On the Complexity of Kinodynamic Planning

John Canny
Berkeley Computer Science Division

Bruce Donald§
Cornell Computer Science Department

John Reif
Duke Computer Science Department

Patrick Xavier§
Cornell Computer Science Department

Abstract: In robotics, kinodynamic planning attempts to solve a motion problem subject to simultaneous kinematic and dynamic constraints. 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. We consider the simplified case of a point mass under Newtonian mechanics, together with velocity and acceleration bounds. The point must be flown from a start to a goal, amidst polyhedral obstacles in 2D or 3D. While exact solutions to this problem are not known, we provide the first provably good approximation algorithm, and show that it runs in polynomial time.

1 Introduction

The *kinodynamic planning problem* 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 in the robotics community, 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. We consider the restricted situation of particle dynamics, and provide a provably good approximation algorithm for the 2- and 3-dimensional cases. 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. Finally, we believe that our algorithm is simple enough that it could be implemented, and might well extend to robot systems with full dynamics.

§Supported in part by the National Science Foundation and the Mathematical Sciences Institute.

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. How-

0272-5428/88/0000/0306$01.00

ever, 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

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

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

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

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.

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.

Finally, we define an *approximate optimal safe solution* Γ_q to be a kinodynamic solution which is "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.

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.

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 running 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 the approximation bound

$$t_q \leq (1+\epsilon)b \qquad (3)$$

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*

[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.

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 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 outdegree of each vertex is fixed at 3^d for dimension d. The algorithm starts at $\mathbf{s}$ 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 $\mathbf{s}$ 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:

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

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 $O(n^2 G + G \log G)$.

3.2 Details and Lemmas

We now describe the key lemmas in our argument. First, we develop some lemmas that concern the approximation of optimal trajectories in the absence of obstacles. Then we generalize these results to the δ_v-safe case, that is, for the case of optimal safe trajectories that respect a δ_v-safety tube.

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 $\mathbf{p}$ 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 $\mathbf{p}$, and Γ respects (1). First we must prove a not very difficult lemma showing that allowing a 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.

Lemma 3.1 *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

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

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

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.

Consider two trajectories $\Gamma_r, \Gamma_q : [0,b] \to TC$. 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, $|\mathbf{p}_q(t) - \mathbf{p}_r(t)|_\infty \le \eta_x$ and $|\dot{\mathbf{p}}_q(t) - \dot{\mathbf{p}}_r(t)|_\infty \le \eta_v$.

Lemma 3.2 (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* $0 < \epsilon \le 1$*,* τ *can be chosen as:*

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

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 $0 \le kN\tau \le T_r$,

$$\begin{aligned} |\mathbf{p}_q(kN\tau) - \mathbf{p}_r(kN\tau)|_\infty &\le \tfrac{a\tau^2}{2} \\ |\dot{\mathbf{p}}_q(kN\tau) - \dot{\mathbf{p}}_r(kN\tau)|_\infty &\le \tfrac{a\tau}{2}. \end{aligned} \tag{7}$$

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 (7) for one-dimensional C. (For d dimensions we just take the largest $N_i, 0 < i \le d$.) 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 (7) 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 $(k+1)N\tau \le T_r$, then $kN\tau < T_r$, and the induction hypothesis states that there exists some (a,τ)-trajectory $\Gamma_q^{(k)}$ such that for $j = 0, \ldots, k$

$$\begin{aligned} |p_q^{(j)} - p_r^{(j)}| &\le \tfrac{a\tau^2}{2} \\ |\dot{p}_q^{(j)} - \dot{p}_r^{(j)}| &\le \tfrac{a\tau}{2}. \end{aligned} \tag{8}$$

We show that in this case there is an (a,τ)-bang extension $\gamma^{(k)}$ to $\Gamma^{(k)}$ so that under $\Gamma^{(k+1)} = \Gamma^{(k)} * \gamma^{(k)}$, (8) holds with $j = k+1$ provided that N is sufficiently large and $(k+1)N\tau \le T_r$. (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,

$$\begin{array}{rcl} \min(\mathcal{Q}_k) & \leq & \inf(\mathcal{R}_k) - \frac{a\tau^2}{2} \\ \max(\mathcal{Q}_k) & \geq & \sup(\mathcal{R}_k) + \frac{a\tau^2}{2}. \end{array} \tag{9}$$

