A Provably Good Approximation Algorithm for Optimal-Time Trajectory Planning*

Bruce Donald
Patrick Xavier

TR 88-947
November 1988

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*Support for this work was provided by the National Science Foundation under grant IRI-88-02390 and by The Mathematical Sciences Institute.

A Provably Good Approximation Algorithm for Optimal-Time Trajectory Planning

Bruce Donald

Patrick Xavier

Cornell Computer Science Department

Abstract: We consider the following problem: given a robot system, find a minimal-time trajectory from a start position and velocity to a goal position and velocity, while avoiding obstacles and respecting dynamic constraints on velocity and acceleration. Based on the theoretical results of [CDRX], we have developed and implemented a new, provably good approximation algorithm for the minimum-time trajectory problem. Our algorithm differs from previous work in three ways. First, it is possible to bound the goodness of the approximation by an error term ϵ. Second, we can polynomially bound the running time (complexity) of our algorithm. Third, we can express the complexity as a polynomial function of the error term. Hence, one supplies the algorithm with the geometric obstacles, dynamics bounds, and the error term ϵ. The algorithm returns a solution that is ϵ-close to optimal, and promises to spend only a polynomial (in $(\frac{1}{\epsilon})$) amount of time computing the answer. In this paper, we describe the algorithm and explain the results in simple terms. We show how it can be applied to robotics, and report on an implementation and experiments.

1 Introduction

The *kinodynamic planning problem* [CDRX] is to synthesize a robot motion subject to simultaneous kinematic constraints (such as avoiding obstacles) and dynamic constraints (such as modulus bounds on velocity, acceleration, and force). A kinodynamic solution is a mapping from time to generalized forces. The resulting motion is governed by a dynamics equation. In robotics, a long-standing open problem is to synthesize *time-optimal* kinodynamic solutions, by which we mean a solution that requires minimal time and respects the kinodynamic constraints.

While there has been a great deal of work on this problem, with the exception of the one-dimensional case, there are no exact algorithms. Among the many proposed approximate or heuristic techniques, there exist no bounds on the goodness of the resulting solutions, or on the time-complexity of the algorithms.

Because the problem is quite hard, we feel that it is reasonable to pursue approximate algorithms, by which we mean algorithms that compute kinodynamic solutions that are "close" to optimal (in some appropriate sense). It is also desirable to find efficient algorithms, since these problems are currently computationally very intensive. In this vein, we feel it is natural to ask the following three questions about such an algorithm:

Open Questions

- How close to optimal is the solution?
- What is the running time (computational complexity) of the algorithm?
- Can the running time be related to the desired solution accuracy?

Hence, ideally, we would like to find an algorithm for which one could guarantee the solution accuracy to some tolerance ϵ. For a fixed ϵ, one hopes such an algorithm would be efficient (i.e., polynomial-time). While one might expect such an algorithm to run longer when a more accurate solution was required, the algorithm would promise to spend only a polynomial amount of time longer, as ϵ was made smaller. We call such an algorithm a *provably good approximation algorithm*. Despite a great amount of previous work, to our knowledge no such algorithm exists, and all the questions above remain open.

Because the problem is quite hard, we have started with a simplified case. We consider the restricted situation of particle dynamics, and provide a provably good approximation algorithm for 2- and 3-dimensional optimal kinodynamic planning. We have developed and implemented an algorithm based on our earlier, theoretical results in [CDRX]. Roughly speaking, we show that if there exists a "safe" optimal-time kinodynamic solution requiring time t, then we can find a "near-optimal" solution that requires time $(1+\epsilon)t$. Furthermore, the running time of our algorithm is polynomial in the both in the closeness of the approximation $\frac{1}{\epsilon}$ and in the geometric complexity. (This is true even in 3 dimensions, where computation of an exact solution can be shown to be $\mathcal{NP}$-hard).

These bounds on solution accuracy and running time are the first that have been obtained for 2D and 3D optimal kinodynamic planning, which has been an open problem in computational robotics for over ten years. Our algorithm is very simple, and we have implemented it. We report on experiments with the implementation later in this paper. Finally, we believe that our approach might well extend to robot systems with full dynamics.

2 Kinodynamic Motion Planning

Kinodynamic planning attempts to solve a motion problem subject to simultaneous kinematic and dynamic constraints. We wish to consider the following problem. A point mass in $\Re^d$ ($d = 2, 3$) must be moved from a start position and velocity $\mathbf{s} = (s, \dot{s})$ to a goal position and velocity $\mathbf{g} = (g, \dot{g})$. In the course of the motion, it must avoid a set of polyhedral obstacles: these are the *kinematic* constraints. The point is commanded to move by applying forces (or equivalently, commanding accelerations). The corresponding motion is governed by Newtonian dynamics. However, there are upper bounds on the magnitude of the commanded accelerations. These bounds are given by an L_∞-norm: for all times t, the acceleration $\mathbf{a}(t)$ is bounded by the inequality

$$|\mathbf{a}(t)|_\infty \leq a_{max}. \tag{1}$$

In addition, we have velocity bounds that the solution must respect. For a smooth path $\mathbf{p}$, we must ensure that

$$|\dot{\mathbf{p}}(t)|_\infty \leq v_{max}. \tag{2}$$

Eqs. (1) and (2) are called *dynamic* constraints.

We will denote our configuration space $\Re^d$ by C, and its phase space by TC. Phase space TC is isomorphic to $\Re^{2d}$ and a point in TC is a (position, velocity) pair such as $\mathbf{g}$ or $\mathbf{s}$.

The *commanded acceleration* is a map $\mathbf{a} : [0, b] \to \Re^d$ for a closed interval $[0, b]$. The *path* $\mathbf{p}$ corresponding to $\mathbf{a}$ is its second integral subject to the initial position and velocity $\mathbf{s}$, and the *trajectory* Γ for $\mathbf{a}$ and $\mathbf{s}$ is the mapping $\Gamma : [0, b] \to TC$ taking a time t to $(\mathbf{p}(t), \dot{\mathbf{p}}(t))$. Thus $\dot{\mathbf{p}}$ is the time derivative of $\mathbf{p}$, and $\mathbf{a} = \ddot{\mathbf{p}}$.

Let us assume the polyhedral obstacles are input as an arrangement $\mathcal{E}$ with n vertices. *Free space* is the complement of these obstacles. A *general kinodynamic planning problem*, then, is a tuple $(\mathcal{E}, \mathbf{s}, \mathbf{g}, a_{max}, v_{max})$. We assume that the set of free configurations is bounded by a square of side length l.

A *solution* to the kinodynamic planning problem is a suitable encoding of the acceleration map $\mathbf{a}$ such that $\Gamma(0) = \mathbf{s}$, $\Gamma(b) = \mathbf{g}$, and Γ obeys the kinematic and dynamic constraints. That is, $\mathbf{p}$ avoids all obstacles, $\dot{\mathbf{p}}$ respects (2), and $\mathbf{a}$ respects (1).

The *time* for a solution $\mathbf{a}$ is simply b. The *optimal* kinodynamic planning problem is to find a kinodynamic solution with minimal time.

Figure 1: A kinodynamic planning problem, showing the obstacles, the start $\mathbf{s}$, the goal $\mathbf{g}$.

However, the theoretically optimal solution may still be unrealizable by a physical robot, (even if it is a point!) This is because robot control systems cannot accurately navigate through tight obstacle channels at high speeds. We would like to take this constraint into account in our analysis, in order to supply a result that is not only theoretically interesting, but also perhaps of practical value. Thus we define the notion of a *δ_v-safe kinodynamic solution.* The intuition behind such solutions is that they avoid obstacles by a safety margin δ_v. Furthermore, this safety margin is an affine function of the trajectory speed. We choose the safety margin *a priori* using two positive scalars c_1 and c_0. One may think of this choice as corresponding to how accurately the dynamical system can control its energy consumption. This variation specifies a tube about the safe path that must remain obstacle-free. We call this tube a *$\delta_v(c_1, c_0)$-tube*; it grows linearly in size with speed.

Formally, a *δ_v-safe kinodynamic solution* has the property that for all times t in $[0, b]$, there exists a ball about $\mathbf{p}(t)$ in free space of radius $\delta_v(t) = c_1|\dot{\mathbf{p}}(t)| + c_0$. (Here "ball" is used in a topological sense: i.e., its shape depends on the metric being used.)

Now, for fixed c_1, c_0, consider the class of all δ_v-safe kinodynamic solutions. We define an *optimal δ_v-safe kinodynamic solution* to be a δ_v-safe solution whose time is minimal in the class of δ_v-safe solutions. We will henceforth abbreviate this to "*optimal safe kinodynamic solution*" since δ_v-safety is the only type we consider here.

We must specify precisely what it means for one kinodynamic solution to "approximate" another. First, we say that a trajectory $\Gamma_q : [0, t_q] \rightarrow TC$ is "ϵ-close" at the start s (resp. goal $\mathbf{g}$) if, at time 0 (resp. time t_q), it achieves the start (resp. goal) position to within an additive error that is polynomial in ϵ, and the start (resp. goal) velocity to within a multiplicative error of $(1 + \epsilon)$, plus or minus another additive error, also polynomial in ϵ. That is, if $\Gamma_q = (\mathbf{p}_q, \dot{\mathbf{p}}_q)$, then both $|\mathbf{p}_q(0) - s|_\infty$ and $|(1 + \epsilon)\dot{\mathbf{p}}_q(0) - \dot{s}|_\infty$ are $\epsilon^{O(1)}$ (and similar conditions hold at the goal). The idea is that by making ϵ smaller, Γ_q can be made to achieve the start and goal as accurately as desired.

Finally, we define an *approximate optimal safe solution* Γ_q to be a kinodynamic solution which is "ϵ-close" at the start and the goal, "near-optimal" in time, and "also safe". By "near optimal", we mean that if the optimal safe solution Γ takes time b, the time t_q required by Γ_q is bounded above by $(1 + \epsilon)b$. By "also safe", we mean nearly δ_v-safe: specifically, that the position component $\mathbf{p}_q$ of Γ_q lies in an obstacle-free safety tube $\delta_v(c'_1, c'_0)$ for constants c'_1, c'_0, where $c'_0 = (1 - \epsilon)c_0$ and $c'_1 = (1 - \epsilon)c_1$. Note that δ_v-safety is actually a *mixed* kinematic and dynamic constraint; it is an example of a kinodynamic constraint that is neither purely kinematic nor purely dynamic.

