An Exact Algorithm for Kinodynamic Planning in the Plane

John Canny *   *Ashutosh Rege* *   *John Reif* †

Computer Science Division
University of California, Berkeley

Computer Science Department
Duke University, Durham, N.C.

Abstract

Planning time-optimal motions has been a major focus of research in robotics. In this paper, we consider the following problem : Given an object in two-dimensional physical space, an initial point and a final point, plan a time-optimal obstacle-avoiding motion for this object subject to bounds on the velocity and acceleration of the object. We give the first algorithm which solves the problem exactly in the case where the velocity and acceleration bounds are given in the L_∞ norm. We further prove the following important results : A tracking lemma and a loop elimination theorem both of which are applicable to the case of arbitrary norms. The latter result implies that, with or without obstacles, a path which intersects itself can be replaced by one which does not do so and which takes time less than or equal to that taken by the original path.

1 Introduction

Consider the following problem : We are given an object in physical space, an initial point and a final point. We have to plan a motion for this object, through physical space, avoiding obstacles present therein. Additionally, the motion should respect certain other constraints, such as given bounds on the velocity and acceleration. Depending on the constraints involved, we can therefore define a broad class of problems. *Kinodynamic planning* deals with synthesizing robot motions subject to both kinematic constraints (such as obstacles) and dynamic constraints (such as bounds on the acceleration and velocity).

A long-standing open problem in robotics has been that of devising algorithms for generating *time-optimal* motions under kinodynamic constraints. This problem has been considered previously in the literature and approximation algorithms have been provided for the two and three dimensional cases ([CDRX],[JHCP]) but with the exception of the one-dimensional case [O], no exact algorithms have been given. In this paper, we provide the first exact algorithm for time-optimal kinodynamic motion planning in the two-dimensional case.

2 The Problem Statement

Consider two-dimensional physical space, i.e. $\Re^2$, with polygonal obstacles. A point mass must be moved from a specified start position and velocity $\mathbf{S} = (\mathbf{s}, \dot{\mathbf{s}})$ to an end position and velocity $\mathbf{F} = (\mathbf{f}, \dot{\mathbf{f}})$ avoiding the obstacles. The point mass is moved by the application of command accelerations (via command forces).

*Supported by NSF Grants IRI-89-58577 and IRI-90-14490 and a David and Lucile Packard Fellowship

†Supported in part by DARPA/ARO contract DAAL03-88-K-0185, Air Force Contract AFSOR-87-0386, ONR contract N00014-K-0310, DARPA/ISTO contract N00014-88-K-0458

Denote the velocity and acceleration of the point mass over time t by $\mathbf{v}(t)$ and $\mathbf{a}(t)$ respectively. The motion is subject to dynamic constraints in the form of upper bounds on the magnitude of the velocity and acceleration in some given norm. That is,

$$\| \mathbf{v}(t) \| \leq v_{max} \tag{1}$$

and,

$$\| \mathbf{a}(t) \| \leq a_{max} \tag{2}$$

In this paper, we consider the L_∞ case, though we also provide partial results for arbitrary norms. The L_∞ case models Cartesian robots such as $RobotWorld^{TM}$. Let us denote by C, the physical space in which the object moves. Let $O \subseteq C$ be the *obstacle* space, i.e. the space occupied by the polygonal obstacles, and let $F \subseteq C$ be the free space. We assume that the obstacle space is specified as an arrangement of n *vertices* with rational coordinates which are joined together by *edges*. Let CP denote phase space, which is isomorphic to $\Re^4$. A point $\mathbf{Q}$ in CP is a pair $(\mathbf{q}, \dot{\mathbf{q}})$, corresponding to position and velocity. Similarly, let OP denote the phase obstacle space corresponding to forbidden positions and velocities, and FP denote phase free space.

In general, a kinodynamic problem is given by a tuple $(O, a_{max}, v_{max}, \mathbf{S}, \mathbf{F})$. Let $\mathbf{a} : [0, a] \rightarrow \Re^2$ be some command acceleration where $[0, a]$ is an interval in time. Denote by $\mathbf{p} : [0, a] \rightarrow C$, the *path* in physical space, and by $\mathbf{\Gamma} : [0, a] \rightarrow CP$, the *trajectory* in phase space, corresponding to this acceleration.

Then, a solution to the problem $(O, a_{max}, v_{max}, \mathbf{s}, \mathbf{f})$ is a command acceleration $\mathbf{a} : [0, a] \rightarrow \Re^2$ such that (1) and (2) are satisfied and $\mathbf{\Gamma}([0, a]) \subseteq FP$, $\mathbf{\Gamma}(0) = \mathbf{S}$, $\mathbf{\Gamma}(a) = \mathbf{F}$. The *time* of solution is a. And a *time-optimal solution* to the given problem is a solution such that the time is minimized.

3 Previous results

The one-dimensional case is studied in [O]. The problem solved there is planning the motion of a particle moving on the real line such that given two "pursuit" functions $f(t)$ and $g(t)$, the position function, say $x(t)$, satisfies $f(t) \leq x(t) \leq g(t)$ for all t. In addition, the motion must obey a bound on the acceleration. A polynomial time exact algorithm is provided for this problem. In the three-dimensional case, finding an exact solution is known to be NP-hard [CR]. Not much is known about lower bounds for finding an exact solution in the 2D case.

There is an extensive literature on time-optimal trajectory planning subject to different constraints. One focus has been to determine the time-optimal control for a manipulator moving along a given path [BDG, SM]. Other work attempts to characterize the time-optimal solutions analytically [Hol, Sch]. Several approximation algorithms have been proposed for time-optimal motion planning. Sahar and Hollerbach [SH] and Shiller and Dubowsky [SD] provide approximation algorithms for robots with several degrees of freedom and full dynamics. These approaches use grid methods to compute approximate solutions but they do not bound the goodness of the approximation. Further, these algorithms run in time exponential in the number of grid points. The first polynomial-time approximation algorithm for the two and three dimensional cases was provided in [CDRX]. More specifically, they provide an algorithm which finds a "safe" trajectory which takes time $(1 + \epsilon)$ times the time for an optimal trajectory. The trajectory is safe in the sense that there exists a "tube" of a certain size (the radius being a function of the particle velocity) around the trajectory which does not pass through through obstacle space. The algorithm runs in time which is polynomial in ϵ^{-1} and in n, the complexity of the input. Donald and Xavier [DX] provide an algorithm for this problem with an improved running time. Jacobs et al. [JHCP] give the first algorithm for generating near time-optimal trajectories for an open-kinematic-chain manipulator with guaranteed bounds on the closeness of the approximation. The time and space required is polynomial in the desired accuracy of approximation. The problem of finding time-optimal motions for a mobile robot subject to nonholonomic constraints is considered in [JRL].

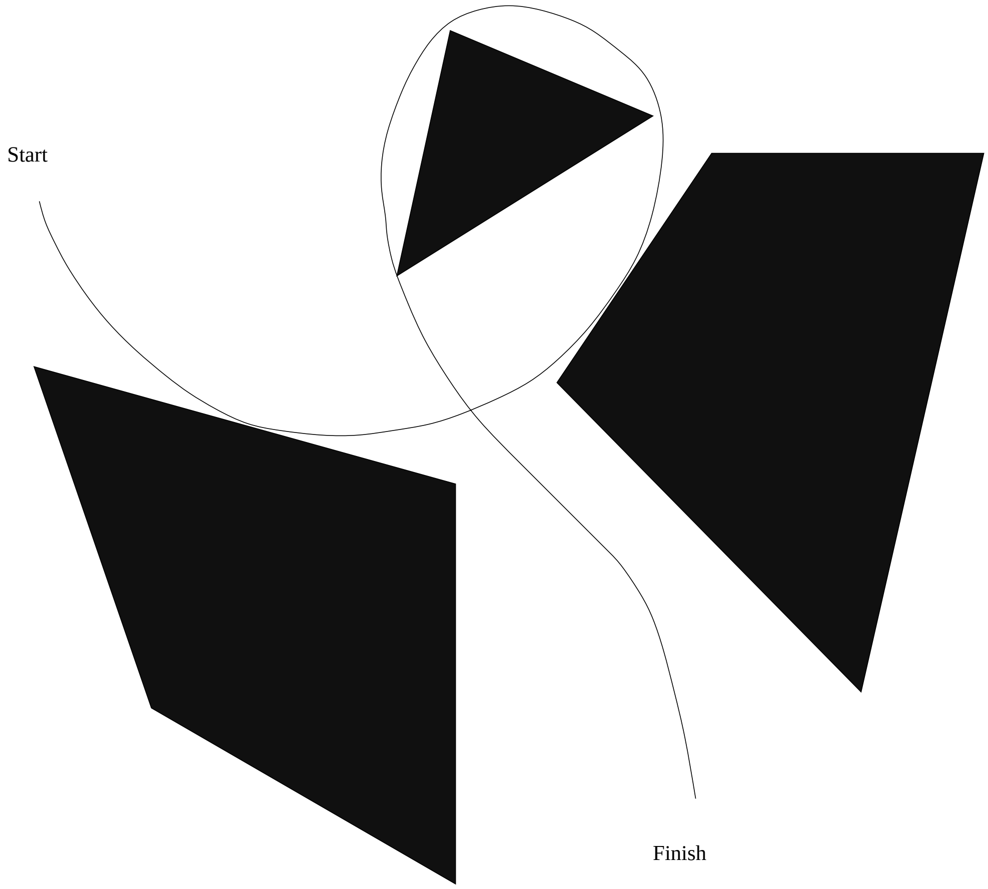


Figure 1: A general trajectory

There are several other interesting variations on the standard problem stated in the previous section, some of which have been considered in the literature. One such problem is to give an algorithm for determining the shortest bounded curvature path in the presence of obstacles (without any dynamic constraints). Fortune and Wilfong [FW] address this problem in the two-dimensional case. They provide an exact algorithm which solves the reachability problem, i.e. it determines if there exists a path between two points which obeys the given constraints but it does not generate the path. Their algorithm runs in time and space exponential in the complexity of the polygonal arrangement. An approximation algorithm for generating a bounded-curvature path is provided in [J]. Other work related to trajectory planning subject to non-holonomic constraints includes [N, LCH].

