Kinodynamic Motion Planning[1]

Bruce Donald[2]

Patrick Xavier[2]

John Canny[3]

John Reif[4]

Abstract:

Kinodynamic planning attempts to solve a robot motion problem subject to simultaneous kinematic and dynamics constraints. In the general problem, given a robot system, we must find a minimal-time trajectory that goes from a start position and velocity to a goal position and velocity while avoiding obstacles by a safety margin and respecting 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] An early version of this work appeared as [CDRX]

[2] Computer Science Department, Cornell University, Ithaca, NY 14853

[3] Computer Science Division, University of California, Berkeley, CA 94720

[4] Computer Science Department, Duke University, Durham, NC

Kinodynamic Motion Planning[5]

Bruce Donald[6]

Cornell Computer Science Department

Patrick Xavier[6]

Cornell Computer Science Department

John Canny

U. C. Berkeley Computer Science Division

John Reif

Duke Computer Science Department

Abstract: Kinodynamic planning attempts to solve a robot motion problem subject to simultaneous kinematic and dynamics constraints. In the general problem, given a robot system, we must find a minimal-time trajectory that goes from a start position and velocity to a goal position and velocity while avoiding obstacles by a safety margin and respecting 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 dynamics constraints, such as

[5]An early version of this work appeared as [CDRX]

[6]This paper describes research done in the Computer Science Robotics Laboratory at Cornell University. Support for our robotics research is provided in part by the National Science Foundation under grants No. IRI-8802390, IRI-9000532 and Presidential Young Investigator award No. IRI-8957316, and in part by the Mathematical Sciences Institute.

modulus bounds on velocity, acceleration, and force. A kinodynamic solution is a mapping from time to generalized forces or accelerations. 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 solutions that require minimal time and respect the kinodynamic constraints.

While there has been a great deal of work on this problem in the robotics community, there are no exact algorithms except for the one-dimensional case. Furthermore, it can be shown that in three dimensions, finding exact solutions is $\mathcal{NP}$-hard.[7] Therefore, it is reasonable to pursue *approximation algorithms* — algorithms that compute kinodynamic solutions that are "close" to optimal. However, 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.

We incorporate safety into the meaning of "optimal" by including a speed-dependent obstacle avoidance margin in the problem parameters. From this viewpoint, it is intuitive that approximation algorithms for kinodynamic planning should trade off planning time (computational complexity) against optimality in terms of: (a) execution time of the motion, (b) strictness in observing the safety margin, and (c) closeness to the desired start and goal positions and velocities.

To analytically express this trade-off we parameterize closeness to an optimal *safe* solution by a tolerance ϵ, and we bound the planning algorithm's running time in terms of this ϵ. Roughly speaking, we show that if there exists a "safe" optimal-time kinodynamic solution requiring time T_{opt}, then we can find a "near-optimal" solution that requires time at most $(1+\epsilon)T_{opt}$. Furthermore, the running time of our algorithm is polynomial both in the closeness of the approximation $\frac{1}{\epsilon}$ and in the geometric complexity. 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.

2 Kinodynamic Motion Planning

2.1 The Kinodynamic Planning Problem

Kinematic constraints, such as joint limits and obstacles, limit the configuration (position) of a robot. *Dynamics constraints* govern the time-derivatives of configuration (independent of obstacles). They include dynamics laws and bounds on velocity, acceleration, and applied force. *Strictly kinodynamic* constraints are obstacle-dependent constraints that govern configuration *and* its time-derivatives but do not fall into either of the previous categories.

[7] It was first observed in [CDRX] that the methods of Canny and Reif [CR] can be extended to demonstrate $\mathcal{NP}$-hardness. For a complete proof see [Xa].

An example of such a constraint is a speed-dependent obstacle-avoidance margin. A constraint is a *kinodynamic constraint* if it belongs to one of the above categories. The *state* of a robot at a given time is its configuration and velocity. The *general kinodynamic planning problem* is, for a given robot, to find a motion that goes from a start state to a goal state while obeying kinodynamic constraints.

We consider the following restricted problem. (See figure 1.) A point mass in $\Re^d$ ($d = 2, 3$) must be moved from a state $\mathbf{S} = (\mathbf{s}, \dot{\mathbf{s}})$ to a goal state $\mathbf{G} = (\mathbf{g}, \dot{\mathbf{g}})$. In the course of the motion, the point must avoid a set of polyhedral obstacles. Movement is controlled by applying forces or commanding accelerations, which are equivalent for a point mass. By using a configuration space approach, this problem is readily extended to cover a rigid non-rotating robot geometrically described by the union $\mathcal{R}$ of convex polyhedra.

We will denote the configuration space $\Re^d$ by C, and its phase space by TC. Phase space TC is the robot state space and is isomorphic to $\Re^{2d}$. Thus, a point in TC is a *(position, velocity)* pair such as $\mathbf{S}$ or $\mathbf{G}$.

A robot motion over a time interval $[0, T_f]$ can be specified by a twice-differentiable map $\mathbf{p} : [0, T_f] \to C$. This map is the *path* of the motion. In kinodynamic planning, the motion must obey dynamics and dynamics constraints, and it is convenient to specify $\dot{\mathbf{p}}$ explicitly. The *trajectory* of a robot motion is the map $\Gamma : [0, T_f] \to TC$ given by $\Gamma(t) = (\mathbf{p}(t), \dot{\mathbf{p}}(t))$. We denote the position and velocity components of a subscripted trajectory Γ_r by $\mathbf{p}_r$ and $\dot{\mathbf{p}}_r$, respectively. While a motion $\mathbf{p}$ can be given directly as a function of time, two equivalent specifications are useful: (a) an initial position $\mathbf{p}_0$ and a velocity function $\mathbf{v} = \dot{\mathbf{p}}$, and (b) an initial state $(\mathbf{p}_0, \mathbf{v}_0)$ and an acceleration function $\mathbf{a} = \ddot{\mathbf{p}}$.

The motion must respect upper bounds on the magnitudes of the acceleration and velocity. At all times t the acceleration $\ddot{\mathbf{p}}(t)$ and the velocity $\dot{\mathbf{p}}(t)$ must obey

$$\|\dot{\mathbf{p}}(t)\|_\infty \leq v_{max}, \qquad \text{and} \tag{1}$$

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

Eqs. (1) and (2) are the *dynamics bounds*. When the meaning is clear from context, we will drop the *max* subscript.

We assume that the obstacles $\mathcal{O}$ are represented by a set of convex, possibly overlapping polyhedra. Suppose these convex polyhedra have a total of n faces overall. We call n the *combinatorial complexity* of $\mathcal{O}$. Note that n is also the number of bounding halfplanes of the obstacles. *Free space* is the complement of these obstacles. We assume that the set of free configurations is bounded by a d-cube of side length l. A *general kinodynamic planning problem*, then, is a tuple $(\mathcal{O}, \mathbf{S}, \mathbf{G}, l, a, v)$.

An *exact solution* to the kinodynamic planning problem is a trajectory Γ such that $\Gamma(0) = \mathbf{S}$, $\Gamma(T_f) = \mathbf{G}$, and Γ obeys the kinodynamic constraints. That is, the path $\mathbf{p}$ avoids all obstacles, the velocity $\dot{\mathbf{p}}$ respects (1), and $\ddot{\mathbf{p}}$ respects (2). The *time* for a solution Γ is simply T_f. The *time-optimal* kinodynamic planning problem is to find a minimal-time kinodynamic solution, which is represented as a suitable encoding of the start state $\Gamma(0)$ and the acceleration function $\mathbf{a}$.

A theoretically time-optimal solution may require unrealizable precision in control or sensing and thus be unexecutable by a physical robot. For this reason, an optimal solution should observe a safety margin; the margin we define is speed-dependent. Furthermore, the safety margin ensures the existence of a "tube" or family of solutions "nearby" in time and in phase-space that "approximate" the optimal safe solution. The existence of such a "tube" of approximating solutions is essential for our approach. Safety margins are both practically motivated and mathematically necessary.

