Provably Good Approximation Algorithms for Optimal Kinodynamic Planning for Cartesian Robots and Open Chain Manipulators

Bruce R. Donald*
Patrick Xavier*

TR 90-1095
February 1990

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

*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 grant no. IRI-8802390 and by a Presidential Young Investigator award, and in part by the Mathematical Sciences Institute.

Provably Good Approximation Algorithms for Optimal Kinodynamic Planning for Cartesian Robots and Open Chain Manipulators

Bruce R. Donald[1]

Patrick Xavier[1]

Cornell University Computer Science Dept., Ithaca NY

Abstract:

We consider the following problem: given a robot system, find a minimal-time trajectory from a start state to a goal state, while avoiding obstacles by a speed-dependent safety-margin and respecting dynamics bounds. In [CDRX] we developed a theoretical, provably good approximation algorithm for the minimum-time trajectory problem for a robot system with decoupled dynamics bounds (*e.g.* a point robot in $\Re^3$). This algorithm differs from previous work in three ways. It is possible (1) to bound the goodness of the approximation by an error term ϵ; (2) to polynomially bound the computational complexity of our algorithm; and (3) to express the complexity as a polynomial function of the error term. Hence, given the geometric obstacles, dynamics bounds, and the error term ϵ, the algorithm returns a solution that is ϵ-close to optimal and requires only a polynomial (in $(\frac{1}{\epsilon})$) amount of time.

We extend the results of [CDRX] in two ways. First, we reanalyze the [CDRX] algorithm for robots with decoupled dynamics bounds. We halve the exponent in the polynomial bounds and prove a better approximation accuracy. These new results indicate that an implementation of the theoretical algorithm could be reasonable. We report on a preliminary implementation of the extended algorithm and experiments.

Second, we extend [CDRX] to d-link, revolute-joint 3D robots will full rigid body dynamics. Specifically, we first prove a generalized trajectory-tracking lemma for robots with coupled dynamics bounds. Then, using this result we describe polynomial-time approximation algorithms for Cartesian robots obeying L_2 dynamics bounds and open kinematic chain manipulators with revolute and prismatic joints; the latter class includes most industrial manipulators. We obtain a general $O\left(n^2(\log n)(\frac{1}{\epsilon})^{6d-1}\right)$ algorithm, where n is the geometric complexity.

[1]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 grant No. IRI-8802390 and by a Presidential Young Investigator award, and in part by the Mathematical Sciences Institute.

Contents

1 Introduction

It has long been the hope of theoretical computer science that by making simplifying assumptions in robotics problems, precise combinatorial algorithms could be obtained, and that these results could later be generalized to cover "real" robots with uncertain control, full dynamics, etc. Thus, for example, it is common to assume point robots (often planar) that can be perfectly position-controlled, and that the dynamics is trivial. Optimization issues—such as finding the "fastest" path—are often ignored. Generalizing the early results and relaxing these assumptions are essential if algorithmic analysis is to remain relevant to robotics, particularly in motion planning; however, this task has proved arduous. In this paper, we present some fruits of that labor. We assume that our robots are 3D arbitrary d-link open kinematic chains with revolute and prismatic joints, and their motion amidst polyhedral obstacles is subject to full Lagrangian rigid body dynamics. Furthermore, we model uncertainty in the robot's control. We give combinatorially precise answers to the question, *What is the minimal time solution trajectory from a start to a goal, subject to obstacle and dynamics constraints, and respecting a speed-dependent "safety" margin?* To prove our result, we employ new techniques that forge a mathematical link between control- and complexity theory.

The *kinodynamic planning problem* [CDRX] is to synthesize a robot motion subject to simultaneous kinematic constraints (such as avoiding obstacles) and dynamics constraints (such as 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. A long-standing robotics problem has been to synthesize *time-optimal kinodynamic solutions*, by which we mean solutions that require minimal time and respect the kinodynamic constraints.

While there has been much work on this problem, there are no algorithms to generate exact solutions, with the exception of the one-dimensional case. Furthermore, lower bounds on the complexity of the exact problem indicate that it is inherently computationally intractable (NP-hard), even for a point robot obeying particle dynamics. Therefore, it is reasonable to pursue *approximation algorithms* — algorithms that compute kinodynamic solutions that are "close" to optimal. Three obvious questions arise about any such algorithm:

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

In general, before [CDRX] no bounds were proven about the goodness of the approximations, on the time-complexity of such algorithms, nor on the relation between the two.

The chief measure of optimality is time. However, because a real robot motion can only be considered safe if it avoids obstacles by some margin, it is natural to incorporate a safety measure into the meaning of "optimal". The problem formulation we introduce in [CDRX] therefore includes a speed-dependent obstacle-avoidance margin in the problem parameters, along with start and goal states, dynamics bounds, and a set of obstacles. We call this margin of error a "safety" function. An optimal trajectory is thus a minimum-time trajectory that respects the safety criterion. An "approximately optimal" trajectory approximately respects the safety criterion and approximately takes the same amount of time as an optimal trajectory. In this manner, the formulation follows our intuition by allowing approximation algorithms for kinodynamic planning to trade off running time against optimality in terms of: (a) execution time of the trajectory, (b) strictness in observing the safety margin, and (c) closeness to the desired start and goal states[2].

To analytically express this trade-off we parameterize closeness[3] to an optimal solution with a tolerance ϵ and bound algorithm running time in terms of this ϵ. We desire an algorithm that runs in polynomial time for a fixed ϵ while guaranteeing solution accuracy ϵ; if there exists a "safe" optimal-time kinodynamic solution requiring time T_{opt}, then the algorithm must find a "nearly-as-safe" solution that requires time at most $(1+\epsilon)T_{opt}$. Furthermore, as ϵ is made smaller to improve the solution quality, the running time should increase at most as a polynomial of $\frac{1}{\epsilon}$. We call such an algorithm a *provably good polynomial-time approximation algorithm.*

[CDRX] considered the restricted situation of particle dynamics, and provided a provably good polynomial-time approximation algorithm for 2- and 3-dimensional optimal kinodynamic planning. Thus, 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. These bounds on solution accuracy and running time were the first obtained for any kinodynamic planning problem in more than one dimension.

Here we extend [CDRX] in two ways: (a) we prove tighter accuracy and complexity bounds for the algorithm; and (b) we extend the approach to a class of robots that obey coupled dynamics bounds.

Specifically, we reanalyze the [CDRX] algorithm. We halve the exponent in the polynomial bounds from $6d$ to $3d$, showing that the algorithm is $O\left(\left(\frac{1}{\epsilon}\right)^{3d}\right)$. At the same time we show that if there exists a "safe" optimal-time kinodynamic solution requiring time T, then the algorithm finds a "nearly-as-safe" solution that requires time T, whereas the results in [CDRX] only show that this motion requires at most $(1+\epsilon)T$. While this analysis was only outlined in [DX], we give complete proofs here. These new results indicate that an implementation of the theoretical algorithm is reasonable, and we report on a preliminary implementation of the extended algorithm and experiments.

To extend the [CDRX] approach to robots with coupled dynamics bounds, we first prove a generalized trajectory-tracking lemma. We apply the Coupled Tracking Lemma to extend the general [CDRX] algorithm. We obtain provably good polynomial-time approximation schemes for a class of robot systems with coupled dynamics that includes

[2]This is a refinement in[DX1].

[3]This closeness is in terms of the performance measure, not spatial closeness.

Cartesian robots obeying L_2 dynamics bounds and open-chain manipulators with revolute and prismatic joints. For these cases, we show that if there exists a "safe" optimal-time kinodynamic solution requiring time T, then the motion the algorithm finds will be "nearly-as-safe" and take at most time $(1+\epsilon)T$. Our algorithms only consider near-extremal accelerations; thus the asymptotic branching-factor of the search is lower than that of the earlier provably good polynomial-time approximation algorithm of [JHCP] for open-chain manipulators. These algorithms are simple, but the new game-theoretic proof techniques we introduce are subtle, and employ tools from disparate parts of computational geometry, robotics, and dynamical systems.

2 Kinodynamic Motion Planning

2.1 The Kinodynamic Planning Problem

Kinodynamic planning attempts to solve a motion problem subject to simultaneous kinematic and dynamic constraints. A robot with d degrees of freedom must move from a start 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, it must avoid a set of obstacles and configuration (*e.g.* joint) limits: these are the *kinematic constraints.* We now describe the problem for two classes of robots: Cartesian robots, which are robots with decoupled dynamics equations, and open kinematic chains. For a Cartesian robot, we use the natural coordinate system in $\Re^d$; For a open-chain manipulator we use joint coordinates, although we allow obstacles to be expressed in real-world $\Re^3$ coordinates. The general problem formulation is easily applied to other classes of robots, such as wheeled robots.

The robot motion is controlled by applying generalized forces according to classical Lagrangian mechanics. Forces are applied at the center of mass of a Cartesian robot; torques and forces are applied at the joints of the open-chain manipulator. Generalized velocities and applied generalized forces must obey certain bounds. In particular, an open-chain manipulator obeys dynamics bounds $(\mathbf{v}_{max}, \mathbf{f}_{max})$ if for all times t the joint velocities $\dot{\mathbf{p}}(t)$ and the applied generalized forces $\mathbf{f}(t)$ obey the the following *dynamics constraints* at each joint i:

$$\begin{aligned} |\dot{p}_i(t)| &\leq v_{max\ i}, \\ |f_i(t)| &\leq f_{max\ i}. \end{aligned} \tag{1}$$

A Cartesian robot obeys L_p dynamics bounds (v_{max}, a_{max}) if for all times t, the acceleration $\mathbf{a}(t)$ and the velocity $\dot{\mathbf{p}}$ obey the following constraints:

$$\begin{aligned} \|\dot{\mathbf{p}}(t)\|_p &\leq v_{max}, \\ \|\mathbf{a}(t)\|_p &\leq a_{max}. \end{aligned} \tag{2}$$

We will denote our configuration space by C, and its phase space (the robot state space) by TC. A point in TC is a (position, velocity) pair such as $\mathbf{S}$ or $\mathbf{G}$.

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

We assume that obstacles are polyhedral and input as an arrangement $\mathcal{O}$ with n vertices. We map these obstacles to c-space obstacles in the configuration space [LoP] [Don]. *Free space* is the complement of the c-space obstacles in C. Note that the free space of a *rigid, non-rotating* polyhedral robot in a d-dimensional world with polyhedral obstacles is the free space of some point robot in a world d-dimensional with polyhedral obstacles, so that kinodynamic planning for a polyhedral Cartesian robot reduces to kinodynamic planning for a point robot.

In general, the problem parameters must include an encoding $\mathcal{M}$ of the physical quantities that determine the robot's dynamics equations which relate state and applied forces to accelerations. Thus, for a given link of an open-chain manipulator, we include the mass, the center of mass, the inertia, and the distance and relative orientation between joints; the dynamics of a Cartesian robot can be similarly encoded. A *general kinodynamic planning problem*, then, is a tuple $\mathcal{K} = (\mathcal{O}, \mathbf{S}, \mathbf{G}, \mathbf{f}_{max}, \mathbf{v}_{max}, \mathcal{M})$. We assume that the set of free configurations is bounded by a workspace d-cube of side length l combined with joint limits.

An *exact solution* to the kinodynamic planning problem is a suitable encoding of the acceleration map $\mathbf{a}$ such that $\Gamma(0) = \mathbf{S}$, $\Gamma(T) = \mathbf{G}$, and Γ obeys the kinematic and dynamic constraints. That is, $\mathbf{p}$ avoids all obstacles, and $\dot{\mathbf{p}}$ and $\mathbf{a}$ respect the appropriate dynamics constraints (1) or (2). The *time* for a solution $\mathbf{a}$ is simply T_f. The *time-optimal kinodynamic planning problem* is to find a kinodynamic solution requiring minimal execution time.

2.2 Optimal and Approximately Optimal Kinodynamic Plans

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; in practice, this margin should be speed-dependent. At the same time, the safety margin ensures the existence of a "tube" or family of "nearby" solutions in time, and in phase-space. Hence, these nearby solutions "approximate" the optimal safe solution both in time and in phase space. The existence of such a "tube" of approximating solutions is essential for our approach; hence safety is not only practically motivated, but also mathematically necessary.

We define a *δ_v-safe kinodynamic solution* as one that 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. One may think of this choice as corresponding to how accurately the dynamical system can control its energy consumption; the precise modeling of control

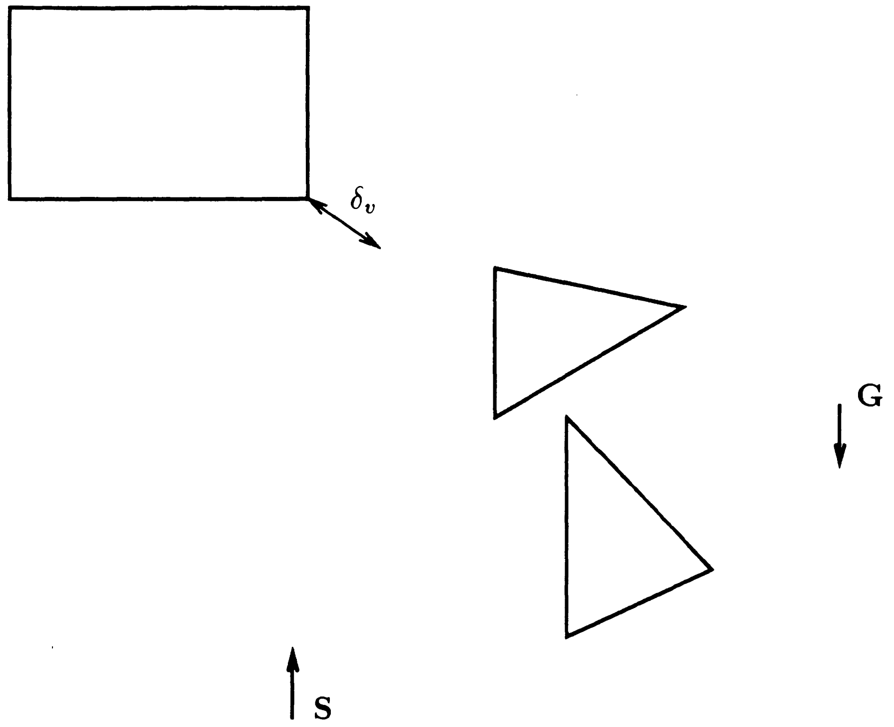


Figure 1: A kinodynamic planning problem for a point robot, showing the obstacles, the start $\mathbf{S} = (\mathbf{s}, \dot{\mathbf{s}})$, the goal $\mathbf{G} = (\mathbf{g}, \dot{\mathbf{g}})$.

---

and sensing error is not the focus of our present research [4]. Two positive scalars c_0 and c_1 characterize the safety margin, which specifies an obstacle-free tube centered about the path. We call this tube a $\delta_v(c_0, c_1)$*-tube*; its diameter grows linearly with speed.

Formally, a solution to the safe kinodynamic planning problem $\mathcal{K} = (\mathcal{O}, \mathbf{S}, \mathbf{G}, \mathbf{f}_{max}, \mathbf{v}_{max}, \mathcal{M}, c_0, c_1)$ 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 $\delta_v(t) = c_0 + c_1\|\dot{\mathbf{p}}(t)\|$. Note that δ_v-safety is actually a *mixed* kinematic and dynamic constraint; it is an example of a kinodynamic constraint that is neither a pure kinematic constraint nor a pure dynamics constraint.

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 safety criterion we consider here. The *optimal kinodynamic planning problem* is to find optimal solutions to the safe kinodynamic planning problem.

We now specify what it means for a kinodynamic solution to be "approximate" in terms of the parameter ϵ. First, we say that a trajectory $\Gamma_q : [0, T_q] \to TC$ is ϵ*-close* at

[4] Using a quadratic safety function yields results similar to those we report in this paper.

the start **S** (resp. goal **G**) if, at time 0 (resp. time T_q), it achieves the start (resp. goal) position to within an additive error that is $O(\epsilon)$ and independent of l and v_{max}, and the start (resp. goal) velocity to within a multiplicative error of ϵ, plus an additive error that is $O(\epsilon)$ and independent of l and v_{max} . That is, if $\Gamma_q = (\mathbf{p}_q, \dot{\mathbf{p}}_q)$, then both $|\mathbf{p}_q(0) - \mathrm{s}|_\infty$ and $|(1+\epsilon)\dot{\mathbf{p}}_q(0) - \dot{s}|_\infty$ are $O(\epsilon)$; similar conditions hold at the goal. Thus, by making ϵ smaller, Γ_q can be made to achieve the start and goal as accurately as desired.

Finally, we define an *approximately optimal solution* Γ_q to be a kinodynamic solution which is "ϵ-close" at the start and the goal, "near-optimal" in time, and "also safe". By "near-optimal", we mean that if the an optimal safe solution Γ takes time T_{opt}, the time T_q required by Γ_q is bounded from above by $(1+\epsilon)T_{opt}$. 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 $\delta_v(c_1', c_0')$, where $c_0' = (1-\epsilon)c_0$ and $c_1' = (1-\epsilon)c_1$. Where no ambiguity will result, we will abbreviate $\delta_v(c_0, c_1)$ by δ_v, and $\delta_v(c_0', c_1')$ by δ_v'.

A glossary of of the many symbols used in this paper can be found in the Appendix.