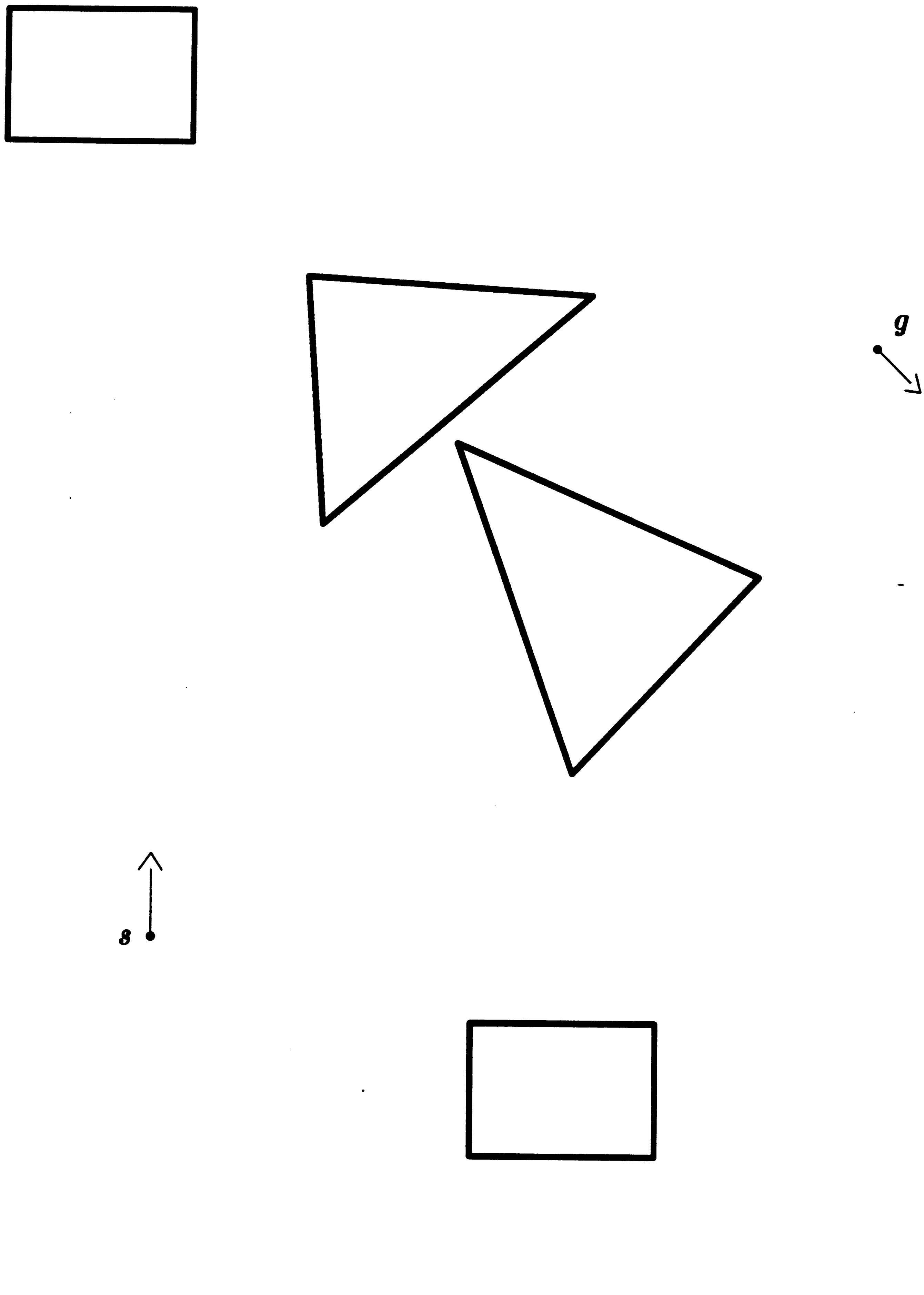

g
s

Figure 2: Γ is the optimal kinodynamic solution. However, it is not δ_v-safe. Γ_r is the optimal δ_v-safe solution. Γ_q illustrates a solution that is almost as fast as Γ_r, and almost as safe. Our algorithm can find solutions such as Γ_q, and provides guarantees as to how fast, and how safe they are, relative to Γ_r.

Figure 3: A near-optimal trajectory that was found by our implementation. (The "spikes" indicate the velocities).

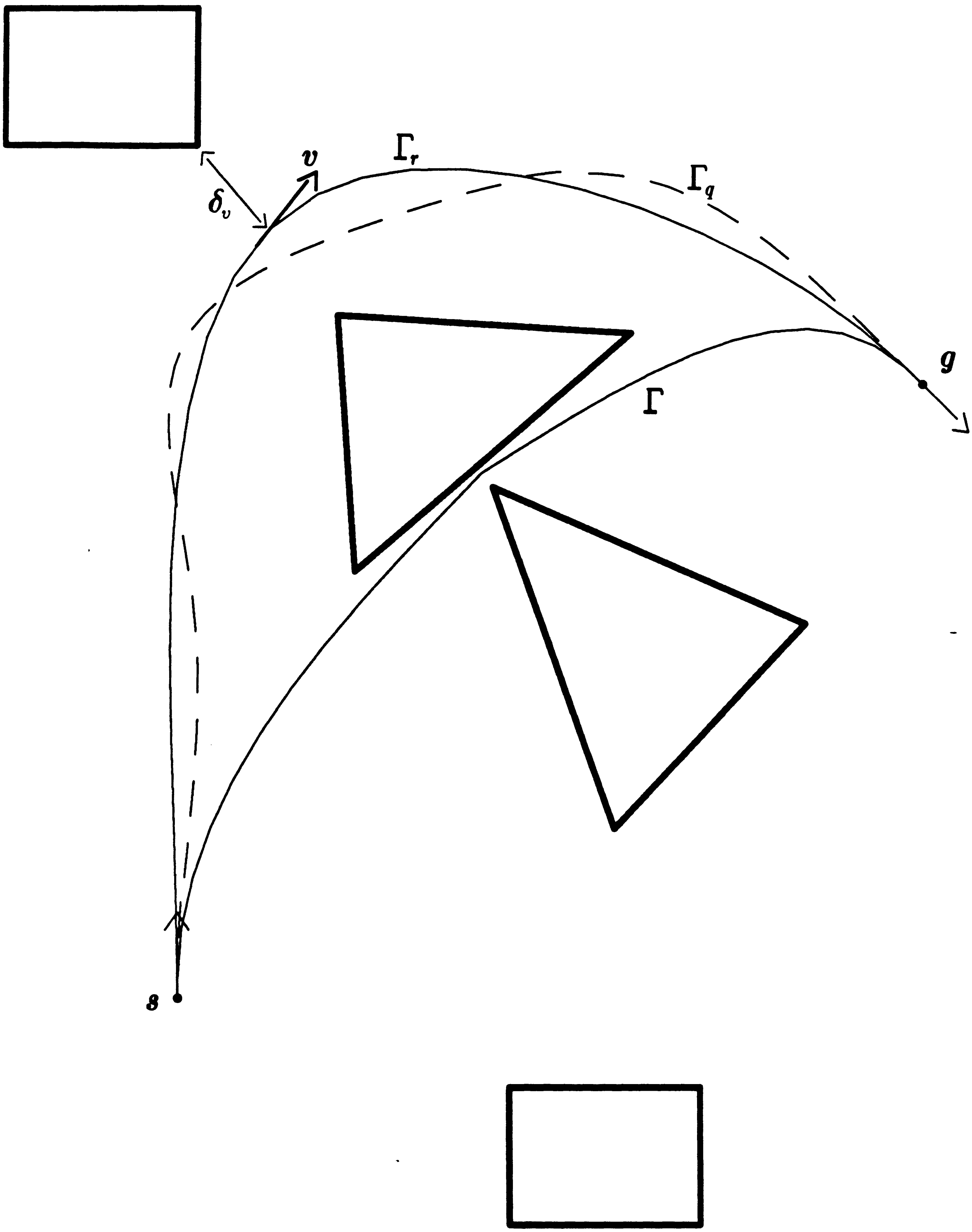
2.1 Statement of Results

In this paper, we assume the workspace has unit diameter ($l = 1$). We describe a provably good approximation algorithm for the optimal safe kinodynamic planning problem $\mathcal{K} = (\mathcal{E}, \mathbf{s}, \mathbf{g}, a_{max}, v_{max}, c_1, c_0, \epsilon)$. The algorithm produces an approximate optimal safe solution. ϵ, which is an input parameter, specifies how close in time the desired solution should be to the optimal, safe solution. c_1, c_0 specify the class of trajectories to be considered "safe."

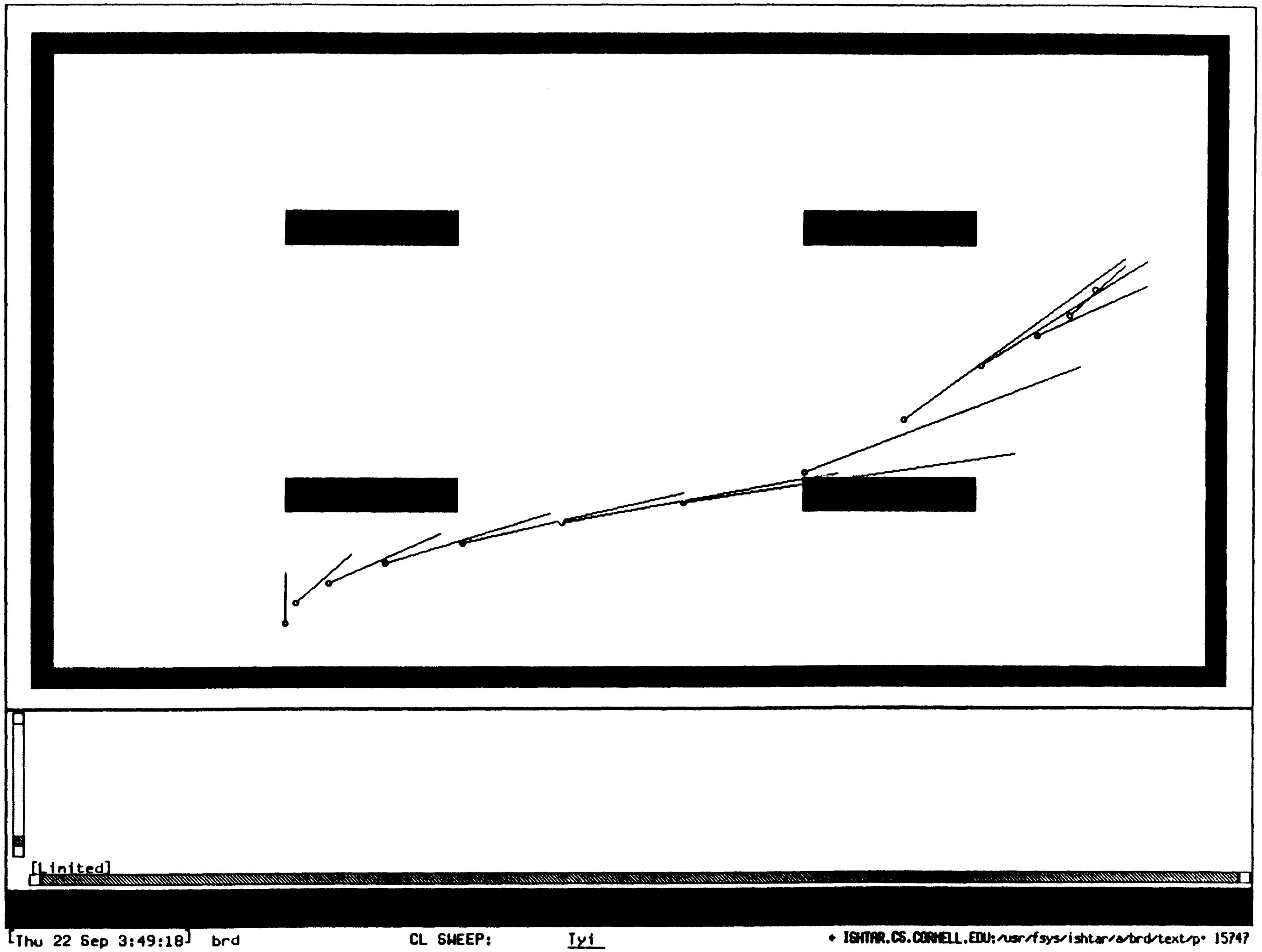
Our algorithm runs in time polynomial in the geometric complexity n, and in the resolution $(\frac{1}{\epsilon})$. Thus we can bound both the goodness of our approximation, and the running time of the algorithm. Furthermore, we can relate the running time to the error term, and show that this relationship is polynomial.