A *δ_v-safe kinodynamic solution* avoids all obstacles by a safety margin δ_v. In this paper, we define this safety margin to be an affine function of the trajectory speed. This first-order choice roughly corresponds to how accurately and quickly a robot senses its position and velocity, combined with how quickly it can correct for velocity errors.[8] Two positive scalars c_0 and c_1 characterize the safety margin, which one can view as an obstacle-free tube centered about the path. Formally, a *δ_v-safe kinodynamic solution* has the property that for all times t in $[0, T_f]$, there exists a ball about $\mathbf{p}(t)$ in free space of radius

$$\delta_v(c_0, c_1)(\dot{\mathbf{p}}(t)) = c_0 + c_1\|\dot{\mathbf{p}}(t)\|.$$

We will drop the parameters c_0 and c_1 in the discussion when confusion will not arise. Note that δ_v-safety is is an example of a kinodynamic constraint that is neither a pure kinematic constraint nor a pure dynamics constraint. A *δ_v-safe kinodynamic planning problem*, then, is a tuple $(\mathcal{O}, \mathbf{S}, \mathbf{G}, a, v, l, c_0, c_1)$. We call a, v, l, c_0, and c_1 the *kinodynamic bounds*.

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

We now specify what it means for a kinodynamic solution Γ_q to be *ϵ-approximately optimal*, where a positive $\epsilon < 1$ parameterizes the closeness of the approximation. First of all, Γ_q must obey the safety margin

$$\delta'_v(c_0, c_1)(\dot{\mathbf{p}}_q) = (1 - \epsilon)\delta_v(c_0, c_1)(\dot{\mathbf{p}}_q). \tag{3}$$

Second, if an optimal safe trajectory takes time T_{opt}, then we require that

$$T_q \leq (1 + \epsilon)T_{opt}.$$

Now, let us say that an approximating state $(\mathbf{x}', \dot{\mathbf{x}}')$ is "ϵ-close" to a reference state $(\mathbf{x}, \dot{\mathbf{x}})$ if

$$\|\mathbf{x} - \mathbf{x}'\| = O(\epsilon), \quad \text{and} \tag{4}$$

$$\left\|\frac{\dot{\mathbf{x}}}{1 + \epsilon} - \dot{\mathbf{x}}'\right\| = O(\epsilon). \tag{5}$$

[8] Consider a one-dimensional system. Recall that $\frac{\partial E}{\partial v} = mv$. Therefore, if the control system allows a maximum velocity error of Δv, and F_{res} force is available for correcting velocity errors, then $\frac{mv\Delta v}{F_{res}}$ distance might be traveled erroneously before the velocity can be corrected. Concisely stated, c_1 characterizes how accurately the robot can control its energy consumption.

For the final criterion, we require that $\Gamma_q(0)$ and $\Gamma_q(T_q)$ be ϵ-close to the desired start and goal states $\mathbf{S}$ and $\mathbf{G}$, respectively.[9]

In order to obtain our result, we must assume four things: a velocity bound, a diameter bound, L_∞-norm acceleration, and safety. Each of these assumptions can be motivated in physical terms. For example, robots exist in the physical world, and hence of course any actual robot will have bounded maximum velocity and a bounded workspace. However, the proofs in this paper do not go through if any of these assumptions is dropped. In [DX2, DX3, RT], we relax the L_∞-norm. Safety, as we shall see, proves to be a crucial assumption.

2.2 Statement of Results

In section 3.2 we describe a provably good approximation algorithm for the optimal safe kinodynamic planning problem. Concisely stated, we show:

Theorem 2.1 *Let $\mathcal{K} = (\mathcal{O}, \mathbf{S}, \mathbf{G}, a, v, l, c_0, c_1)$ be an optimal safe kinodynamic planning problem. Let $0 < \epsilon < 1$. Let n be the combinatorial complexity of the obstacles $\mathcal{O}$.*

Suppose there is a $\delta_v(c_0, c_1)$-safe trajectory that obeys the dynamics bounds a and v and goes from $\mathbf{S}$ to $\mathbf{G}$ in time T_{opt}. Then the algorithm finds a $\delta'_v(c_0, c_1)$-safe trajectory that obeys the dynamics bounds, takes at most time $(1+\epsilon)T_{opt}$, and goes from some $\mathbf{S}^*$ to some $\mathbf{G}^*$ such that $\mathbf{S}^*$ is ϵ-close to $\mathbf{S}$ and $\mathbf{G}^*$ is ϵ-close to $\mathbf{G}$.

The running time of the algorithm is

$$O\left(n\left[\frac{lv\gamma^3}{\epsilon^6}\right]^d\right),$$

$d = 2, 3$, *where*

$$\gamma = \max\left(\frac{a(c_1+1)}{c_0}, \sqrt{\frac{a(c_1+1)}{2c_0}}, \frac{a}{v}\right).$$

An optimal safe kinodynamic planning problem has two complexity components. The *combinatorial complexity* is the number n of bounding halfplanes on the obstacles $\mathcal{O}$. The algebraic complexity of the kinodynamic bounds (a, v, l, c_0, c_1) is the number of bits needs to encode them. Our algorithm is an ϵ-approximation scheme that is *fully polynomial* in the combinatorial complexity of the geometry and *pseudo-polynomial*[10] in the algebraic complexity of the kinodynamic bounds.

Note that we cannot claim that the approximately optimal safe solution is necessarily near to a truly optimal safe solution in position. In this respect it is useful to compare our

[9] Note that the definition of "ϵ-close" is not symmetric because of the velocity condition in (5). The condition $\|\dot{\mathbf{x}} - \dot{\mathbf{x}}'\| = O(\epsilon)$ may perhaps seem more intuitive. While the results here satisfy this definition, (5) allows simpler proofs and is necessary for natural extensions of our work beyond the scope of this paper.

[10] That is, the algorithm has a running time that is polynomial in the quantities a, v, l, c_0, and c_1, but it is not polynomial in the size (bit-complexity) of their encodings. See [PS] for a discussion of pseudopolynomiality.

result to Papadimitriou's fully polynomial approximation scheme for 3D Euclidean shortest path [Pap]. Specifically, neither method necessarily finds a solution that is spatially close to the optimal path, but merely one that has a length (time) that is not too much longer than the optimal length (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.

These above results can be extended to a rigid, non-rotating robot whose geometry is given by a union $\mathcal{R}$ of convex polyhedra. This configuration space transformation has been discussed extensively in the literature (see, eg, [LoP]). The algorithm of [LoP] could be used as a preprocess to reduce the planning problem for $\mathcal{R}$ amidst $\mathcal{O}$ to the point navigation problem we discuss. Since the dynamics equations for such a robot are identical to those of a point robot, we only need to map the problem to this configuration space and apply the algorithm.

2.3 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 dynamics 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 can be viewed as a version of optimal kinodynamic planning with the acceleration bound a 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 $\frac{1}{a}\|\dot{\mathbf{p}}\|^2$. These algorithms might lead in time to an exact solution to kinodynamic problems in 2D and 3D.

3 Algorithm and Analysis

3.1 The General Idea

Our approach transforms the problem of finding an approximately minimal-time trajectory to finding the shortest path in a directed graph. The vertices of the graph "discretize" the statespace TC, and the edges of the graph correspond to trajectory segments that each take time τ, a parameter computed by the algorithm.

Given the acceleration bound a, let $\mathcal{A}$ be the set of constant accelerations whose components are members of $\{-a, 0, a\}$. Let us choose a timestep τ such that velocity bound v is a multiple[11] of $a\tau$. Applying a member of $\mathcal{A}$ for duration τ is called an *(a,τ)-bang*. (See figure 2.) We also use this term to refer to the resulting trajectory segment: we say that there is an *(a,τ)-bang* from state $\mathbf{X}$ to state $\mathbf{Y}$ if following an (a,τ)-bang moves from $\mathbf{X}$ to $\mathbf{Y}$.