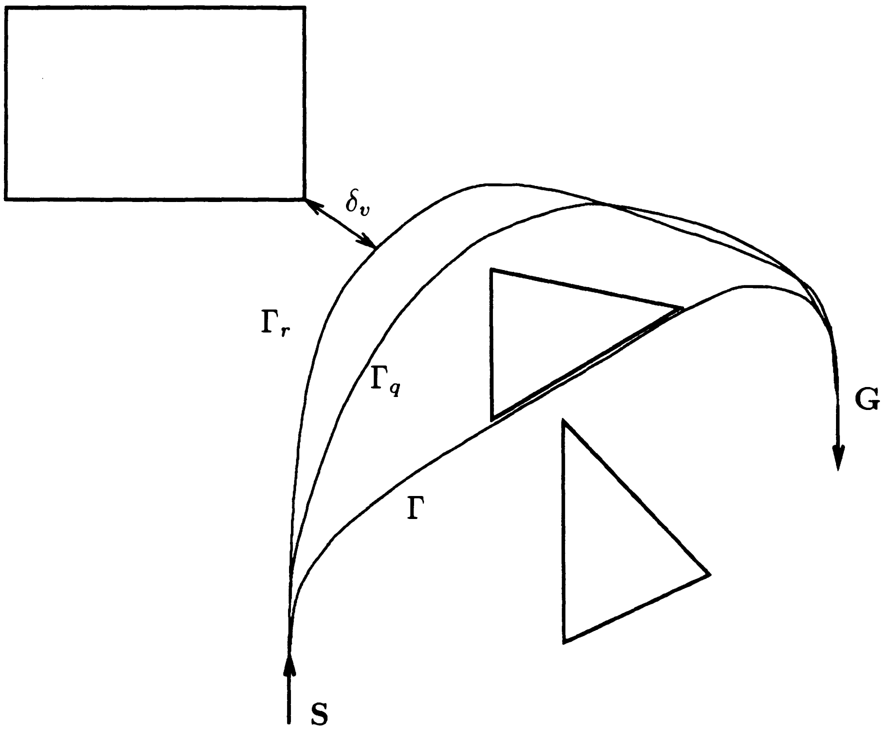


Figure 2: Γ is the time-optimal kinodynamic solution for the problem in figure 1. However, it is not δ_v-safe. Γ_r is the optimal δ_v-safe solution. Γ_q illustrates a solution that is almost as fast as Γ_r, and almost as safe. Our algorithm can find solutions such as Γ_q, and provides guarantees as to how fast and how safe they are, relative to Γ_r, and how close they are to **S** and **G**.

2.3 Statement of Results for a Cartesian Robot Obeying L_∞ Dynamics Bounds

We describe a provably good approximation algorithm for the optimal kinodynamic planning problem $\mathcal{K}_{L\infty} = (\mathcal{O}, \mathbf{S}, \mathbf{G}, a_{max}, v_{max}, l, c_0, c_1)$, where l is the workspace diameter, for polyhedral Cartesian robots obeying L_∞ dynamics bounds. Concisely stated, we show:

Theorem 2.1 *Let $(\mathcal{O}, \mathbf{S}, \mathbf{G}, a_{max}, v_{max}, l, c_0, c_1)$ be an optimal kinodynamic planning problem for a polyhedral Cartesian robot obeying L_∞ dynamics bounds. Let $0 < \epsilon < 1$.*

Suppose there is a $\delta_v(c_0, c_1)$-safe trajectory from $\mathbf{S}$ to $\mathbf{G}$ taking time T_{opt}. Then the algorithm finds a $(1-\epsilon)\delta_v(c_0, c_1)$-safe trajectory taking time at most T_{opt} from some $\mathbf{S}^* = (\mathbf{s}^*, \dot{\mathbf{s}}^*)$ to some $\mathbf{G}^* = (\mathbf{g}^*, \dot{\mathbf{g}}^*)$ such that $\mathbf{S}^*$ and $\mathbf{G}^*$ are within $O(\epsilon)$ of $\mathbf{S}$ and $\mathbf{G}$, respectively.

The running time of the algorithm is $O\left(n^2(\log n)\left[\frac{v_{max}(a_{max}c_1 + v_{max})^3 l}{a_{max}^2 c_0^3 \epsilon^3}\right]^d\right)$, $d = 2, 3$, where n is the geometric complexity of the problem. Thus, for a given robot in a world of fixed diameter, the algorithm has an asymptotic time bound of $O\left(n^2(\log n)\left(\frac{1}{\epsilon}\right)^{3d}\right)$.

Thus, the algorithm will find a trajectory that takes at most as long as the optimal trajectory but that is not guaranteed to be as safe. The bounds in this theorem are a significant improvement over the [CDRX] result of $O\left(n^2(\log n)\left(\frac{1}{\epsilon}\right)^{6d}\right)$ and $(1+\epsilon)T_{opt}$.

The theorem states that the 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 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 $\mathcal{O}$. The *algebraic complexity of the geometry* is the number of bits necessary to encode the coordinates of the vertices of $\mathcal{O}$, 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 as in [CDRX] 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 this with Papadimitriou's fully polynomial approximation scheme for 3D Euclidian 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.

We have completed a preliminary Common Lisp implementation of this algorithm in two dimensions, and we have run simple experiments on this implementation.

2.4 Results In Cases of Coupled Dynamics Bounds

A robot system obeys *coupled dynamics bounds* if (a) its dynamics equations cannot be separated into the dynamics equations of d one-dimensional systems or (b) or there is no set of orthogonal axes such that the velocity (acceleration) bound in each axial direction is independent of the bounds in all of the other axial directions.

We describe provably good approximation algorithms for the optimal (safe) kinodynamic planning problems for Cartesian Robots with L_2 dynamics bounds ($\mathcal{K}_{L2} = (\mathcal{O}, \mathbf{S}, \mathbf{G}, a_{max}, v_{max}, l, c_0, c_1)$) and for open-chain manipulators with revolute and prismatic joints ($\mathcal{K}_{oc} = (\mathcal{O}, \mathbf{S}, \mathbf{G}, \mathbf{f}_{max}, \mathbf{v}_{max}, \mathcal{M}, c_0, c_1)$). The algorithms produce approximately optimal solutions relative to the safety parameters c_0 and c_1. ϵ, the approximation parameter, specifies how close in time the desired solution should be to the optimal, safe solution. Thus, if the optimal $\delta_v(c_0, c_1)$-safe solution Γ takes time T_{opt}, then the algorithm finds an "ϵ-close", "also-safe" solution Γ_q such that $T_q \leq (1+\epsilon)T_{opt}$.

These algorithms run in time polynomial in the geometric complexity n and in the resolution ($\frac{1}{\epsilon}$). The coefficient of the polynomial for an open chain robot depends on certain constants determined from its physical properties. Again, our algorithms are ϵ-approximation schemes *fully polynomial* in the combinatorial and algebraic complexity of the geometry, and *pseudo-polynomial* in the kinodynamic bounds.

Concisely stated, we show the following theorems.

Theorem 2.2 *Let $(\mathcal{O}, \mathbf{S}, \mathbf{G}, a_{max}, v_{max}, l, c_0, c_1)$ be an optimal kinodynamic planning problem for a polyhedral Cartesian robot obeying L_2 dynamics bounds. Let $0 < \epsilon < 1$.*

Suppose there is a $\delta_v(c_0, c_1)$-safe trajectory from $\mathbf{S}$ to $\mathbf{G}$ taking time T_{opt}. Then the algorithm finds a $(1-\epsilon)\delta_v(c_0, c_1)$-safe trajectory taking time at most $T_{opt}(1+\epsilon)$ from some $\mathbf{S}^* = (\mathbf{s}^*, \dot{\mathbf{s}}^*)$ to some $\mathbf{G}^* = (\mathbf{g}^*, \dot{\mathbf{g}}^*)$ such that $\mathbf{S}^*$ and $\mathbf{G}^*$ are ϵ-close to $\mathbf{S}$ and $\mathbf{G}$, respectively.

The asymptotic running time of the algorithm is $O\left(n^2(\log n)\left(\frac{1}{\epsilon}\right)^{6d-1}\right)$, $d = 2, 3$, where n is the geometric complexity of the problem.

Theorem 2.3 *Let $(\mathcal{O}, \mathbf{S}, \mathbf{G}, \mathbf{f}_{max}, \mathbf{v}_{max}, \mathcal{M}, c_0, c_1)$ be an optimal kinodynamic planning problem for an open-chain manipulator with revolute and prismatic joints. Let $0 < \epsilon < 1$.*

Suppose there is a $\delta_v(c_0, c_1)$-safe trajectory from $\mathbf{S}$ to $\mathbf{G}$ taking time T_{opt}. Then the algorithm finds a $(1-\epsilon)\delta_v(c_0, c_1)$-safe trajectory taking time at most $T_{opt}(1+\epsilon)$ from some $\mathbf{S}^* = (\mathbf{s}^*, \dot{\mathbf{s}}^*)$ to some $\mathbf{G}^* = (\mathbf{g}^*, \dot{\mathbf{g}}^*)$ such that $\mathbf{S}^*$ and $\mathbf{G}^*$ are ϵ-close to $\mathbf{S}$ and $\mathbf{G}$, respectively.

The asymptotic running time of the algorithm is $O\left(p(n)(\log n)\left(\frac{1}{\epsilon}\right)^{6d-1}\right)$ where n is the geometric complexity of the problem, and $p(n)$ is the time-complexity of checking the $(1-\epsilon)\delta_v((c_0, c_1)$-safety of a constant-acceleration trajectory segment.

In Section 5.6 we show that $p(n)$ is roughly $n^2 \log n$. As we discuss in that section, it turns out that the collision detection requires the solution of trignometric equations that cannot be tranformed into algebraic equations using the usual substitution methods, such as [Can]. Thus, provably good numerical techniques must be used to approximately solve the collision detection problem. If in the problem formulation obstacles were represented algebraically *in configuration space*, then $p(n)$ would be $n^2 \log n$.

2.5 Straightforward Extensions

As will be discussed later, [CDRX] approach reduces the problem of finding an approximately optimal trajectory to that of finding the shortest path between two vertices of a uniform-cost graph whose vertices correspond to system states. On such graphs the single source or single sink shortest path problem can be solved with the same time-complexity.

The [CDRX] algorithm and the extensions discussed in this paper can be easily adapted to solve two analogous problems. In particular, we can modify these algorithms to build tables from which we can read off a solution from the source (to the sink) to any goal state (from any start state). The tables are directed graphs of size $O\left(\left(\frac{1}{\epsilon}\right)^{3d}\right)$ for L_∞-dynamics bounds Cartesian robots and $O\left(\left(\frac{1}{\epsilon}\right)^{5d}\right)$ for the L_2-dynamics bounds Cartesian robots and open chain manipulators. We describe the modification in Section 3.3 for details.

2.6 Review of Previous and Related 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 singular-extremal or "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.

The polyhedral euclidian shortest path problem is a version of optimal kinodynamic planning with the acceleration bound a_{max} is set to 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. [Pap] gives a fully-polynomial approximation algorithm for the shortest path problem. Ó'Dúnlaing [Odu] 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] and Jacobs and Canny [JC]. 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.

As mentioned before, [CDRX] introduces the algorithmic approach to kinodynamic planning and provides the first a provably good approximation algorithm for 2- and 3-dimensional optimal kinodynamic planning. [DX] outlines the improvements to these results proven in later in this paper.

[JHCP] give the first polynomial-time approximation method for optimal kinodynamic planning for robots with rotational dynamics – the first for robot systems with state dependent dynamics. Their work also introduces new algorithmic and proof techniques, including (a) discretizing acceleration-space according to the problem parameters and (b) reducing state-dependent dynamics to being locally constant. Our approach toward state-dependent dynamics is similar to theirs; however, as a result of our Coupled Tracking Lemma, our algorithm only needs to consider trajectory segments that are approximately at torque extremals. In concurrent independent work [RT] obtain a polynomial-time approximation scheme for Cartesian robots obeying L_2 dynamics bounds.

2.7 A Road Map of this Paper

As we are presenting several results requiring substantial technical discussion and full proofs, brief descriptions of the following sections might be useful to the reader.

We begin Section 3 by introducing the basic approach of [CDRX]. This review section is necessary to understand our new results. In Sections 3.1 and 3.2 we describe an underlying grid in phase-space and a finite set of constant accelerations that generate trajectory segments between interstices of this grid. In Section 3.3 we describe the [CDRX] algorithm.

Section 4 contains the lemmas and the theorem needed to obtain our improved accuracy and complexity bounds for Cartesian robots obeying L_∞ dynamics bounds. After presenting the necessary proofs, we use the theorem to derive the tighter bounds (given in Theorem 2.1).

In Section 5 we sketch our approach to approximately optimal kinodynamic planning for robots obeying coupled dynamics bounds. The basic idea, presented in Sections 5.2 and 5.3, is a generalization of the [CDRX] algorithm presented in Section 3. We review time-rescaling [Hol, CDRX] and Lagrangian dynamics in Section 5.4. We present Lemma 5.1 (The Coupled Tracking Lemma) and briefly describe how to apply it in Section 5.5. The lemma, which allows us to track the trajectories of certain systems obeying coupled dynamics bounds, is central to showing the accuracy and complexity bounds of our algorithm. We describe how to do safety-checking in Section 5.6.

We flesh out the details of the approach sketched in Section 5 in Section 6. After an overview, we give full proofs of the Coupled Tracking Lemma and a similar but stronger lemma that is easily applied to systems with state-dependent dynamics bounds, such as revolute-jointed arms. Then, using the Coupled Tracking Lemma, we obtain a provably good polynomial-time approximation algorithm for Cartesian robots obeying L_2 acceleration and velocity bounds in Section 6.3. In Section 6.4 we show in detail how to obtain the polynomial bound for kinodynamic planning for open-chain manipulators, using the stronger lemma.

We discuss our implementation and some experiments for the L_∞ case in two-dimensions in Section 7.

3 Kinodynamic Planning with L_∞ Dynamics Bounds

We now describe the approach introduced in [CDRX]. In Section 4 we present our new complexity and accuracy analysis in detail. We obtain our results (presented in Sections 5 through 6.4) for kinodynamic planning with coupled dynamics bounds by extending this 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 "gridpoint", and we will call the set of these points the TC-grid.

One could command moves from one gridpoint to another using piecewise-constant extremal accelerations. When all controls are extremal almost everywhere, such a motion is called "bang-bang". Borrowing from this terminology, we will call an acceleration generated by controls that are all extremal or null a "bang". Of course, not every gridpoint neighbor will be reachable with a single bang if an arbitrary discretization is used. However, it is possible to choose a finite set of constant extremal accelerations such that given a fixed timestep τ and a start state **S**, all states reachable after any sequence of "bangs" of duration τ generated by this set are gridpoints of a regular grid. Furthermore, if the statespace is bounded and the start state is not too close to a boundary, then the reachable gridpoints fill this grid as the number of bangs allowed grows.

The smaller τ is, the finer the underlying TC-grid. It seems intuitively plausible that for a sufficiently "fine" grid and a small enough τ, some trajectory resulting from a sequence of bangs should approximate the optimal safe trajectory. We will call a trajectory that consists of bang-accelerations between grid points a "grid-bang trajectory". Since each trajectory segment takes time τ, a path between two states in the TC-grid requiring a minimal number of segments is a minimal-time grid-bang trajectory.

Thus, the key issues are to choose the set of bangs and the grid spacing correctly and to prove that the approximation bound

$$T_q \leq T_{opt} \tag{3}$$

holds for some grid-bang trajectory Γ_q. Furthermore, the size of the grid must be polynomial in $\frac{1}{\epsilon}$. The proofs of these properties require certain non-trivial constructions.

3.2 The TC-grid

We now describe (a) the set of constant accelerations used to generate bangs, and (b) the TC-grid these bangs induce, given an appropriate start state. Later, we give a condition on the timestep τ used to generate bangs (Theorem 4.5). Note that the set of accelerations we describe in this section is adequate only for kinodynamic planning with L_∞ dynamics bounds.

Given a_{max} and v_{max}, let $\mathcal{A}$ be the set of constant accelerations whose components are members of $\{-a_{max}, 0, a_{max}\}$. Let us choose a timestep τ such that v_{max} is a multiple of $a_{max}\tau$. Applying a member of $\mathcal{A}$ for duration τ is called an *(a_{max}, τ)-bang.* (See figure 4.) We also use this term to refer to the resulting trajectory — we say that there is an (a_{max}, τ)-bang from $\mathbf{X}$ to $\mathbf{Y}$ if following an (a_{max}, τ)-bang moves from state $\mathbf{X}$ to state $\mathbf{Y}$.) We call a trajectory that results from a sequence of (a_{max}, τ)-bangs an *(a_{max}, τ)-bang trajectory.*

Suppose we start at a state $\mathbf{S} = (\mathrm{s}, \dot{\mathrm{s}})$ whose velocity component $\dot{\mathrm{s}}$ is a vector of multiples of $a_{max}\tau$. Let $(\mathbf{p}, \dot{\mathbf{p}})$ be a state reachable from $\mathbf{S}$ by some sequence of (a_{max}, τ)-bangs. Then for each spatial coordinate i, either

$$\begin{aligned} p_i &= s_i + m_i a_{max}\tau^2 \quad \text{and} \\ \dot{p}_i &= \dot{s}_i + 2n_i a_{max}\tau, \quad \text{or} \end{aligned} \tag{4}$$

$$\begin{aligned} p_i &= s_i + \left(\tfrac{2m_i+1}{2}\right) a_{max}\tau^2 \quad \text{and} \\ \dot{p}_i &= \dot{s}_i + (2n_i+1) a_{max}\tau, \end{aligned} \tag{5}$$

for some integers m_i and n_i. Thus, all states reachable from $\mathbf{S}$ under a sequence of (a_{max}, τ)-bangs lie on some regular grid, in particular the one whose gridpoints are given by (4) and (5). We choose this as the discretization for our TC-grid.