More precisely, we observe that an optimal safe kinodynamic planning problem $\mathcal{K}$ has three components: The *combinatorial complexity* of $\mathcal{K}$ is the number n of vertices in the arrangement of obstacles $\mathcal{E}$. The *algebraic complexity of the geometry* is the number of bits necessary to encode the coordinates of the vertices of $\mathcal{E}$, and the start and goal states. The *algebraic complexity of the kinodynamic bounds* is the number of bits necessary to encode the kinodynamic bounds $(a_{max}, v_{max}, c_1, \frac{1}{c_0})$. In the language of combinatorial optimization [PS], we show that our algorithm is an ϵ-approximation scheme that is *fully polynomial* in the combinatorial and algebraic complexity of the geometry, and *pseudo-polynomial* in the kinodynamic bounds.

Note, however, that we cannot claim that the approximate optimal safe solution is necessarily near (in position space) to the (true) optimal safe solution. In this respect it is useful to compare Papadimitriou's fully polynomial approximation scheme for 3D Euclidean shortest path [Pap]. Specifically, neither method finds a solution that is necessarily close (in position space) to the optimal path, but merely one that has a length (or, in our case time) that is not too much longer than the optimal length (resp. time). In fact, the results of [CR] imply that finding a path that is position-space close to the shortest path, or even one that is homotopic to the optimal is $\mathcal{NP}$-hard.



v
Γ_r
Γ_q
δ_v
g
Γ
s


[Linited]
Thu 22 Sep 3:49:18 brd
CL SWEEP:
Iyi
• ISHTAR.CS.CORNELL.EDU:/usr/fsys/ishtar/a/brd/text/p• 15747

2.2 Review of Previous Work

For a review of issues in robotics and algorithmic motion planning, see [Bra, Y]. There exists a large body of work on optimal control in the control theory and robotics literature. For example, see [Hol, BDG, Sch, SS1, SS2]. Much of this work attempts an analytic characterization of time-optimal solutions—for example, to prove that in certain cases piecewise-extremal ("bang-bang") controls, with a finite number of switchings, suffice. This has led to many interesting and deep subresults. For example, [BDG, Hol] show how given a *particular* trajectory $\Gamma = (\mathbf{p}, \dot{\mathbf{p}})$, its velocity profile can be rescaled so as to respect dynamic constraints and to be time-optimal. Using these ideas, a number of authors have proposed heuristic or approximate algorithms for what is hoped to be near time-optimal trajectory planning. In particular, Sahar and Hollerbach [SH] and Shiller and Dubowsky [SD] both implemented algorithms which employ a fixed-resolution configuration-space or phase-space grid to compute, approximately, near minimal-time trajectories for robots with several degrees of freedom (and full dynamics). They did not bound the goodness of their approximation, nor the [illegible]g time of their algorithm. However, their grid methods take time which grows exponentially with the number of grid points, or the resolution. We provide the first polynomial-time algorithm.

The polyhedral euclidean shortest path problem is a version of optimal kinodynamic planning with the acceleration bound a_{max} set equal to infinity. This observation may be used to extend the results of [CR] to show that in 3D, optimal kinodynamic planning is $\mathcal{NP}$-hard. In other work, Ó'Dúnlaing [O] provides an exact algorithm for one-dimensional kinodynamic planning. These methods may extend to the 2- and 3D cases as well. Kinodynamic planning in 2D is related to the problem of planning with non-holonomic constraints, as studied by Fortune and Wilfong [FW, W]. In this problem, a robot with wheels and a bounded minimum turning radius must be moved. To make the analogy clear, in our case, the minimum turning radius is $\frac{1}{a_{max}}\|\dot{\mathbf{p}}\|^2$. These algorithms might lead in time to an exact solution to kinodynamic problems in 2D and 3D.

3 Description of the Approach

3.1 The Basic Idea

The phase space TC is the state space for the particle robot. One can imagine a regular discretization of phase space TC. This discretization can be thought of as a "grid"; a point in the interstices is a "grid point". One could command a move from one grid point to another using piecewise-maximal accelerations. Such a motion is called a "bang" in the controls literature. Of course, not all grid-point neighbors will be reachable for an arbitrary discretization. However, it seems intuitively plausible that for a sufficiently "fine" grid, a grid-point bang path might approximate the optimal time path. We will call a trajectory that consists of bang-accelerations between grid points a "grid-bang trajectory". The key issues are to choose the grid spacing correctly and to prove that

Figure 4: A picture of the phase space grid in 1D. A plot of x versus $\dot{x}$. The cross indicates zero position and velocity. The dashed arrows from $(0,0)$ indicate the two states reachable after time step τ, under $+$ and $-$ (a,τ)-bangs. The six states reachable at time 2τ are indicated by dotted arrows. The 14 states reachable at time 3τ are indicated by solid arrows. The grid spacing is $a\tau^2$ in x, and $a\tau$ in $\dot{x}$.

---

the approximation bound

$$t_q \leq (1+\epsilon)b \tag{3}$$

holds. (b is the true optimal safe solution time, and t_q is the grid-bang approximate optimal safe solution time). Furthermore, the grid must be polynomial in size. The proofs of these properties require certain non-trivial constructions.

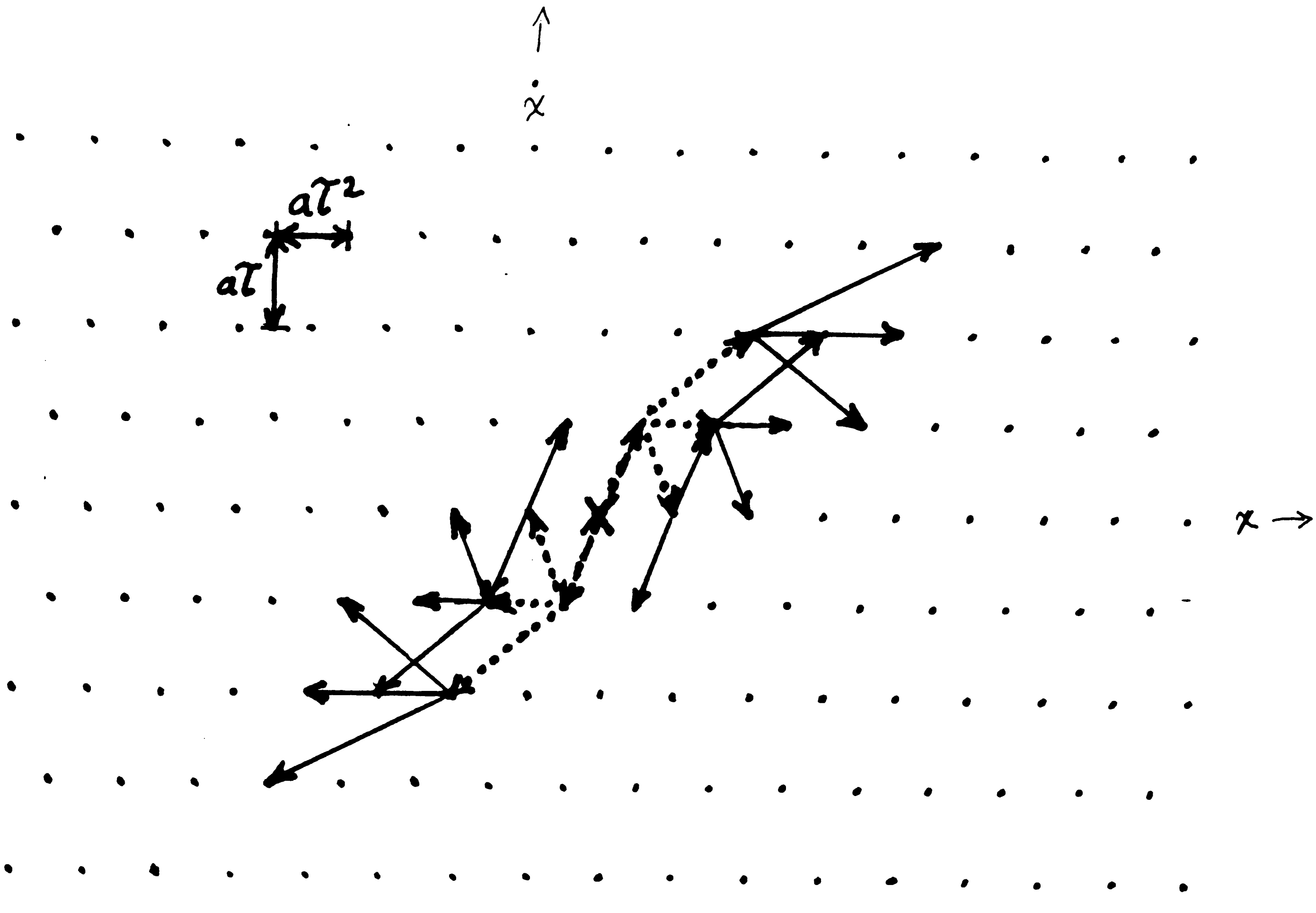
Our idea is to use a non-linear grid spacing.[1] The spacing is a function of the velocity. For a grid point $(\mathbf{x}, \dot{\mathbf{x}})$, let us call the distance from $(\mathbf{x}, \dot{\mathbf{x}})$ to each of its grid neighbors the *local grid spacing at* $(\mathbf{x}, \dot{\mathbf{x}})$. This distance has both a position and a velocity component. The local grid spacing in the velocity dimensions of phase space will be constant. The local grid spacing in a position dimension i will be an affine function of the local velocity $\dot{x}_i$. This will ensure that from a state $(\mathbf{x}, \dot{\mathbf{x}})$, all neighbors will be reachable under a single acceleration bang. Our approach is further distinguished from uniform grid algorithms in that at each time step, one is compelled to move to a neighbor. The algorithm can find a grid-bang trajectory that lies within some tube (in phase space) of the optimal trajectory. By staying within this tube, we can hope to achieve a constant multiplicative error bound in terms of time.

Wlog, we assume that the start $\mathbf{s}$ and goal $\mathbf{g}$ lie on the grid. Our algorithm first chooses a time-step τ that we derive as a function of a_{max}, ϵ, c_0, and c_1. Next, the algorithm performs a breadth-first search, that takes time $O(G \log G)$ in the number G of grid points (however, collision avoidance and δ_v-safety introduce an additional quadratic factor in the geometric complexity n).

More specifically, a queue is initialized to contain the start point $\mathbf{s}$. At each step of the algorithm, a state (phase space point) $(\mathbf{x}, \dot{\mathbf{x}})$ is popped off the queue. We check to see whether it is the goal. If it is, then we halt; the approximate grid-bang path has been found. We consider bangs (maximal accelerations) of duration τ along the major axes from $(\mathbf{x}, \dot{\mathbf{x}})$, in both $+$ and $-$ directions. For example, in 2D, we would consider bangs $\mathbf{a}$ of the form $a_1\hat{\imath} + a_2\hat{\jmath}$ for $a_i \in \{0, a_{max}, -a_{max}\}$. After time τ, the final phase space coordinates for each bang are taken as the new set of neighbors. (It is easy to verify that these grid points have a position offset that is affine in $\dot{\mathbf{x}}$, and a constant velocity offset). Those neighbors that have not been explored are pushed on the back of the queue. They must retain a pointer to their father grid point. So a queue entry must consist of (a) the grid-point $(\mathbf{x}, \dot{\mathbf{x}})$, (b) the grid-point's parent, and (c) the commanded acceleration $\mathbf{a}$ to

---

[1]Although our discretization does form a lattice in TC, the reachable neighbors are not the geometric neighbors. We continue to use the term "grid" for intuition.



$\dot{x}$
$a\tau^2$
$a\tau$
x

get there from the parent in time τ. A balanced tree can be employed to keep track of which grid points have already been explored.