4 Statement of Results and Overview of Approach

In this paper, we give the first algorithm to generate an *exact* time-optimal solution to the two dimensional problem $(O, a_{max}, v_{max}, \mathbf{S}, \mathbf{F})$ where $\mathbf{S} = (\mathbf{s}, \dot{\mathbf{s}}) = (\mathbf{s}, \mathbf{0})$ and $\mathbf{F} = (\mathbf{f}, \dot{\mathbf{f}}) = (\mathbf{f}, \mathbf{0})$ (i.e. the object starts and ends at rest). The velocity and acceleration are assumed to be bounded in the L_∞ norm. We also provide partial results for arbitrary norms, in particular the L_2 norm. The algorithm requires space which is polynomial in the input (i.e. PSPACE) and runs in exponential time.

Let us first attempt to get an intuitive feel for the problem. In general, the time-optimal trajectory from one point to another will have the shape shown in Fig. 1. As shown there, the trajectory will be made up of series of obstacle contacts connected by segments which do not touch the obstacles. These segments (including their start and end points, which touch the obstacles) are necessarily optimal trajectories between the states (position and velocity) at the start and end points of the segment. Let us suppose that given two such states in phase space, we can describe the exact form of the optimal segment between the two. Let us further suppose that we know the number of contact points in the time-optimal trajectory (but not the values of the positions and velocities at the contact points). We can now imagine using some kind of search technique using the positions and velocities at the contact points as variables, which we can instantiate with values, to find out the time-optimal trajectory.

The approach described rests strongly on three suppositions : The first is that we can describe the optimal path between two states in the absence of obstacles. We will show that this is indeed possible and that the path can be described in the form of boolean expressions of polynomial inequalities.

Second and most important is the assumption that we know an upper bound on the number of contact points in the time-optimal trajectory. It is entirely possible, *a priori*, for a time-optimal trajectory to contain an arbitrary number of loops in the path it generates. In that case, our approach would fail since even if we found the optimal trajectory with a certain number of contacts, it would be possible that adding more contacts would give us trajectories which take less time. We prove that this is not possible; a time-optimal trajectory which visits the same vertex or edge of the configuration more than once can be replaced by one which visits the particular vertex or edge only once. It then follows that a time-optimal trajectory comprises a (known) bounded number of segments (or less) and we can then search over all trajectories which contain that number of segments (or less).

The final assumption is that we can do the search over phase space for the optimal trajectory. We will show that this is possible using algorithms for the theory of the reals. In order to use the theory of the reals, however, it is important that we can describe the segments using boolean expressions involving polynomial inequalities. As mentioned above, we will show that the segments can indeed be described as such.

In light of the above discussion, we can summarize our approach as follows :

- We first define and characterize, in the case of the L_∞ norm, a certain class of solutions which we call the *canonical* solutions, which satisfy certain homotopy properties. The canonical solutions are made up of segments which can be described using polynomial inequalities.

- Then, we prove a tracking lemma which provides bounds on the time difference between a given trajectory and (a slower) one which tracks it. This lemma is applicable to arbitrary norms and is of independent interest.

- We then prove the Loop Elimination Theorem which shows that any solution which generates a self-intersecting path can be replaced by one which does not give rise to a self-intersecting path. This theorem too applies to arbitrary norms.

- Next, we show that there exists a canonical solution which gives rise to a non-self-intersecting path and which is time-optimal.

- Finally, we show how to use the theory of the reals (with a bounded number of quantifier alternations) to obtain the canonical solution and hence the associated trajectory.

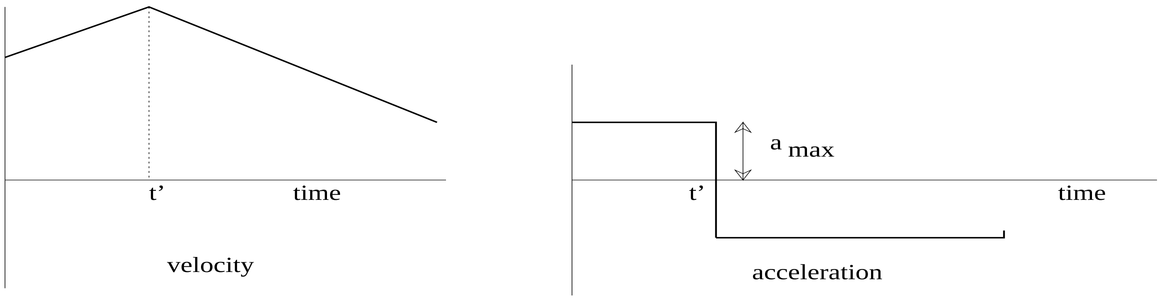


Figure 2: Time-optimal velocity and acceleration in 1-d

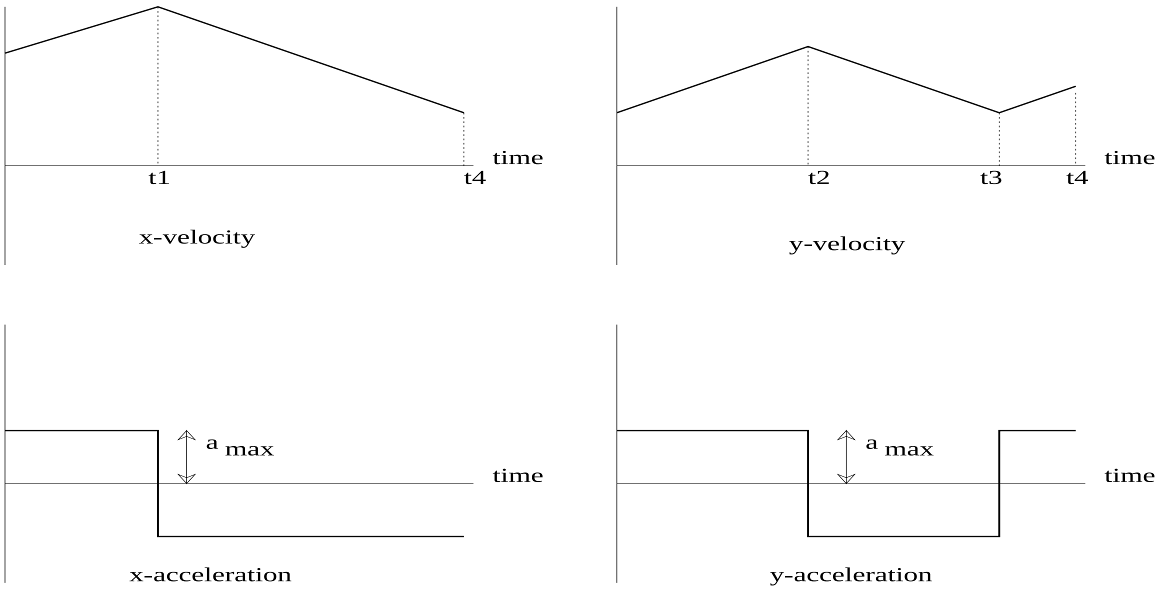


Figure 3: Canonical motion between two points

5 The canonical solutions

Let us first consider the one-dimensional problem : If we do not have a bound on the velocity, it can be easily seen that the fastest way to get from a point s with velocity $\dot{s}$ to another point f with velocity $\dot{f}$ is to apply a "bang-bang" command acceleration as shown in Fig. 2. Consider now the motion of the given object in two dimensional physical space without the presence of obstacles, where the acceleration is bounded under the L_∞ norm. Suppose the particle is to be moved from a point $\mathbf{S} = (\mathbf{s}, \dot{\mathbf{s}})$ to $\mathbf{F} = (\mathbf{f}, \dot{\mathbf{f}})$. Since the velocity and acceleration are bounded in the L_∞ norm, the bounds in the x and y directions are decoupled. We could then treat the two-dimensional problem as two one-dimensional problems and apply bang-bang accelerations in both directions. However, in general, the time-optimal motion in one of the directions will take more time than the other. We will refer to that direction as the *saturated direction*. Assume that the saturated direction is the x direction. Then, as shown in Fig. 3, we can apply a bang-bang acceleration in the x direction. Since the time-optimal motion in the y direction takes time at most equal to that in the x direction, it follows that we can arrange the velocity profile in the y direction in a triple-bang fashion as shown in Fig. 3 such that the area under the velocity curve is exactly the distance to be traversed in the y direction. Since there is only one position of the middle segment of the velocity profile in the y direction which corresponds to the right y distance, the motion thus obtained is uniquely specified. Let the velocity profiles in the x and y directions have the general shape shown in Fig. 3. Let $\mathbf{s} = (s_x, s_y)$, $\dot{\mathbf{s}} = (\dot{s}_x, \dot{s}_y)$, $\mathbf{f} = (f_x, f_y)$ and $\dot{\mathbf{f}} = (\dot{f}_x, \dot{f}_y)$. For time $t \leq t_1$, the equations for the path described in the x and y directions are given by :

$$x = s_x + \dot{s}_x t + \frac{1}{2} a_{max} t^2 \tag{3}$$

and

$$y = s_y + \dot{s}_y t + \frac{1}{2} a_{max} t^2 \tag{4}$$

Similarly, for $t_1 \leq t \leq t_2$, we have,

$$x = s_x + \dot{s}_x t + \frac{1}{2} a_{max} t^2 - a_{max}(t - t_1)^2 \tag{5}$$

and y is given by (4). For $t_2 \leq t \leq t_3$, x is given by (5) and

$$y = s_y + \dot{s}_y t + \frac{1}{2} a_{max} t^2 - a_{max}(t - t_2)^2$$

Finally for $t \geq t_3$, x is given by (5) and

$$y = s_y + 2\dot{s}_y t - \dot{s}_y t_3 - \frac{1}{2} a_{max} t^2 + a_{max}((t - t_3)^2 - (t - t_2)^2) \tag{6}$$

If we let $s_f - s_x = \Delta x \geq 0$, we get

$$t_1 = \frac{\dot{s}_x}{a_{max}} + \frac{1}{a_{max}} \sqrt{\frac{\dot{s}_x^2 + \dot{f}_x^2}{2} + a_{max} \Delta x} \tag{7}$$

We can obtain similar expressions for t_2 and t_3. Thus, given the initial and final positions and velocities of a canonical segment we can write parametrized expressions (with the positions and velocities as indeterminates) describing the parabolic pieces which comprise the path. The parameter we use is time t. In section 8 we will show how we can use such expressions to write predicates which enable us to generate the optimal trajectory. The above analysis holds for the type of velocity profiles shown in Fig. 3. We can get different types of velocity profiles depending on the relative values of the velocities as shown in Fig. 4. For each case we can obtain the extremal time instants as above thereby giving us similar expressions which describe the corresponding path.