Recall that for a time t and velocity $\dot{\mathbf{p}}(t)$, $\delta'_v(t) = (1-\epsilon)\delta_v(t)$ and that a state $(\mathbf{p}(t), \dot{\mathbf{p}}(t))$ is δ'_v-safe if and only if the ball of radius $\delta'_v(t)$ about $\mathbf{p}(t)$ lies in free space. Now, let $\mathcal{V}$ be the set of TC-gridpoints that obey δ'_v-safety and the dynamics bounds. Let $\mathcal{E}$ be the set of pairs $(\mathbf{X}, \mathbf{Y})$ of members of $\mathcal{V}$ such that there is an (a_{max}, τ)-bang from $\mathbf{X}$ to $\mathbf{Y}$ that observes δ'_v-safety and the velocity bounds. Thus, we view the discrete search space as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We call this graph the TC-graph. Vertices of the graph are grid points, and directed edges correspond to (a_{max}, τ)-bangs. Thus, two vertices $\mathbf{X}$ and $\mathbf{Y}$ are connected by an edge if and only if Y is reachable from X by an (a_{max}, τ)-bang. The vertices connected to X by edges from X are called its neighbors.

Since each (a_{max}, τ)-bang takes time τ, a shortest path from a vertex $\mathbf{X}$ to a vertex $\mathbf{Y}$ in the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ corresponds to the minimal-time δ'_v-safe (a_{max}, τ)-bang trajectory from state $\mathbf{X}$ to state $\mathbf{Y}$. It remains, of course, to show how this trajectory approximates the minimal-time, δ_v-safe solution, and that $\mathcal{G}$ has polynomial size in $(\frac{1}{\epsilon})$.

3.3 The Algorithm

Our algorithm first chooses a time-step τ as a function of a_{max}, ϵ, c_0, and c_1; we derive an upper bound for τ in Lemma 4.5. Next, the algorithm chooses the TC-gridpoint $\mathbf{S}^*$ that most closely approximates $\mathbf{S}$. It then searches for the shortest path in the graph $\mathcal{G}$ from $\mathbf{S}^*$ to any vertex that is within $(\frac{a_{max}\tau^2}{2}, \frac{a_{max}\tau}{2})$ of G. We use a breadth-first search to explore the graph. Safety checking, which is needed to determine whether an (a_{max}, τ)-bang is an edge in the graph, is done for each bang during the search.

We claim that if an optimal δ_v-safe trajectory Γ_{opt} from $\mathbf{S}$ to $\mathbf{G}$ takes time T_{opt}, then the algorithm will find a δ_v'-safe trajectory that approximates $\mathbf{S}$ and $\mathbf{G}$ to within $(\frac{a_{max}\tau^2}{2}, \frac{a_{max}\tau}{2})$ and that takes time at most T_{opt}. To show that this is correct we prove a lemma that there is a τ proportional to ϵ such that there exists an (a_{max}, τ)-bang trajectory Γ_q that approximates Γ_{opt} closely enough so that it is δ_v'-safe and that $\Gamma_q(0)$ and $\Gamma_q(T_{opt})$ are within $(\frac{a_{max}\tau^2}{2}, \frac{a_{max}\tau}{2})$ of $\mathbf{S}$ and $\mathbf{G}$.

The maximal out-degree in $\mathcal{G}$ is 3^d; this is the branching factor of the search. The number of TC-gridpoints is proportional to $\frac{lv_{max}}{a^2\tau^3}$. Safety-checking is $O(n^2(\log n))$ $(d = 3)$ per (a_{max}, τ)-bang, as is [CDRX]. Thus we get the complexity bound $O(n^2(\log n)(\frac{1}{\epsilon})^{3d})$.

The algorithm can be adapted to solve the approximately optimal single-source kinodynamic planning problem by terminating the search when it finds no new states, instead of terminating it when it first finds a vertex within $(\frac{a_{max}\tau^2}{2}, \frac{a_{max}\tau}{2})$ of G. To solve the single-sink problem, we compute the TC-gridpoint $\mathbf{G}^*$ closest to $\mathbf{G}$, and do a breadth-first search of the graph, *following the edges in $\mathcal{E}$ in their reverse direction.* The algorithms for these two problems are no more expensive than the state-state algorithm in the worst case, since a backpointer must be kept for each state explored so that the algorithm can return a solution.

4 Better Bounds for a Cartesian Robot with L_∞ Dynamics Bounds

4.1 Overview

We now outline the argument proving the parts of Theorem 2.1 concerning the goodness of the approximation and the running-time of the algorithm (except for safety-checking). First, we need two definitions.

definition .1 *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* $\mathbf{p}$.

definition .2 *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,

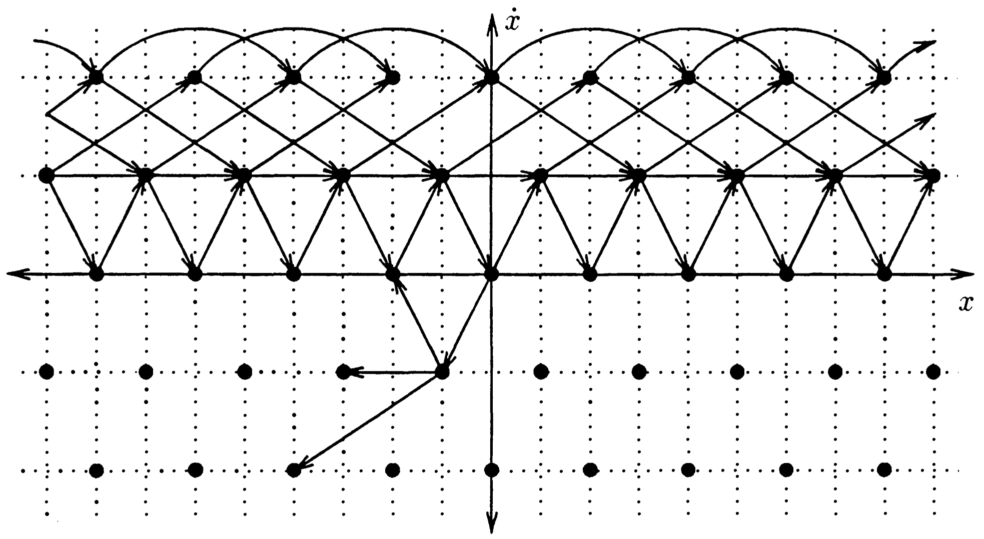


Figure 3: A section of a phase space grid in 1D, centered about $(0, 0)$. The neighbor relation of the TC-grid is partially shown by the arcs between states.

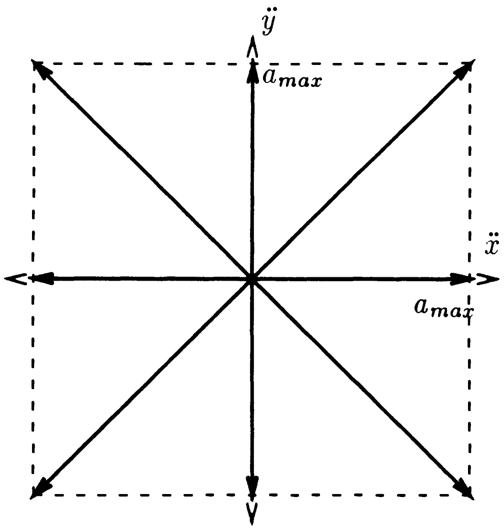
$$\begin{array}{rcll} \|\mathbf{p}_a(t) - \mathbf{p}_b(t)\|_\infty & \leq & \eta_x, & \textit{and} \\ \|\dot{\mathbf{p}}_a(t) - \dot{\mathbf{p}}_b(t)\|_\infty & \leq & \eta_v. & \end{array} \tag{6}$$

Suppose that $\Gamma_r(t) = (\mathbf{p}_r(t), \dot{\mathbf{p}}(t))$ is a trajectory from $\mathbf{S}$ to $\mathbf{G}$ taking time T_r, and suppose furthermore that Γ_r obeys bounds a_{max} and v_{max} and is $\delta_v(c_0, c_1)$-safe. Let $0 < \epsilon < 1$.

In Section 4.2, we show that if trajectory Γ_q tracks Γ_r to a tight enough tolerance (η_x, η_v), then Γ_q will be $(1 - \epsilon)\delta_v(c_0, c_1)$-safe. (Lemma 4.1.)

In Section 4.3 we show how to choose τ (given (η_x, η_v), a_{max}, and v_{max}) so that there will be some (a_{max}, τ)-bang trajectory Γ_q that tracks Γ_r to tolerance (η_x, η_v). This choice of τ is independent of $\mathbf{S}$ and $\mathbf{G}$, and defines the τ chosen by the algorithm. (Tracking Lemma.)

In Section 4.4 we show how to choose a sufficiently small (η_x, η_v) such that this τ will still be proportional to ϵ. It follows that the number of TC-gridpoints is polynomial in $O\left(\left(\frac{1}{\epsilon}\right)^{3d}\right)$. Since each TC-graph vertex has a maximum out-degree of 3^d, and checking the $(1 - \epsilon)\delta_v(c_0, c_1)$-safety of an (a_{max}, τ)-bang is $O(n^2(\log n))$ (in 3D), it follows that searching the TC-graph for a shortest path from $\mathbf{S}^*$ to $\mathbf{G}^*$ in the TC-graph takes time $O\left(n^2(\log n)\left(\frac{1}{\epsilon}\right)^{3d}\right)$. Details of the proofs give us Theorem 2.1.



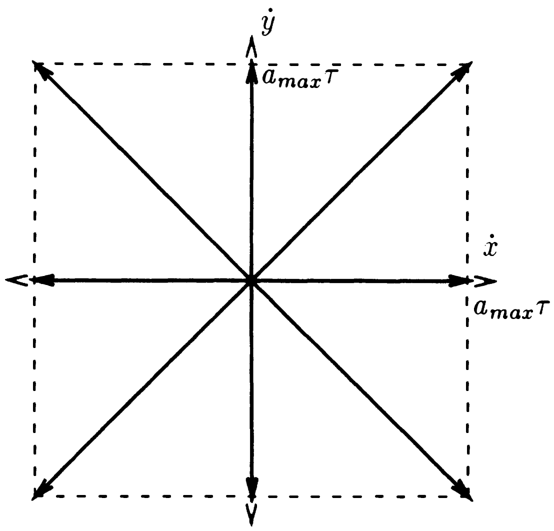


Figure 4: For L_∞ dynamics bounds, from [CDRX]: extremal accelerations (left) that generate (a_{max}, τ)-bangs (right).

For a discussion of how to do safety-checking in time $O(n^2(\log n))$ in 3D and $O(n)$ in 2D, see [CDRX].

4.2 Safe Tracking

Lemma 4.1 (The Safe Tracking Lemma) *Suppose that δ_v is specified by c_0 and c_1 and $0 < \epsilon < 1$, and let $\delta_v{}' = (1-\epsilon)\delta_v$, and that Γ_r is a δ-safe trajectory. Then there exists a tolerance (η_x, η_v) such that for any trajectory Γ_q, the following hold:*

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

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

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

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 $t_x \in [0, T_r]$,

$$\|\mathbf{x} - \mathbf{p}_q(t_x)\| < c_0' + c_1'\|\dot{\mathbf{p}}_q(t_x)\|. \tag{8}$$

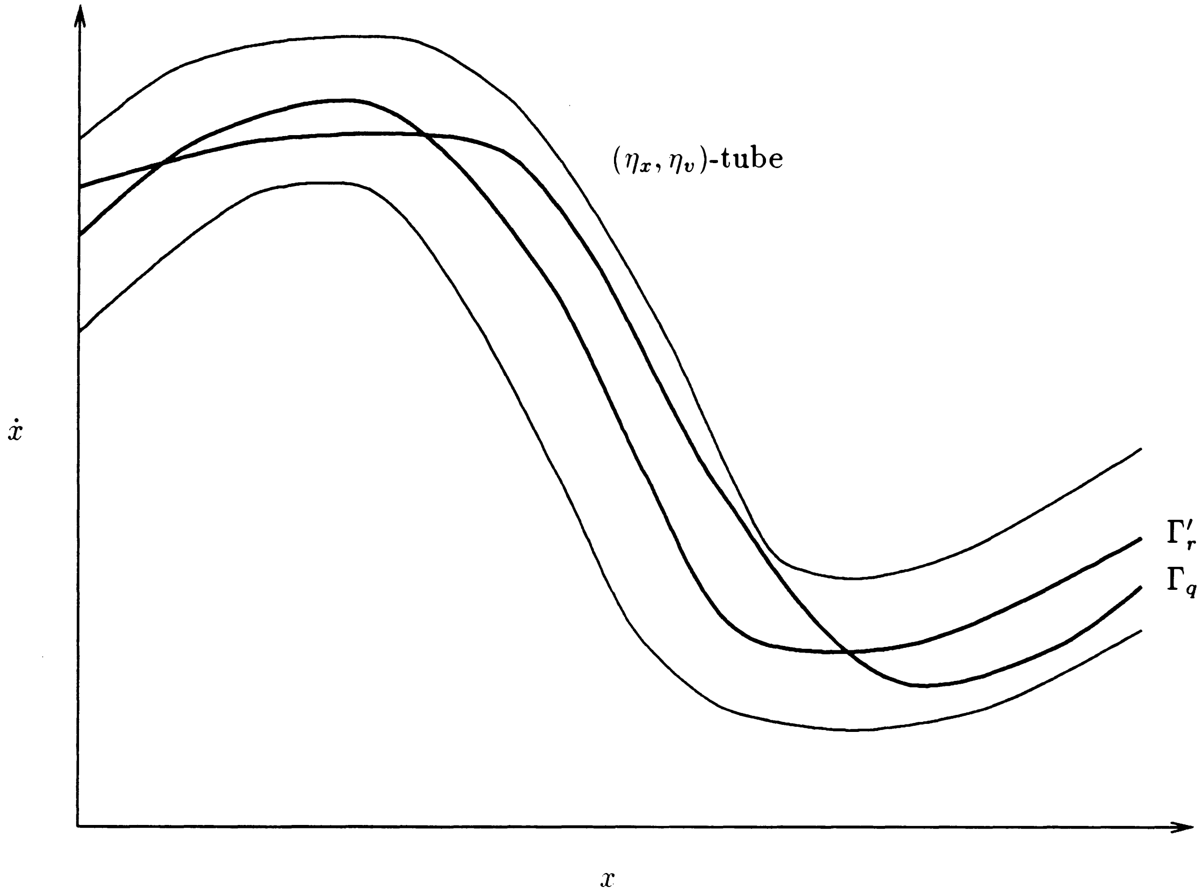


Figure 5: An (η_x, η_v)-tube projected into phase-space. Trajectory Γ_q approximately tracks Γ'_r to tolerance (η_x, η_v).

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

$$\begin{aligned} \|\mathbf{x} - \mathbf{p}_r(t_x)\| &\leq \|\mathbf{x} - \mathbf{p}_q(t_x)\| + \eta_x, \quad \text{and} \\ \|\dot{\mathbf{p}}_q(t_x)\| &\leq \|\dot{\mathbf{p}}_r(t_x)\| + \eta_v. \end{aligned} \tag{9}$$

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 (8) and adding η_x to the right-hand side,

$$\|\mathbf{x} - \mathbf{p}_r(t_x)\| < c'_0 + \eta_x + c'_1(\|\dot{\mathbf{p}}_r(t_x)\| + \eta_v). \tag{10}$$

Now, suppose that $\beta > 0$ and that η_x and η_v satisfy (7). It follows that $\eta_x + (1-\epsilon)c_1\eta_v \leq \epsilon c_0$, and thus

$$c_0' + \eta_x + c_1'\eta_v \leq c_0. \tag{11}$$

But then,

$$c_0' + \eta_x + c_1'\left(\left\|\dot{\mathbf{p}}_r\left(\frac{t_x}{1+\epsilon}\right)\right\| + \eta_v\right) \leq c_0 + c_1\left\|\dot{\mathbf{p}}_r\left(\frac{t_x}{1+\epsilon}\right)\right\|. \tag{12}$$

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. □

Note that this lemma is independent of norm.

4.3 The Strong Tracking Lemma

4.3.1 Preliminary Discussion of the Lemma

Our tighter approximation and time-complexity bounds for the [CDRX] rely on the following lemma.

Lemma 4.2 (The Strong Tracking Lemma) *Suppose a trajectory Γ_r respects acceleration bounds a_{max} and takes time T_r respecting all kinodynamic bounds as above. Then in the absence of obstacles, the following hold.*

(a) For any positive η_x, η_v, there exists a time spacing τ, a choice of position space grid alignment, and an (a_{max}, τ)-grid-bang trajectory Γ_q with bounded acceleration a_{max} that approximately tracks Γ_r to tolerance (η_x, η_v) in time T_r such that $p_q(0) = p_r(0)$, $v_q(0) - v_r(0) < a_{max}\tau$, $|p_q(T_r) - p_r(T_r)| < 3a_{max}\tau^2$, and $|v_q(T_r) - v_r(T_r)| < 2a_{max}\tau$.