Suppose $\mathbf{S}^* = (\mathbf{s}^*, \dot{\mathbf{s}}^*) \in TC$ such that $\dot{\mathbf{s}}^*$ is a vector of multiples of $a\tau$. Suppose that $(\mathbf{p}, \dot{\mathbf{p}})$ is a state reachable from $\mathbf{S}^*$ by some sequence of (a,τ)-bangs. Then for each coordinate i,

$$\begin{array}{rcl} p_i & = & s_i^* + \frac{m_i}{2} a\tau^2 \quad \text{and} \\ \dot{p}_i & = & \dot{s}_i^* + n_i a\tau \end{array} \tag{6}$$

for some integers m_i and n_i. Thus, all states reachable from $\mathbf{S}^*$ under a sequence of (a,τ)-bangs belong to a set of states that lie at the interstices of an underlying, regular grid embedded in TC. This grid has spacings of $\frac{a\tau^2}{2}$ in position and $\frac{a\tau}{2}$ in velocity. We call this set of intersticial states the *TC-grid*, and each of these states a *TC gridpoint*.[12] We call a trajectory that results from a sequence of (a,τ)-bangs between TC-gridpoints an *(a,τ)-grid-bang trajectory*.

Recall (3). We say that state $(\mathbf{x}, \dot{\mathbf{x}})$ obeys δ'_v-safety if the ball of radius $\delta'_v(\|\dot{\mathbf{x}}\|)$ about $\mathbf{x}$ lies in free space. If τ is small enough, then a δ_v-safe trajectory will imply the existence of a δ'_v-safe (a,τ)-grid-bang trajectory that meets the other approximation requirements. Since each (a,τ)-bang takes time τ, finding a minimal-time δ'_v-safe (a,τ)-grid-bang trajectory between TC-gridpoints $\mathbf{X}$ and $\mathbf{Y}$ is identical to finding the shortest path in a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ embedded in TC. The vertices $v_i \in \mathcal{V}$ are the TC-gridpoints, and the edges $e_j \in \mathcal{E}$ are the δ'_v-safe (a,τ)-bangs between pairs of these vertices. We say that τ, $\mathbf{S}^*$, the kinodynamic parameters, and ϵ *induce* the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

---

[11] We use *multiple* to mean "integer multiple."

[12] If the grid-spacing in velocity is $a\tau$, then the closest velocity grid-coordinate is always at most $\frac{a\tau}{2}$ away; this is what is needed for our proofs. The spacing along the velocity axis is $a\tau$. At any *fixed* grid-velocity (multiple of $a\tau$), the spacing along the position axis is $a\tau^2$. However, the grid positions for odd multiples of $a\tau$ (velocity) are offset by $\frac{a\tau^2}{2}$ from the grid positions for even multiples of $a\tau$. Hence, all the relevant states lie at the interstices of an *underlying*, regular grid with spacings of $\frac{a\tau^2}{2}$ in position and $a\tau$ in velocity. Thus, the mapping from states to the interstices of the underlying grid is one-to-one, but not onto. The size of the underlying grid provides a bound on the number of reachable states. See [Xa] for further discussion.

3.2 The Algorithm

To explain the algorithm we need two more definitions. First, given two non-negative scalars η_x and η_v, we say that *state* $\mathbf{X}$ *is within* (η_x, η_v) *of state* $\mathbf{Y}$ if $\|\mathbf{x}-\mathbf{y}\| \leq \eta_x$ and $\|\dot{\mathbf{x}}-\dot{\mathbf{y}}\| \leq \eta_v$. Second, consider two trajectories Γ_a, $\Gamma_b : [0,T] \to TC$. Given two scalars η_x and η_v, we say that we say that Γ_a *approximately tracks* Γ_b *to tolerance* (η_x, η_v) *in the* L_∞*-norm* if for all times t,

$$\begin{aligned} \|\mathbf{p}_a(t) - \mathbf{p}_b(t)\|_\infty &\leq \eta_x, \quad \text{and} \\ \|\dot{\mathbf{p}}_a(t) - \dot{\mathbf{p}}_b(t)\|_\infty &\leq \eta_v. \end{aligned}$$

Given problem $(\mathcal{O}, \mathbf{S}, \mathbf{G}, a, v, l, c_0, c_1)$ and approximation parameter ϵ, our algorithm does the following:

1. It chooses a timestep τ as a function of a, v, ϵ, c_0, and c_1. Specifically, the algorithm chooses the largest τ such that $\tau \leq \frac{\epsilon v}{a}$, $a\tau | v$, and
$$\tau \leq \frac{\epsilon}{13}\min\left(\sqrt{\frac{2c_0\epsilon}{a(c_1+1)}}, \frac{c_0\epsilon}{a(c_1+1)}\right).$$
2. Next, the algorithm chooses the starting TC-gridpoint $\mathbf{S}^*$ according to the following: $\mathbf{s}^* = \mathbf{s}$, and $\dot{\mathbf{s}}^*$ is the multiple of $a\tau$ closest to $\frac{\dot{\mathbf{s}}}{1+\epsilon}$.
3. It then searches for the shortest path in the induced embedded graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, described above, from $\mathbf{S}^*$ to any state (vertex) that is within $(\frac{a\tau^2}{2}, \frac{a\tau}{2})$ of $(\mathbf{g}, \frac{\dot{\mathbf{g}}}{1+\epsilon})$. The algorithm explores the graph using breadth-first search, checking the δ'_v-safety of each (a,τ)-grid-bang it considers.

To show the correctness and complexity of the algorithm we must show how to choose τ so that the following holds: if there exists a δ_v-safe trajectory from $\mathbf{S}$ to $\mathbf{G}$ taking time T, then there also exists a δ'_v-safe (a,τ)-grid-bang trajectory between states ϵ-close to $\mathbf{S}$ and $\mathbf{G}$ that takes time $(1+\epsilon)T$.

We first observe that if trajectory Γ_{opt} obeys dynamics bounds a and v, then there is a time-rescaled [Hol] trajectory Γ'_{opt} that takes time $(1+\epsilon)T_{opt}$ and that obeys dynamics bounds $\frac{a}{(1+\epsilon)^2}$ and $\frac{v}{(1+\epsilon)}$. We then choose η_x and η_v that guarantee that if Γ_q tracks Γ'_{opt} to tolerance (η_x, η_v), then it will be δ'_v-safe. We then show there is a τ proportional to ϵ^2 such that there exists an (a,τ)-grid-bang trajectory Γ_q that (a) approximately tracks Γ'_{opt} to this tolerance, and (b) is within $(\frac{a\tau^2}{2}, \frac{a\tau}{2})$ of Γ'_{opt} when $t = 0$ and when $t = (1+\epsilon)T_{opt}$. The latter implies ϵ-closeness.

Finally, we show that δ_v-safety-checking is $O(n)$ per (a,τ)-bang. Recalling that TC-gridpoints have the form (6), we find that $|\mathcal{V}|$ is $O\left(\frac{lv}{a^2\tau^3}\right)^d$. The definition of (a,τ)-bang implies that the maximal out-degree in $\mathcal{G}$ is 3^d. Thus we get the complexity bound in Theorem 2.1.

3.3 Time-Rescaling and Safe Tracking

We say that a path $\mathbf{p}$ is *traversed* by a trajectory Γ if the image of the position component of Γ is equal to the image of $\mathbf{p}$.