The previous discussion assumed that there was no bound on the velocity of the particle. If we have a bound on the velocity then the motion in the saturated direction would, in general, appear as shown in Fig. 5(a). Similarly the motion in the unsaturated direction would have the general form shown in Fig. 5(b). We can then determine the extremal time instances in the same manner as before and obtain expressions for the path. In the rest of the paper we will assume no bound on the velocity; the arguments given can be easily modified to deal with the case of bounded velocity.

Call a point, such as t' in Fig. 2, where the acceleration changes direction a *switch*. Now consider the motion of the object through physical space with obstacles. As the object moves from $\mathbf{S}$ to $\mathbf{F}$, the path might touch vertices or edges of the obstacle space. We will call such points on the path (or on the trajectory) *contact points* and will refer to the portion of a path (or trajectory) between two consecutive contact points as a *segment* of the path (or trajectory). For the sake of completeness we will also include the start and

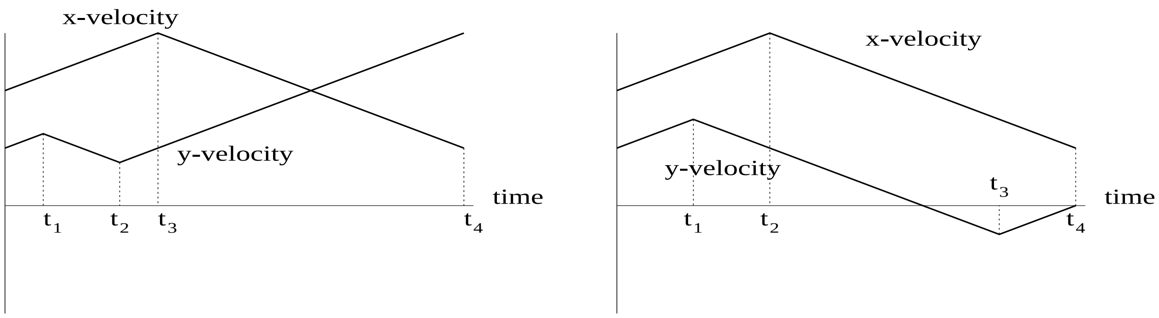


Figure 4: Other types of velocity profiles

finish points of the trajectory as contact points. If the two consecutive contact points are $\mathbf{z}$ and $\mathbf{z}'$, we will denote the segment between them by $\overline{\mathbf{z},\mathbf{z}'}$. Consider a segment of a path where the command acceleration in both the x and y directions is bang-bang and the motion is saturated in at least one direction. If the velocity profile is such that it is saturated in at least one direction and is arranged in the unsaturated direction as shown in Fig. 3, then we will call the segment a *canonical segment.* Finally, we have,

Definition 1 *Let* $\mathbf{a} : [0, b] \rightarrow \Re^2$ *be a solution to (i.e. the command acceleration for) a given problem* $(O, a_{max}, v_{max}, \mathbf{S}, \mathbf{F})$. *We say that this solution is* canonical *if it is made up of a finite number of canonical segments.*

Thus, equivalently a solution is canonical if it is made up of a finite number of constant saturated acceleration steps.

5.1 The homotopy property of the canonical solutions

We now prove an important property of the canonical solutions which we will need in Section 7 to prove the existence of time-optimal canonical solutions. Assume that the physical space has no obstacles. As shown above, the canonical solution between two points in CP is then uniquely defined. If $\mathbf{Q}$ and $\mathbf{R}$ are two points in CP then denote the canonical solution between Q and R by $\mathbf{a}_{\mathrm{QR}}$ and the corresponding canonical trajectory and path by $\mathbf{\Gamma}_{\mathrm{QR}}$ and $\mathbf{p}_{\mathrm{QR}}$ respectively.

Lemma 1 *Let* $\mathbf{Q}$, $\mathbf{R_1}$ *and* $\mathbf{R_2}$ *be three points in* CP. *Then* $\mathbf{p}_{\mathrm{QR}_1}$ *is homotopic to* $\mathbf{p}_{\mathrm{QR}_2}$ *and* $\mathbf{\Gamma}_{\mathrm{QR}_1}$ *is homotopic to* $\mathbf{\Gamma}_{\mathrm{QR}_2}$.

Thus, there exists a continuous deformation of one path to another (and similarly of one trajectory to another). Or, in other words, the position and velocity are continuous functions of the end-point position and velocity.

Proof: We will show that if the end-point position and velocity are varied continuously, at each step we will obtain a continuous deformation of the path we started with. Let $\mathbf{R} \in CP$ denote the end-point at any intermediate step of the deformation, i.e., $\mathbf{R}$ changes continuously from $\mathbf{R_1}$ to $\mathbf{R_2}$. There are essentially two cases : in the first, the saturated direction remains unchanged as $\mathbf{R}$ is changed. The assertion then follows from the fact that the switch points t_1, t_2, t_3 and t_4 as obtained above are continuous functions of the end-point position and velocity and hence change continuously as the $\mathbf{R}$ is changed continuously. Hence the x and y displacements and the velocities in the two directions at any instant change continuously with $\mathbf{R}$. This means the path and trajectory obtained at each instant change continuously. On the other hand, if the saturated direction changes, it is possible that the path doesn't change continuously but rather jumps

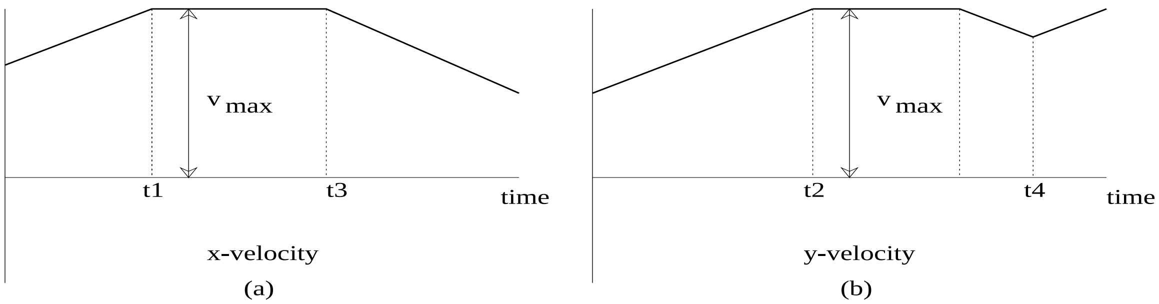


Figure 5: Canonical motion with velocity bound

around. However, the saturated direction changes if and only if two of the switch points in the unsaturated direction merge. But this can only happen in a continuous fashion since the switch points are continuous functions of the end-point position and velocity. ■

The importance of this lemma lies in the fact that it enables us to continuously deform a given trajectory into a canonical one as will be seen in section 7.

6 The loop elimination theorem

In this section, we show that any solution path which contains a loop, or more generally, intersects itself, can be replaced by one which does not intersect itself and which takes time which is less than or equal to that taken by the original trajectory. This enables us to restrict attention to loop-free paths in our search for an optimal trajectory using the theory of the reals.

6.1 A tracking lemma

In order to prove the loop elimination theorem, we prove a tracking lemma. The essence of this lemma is as follows : given any trajectory in free space, we show that there exists another trajectory starting at rest, which follows the same path in physical space, such that the time difference between the two trajectories at any given point on the path is bounded from above. This is done by establishing an invariant relating the two trajectories which holds at all points along the path. We prove this lemma for *any norm* and in particular for the L_∞ and L_2 norms. Though we use this lemma to prove the loop elimination theorem, it is of independent interest since it provides a simple and elegant characterization of the relationship between the original trajectory and the tracking trajectory.

Notation : Let $\mathbf{a} : [0, b] \rightarrow \Re^2$ be a solution to some given problem. Denote by $\tilde{\mathbf{v}}(s)$ the arc-parametrized velocity of the trajectory corresponding to this solution, where s denotes arc distance traversed along the path. Let $\tilde{\mathbf{a}}(s)$ denote the arc-parametrized acceleration and $\tilde{\mathbf{p}}(s)$ denote the arc-parametrized path. Let $t(s)$ denote the time taken by the solution to traverse a distance of s along the path. Let $|| \cdot ||$ denote any norm under which the bound on the acceleration is given.

We observe that any norm gives rise to an equinormal contour which has the following properties : The contour is convex and symmetrical about the origin.

