement of F in E. $\mathcal{FC}^+$ is simply derived from $\mathcal{FC}$:

$$\mathcal{FC}^+ = \{(q, \dot{q}, t) \mid \exists i \in \{1, n\},\ \mathcal{A}(q) \cap \mathcal{B}_i(t) \neq \emptyset\}$$

and so is $\mathcal{FS}^+$ from $\mathcal{FS}$:

$$\mathcal{FS}^+ = \{(q, \dot{q}, t) | \dot{x}^2 + \dot{y}^2 \leq 0 \text{ or } \dot{x}^2 + \dot{y}^2 > v_{max}{}^2\}$$

The set of every admissible state-time is the *admissible state-time space* of $\mathcal{A}$, it is denoted by $\mathcal{AST}$.

In this framework planning a trajectory between an initial state $(q_i,\ \dot{q}_i)$ and a final state $(q_f,\ \dot{q}_f)$ can be regarded as constructing a curve $\Gamma : [0,1] \longrightarrow \mathcal{AST}$ between the initial state-time $(q_i,\ \dot{q}_i,\ 0)$ and a final state-time $(q_f,\ \dot{q}_f,\ t_f)$ where t_f is the time of the trajectory. Such a trajectory being included in $\mathcal{AST}$, it does respect the no-collision and state constraint (12). Besides the acceleration profile $\ddot{q} : [0, t_f] \longrightarrow \mathrm{R}^2$ corresponding to Γ must respect the constraints (9), (10) and (11)[3]. Finally Γ must be *strictly monotone in time* since no object can travel in the past, i.e. it must satisfy the following constraint:

$$\forall \gamma, \gamma' \in [0,1], \quad \Gamma(\gamma) = (q, \dot{q}, t),\ \Gamma(\gamma') = (q', \dot{q}', t'); \\ \gamma < \gamma' \Rightarrow t < t'$$

In short the problem to be solved can be stated formally as such: let $(q_i,\ \dot{q}_i)$ be $\mathcal{A}$'s starting state and $(q_f,\ \dot{q}_f)$ its goal state. A trajectory $\Gamma : [0,1] \longrightarrow \mathcal{AST}$ is a solution to the problem at hand if and only if:

1. $\Gamma(0) = (q_i, \dot{q}_i, 0)$ and $\Gamma(1) = (q_f, \dot{q}_f, t_f)$.
2. $\forall \gamma \in [0, t_f],\ \ddot{q}(\gamma) \in \mathcal{AA}(\dot{q}(\gamma))$.
3. Γ is strictly monotone in time.

Naturally we are interested in finding a time-optimal trajectory, i.e. a trajectory such that t_f should be minimal.

6 A Solution Algorithm

6.1 The General Idea

The method we have developed in order to solve the problem at hand was initially motivated by the paradigm of near-time-optimization [2, 7]. We compute an approximate time-optimal solution by performing the search over a restricted set of *canonical trajectories*. These canonical trajectories are defined as having a piecewise constant acceleration $\ddot{q}$ that can only change its value at given times $k\tau$ where τ is a time-step and k some positive integer. Besides $\ddot{q}$ is selected among a finite and discrete set. Under these assumptions, it is possible to transform the problem of finding the time-optimal canonical trajectory into that of finding the shortest path in a directed graph $\mathcal{G}$ embedded in $\mathcal{ST}$. The vertices of $\mathcal{G}$ form a regular grid embedded in $\mathcal{ST}$ while the edges correspond to canonical trajectory segments, each of which lasting τ. The next sections respectively present the canonical trajectories, the graph $\mathcal{G}$ and the search algorithm.

[3] $\dot{q}$ and q are respectively defined as the first and second integral of $\ddot{q}$ subject to an initial state.

6.2 The Canonical Trajectories

Before defining a canonical trajectory, let us recall that $\mathcal{AA}(\dot{q})$ characterizes the set of accelerations that can be applied to $\mathcal{A}$ at a given velocity $\dot{q}$. However this set changes as soon as $\mathcal{A}$'s velocity changes. Accordingly, if a constant acceleration $\ddot{q}$ is applied to $\mathcal{A}$ for a duration τ, it is not certain that $\ddot{q}$ will always remain in $\mathcal{AA}(\dot{q})$ as time passes. This observation leads us to define $\mathcal{CA}(\dot{q}, \tau)$ which is the set of accelerations that can be applied for a duration τ to $\mathcal{A}$ at a given velocity $\dot{q}$ and such that the corresponding trajectory violates none of the dynamic constraints (9), (10) and (11). $\mathcal{CA}(\dot{q}, \tau)$ is the *conservative acceleration space*, it is formally defined in appendix A.