Lemma 3.1 *If path $\mathbf{p}$ is traversed in time T_r by a trajectory Γ_r under acceleration bound a and velocity bound v, then there exists some Γ'_r that traverses $\mathbf{p}$ in time $T_r(1+\epsilon)$ while obeying acceleration bound $\frac{a}{(1+\epsilon)^2}$ and velocity bound $\frac{v}{1+\epsilon}$. In particular, this is true of $\Gamma'_r = (\mathbf{p}'_r, \dot{\mathbf{p}}'_r)$, where*

$$\begin{array}{rcl} \mathbf{p}'_r(t) & = & \mathbf{p}_r\left(\frac{t}{1+\epsilon}\right); \\ \dot{\mathbf{p}}'_r(t) & = & \frac{1}{1+\epsilon}\dot{\mathbf{p}}_r\left(\frac{t}{1+\epsilon}\right). \end{array} \tag{7}$$

Proof: Follows from the results of [Hol] or from direct computation using (7). $\Box$

To prove the main theorem we need to note that (7) preserves δ_v-safety:

Observation 3.2 *Suppose Γ_r is a $\delta_v(c_0, c_1)$-safe trajectory from $\mathbf{S}$ to $\mathbf{G}$ that takes time T_r and obeys bounds v and a. Then Γ'_r as defined by (7) is $\delta_v(c_0, c_1)$-safe, obeys bounds $\frac{v}{1+\epsilon}$ and $\frac{a}{(1+\epsilon)^2}$, and goes from $\mathbf{S}' = (\mathbf{s}, \frac{\dot{\mathbf{s}}}{1+\epsilon})$ to $\mathbf{G}' = (\mathbf{g}, \frac{\dot{\mathbf{g}}}{1+\epsilon})$.*

Intuitively, we expect that if a trajectory Γ_q tracks Γ'_r as defined in (7) closely enough, then Γ_q will be $(1-\epsilon)\delta_v$-safe. We have the following lemma, which is independent of norm:

Lemma 3.3 (The Safe Tracking Lemma) *Let δ_v be specified by c_0 and c_1, and let $0 < \epsilon < 1$. Let the tracking tolerance (η_x, η_v) satisfy the condition*[13]

$$\begin{array}{rcl} \eta_v & \leq & \frac{c_0\epsilon}{c_1+1}, \quad \text{and} \\ \eta_x & \leq & \frac{c_0\epsilon}{c_1+1}. \end{array} \tag{8}$$

Suppose that 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.

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 that $\mathbf{x} \in C$ lies inside the δ'_v-tube induced by Γ_q. Then for some $t_x \in [0, (1+\epsilon)T_r]$,

$$\|\mathbf{x} - \mathbf{p}_q(t_x)\| < c'_0 + c'_1\|\dot{\mathbf{p}}_q(t_x)\|. \tag{9}$$

Let $B_\eta(\mathbf{x})$ denote the ball of radius η about $\mathbf{x}$. If $\Gamma_q(t)$ tracks $\Gamma'_r(t)$ to tolerance (η_x, η_v), then $\mathbf{p}_q(t_x) \in B_{\eta_x}(\mathbf{p}'_r(t_x))$ and $\dot{\mathbf{p}}_q(t_x) \in B_{\eta_v}(\dot{\mathbf{p}}'_r(t_x))$, Therefore

$$\begin{array}{rcl} \|\mathbf{x} - \mathbf{p}'_r(t_x)\| & \leq & \|\mathbf{x} - \mathbf{p}_q(t_x)\| + \eta_x, \quad \text{and} \\ \|\dot{\mathbf{p}}_q(t_x)\| & \leq & \|\dot{\mathbf{p}}'_r(t_x)\| + \eta_v. \end{array}$$

[13] We write two inequalities because η_x and η_v have different dimensions (units).

Since $\mathbf{p}_r'(t_x) = \mathbf{p}_r(\frac{t_x}{1+\epsilon})$ and $\dot{\mathbf{p}}_r'(t_x) = \frac{1}{1+\epsilon}\dot{\mathbf{p}}_r(\frac{t_x}{1+\epsilon})$, by substituting into (9) and adding η_x to the right-hand side,

$$\left\|\mathbf{x} - \mathbf{p}_r\left(\frac{t_x}{1+\epsilon}\right)\right\| < c_0' + \eta_x + c_1'\left(\left\|\dot{\mathbf{p}}_r\left(\frac{t_x}{1+\epsilon}\right)\right\| + \eta_v\right). \tag{10}$$

Now, suppose that $\beta > 0$ and that η_x and η_v satisfy the following condition:

$$\begin{array}{rcl} \eta_v & \leq & \frac{c_0\epsilon}{c_1(1-\epsilon)+\beta} \\ \eta_x & \leq & \beta\eta_v. \end{array} \tag{11}$$

Simple manipulation then shows that $\eta_x + (1-\epsilon)c_1\eta_v \leq \epsilon c_0$. Thus,

$$c_0' + \eta_x + c_1'\eta_v \leq c_0. \tag{12}$$

But then,

$$c_0' + \eta_x + c_1'\left(\left\|\dot{\mathbf{p}}_r\left(\frac{t_x}{1+\epsilon}\right)\right\| + \eta_v\right) \leq c_0 + c_1\left\|\dot{\mathbf{p}}_r\left(\frac{t_x}{1+\epsilon}\right)\right\|. \tag{13}$$

This implies that $\|\mathbf{x} - \mathbf{p}_r(\frac{t_x}{1+\epsilon})\| < c_0 + c_1\|\dot{\mathbf{p}}_r(\frac{t_x}{1+\epsilon})\|$ via (10). Therefore, $\mathbf{x}$ lies inside the δ_v-tube induced by Γ_r.

Since $\mathbf{x}$ is an arbitrary point inside the δ_v'-tube induced by Γ_q, it follows that the δ_v'-tube induced by Γ_q lies entirely inside the δ_v-tube induced by Γ_r. Recall the hypotheses concerning c_0, c_1, and ϵ. Choosing $\beta = 1$, we see that condition (8) and the other hypotheses of the lemma together ensure that (11) is satisfied. $\square$

3.4 The Tracking Lemma

The Tracking Lemma relates a timestep size τ to a tracking tolerance. In particular, it tells how to choose τ to assure that in the absence of obstacles, for every Γ_u that obeys dynamics bounds $\frac{a}{(1+\epsilon)^2}$ and $\frac{v}{1+\epsilon}$ there will exist an (a,τ)-grid-bang trajectory Γ_q that tracks Γ_u to tolerance (η_x, η_v). We first need the following.

Lemma 3.4 *Let $\epsilon > 0$, and let Γ_u be a trajectory respecting dynamics bounds $\frac{a}{(1+\epsilon)^2}$ and $\frac{v}{1+\epsilon}$. Let $\tau \leq \frac{\epsilon v}{a}$, and let $a\tau | v$.*

Suppose that

$$N \geq \frac{6}{\epsilon}. \tag{14}$$

Then if $(\mathbf{p}_{q0}, \dot{\mathbf{p}}_{q0})$ is a TC-gridpoint such that $\|\mathbf{p}_{q0} - \mathbf{p}_u(0)\| \leq \frac{a\tau^2}{2}$ and $\|\dot{\mathbf{p}}_{q0} - \dot{\mathbf{p}}_u(0)\| \leq \frac{a\tau}{2}$, there exists an (a,τ)-grid-bang trajectory Γ_q such that $\Gamma_q(0) = (\mathbf{p}_{q0}, \dot{\mathbf{p}}_{q0})$ and that $\Gamma_q(N\tau)$ is within $(\frac{a\tau^2}{2}, \frac{a\tau}{2})$ of $\Gamma_u(N\tau)$.

Proof:

Since we are using the L_∞-norm, it is sufficient to consider the case of a one-dimensional configuration space C. We show that the lower bound on N given by (14) is sufficiently large to guarantee that if Γ_u meets the hypotheses of the lemma then some (a,τ)-grid-bang trajectory can meet the endpoint conditions.