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 $\Delta \dot{p}_r^{(k)}$

$$\begin{array}{rcccl} \dot{p}_r^{(k)} - \frac{a}{2\tau} & \leq & \dot{p}_q^{(k)} & < & \dot{p}_r^{(k)}, \quad \text{and} \\ \dot{p}_r^{(k+1)} - \frac{a}{2\tau} & \leq & \dot{p}_q^{(k+1)} & < & \dot{p}_r^{(k+1)}. \end{array} \tag{10}$$

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 $\Delta \dot{p}_r^{(k)}$

$$\begin{array}{rcl} \sup(\mathcal{R}_k) & = & \dot{p}_r^{(k)} N\tau + \frac{\tilde{a}t_k^2}{2} + \frac{\tilde{a}(N\tau - t_k)^2}{4} \\ & & + \tilde{a}t_k(N\tau - t_k) \end{array} \tag{11}$$

When $\Delta \dot{p}_r^{(k)}$ is negative, eqs. (11-14) and (16) are very similar; here, we assume the non-negative case throughout. Let $b_k = |\tilde{b}_k|$. Then, similarly to the above, we obtain

$$\begin{array}{rcl} \max(\mathcal{Q}_k) & \geq & \dot{p}_q^{(k)} N\tau + \frac{ab_k^2\tau^2}{2} + \frac{a\tau^2(N-b_k)^2}{4} \\ & & + a\tau^2 b_k(N - b_k) - \frac{a\tau^2}{2} \end{array} \tag{12}$$

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

$$\begin{array}{rcl} \max(\mathcal{Q}_k) & \geq & \dot{p}_q^{(k)} N\tau + \frac{aN^2\tau^2}{4} + \frac{ab_k N\tau^2}{2} \\ & & - \frac{ab_k^2\tau^2}{4} - \frac{a\tau^2}{2} \end{array} \tag{13}$$

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 (13) 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 13, let us define the function $\Phi_{N,\tau,k} : \Re \to \Re$ so that $\Phi_{N,\tau,k}(b_k)$ is a lower bound on $\max(\mathcal{Q}_k)$ for $b_k \geq 0$:

$$\begin{array}{rcl} \Phi_{N,\tau,k}(x) & = & \dot{p}_q^{(k)} N\tau + \frac{aN^2\tau^2}{4} + \frac{aN\tau^2 x}{2} \\ & & - \frac{a\tau^2 x^2}{4} - \frac{a\tau^2}{2} \end{array} \tag{14}$$

Since the quadratic terms of $\Phi_{N,\tau,k}(x)$ have negative sign and $\Phi_{N,\tau,k}$ is continuous, if $\Phi_{N,\tau,k}(\lfloor \frac{t_k}{\tau(1+\epsilon)^2} \rfloor) \leq \Phi_{N,\tau,k}(\lceil \frac{t_k}{\tau(1+\epsilon)^2} \rceil)$, then $\Phi_{N,\tau,k}(\frac{t_k}{\tau(1+\epsilon)^2} - 1) < \Phi_{N,\tau,k}(\lfloor \frac{t_k}{\tau(1+\epsilon)^2} \rfloor)$, 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 $\Phi_{N,\tau,k}(\frac{t_k}{\tau(1+\epsilon)^2} + 1) < \Phi_{N,\tau,k}(\lceil \frac{t_k}{\tau(1+\epsilon)^2} \rceil)$. Thus, the following is a sufficient condition for the max case of (9):

$$\Phi_{N,\tau,k}\left(\frac{t_k}{\tau(1+\epsilon)^2} \pm 1\right) - \sup(\mathcal{R}_k) - \frac{a\tau^2}{2} \geq 0. \tag{15}$$

We now apply (15) 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 (11) and (13) and doing some manipulation, we obtain:

$$\Phi_{N,\tau,k}(\frac{t_k}{\tau(1+\epsilon)^2} \pm 1) - \sup(\mathcal{R}_k) - \frac{a\tau^2}{2} \geq \frac{-aN\tau^2}{2} + \frac{(1-\zeta^2)aN^2\tau^2}{4} \pm \frac{aN\tau^2}{2} + \frac{\zeta^2(1-\zeta^2)a\alpha^2\tau^2}{4} \mp \frac{\zeta^2 a\alpha\tau^2}{2} - \frac{5a\tau^2}{4} \quad (16)$$