(b) Moreover, τ is polynomial in η_x, η_v, and $\frac{v_{max}}{a}$. Specifically, τ can be chosen as:

$$\tau \leq \min\left(\frac{\eta_x}{5v_{max}}, \frac{\eta_v}{2a}, \frac{v_{max}}{2a}\right) \tag{13}$$

(c) Finally, for a given choice of τ, $\Gamma_q(0) = \Gamma_r(0)$ and $\Gamma_q(T_r) = \Gamma_r(T_r)$ if and only if $\Gamma_q(0)$ and $\Gamma_r(T_r)$ are gridpoints and T_r is a multiple of $a_{max}\tau$.

As in [CDRX] it will be sufficient to consider the one-dimensional case, since we are using the L_∞-norm for dynamics bounds. Let v_{max}, a_{max}, and τ be given such that $\frac{v_{max}}{(a\tau)}$ is integral. (We subscript v_{max} because we discuss several velocity functions; the only acceleration mentioned is the bound a_{max}.) Assume that trajectory Γ_r obeys the velocity and acceleration bounds. *Wlog*, let $T_r = N\tau$ for some integer N. Let us call a function that describes the velocity of some grid-bang trajectory a *grid-bang velocity function.*

The approach involves four stages. First. in Section 4.3.3 we fix τ, and given an arbitrary trajectory Γ_r obeying the dynamics bounds, we show there exists a pair of grid-bang velocity functions bounding v_r from above and below while staying within a constant of v_r. (See figure 6.) Second, in Section 4.3.4 using the bounding trajectories we show there exists a grid-bang trajectory Γ_{greedy} that tracks Γ_r to a tolerance that is a function of the dynamics bounds and τ; Γ_{greedy} approximates Γ_r to within $(0, a\tau)$ at time 0. (See figure 8.) Then, using Γ_{greedy}, we show a lemma that there exists a grid-bang trajectory Γ_q tracks Γ_r only slightly less closely but approximates it better at T_r. (See figure 9.) We apply this last lemma to obtain the Strong Tracking Lemma.

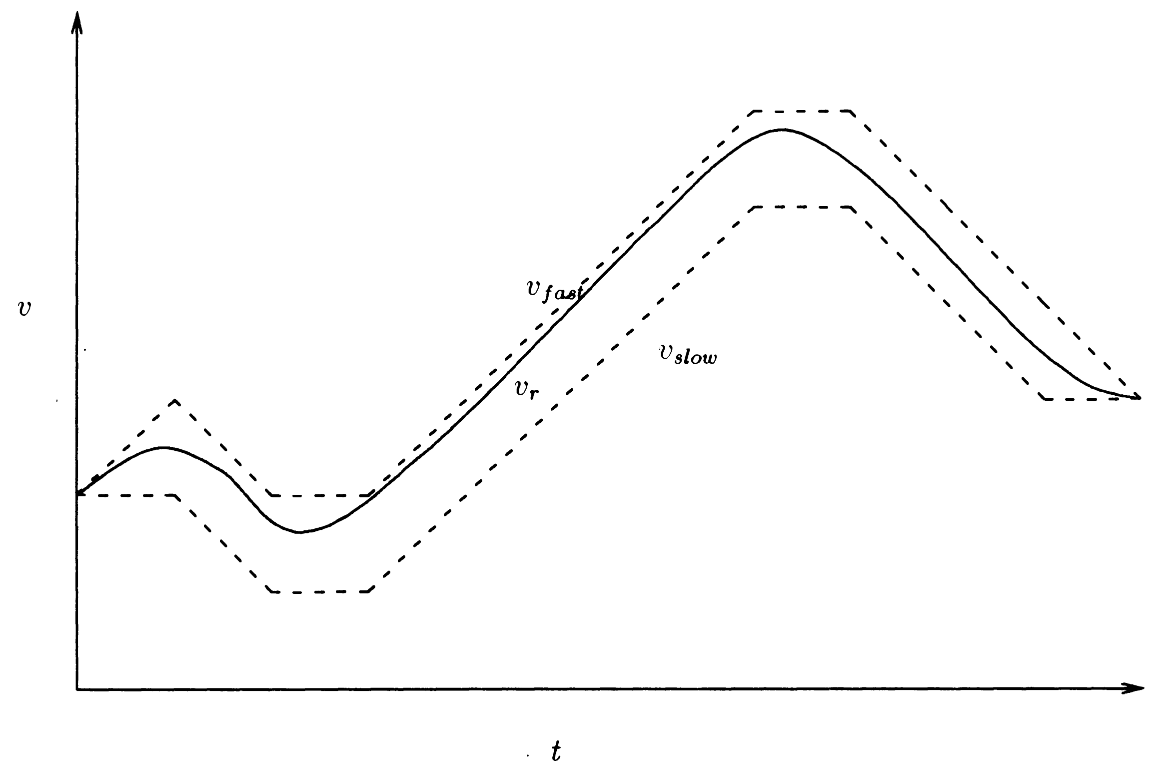


Figure 6: v_r is bounded above and below by (a_{max}, τ)-bang velocity functions v_{fast} and v_{slow}, which stay within $a_{max}\tau$ of v_r.

4.3.2 Describing a Grid-Bang Trajectory

Observe that a that a grid-bang trajectory Γ lasting for $N\tau$ is uniquely described by the initial state $(\mathbf{p}(0), \dot{\mathbf{p}}(0))$ and a sequence of vectors $\{\mathbf{c}_0, \ldots, \mathbf{c}_{N-1}\}$ whose individual components are members of $\{-1, 0, 1\}$. Let $\mathbf{A}$ be the diagonal matrix of acceleration bounds; that is, $\mathbf{A}_{ii}$ is the acceleration bound in the i^{th} coordinate direction. Then we have for for $t \in [n\tau, (n+1)\tau]$:

$$\begin{aligned} \mathbf{v}(t) &= \mathbf{v}(n\tau) + \tfrac{(t-n\tau)}{\tau}\mathbf{A}\mathbf{c}_n \\ \mathbf{p}(t) &= \mathbf{p}(n\tau) + \textstyle\int_{n\tau}^{t} \mathbf{v}(s)ds. \end{aligned} \tag{14}$$

4.3.3 A Simple Lemma

Lemma 4.3 *If the hypotheses of Lemma 4.2 are met, then there exist two grid-bang velocity functions $v_{slow} : [0, N\tau] \to [-v_{max}, v_{max}]$ and $v_{fast} : [0, N\tau] \to [-v_{max}, v_{max}]$ that meet the following three conditions:*

1. $v_r(0) - a_{max}\tau < v_{slow}(0) \le v_r(0) \le v_{fast}(0) < v_r(0) + a_{max}\tau$, *and* $v_r(T_r) - a_{max}\tau < v_{slow}(T_r) \le v_r(T_r) \le v_{fast}(T_r) < v_r(T_r) + a_{max}\tau$,
2. $v_{slow}(t) \le v_r(t) \le v_{fast}(t)$ *for all* $t \in [0, T_r]$, *and*
3. $v_{fast}(t) - v_r(t) \le 2a_{max}\tau$ *and* $v_r(t) - v_{slow}(t) \le 2a_{max}\tau$ *for all* $t \in [0, T_r]$.

Furthermore, if $v_r(0)$ and $v_r(T_r)$ are multiples of $a_{max}\tau$, then $v_{slow}(0) = v_r(0) = v_{fast}(0)$ and $v_{slow}(T_r) = v_r(T_r) = v_{fast}(T_r)$.

Proof: Given v_r, we show by construction the existence of two functions for which 2) and 3) hold; we then modify the constructed functions so that 1) also holds. Let us define rationals u_n and w_n for integers $n \in [0, N-1]$:

$$\begin{aligned} u_n &= \left\lfloor \tfrac{\min_t v_r(t)}{a_{max}\tau} \right\rfloor a_{max}\tau \\ w_n &= \left\lceil \tfrac{\max_t v_r(t)}{a_{max}\tau} \right\rceil a_{max}\tau, \qquad \text{where } t \in [n\tau, (n+1)\tau] \end{aligned} \tag{15}$$

Let $\tilde{v}_{slow}$ and $\tilde{v}_{fast}$ be defined as follows:

$$\begin{aligned} \tilde{v}_{slow}(0) &= \left\lfloor \tfrac{v_r(0)}{a_{max}\tau} \right\rfloor a_{max}\tau, \\ \tilde{v}_{slow}(t) &= \tilde{v}_{slow}(n\tau) + \tfrac{t-n\tau}{\tau}(u_n - \tilde{v}_{slow}(n\tau)) \qquad \text{for} \quad t \in [n\tau, (n+1)\tau], \\ \tilde{v}_{fast}(0) &= \left\lceil \tfrac{v_r(0)}{a_{max}\tau} \right\rceil a_{max}\tau, \\ \tilde{v}_{fast}(t) &= \tilde{v}_{fast}(n\tau) + \tfrac{t-n\tau}{\tau}(w_n - \tilde{v}_{fast}(n\tau)) \qquad \text{for } t \in [n\tau, (n+1)\tau]. \end{aligned} \tag{16}$$

It is easy to see that $\tilde{v}_{slow}$ and $\tilde{v}_{fast}$ satisfy the second and third conditions of lemma 4.3. If the first condition is not met, then either (a)$\tilde{v}_{slow}(T_r) \le v_r(T_r) - a_{max}\tau$ or (b) $\tilde{v}_{fast}(T_r) \ge v_r(T_r) + a_{max}\tau$. If (a) is the case then $v_{fast} = \tilde{v}_{fast}$, and we alter $\tilde{v}_{slow}$ to get v_{slow}; if (b) is the case then $v_{slow} = \tilde{v}_{slow}$, and we alter $\tilde{v}_{fast}$ to get v_{fast}.

We only describe the construction for the latter case; the former is similar. Let $\tilde{v}_{fast}(0)$ and $\{\tilde{c}_i^{fast}\}$ describe $\tilde{v}_{fast}$, and let m be the least integer for which $\tilde{v}_{fast}(k\tau) \ge$

$v_r(k\tau) + a_{max}\tau$ for all integers $k \in [m, N]$. Then either (a) $\tilde{c}^{fast}_{m-1} = 1$ and $\tilde{v}_{fast}((m-1)\tau) - v_r((m-1)\tau \leq a_{max}\tau$, or (b) $\tilde{c}^{fast}_{m-1} = 0$. In the case of (a), for some $t \in [(m-1)\tau, m\tau]$, $v_r(t) \geq \tilde{v}_{fast}((m-1)\tau)$; if k is the greatest integer less than m such that $c_k \neq 1$, then v_{fast} has parameters identical to those of $\tilde{v}_{fast}$, except that $c^{fast}_k = \tilde{c}^{fast}_k + 1$ and $c^{fast}_{m-1} = -1$. (See fig. 4.3.3 (a).) In the case of (b), for some $t \in [(m-1)\tau, m\tau]$, $\tilde{v}_{fast}((m-1)\tau) - a_{max}\tau \leq v_r(t) < \tilde{v}_{fast}((m-1)\tau)$; v_{fast} has parameters identical to those of $\tilde{v}_{fast}$, except that $c^{fast}_{m-1} = -1$. (See fig. 4.3.3 (b).) This construction also is sufficient to establish the "furthermore". □

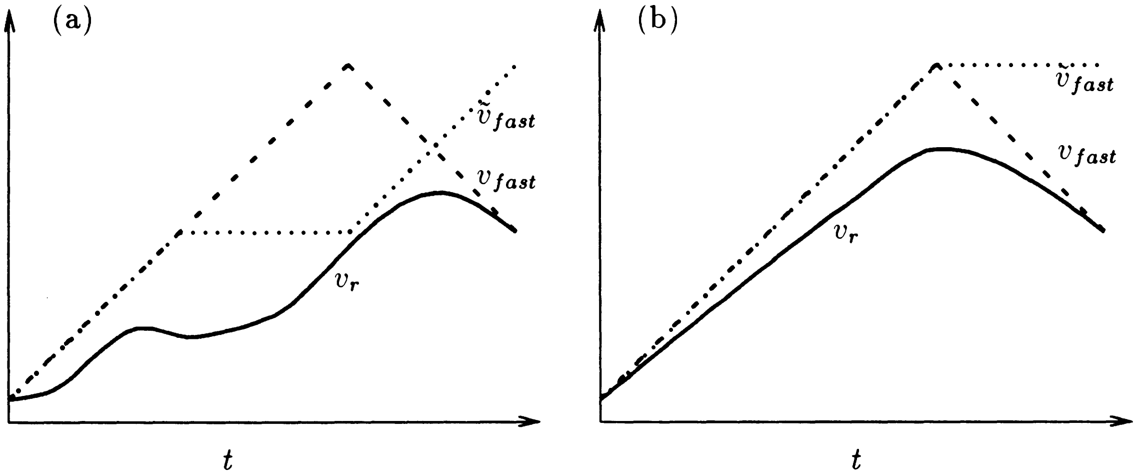


Figure 7: Examples of the two cases where $\tilde{v}_{fast}$ is altered to obtain v_{fast}.

4.3.4 Greedy Tracking

We now iteratively define a (a_{max}, τ)-bang trajectory Γ_{greedy} that attempts to track Γ_r. Γ_r is inductively definable via the following; there are $N+1$ stages. Note that while we have already implicitly chosen the grid alignment in the velocity dimensions of state space, we have not done so for the position dimensions; we chose the *position space alignment* of the grid in the construction.

- Stage $0'$: $p_{greedy}(0) = p_r(0)$. If $v_r(0)$ is a multiple of $a_{max}\tau$, $v_{greedy}(0) = v_r(0)$. Otherwise, if $v_r(T_r) \leq v_r(0)$, then $v_{greedy}(0) = \lfloor v_r(0) \rfloor$; if $v_r(T_r) > v_r(0)$, then $v_{greedy}(0) = \lceil v_r(0) \rceil$. The origin of the grid is chosen so that $(p_{greedy}(0), v_{greedy}(0))$ is a gridpoint. In this manner, $p_r(0)$ induces the *position space alignment* of the grid.

- Stage n: We define $v_{greedy}(t)$ for all $t \in [n\tau, (n+1)\tau]$. We choose $c_n \in \{-1, 0, 1\}$, and set $v_{greedy}(t) = v_{greedy}(n\tau) + c_n a_{max}(t - n\tau)$. We now describe how c_n is chosen.

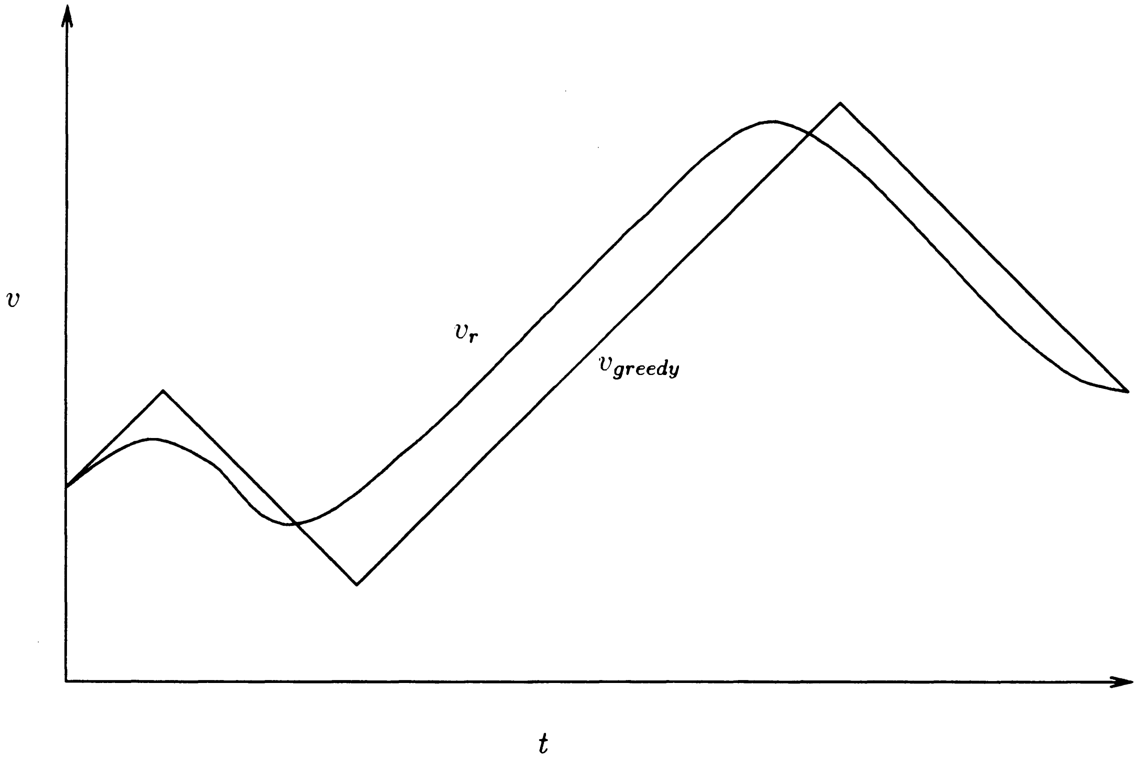


Figure 8: v_{greedy} stays within $a_{max}\tau$ of v_r. By construction v_{greedy} is bounded above and below by (a_{max}, τ)-bang velocity functions v_{fast} and v_{slow}, which stay within $a_{max}\tau$ of v_r.