As mentioned earlier, a canonical trajectory has a piecewise constant acceleration $\ddot{q}$ that can only change its value at given times. Let $\dot{q}$ be $\mathcal{A}$'s velocity at a given time. As in [2], $\ddot{q}$ is selected out of nine extremum values only. One of these values is the null acceleration while the eight others correspond to accelerations which are either minimum, null or maximum in the tangent and normal directions of motion, thus corresponding to extremum accelerating/braking and steering commands. In our case, these extremum accelerations lie on eight *extremum vectors* defined by the $\ddot{x}\ddot{y}$-rectangle underlying $\mathcal{AA}(\dot{q})$ (cf. §3.2). These vectors respectively point towards the vertices and the midpoints of the sides of this rectangle (Fig. 4).

From a practical point of view, the acceleration space, i.e. the $\ddot{x}\ddot{y}$-space is discretized — an acceleration-step $\delta_{\ddot{x}}$ (resp. $\delta_{\ddot{y}}$) is chosen for the $\ddot{x}$ (resp. $\ddot{y}$) axis — and the acceleration applied to $\mathcal{A}$ at each time-step is selected from this discrete space. Let $\mathcal{DA}(\dot{q}, \tau)$ be the set of the nine extremum accelerations allowed for a given velocity $\dot{q}$ and time-step τ, it is informally defined as:

$$\mathcal{DA}(\dot{q}, \tau) = \left\{ \ddot{q} \;\middle|\; \begin{array}{l} \ddot{q} = (\alpha\delta_{\ddot{x}}, \beta\delta_{\ddot{y}}) \textit{ where } (\alpha, \beta) \in \mathrm{N}^2 \\ \ddot{q} \in \mathcal{CA}(\dot{q}, \tau) \\ \ddot{q} = (0,0) \textit{ or the maximum acceleration} \\ \textit{along each extremum vectors} \end{array} \right\}$$

As we will see further down, choosing $\ddot{q}$ out of $\mathcal{DA}(\dot{q}, \tau)$ yields a regular grid in $\mathcal{ST}$. Let $\Gamma : [0,1] \longrightarrow \mathcal{ST}$ be a trajectory and $\ddot{q} : [0, t_f] \longrightarrow \mathrm{R}^2$ its acceleration profile. Γ is a **canonical trajectory** if and only if:

- $\ddot{q}$ changes its value at times $k\tau$, $k \in \mathrm{N}$.
- $\ddot{q}(k\tau) \in \mathcal{DA}(\dot{q}(k\tau), \tau)$.

6.3 The State-Time Graph $\mathcal{G}$

Let s be a state-time, i.e. a point of $\mathcal{ST}$. It is a triple $(q, \dot{q}, t)$, where $q = (x, y)$. It can equivalently be represented by $s(t) = (q(t), \dot{q}(t))$. Let $s(k\tau) = (q(k\tau), \dot{q}(k\tau))$ be a state-time of $\mathcal{A}$ and $s((k+1)\tau)$ one of the state-times that $\mathcal{A}$ can reach by a canonical trajectory of duration τ. $s((k+1)\tau)$ is obtained by applying an acceleration $\ddot{q} \in \mathcal{DA}(\dot{q}(k\tau), \tau)$ to $\mathcal{A}$ for the duration τ. Accordingly,

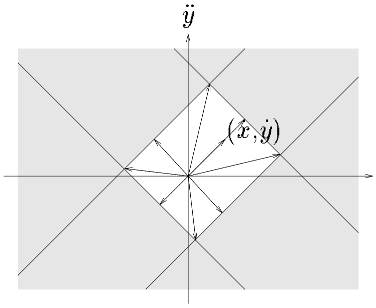

Figure 4: the eight extremum vectors for a given velocity.