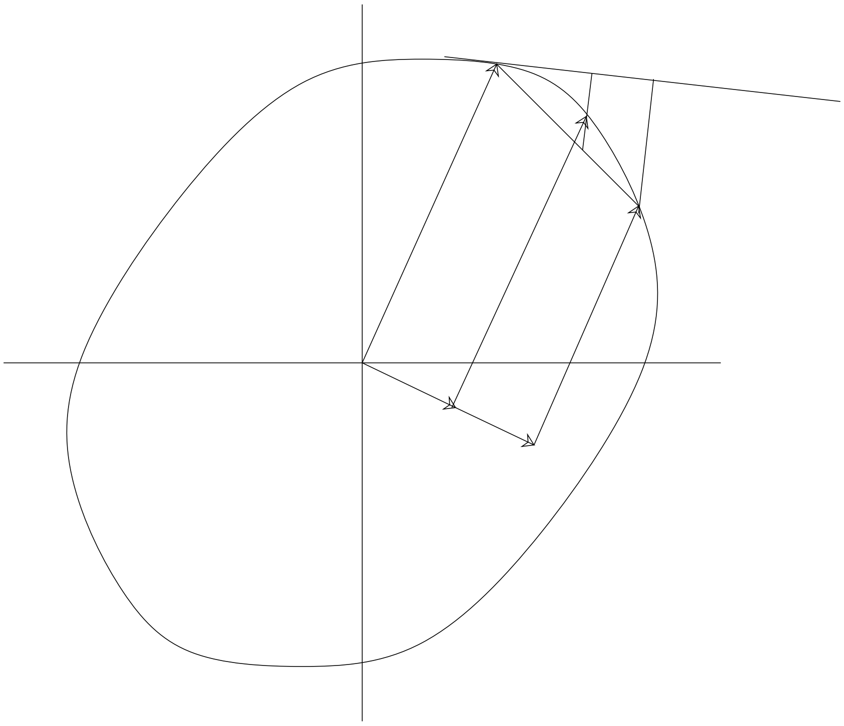


Figure 6:

Lemma 2 *(Tracking Lemma) Consider a command acceleration* $\mathbf{a_1} : [0, b_1] \rightarrow \Re^2$ *Then there exists a command acceleration* $\mathbf{a_2}$ *such that*

1. $\mathbf{v_2}(0) = 0$ *(the second trajectory starts at rest)*
2. $\vartheta(s)$ *is non-increasing, where* $\vartheta(s) = \| \Delta\tilde{\mathbf{v}}(s) \| + \Delta t(s) a_{max}$, $\Delta\tilde{\mathbf{v}}(s) = \tilde{\mathbf{v}}_\mathbf{1}(s) - \tilde{\mathbf{v}}_\mathbf{2}(s)$ *and* $\Delta t(s) = t_2(s) - t_1(s)$.
3. $\tilde{\mathbf{p}}_\mathbf{1}(s) = \tilde{\mathbf{p}}_\mathbf{2}(s)$ *(i.e. the paths described by the two trajectories are the same).*

Proof: We want to find a command acceleration which will satisfy conditions 1 to 3 in the statement of the lemma. Our strategy will be simple : we will follow the original path and at every point on the path we will apply the maximum possible acceleration in the tangential direction till we (possibly) encounter a point where the velocity in the second case equals that in the first case. From this point on, we will apply the same acceleration as that in the original trajectory.

Consider Fig. 6 representing the accelerations and velocities at some point s on the two paths. The bounding closed curve represents the equinormal contour corresponding to a_{max} of the norm under consideration. We have scaled the velocity $\tilde{\mathbf{v}}_\mathbf{1}$ so that it has the same norm value as a_{max}. Both motions give rise to a centripetal acceleration of magnitude $\| \tilde{\mathbf{v}}_\mathbf{1}(s) \|_2^2 / \rho(s)$ in the original trajectory and $\| \tilde{\mathbf{v}}_\mathbf{2}(s) \|_2^2 / \rho(s)$ in the second case. Here $\rho(s)$ is the radius of curvature at s. Since the tracking trajectory is constrained to follow the original one at each point along its path, the centripetal acceleration for the tracking trajectory has to be in the same direction as that of the original trajectory. Both accelerations are perpendicular to the velocity direction and point toward the instantaneous center of rotation. We show these accelerations $\tilde{\mathbf{a}}_{\mathbf{c_1}}$ and $\tilde{\mathbf{a}}_{\mathbf{c_2}}$ in Fig. 6. At every point, the velocity in the second case is at most that in the first case. Thus the centripetal acceleration in the tracking trajectory is at most that in the original trajectory. The centripetal acceleration for the tracking trajectory is constrained in direction and magnitude as described above; however once the centripetal acceleration is fixed we are free to choose the magnitude of the tangential acceleration subject to the norm constraints. We will select the tangential acceleration $\tilde{\mathbf{a}}_{\mathbf{t_2}}(s)$ applied to

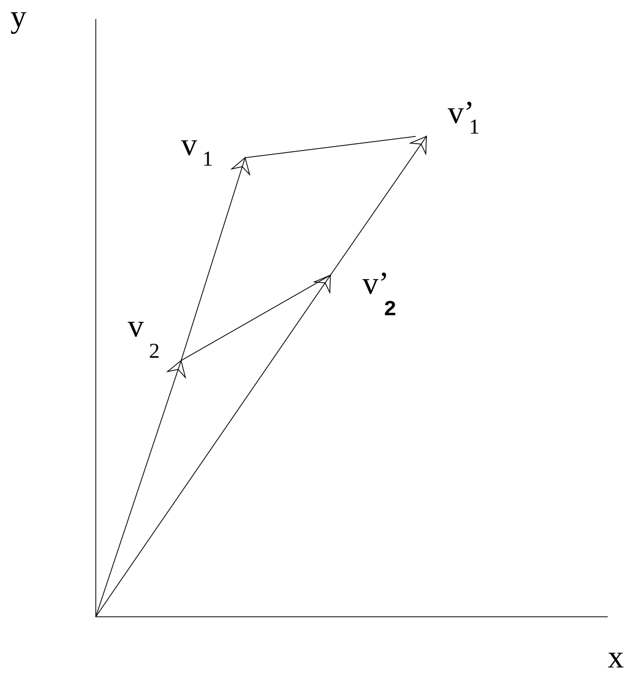


Figure 7:

be the maximum possible in the tangential direction as shown in Fig. 6. The acceleration $\tilde{\mathbf{a}}_{\mathbf{2}}(s)$ is then obtained as a vector sum of the normal (centripetal) and tangential accelerations. Since we are following the original path at each point and since we start with a zero velocity, conditions 1 and 3 in the lemma are satisfied.

We will now prove that condition 2 is also satisfied. We will show that for two points with arc distance s and $s + \delta s$, where δs is sufficiently small, the difference $\delta\vartheta(s) = \vartheta(s + \delta s) - \vartheta(s)$ is non-positive. We have

$$\delta\vartheta(s) = \delta \parallel \Delta\tilde{\mathbf{v}}(s) \parallel + a_{max}\delta\Delta t(s)$$

The general relationship between $\tilde{\mathbf{v}}_{\mathbf{1}}(s)$, $\tilde{\mathbf{v}}_{\mathbf{2}}(s)$, $\tilde{\mathbf{v}}'_{\mathbf{1}} = \tilde{\mathbf{v}}_{\mathbf{1}}(s + \delta s)$ and $\tilde{\mathbf{v}}'_{\mathbf{2}} = \tilde{\mathbf{v}}_{\mathbf{2}}(s + \delta s)$ is as shown in Fig. 7. Since $\tilde{\mathbf{v}}'_{\mathbf{1}}$ and $\tilde{\mathbf{v}}'_{\mathbf{2}}$ are in the same direction,

$$\parallel \Delta\tilde{\mathbf{v}}(s + \delta s) \parallel = \parallel \tilde{\mathbf{v}}'_{\mathbf{1}} \parallel - \parallel \tilde{\mathbf{v}}'_{\mathbf{2}} \parallel$$

Similarly,

$$\parallel \Delta\tilde{\mathbf{v}}(s) \parallel = \parallel \tilde{\mathbf{v}}_{\mathbf{1}} \parallel - \parallel \tilde{\mathbf{v}}_{\mathbf{2}} \parallel$$

And,

$$\begin{aligned} \delta\Delta t(s) &= t_2(s + \delta s) - t_1(s + \delta s) - t_2(s) - t_1(s) \\ &= (t_2(s + \delta s) - t_2(s)) - (t_1(s + \delta s) - t_1(s)) \end{aligned}$$

or,

$$\delta\Delta t(s) = \delta t_2(s) - \delta t_1(s)$$

Thus,

$$\delta\vartheta(s) = (\parallel \tilde{\mathbf{v}}'_{\mathbf{1}} \parallel - \parallel \tilde{\mathbf{v}}'_{\mathbf{2}} \parallel) - (\parallel \tilde{\mathbf{v}}_{\mathbf{1}} \parallel - \parallel \tilde{\mathbf{v}}_{\mathbf{2}} \parallel) + \ a_{max}(\delta t_2(s) - \delta t_1(s))$$