Let $\epsilon > 0$, $\tau \leq \frac{\epsilon v}{a}$, and $a\tau | v$. Let $p_u(0)$, $\dot{p_u}(0)$, and $\dot{p_u}(N\tau)$ be fixed, and consider some Γ_u that satisfies these endpoint conditions and the hypotheses of the lemma. Let $(p_{q0}, \dot{p}_{q0})$ be a TC-gridpoint within $(\frac{a\tau^2}{2}, \frac{a\tau}{2})$ of $\Gamma_u(0)$. To find a sufficiently large N, we introduce variables b and $\mathcal{Q}$. These variables depend on Γ_u. Let b be an integer such that such that

$$|\dot{p}_{q0} + ba\tau - \dot{p}_u(N\tau)| \leq \frac{a\tau}{2}.$$

Now, define $\mathcal{Q}$ to be the collection of all (a,τ)-grid-bang trajectories of time length $N\tau$ starting at $(p_{q0}, \dot{p}_{q0})$ with net velocity change $ba\tau$. The positions reached by the different trajectories in $\mathcal{Q}$ at time $N\tau$ form a set of discrete points spaced $a\tau$ apart. Call these positions the *$\mathcal{Q}$-positions*. If the range of $\mathcal{Q}$-positions spans the range of possible $\Gamma_u(N\tau)$, then for some trajectory $\Gamma_q \in \mathcal{Q}$, $|p_q(N\tau) - p_u(N\tau)| \leq \frac{a\tau^2}{2}$.

We show how to choose N so that the maximum $\mathcal{Q}$-position exceeds the maximum possible $p_u(N\tau)$. A similar argument shows that the same N guarantees that the minimum $\mathcal{Q}$-position is less than the minimum possible $p_u(N\tau)$. For the remainder of this proof, we will refer to the trajectories that achieve the maximum $\mathcal{Q}$-position and the maximum possible $p_u(N\tau)$ as Γ_q and Γ_u, respectively.

The Γ_q that attains the maximum $\mathcal{Q}$-position obeys either (a) full positive acceleration, possibly followed by zero acceleration for one timestep, followed by full negative acceleration[14], or (b) full positive acceleration until its velocity is v, followed by zero acceleration, followed by full negative acceleration. Similarly, a Γ_u that maximizes $p_u(N\tau)$ obeys either (c) full positive acceleration followed by full negative acceleration, or (d) full positive acceleration until its velocity is $\frac{v}{(1+\epsilon)}$, followed by zero acceleration, followed by full negative acceleration.

Consider $\dot{p}_q$ and $\dot{p}_u$ and their role in determining $p_q(N\tau)$ and $p_u(N\tau)$. In the worst case, $\dot{p}_{q0} = \dot{p}_u(0) - \frac{a\tau}{2}$ and $\dot{p}_q(N\tau) = \dot{p}_u(N\tau) - \frac{a\tau}{2}$. If $p_q(N\tau)$ is to be greater than $p_u(N\tau)$, then we can divide the interval $[0, N\tau]$ into three intervals during which Γ_q "loses ground to" Γ_u, "gains on" Γ_u, and "loses ground to" Γ_u. (See figure 3.) In other words, there are times t_c and t_l, $0 < t_c < t_l < N\tau$, such that

$$\begin{array}{rcll} \dot{p}_q(t) & < & \dot{p}_u(t) & \text{if } 0 < t < t_c; \\ \dot{p}_q(t) & > & \dot{p}_u(t) & \text{if } t_c < t < t_l; \quad \text{and} \\ \dot{p}_q(t) & < & \dot{p}_u(t) & \text{if } t_l < t < N\tau. \end{array}$$

Now, when Γ_q is accelerating full-positive,

$$\ddot{p}_q(t) - \ddot{p}_u(t) \geq a\left(1 - \frac{1}{(1+\epsilon)^2}\right) > \frac{\epsilon a}{2}.$$

[14]The zero acceleration timestep in the first case occurs if $N - \frac{\dot{p}_q(N\tau) - \dot{p}_{q0}}{a\tau}$ is odd.

Similarly, when Γ_q is accelerating full-negative,

$$\ddot{p}_q(t) - \ddot{p}_u(t) < \frac{\epsilon a}{2}.$$

Hence, when $0 \leq t < t_c$, $\ddot{p}_q(t) - \ddot{p}_u(t) > \frac{\epsilon a}{2}$, and when $t_l < t \leq N\tau$, $\ddot{p}_q(t) - \ddot{p}_u(t) < -\frac{\epsilon a}{2}$. Then, because $\dot{p}_{q0} \geq \dot{p}_u(0) - \frac{a\tau}{2}$ and $\dot{p}_q(N\tau) \geq \dot{p}_u(N\tau) - \frac{a\tau}{2}$,

$$\begin{array}{rcl} \int_0^{t_c} (\dot{p}_u(t) - \dot{p}_q(t))dt & < & \frac{a\tau^2}{4\epsilon}, \quad \text{and} \\ \int_{t_l}^{N\tau} (\dot{p}_u(t) - \dot{p}_q(t))dt & < & \frac{a\tau^2}{4\epsilon}. \end{array}$$

Therefore, it is sufficient to choose an N that guarantees

$$\int_{t_c}^{t_l} (\dot{p}_q(t) - \dot{p}_u(t))dt \geq \frac{a\tau^2}{2\epsilon}. \tag{15}$$

Consider the behavior of $\dot{p}_q(t) - \dot{p}_u(t)$ between times t_c and t_l. For now, suppose that $\dot{p}_q(t) < v$ during this time. Then, for an interval of time I_c beginning with t_c, both $\dot{p}_q(t)$ and $\dot{p}_q(t) - \dot{p}_u(t)$ increase; for an interval I_l beginning at most one timestep after I_c and ending with t_l, both $\dot{p}_q(t)$ and $\dot{p}_q(t) - \dot{p}_u(t)$ decrease. Furthermore,

$$\begin{array}{rcll} \dot{p}_q(t) - \dot{p}_u(t) & > & \frac{a\epsilon}{2}(t - t_c) & \text{during } I_c, \text{ and} \\ \dot{p}_q(t) - \dot{p}_u(t) & > & \frac{a\epsilon}{2}(t_l - t) & \text{during } I_l. \end{array}$$

Using some manipulation, we then see that if $\dot{p}_q(t) < v$ for all $t \in [t_c, t_l]$, the condition

$$t_l - t_c \geq \frac{2\tau}{\epsilon} + 1 \tag{16}$$

guarantees that (15) is true.

Now, suppose $\dot{p}_q(t) = v$ for some interval $I_m \subset [t_c, t_l]$. (See figure 4.) Then, for an interval of time I_c immediately preceeding I_m and beginning with t_c, both $\dot{p}_q(t)$ and $\dot{p}_q(t) - \dot{p}_u(t)$ increase; for an interval I_l immediately after I_m and ending with t_l, both $\dot{p}_q(t)$ and $\dot{p}_q(t) - \dot{p}_u(t)$ decrease. However, during I_m, $\dot{p}_q(t) - \dot{p}_u(t) \geq a\tau$, since $\tau \leq \frac{\epsilon v}{a}$. It follows that (16) again guarantees that (15) is true.

We observe that

$$\begin{array}{rcl} t_c & \leq & \frac{\tau}{\epsilon}, \text{ and} \\ N\tau - t_l & \leq & \frac{\tau}{\epsilon}. \end{array}$$

Recalling (16), we see that the following choice of N guarantees that the range of $\mathcal{Q}$-positions will be adequate:

$$N = 4\left\lceil \frac{1}{\epsilon} \right\rceil + 1.$$

Using the fact that $0 < \epsilon < 1$, we obtain the sufficient condition (14). $\Box$

Lemma 3.5 (The Tracking Lemma) *Let $\epsilon > 0$. Let Γ_u be a trajectory that respects dynamics bounds $\frac{a}{(1+\epsilon)^2}$ and $\frac{v}{1+\epsilon}$ and takes time T_u. Let η_x and η_v be positive. Let $\tau \leq \frac{\epsilon v}{a}$*