---

1. c_n is chosen to minimize $|p_r((n+1)\tau)-p_{greedy}((n+1)\tau)|$ subject to the condition $v_{slow}(t) \leq v_{greedy}(t) \leq v_{fast}(t)$, given $c_0, \ldots, c_{n-1}$.
2. Let the lag $y = p_r((n+1)\tau) - p_{greedy}((n+1)\tau)$. If $|y| > 2v_{max}\tau$, then (a) for the greatest integer $m < n$ such that $c_m \neq sgn(y)$, set $c_m = c_m + sgn(y)$, and (b) if j, $m < j \leq n$, is the integer such that setting $c_j = c_j - sgn(y)$ minimizes $|p_r((n+1)\tau) - p_{greedy}((n+1)\tau)|$ subject to the constraint that $|p_{greedy}(t) - p_r(t)| \leq 2v_{max}\tau + a_{max}\tau^2$, then set $c_j = c_j - sgn(y)$. To see this is consistent, observe that $|y| > 2v_{max}\tau$ implies that for $k = m, m+1, m+2$,

$$sgn(y)(p_{greedy}(k\tau) - p_r(k\tau)) \leq 2v_{max}. \tag{17}$$

By this definition, for all $t \in [0, T_r]$, $v_{slow}(t) \leq v_{greedy}(t) \leq v_{fast}(t)$. Furthermore, we can show that Γ_{greedy} tracks Γ_r to a constant tolerance that is linear in the acceleration and velocity bounds and quadratic in timestep size. Using this property will allow us to prove the Strong Tracking Lemma.

Lemma 4.4 *For all times $t \in [0, T_r]$, $|p_r(t) - p_{greedy}(t)| < 2v_{max}\tau + 2a_{max}\tau^2$.*

Proof: Suppose not. First of all, we observe that in the definition of Γ_{greedy}, Stage n never causes a violation at time $t < n\tau$. Then, let t_w be the least time such that $|p_r(t_w) - p_{greedy}(t_w)| > 2v_{max}\tau + 2a_{max}\tau^2$. *Wlog* suppose that the sign of the difference is positive; the negative case is similar. Since $p_r(t_w) - p_{greedy}(t_w) > 2v_{max}\tau$, there exists a least integer $m < n$ such that (a) for all $t \in [m\tau, t_w]$, $v_r(t) > v_{greedy}(t)$, and (b) for all integers j such that $m \leq j < \lfloor \frac{t_w}{\tau} \rfloor$, $c_j = 1$ and $v_r(m\tau) - v_{greedy}(m\tau) \leq a_{max}\tau$.

$m\tau$ is basically an upper bound on when Γ_q starts falling further and further behind Γ_r. Thus during $[m\tau, \lfloor \frac{t_w}{\tau} \rfloor \tau]$, Γ_r must be accelerating positive maximally; such a period of maximal acceleration can last at most $\frac{2v_{max}}{a}$.

It follows from the definition of Γ_{greedy} that for all $t \in [m\tau, \lfloor \frac{t_w}{\tau} \rfloor \tau)$, $0 < v_r(t) - v_{greedy}(t) < a_{max}\tau$. Then either (a) $p_r(m\tau) - p_{greedy}(m\tau) \leq \frac{(a_{max}\tau^2)}{2}$ and $\lfloor \frac{t_w}{\tau} \rfloor \tau - m \leq \frac{2v_{max}}{a_{max}\tau}$, or (b) $p_r(m\tau) - p_{greedy}(m\tau) \leq 2a_{max}\tau^2$ and $t_w - m\tau < \frac{2v_{max}}{a} - \tau$. The disjunction follows from an examination of the definition of Γ_{greedy}. As result, $p_r(t_w) - p_{greedy}(t_w) < 2v_{max}\tau + 2a_{max}\tau^2$, and we have a contradiction. □

Using Γ_{greedy} given above, we can prove a lemma showing the existence of a trajectory that tracks Γ_r a little less closely but that is within $(3a_{max}\tau^2, a_{max}\tau)$ of $\Gamma_r(T_r)$ at time T_r.

Lemma 4.5 *If the hypotheses from the beginning of this section hold then the following are true.*

(a) There exists a grid-bang Γ_q that tracks Γ_r to tolerance $(4v_{max}\tau + 2a_{max}\tau^2, 2a_{max}\tau)$ such that $p_q(0) = p_r(0)$, $|p_q(T_r) - p_r(T_r)| < a_{max}\tau^2$, and $|v_q(T_r) - v_r(T_r)| < a_{max}\tau$.

(b) Furthermore, if $\Gamma_r(0)$ and $\Gamma_r(T_r)$ are gridpoints, then there exists a grid-bang Γ_q such that $\Gamma_q(0) = \Gamma_r(0)$ and $\Gamma_q(T_r) = \Gamma_r(T_r)$ in addition to the above.

Proof: It follows from lemmas 4.3 and 4.4 and the definition of v_{greedy} that the grid-bang trajectory Γ_{greedy} tracks Γ_r to tolerance $(2v_{max}\tau + 2a_{max}\tau^2, 2a_{max}\tau)$, that $\Gamma_r(0) = \Gamma_{greedy}(0)$, and that $|v_r(T_r) - v_{greedy}(T_r)| < a_{max}\tau$. *Wlog* assume that $p_r(T_r) > p_{greedy}(T_r)$; the "$<$" case is similar. We show by construction how Γ_{greedy} can be modified into a Γ_q that satisfies the lemma. Each stage n in the following "algorithm" generates trajectory $\Gamma_q^{(n)}$.

- Stage 0: Let $\Gamma_q^{(0)} = \Gamma_{greedy}$.

- Stage n: We obtain $\Gamma_q^{(n)}$ by modifying $v_q^{(n-1)}$. Let m be the greatest integer such that $v_q^{(n-1)}(m\tau) < v_{fast}(m\tau)$. Let k be the largest integer such that $k < m$, $c_k^{(n-1)} \neq 1$, and $v_q^{(n-1)}((k+1)\tau) < v_{fast}((k+1)\tau)$. Set $c_k^{(n)} = c_k^{(n-1)} + 1$. Then, choose the integer j, $k < j \leq m$, that minimizes $|p_r(T_r) - p_q^{(n)}(T_r)|$ when $\Gamma_q^{(n)}$ is obtained by setting $c_j^{(n)} = c_j^{(n-1)} - 1$ and leaving the other $c_i^{(n)}$ unchanged from $c_i^{(n-1)}$. Set $c_j^{(n)} = c_j^{(n-1)} - 1$, and set $c_i^{(n)} = c_i^{(n-1)}$ for all i such that $i \neq k$ and $i \neq j$ to complete the definition of $\Gamma_q^{(n)}$. If $p_r(T_r) = p_q^{(n)}(T_r)$. then $\Gamma_q = \Gamma_q^{(n)}$; if $\Gamma_r(T_r)$ is not a gridpoint and $|p_r(T_r) - p_q^{(n)}(T_r)| \leq a_{max}\tau^2$, then $\Gamma_q = \Gamma_q^{(n)}$; otherwise go to the next stage.

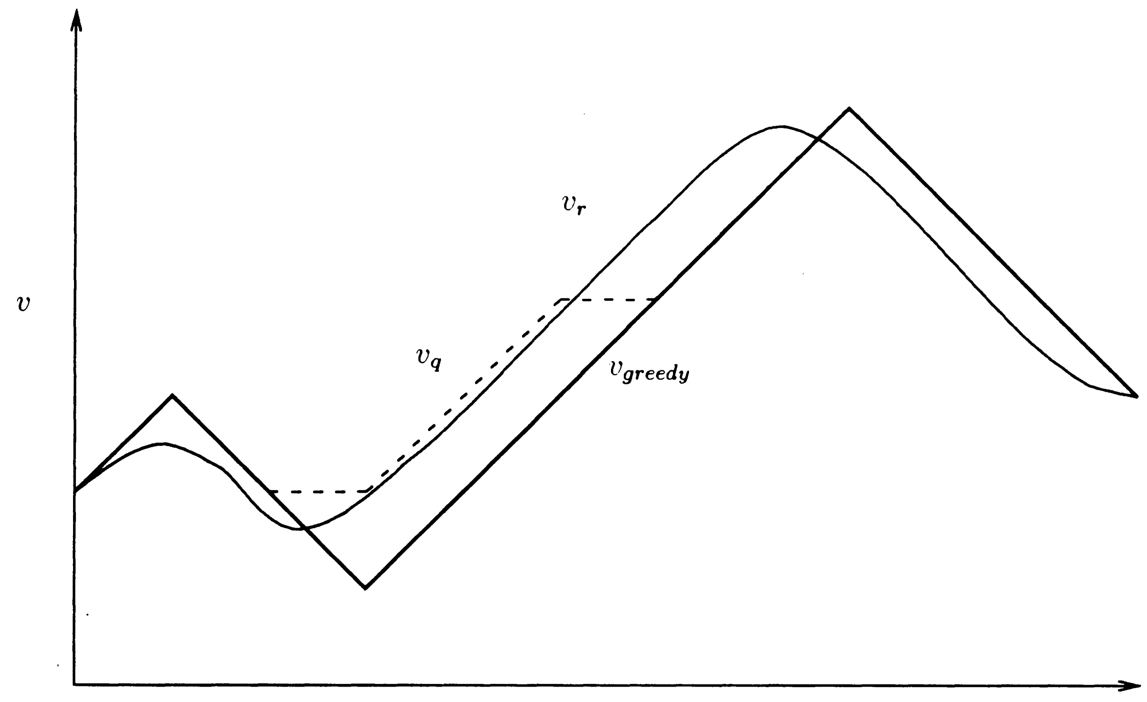


Figure 9: v_{greedy} is modified to v_q so that $\int_0^{t_r} v_q(t)dt$ is within $a_{max}\tau^2$ of $\int_0^{t_r} v_r(t)dt$.

Note that in the first stage where it is possible to choose j so that $p_q^{(n)}(T_r)$ exceeds $p_r(T_r)$ it is also possible to choose j so that $p_r(T_r) = p_q^{(n)}(T_r)$ (assuming that $p_r(T_r)$ is a gridpoint). Thus, the corrections change $p_q^{(n)}(T_r)$ monotonically (as opposed to oscillating). The choice of $\Gamma_{greedy}(0)$ guarantees that Γ_{greedy} can be modified so that eventually $p_q^{(n)}(T_r)$ is within the claimed tolerance of $p_r(T_r)$. Then, since for each stage n, $p_q^{(n+1)}(T_r) \geq p_q^{(n)}(T_r) + a_{max}\tau^2$, the construction terminates in at most $\frac{(2v_{max})}{a} + 2$ stages.

Observe that at a given stage n, for all $t \in [m\tau, T_r]$, $p_q^{(n-1)}(t) < p_r(t)$, and at that stage $m - k \leq \frac{2v_{max}}{a}$. It follows that it is sufficient to increase the positional tracking tolerance by $2v_{max}\tau$. Since $v_{slow}(t) \leq v_q(t) \leq v_{fast}(t)$ when $0 \leq t \leq T_r$, it follows that the velocity component of the tolerance is $2a_{max}\tau$. $\square$

4.3.5 Strong Tracking

To prove lemma 4.2 we need the following simple corollary below.

Corollary 4.6 *If T_r is not a multiple of τ, then there exists a grid-bang Γ_q that tracks Γ_r to tolerance $(4v_{max}\tau + 2a_{max}\tau^2, 2a_{max}\tau)$ such that $p_q(0) = p_r(0)$, $|p_q(T_r) - p_r(T_r)| < 3a_{max}\tau^2$,* *and* $|v_q(T_r) - v_r(T_r)| < 2a_{max}\tau$.

Proof: Extend Γ_r arbitrarily while obeying dynamics bounds to get a trajectory Γ_r^* lasting for time $T_r^* = \lceil \frac{T_r}{\tau} \rceil \tau$. Apply Lemma [grid-tracking] to get Γ_q. $|v_q(T_r) - v_r(T_r)| < 2a_{max}\tau$ follows from the tracking tolerance, and $|p_q(T_r) - p_r(T_r)| < 3a_{max}\tau^2$ follows from applying $|p_q(T_r^*) - p_r'(T_r^*)| < a_{max}\tau^2$ and the velocity part of the tracking tolerance together. □

We now remove the restriction that τ divide T_r and use the lemmas and corollary above to prove lemma 4.2.

Proof (of lemma 4.2): We apply lemma 4.5 and corollary 4.6 to obtain the sufficient condition $4v_{max}\tau + 2a_{max}\tau^2 \leq \eta_x$ and $2a_{max}\tau \leq \eta_v$. Requiring that $\tau \leq \frac{v_{max}}{2a}$ and substituting yields (a), (b), and the "if" case from (c). By applying lemmas 4.3 and 4.4 we obtain (a) and (b) in the non-gridpoint case; note that although Γ_q might be defined for a time greater than T_r, $|p_q(T_r) - p_r(T_r)| \leq 3a_{max}\tau^2$ and $|v_q(T_r) - v_r(T_r)| \leq 2a_{max}\tau$. □

With some simple calculation we can show that if the dynamics bounds, $\Gamma_r(0)$, $\Gamma_r(T_r)$, and T_r can be given in rational numbers, then it possible to choose τ so that T_r is a multiple of τ, v_{max} is a multiple of $a_{max}\tau$, and $\Gamma_r(0)$ and $\Gamma_r(T_r)$ lie on the grid. However, $\frac{1}{\tau}$ will depend on the least common denominator of a_{max} and the components of $\Gamma_r(0)$ and $\Gamma_r(T_r)$.

4.4 A δ_v'-Safe ("Also Safe") Grid-Bang Trajectory

Recall that by lemma 4.2, τ is polynomially dependent on η_x and η_v. Applying lemmas 4.2 and 4.1 and choosing β to maximize the upper bound on τ yields the following theorem:

Theorem 4.7 *Given acceleration bounds a_{max}, obstacles $\mathcal{O}$, and positive scalars $\epsilon \leq 1$, c_0, and c_1, for any $\delta_v(c_0, c_1)$-safe trajectory Γ_r taking time T_r, there exists a time spacing τ and an (a_{max}, τ)-grid-bang, δ_v'-safe trajectory Γ_q with the following properties:*

1. $\mathbf{p}_q(0) = \mathbf{p}_r(0)$;
2. $|\mathbf{v}_q(0) - \mathbf{v}_r(0)| \leq a_{max}\tau$;
3. $|\mathbf{p}_q(T_r) - \mathbf{p}_r(T_r)| \leq 3a_{max}\tau^2$;
4. $|\mathbf{v}_q(T_r) - \mathbf{v}_r(T_r)| \leq 2a_{max}\tau$;
5. *If τ divides T_r and Γ_r begins and ends on the (a_{max}, τ)-grid, then the above are valid with 0 substituted for the right-hand terms in the inequalities.*

In particular, the following choice of τ suffices:

$$\tau \leq \min\left(\frac{c_0\epsilon}{2ac_1(1-\epsilon) + 5v_{max}}, \frac{v_{max}}{2a}\right). \tag{18}$$

Proof: Suppose Γ_r is a $\delta_v(c_0, c_1)$-safe trajectory taking time T_r obeying acceleration bound a_{max}. Then by lemma 4.1 the choice of a tolerance (η_x, η_v) given in (7) 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 $\mathcal{O}$, Γ_q is therefore δ'_v-safe. Given the tolerance (η_x, η_v), by lemma 4.2 there is a time-spacing τ such that some(a_{max}, 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 (18) yields a maximal τ as given by (13). Let us therefore define for $\beta \geq 0$

$$\begin{aligned} \tau_x(\beta) &= \tfrac{c_0\epsilon\beta}{5v_{max}(c_1(1-\epsilon)+\beta)} \\ \tau_v(\beta) &= \tfrac{c_0\epsilon}{2a(c_1(1-\epsilon)+\beta)} \\ \tau(\beta) &= \min(\tau_x(\beta), \tau_v(\beta)). \end{aligned} \tag{19}$$

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

$$\beta = \frac{2v_{max}}{5a}. \tag{20}$$

Applying either τ_x or τ_v to this β yields the desired τ in (18). □

4.5 Final Bounds

To determine the total complexity of the algorithm we must bound the number of (a_{max}, τ)-gridpoints for a Cartesian robot with maximum (L_∞) speed v_{max} in a d-dimensional free-space of diameter l. We so this by simply calculating the number of gridpoints reachable under grid-bang commanded motion [CDRX]. Let $G_\infty(a_{max}, \tau, v_{max}, l, d)$ denote this bound. Then

$$G_\infty(a_{max}, \tau, v_{max}, l, d) = \left(\left(\frac{2v_{max}}{a_{max}\tau} + 1\right)\left(\frac{l}{a\tau^2} + 1\right)\right)^d \tag{21}$$

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_{max}, τ)-grid-bang path, while avoiding obstacles as prescribed by a safety function δ'_v, in polynomial time. Noting that checking the safety of trajectory segments requires time $O(n^2 \log(n))$ [CDRX], we have shown Theorem 2.1.

The computed trajectory Γ_q satisfies the time approximation $T_q \leq T$, in addition to respecting the kinodynamic constraints and being δ'_v-safe. Note that by Theorem 4.7 we can show that if **S** and **G** lie on the (a_{max}, τ)-grid, then $\Gamma_q(0) = \mathbf{S}$, $T_q \leq T$, and

$\Gamma_q(T_q) = \mathbf{G}$. If the components of $\mathbf{S}$, $\mathbf{G}$, v_{max}, and a_{max} are all rationals, τ can always be chosen to satisfy this condition; however, it is inversely dependent on the least common denominator of these parameters.