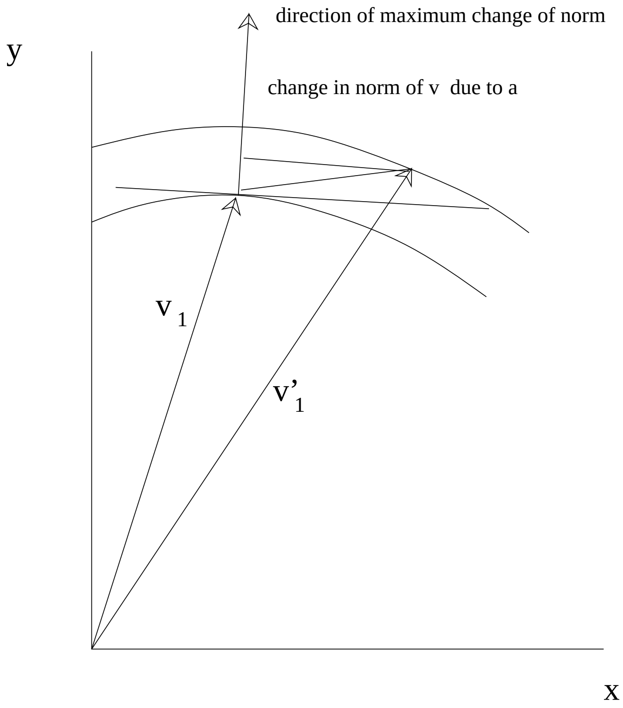


Figure 8:

or,

$$\delta\vartheta(s) = (\| \tilde{\mathbf{v}}'_{\mathbf{1}} \| - \| \tilde{\mathbf{v}}_{\mathbf{1}} \|) - (\| \tilde{\mathbf{v}}'_{\mathbf{2}} \| - \| \tilde{\mathbf{v}}_{\mathbf{2}} \|) + \ a_{max}(\delta t_2(s) - \delta t_1(s)) \tag{8}$$

We can assume, *wlog*, the general picture shown in Fig. 7. As δs approaches 0, the difference $(\| \tilde{\mathbf{v}}'_{\mathbf{1}} \| - \| \tilde{\mathbf{v}}_{\mathbf{1}} \|)$ in the norms of $\tilde{\mathbf{v}}_{\mathbf{1}}$ and $\tilde{\mathbf{v}}'_{\mathbf{1}}$ will be measured in the direction perpendicular to the norm contour at $\tilde{\mathbf{v}}_{\mathbf{1}}$. As shown in Fig. 8, the maximum change in $\| \tilde{\mathbf{v}}_{\mathbf{1}} \|$ will occur in this direction. Since, in general, $\tilde{\mathbf{a}}_{\mathbf{1}}$ will be applied in another direction, we will not achieve the maximum change. The discrepancy between the two can be measured by taking the difference between the maximum change in velocity and the projection of the given change in velocity, $\tilde{\mathbf{a}}_{\mathbf{1}}\delta t_1$ in the direction of the maximum change, as shown in Fig. 8. As $\delta s \rightarrow 0$, this measures the actual discrepancy. Similarly, we can get the discrepancy in the $\tilde{\mathbf{v}}_{\mathbf{2}}$ case. Since the direction of $\tilde{\mathbf{a}}_{\mathbf{1}}\delta t_1$ and $\tilde{\mathbf{a}}_{\mathbf{1}}$ is the same (as is that of $\tilde{\mathbf{a}}_{\mathbf{2}}\delta t_2$ and $\tilde{\mathbf{a}}_{\mathbf{2}}$), we can translate the discrepancies in velocities to those in accelerations as shown in Fig. 6. As can be seen from Fig. 6, the discrepancies, given by ϵ_1 and ϵ_2, are proportional to the centripetal accelerations. The actual discrepancy for $\tilde{\mathbf{a}}_{\mathbf{2}}$ has value at most ϵ_2 shown in Fig. 6 since due to the convexity of the norm, the point p lies above the point q. Therefore, we have

$$\frac{\epsilon_1}{\epsilon_2} = \frac{{v_1}^2}{{v_2}^2} \tag{9}$$

where v_1 and v_2 denote the magnitude of the corresponding velocities in the L_2 norm. Now, in (8) we have $(\| \tilde{\mathbf{v}}'_{\mathbf{1}} \| - \| \tilde{\mathbf{v}}_{\mathbf{1}} \|) = (a_{max} - \epsilon_1)\delta t_1$ and $(\| \tilde{\mathbf{v}}'_{\mathbf{2}} \| - \| \tilde{\mathbf{v}}_{\mathbf{2}} \|) = (a_{max} - \epsilon_2)\delta t_2$. Combining with (9) and rearranging we get

$$\begin{aligned} \delta\vartheta(s) &= \epsilon_2\delta t_2 - \epsilon_1\delta t_1 \\ &= \epsilon_1\left(\frac{{v_2}^2}{{v_1}^2}\right)\delta t_2 - \epsilon_1\delta t_1 \end{aligned}$$

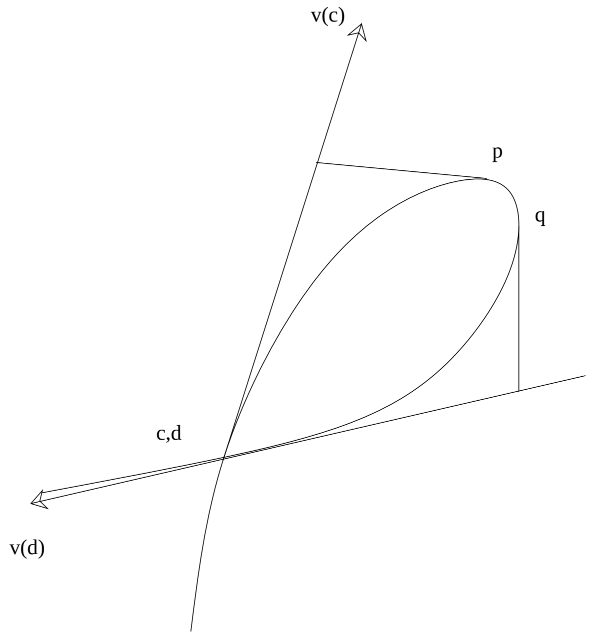


Figure 9:

$$= \quad \epsilon_1 \left(\frac{{v_2}^2 \delta t_2 - {v_1}^2 \delta t_1}{{v_1}^2} \right) \tag{10}$$

As $\delta s \rightarrow 0$, we have $v_1 = (\delta s / \delta t_1)$ and $v_2 = (\delta s / \delta t_2)$ and therefore $v_1 \delta t_1 = v_2 \delta t_2$. Therefore,

$$\delta \vartheta(s) = \epsilon_1 \delta t_1 \left(\frac{v_2 - v_1}{v_1} \right) \tag{11}$$

Since $v_2 \leq v_1$, $\delta\vartheta(s) \leq 0$ as $\delta s \rightarrow 0$. In other words $\frac{\delta\vartheta(s)}{\delta s}$ is non-positive i.e. $\vartheta(s)$ is non-increasing. ■

6.2 A lower bound on the loop time

We will now prove a lower bound on the time spent in any loop by the object as a function of the loop entry and exit velocities.

Lemma 3 *Let* $\mathbf{a} : [0, b] \rightarrow \Re^2$ *be a command acceleration such that* $\tilde{\mathbf{p}} : [0, l] \rightarrow C$*, the corresponding arc-parametrized path, has the property* $\tilde{\mathbf{p}}(c) = \tilde{\mathbf{p}}(d)$ *(i.e. the path contains a loop). Then the time taken by the corresponding trajectory is at least*

$$\frac{\| \tilde{\mathbf{v}}(c) \| + \| \tilde{\mathbf{v}}(d) \|}{a_{max}}.$$

Proof: We can assume, *wlog*, the picture in Figure 9. Here, c denotes the entry point of the loop and d the exit point. Fig. 10 shows the velocity vectors $\mathbf{v}(c) = \tilde{\mathbf{v}}(c)$ and $\mathbf{v}(d) = \tilde{\mathbf{v}}(d)$ corresponding to the two points c and d with the norm contour of the given norm. Only the norm contour corresponding to the norm value of $\mathbf{v}(c)$ is shown for the sake of clarity. As the object traverses the loop the velocity moves in a

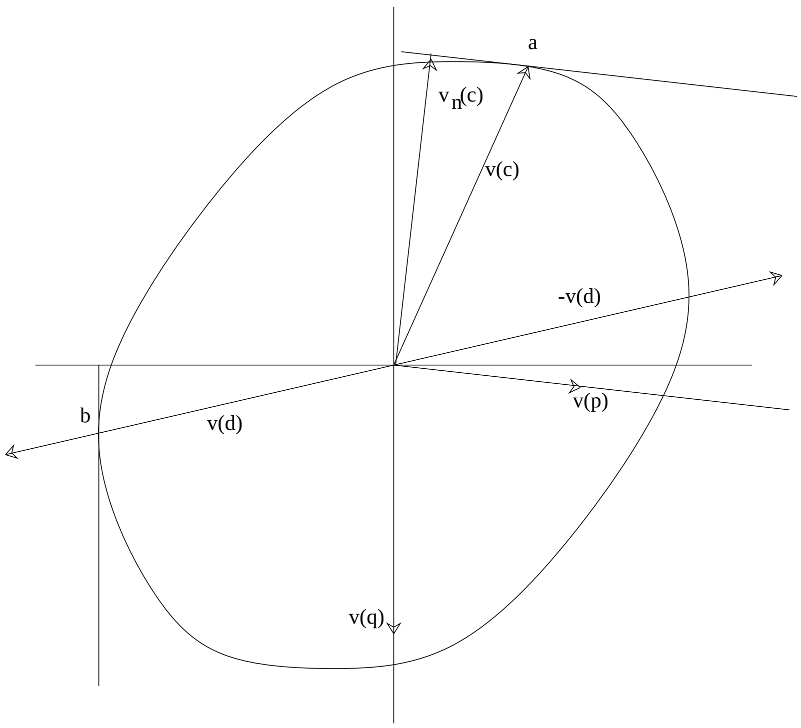


Figure 10:

clockwise manner. Our aim will be to split the motion along the loop into two non-overlapping pieces with a lower bound on the time taken by each piece. To that end, consider the tangent to the norm contour at the point a corresponding to the velocity $\mathbf{v}(c)$. Consider the projection $\mathbf{v_n}(c)$ of $\mathbf{v}(c)$ along the direction perpendicular to the tangent at a. Define point p in Fig. 9 to be given by the velocity direction in the loop parallel to the tangent at a in Fig. 10. This velocity is shown as $\mathbf{v}(p)$ in Fig. 10. When the object reaches this point in the loop the projection $\mathbf{v_n}(c)$ is exactly zero. We will show that the time required between the point c and the point p is at least $\frac{\|\tilde{\mathbf{v}}(c)\|}{a_{max}}$. At each instant, as the object traverses the loop between c and p, the maximum rate of decrease of the projection of $\mathbf{v}$ along the $\mathbf{v_n}(c)$ direction is obtained by applying maximum acceleration along the $-\mathbf{v}(c)$ direction (Denote this projection by $\mathbf{a_n}$). This can be seen as follows : Suppose we apply the acceleration in some other direction. Because of the convexity of the norm contour, the magnitude of the projection of this acceleration applied in the direction opposite to that of $\mathbf{v_n}(c)$ is less than that of the projection $\mathbf{a_n}$. This is true independent of the position in the loop the object is in. Hence at each step, to obtain the fastest decrease in the projection of $\mathbf{v}$ in $\mathbf{v_n}(c)$ direction, we should apply acceleration in the $-\mathbf{v}(c)$ direction.