Suppose that

$$\tau \leq \frac{\epsilon}{13} \min\left(\sqrt{\frac{2c_0\epsilon}{a(c_1+1)}}, \frac{c_0\epsilon}{a(c_1+1)}\right), \tag{17}$$

and $a\tau | v$. Furthermore, let N be given by (14), and suppose that $T_u \geq N\tau$. Then in the absence of obstacles there exists an (a,τ)-grid-bang trajectory Γ_q respecting dynamics bounds v and a that approximately tracks Γ_u to tolerance (η_x, η_v) during $[0, T_u]$ and obeys the following conditions:

$$\begin{array}{rcl} \mathbf{p}_q(0) & = & \mathbf{p}_u(0), \\ \|\dot{\mathbf{p}}_q(0) - \dot{\mathbf{p}}_u(0)\|_\infty & \leq & \frac{a\tau}{2}, \\ \|\mathbf{p}_q(T_u) - \mathbf{p}_u(T_u)\|_\infty & \leq & a\tau^2, \qquad \text{and} \\ \|\dot{\mathbf{p}}_q(T_u) - \dot{\mathbf{p}}_u(T_u)\|_\infty & \leq & a\tau. \end{array} \tag{18}$$

Proof:

We show that the the upper bound (17) is correct. Let the hypotheses of the lemma be satisfied. Let N be given by (14). Then it follows from Lemma 3.4 that there is an (a,τ)-grid-bang trajectory Γ_q such that for any positive integer k satisfying $kN\tau \leq T_u$,

$$\begin{array}{rcl} \|\mathbf{p}_q(kN\tau) - \mathbf{p}_u(kN\tau)\|_\infty & \leq & \frac{a\tau^2}{2}, \text{ and} \\ \|\dot{\mathbf{p}}_q(kN\tau) - \dot{\mathbf{p}}_u(kN\tau)\|_\infty & \leq & \frac{a\tau}{2}. \end{array} \tag{19}$$

This can be shown by induction on k.

Now, for all t, $\|\ddot{\mathbf{p}}_u(t) - \ddot{\mathbf{p}}_q(t)\|_\infty \leq 2a$. By considering the relative velocity in the worst case, where along some axis

$$\begin{array}{rcl} |p_q(kN\tau) - p_u(kN\tau)| & = & \frac{a\tau^2}{2}, \\ |\dot{p}_q(kN\tau) - \dot{p}_u(kN\tau)| & = & \frac{a\tau}{2}, \\ |p_q((k+1)N\tau) - p_u((k+1)N\tau)| & = & -\frac{a\tau^2}{2}, \text{ and} \\ |\dot{p}_q((k+1)N\tau) - \dot{p}_u((k+1)N\tau)| & = & -\frac{a\tau}{2}, \end{array}$$

we conclude that for all $t \in [0, T_u]$,

$$\|\mathbf{p}_q(t) - \mathbf{p}_u(t)\|_\infty < \frac{a(N+1)^2\tau^2}{2} \tag{20}$$

To guarantee that the right-hand side of (20) is less than η_x, it is sufficient that

$$\tau < \sqrt{\frac{2\eta_x}{a}}\left(\frac{1}{N+1}\right). \tag{21}$$

Since $\|\dot{\mathbf{p}}_q(t) - \dot{\mathbf{p}}_u(t)\|_\infty < a(N+1)\tau$ when $0 \leq t \leq T_u$, for the velocity case we require that

$$\tau \leq \frac{\eta_v}{a}\left(\frac{1}{N+1}\right). \tag{22}$$

If $T_u > N\tau$ is not a multiple of $N\tau$, then for some natural numbers $n < N$ and k, $(kN+n)\tau$ is within $\frac{\tau}{2}$ of T_u. Substituting twice the value for N from Lemma 3.4 and rounding to simplify yields the condition (17). □

3.5 Safety Checking

We describe how to check whether an (a, τ)-bang violates the speed-dependent safety-margin $\delta_v(c_0, c_1)$ in $O(n)$ time. We review some basic computational geometry, describe the special case when $c_1 = 0$, and then extend the method to the general case.

As noted above, we assume that obstacles are the union of convex polyhedra. For now, let the safety margin be a constant $c_0 > 0$, and define the B_{c_0} to be the L_∞ ball with radius c_0. Staying c_0-safe relative to a convex polyhedron A is then equivalent to avoiding $\overline{A} = A \oplus B_{c_0}$, where "$\oplus$" denotes the Minkowski sum. Since B_{c_0} is a d-cube, $\overline{A}$ is also a convex polyhedron and has $O(|faces(A)|)$ faces. By taking the Minkowski sum of each of the obstacles with B_{c_0} we obtain the *expanded obstacles.*

Suppose $\overline{A}$ has faces $\{ F_0, \ldots, F_m \}$ lying on the boundary planes of the closed half-spaces $\{ H_0, \ldots, H_m \}$. The boundary plane of each H_i is the kernel of an affine function f_i. If $\mathbf{n}_i$ is a unit vector in the outward normal direction from the boundary plane of H_i, and $\mathbf{y}_i$ is any point on this boundary then

$$f_i(\mathbf{x}) = \langle \mathbf{n}_i, \mathbf{x} \rangle - \langle \mathbf{n}_i, \mathbf{y}_i \rangle. \tag{23}$$

The polyhedron $\overline{A}$ is thus described by a set of functions $\mathcal{F} = \{ f_0, \ldots, f_m \}$.

A point $\mathbf{x}$ is on the boundary of $\overline{A}$ if and only if it lies on some *closed* face F_k of $\overline{A}$. Equivalently, $f_k(\mathbf{x}) = 0$, and for all f_j that determine an edge of F_k, $f_j(\mathbf{x}) \leq 0$. Since for a convex polyhedron the numbers of edges and faces are linearly related and an edge is common to two faces, determining whether $\mathbf{x}$ lies on the boundary of any of the expanded of obstacles takes total time $O(n)$.

Without loss of generality, suppose that (a, τ)-bang $\mathbf{p}$ begins at $t = 0$ and that $\mathbf{p}(0)$ is c_0-safe. We then can check the c_0-safety of $\mathbf{p}(t)$ by determining whether $\mathbf{p}(t)$ violates the boundary of an expanded obstacle. For a face F_k of $\overline{A}$, we only need to solve $f_k(\mathbf{p}(t)) = 0$, and for each solution t_s check whether $f_j(\mathbf{p}(t_s)) > 0$ for some f_j that determines an edge of F_k with f_k.

Now, consider the speed-dependent safety function δ_v. For each time t, the point $\mathbf{p}(t)$ should avoid the obstacles expanded by an L_∞ ball with radius $\delta_v(\dot{\mathbf{p}}(t))$.

In other words, $\mathbf{p}(t)$ is δ_v-safe relative to a convex polyhedron A if and only if it avoids the expanded obstacle $\tilde{A}(\dot{\mathbf{p}}(t)) = A \oplus B_{\delta_v}(\dot{\mathbf{p}}(t))$, where $B_{\delta_v}(\dot{\mathbf{p}}(t))$ is the L_∞ ball with radius $\delta_v(\dot{\mathbf{p}}(t))$. $\tilde{A}(\dot{\mathbf{p}}(t))$ is described similarly to $\overline{A}$, by a set of functions $\tilde{\mathcal{F}} = \{ \tilde{f}_0, \ldots, \tilde{f}_m \}$. For each $f_i \in \mathcal{F}$,

$$\tilde{f}_i(\mathbf{p}(t), \dot{\mathbf{p}}(t)) = \langle \mathbf{n}_i, \mathbf{p}(t) \rangle - \langle \mathbf{n}_i, \mathbf{y}_i + \mathbf{q}_i \|\dot{\mathbf{p}}(t)\|_\infty \rangle, \tag{24}$$

where $\mathbf{q}_i$ is a constant vector that depends only on $\mathbf{n}_i$. To check whether $\mathbf{p}(t)$ violates the faces of $\tilde{A}$, we use the $\tilde{f}_i \in \tilde{\mathcal{F}}$ the same way we use the $f_i \in \mathcal{F}$ above.

Since $\mathbf{p}(t)$ is quadratic, $\tilde{f}_k(\mathbf{p}(t), \dot{\mathbf{p}}(t)) = 0$ has solutions of the form $t = a + \sqrt{b}$. When computing the inequalities we can square twice to eliminate the radical, and thus it is adequate to compute square roots symbolically. This implies that safety checking never requires numbers longer (in the number of bits) than a constant-multiple of the length of the longest number in the input. Therefore, we can still use the real-RAM computation model. By the same argument as in checking whether a point is in on an expanded obstacle boundary, we need to solve $O(n)$ equations and check $O(n)$ inequalities, overall. Therefore, the cost of safety-checking is $O(n)$ per (a, τ)-bang.