To see that this algorithm is correct, we view the discrete search space as a directed graph. Vertices of the graph are grid points. Directed edges correspond to $(\mathbf{a}, \tau)$-bangs. Two vertices u and w are connected by an edge iff w is reachable from u by an $(\mathbf{a}, \tau)$-bang. The vertices connected to u by edges from u are called its neighbors. Since we have a normal form for the bangs $\mathbf{a}$, the out-degree of each vertex is fixed at 3^d for dimension d. The algorithm starts at s and begins constructing edges of the graph. The search terminates when either a path to the goal $\mathbf{g}$ is found, or the maximal connected component from s has been explored. Thus the algorithm reduces to directed graph search. We will show that the number G of vertices (grid-points) is polynomial. Since we are seeking a shortest path in a graph where all edges represent the same time step, we can use the breadth-first search algorithm above, which takes time $O(G \log G)$.

There are several complications. First, when a neighbor is generated, the resulting bang trajectory is considered:

$$\begin{array}{rcl} \Gamma_i : [0, \tau] & \rightarrow & TC \\ t & \mapsto & (\mathbf{p}, \dot{\mathbf{p}}) = (\mathbf{x} + \dot{\mathbf{x}}t + \frac{1}{2}\mathbf{a}t^2, \dot{\mathbf{x}} + \mathbf{a}t). \end{array} \tag{4}$$

Our planner must ensure that (a) $\mathbf{p}$ does not intersect any obstacles, (b) $\dot{\mathbf{p}}$ does not violate the velocity bounds (2), and (c) for all times t, $\mathbf{p}(t)$ is no closer than $c_1' |\dot{\mathbf{p}}(t)| + c_0'$ to any obstacle. (a) and (c) may be tested together by "growing" the obstacles affinely in TC as the velocity increases. Any bang-neighbor violating (b) or (c) is considered unreachable from $(\mathbf{x}, \dot{\mathbf{x}})$, and so is left unconnected in the directed graph.

If the obstacles are "grown" affinely with velocity, their boundaries form $O(n)$ algebraic surfaces in phase space. Collision detection for a single (a, τ)-bang can be accomplished by intersecting the quadratic trajectory (parameterized by time) with these surfaces, and performing $O(n)$ sign-tests on each intersection point. Thus collision detection can be done in time $O(n^2)$ per bang, and the overall complexity of the algorithm is $O(n^2 G + G \log G)$.

4 Outline of the Proof

We can now give a very basic overview of the approach we have used to show that our algorithm is a provably good approximation algorithm. Full proofs can be found in an appendix.

4.1 Time Scaling

We denote the position and velocity components of a subscripted trajectory Γ_r by $\mathbf{p}_r$ and $\dot{\mathbf{p}}_r$, resp. We say that a path p is *traversed* by a trajectory Γ under acceleration bound a_{max} if the image of the position component of Γ is equal to the image of p, and Γ respects (1). First we must prove a not very difficult lemma showing that allowing a

Figure 5: A tube in phase-space. Trajectory Γ_q approximately tracks Γ'_r to tolerance (η_x, η_y).

---

multiplicative time error of $(1+\epsilon)$ permits a trajectory to be traversed with a tighter acceleration bound. Intuitively, this permits an approximate, grid-bang trajectory with acceleration bound a to "keep up with" a trajectory that respects a smaller acceleration bound $\frac{a}{(1+\epsilon)^2}$, regardless of curvature.

Specifically, suppose we are given a trajectory Γ_r which is an optimal safe kinodynamic solution (see fig 2). Let $\zeta = \frac{1}{(1+\epsilon)}$. Then

$$\Gamma_r'(t) = \Big(\mathbf{p}_r(\zeta t), \zeta \dot{\mathbf{p}}_r(\zeta t)\Big) \tag{5}$$

is obtained by rescaling Γ_r; its image in position space is the same. Next, we must prove that there exists some (a,τ)-grid-bang trajectory Γ_q that is "sufficiently near" Γ'_r in phase space. The bound on solution accuracy will then follow from this relationship of "sufficient nearness." The bound on time complexity will follow by showing that the (a,τ)-grid need only have polynomial size—or equivalently, that τ, the time step, is only polynomially small. Thus the proof may be sketched as

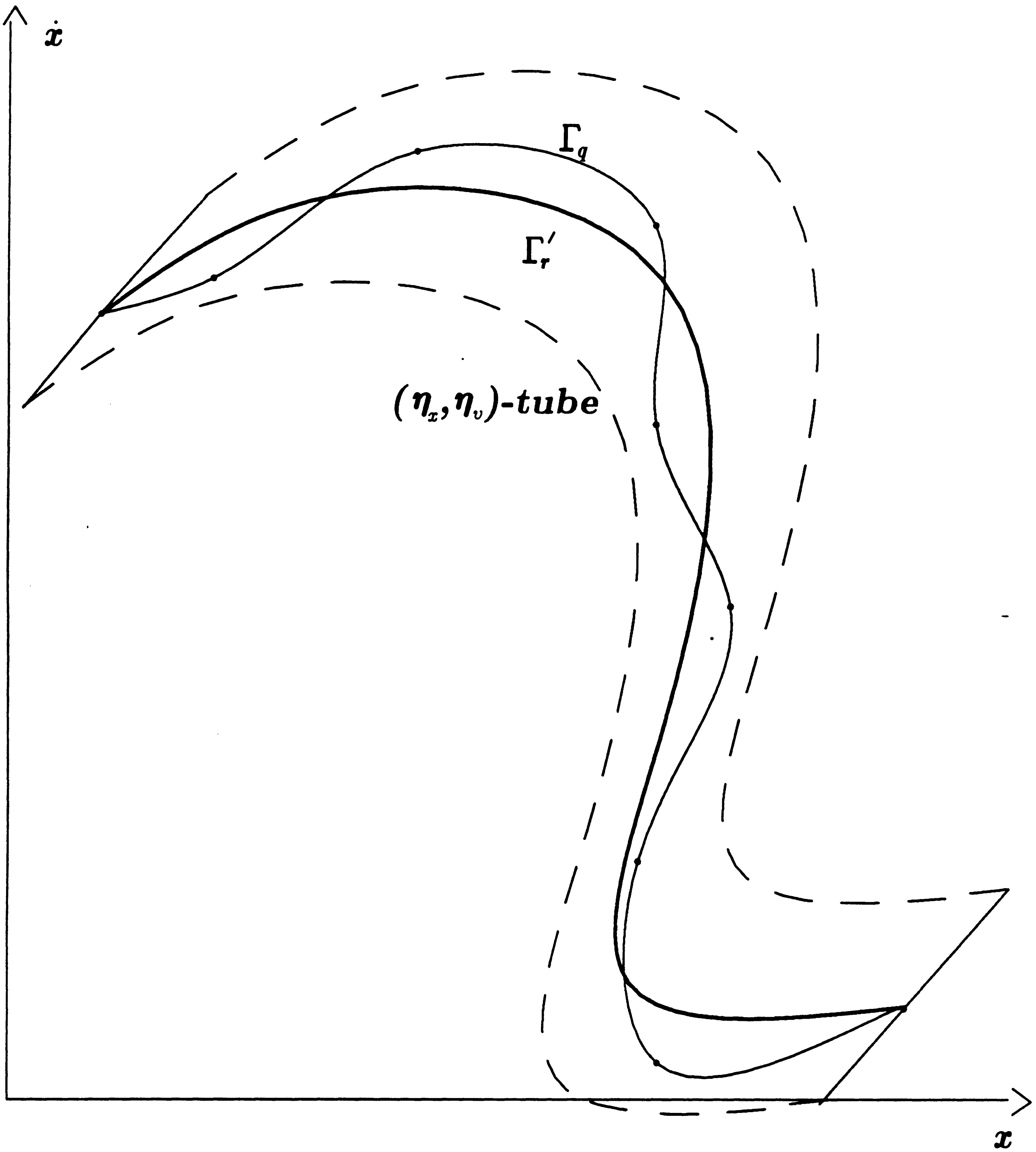
$$\Gamma_r \rightarrow \Gamma'_r \rightarrow \Gamma_q.$$

Now, a time spacing τ and an acceleration bound a define a non-linear grid on phase space, as in sec. 3.1. We call this an *(a,τ)-grid.* Recall that a *grid-bang* (or *(a,τ)-grid-bang*) trajectory starts on the grid, and is defined by a finite number of "bangs" (maximal or zero accelerations under the L_∞-norm) each of duration τ. We may think of it as successive bangs "between" grid points. Our goal is to choose a sufficiently small τ such that (3) holds, but still maintain a polynomial-size number of grid points. To this end, we must be able to show that τ can be chosen such that for any safe trajectory, there exists a "nearby" grid-bang trajectory that is "almost as fast." Since the simple structure of our grid assures neighbor reachability tautologously, it is easy to see that our algorithm will find such a grid-bang path if it exists, and is the fastest.

4.2 Tubes in Phase Space

It remains to make precise our notion of "sufficient nearness." This notion may be captured by a "tube" about a phase space trajectory. A tube about a trajectory (in phase space) is a neighborhood about the the trajectory image. (see fig. 5). It is specified by two metric parameters, one for position space, and one for velocity space. More precisely, consider two trajectories Γ_r, $\Gamma_q : [0,b] \rightarrow TC$. Given two scalars η_x and η_v, we say that we say that Γ_q *approximately tracks* Γ_r *to tolerance* (η_x, η_v) *in the* L_∞*-norm* if for all times t, $|\mathbf{p}_q(t) - \mathbf{p}_r(t)|_\infty \leq \eta_x$ and $|\dot{\mathbf{p}}_q(t) - \dot{\mathbf{p}}_r(t)|_\infty \leq \eta_v$.

Then, using certain non-trivial arguments, we are able to show the following *Tracking Lemma.* It essentially says the following: suppose we rescale the optimal trajectory Γ_r



$\dot{x}$
Γ_q
Γ_r'
(η_x, η_v)-tube
x

Figure 6: Knot points at which the trajectories Γ_q and Γ'_r are close in phase space. The knot times are times $kN\tau$ for integer $k \geq 0$.

---

to obtain Γ'_r (as above). Then we can choose τ so that the following hold. First, for any (η_x, η_v)-tube, τ can be chosen such that there is an (a, τ)-grid-bang solution within the tube about Γ'_r. Second, τ need only be polynomially small in the tube size (and in ϵ).

The Tracking Lemma: *Suppose a trajectory Γ'_r respects acceleration bounds $\frac{a}{(1+\epsilon)^2}$ and takes time T_r. Then in the absence of obstacles*

(a) for any positive η_x, η_v, there exists a time spacing τ and an (a, τ)-grid-bang trajectory Γ_q with bounded acceleration a that approximately tracks Γ'_r to tolerance (η_x, η_v) in time T_r.