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

$$\Phi_{N,\tau,k}\left(\frac{t_k}{\tau(1+\epsilon)^2} \pm 1\right) - \sup(\mathcal{R}_k) - \frac{a\tau^2}{2} \geq \frac{a\tau^2}{4}\Big[(1-\zeta^2)N^2 - 4N + \zeta^2(1-\zeta^2)\alpha^2 - 2\zeta^2\alpha - 5\Big]. \quad (17)$$

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

$$N^2(1-\zeta^2) - 6N - 5 \geq 0. \quad (18)$$

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

$$N \geq \frac{6 + \sqrt{36 + 20(1-\zeta^2)}}{2(1-\zeta^2)}. \quad (19)$$

For all $\epsilon > 0$, (19) 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 (19) and therefore (18) and (9). 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 (8) 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 (7) 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

$$\tau^2 < \frac{2\eta_x}{a(3N+1)}. \quad (20)$$

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

$$\tau \leq \frac{\eta_v}{a}(\frac{1}{N+1}). \quad (21)$$

Substituting our bound for N into (20) and (21) taking the minimum, we obtain the bound in part (b) of the lemma (eq. (6)). □

Lemma 3.2 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 satisfies (6), there exists an (a, τ)-grid-bang trajectory that "follows" such a Γ_r in a weaker sense. Specifically, as in lemma 3.1, let $\zeta = \frac{1}{(1+\epsilon)}$ and let Γ_r' be given by (5). If Γ_r is a-bounded, then Γ_r' is $\zeta^2 a$-bounded, and thus by lemma 3.2, there is an (a, τ)-grid-bang trajectory that approximately tracks Γ_r' to tolerance (η_x, η_v). It then follows from (5) 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 lemma 3.2 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' given by (5) must also be 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 δ_v' such that for all speeds y and some constant ϵ_v,

$$\delta_v'(y) \leq \delta_v(y) + \epsilon_v, \quad (22)$$

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.

Lemma 3.3 (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 (5), 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:

$$\begin{aligned} \eta_v &\leq \frac{c_0\epsilon}{c_1(1-\epsilon)+\beta} \\ \eta_x &\leq \beta\eta_v. \end{aligned} \tag{23}$$

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 closed 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

$$c'_0 + \eta_x + c'_1(|\dot{\mathbf{p}}_r(\zeta t_x)|_\infty + \eta_v) \leq c_0 + c_1|\dot{\mathbf{p}}_r(\zeta t_x)|_\infty \tag{24}$$

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 (24) is

$$\eta_x + (1-\epsilon)c_1\eta_v \leq \epsilon c_0. \tag{25}$$

Thus, (25) 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 (25). □

Recall that by lemma 3.2, τ is polynomially dependent on η_x and η_v. Applying lemmas 3.2 and 3.3 and choosing β to maximize the upper bound on τ yields the following theorem:

Theorem 3.4 *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:*

$$\tau \leq \frac{2c_0\epsilon^2}{12ac_1 + \sqrt{144a^2c_1^2 + 68ac_0}}. \tag{26}$$

Proof: Suppose Γ_r is a $\delta_v(c_0, c_1)$-safe trajectory taking time T_r obeying acceleration bound a. Applying lemma 3.1, the trajectory Γ'_r as given in (5) respects $\frac{a}{(1+\epsilon)^2}$ and traverses Γ_r in time $(1+\epsilon)T_r$. Then by lemma 3.3 the choice of a tolerance (η_x, η_v) given in (23) 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 3.2 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 (26) yields a maximal τ as given by (6). Let us therefore define for $\beta \geq 0$

$$\begin{aligned} \tau_x(\beta) &= \sqrt{\frac{c_0\epsilon^2\beta}{204a(c_1(1-\epsilon)+\beta)}} \\ \tau_v(\beta) &= \frac{c_0\epsilon^2}{17a(c_1(1-\epsilon)+\beta)} \\ \tau(\beta) &= \min(\tau_x(\beta), \tau_v(\beta)). \end{aligned} \tag{27}$$

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

$$\beta = \frac{\sqrt{144a^2c_1^2(1-\epsilon)^2 + 17ac_0\epsilon^2} - 144ac_1(1-\epsilon)}{24a}. \tag{28}$$