5 Robots with Coupled Dynamics Bounds

5.1 Overview

Extending the [CDRX] approach to robots with coupled dynamics bounds requires a tracking lemma for coupled systems. Lemmas 5.1 and 5.2 guarantee that we can track trajectory Γ_r using any of a class of discrete sets of acceleration functions and a maximum timestep given by the lemma. This class includes sets of (a) near-extremal piecewise constant accelerations and sets of (b) extremal accelerations that result from applying piecewise constant torques. According to these lemmas, if for each step we are allowed acceleration functions that satisfy conditions dependent on the dynamics bounds obeyed by Γ_r, then we can find a trajectory Γ_q that approximately tracks Γ_r and that can be computed from these functions using a fixed timestep that depends on (a) deviation from Γ_r's dynamics bounds and (b) the tracking tolerance. That is, the segment of Γ_q over the duration of a given timestep will be generated by a member of a set of pre-computed controls.

We are motivated to prove such a general lemma by the hope that we could use it to obtain bounds for a kinodynamic planning algorithm that generates trajectories along control extremals. The reason for considering extremal trajectory segments in the context of an approximation algorithm is subtle. Although our main goal is a (polynomial-time) provable approximation algorithm, we would hope that if the algorithm produces a solution $\mathbf{a}_{\mathcal{K}}$ for a problem $\mathcal{K}$ then there might be a "nearby" problem $\mathcal{K}'$ for which $\mathbf{a}_{\mathcal{K}}$ is the optimal solution. Intuitively, this would correspond to a proof that our solution is "stable". For this to occur, in general it would be necessary for the algorithm to consider trajectory segments that correspond to control extremals. At the time being, though, we have the lemma but can only a weaker result using trajectory segments that result from near-extremal controls that lie on a regular grid.

We first apply the lemma in a simple way to obtain a polynomial-time approximation algorithm for a Cartesian robot with L_2 bounds. Then, using global dynamics bounds [HP], we obtain a polynomial-time approximation algorithm for an open-chain manipulator. We show that a factor of ϵ advantage in torque bounds obeyed by Γ_r allows us to do the following: given a tracking tolerance (η_x, η_y), we can choose a set of acceleration functions whose number is polynomial in $\frac{1}{\epsilon}$ at each timestep and track Γ_r to tolerance (η_x, η_y) using a timestep that is polynomially dependent on η_x, η_v, and ϵ. Choosing these acceleration functions to be constant functions lying at the interstices of a regular grid then results in a polynomial-sized search space and gives us the complexity bound. The reduction of time-dependent coupled dynamics to piecewise constant accelerations is similar to that in [JCHP].

5.2 The Basic Idea

The the approach is a generalization of the [CDRX] algorithm (Section 3). It first chooses an underlying acceleration-space discretization with grid-spacing $\boldsymbol{\mu}$ and a timestep τ that we derive as functions of the velocity and generalized force bounds, robot dynamics, safety parameters c_0 and c_1, and the approximation parameter ϵ. $\boldsymbol{\mu}_i$ is the acceleration grid-spacing in dimension i, and each $\boldsymbol{\mu}_i$ shrinks polynomially in ϵ. We call the set of gridpoints the $\boldsymbol{\mu}$-grid. The choice of $\boldsymbol{\mu}$ and τ induces an underlying regular grid covering the state space of the robot (the phase space TC). This implies that if the j^{th} joint is revolute, then $\boldsymbol{\mu}_j$ must be chosen so that $K_j\boldsymbol{\mu}_j\tau^2 = 4\pi$ for some integer K_j. Again (recall Section 3.1) we call this grid the TC-grid.[5]

Wlog, we assume that the start **S** and goal **G** (with velocity terms rescaled) lie on the TC-grid. After choosing $\boldsymbol{\mu}$ and τ, the algorithm performs a breadth-first search of state-space starting from **S** that takes time $O(G)$ in the number G of TC-gridpoints. This search phase is similar to the search phase of the L_∞ algorithm, and only δ'_v-safe trajectory segments are considered. We can view this phase as the search for a shortest path between two vertices on a directed graph in which the vertices are TC-gridpoints and the edges are trajectory segments generated by constant near-extremal $\boldsymbol{\mu}$-grid accelerations of duration τ. Thus, when expanding the search frontier, our algorithm considers accelerations that lie within $\boldsymbol{\mu}$ of a $(d-1)$-surface; this distinguishes it from the algorithm of [JHCP], which must consider accelerations all accelerations on a grid in a d-dimensional solid.

We can modify the algorithm to solve the approximately optimal single-source and single-sink kinodynamic planning problems by the same technique we use on the algorithm of Section 3. Thus, the algorithms for these two problems are no more expensive than the state-state algorithm in the worst case, since a backpointer must be kept for each state explored so that the algorithm can return a solution.

5.3 Key Concepts

For any given TC-gridpoint $(\mathbf{p}, \dot{\mathbf{p}})$ and timestep τ there is a set $\tilde{\mathcal{A}}(\mathbf{p}, \dot{\mathbf{p}})$ of constant accelerations such that a trajectory starting at $(\mathbf{p}, \dot{\mathbf{p}})$ and obeying acceleration $\mathbf{a} \in \tilde{\mathcal{A}}(\mathbf{p}, \dot{\mathbf{p}})$ for duration τ observes the dynamics bounds. For a given state $(\mathbf{p}, \dot{\mathbf{p}})$, the algorithm considers members of $\tilde{\mathcal{A}}(\mathbf{p}, \dot{\mathbf{p}})$ that lie on the $\boldsymbol{\mu}$-grid and are within a grid spacing of the boundary of $\tilde{\mathcal{A}}(\mathbf{p}, \dot{\mathbf{p}})$; the size of this set will be polynomial in $(\min_i \boldsymbol{\mu}_i)^{-1}$ and thus in $\frac{1}{\epsilon}$. We say that these accelerations are $(\boldsymbol{\mu}, \tau)$-*extremal* and call the set of these constant accelerations the $(\boldsymbol{\mu}, \tau)$-*extremal shell* of accelerations. We call applying a $(\boldsymbol{\mu}, \tau)$-extremal acceleration for duration τ a $(\boldsymbol{\mu}, \tau)$-*bang*[6]; this term also refers to the resulting trajectory segment. These accelerations are analogous to the (a, τ)-bangs of [CDRX] and previous sections.

[5] To see the induced discretization is regular, consider the set of all points in TC reachable from a start state via any sequence of accelerations $\boldsymbol{\mu}$-grid each followed for time τ.

[6] Strictly speaking, they are *pseudo-bangs*, since in general they are only merely close to being extremal.

A $(\boldsymbol{\mu}, \tau)$*-bang trajectory* starts on the TC-grid, and is defined by a finite number of $(\boldsymbol{\mu}, \tau)$-bangs each of duration τ. We may think of it as successive bangs "between" TC-gridpoints. It seems intuitively plausible that for a sufficiently "fine" grid, some $(\boldsymbol{\mu}, \tau)$-bang trajectory might approximate the optimal-time trajectory. The key issues are to choose the grid spacing correctly to prove that the approximation bound

$$T_q \leq (1+\epsilon)T_{opt} \tag{22}$$

holds. (T_{opt} is the true optimal safe solution time, and T_q is the grid-bang solution time.) Furthermore, the grid must be polynomial in size.

Our goal is to choose sufficiently small $\boldsymbol{\mu}_i$ and τ such that (22) holds, but still maintain a polynomial-size number of grid points. To this end, we can show that $\boldsymbol{\mu}$ and τ can be chosen such that for any safe trajectory, there exists a "nearby" -bang trajectory that is as 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 $(\boldsymbol{\mu}, \tau)$-bang trajectory if it exists, and is the fastest.

The notion of "sufficient nearness" is again captured by a "tube" about a phase space trajectory. A tube about a trajectory (in phase space) is a neighborhood about the the trajectory image specified by two metric parameters for position space and for velocity space. The definition of "approximately tracks" is the same as in section 4.1.

5.4 Putting It Together

The robot motion obeys the following equation derived from Lagrangian mechanics [Hol]:

$$\mathbf{f}(t) = \mathbf{M}(\mathbf{p}(t))\mathbf{a}(t) + [\dot{\mathbf{p}}^T(t)\mathbf{C}(\mathbf{p}(t))\dot{\mathbf{p}}(t)] + \mathbf{G}(\mathbf{p}(t)). \tag{23}$$

The inertia matrix $\mathbf{M}$ is symmetric positive definite; for a Cartesian robot it is diagonal. $\mathbf{C}(\mathbf{p}(t))$ is a $d \times d \times d$ tensor that encodes centripetal and Coriolis forces. $\mathbf{G}(\mathbf{p}(t))$ is the gravity term.

Suppose a trajectory $\Gamma_r(t) = (\mathbf{p}_r(t), \dot{\mathbf{p}}(t))$ obeys the bounds $\mathbf{f}_{max}$. Let $\zeta = \frac{1}{1+\epsilon}$. For the given trajectory $\Gamma_r(t)$, we define the ϵ*-time-rescaled trajectory* $\Gamma'_r(t)$ as follows:

$$\Gamma'_r(t) = (\mathbf{p}_r(\zeta t)), \zeta \mathbf{p}_r(\zeta t)) \,. \tag{24}$$

Using time-rescaling [Hol] we find that at time $(1+\epsilon)t)$

$$\mathbf{f}'_r((1+\epsilon)t) = \frac{\mathbf{f}_r(t)}{(1+\epsilon)^2} + \frac{2\epsilon+\epsilon^2}{(1+\epsilon)^2}\mathbf{G}(\mathbf{p}_r(t)) \tag{25}$$

This means that $\Gamma'_r(t)$ obeys some L_∞-smaller bounds $\mathbf{f}'_{max}$.

Intuitively, we expect that if a trajectory Γ_q tracks Γ'_r as defined in (24) closely enough, then Γ_q will be $(1-\epsilon)\delta_v$-safe. As it turns out, coupled dynamics do not affect the applicability of Safe Tracking Lemma from [CDRX]. Thus, we show the correctness of our algorithm by showing that in the worst case, for some optimal trajectory Γ_r, our algorithm can find a trajectory Γ_q that tracks Γ'_r to tolerance a (η_x, η_v).

5.5 The Coupled Tracking Lemma

5.5.1 Motivation and Statement of the Lemma

Let $\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ denote the set of possible accelerations for a robot in state $(\mathbf{p}, \dot{\mathbf{p}})$ under bounds $\mathbf{f}_{max}$. Let $\mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$ denote the set of possible accelerations for the same robot in state $(\mathbf{p}, \dot{\mathbf{p}})$ but subject to bounds $\mathbf{f}'_{max}$. Clearly, $\mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}}) \subset \mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$. If $f'_{max\ i} < f_{max\ i}$ for each i, then we might expect that $\mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$ will lie entirely inside $\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ by some margin κ.

Consider the intersection of the $\boldsymbol{\mu}$-grid and $\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}}) - \mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$. Call this $\mathcal{H}(\mathbf{p}, \dot{\mathbf{p}})$, and note that it is a discrete set whose size is proportional to all $\frac{1}{\boldsymbol{\mu}_i}$ If the $\boldsymbol{\mu}$-grid is fine enough, then we might expect that for any $\mathbf{a} \in \mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$ and any d-vector σ of 1's and –1's, we could find some $a^{\mathcal{H}} \in \mathcal{H}(\mathbf{p}, \dot{\mathbf{p}})$ such that

$$\forall i, \sigma_i(a_i^{\mathcal{H}} - a_i) \geq \kappa_l \tag{26}$$

for a constant κ_l. Informally, we would expect that with respect to any given direction σ we can find an $a^{\mathcal{H}}(\sigma)$ that has a such a κ_l advantage over $\mathbf{a}$.

Since $\mathbf{p}$ and $\dot{\mathbf{p}}$ are functions of time, we generalize $\mathcal{A}$ to denote a function $\mathcal{A}$: *time* $\rightarrow$ $\{X : X$ *is a set of accelerations*$\}$. Let $\mathcal{H}$ be a set of acceleration functions. Suppose that for interval $\mathcal{I} = [\tau_I, \tau_I + \Delta\tau_I]$ the following is true: if $\mathbf{a} \in \mathcal{A}(t)$ for all $t \in \mathcal{I}$ and is a d-vector of 1's and –1's, then there is some $a^{\mathcal{H}} \in \mathcal{H}$ such that

$$\sigma_i \left(a_i^{\mathcal{H}}(t) - \frac{\int_{\tau_I}^{\tau_I + \Delta\tau_I} a_i(s) ds}{\Delta\tau_I} \right) \geq \kappa_l \tag{27}$$

for each i and each $t \in \mathcal{I}$. Then we say that $\mathcal{H}$ *has a uniform* κ_l *advantage over* $\mathcal{A}$ *during* $\mathcal{I}$.

The Coupled Tracking Lemma relates a tracking tolerance (η_x, η_v), a κ_l acceleration advantage, and global acceleration bounds to an upper bound on τ.

In some fixed coordinate system, let $\mathcal{A}$ be a function $\mathcal{A}$: *time* $\rightarrow$ $\{X : X$ *is a set of accelerations*$\}$. We say that a trajectory $\Gamma_r(t)$ *respects* $\mathcal{A}$ if for all t, $\mathbf{a}_r(t) \in \mathcal{A}(t)$.

Lemma 5.1 (**Coupled Tracking Lemma**) *Let $\mathcal{A}$ be a function $\mathcal{A}$: time $\rightarrow$ $\{X : X$ is a set of accelerations$\}$. Let τ_0 be given. Suppose that for each $\tau \leq \tau_0$ there is a set of acceleration functions $\mathcal{H}_\tau^{(n)}$ such that for each n, in $[n\tau, (n+1)\tau]$, $\mathcal{H}_\tau^{(n)}$ has a uniform κ_l advantage over $\mathcal{A}$. Suppose that for all t and i, $|a_i(t) - a'_i(t)| \leq 2A_{max}$ if $\mathbf{a}, \mathbf{a}' \in \mathcal{A}(t)$, and that for all n and i, $|a_i(t) - a'_i(t)| \leq 2(A_{max} + \kappa_h)$ if $\mathbf{a}_i, \mathbf{a}' \in \mathcal{H}_\tau^{(n)}$.*

Then, for any L_∞ tracking tolerance (η_x, η_y), there exists a timestep τ such that if a trajectory Γ_r respects $\mathcal{A}$, then there exists a trajectory Γ_q such that

(a) $\Gamma_r(0) = \Gamma_q(0)$;

(b) Γ_q tracks Γ_r to tolerance (η_x, η_y); and

(c) For each n, there is some $\mathbf{a}^{(n)} \in \mathcal{H}_\tau^{(n)}$ such that $\mathbf{a}_q(t) = \mathbf{a}^{(n)}(t)$ for all $t \in [n\tau, (n+1)\tau]$.

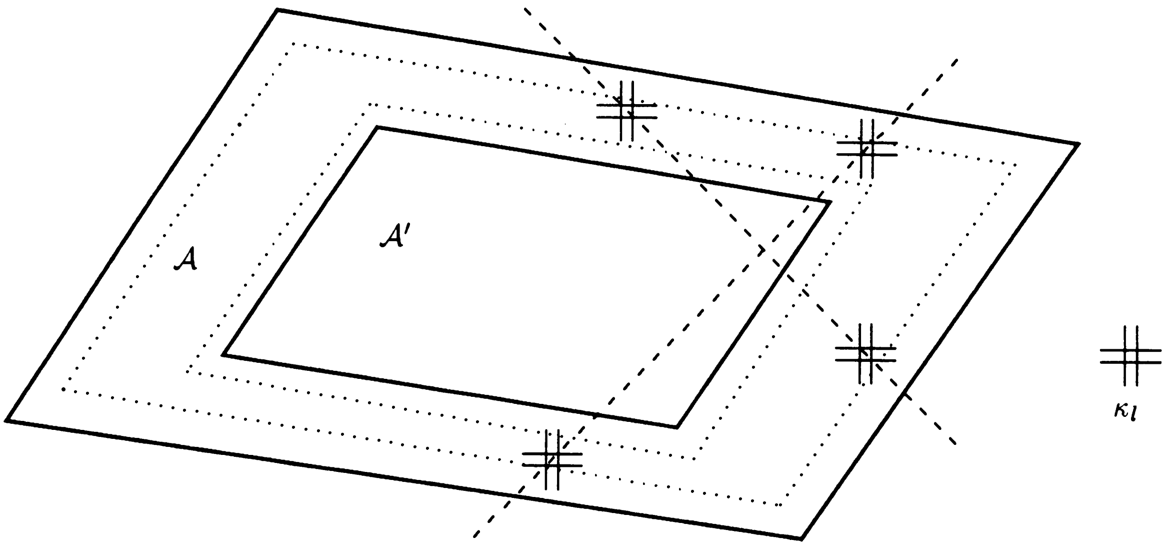


Figure 10: Sets in acceleration space. $\mathcal{A}$ obeys bound $\mathbf{f}_{max}$; $\mathcal{A}'$ obeys $\mathbf{f}'_{max}$. Dotted lines show boundaries perturbed by $h_{q1}(\eta_x, \eta_v) + h_{q0}(\tau)$ and $h_r(\tau)$. Also shown are witnesses to $\mathcal{A}$'s κ_h advantage over a member of $\mathcal{A}'$.