3.6 Proving the Main Theorem

We can now prove Theorem 2.1.

Proof of Theorem 2.1:

Let $\mathcal{K} = (\mathcal{O}, \mathbf{S}, \mathbf{G}, a, v, l, c_0, c_1)$ be an instance of the optimal safe kinodynamic planning problem. Let $0 < \epsilon < 1$.

Suppose Γ_{opt} is a $\delta_v(c_0, c_1)$-safe trajectory that obeys the dynamics bounds a and v and goes from $\mathbf{S}$ to $\mathbf{G}$ in time T_{opt}. By Lemma 3.1 and Observation 3.2 that follows it, there is a trajectory Γ'_{opt} that is $\delta_v(c_0, c_1)$-safe, obeys bounds $\frac{v}{1+\epsilon}$ and $\frac{a}{(1+\epsilon)^2}$, takes time $(1+\epsilon)T_{opt}$, and goes from $\mathbf{S}' = (\mathbf{s}, \frac{\dot{\mathbf{s}}}{1+\epsilon})$ to $\mathbf{G}' = (\mathbf{g}, \frac{\dot{\mathbf{g}}}{1+\epsilon})$.

Now suppose we run the algorithm described in Section 3.2. The choice of τ in the algorithm matches the conditions in Lemma 3.5 when the values of η_x and η_v from equation (8) in Lemma 3.3 are substituted into equation (17). Furthermore, the algorithm's choice of $\mathbf{S}^*$ obeys the condition on $(\mathbf{p}_{q0}, \dot{\mathbf{p}}_{q0})$ in Lemma 3.4. Therefore, some (a, τ)-grid-bang trajectory beginning at $\mathbf{S}^*$ tracks Γ'_{opt} closely enough to be $\delta'_v(c_0, c_1)$-safe, to obey the dynamics bounds a and v, and to take time $(1+\epsilon)$ to reach a state $\mathbf{G}^*$ within tolerance $(\frac{a\tau^2}{2}, \frac{a\tau}{2})$ of $\mathbf{G}'$.

Breadth-first search guarantees such a trajectory will be found if there is no $\delta'_v(c_0, c_1)$-safe (a, τ)-grid-bang trajectory beginning at $\mathbf{S}^*$ that obeys the dynamics bounds and comes adequately close to $\mathbf{G}'$ in less time. Thus, the algorithm will find a trajectory meeting the conditions of the theorem.

To establish the time bound, we now bound the number $G_\infty(a, \tau, v, l, d)$ of TC-gridpoints for a point robot with maximum (L_∞) speed v in a d-dimensional free-space of diameter l. Without loss of generality, choose $\mathbf{s}^*$ to be the "zero" position. Recalling the canonical form of a TC-gridpoint from (6), we conclude

$$G_\infty(a, \tau, v, l, d) = O\left(\left(\frac{vl}{a^2\tau^3}\right)^d\right).$$

Since the number of (a, τ)-bangs from a state is constant (3^d) and the cost of checking the safety of a bang is $O(n)$, the total complexity of the algorithm can be obtained by substituting in τ from (17). Since $\epsilon < 1$, we can use

$$\tau \leq \frac{\epsilon^2}{13} \min\left(\sqrt{\frac{2c_0}{a(c_1+1)}}, \frac{c_0}{a(c_1+1)}\right)$$

instead of (17) to get the bound in the theorem.

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. For work in this direction, see [DX1,DX2,DX3, Xa].

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). For work in this direction, see [Xa].

4. Exact algorithms should be explored. For work in this direction, see [CRR].

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 non-holonomic 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.

From a combinatorial standpoint, we believe that in order to obtain near- (ϵ-) optimality for the figure-skating problem, a grid such as ours would have to have at least exponential size. In particular, we conjecture that the grid spacing may be a superpolynomial function of the minimum distance between obstacles.

6. It would be interesting to extend our approach to 2-norm velocity and acceleration bounds. For work in this direction, see [DX2, DX3, RT, Xa].

7. It would be of value to extend our approach to to manipulator systems with full rotational dynamics. For example, one might consider the rigid body dynamics of open kinematic chains with revolute and prismatic joints. Finding near-optimal kinodynamic solutions in these cases would be of great interest. For work in this direction, see [JHCP, DX2, DX3, Xa].

In addition, there is a great deal of interesting experimental work to be done, in reducing these algorithms to practice, and on developing search heuristics. For work on implementation of our approach, and experiments, see [DX1, Xa]. 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.

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

5 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).

[C] Canny, J. F. *Collision Detection for Moving Polyhedra,* PAMI-8(2) (1986).

[CDRX] Canny, J., Donald, B., Reif J., and Xavier, P. *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).

[CRR] Canny, J. and Rege, A. and Reif, J. *An Exact Algorithm for Kinodynamic Planning in the Plane,* Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkeley, Calif. pp. 271–280 (1990).

[DX1] Donald, B.,and Xavier, P. *A Provably Good Approximation Algorithm for Optimal-Time Trajectory Planning,* IEEE Int. Conf. On Robotics and Automation, Scottsdale, AZ (1989).

[DX2] Donald B. and Xavier P. *Near-Optimal Kinodynamic Planning for Robots With Coupled Dynamics Bounds, Fourth IEEE International Symposium on Intelligent Control,* (Sept. 24-26) Albany, NY (1989).

[DX3] Donald B. and Xavier P. *Provably Good Approximation Algorithms for Optimal Kinodynamic Planning for Cartesian Robots and Open Chain Manipulators,* Proc. 6^{th} ACM Symposium on Computational Geometry, Berkeley, CA (June). (1990).

[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).

[JHCP] Jacobs, P., Heinzinger, G., Canny, J., and Paden, B. *Planning Guaranteed Near-Time-Optimal Planning in a Cluttered Workspace,* Int'l Workshop onf Sensorial Integration for Industrial Robots: Architectures & Applications, Zaragoza, Spain. November (1989).

[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).

[RT] Reif, J., and Tate, S. *Approximate Kinodynamic Planning Using L_2-norm Dynamic Bounds,* Duke University Tech. Report (1989).

[SH] Sahar, G., and Hollerbach, J. *Planning of Minimum-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).

[Xa] Xavier, P. G. *Approximation Algorithms for Optimal Kinodynamic Robot Plans,* PhD Dissertation, Computer Science Department, Cornell University, Ithaca, NY. Also CU-CS technical report. (1992).

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

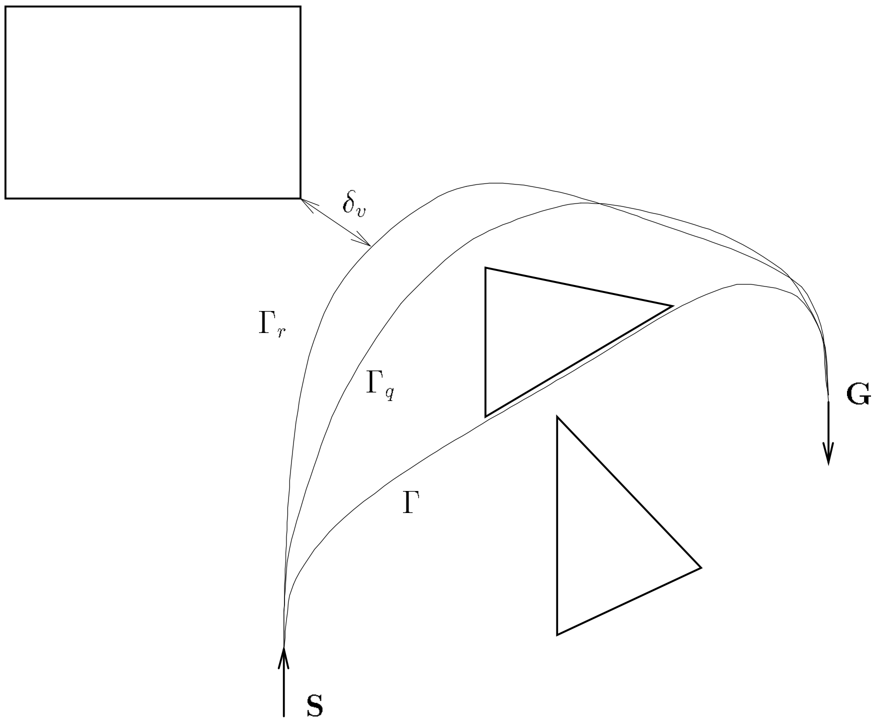

Figure 1: A kinodynamic planning problem for a point robot, showing the obstacles, the start **S**, the goal **G**, and three solutions: time-optimal Γ, optimal (safe) Γ_r, and approximately optimal Γ_q, which happens to be exact at the start and goal.