(b) Moreover, τ is polynomial in ϵ, η_x, and η_v. Specifically, when $0 < \epsilon \leq 1$, τ can be chosen as:

$$\tau \leq \min\left(\sqrt{\frac{\eta_x \epsilon}{17a}}, \frac{\eta_v \epsilon}{12a}\right). \tag{6}$$

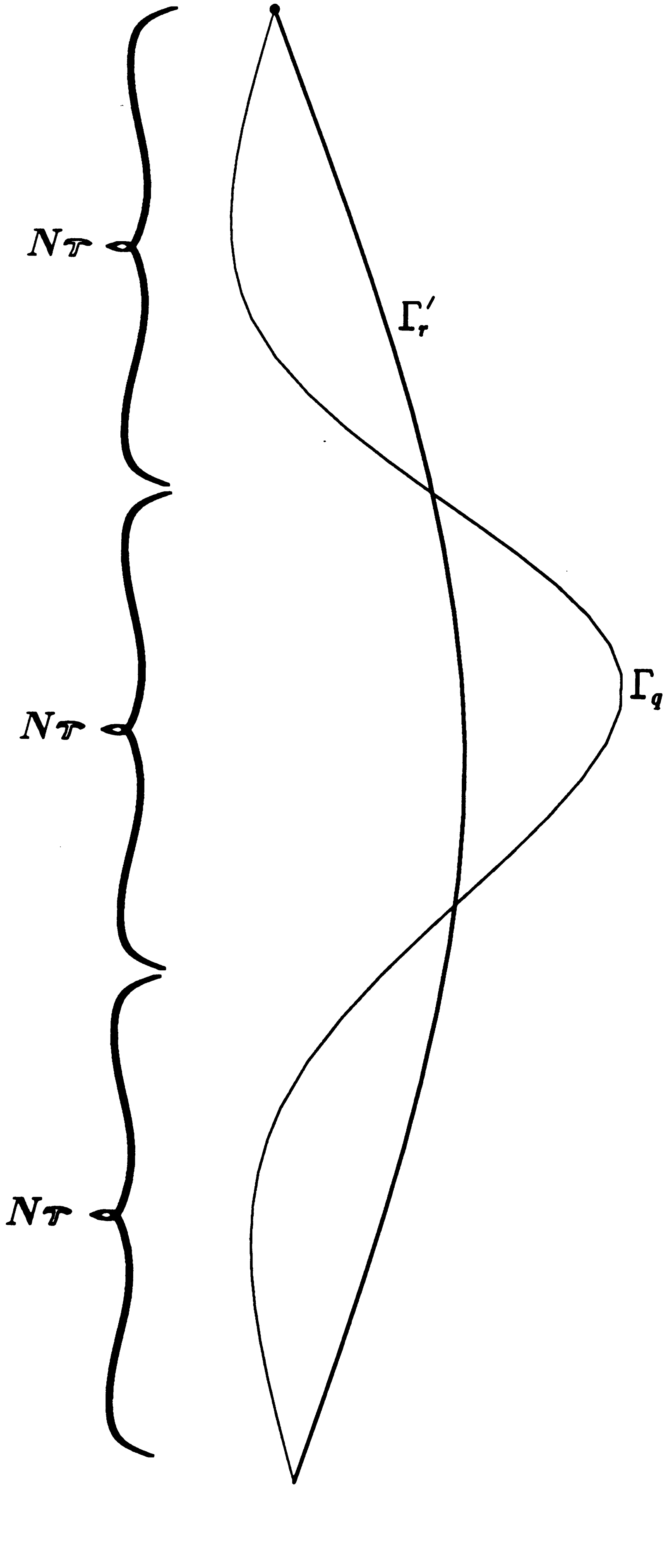
We now give some intuition for how one proves the Tracking Lemma. The essential technique is reminiscent of constructing spline-curve approximations. What we can show is that there exist "knot" points along Γ'_r and Γ_q at which the two trajectories are "close" in both position and velocity. See fig 6. Furthermore, these knot points are equally spaced in time; these times are called *knot times.* More precisely, we suppose that τ has been fixed (arbitrarily). Then, at every knot time t, we can show that the position components $\mathbf{p}_q(t)$ and $\mathbf{p}'_r(t)$ are no further apart than $\frac{1}{2}a\tau^2$ (in the ∞-norm). Similarly, the velocity components are no farther apart than $\frac{1}{2}a\tau$. Finally, the knot points are given by the sequence of times

$$0, N\tau, 2N\tau, 3N\tau, 4N\tau, \ldots$$

where N is an integer dependent on ϵ. In fact, it turns out that choosing N greater then 10.5ϵ suffices. Hence, every $N\tau$ seconds, the trajectories are guaranteed to converge. Now, since the convergance distance $(\frac{1}{2}a\tau^2, \frac{1}{2}a\tau)$ can be made arbitrarily small by chosing a small τ, we must then calculate how far apart the trajectories can stray between knot points. This calculation is not hard; for example, in the position case we must choose τ to ensure that $\frac{1}{2}(a\tau^2 + aN\tau^2) + aN\tau^2 < \eta_x$.

4.3 More Phase Space Tubes: The Case With Obstacles

Recall the construction of Γ'_r from Γ_r by rescaling. The Tracking Lemma does not yield a polynomially-small τ that guarantees that a trajectory Γ_r respecting acceleration bound a is tracked to a tolerance (η_x, η_v) by some (a, τ)-grid-bang trajectory. However, if τ



$N\tau$
$N\tau$
$N\tau$
Γ_r'
Γ_q

satisfies (6), there exists an (a,τ)-grid-bang trajectory that "follows" such a Γ_r in a weaker sense. Specifically, let $\zeta = \frac{1}{(1+\epsilon)}$ and let $\Gamma_r{}'$ be given by (5). If Γ_r is a-bounded, then $\Gamma_r{}'$ is $\zeta^2 a$-bounded, and thus by the Tracking Lemma, there is an (a,τ)-grid-bang trajectory that approximately tracks $\Gamma_r{}'$ to tolerance (η_x, η_v). It then follows from the definition of Γ_r' and the definition of "approximately tracks" that for all $t \in [0, (1+\epsilon)T_r]$ we have $|\mathbf{p}_q(t) - \mathbf{p}_r(\zeta t)|_\infty \leq \eta_x$.

We use this observation to motivate an extension of the Tracking Lemma to obstacle-avoiding trajectories. Recall that δ_v is an affine function of speed completely specified by two constants c_0 and c_1 which are input to the algorithm; henceforth we will abbreviate $\delta_v(c_0, c_1)$ by δ_v. Suppose that Γ_r is δ_v-safe, and recall the δ_v-tube for Γ_r (see sec. 2). It is then clear that Γ_r' must also tube δ_v-safe. Naively applying the observation, we might expect that if a trajectory Γ_q tracks Γ_r' imperfectly but closely enough, then the δ_v-tube induced by Γ_q would lie entirely in δ_v-tube induced by Γ_r. Since this is not generally true, it is natural to try a slightly weaker conjecture: for any $\delta_v{}'$ such that for all speeds y and some constant ϵ_v,

$$\delta_v{}'(y) \leq \delta_v(y) + \epsilon_v,$$

if Γ_q tracks Γ_r' closely enough, the δ_v'-tube induced by Γ_q will lie within the δ_v be induced by Γ_r. This conjecture is formalized by the following lemma.

The Safe Tracking Lemma: *Suppose that δ_v is specified by c_0 and c_1 and $0 < \epsilon < 1$, and let $\delta_v{}' = (1-\epsilon)\delta_v$. Then for a given acceleration bound a there exists a tolerance (η_x, η_v) such that for any trajectories Γ_r and Γ_r' as above (12), the following hold:*

(a) If Γ_q tracks Γ_r' to tolerance (η_x, η_v), then the δ_v'-tube induced by Γ_q lies within the δ_v-tube induced by Γ_r.

(b) Furthermore, for any positive β, the following choices suffice:

$$\begin{aligned} \eta_v &\leq \tfrac{c_0\epsilon}{c_1(1-\epsilon)+\beta} \\ \eta_x &\leq \beta\eta_v. \end{aligned} \tag{7}$$

Now, since we know that τ is polynomially dependent on η_x and η_v, we can apply the lemmas above and choose β to maximize the upper bound on τ. This yields the following choice of tau, which suffices for $0 < \epsilon < 1$:

$$\tau \leq \frac{2c_0\epsilon^2}{12ac_1 + \sqrt{144a^2c_1^2 + 68ac_0}}. \tag{8}$$

4.4 Bounding the Number of Grid Points

We now bound the number of (a,τ)-gridpoints for a point robot with maximum (L_∞) speed v_{max} in a d-dimensional free-space of diameter l. Let $G_\infty(a, \tau, v_{max}, l, d)$ denote this bound. Then

$$G_\infty(a,\tau,v_{max},l,d) = (G_\infty(a,\tau,v_{max},l,1))^d. \tag{9}$$

It is clear that $G_\infty(a,\tau,v_{max},l,1)$ is equal to the maximum number of possible velocities at any given time $k\tau$ multiplied by the maximum number of possible positions at that time. Since at each timestep the change in velocity is $a\tau$, $-a\tau$, or 0, the number of possible velocities is at most $\frac{2v_{max}}{a\tau} + 1$. To see that the number of possible positions at a given velocity is at most $\frac{l}{a\tau^2} + 1$, let v_k denote the velocity and x_k the position at timestep k for all k. Then $x_{k+1} = v_k\tau + \sigma(k)\frac{a\tau^2}{2}$, where $\sigma(k) \in \{-1,0,1\}$. *Wlog*, let $v_0 = 0$ and $x_0 = 0$. Since $v_k = c_k a\tau$ for some integer c_k, by using induction we can show that

$$\begin{aligned} x_k &= \frac{(2\Upsilon_k+1)a\tau^2}{2} && \text{if } c_k \text{ odd} \\ x_k &= \frac{2\Upsilon_k a\tau^2}{2} && \text{if } c_k \text{ even,} \end{aligned} \tag{10}$$

where Υ_k is another integer. It follows directly from (9) and (10) that

$$G_\infty(a,\tau,v_{max},l,d) = \left(\left(\frac{2v_{max}}{a\tau}+1\right)\left(\frac{l}{a\tau^2}+1\right)\right)^d. \tag{11}$$

Hence, in a bounded workspace with velocity limits, a polynomial-sized grid suffices to obtain an approximate optimal safe solution. It is easy to see that this polynomial-sized grid can be searched for the optimal (a,τ)-grid-bang path, while avoiding obstacles as prescribed by a safety function δ'_v, in polynomial time. We formalize this claim below:

Theorem: *Given acceleration bounds a, velocity bounds v_{max}, environment diameter l, and positive scalars ϵ, c_0, and c_1, the (a,τ)-grid with τ chosen to satisfy (8) has polynomial size. In addition given obstacles $\mathcal{E}$, start* $\mathbf{s}$, *and goal* $\mathbf{g}$, *a minimal-time (a,τ)-grid-bang, δ'_v-safe trajectory Γ_q from* $\mathbf{s}$ *to* $\mathbf{g}$ *can be computed in polynomial time.*

Note that the computed trajectory Γ_q satisfies the time approximation $T_q \le (1+\epsilon)T$, in addition to respecting the kinodynamic constraints and being δ'_v-safe.

5 Implementation and Experiments

To get some idea for how our algorithm might perform in practice, we have completed a preliminary implementation for the two dimensional case. The implementation runs on a Symbolics 3620 in Zetalisp. Currently, the planner implements collision-avoidance but not δ_v-safety; that is, the paths found are collision-free but not necessarily δ_v-safe. We intend to implement δ_v-safety by the final version of this paper. Since δ_v-safety reduces the search space, this may result in a performance improvement.

An example of a solution found by our planner is shown in fig. 3. The start position is at lower left, and the start velocity is in the y direction. The goal, at the upper right, has

Figure 7: This figure shows the commanded accelerations for the solution in fig. 3. Eg., at the start configuration, the commanded acceleration for the first time-step is a bang in the $+x$ direction. The large "diagonal" accelerations correspond to simultaneous ∞-norm bangs in the $-x$ and y or $-x$ and $-y$ directions. Thus this figure represents the "plan" of commanded forces.

zero velocity. The small circles indicate the position component of the planned trajectory at each time step. The "lines" or "spikes" attached to each circle indicate the velocity at that position.