From the above discussion, it follows that the time taken by the object to move from c to p in the loop has a lower bound given by the time taken to move between those two points with the acceleration held fixed in the $-\mathbf{v}(c)$ direction. For this motion we have, since the acceleration is applied in the $-\mathbf{v}(c)$ direction at all times, the time taken to achieve $\mathbf{v_n} = 0$ is given by

$$\frac{\| \mathbf{v}(c) \|}{a_{max}}$$

Similarly, we can define a point q in the loop such that the (reverse) motion from d to q takes time at least $\frac{\|\mathbf{v}(d)\|}{a_{max}}$. Here the point q is defined (Fig. 9), as point p was, by the velocity in the direction parallel to the tangent to the contour at b corresponding to the velocity $\mathbf{v}(d)$ (Fig. 10). However, it is possible *a priori* for the intervals corresponding to the motions from c to p and from q to d to overlap. This would mean that q appeared earlier in the loop than p. This is not possible due to the following three observations : The norm contour is convex; it is symmetric and lastly the loop involves a rotation of at least 180°. From symmetry the tangent direction to the norm contour at the point b corresponding to the velocity vector $-\mathbf{v}(d)$ as shown in Fig. 10 is the same as that corresponding to the vector $\mathbf{v}(d)$. Since the loop involves a rotation of at least 180°, the vector $-\mathbf{v}(d)$ lies within an angle $< 180°$ in the clockwise direction from the vector $\mathbf{v}(c)$. Finally, from convexity of the norm contour and the previous observations it follows that the tangent direction corresponding to $-\mathbf{v}(d)$ makes an angle greater than that corresponding to $\mathbf{v}(c)$. Since the points p and q are defined by these tangent directions (respectively), it follows that q cannot appear earlier than p in the loop. In the limiting case, the object leaves the loop in a direction exactly opposite to the entry direction. In this case the two tangent directions coincide as do the points p and q. We can therefore conclude that the two intervals are disjoint and the lemma follows.

■

6.3 Proof of the loop elimination theorem

We are now ready to prove the loop elimination theorem in the case of arbitrary norms :

Theorem 1 *Consider a solution* $\mathbf{a_1} : [0, b_1] \rightarrow \Re^2$ *to some given problem* $(O, a_{max}, v_{max}, \mathbf{S}, \mathbf{F})$ *where* $\dot{\mathbf{s}} = \dot{\mathbf{f}} = \mathbf{0}$. *Let* $\tilde{\mathbf{p}}_\mathbf{1} : [0, l_1] \rightarrow C$ *be the arc-parametrized path corresponding to this solution. If there exist* c, $d \in [0, l_1]$, $c < d$, *such that* $\tilde{\mathbf{p}}_\mathbf{1}(c) = \tilde{\mathbf{p}}_\mathbf{1}(d)$ *(i.e. the path intersects itself) then there exists a solution* $\mathbf{a_2} : [0, b_2] \rightarrow \Re^2$ *with corresponding path* $\tilde{\mathbf{p}}_\mathbf{2} : [0, l_2] \rightarrow C$ *such that*

1. $b_2 \leq b_1$.

2. *there do not exist points* c' *and* d' *such that* $c' \neq d'$ *and* $\tilde{\mathbf{p}}_\mathbf{2}(c') = \tilde{\mathbf{p}}_\mathbf{2}(d')$, *(i.e. the path does not intersect itself).*

Proof: The idea of the proof is as follows : We will construct the new trajectory by moving along the path of the original trajectory till we come to rest at the loop intersection point. At this point, we will ignore the loop and continue along the rest of the path. We will show that coming to a stop at the intersection does not penalize us and that the new trajectory takes time which is less than or equal to that taken by the original one.

Consider the portion of the (arc-parametrized) trajectory from d to l_1. By Lemma 1 above, there exists a solution $\tilde{\mathbf{a}}_\mathbf{2}$ defined on $[d, l_1]$ such that $\tilde{\mathbf{p}}_\mathbf{2}(s) = \tilde{\mathbf{p}}_\mathbf{1}(s)$ and $\vartheta(s)$ is non-increasing in this interval and $\tilde{\mathbf{v}}_\mathbf{2}(d) = 0$. Similarly, we can define $\tilde{\mathbf{a}}_\mathbf{2}$ on the interval [0,c] by applying the above lemma, in the reverse direction, from c to 0. Now, $\tilde{\mathbf{v}}_\mathbf{2}(c) = 0$ and $\tilde{\mathbf{v}}_\mathbf{2}(d) = 0$. And c and d correspond to the same point in C. So, we can now define the new acceleration $\tilde{\mathbf{a}}_\mathbf{2}$ as stated with $c = d$. If there are any more loops or self-intersections, we can repeat the construction to eliminate them. (We note that this includes trivial loops of the kind where the particle just moves back and forth across the same portion of the path.) Since both pieces constitute a solution to and follow the path of the original trajectory, the combined path is a legal path and the combined solution is a solution to the original problem. We now show that for each loop elimination, we get a solution which takes time less than or equal to that of the trajectory in the previous iteration.

Consider the portions of the original and the new trajectories from d to l_1. (By abuse of notation, we will use the same arc-parametrization for this piece of both trajectories.) Treat these two trajectories as complete trajectories in themselves from d to l_1. Thus, e.g. $\Delta t(s)$ measures the time difference starting from d. But $\vartheta(d) = \| \Delta\tilde{\mathbf{v}}(d) \| + \Delta t(d) a_{max}$. Putting $\Delta t(d) = 0$ and $\tilde{\mathbf{v}}_\mathbf{2}(d) = 0$, we get $\vartheta(d) = \| \tilde{\mathbf{v}}_\mathbf{1}(d) \|$. Similarly, we get $\vartheta(l_1) = \Delta t(l_1) a_{max}$. In like manner, consider the portions of the original and new trajectories from c to 0 in reverse arc-parametrization. We get $\vartheta(c) = \| \tilde{\mathbf{v}}_{\mathbf{1}}(c) \|$. and $\vartheta(0) = \Delta t(0) a_{max}$ where we abuse our notation and denote the "time difference" between these two segments at 0 by $\Delta t(0)$. From Lemma 1 we have $\vartheta(d) \geq \vartheta(l_1)$ and $\vartheta(c) \geq \vartheta(0)$. Thus,

$$\| \tilde{\mathbf{v}}_{\mathbf{1}}(c) \| + \| \tilde{\mathbf{v}}_{\mathbf{1}}(d) \| \geq (\Delta t(0) + \Delta t(l_1)) a_{max} \tag{12}$$

From Lemma 3, we have the time taken in the loop given by

$$t_1(d) - t(c) \geq \frac{\| \tilde{\mathbf{v}}_{\mathbf{1}}(c) \| + \| \tilde{\mathbf{v}}_{\mathbf{1}}(d) \|}{a_{max}}$$

Combining this with (12) we get,

$$t_1(d) - t(c) \geq \Delta t(0) + \Delta t(l_1)$$

Thus, the time taken in the loop is at least as much as the time difference between the two trajectories. Ergo, we can eliminate the loop to get a new trajectory, without the loop, which takes time less than or equal to that taken by original trajectory. Repeating this procedure, we obtain a faster trajectory which has a path which does not intersect itself.

■

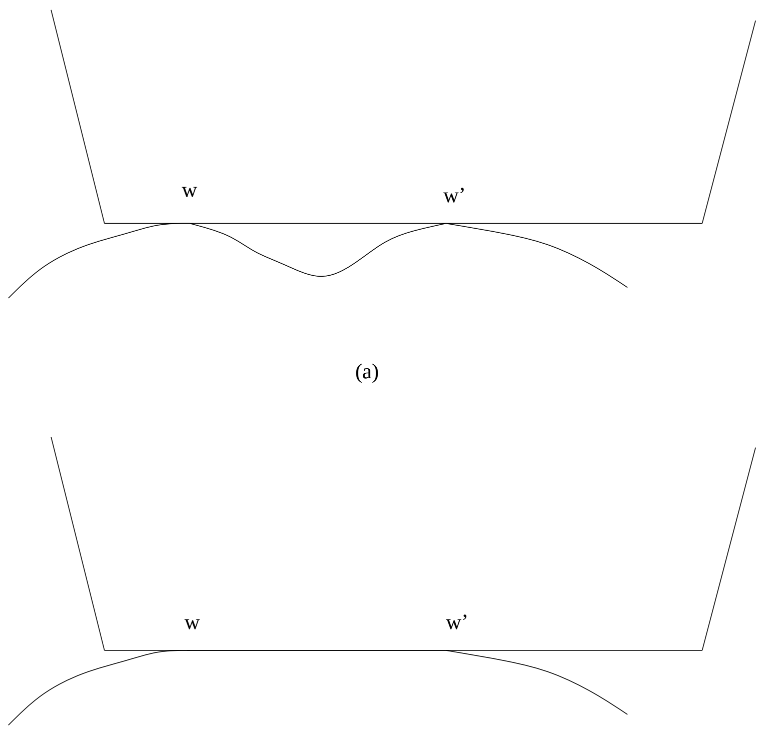
7 Optimality of the canonical trajectory

In this section, we show that for any problem there exists a canonical solution, which is time-optimal.