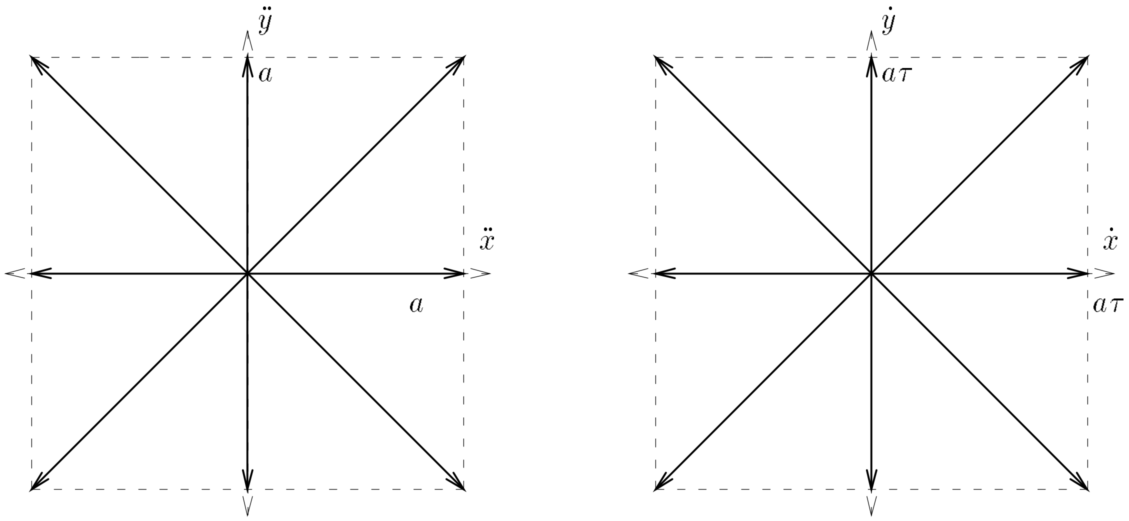

Figure 2: Extremal accelerations (left) that generate (a, τ)-bangs (right).

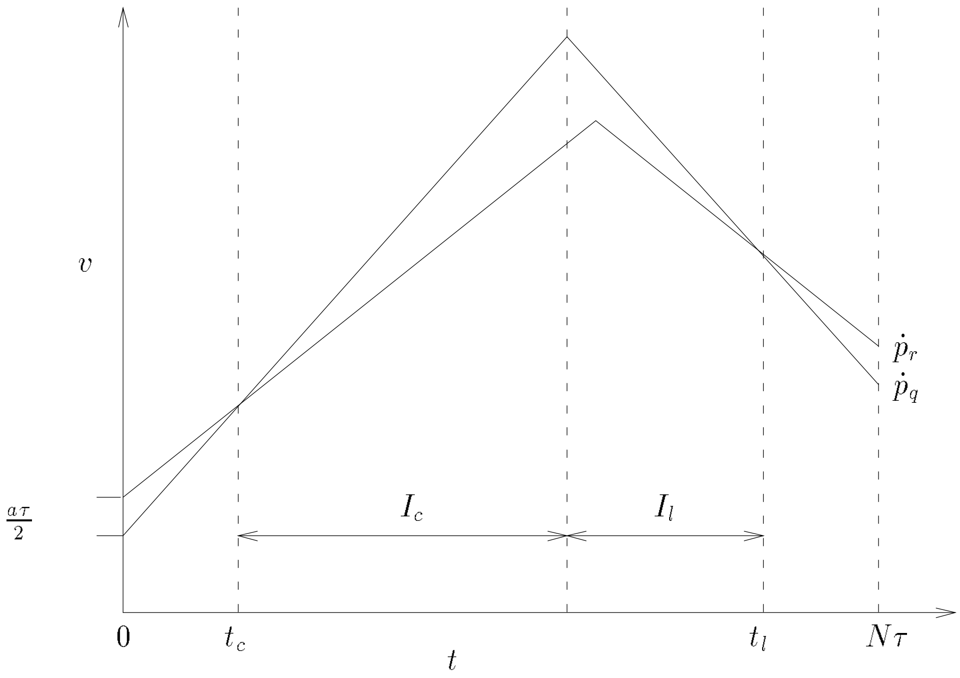

Figure 3: An example of $\dot{p}_r$ and $\dot{p}_q$ that achieve the maximum position subject to conditions at times 0 and $N\tau$. In this case, $\dot{p}_q$ never reaches the maximum allowed velocity v. N must be large enough so that the distance Γ_q gains over Γ_r during I_c and I_l makes up for the distance Γ_q loses to Γ_r during $[0, t_c]$ and $[t_l, N\tau]$.

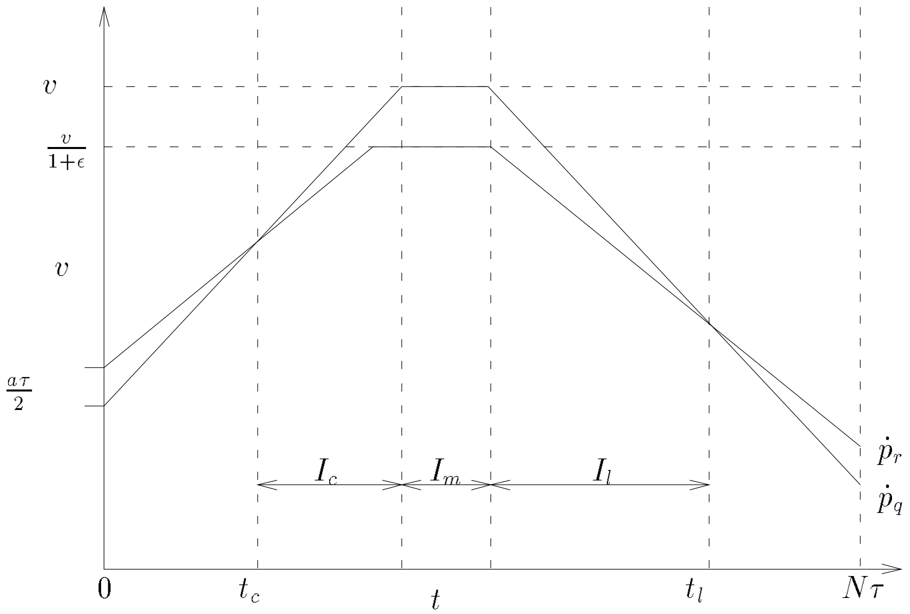

Figure 4: An example of $\dot{p}_r$ and $\dot{p}_q$ that achieve the maximum position subject to conditions at times 0 and $N\tau$. In this case, $\dot{p}_q$ sustains the maximum allowed velocity v for the interval I_m. N must be large enough so that the distance Γ_q gains over Γ_r during I_c, I_m, and I_l makes up for the distance Γ_q loses to Γ_r during $[0, t_c]$ and $[t_l, N\tau]$. Note that in this figure, the condition $\tau \leq \frac{\epsilon v}{a}$ is not met.