Furthermore, it is sufficient that $\tau \leq \tau_0$ and that

$$\begin{aligned} \tau &\leq \sqrt{\frac{\eta_x \kappa_l}{8(A_{max}+\kappa_h)^2 + 2\kappa_l(A_{max}+\kappa_h)}}; \\ \tau &\leq \frac{\eta_v}{3(A_{max}+\kappa_h)}. \end{aligned} \tag{28}$$

We prove the lemma in Section 6.2. While this lemma is sufficient to prove our result for Cartesian robots, we use a more complicated version of this lemma in obtaining our result for open-chain manipulators.

Lemma 5.2 (Inductive Coupled Tracking Lemma) *Let $\mathcal{A}$ be a function $\mathcal{A}$: time $\rightarrow$ $\{X : X$ is a set of accelerations$\}$ such that if $\mathbf{a}, \mathbf{a}' \in \mathcal{A}(t)$, then for all t and i, $|a_i(t) - a_i'(t)| \leq 2A_{max}$. Let $\mathbf{v}_{max}$ be a velocity bound. Let $\Gamma_r(t)$ respect $\mathcal{A}(t)$ and $\mathbf{v}_{max}$.*

Suppose that there exist (η_{x0}, η_{v0}) and τ_0 such that for all $\tau \leq \tau_0$ the following is true: for all n, if $\Gamma_q(n\tau)$ is within (η_{x0}, η_{v0}) of $\Gamma_r(n\tau)$, then there is a set of acceleration function $\mathcal{H}_q^{(n)}$ such that for each n in $[n\tau, (n+1)\tau]$, (a) $\mathcal{H}_q^{(n)}$ has a uniform κ_l advantage over $\mathcal{A}$, and (b) $|a_i(t) - a_i'(t)| \leq 2(A_{max} + \kappa_h)$ if $\mathbf{a}, \mathbf{a}' \in \mathcal{H}_q^{(n)}$

Then, for any L_∞ tracking tolerance (η_x, η_y), there exists a timestep τ such that if a trajectory Γ_r respects $\mathcal{A}$, then there exists a trajectory Γ_q such that

(a) $\Gamma_r(0) = \Gamma_q(0)$;

(b) Γ_q tracks Γ_r to tolerance (η_x, η_y); and

(c) For each n, there is some $\mathbf{a}_{(n)} \in \mathcal{H}^{(n)}$ such that $\mathbf{a}_q(t) = \mathbf{a}_{(n)}(t)$ for all $t \in [n\tau_0, (n+1)\tau_0]$.

It is sufficient that

$$\begin{aligned} \tau &\leq \sqrt{\frac{\eta_x \kappa_l}{7(A_{max}+\kappa_h)^2+2\kappa_l(A_{max}+\kappa_h)}}; \\ \tau &\leq \frac{\eta_v}{3(A_{max}+\kappa_h)}. \end{aligned} \tag{29}$$

5.5.2 Applying the Coupled Tracking Lemmas

Using this lemma and global dynamics bounds we can show the following. First, for any $\eta_x, \eta_v > 0$, we can choose $\boldsymbol{\mu}$ and τ such that for any trajectory Γ_r' obeying $\mathbf{f}'_{max}$, there is an $(\boldsymbol{\mu}, \tau)$-grid-bang trajectory within the (η_x, η_v)-tube about Γ_r' and obeying $\mathbf{f}_{max}$. Second, for the robot classes we consider, τ and the $\boldsymbol{\mu}_i$ need only be polynomially small in the tube size and ϵ. This gives us the complexity bound in Theorem 2.3.

Furthermore, by using the Coupled Tracking Lemma and global dynamics bounds, we can show the following: by using trajectory segments that correspond to the application of constant generalized forces that lie *on* the force-extremal surface for duration τ, we can find a δ_v'-safe trajectory Γ_q taking time $\leq T(1+\epsilon)$ if a δ_v-safe trajectory Γ_r taking time T exists. A constant set of gridpoints on the surface of force-extremals is adequate for the robots we consider. However, we do not yet been able to polynomially bound the number of states an algorithm using this set of force controls would have to search to find a provably good approximation. This is why we use $(\boldsymbol{\mu}, \tau)$-bangs in our algorithm.

For Cartesian robots obeying L_2 dynamics bounds, it is straightforward to choose $\boldsymbol{\mu}$ and τ by using Lemmas 4.1 and 5.1. If $0 < \epsilon < 1$, then $\frac{\epsilon}{2} < 1 - \frac{1}{(1+\epsilon)^2} < 2\epsilon$. Choosing $\mu \leq \frac{\epsilon a_{max}}{6}$ implies that if $\kappa_l \leq \mu$ and *any* τ_0, then for all t the (μ, τ_0)-extremal shell $\mathcal{H}(\mathbf{p}_q(t), \dot{\mathbf{p}}_q(t))$ has a uniform κ_l-advantage over $\mathcal{A}_r'(\mathbf{p}_r'(t), \dot{\mathbf{p}}_r'(t))$, *regardless of how closely* Γ_q *tracks* Γ_r'. Applying the two lemmas we obtain a τ proportional to ϵ.

For open-chain manipulators we must choose $\boldsymbol{\mu}$ and τ small enough to allow for changes in $\mathcal{A}(\mathbf{p}_q, \dot{\mathbf{p}}_q)$ and $\mathcal{A}'(\mathbf{p}_r', \dot{\mathbf{p}}_r')$ during the timestep and to how inexactly Γ_q tracks Γ_r'. We first show that there exist maximal μ, τ_0, η_{x0}, and η_{v0} such that (a) if $\Gamma_q(n\tau_0)$ is within (η_{x0}, η_{v0}) of $\Gamma_r'(n\tau_0)$, then the $(\boldsymbol{\mu}, \tau_0)$-extremal shell $\mathcal{H}(\mathbf{p}_q(n\tau_0), \dot{\mathbf{p}}_q(n\tau_0))$ has a uniform κ_l-advantage over $\mathcal{A}_r'(\mathbf{p}_r'(t), \dot{\mathbf{p}}_r'(t))$ in the time interval $[n\tau_0, (n+1)\tau_0]$; and (b) τ_0 is adequately small so that we can apply Lemma 5.2 to show that there exists a (μ, τ)-bang Γ_q that tracks Γ_r' to tolerance (η_{x0}, η_{v0}). It follows that we can apply the lemma as long as $\boldsymbol{\mu}_i \leq \mu$ for all i and $\tau \leq \tau_0$. We show this in Section 6.4.

5.6 Safety-Checking

Observe that the kinodynamic solutions our algorithm obtains have piecewise-constant acceleration $\ddot{\mathbf{p}}_q$. Hence the solution trajectory Γ_q is algebraic: $\mathbf{p}_q$ (resp. $\dot{\mathbf{p}}_q$) is piecewise quadratic (resp. linear) in time t. Hence, for Cartesian manipulators, safety checking amounts to intersection detection between Γ_q and a set of polyhedral obstacles in TC. Each collision is detected as the intersection of the quadratic Γ_q with a set of polyhedra. Checking for safety violations is done exactly as in the L_∞ dynamics bounds case in

[CDRX]. The obstacles are "grown" affinely with speed, so their boundaries form $O(n)$ algebraic surfaces in phase space. Collision detection for a single (a_{max}, 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. As shown in [CDRX], this takes time $O(n^2 \log n)$ per bang.

For open chain manipulators revolute joints, safety is checked in the same manner (by affinely growing the obstacles); however, computationally, safety checking is more complicated. Recall $(\boldsymbol{\mu}, \tau)$-bangs have position components that are quadratic in t and velocity components that are linear in t. We would like to apply an algebraic collision detection predicate such as that of [C]. It is possible to describe the c-space obstacles algebraically by using the substitution $u_j = \tan \frac{\theta_j}{2}$ for each revolute joint j. However there is no way to algebraically parameterize the path that results from this substitution if the configuration space has more than one dimension.

Specifically, let θ_j denote the angle of each revolute joint j. Then c-space obstacle surfaces are the zeroes of functions with terms $\cos \theta_j$ or $\sin \theta_j$. In the coordinate system given by the substitution $u_j = \tan \frac{\theta_j}{2}$, c-space obstacle surfaces are the zeroes of functions that are algebraic in the u_j. Collision detection is algebraic when there are two or more revolute joints if and only if each u_j is a polynomial in t. Since our algorithm only considers motions where each θ_j is a quadratic polynomial in t, we cannot in general perform safety-checking algebraically.

Thus, our collision predicate will contain terms that contain products of tangents or sines and cosines of components of $\mathbf{p}$ and $\dot{\mathbf{p}}$. Since each timestep is finite, and in practice, very short, these transcendental terms can be approximated by polynomials; thus approximate detection of δ_v-safety violations can be done algebraically. The resulting error ϵ_s in checking safety can be bounded and made arbitrarily small by increasing the degree of the polynomial. This error can be accounted for by adjusting c_0'. Then one could implement either a conservative algorithm, which only find solutions that are $(\delta_v' + \epsilon_s)$-safe, or an "optimistic" algorithm that finds solutions that are $(\delta_v' - \epsilon_s)$-safe.

To summarize, under a combinatorial model that charges unit cost for finding the roots of a univariate trignometric polynomial of fixed degree and size, safety checking can be done in time $O(n^2 \log n)$. In practice, numerical methods would be used to approximately solve the polynomial in t; however, the error in these methods can be bounded and incorporated into the safety margin.

6 Kinodynamic Planning for Robots with Coupled Dynamics: a Provably-Good Polynomial Approximation Algorithm Using Near-Extremal Controls

6.1 Overview

In Section 6.2 prove the Coupled Tracking Lemma (5.1) and the Inductive Coupled Tracking Lemma 5.2). We apply the lemma in Section 6.3 to obtain a provably good polynomial-time approximation algorithm for kinodynamic planning for Cartesian Robots obeying L_2 dynamics bounds. This development is direct and parallels that in Sections 4.4 and 4.5. The development necessary to derive conditions on μ and τ for the open-chain manipulator problem is more complicated and presented in Section 6.4.

6.2 Proving the Coupled Tracking Lemma

In a system with decoupled dynamics bounds, a maximal acceleration along any one axis can be attained without influencing what magnitudes of accelerations are possible in any other direction. This is why tracking the trajectory of a robot obeying L_∞ bounds reduces to the one-dimensional case. If a system has coupled dynamics bounds, attaining a maximal acceleration along one axis can limit the acceleration along another axis to zero or to the negative or positive direction. Thus, we cannot readily use velocity limits to bound how long a tracking trajectory loses ground. Furthermore, in general it is not possible to limit the a tracking trajectory to τ_0-constant accelerations in whose i^{th} components are members $\{-\mathbf{a}_{max\ i}, 0, \mathbf{a}_{max\ i}\}$. The observation that $\mathcal{H}^{(n)}$ has a κ_l advantage over $\mathcal{A}$ during n^{th} time interval allows us to motivate a proof of Coupled Tracking Lemma by examining a game in a one dimensional space. In this game certain rules simulate dynamics, and the adversary's partial control of our movement covers all cases when "dynamics work against us".

6.2.1 The Tracking Game

Consider a game in which you are trying to track an adversary in a one dimensional space. The game is a series of rounds, each of which simulates motions taking time τ. The game begins with adversary in state (x_0, v_0) and you in state (x'_0, v'_0). In each round n the adversary plays first by choosing *for himself* an acceleration function $a_n(t)$ bounded above and below by $a_{max\ n}(t)$ and $a_{min\ n}(t)$, where $|a_{max\ n} - a_{min\ n} \leq 2A_{max}|$. His state (x_{n+1}, v_{n+1}) at the end of the round (beginning of the next) is computed from his state (x_n, v_n) and this acceleration function by integration.

For your play in round n, you then choose between HIGH and LOW; this choice limits the acceleration function that the adversary can choose *for you* to determine your state (x'_{n+1}, v'_{n+1}) at the end of the round. Note that you play only HIGH or LOW;

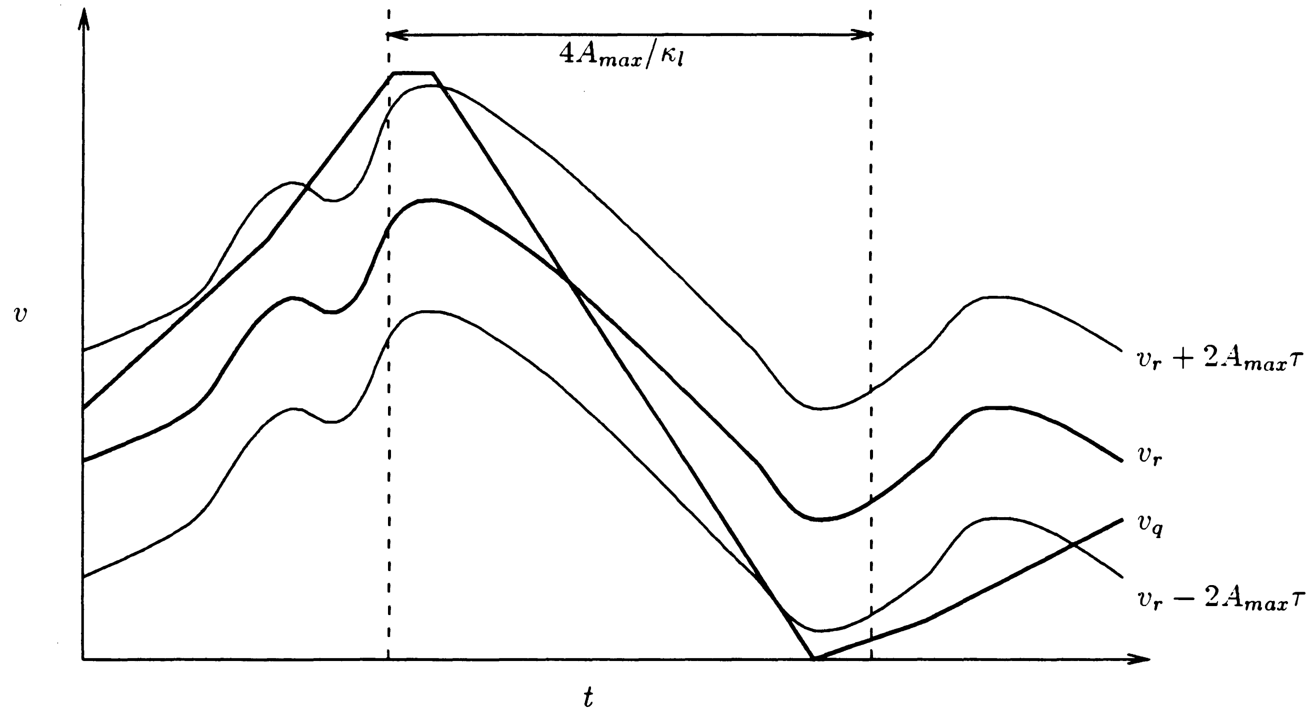


Figure 11: Intuition for proving Lemmas 5.1 and 5.2: a game in 1D. We try to track an adversary whose velocity v_r respects A_{max}. We generate a v_q timestep by timestep, trying to minimize $|v_q(t) - v_r(t)|$ and $|\int(v_r(t) - v_q(t))dt|$, but our control system is so sloppy that it can only guarantee (a) that our acceleration $a(t)$ is above or below (we control this) the adversary's average over the timestep by κ_l and (b) that $|a(t)| \leq A_{max} + \kappa_h$. Mimicking the adversary's velocity v_r to within $(A_{max} + \kappa_h)\tau$ is easy; it follows that we can do the same for $v_r + 2A_{max}\tau$ and $v_r - 2A_{max}\tau$. Furthermore, we can go from being faster than $v_r + 2A_{max}\tau$ to slower than $v_r - 2A_{max}$ (and vice-versa) within time $4A_{max}\tau/\kappa_l$. Thus $|\int(v_r(t) - v_q(t))dt|$ can be kept within a constant bound dependent on τ. The game conditions give us a κ_l advantage for each timestep. When several games are played at once, the control sloppiness is a harsher condition than being limited to a (μ, τ)-extremal shell of accelerations.

---

the adversary chooses *your* acceleration *as well as his own.* However, his choice of your acceleration is constrained by your play.

Specifically, if you play HIGH, then your adversary can choose any function $a'_n(t)$ such that $\frac{v_{n+1}-v_n}{\tau} + \kappa_l \leq a'_n(t) \leq a_{max\ n} + \kappa_h$ for $t \in [n\tau, (n+1)\tau]$; if you play LOW, then your adversary's choice must obey the condition $-a_{min\ n} - \kappa_h \leq a'_n(t) \leq \frac{v_{n+1}-v_i}{\tau} - \kappa_l$. Your state (x'_{n+1}, v'_{n+1}) at the end of the round is computed using your state at the end of the previous round and $a'_n(t)$.

6.2.2 A Winning Simple Strategy for the Tracking Game

Recall the proof of the Strong Tracking Lemma in section 4.3, including the lemmas leading to the proof. For a trajectory Γ_r we show the existence of a tracking trajectory

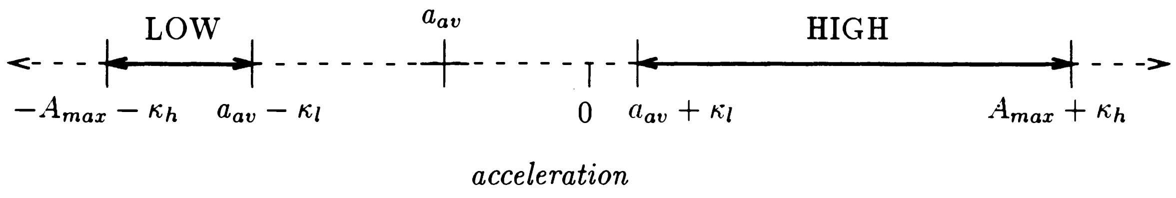


Figure 12: Details from a game against an adversary. Suppose in the current round the adversary's average acceleration is a_{av}. Then, if you choose HIGH, the adversary assigns you an acceleration function whose range lies in $[a_{av} + \kappa_l, A_{max} + \kappa_h]$; if you choose LOW, the adversary assigns you an acceleration function whose range lies in $[-A_{max} - \kappa_h, a_{av} - \kappa_l]$. This function controls your motion for the current round, which covers a time interval τ.