Several points are worthy of comment. See fig. 7, which shows the commanded accelerations for the solution in fig. 3. Eg., at the start configuration, the commanded acceleration for the first time-step is a bang in the $+x$ direction. The large "diagonal" accelerations correspond to simultaneous ∞-norm bangs in the $-x$ and y or $-x$ and $-y$ directions. Thus this figure represents the "plan" of commanded forces. At first glance, the plan may appear to exhibit chattering. However, consider the plan as two simultaneous controls in x and y (that is, consider the projections of the commanded accelerations). From this viewpoint, viewing the x and y controls independently, it is clear that the plan is actually bang-bang in x, and bang-bang in y independently. That is, in x, the plan starts out with a full positive acceleration. Roughly half-way along, it switches to a full negative acceleration. However, in y, at first here is no y acceleration. Intuitively, the reason for this may be that a y acceleration would cause a collision. After the obstacle near the start is cleared, the plan accelerates full positively in y, followed by a full negative acceleration in y (and a small coast at the end). At this point, however, we cannot say anything more precise than that the plan has (essentially) a global bang-bang-like structure in the independent x, y controls.

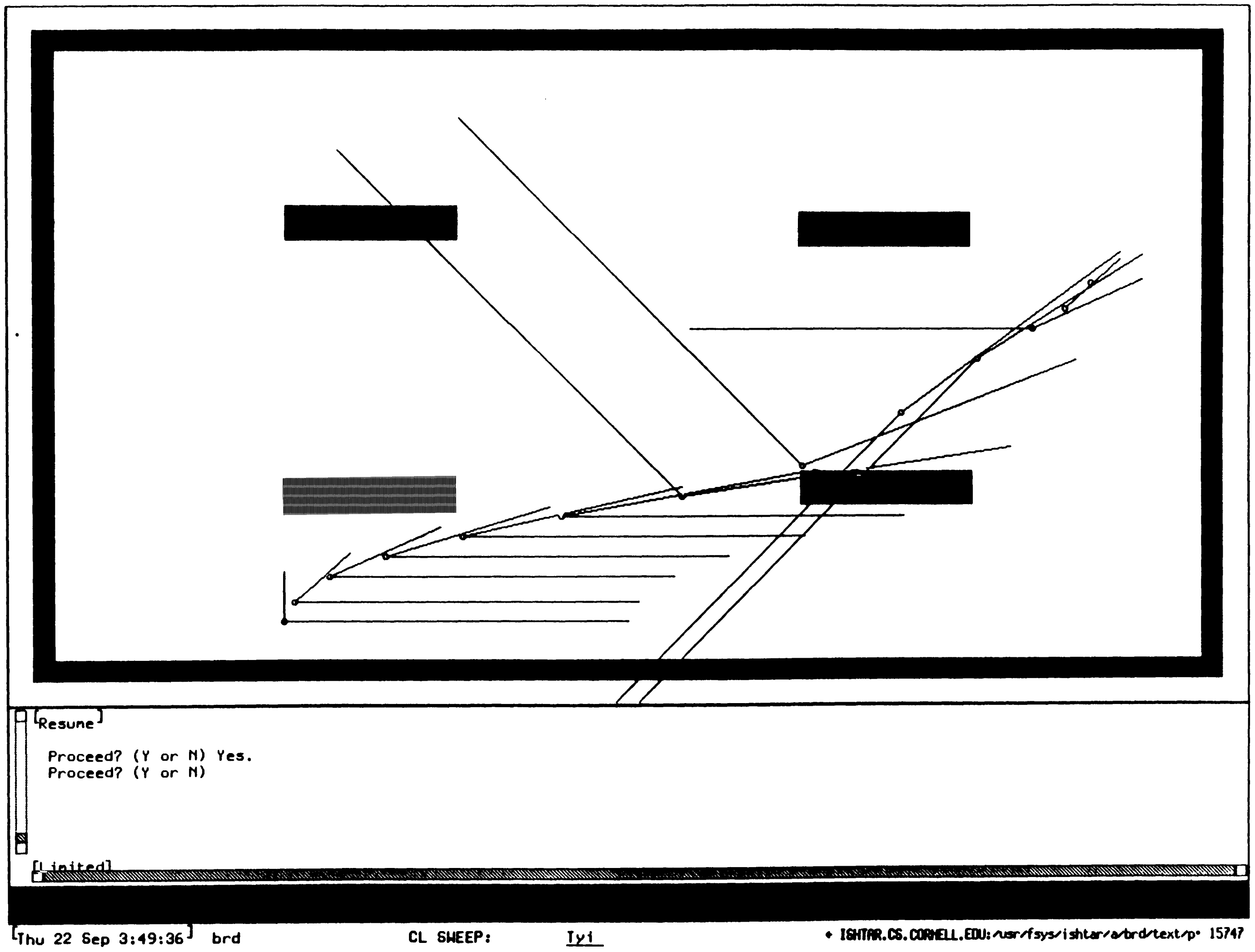
Our planner is currently implemented as a breadth-first search, with no heuristics to prune the search space. The solution in figs. 3-7 took several hours to generate. It would be of great interest to reimplement the planner using A^* search, assuming that appropriate estimation functions could be found. We hope that such techniques can be used to improve the performance of the algorithm in practice. Much work remains in tuning the algorithm; we have just taken the first step.

In addition, in the future, we intend to explore extending our algorithm to the case of rotational dynamics. A practical, provably-good planner for this case would clearly be of great interest.

6 Conclusions

In this paper we described the first implementation of a polynomial-time, provably good approximation algorithm for kinodynamic planning. We feel that kinodynamic planning represents a new direction in algorithmic motion planning, and expect to see much progress in this area.

There are many directions for future research:


Resune
Proceed? (Y or N) Yes.
Proceed? (Y or N)
[Linited]
Thu 22 Sep 3:49:36 brd
CL SWEEP:
Iyi
* ISHTAR.CS.CORNELL.EDU:/usr/fsys/ishtar/a/brd/text/p* 15747

1. The complexity of our algorithm can probably be improved.
2. Other search algorithms, such as A^*, may be employed in place of a breadth-first search.
3. Precise lower bounds for kinodynamic planning should be established (especially in the 2D case).
4. Exact algorithms should be explored.
5. We conjecture that if contact is allowed (rather than δ_v-safety) then the complexity of the problem increases considerably. More specifically, one can imagine three related kinodynamic planning problems:

   (a) The first is explored in this paper, where the robot must avoid obstacles by a speed-dependent safety margin.

   (b) A second problem might be likened to figure skating: forbidden regions are marked out in the plane (the "ice"), and a path with velocity-dependent nonholonomic constraints must be synthesized. The "obstacles" may be grazed but not crossed. However, the forbidden regions exert no reaction forces on the robot, even when in contact. This second problem corresponds to theoretical "true" optimality.

   (c) One can also imagine a third problem in which the reaction forces (impact, constraint forces, and friction) of the obstacle surfaces are taken into account.

   Finally, one may consider the optimization version of each of these problems. Note that while the theoretical formulation of the "figure skating" problem is quite clean, it may be rather far from practical interest.
6. It would be interesting to extend our approach to 2-norm velocity and acceleration bounds, and to manipulator systems with full rotational dynamics. For example, one might consider the rigid body dynamics of a planar polygon or a two link planar manipulator. Finding near-optimal kinodynamic solutions in these cases would be of great interest.

In addition, there is a great deal of interesting heuristic and experimental work to be done, in reducing these algorithms to practice. Computational kinodynamics seems a particularly fruitful area in which to pursue fast, provably good approximation algorithms, since while the problems are of considerable intrinsic interest, exact solutions may well be intractable. Finally, since the problem has an optimization flavor, the algorithms and proof techniques draw on several branches of computer science and robotics.

A Proofs

A.1 Details and Lemmas

We now give the proofs, so that the reader may check our argument if desired. For more details, see [CDRX], where full proofs will appear.

Lemma A.1 *If* $\mathbf{p}$ *is traversed in time* T_r *by a trajectory* Γ_r *under acceleration bound* a, *then there exists some* Γ'_r *that also traverses* $\mathbf{p}$ *with acceleration bounds* $\frac{a}{(1+\epsilon)^2}$ *in time* $T_r(1+\epsilon)$.

Proof: Given $\Gamma_r = (\mathbf{p}_r, \dot{\mathbf{p}}_r)$, we construct Γ'_r. Let $\zeta = \frac{1}{(1+\epsilon)}$ and let $\Gamma_r{}'$ be given by

$$\Gamma_r{}'(t) = \Big(\mathbf{p}_r(\zeta t), \zeta\dot{\mathbf{p}}_r(\zeta t)\Big). \tag{12}$$

Then the proof follows by checking position boundary conditions, and differentiating to obtain the acceleration bound. □

Lemma A.2 (The Tracking Lemma) *Suppose a trajectory* Γ_r *respects acceleration bounds* $\frac{a}{(1+\epsilon)^2}$ *and takes time* T_r. *Then in the absence of obstacles*

(a) for any positive η_x, η_v, *there exists a time spacing* τ *and an* (a,τ)*-grid-bang trajectory* Γ_q *with bounded acceleration* a *that approximately tracks* Γ_r *to tolerance* (η_x, η_v) *in time* T_r.

(b) Moreover, τ *is polynomial in* ϵ, η_x, *and* η_v. *Specifically, when* $0 < \epsilon \leq 1$, τ *can be chosen as:*

$$\tau \leq \min\left(\sqrt{\frac{\eta_x\epsilon}{17a}}, \frac{\eta_v\epsilon}{12a}\right). \tag{13}$$

Proof: We first show that given $\epsilon > 0$ we can find an integer N such that for any trajectory Γ_r respecting acceleration bounds $\frac{a}{(1+\epsilon)^2}$ and its running time T_r the following holds: for any time-spacing $\tau > 0$ there exists an (a,τ)-grid-bang trajectory Γ_q such that for all integers k such that $0 \leq kN\tau \leq T_r$,

$$\begin{aligned} |\mathbf{p}_q(kN\tau) - \mathbf{p}_r(kN\tau)|_\infty &\leq \tfrac{a\tau^2}{2} \\ |\dot{\mathbf{p}}_q(kN\tau) - \dot{\mathbf{p}}_r(kN\tau)|_\infty &\leq \tfrac{a\tau}{2}. \end{aligned} \tag{14}$$

Note that if $kN\tau > T_r$ the "for all" condition is vacuously true.

Since we are using the L_∞-norm, it is sufficient to show (14) for one-dimensional C. (For d dimensions we just take the largest $N_i, 0 < i \leq d$.) To make the proof more readable, we introduce a less cumbersome notation: for a map p or Γ to C, TC, or $\Re$, we denote its value at $kN\tau$ by $p^{(k)}$ or $\Gamma^{(k)}$, etc.; we denote by $\Delta p^{(k)}$ the quantity $p^{(k+1)} - p^{(k)}$.

Note that the dependence of $p^{(k)}$, $\Gamma^{(k)}$, etc. on τ is not apparent in this notation but will be obvious in the context of the proof.

The proof of (14) is by induction on k. We find a sufficiently large N that is independent of k and τ during a construction in the induction step. The $k = 0$ base case is trivial because Γ_r begins on a gridpoint by hypothesis.