we have:

$$\begin{cases} x((k+1)\tau) &= x(k\tau) + \dot{x}(k\tau)\tau + \ddot{x}\tau^2/2 \\ y((k+1)\tau) &= y(k\tau) + \dot{y}(k\tau)\tau + \ddot{y}\tau^2/2 \\ \dot{x}((k+1)\tau) &= \dot{x}(k\tau) + \ddot{x}\tau \\ \dot{y}((k+1)\tau) &= \dot{y}(k\tau) + \ddot{y}\tau \end{cases}$$

By analogy with [2], the trajectory between $s(k\tau)$ and $s((k+1)\tau)$ is called a $(\ddot{q},\tau)$-**bang**. The state-time $s((k+1)\tau)$ is reachable from $s(k\tau)$. Obviously a canonical trajectory is made up of a sequence of $(\ddot{q},\tau)$-bangs.

Let $s(m\tau)$, $m \geq k$, be a state-time reachable from $s(k\tau)$. Assuming that $\dot{x}(k\tau)$ is a multiple of $\delta_{\ddot{x}}\tau$ and that $\dot{y}(k\tau)$ is a multiple of $\delta_{\ddot{y}}\tau$, it can be shown that the following relations hold for some integers α_1, α_2, α_3 and α_4:

$$\begin{cases} x(m\tau) &= x(k\tau) + \alpha_1\delta_{\ddot{x}}\tau^2/2 \\ y(m\tau) &= y(k\tau) + \alpha_2\delta_{\ddot{y}}\tau^2/2 \\ \dot{x}(m\tau) &= \dot{x}(k\tau) + \alpha_3\delta_{\ddot{x}}\tau \\ \dot{y}(m\tau) &= \dot{y}(k\tau) + \alpha_4\delta_{\ddot{y}}\tau \end{cases}$$

Thus all the state-times reachable from one given state-time by a canonical trajectory lie on a regular grid embedded in $\mathcal{ST}$. This grid has spacings of $\delta_{\ddot{x}}\tau^2/2$ in x, of $\delta_{\ddot{y}}\tau^2/2$ in y, of $\delta_{\ddot{x}}\tau$ in $\dot{x}$, of $\delta_{\ddot{y}}\tau$ in $\dot{y}$ and of τ in time.

Consequently it becomes possible to define a directed graph $\mathcal{G}$ embedded in $\mathcal{ST}$. The nodes of $\mathcal{G}$ are the grid-points while the edges of $\mathcal{G}$ are $(\ddot{q},\tau)$-bangs between pairs of nodes. $\mathcal{G}$ is called the **state-time graph**. Let η be a node in $\mathcal{G}$, the state-times reachable from η by a $(\ddot{q},\tau)$-bang lie on the grid, they are nodes of $\mathcal{G}$. An edge between η and one of its neighbours represents the corresponding $(\ddot{q},\tau)$-bang. A sequence of edges between two nodes defines a canonical trajectory. The time of such a canonical trajectory is trivially equal to τ times the number of edges in the trajectory. Therefore the shortest path between two nodes *is* the time-optimal canonical trajectory between these nodes.

Let $\mathbf{s} = (q_i, \dot{q}_i)$ be the initial state of $\mathcal{A}$ and $\mathbf{g} = (q_f, \dot{q}_f)$ be its goal state. Without loss of generality it is assumed that the corresponding initial state-time $\mathbf{s}^* = (q_i, \dot{q}_i, 0)$ and the corresponding set of goal state-times $\mathbf{G}^* = \{(q_f, \dot{q}_f, k\tau) \text{ with } k \geq 0\}$ are grid-points. Accordingly, searching for a time-optimal canonical trajectory between $\mathbf{s}$ and $\mathbf{g}$ is equivalent to searching a shortest path in $\mathcal{G}$ between the node $\mathbf{s}^*$ and a node in $\mathbf{G}^*$.

6.4 Searching the State-Time Graph

Most classically, we use an A^* algorithm to search $\mathcal{G}$ [10]. From a practical point of view, the state-time graph $\mathcal{G}$ is embedded in a compact region of $\mathcal{ST}$. More precisely, the time component of the grid-points is upper bounded by a certain value t_{max} which can be viewed as a time-out. The number of grid-points is therefore finite and so is $\mathcal{G}$. Accordingly, the search for the time-optimal canonical trajectory can be carried out in a finite amount of time.