Applying either τ_x or τ_v to this β yields the desired τ in (26). □

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

$$G_\infty(a, \tau, v_{max}, l, d) = (G_\infty(a, \tau, v_{max}, l, 1))^d. \tag{29}$$

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

$$\begin{aligned} x_k &= \frac{(2\Upsilon_k+1)a\tau^2}{2} && \text{if } c_k \text{ odd} \\ x_k &= \frac{2\Upsilon_k a\tau^2}{2} && \text{if } c_k \text{ even,} \end{aligned} \tag{30}$$

where Υ_k is another integer. It follows directly from (29) and (30) that

$$G_\infty(a,\tau,v_{max},l,d)=\left(\left(\frac{2v_{max}}{a\tau}+1\right)\left(\frac{l}{a\tau^2}+1\right)\right)^d. \tag{31}$$

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:

Corollary 3.5 *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 (26) has polynomial size. In addition given obstacles $\mathcal{E}$, start* s, *and goal* g, *a minimal-time (a,τ)-grid-bang, δ'_v-safe trajectory Γ_q from* s *to* 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.

4 Conclusions

In this paper we described the first 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:

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.

5 Appendix: Some Details on the Algebraic Complexity

In the language of combinatorial optimization [PS], we wish to 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. Recall 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 $(a_{max}, v_{max}, c_1, \frac{1}{c_0})$.

All numbers in the input are taken to be rational. We have shown that the number of steps required in the algorithm is polynomial in the combinatorial complexity and $(\frac{1}{\epsilon})$. Since the algorithm requires a certain amount of numerical computation, we now show that the number of bits required for any number in any computation in the algorithm is also bounded polynomially and that thus our algorithm is fully polynomial in the algebraic complexity of the geometry as well.

From theorem 3.4 we an have upper bound on what τ can be in the discretization of the state-space. Examining (26), we see that it is easy to obtain an acceptable τ whose length is polynomial in the length of the input parameters and that is within a polynomial factor (in a, c_0, and c_1) of the optimal τ. Let $D = 144a^2c_1^2 + 68ac_0$. Then

$$\begin{aligned} \tau &= \frac{2c_0\epsilon^2}{12ac_1+144a^2c_1^2+68ac_0} && \text{if } D > 1, \\ \tau &= \frac{2c_0\epsilon^2}{12ac_1+1} && \text{if } D \leq 1. \end{aligned} \qquad (32)$$

It is clear from the development following the proof of theorem 3.4 that if the initial state, dynamics bounds, and timestep τ for $(\mathbf{a},\tau)$-grid-bang trajectory Γ_q are given in rational numbers and rational vectors, then at any time $n\tau$ (n a non-negative integer) the state $\Gamma_q(n\tau)$ is given by a rational vector. Furthermore, the number of bits needed to compute these states and single-step reachability between states in the absence of obstacles is polynomial in the number of bits in the rationals in the problem instance. The only other computations that we need to consider here are those needed to determine whether a particular $(\mathbf{a},\tau)$-bang at a given state $(\mathbf{p},\dot{\mathbf{p}})$ results in violating δ_v-safety during the current timetep. *Wlog* we consider the $\Re^3$ case, first reviewing the representation of convex polyhedral objects.

Obstacles are represented as intersections of closed half-spaces. The boundary plane of each half-space H is the kernel of an affine function f_H, and where a point $\mathbf{p}$ lies relative to H (interior, exterior, or on the boundary) can be determined from the sign of $f_H(\mathbf{p})$. Thus, if a polyhedron A is the intersection of the half-spaces described by the set of functions $\mathcal{A} = \{ f_0, \ldots, f_k \}$ where $f_i(\mathbf{p}) > 0$ determines that $\mathbf{p}$ is outside closed half-space H_i, then $\mathbf{p}$ is a point on the boundary of A if and only if for all $f_i \in \mathcal{A}$, $f_i(\mathbf{p}) \leq 0$, and for some $f_j \in \mathcal{A}$, $f_j(\mathbf{p}) = 0$. If $\mathbf{n}_i$ is a unit vector in the outward normal direction from the boundary plane of H_i, and $\mathbf{x}_i$ is any point on this boundary then we can use the function

$$f_i(\mathbf{p}) = \langle \mathbf{n}_i, \mathbf{p} \rangle - \langle \mathbf{n}_i, \mathbf{x}_i \rangle. \qquad (33)$$