Consider the induction step *for an arbitrary* $\tau > 0$. If $(k+1)N\tau > T_r$, then the induction step holds trivially. If $(k+1)N\tau \leq T_r$, then $kN\tau < T_r$, and the induction hypothesis states that there exists some (a,τ)-trajectory $\Gamma_q^{(k)}$ such that for $j = 0, \ldots, k$

$$\begin{aligned} |p_q^{(j)} - p_r^{(j)}| &\leq \tfrac{a\tau^2}{2} \\ |\dot{p}_q^{(j)} - \dot{p}_r^{(j)}| &\leq \tfrac{a\tau}{2}. \end{aligned} \tag{15}$$

We show that in this case there is an (a,τ)-bang extension $\gamma^{(k)}$ to $\Gamma^{(k)}$ so that under $\Gamma^{(k+1)} = \Gamma^{(k)} * \gamma^{(k)}$, (15) holds with $j = k+1$ provided that N is sufficiently large and $(k+1)N\tau \leq T_r$. (Here "$*$" denotes trajectory composition, which is similar to path composition.)

Assume that $N \geq \frac{1+\epsilon}{\epsilon}$ and let $\tilde{a} = \frac{a}{(1+\epsilon)^2}$ and $t_k = \frac{|\Delta\dot{p}_r^{(k)}|}{\tilde{a}}$. Then, we can choose an integer $\tilde{b}_k$ such that $|\tilde{b}_k| < N$, and $|\dot{p}_q^{(k)} + a\tilde{b}_k\tau - \dot{p}_r^{(k+1)}| \leq \frac{a\tau}{2}$. We set $\dot{p}_q^{(k+1)} = \dot{p}_q^{(k)} + a\tilde{b}_k\tau$. We can then construct the set $\mathcal{Q}_k$ of possible $\Delta p_q^{(k)}$ for this choice of $\dot{p}_q^{(k+1)}$; that is, we find all possible $\Delta p_q^{(k)}$ under (a,τ)-bang extensions $\gamma^{(k)}$ of $\Gamma^{(k)}$ such that $\dot{p}_q^{(k+1)} = \dot{p}_q^{(k)} + a\tilde{b}_k\tau$. By considering $\dot{p}_q(t)$ and recalling that $p_q(t)$ is its integral for t between $kN\tau$ and $(k+1)N\tau$ under the above constraints, we conclude first, that $\max(\mathcal{Q}_k) = \min(\mathcal{Q}_k) + \beta_k a\tau^2$ for some integer β_k, and second, that $u \in \mathcal{Q}_k$ if and only if $u = \min(\mathcal{Q}_k) + na\tau^2$ for some integer $n \in [0, \beta_k]$. Hence, we say that the suitable (a,τ)-bang extensions $\gamma^{(k)}$ of $\Gamma^{(k)}$ achieve the full range of $[\min(\mathcal{Q}_k), \max(\mathcal{Q}_k)]$ with grid spacing $a\tau^2$. Now, in a corresponding manner, we define $\mathcal{R}_k$ to be the (infinite) set of all possible $\Delta p_r^{(k)}$, for a given $\dot{p}_r^{(k)}$ and $\dot{p}_r^{(k+1)}$.

Therefore, we must show that for sufficiently large N, with $\dot{p}_q^{(k+1)}$ given by the above,

$$\begin{aligned} \min(\mathcal{Q}_k) &\leq \inf(\mathcal{R}_k) - \tfrac{a\tau^2}{2} \\ \max(\mathcal{Q}_k) &\geq \sup(\mathcal{R}_k) + \tfrac{a\tau^2}{2}. \end{aligned} \tag{16}$$

We describe how to choose N sufficiently large to ensure the max-sup inequality holds; the argument for the min-inf case is similar and yields the same N. The $\sup(\mathcal{R}_k)$ for a given $\Delta\dot{p}_r^{(k)}$ arises when Γ_r accelerates fully in the positive direction for as long as possible with the restriction of achieving $\Delta\dot{p}_r^{(k)}$. It is easy to see that in the worst case for a given $\Delta\dot{p}_r^{(k)}$

$$\begin{aligned} \dot{p}_r^{(k)} - \tfrac{a}{2\tau} &\leq \dot{p}_q^{(k)} < \dot{p}_r^{(k)}, \quad \text{and} \\ \dot{p}_r^{(k+1)} - \tfrac{a}{2\tau} &\leq \dot{p}_q^{(k+1)} < \dot{p}_r^{(k+1)}. \end{aligned} \tag{17}$$

If $\Delta\dot{p}_r^{(k)} \geq 0$, then Γ_r accelerates full-positively for the interval $(kN\tau, kN\tau + \frac{t_k + N\tau}{2})$ and full-negatively for the interval $(kN\tau + \frac{t_k+N\tau}{2}, (k+1)N\tau)$; otherwise, Γ_r accelerates full-positively for the interval $(kN\tau, kN\tau + \frac{N\tau - t_k}{2})$ and full-negatively for the interval $(kN\tau + \frac{N\tau - t_k}{2}, (k+1)N\tau)$. Thus, for a given non-negative $\Delta\dot{p}_r^{(k)}$

$$\begin{aligned} \sup(\mathcal{R}_k) &= \dot{p}_r^{(k)}N\tau + \tfrac{\tilde{a}t_k^2}{2} + \tfrac{\tilde{a}(N\tau - t_k)^2}{4} \\ &\quad + \tilde{a}t_k(N\tau - t_k) \end{aligned} \tag{18}$$

When $\Delta\dot{p}_r^{(k)}$ is negative, eqs. (18-21) and (23) are very similar; here, we assume the non-negative case throughout. Let $b_k = |\tilde{b}_k|$. Then, similarly to the above, we obtain

$$\begin{aligned} \max(\mathcal{Q}_k) &\geq \dot{p}_q^{(k)}N\tau + \tfrac{ab_k^2\tau^2}{2} + \tfrac{a\tau^2(N-b_k)^2}{4} \\ &\quad + a\tau^2 b_k(N - b_k) - \tfrac{a\tau^2}{2} \end{aligned} \tag{19}$$

where the "$-\frac{a\tau^2}{2}$" is due to the possibility of $N - b_k$ being odd. Combining common terms in N and b_k, we get

$$\begin{aligned} \max(\mathcal{Q}_k) &\geq \dot{p}_q^{(k)}N\tau + \tfrac{aN^2\tau^2}{4} + \tfrac{ab_kN\tau^2}{2} \\ &\quad - \tfrac{ab_k^2\tau^2}{4} - \tfrac{a\tau^2}{2} \end{aligned} \tag{20}$$

Since $b_k = \lfloor \frac{t_k}{\tau(1+\epsilon)^2} \rfloor$ or $b_k = \lceil \frac{t_k}{\tau(1+\epsilon)^2} \rceil$, in (20) we must choose the b_k that minimizes $\max(\mathcal{Q}_k)$. Now we consider the various possible values of b_k corresponding to different possible velocities at times $kN\tau$ and $(k+1)N\tau$. Using the RHS of 20, let us define the function $\Phi_{N,\tau,k} : \Re \rightarrow \Re$ so that $\Phi_{N,\tau,k}(b_k)$ is a lower bound on $\max(\mathcal{Q}_k)$ for $b_k \geq 0$:

$$\begin{aligned} \Phi_{N,\tau,k}(x) &= \dot{p}_q^{(k)}N\tau + \tfrac{aN^2\tau^2}{4} + \tfrac{aN\tau^2 x}{2} \\ &\quad - \tfrac{a\tau^2 x^2}{4} - \tfrac{a\tau^2}{2} \end{aligned} \tag{21}$$

Since the quadratic terms of $\Phi_{N,\tau,k}(x)$ have negative sign and $\Phi_{N,\tau,k}$ is continuous, if $\Phi_{N,\tau,k}(\lfloor \frac{t_k}{\tau(1+\epsilon)^2} \rfloor) \leq \Phi_{N,\tau,k}(\lceil \frac{t}{\tau\, 1+\epsilon} \rceil)$, then $\Phi_{N,\tau,k}(\frac{t}{\tau\, 1+\epsilon} - 1) < \Phi_{N,\tau,k}(\lfloor \frac{t_k}{\tau(1+\epsilon)^2} \rfloor)$, and if $\Phi_{N,\tau,k}(\lfloor \frac{t_k}{\tau(1+\epsilon)^2} \rfloor) \geq \Phi_{N,\tau,k}(\lceil \frac{t_k}{\tau(1+\epsilon)^2} \rceil)$, then $\Phi_{N,\tau,k}(\frac{t}{\tau(1+\epsilon)^2} + 1) < \Phi_{N,\tau,k}(\lceil \frac{t}{\tau(1+\epsilon)^2} \rceil)$. Thus, the following is a sufficient condition for the max case of (16):

$$\Phi_{N,\tau,k}\left(\frac{t_k}{\tau(1+\epsilon)^2} \pm 1\right) - \sup(\mathcal{R}_k) - \frac{a\tau^2}{2} \geq 0. \tag{22}$$

We now apply (22) to determine how large N must be. Let $\alpha = t_k/\tau$ and $\zeta = \frac{1}{1+\epsilon}$. Then, $t_k = \alpha\tau$ and $\frac{t_k}{\tau(1+\epsilon)^2} \pm 1 = \alpha\zeta^2 \pm 1$. Using (18) and (20) and doing some manipulation, we obtain:

$$\begin{aligned} &\Phi_{N,\tau,k}(\frac{t_k}{\tau(1+\epsilon)^2} \pm 1) - \sup(\mathcal{R}_k) - \frac{a\tau^2}{2} \geq \\ &\tfrac{-aN\tau^2}{2} + \tfrac{(1-\zeta^2)aN^2\tau^2}{4} \pm \tfrac{aN\tau^2}{2} \\ &+ \tfrac{\zeta^2(1-\zeta^2)a\alpha^2\tau^2}{4} \mp \tfrac{\zeta^2 a\alpha\tau^2}{2} - \tfrac{5a\tau^2}{4} \end{aligned} \tag{23}$$

After simplification, we get for both $\Delta\dot{p}_r^{(k)} \geq 0$ and $\Delta\dot{p}_r^{(k)} \leq 0$:

$$\begin{aligned}
&\Phi_{N,\tau,k}\left(\frac{t_k}{\tau(1+\epsilon)^2} \pm 1\right) - \sup(\mathcal{R}_k) - \frac{a\tau^2}{2} \geq \\
&\frac{a\tau^2}{4}\Big[(1-\zeta^2)N^2 - 4N \\
&\qquad + \zeta^2(1-\zeta^2)\alpha^2 - 2\zeta^2\alpha - 5\Big].
\end{aligned} \tag{24}$$

Since $0 \leq \alpha \leq N$, a sufficient condition for the right-hand side of (17) to be non-negative is

$$N^2(1-\zeta^2) - 6N - 5 \geq 0. \tag{25}$$

Choosing N to be positive, we see that this is guaranteed if

$$N \geq \frac{6+\sqrt{36+20(1-\zeta^2)}}{2(1-\zeta^2)}. \tag{26}$$

For all $\epsilon > 0$, (26) is implied by the condition $N \geq 7(1+\frac{1}{\epsilon^2+2\epsilon})$. Since for $0 < \epsilon \leq 1$, $\frac{3}{2\epsilon} \geq 1+\frac{1}{\epsilon^2+2\epsilon}$, a choice of $N \geq \frac{21}{2\epsilon}$ thus implies (26) and therefore (25) and (16). Therefore, because τ is arbitrary, we have shown that $N \geq \frac{21}{2\epsilon}$ is sufficiently large enough for the induction step to go through—that is, if $N \geq \frac{21}{2\epsilon}$, then (15) holds with $j = k+1$ for arbitrary τ such that $(k+1)N\tau \leq T_r$.