Implementation and Experiments. The algorithm presented earlier is currently being implemented in C on a Sun SPARC-station. It is the extension of the algorithm described in [4].

7 Conclusion and Discussion

In this paper we have studied *dynamic trajectory planning*, which is defined as trajectory planning for a robot whose motion capabilities are restricted by kinematic and dynamic constraints, and stationary and moving obstacles. Our main purpose was to take into account all these constraints in the planning process. We have addressed the case of a car-like robot moving on R^2.

To begin with we have chosen a simple yet rich enough model for our robot. Then we have formulated the motion planning problem at hand in the state-time space[4] framework. In this framework part of the constraints at hand are translated into static forbidden regions of state-time space, and a trajectory maps to a state-time curve which must respect the remaining constraints [4]. Finally we have presented an approximate solution to the problem at hand. The search for the time-optimal trajectory is performed over a restricted set of *canonical trajectories*. These canonical trajectories are defined as having a piecewise constant acceleration selected out of a finite and discrete set and which can only change its value at given times — a time-step τ is chosen. Under these assumptions, it is possible to transform the problem of finding the time-optimal canonical trajectory into that of finding the shortest path in a directed graph embedded in state-time space.

The running time of the search algorithm depends on the size of the graph which is to be explored. In turn this size is directly related to the value of the time-step τ — the smaller τ, the higher the number of vertices in the graph. On the other hand we intuitively feel that the quality of the approximation is also related to the value of τ — the smaller τ, the better the approximation. Thus it is possible to trade off the computation time against the solution quality. This property is very important and we would like to advocate this type of approach when dealing with an actual dynamic workspace. In such a workspace it is usually impossible to have a full a priori knowledge of the motion of the moving obstacles. The knowledge we have of their motions is more likely to be restricted to a certain time interval, a *time horizon*. This time horizon may represent the duration over which an estimate of the motions

[4]The state-time space of a robot is its state space plus the time dimension.

of the moving obstacles is sound. The main consequence of this assumption is to set an upper bound on the time available to plan the motion of our robot (in a highly dynamic workspace, this upper bound may be very low). In this case an approach such as the one we have presented is most interesting because its average running time can be tuned w.r.t. the time horizon considered.

Besides completing the implementation of our search algorithm, future directions of work should include:

- Considering more complex models for car-like robots.
- Trying to establish that our search algorithm is *provably good* [2], i.e. that any trajectory of duration t can be approximated by a canonical trajectory of duration at most $(1+\epsilon)t$ for a correct choice of τ. Such a result should tell us exactly how close to the time-optimal trajectory the time-optimal canonical trajectory is.

A Appendix

Let $\dot{q}$ be $\mathcal{A}$'s current velocity. The *conservative acceleration space* $\mathcal{CA}(\dot{q},\tau)$ is the set of accelerations that can be applied to $\mathcal{A}$ for a duration τ, and such that the corresponding $(\ddot{q},\tau)$-bang trajectory violates none of the constraints (9), (10) and (11). It is a subset of $\mathcal{AA}(\dot{q})$, i.e. the set of accelerations that can be instantaneously applied to $\mathcal{A}$. In order to characterize $\mathcal{CA}(\dot{q},\tau)$, let us study the conditions for (9), (10) and (11) to be verified along a $(\ddot{q},\tau)$-bang where $\ddot{q} \in \mathcal{AA}(\dot{q})$.

First, let us consider (11). $\ddot{q}$ is constant over $[0,\tau]$ and so is $\ddot{x}^2+\ddot{y}^2$. Accordingly, (11), which is true at time 0 since $\ddot{q} \in \mathcal{AA}(\dot{q})$, remains true along the $(\ddot{q},\tau)$-bang.

As for (10), a_t keeps on increasing over $[0,\tau]$ since its time derivative is positive. Accordingly, if (10),which is true at time 0, is also true at time τ, then it is true along the $(\ddot{q},\tau)$-bang.