Recall the description of the algorithm in section 3.1. We now describe how to check whether a particular $(\mathbf{a},\tau)$-bang from a state $(\mathbf{p},\dot{\mathbf{p}})$ would violate δ_v-safety with respect to a particular obstacle A during the execution of the command. To see the key observation, let safety margin δ be fixed, and define the set $B_\delta = \{\mathbf{y} \in \Re^3 \mid \ |\mathbf{y}|_\infty \leq \delta\}$. Then, staying δ-safe relative to A is equivalent to avoiding $A' = A \oplus B_\delta$, where "$\oplus$" denotes the Minkowski sum. A' is also polyhedral, and $|\mathcal{A}'(\delta)|$ has size linear in $|verts(A)|$, $|edges(A)|$, and $|\mathcal{A}|$. Using [LoP] it is clear that computing a half-space intersection description $\mathcal{A}'$ of A' can be done in time $O(n^2 \log n)$ in $\Re^3$ and $O(n)$ in $\Re^2$, where n is the number of vertices in A. By allowing δ to vary, we effectively lift the collision-avoidance problem from three dimensions to four.

In the general case δ is a positive affine function of $v = |\dot{\mathbf{p}}|_\infty$, and we can describe members H_i of $\mathcal{A}'$ with functions $f_i : \Re^4 \rightarrow \Re$ of the form

$$f_i(\mathbf{p}, v) = \langle \mathbf{n}_i, \mathbf{p} \rangle - \langle \mathbf{n}_i, \mathbf{x}_i + c_1 v \hat{\mathbf{q}} \rangle \qquad (34)$$

where $\hat{\mathbf{q}}$ is a unit ∞-norm vector in a grid-bang direction. A state $(\mathbf{p},\dot{\mathbf{p}})$ is δ_v-safe relative to A if and only if for all $f_i \in \mathcal{A}'$, $f_i(\mathbf{p}, |\dot{\mathbf{p}}|_\infty) > 0$;

$f_i(\mathbf{p}, v) = 0$ if and only if $(\mathbf{p}, v)$ lies on the boundary hyperplane. Because the components of $\mathbf{n}_i$ and $\mathbf{x}_i$ are rational and of polynomial length, it follows that the coefficients of $f_i(\mathbf{p}, v)$ must also be rational.

Define $\tilde{\mathcal{A}}$ to be the set of composite functions $\tilde{f}_i(t) = f_i(\mathbf{p}(t), |\dot{\mathbf{p}}(t)|_\infty)$. Since for any non-negative integer m $(\mathbf{p}(m\tau), \dot{\mathbf{p}}(m\tau))$ is a vector of rationals, under any $(\mathbf{a}, \tau)$-bang from $m\tau$ to $(m+1)\tau$ the components of $(\mathbf{p}(t), \dot{\mathbf{p}}(t))$ are quadratic functions in $(t - m\tau)$ with rational coefficients. Thus, the $\tilde{f}_i$ are quadratic in t. (Strictly speaking, for each $\tilde{f}_i$ in this set there are three sets of coefficients because dominating component of $\dot{\mathbf{p}}$ can change, but this is easily taken into account.) Therefore, checking for a δ_v-safety violation only requires deciding whether there is some $t_c \in [m\tau, (m+1)\tau]$ such that $\tilde{f}_i(t_c) \leq 0$ for all $\tilde{f}_i$ and $\tilde{f}_j(t_c) = 0$ for some $\tilde{f}_j$, where all $\tilde{f}_i \in \tilde{\mathcal{A}}$ are quadratic functions in t with rational coefficients. This checking clearly can be done in polynomial time and without any number exceeding a polynomial number of bits. Hence, our algorithm is polynomial in the algebraic complexity of the geometry.

By substituting the value for τ (32) into the bound on the grid size (31), we see that our algorithm is polynomial in the kinodynamic bounds $(a_{max}, v_{max}, c_1, \frac{1}{c_0})$, but not in the size of their bit encodings. Hence, the algorithm is pseudo-polynomial in the kinodynamic bounds.

Acknowledgements
Donald and Xavier would like to thank Michael Erdmann for comments and discussions on the substance and presentation of the proofs, which we found very helpful.

6 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).

[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.