Now that we have shown an upper bound for how large N must be in (14) independent of τ, we can choose τ such that $|p_q(t) - p_r(t)| \leq \eta_x$, and $|\dot{p}_q(t) - \dot{p}_r(t)| \leq \eta_v$. Since for all k, $|p_q(kN\tau) - p_r(kN\tau)| \leq \frac{1}{2}a\tau^2$ and $|\dot{p}_q(kN\tau) - \dot{p}_r(kN\tau)| \leq \frac{1}{2}a\tau$, and for all t, $|\ddot{p}_r(t) - \ddot{p}_q(t)| \leq 2a$, we can simply choose τ such that $\frac{1}{2}(a\tau^2 + aN\tau^2) + aN\tau^2 < \eta_x$. Thus, for the position case we require that

$$\tau^2 < \frac{2\eta_x}{a(3N+1)}. \tag{27}$$

Since $|\dot{p}_q(t) - \dot{p}_r(t)| < a(N+1)\tau$ for $0 \leq t \leq T_r$ or else $|\dot{p}_q^{(k)} - \dot{p}_r^{(k)}| \leq \frac{1}{2}a\tau$ gets violated, for the velocity case we require that

$$\tau \leq \frac{\eta_v}{a}\left(\frac{1}{N+1}\right). \tag{28}$$

Substituting our bound for N into (27) and (28) taking the minimum, we obtain the bound in part (b) of the lemma (eq. (13)). □

Lemma A.3 (The Safe Tracking Lemma) *Suppose that δ_v is specified by c_0 and c_1 and $0 < \epsilon < 1$, and let $\delta_v{}' = (1-\epsilon)\delta_v$. Then for a given acceleration bound a there exists a tolerance (η_x, η_v) such that for any trajectories Γ_r and Γ_r' as above (12), the following hold:*

(a) If Γ_q tracks Γ_r' to tolerance (η_x, η_v), then the δ_v'-tube induced by Γ_q lies within the δ_v-tube induced by Γ_r.

(b) Furthermore, for any positive β, the following choices suffice:

$$\begin{aligned} \eta_v &\leq \tfrac{c_0\epsilon}{c_1(1-\epsilon)+\beta} \\ \eta_x &\leq \beta\eta_v. \end{aligned} \tag{29}$$

Proof: We find positive real numbers η_x and η_v such that if Γ_q tracks Γ_r' to tolerance (η_x, η_v), then the δ_v'-tube induced by Γ_q lies entirely inside the δ_v-tube induced by Γ_r. Henceforth, let $c_0' = (1-\epsilon)c_0$ and $c_1' = (1-\epsilon)c_1$.

Suppose $\mathbf{x} \in C$ lies inside the δ_v'-tube induced by Γ_q. Then for some $t_x \in [0, (1+\epsilon)T_r]$, $|\mathbf{x} - \mathbf{p}_q(t_x)| < c_0' + c_1'|\dot{\mathbf{p}}_q(t_x)|$. If Γ_q tracks Γ_r' to tolerance (η_x, η_v), then $\mathbf{p}_q(t_x) \in B_{\eta_x}(p_r'(t_x))$ and $\dot{\mathbf{p}}_q(t_x) \in B_{\eta_v}(\dot{\mathbf{p}}_r'(t_x))$. ($B_\eta(\mathbf{p})$ denotes the cl[illegible] η-ball around $\mathbf{p}$ in an arbitrary metric.) Therefore, $|\mathbf{x} - \mathbf{p}_r'(t_x)| \leq |\mathbf{x} - \mathbf{p}_q(t_x)| + \eta_x$ and $|\dot{\mathbf{p}}_q(t_x)| \leq |\dot{\mathbf{p}}_r'(t_x)| + \eta_v$. Since $\mathbf{p}_r'(t_x) = \mathbf{p}_r(\zeta t_x)$ and $\dot{\mathbf{p}}_r'(t_x) = \zeta\dot{\mathbf{p}}_r(\zeta t_x)$, $|\mathbf{x} - \mathbf{p}_r(\zeta t_x)| < c_0' + \eta_x + c_1'(|\dot{\mathbf{p}}_r(\zeta t_x)| + \eta_v)$. Therefore, the condition

$$c_0' + \eta_x + c_1'(|\dot{\mathbf{p}}_r(\zeta t_x)|_\infty + \eta_v) \leq c_0 + c_1|\dot{\mathbf{p}}_r(\zeta t_x)|_\infty \tag{30}$$

implies that $|\mathbf{x} - \mathbf{p}_r(\zeta t_x)| < c_0 + c_1|\dot{\mathbf{p}}_r(\zeta t_x)|$ for some $t_x \in [0, (1+\epsilon)T_r]$. Simplifying, we find that a sufficient condition for (30) is

$$\eta_x + (1-\epsilon)c_1\eta_v \leq \epsilon c_0. \tag{31}$$

Thus, (31) implies that if Γ_q tracks Γ_r' to tolerance (η_x, η_v), then the δ_v'-tube induced by Γ_q lies entirely inside the δ_v-tube induced by Γ_r. Both parts of the lemma are obtained by letting $\eta_x = \beta\eta_v$ and observing (31). □

Recall that by lemma A.2, τ is polynomially dependent on η_x and η_v. Applying lemmas A.2 and A.3 and choosing β to maximize the upper bound on τ yields the following theorem:

Theorem A.4 *Given acceleration bounds a, obstacles $\mathcal{E}$, and positive scalars $\epsilon \leq 1$, c_0, and c_1, for any $\delta_v(c_0, c_1)$-safe trajectory taking time T, there exists a time spacing τ and an (a, τ)-grid-bang, δ_v'-safe trajectory Γ_q taking time at most $(1+\epsilon)T$. In particular, the following choice of τ suffices:*

$$\tau \leq \frac{2c_0\epsilon^2}{12ac_1 + \sqrt{144a^2c_1^2 + 68ac_0}}. \tag{32}$$

Proof: Suppose Γ_r is a $\delta_v(c_0, c_1)$-safe trajectory taking time T_r obeying acceleration bound a. Applying lemma A.1, the trajectory Γ'_r as given in (12) respects $\frac{a}{(1+\epsilon)^2}$ and traverses Γ_r in time $(1+\epsilon)T_r$. Then by lemma A.3 the choice of a tolerance (η_x, η_v) given in (29) ensures that if a trajectory Γ_q approximately tracks Γ'_r to tolerance (η_x, η_v), then the δ'_v-tube induced by Γ_q lies entirely inside the δ_v-tube induced by Γ_r. Since the δ_v-tube induced by Γ_r intersects no obstacles in $\mathcal{E}$, Γ_q is therefore δ'_v-safe. Given the tolerance (η_x, η_v), by lemma A.2 there is a time-spacing τ such that some(a, τ)-grid-bang trajectory approximately tracks Γ'_r to tolerance (η_x, η_v).

To get the desired bounds, we must choose β so that using (32) yields a maximal τ as given by (13). Let us therefore define for $\beta \geq 0$

$$\begin{aligned} \tau_x(\beta) &= \sqrt{\frac{c_0\epsilon^2\beta}{204a(c_1(1-\epsilon)+\beta)}} \\ \tau_v(\beta) &= \frac{c_0\epsilon^2}{17a(c_1(1-\epsilon)+\beta)} \\ \tau(\beta) &= \min(\tau_x(\beta), \tau_v(\beta)). \end{aligned} \tag{33}$$

By inspection, $\tau_x(0) < \tau_v(0)$, τ_x is monotonically increasing, and τ_v is monotonically decreasing. Thus, $\tau(\beta)$ is maximized when $\tau_x(\beta) = \tau_v(\beta)$. Requiring β to be positive and doing a little computation, we find that $\tau(\beta)$ is maximized when

$$\beta = \frac{\sqrt{144a^2c_1^2(1-\epsilon)^2 + 17ac_0\epsilon^2} - 144ac_1(1-\epsilon)}{24a}. \tag{34}$$

Applying either τ_x or τ_v to this β yields the desired τ in (32). □

Using our bound on the number of gridpoints (11), we hence conclude that in a bounded workspace with velocity limits, a polynomial-sized grid suffices to obtain an approximate optimal safe solution. It is easy to see that this polynomial-sized grid can be searched for the optimal (a, τ)-grid-bang path, while avoiding obstacles as prescribed by a safety function δ'_v, in polynomial time. We formalize this claim below:

Corollary A.5 *Given acceleration bounds a, velocity bounds v_{max}, environment diameter l, and positive scalars ϵ, c_0, and c_1, the (a, τ)-grid with τ chosen to satisfy (32) has polynomial size. In addition given obstacles $\mathcal{E}$, start* $\mathbf{s}$*, and goal* $\mathbf{g}$*, a minimal-time (a, τ)-grid-bang, δ'_v-safe trajectory Γ_q from* $\mathbf{s}$ *to* $\mathbf{g}$ *can be computed in polynomial time.*

Note that the computed trajectory Γ_q satisfies the time approximation $T_q \leq (1+\epsilon)T$, in addition to respecting the kinodynamic constraints and being δ'_v-safe.

B References

[BDG] Bobrow, J., Dubowsky, S., and Gibson, J. *Time-Optimal Control of Robot Manipulators,* Int. Jour. Robotics Research 4 (3) (1985).

[Bra] **Brady, M. et. al.** (eds). *Robot Motion: Planning and Control.,* Cambridge, Mass.: MIT Press. (1982).

[CDRX] **Canny, J., Donald, B., Reif J., and Xavier, P.** Submitted to: *On The Complexity of Kinodynamic Planning,* 29^{th} Symposium on the Foundations of Computer Science, White Plains NY (1988).

[CR] **Canny J., and Reif., J** *New Lower Bound Techniques for Robot Motion Planning,* FOCS, Los Angeles (1987).

[FW] **Fortune, S., and Wilfong, G.** *Planning Constrained Motion,* STOC, Chicago (1988).

[Hol] **Hollerbach, J. M.** *Dynamic Scaling of Manipulator Trajectories,* MIT A.I. Lab Memo 700 (1983).

[LoP] **Lozano–Pérez, T.** *Spatial Planning: A Configuration Space Approach,* IEEE Trans. on Computers (C–32):108–120 (1983).

[O] **Ó'Dúnlaing** *Motion Planning with Inertial Constraints,* NYU Robotics Laboratory TR 230 (1986).

[Pap] **Papadimitriou, C.** *An Algorithm for Shortest Path Motion in Three Dimensions,* Information Proc. Letters, North Holland **20** pp. 259-263 (1985).

[PS] **Papadimitriou, C. and Steiglitz** *Combinatorial Optimization: Algorithms and Complexity,* Prentice-Hall, Englewood Cliffs, NY (1982).

[SH] **Sahar, G., and Hollerbach, J.** *Planning of Mimimum-Time Trajectories for Robot Arms,* IEEE Int. Conf. On Robotics and Automation, St. Louis (1985).

[Sch] **Schaettler, H.M.** *On The Optimality of Bang-Bang Trajectories in* $\Re^3$, Bull. AMS **16** (1). pp 113-6 (1987).

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

[SS1] **Sontag, E., and Sussmann, H** *Remarks on the Time-Optimal Control of Two-Link Manipulators,* Proc. 24^{th} Conf. on Decision and Control, Ft. Lauderdale (1985).

[SS2] **Sontag, E., and Sussmann, H** *Time-Optimal Control of Manipulators,* Dept. mathematics, Rutgers, New Brunswick, NJ (1986).

[W] **Wilfong, G.** *Motion Planning for an Autonomous Vehicle,* IEEE Int. Conf. On Robotics and Automation, Philadelphia (1988).

[Y] **Yap, C.**, "Algorithmic Motion Planning", in *Advances in Robotics: Volume 1*, edited by J. Schwartz and C. Yap, Lawrence Erlbaum Associates, 1986.