Finally let us consider (9). It can be rewritten as such:

$$|\kappa| = \frac{|a_n|}{v^2} = \frac{|-\dot{y}\ddot{x}+\dot{x}\ddot{y}|}{\sqrt{\dot{x}^2+\dot{y}^2}^3} \leq \kappa_{max}$$

$-\dot{y}\ddot{x}+\dot{x}\ddot{y}$ is constant since its time derivative is null. Accordingly, κ is maximum when v is minimum. v reaches a minimum when its time derivative is null, i.e. for a value of t equal to $t_m = -\dot{x}\ddot{x} - \dot{y}\ddot{y}/\ddot{x}^2+\ddot{y}^2$. If $t_m \notin]0,\tau[$ then κ is monotone over $[0,\tau]$. and (9) is true over the $(\ddot{q},\tau)$-bang if and only if it is true at time 0 and τ. Otherwise, i.e. if $t_m \in]0,\tau[$, (9) must also be true at time t_m.

This observation leads us to define $\mathcal{RA}(\dot{q},\tau)$ which is the subset of $\mathcal{AA}(\dot{q})$ such that, if $\ddot{q}$ belongs to $\mathcal{RA}(\dot{q},\tau)$, then κ reaches its maximum in $]0,\tau[$. The relation $0 < t_m < \tau$ yields the two following inequalities:

$$\begin{cases} 0 < \frac{-\dot{x}\ddot{x}-\dot{y}\ddot{y}}{\ddot{x}^2+\ddot{y}^2} \\ \frac{-\dot{x}\ddot{x}-\dot{y}\ddot{y}}{\ddot{x}^2+\ddot{y}^2} < \tau \end{cases}$$

which can be rewritten as such:

$$\begin{cases} \ddot{q}.\dot{q} < 0 \\ \left\|\ddot{q} - \frac{1}{2\tau}\dot{q}\right\| > \|\frac{\dot{q}}{2\tau}\| \end{cases}$$

Accordingly, $\mathcal{RA}(\dot{q},\tau)$ is formally defined as:

$$\mathcal{RA}(\dot{q},\tau) = \left\{ \ddot{q} \in \mathcal{AA}(\dot{q}) \;\middle|\; \begin{array}{l} \ddot{q}.\dot{q} < 0 \\ \left\|\ddot{q} - \frac{1}{2\tau}\dot{q}\right\| > \|\frac{\dot{q}}{2\tau}\| \end{array} \right\}$$

Trivially $\mathcal{RA}(\dot{q},\tau)$ is the intersection between $\mathcal{AA}(\dot{q})$, a half-plane perpendicular to $\dot{q}$ and the exterior of a disk centered in $-\dot{q}/2\tau$ of radius $\|\dot{q}/2\tau\|$.

Finally it becomes possible to formally define $\mathcal{CA}(\dot{q},\tau)$:

$$\mathcal{CA}(\dot{q},\tau) = \left\{ \ddot{q} \;\middle|\; \begin{array}{l} \ddot{q} \in \mathcal{AA}(\dot{q}) \\ \ddot{q} \in \mathcal{AA}(\dot{q}+\ddot{q}\tau) \\ \ddot{q} \in \mathcal{RA}(\dot{q},\tau) \Rightarrow \ddot{q} \in \mathcal{AA}(\dot{q}+\ddot{q}t_m) \end{array} \right\}$$

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. 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*, pages 306–316, White Plains, NY (USA), Nov. 1988.

[3] M. Erdmann and T. Lozano-Perez. On multiple moving objects. A.I. Memo 883, MIT AI Lab., Boston, MA (USA), May 1986.

[4] 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 1393–1400, Yokohama (J), July 1993.

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

[6] K. Fujimura and H. Samet. Motion planning in a dynamic domain. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 324–330, Cincinnatti, OH (USA), May 1990.

[7] P. Jacobs, G. Heinzinger, J. Canny, and B. Paden. Planning guaranteed near-time-optimal trajectories for a manipulator in a cluterred workspace. Research Report ESRC 89-20/RAMP 89-15, Engineering Systems Research Center, Univ. of California., Berkeley, CA (USA), Oct. 1989.

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

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

[10] N.J. Nilsson. *Principles of artificial intelligence.* Morgan Kaufmann, Los Altos, CA (USA), 1980.

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

[12] 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.

[13] 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), Apr. 1988.