Theorem 2 *Let $\mathbf{a_1} : [0, b_1] \rightarrow \Re^2$ be an optimal solution to some problem $(O, a_{max}, v_{max}, \mathbf{S}, \mathbf{F})$. Then there exists an optimal solution $\mathbf{a_2}$ to this problem which is canonical and which has $O(n)$ segments.*

Proof: We will show the existence of an optimal canonical trajectory by deforming the path corresponding to the given acceleration homotopically into one which is canonical. Starting at $\mathbf{s}$, we move along the path $\mathbf{p_1}$ till we reach a point $\mathbf{z}$ which has the following property : assume that the trajectory from $(\mathbf{s}, \dot{\mathbf{s}})$ to $(\mathbf{z}, \dot{\mathbf{z}}) = \mathbf{Z}$ is canonical. Thus we get a new trajectory from $\mathbf{S}$ to $\mathbf{F}$ such that the trajectory passes through $\mathbf{S}$ and $\mathbf{Z}$ but such that the new trajectory has a canonical portion from $\mathbf{S}$ to $\mathbf{Z}$. The point $\mathbf{Z}$ is extremal in the sense that the canonical path from $\mathbf{s}$ to $\mathbf{z}$ is tangent to a vertex or to a segment of the obstacle space but is a legal path through C. In other words, we "canonize" the trajectory as far as we can till it touches some contact point. (We will refer to points in CP such as $\mathbf{Z} = (\mathbf{z}, \dot{\mathbf{z}})$, where $\mathbf{z}$ is a contact point in C, as "contact points".) Denote this new trajectory by $\mathbf{\Gamma_1}$. It is possible for the path corresponding to this trajectory to go through more than one contact point. Let $\mathbf{Z_1}$ be the furthest contact point down the trajectory between $\mathbf{S}$ and $\mathbf{Z}$. Since the portion of the trajectory between $\mathbf{S}$ and $\mathbf{Z}$ is canonical, it follows that the portion between $\mathbf{S}$ and $\mathbf{Z_1}$ is a canonical segment. We now repeat the process with the new trajectory $\mathbf{\Gamma_1}$, starting at $\mathbf{Z_1}$ to obtain another trajectory $\mathbf{\Gamma_2}$ and so on. Thus, at the k^{th} step, we have a trajectory $\mathbf{\Gamma_k}$ and contact points $\mathbf{Z_1}$ to $\mathbf{Z_k}$ on it such that the segments $\overline{\mathbf{Z_i}, \mathbf{Z_{i+1}}}$ (for $1 \leq i \leq k-1$) are canonical. Then, we canonize the trajectory from $\mathbf{Z_k}$ till we obtain an extremal point $\mathbf{Z}'$ such that the trajectory from $\mathbf{Z_k}$ to $\mathbf{Z}'$ has a contact point $\mathbf{Z_{k+1}}$ which is closest to $\mathbf{Z}'$.

We need to prove several things in order to complete the proof of the theorem. First, we need to show that for any k the deformation of the trajectory $\mathbf{\Gamma_k}$ into $\mathbf{\Gamma_{k+1}}$ is continuous so as to ensure that even if the saturated direction changes the trajectory obtained does not pass through obstacle space. I.e. we need



(a)

(b)

Figure 11:

to ensure that there exists a point $\mathbf{Z}'$ such that the canonized section between $\mathbf{Z_k}$ and $\mathbf{Z}'$ passes through a contact point $\mathbf{Z_{i+1}}$. This can be done only by ensuring that we deform the trajectory continuously. But this follows from Lemma 1 proved in Section 5.1. We showed that the position and velocity were continuous functions of the end-point position and velocity. At the k^{th} step, we move along the path continuously from $\mathbf{z_k}$. We are therefore varying the position and velocity of the end-point continuously. Accordingly, the canonical segment so obtained, starting at $\mathbf{Z_k}$, also varies continuously. Hence, we are assured that either we will reach the final position or the segment will touch a point $\mathbf{z_{k+1}}$ of obstacle space. We then repeat the process starting at $\mathbf{Z_{k+1}}$. This ensures the validity of the above process of obtaining a canonical trajectory from the given one.

Next, we show that we can obtain a trajectory which has $O(n)$ contact points or we can transform it into one which does. If the trajectory passes through the same contact point more than once then it has a loop and by the loop elimination theorem we can replace it with another trajectory which does not pass through the same point more than once. This clearly does not affect the fact that the trajectory is made of canonical segments. As shown in Fig. 7(a), it is possible for the path corresponding to this new trajectory to touch the same edge of the obstacle space at different points. Let $\mathbf{w}$ and $\mathbf{w}'$ denote the first and last points which touch the same edge as shown in the Fig. 7(a). We note that the velocity at $\mathbf{w}'$ either has the same or opposite direction as that at $\mathbf{w}$. We can therefore treat the section of the trajectory from $\mathbf{w}$ to $\mathbf{w}'$ as a one-dimensional problem. Hence we can replace the entire section with a path running along the edge from $\mathbf{w}$ to $\mathbf{w}'$ as shown in Fig. 7(b) - the command acceleration for this section being bang-bang and parallel to the edge. This transformation does not affect the canonical nature of the trajectory since in the one-dimensional case a single bang-bang switch gives the optimal trajectory. If we treat only the first and last points along such a path (i.e. one moving along an edge) as contact points, we see that each edge can contribute only two contact points to the path. Other configurations which may arise with contacts at a given segment are treated similarly. Thus, the path is made up of $O(n)$ canonical segments.

Finally, we see that the trajectory obtained is time-optimal : for all k, $\mathbf{\Gamma_{k+1}}$ takes time less than or equal to that taken by $\mathbf{\Gamma_k}$. This follows from the fact that in $\mathbf{\Gamma_k}$ the section from $\mathbf{Z_k}$ to $\mathbf{Z}'$ is bang-bang in the constrained direction and therefore takes time less than or equal to that taken by $\mathbf{\Gamma_k}$ between the same points. The theorem therefore follows.

■

8 Generating the optimal canonical trajectory

In this section, we will sketch the process of generating the optimal canonical trajectory. From the previous section, we know that there exists an optimal canonical trajectory from **S** to **F** which passes through at most n distinct vertices and has at most $2n$ contact points which touch edges. In this section, we show how to find it using only a polynomial (in the input) amount of space. We will use the theory of the reals to that end. An expression in the first order theory of the reals has the form

$$Q_1 x_1 \ldots Q_n x_n \quad P(x_1, \ldots, x_n)$$

where each Q_i is one of the quantifiers $\exists$ or $\forall$ and $P(x_1, \ldots, x_n)$ is a quantifier free boolean formula with atomic predicates of the form $f_j(x_1, \ldots, x_n) = 0$ or $g_k(x_1, \ldots, x_n) \geq 0$, f_j and g_k being real polynomials.

For any two (indeterminate) points in free space, with two (indeterminate) velocities, we can write a collection of polynomial clauses which describe the canonical segment between the two points, in terms of the end-point indeterminates, as was shown in Section 5. Combining these clauses with other clauses which check that the segment does not pass through obstacle space, we get a clause $Q(\mathbf{z_i}, \mathbf{z_{i+1}}, \mathbf{v_i}, \mathbf{v_{i+1}}, t_i, t_{i+1})$ which is true iff there exists a canonical segment (not touching the obstacle space) joining $\mathbf{z_i}$ and $\mathbf{z_{i+1}}$ with velocities $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$ and at times t_i and t_{i+1} respectively. Let M denote a sequence of vertices and edges, i.e. M describes a possible sequence of contact points for a canonical trajectory. For every vertex point in S, we have two indeterminates, the velocities in the x and y directions. Similarly for every edge, we have first of all two contact points, and for each contact point we have two indeterminates - one is the distance along the edge and the other is the velocity magnitude (since we know the velocity direction is along the edge from the first contact point to the second one). For each point $\mathbf{z_i}$ (whether a vertex or edge point), we will represent its indeterminates by $\mathbf{p_i} \in \Re^2$. Let $R_M(\mathbf{z_0}, \mathbf{z_{3n+1}}, t)$ (where $(\mathbf{z_0}, \mathbf{v_0}) = \mathbf{S}$ and $(\mathbf{z_{3n+1}}, \mathbf{v_{3n+1}}) = \mathbf{F}$) be true iff there exists a path of at most $3n+1$ canonical segments from **S** to **F** following the sequence M. We can write R_M in terms of Q as follows,

$$\exists \mathbf{p_0}\ \exists \mathbf{p_1} \ldots \exists \mathbf{p_{3n+1}}\ \exists t_0\ \exists t_1 \ldots \exists t_{3n+1} \bigcap_{i=0,\ldots,n} Q(\mathbf{z_i}, \mathbf{z_{i+1}}, \mathbf{v_i}, \mathbf{v_{i+1}}, t_i, t_{i+1}) \cap (t = t_{3n+1}) \tag{13}$$

where the $\mathbf{p_i}$'s denote the indeterminates corresponding to each contact point as mentioned above. We will try every possible sequence of vertices and edges. At each stage we will compare a new sequence M' with the previous best, say M, using the following formula,

$$\exists t'\ \forall t\ (R_{M'}(\mathbf{z_0}, \mathbf{z_{3n+1}}, t')) \wedge (R_M(\mathbf{z_0}, \mathbf{z_{3n+1}}, t) \Rightarrow (t' < t)) \tag{14}$$

We note that the number of alternations between $\forall$'s and $\exists$'s is a constant (the Q's each have a $\forall$ but all the Q's are placed together, so there is only one alternation). We can therefore use Renegar's PSPACE algorithm [Re] for the theory of reals with a bounded number of quantifier alternations. This algorithm will give as output a polynomial of degree $2^{N^{O(1)}}$, where N is the size of the input, such that the total time and velocities of the via points are roots of this polynomial. At each call to above algorithm, we only need remember the previous best sequence, thus we need only a polynomial amount of space to perform this computation. At the end of this computation we will have an polynomial which will encode the optimal trajectory in the form of its roots. We can now apply Neff's algorithm [Ne] to this polynomial. Neff's algorithm takes as input a polynomial p of degree k with all coefficients with absolute values bounded above by 2^m and a specified integer μ and determines all roots of the polynomial with error less than $2^{-\mu}$. It runs in parallel time $O(\log^3(k+m+\mu))$ with at most $POLY(k+m+\mu)$ number of processors, which also implies that it takes space which is poly-logarithmic in the input quantities. Thus applying Neff's algorithm to our polynomial with $k = \mu = m = 2^{N^{O(1)}}$, we get an exponential number of bits of the time and contact velocities of the optimal trajectory using a polynomial (in the input N) amount of space.

We note that the output of Renegar's algorithm is a polynomial of exponential degree and size and this polynomial is the input to Neff's algorithm. However the entire computation can still be carried out in PSPACE using the standard PSPACE simulation of two PSPACE machines interacting as above via an exponential size tape, where both machines use polynomial size work tapes. Therefore, we have

Theorem 3 *Under the L_∞ norm, the two-dimensional kinodynamic problem $(O, a_{max}, v_{max}, \mathbf{S}, \mathbf{F})$, where $\mathbf{S} = (\mathbf{s}, \mathbf{0})$ and $\mathbf{F} = (\mathbf{f}, \mathbf{0})$, is in PSPACE.*

8.1 Comparison with Approximation Algorithms

Although our algorithm can supply an exact description of the minimum time trajectory, i.e. a polynomial with the contact velocities and time as roots, a robot controller would require the numerical approximation to the trajectory which we obtain using Neff's algorithm as described above. It is therefore worth comparing our method with numerical approximation schemes such as [CDRX] or [JHCP] for the minimum time problem. It could be argued that we could use such a grid-based approximation algorithm (which normally runs in polynomial time on a polynomially large grid) and run it on an exponentially large grid to get an equivalent solution. This however is not the case. There are three critical differences between the *exact* method of this paper and a numerical approximation approach :

(i) The first difference is that the path(s) the exact method would find *is(are)* the actual minimum-time path(s) whereas the numerical approach makes no such guarantees; the path found by the numerical approach as in [CDRX] need not even be homotopic to the minimum-time path.

(ii) The second difference is that the grid-based approach is restricted to points which lie on the grid and as such is limited by the resolution of the grid. If an exponential number of points where used, the grid algorithm would return points within an accuracy of a polynomial number of bits whereas the exact algorithm would return an exponential number of bits of the contact velocities etc. and would give a far more accurate description of the minimum-time trajectory in the same time.

(iii) The exact algorithm uses only polynomial space. The grid approach would perform a bread-first-search on an exponential size graph and it is not obvious that this can be done in PSPACE.

9 Conclusions and Open Problems

In this paper we gave the first algorithm for generating an exact time-optimal trajectory for a kinodynamic planning problem in the plane. This algorithm runs in PSPACE and takes exponential time. We characterized the nature of a class of solutions which provides us with a time-optimal solution. We proved a tracking lemma which is applicable to arbitrary norms. This lemma should prove to be useful in dealing with the different variations to this problem and is also of general interest since it provides bounds on the time difference between a given trajectory and one which tracks it. We then showed how to eliminate loops from a given trajectory in order to limit the number of contact points a time-optimal trajectory might have. Finally, we drew upon the theory of reals (with a bounded number of alternations between quantifiers) to obtain a time-optimal solution. Several questions arise almost immediately :

1. The complexity of this problem - is it NP-hard or can it be done in polynomial time ?
2. Can the above algorithm be modified to handle the case of non-zero velocities at the beginning and end of the trajectory ?
3. Can the above approach be extended to the case of arbitrary norms, especially the L_2 norm? (See [RT] for work in this direction.) We have shown how to eliminate loops for arbitrary norms; the main problem now is characterizing the time-optimal solutions.
4. Exact algorithms for the three-dimensional case.

The second problem mentioned above requires further elaboration : If we allow for non-zero start and end velocities, it is obvious that a time-optimal trajectory would, in general, contain loops (e.g. the trivial case where the start and end position are the same but have different velocities). It seems intuitive that the number of loops required would be some function of the terminal velocities. Each loop would add an extra segment to the optimal trajectory and hence an extra clause to the formula describing the optimal trajectory. Thus, if the function were poly-logarithmic, we would only have a polynomial (in the input) number of clauses added to the final description and the algorithm would run as before. However, it is not clear at present what form this function would take.

There are several other open problems within the broad framework of kinodynamic planning. One of the most interesting problems, from both theoretical and practical viewpoints, is that of generating time-optimal motions for nonholonomic systems, i.e. systems for which the dynamical constraints are not integrable [BL]. [JRL] consider the problem of generating time-optimal motions for an Hilare-like mobile robot (i.e. a robot with two independently controlled wheels). The form of time-optimal trajectories for such systems, even without obstacles, is not known at present. As mentioned earlier, Fortune and Wilfong [FW] and Jacobs [J] addresses the problem of determining the shortest bounded curvature path in the plane in the presence of obstacles. However, at present, only an approximation algorithm is known for this problem and it would be of significant interest to determine the complexity of this problem and to provide an exact algorithm for it. Additionally, a considerable amount of work needs to be done to translate such results into practice.

10 References

[BDG] J.E. Bobrow, S. Dubowsky and J.S. Gibson, "On the Optimal Control of Robotic Manipulators with Actuator Constraints", *Proc. of the ACC*, San Francisco, (1983), pp. 782-787.

[BL] J. Barraquand and J.-C. Latombe, "On Non-holonomic Mobile Robots and Optimal Maneuvering", *Revue d'Intelligence Artificielle*, 3(2), 1989.

[CDRX] Canny, J., Donald, B., Reif J. and Xavier, P., "On the Complexity of Kinodynamic Planning", *Proc. 29th IEEE Symposium on Foundations of Computer Science*, New York, (1988), pp. 306-318.

[CR] J.F. Canny and J. Reif, "New lower bound techniques for robot motion planning," in *28th IEEE Symposium on Foundations of Computer Science*, (Los Angeles), 1987.

[DX] Donald, B. and Xavier, P., "A Provably Good Approximation Algorithm for Optimal-Time Trajectory Planning", *Proc. IEEE Int. Conf. on Robotics and Automation*, Scottsdale (1989), pp. 958-963.

[FW] Fortune, S. and Wilfong, G., "Planning Constrained Motion" *Proc. ACM Symposium on the Theory of Computing*, Chicago, (1988), pp. 445-459.

[Hol] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," in *Proceedings of the ACC*, (San Francisco, CA), pp. 752–756, The American Automatic Control Council, June 1983.

[J] P. Jacobs, "Planning Robot Motion with Dynamic Constraints", Ph.D. dissertation, EECS Department, University of California, Berkeley (1989).

[JHCP] P. Jacobs, G. Heinzinger, J. Canny and B. Paden, "Planning Guaranteed Near-Time-Optimal Trajectories for a Manipulator in a Cluttered Workspace", in *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, (1990), to appear in *IEEE Journal of Robotics and Automation.*

[JRL] P. Jacobs, A. Rege, J.-P. Laumond, "Non-holonomic Motion Planning for Hilare-like Mobile Robots", in *Intelligent Robotics, Proceedings of the International Symposium on Intelligent Robotics*, (Bangalore, India), pp. 338-347, January 1991.

[LCH] Z. Li, J. Canny, and G. Heinzinger, "Robot motion planning with nonholonomic constraints", in *International Symposium on Robotics Research*, August 1989.

[Ne] Neff, C.A., "Specified Precision Polynomial Root Isolation is in NC", in *Proceedings of the 31st Annual Symposium on the Foundations of Computer Science*, (St. Louis), pp. 152-162, Oct. 1990.

[N] W. Nelson, "Continuous-curvature paths for autonomous vehicles", in *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, (Scottsdale), pp. 1260–1264, May 1989.

[O] C. Ó'Dúnlaing, "Motion planning with inertial constraints", *Algorithmica*, vol. 2(4), (1987), pp. 431-475

[RT] Reif, J. and Tate, S., "Approximate kinodynamic planning using L_2 norm dynamic bounds", Duke University Tech. Report (1989).

[Re] Renegar, J., "On the computational complexity and geometry of the first order theory of the reals", Technical Report no. 853, School of O.R. and I.E., Cornell University, NY, July 1989.

[SH] G. Sahar and J. M. Hollerbach, "Planning of minimum-time trajectories for robot arms", Tech. Rep. A.I. Memo No. 804, MIT, November 1984.

[Sch] H.M. Schaettler, "On the Optimality of Bang-Bang Trajectories in $\Re^3$", *Bull. AMS*, vol. 18(1), pp. 113-6 (1987).

[SD] Z. Shiller and S. Dubowsky, "Global Time-Optimal Motions of Robotic Manipulators in the Presence of Obstacles", *IEEE Int. Conf. on Robotics and Automation*, Philadelphia, (1988).

[SM] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints", *IEEE Transactions on Automatic Control*, vol. AC-30, pp. 531–541, June 1985.