Γ_q. In anthropomorphic terms, Γ_q "tries" to go faster than $v_r(t)$ when it is behind in position and slower than $v_r(t)$ when it is ahead. Given a velocity function $v_r(t)$ we show the existence of velocity functions $v_{fast}(t)$ and v_{slow} that (a) bound $v_r(t)$ closely from above and below, respectively, and (b) bound the number of timesteps that a grid-bang velocity $v_q(t)$ between v_{fast} and v_{slow} needs to get from being slower than $v_r(t)$ to being faster, or vice-versa. In turn, this gives us a bound on how far Γ_q deviates from Γ_r in position.

This motivates a simple high-level strategy for the Simple Game: try to go faster than the adversary if you have fallen behind; and try to go slower than the adversary if you have gotten ahead; but never go much faster or much slower than the adversary. The uniform κ_l advantage you have over the adversary's average acceleration during a round assures that we can follow such a strategy. (See fig. 11.)

We now give the detailed strategy in terms of the game parameters. Let

$$\begin{aligned} A_{max\ n} &= \max_{t\in[n\tau,(n+1)\tau]} a_{max}(t); \\ A_{min\ n} &= \min_{t\in[n\tau,(n+1)\tau]} a_{max}(t). \end{aligned} \tag{30}$$

Simple Strategy

Assume that at the beginning of round n that

$$-3(A_{max} + \kappa_h)\tau \leq v_n - v'_n \leq 3(A_{max} + \kappa_h)\tau. \tag{31}$$

Then choose HIGH or LOW according to the following rules.

- If $|x_n - x'_n| \leq 2(A_{max} + \kappa_h)\tau^2$, then choose HIGH if $\left|\frac{v_{n+1} - v'_n}{\tau} - A_{max\ n} - \kappa_h\right| \leq \left|\frac{v_{n+1} - v'_n}{\tau} - A_{min\ n} + \kappa_h\right|$, LOW otherwise.

- Else if $x_n - x'_n > 2(A_{max} + \kappa_h)\tau^2$, then choose HIGH if $v'_i \leq v_{n+1} + 2(A_{max\ n} + \kappa_h)\tau$, LOW otherwise.

- Else choose LOW if $v'_n \geq v_{n+1} + 2(A_{min\ n} + \kappa_h)\tau$, HIGH otherwise.

We make the following claim

Claim 6.1 *The Simple Strategy ensures that if* $|x_0 - x_0'| \leq 2(A_{max} + \kappa_h)\tau^2$ *and* $|v_0 - v_0'| \leq A_{max}\tau$ *then at the beginning of each round* n *(a) (31) is true, and (b) the following holds:*

$$|x_n - x_n'| \leq \frac{7(A_{max} + \kappa_h)A_{max}\tau^2}{8\kappa_l} + \frac{49(A_{max} + \kappa_h)^2\tau^2}{8\kappa_l} + 2(A_{max} + \kappa_h)\tau^2 \qquad (32)$$

Furthermore, for all $t \in [0, n\tau]$*:*

$$\begin{aligned} |v_r(t) - v_q(t)| &\leq 3(A_{max} + \kappa_h)\tau; \\ |x_r(t) - x_q(t)| &\leq \tfrac{7(A_{max}+\kappa_h)A_{max}\tau^2}{8\kappa_l} + \tfrac{49(A_{max}+\kappa_h)^2\tau^2}{8\kappa_l} + 2(A_{max} + \kappa_h)\tau^2. \end{aligned} \qquad (33)$$

Proof: We see by inspection of the strategy that (a) holds. If the condition at $n = 0$ holds, then we can see by induction that one of the following cases holds for each relevant n, thus implying (b).

Case 1 Suppose that $|x_n - x_n'| \leq 2(A_{max} + \kappa_h)\tau^2$. Furthermore, suppose that the difference on the left-hand side is positive; the negative case is similar. Since you are guaranteed to be able to gain on the adversary in position during round j if $v_j + \frac{A_{max}\tau}{4} < v_j'$, no more than $\frac{7(A_{max}+\kappa_h)}{2\kappa_l}$ rounds will elapse before you are guaranteed to be able to gain on the adversary or not fall further behind. Let $n + k$ be the first round after n during which it is sure that you can gain. Then for j such that $n \leq j < n + k$,

$$v_j + \frac{A_{max}\tau}{4} \geq v_j' > v_j - 3(A_{max} + \kappa_h)\tau \qquad \text{and} \qquad (34)$$

$$x_j - x_j' \leq \frac{7(A_{max} + \kappa_h)A_{max}\tau^2}{8\kappa_l} + \frac{49(A_{max} + \kappa_h)^2\tau^2}{8\kappa_l} + 2(A_{max} + \kappa_h)\tau^2 \quad (35)$$

Round $n + k$ is covered by Case 1 or Case 2. (If $x_n - x_n' < 0$, then round $n + k$ would be covered by Case 1 or Case 3.)

Case 2 Suppose that $x_n - x_n' > 2(A_{max} + \kappa_h)\tau^2$ and $v_n + \frac{A_{max}\tau}{4} < v_n' \leq v_n + 3(A_{max} + \kappa_h)\tau$. Then $-\frac{3(A_{max}+\kappa_h)\tau^2}{2} < x_{i+1} - x_{i+1}' < x_n - x_n' - \frac{(A_{max}+2\kappa_h)\tau^2}{4}$, and $v_{i+1} + \frac{(A_{max}+4\kappa_h)\tau}{4} \leq v_{i+1}' < v_{i+1} + 3(A_{max} + \kappa_h)\tau$. Thus, the conditions of the lemma hold for $n + 1$, and round $n + 1$ will be covered by Case 1 or Case 2.

Case 3 Suppose that $x_n - x_n' < -2(A_{max} + \kappa_h)\tau^2$ and $v_n - \frac{A_{max}\tau}{4} > v_n' \geq v_n - 3(A_{max} + \kappa_h)\tau$. Then $\frac{3(a_{max}+\kappa_h)\tau^2}{2} > x_{i+1} - x_{i+1}' > x_n - x_n' + \frac{\kappa_h\tau^2}{2}$, and $v_{i+1} - \frac{(a_{max}+4\kappa_h)\tau}{4} \geq v_{i+1}' > v_{i+1} - 3(a_{max} + \kappa_h)\tau$. Thus, the conditions of the lemma hold for $i + 1$, and round $i + 1$ will be covered by Case 1 or Case 3.

Case 4 Either $x_n - x_n' > 2(a_{max} + \kappa_h)\tau^2$ and $v_n + \frac{a_{max}\tau}{4} \geq v_n'$ or $x_n - x_n' < -2(a_{max} + \kappa_h)\tau^2$ and $v_n - \frac{a_{max}\tau}{4} \leq v_n'$. This can only arise between the most recent step when Case 1 applies and the next step when Case 2 or Case 3 applies. Thus, from the discussion of Case 1, we see that (32) hold. □

Simple arithmetic manipulation shows the following corollary.

Corollary 6.2 *For us to be able to track the adversary to a tolerance of* (η_x, η_v), *it is sufficient that*

$$\tau \leq \min\left(\sqrt{\frac{\eta_x \kappa_l}{7(A_{max}+\kappa_h)^2 + 2\kappa_l(A_{max}+\kappa_h)}}, \frac{\eta_v}{3(A_{max}+\kappa_h)}\right) \tag{36}$$

We now use facts proven about the Tracking Game to obtain a proof of Lemma 5.1.

Proof (of Lemma 5.1): Suppose that for $\mathcal{A}$ and each $\tau \leq \tau_0$ there is a family of $\mathcal{H}(n)$ meeting the hypothesis of the lemma. Then, for any $\tau \leq \tau_0$ consider d playings of the Simple Game in which your adversary's i^{th} trajectory is $\Gamma_{r,i}$ and $\Gamma_{r,i}(0)$ and $\Gamma_{q,i}(0)$ meet the hypotheses of Claim 6.1. Suppose all d games are played simultaneously. During each round n, for each of the 2^d combinations of HIGH and LOW, there is some function $\mathbf{a}^{\mathcal{H}}(t) \in \mathcal{H}(n)$ such that the adversary could legally return $\mathbf{a}_i^{\mathcal{H}}(t)$ in the i^{th} game, provided you follow the Simple Strategy. Since the conditions on the members of $\mathcal{H}^{(n)}$ and $\mathbf{a}_r(t)$ in the lemma imply those of the game, we can thus apply corollary 6.2. By the corollary you are able to track the adversary to a tolerance of (η_x, η_v) in all d games simultaneously. Therefore, if for all n, for all $t \in [n\tau, (n+1)\tau]$, $\mathbf{a}_q(t) = \mathbf{a}_i^{\mathcal{H}}(t)$, then Γ_q tracks Γ_r to L_∞ tolerance (η_x, η_v). □

A similar argument proves Lemma 5.2.

Proof (of Lemma 5.2): Given $\mathcal{A}$, $\mathbf{v}_{max}$, and Γ_r, suppose that (η_{x0}, η_{v0}) and τ_0 meet the hypotheses of the lemma.

Let (η_x, η_y) be given. Choose τ such that

$$\begin{aligned} \tau &\leq \tau_0; \\ 3(A_{max}+\kappa_h)\tau &\leq \min(\eta_v, \eta_{v0}); \quad \text{and} \\ \tfrac{7(A_{max}+\kappa_h)A_{max}\tau^2}{8\kappa_l} + \tfrac{49(A_{max}+\kappa_h)^2\tau^2}{8\kappa_l} + 2(A_{max}+\kappa_h)\tau^2 &\leq \min(\eta_x, \eta_{x0}). \end{aligned} \tag{37}$$

Consider d playings of the Simple Game in which your adversary's i^{th} trajectory is $\Gamma_{r,i}$ and $\Gamma_{r,i}(0)$ and $\Gamma_{q,i}(0)$ meet the hypotheses of Claim 6.1. Suppose all d games are played simultaneously.

We consider play during round n, assuming that you have followed the Simple Strategy and play has proceeded legally through the end of the previous round. By Claim 6.1, in each game i, $\Gamma_{q,i}(n\tau)$ is within (η_{x0}, η_{v0}) of $\Gamma_{r,i}(n\tau)$. Then during round n, for each of the 2^d combinations of HIGH and LOW, there is some function $\mathbf{a}^{\mathcal{H}}(t) \in \mathcal{H}_q^{(n)}$ such that the adversary can legally return $\mathbf{a}_i^{\mathcal{H}}(t)$ in the i^{th} game, provided you follow the Simple Strategy. Suppose you do so. Then by the claim, in each game i, $\Gamma_{q,i}((n+1)\tau)$ is within (η_{x0}, η_{v0}) of $\Gamma_{r,i}((n+1)\tau)$. By induction, we see that (33) holds for the duration of the game.

Therefore, if for all n, for all $t \in [n\tau, (n+1)\tau]$, $\mathbf{a}_q(t) = \mathbf{a}_i^{\mathcal{H}}(t)$, then Γ_q tracks Γ_r to L_∞ tolerance (η_x, η_v). □

6.3 A Polynomial-time Approximation Algorithm for a Cartesian Robot Obeying L_2 Acceleration and Velocity Bounds

Using time-rescaling [Hol] as in [CDRX], we can apply Lemma 5.1 to get a polynomial-time approximation algorithm for a Cartesian robot obeying L_2 acceleration and velocity bounds. Assume that $\epsilon < 1$. Recall that if a path $\mathbf{p}$ in $\Re^d$ is traversed by a trajectory Γ_r under acceleration bound a_{max} and velocity bound v, then there exists a Γ_r' that also traverses $\mathbf{p}$ with acceleration bounds $\frac{a_{max}}{(1+\epsilon)^2}$ and velocity bounds $\frac{v_{max}}{1+\epsilon}$. (This is shown explicitly in [CDRX].)

Observe that if $0 < \epsilon < 1$, then $\frac{\epsilon}{2} < 1 - \frac{1}{(1+\epsilon)^2} < 2\epsilon$. Let us grid acceleration space with a grid-spacing of

$$\mu \leq \frac{\epsilon a_{max}}{6}. \tag{38}$$

Recall the definitions of the (μ,τ)-*extremal shell* and a (μ,τ)-bang. Let $\mathcal{A}$ be the set of all accelerations obeying the L_2 bound $A_{max} = \frac{a_{max}}{(1+\epsilon)^2}$. Observe that for any τ the (μ,τ)-shell obeys the conditions on $\mathcal{H}^{(n)}$ in the Coupled Tracking Lemma. Setting $\kappa_l \geq \frac{\epsilon a_{max}}{6}$ and observing that $A_{max} + \kappa_h = a_{max}$, we apply the Coupled Tracking Lemma to get the following lemma by simple substitution.

Lemma 6.3 *Suppose a trajectory Γ_r' obeys L_2 acceleration bounds $\frac{a_{max}}{(1+\epsilon)^2}$ and takes time T_r. Then in the absence of obstacles*

(a) for any positive η_x and η_v, for any acceleration grid spacing μ obeying (38) there exists a time-spacing τ, and an (a_{max},μ,τ)-bang trajectory Γ_q with bounded acceleration a that approximately tracks Γ_r' to L_∞ tolerance (η_x,η_v) in time T_r.

(b) Moreover, a sufficient condition for the choice of τ is that:

$$\tau \leq \min\left(\sqrt{\frac{\eta_x \epsilon}{a_{max}(48+2\epsilon)}}, \frac{\eta_v}{3a_{max}}\right) \tag{39}$$

(c) If Γ_r obeys L_2 velocity bound $\frac{v_{max}}{1+\epsilon}$, then adding the condition

$$\tau \leq \frac{\epsilon v_{max}}{3a_{max}(1+\epsilon)} \tag{40}$$

to (b) ensures that there exists a Γ_q that obeys L_2 velocity bound v_{max} in addition to meeting the conditions of (a).

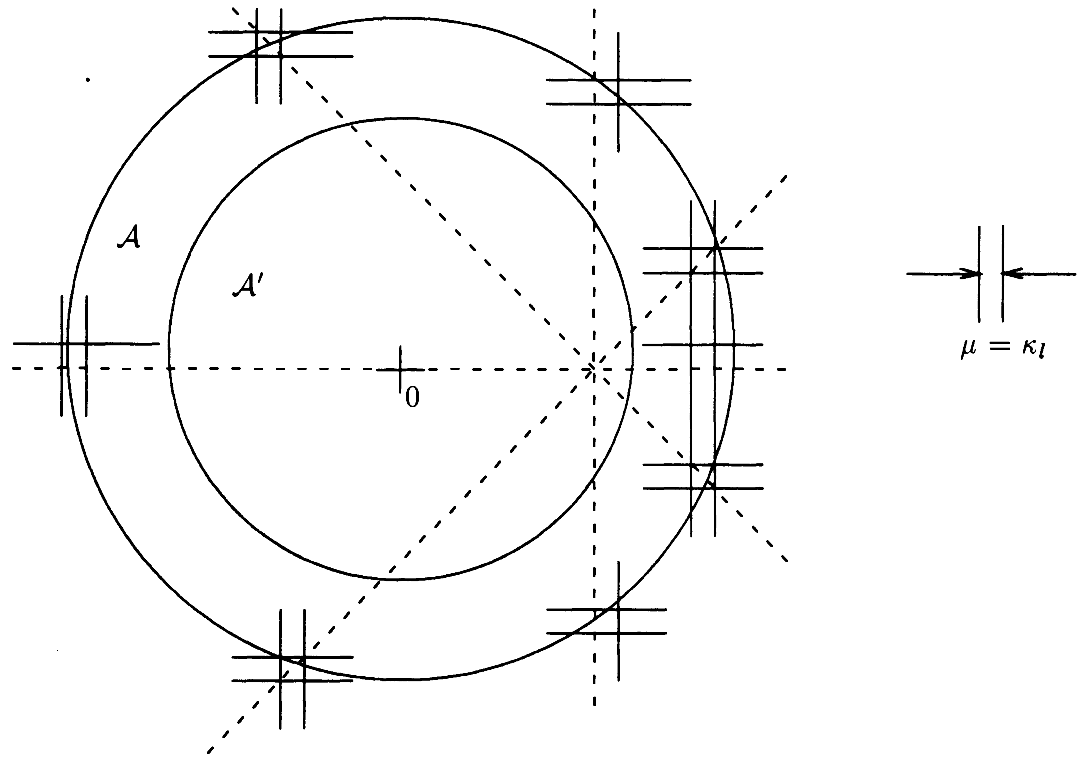


Figure 13: Allowable accelerations for the L_2 point robot. $\mathcal{A}$ obeys a_{max}, $\mathcal{A}'$ obeys $\frac{a_{max}}{(1+\epsilon)^2}$. For any $\mathbf{a} \in \mathcal{A}'$, *wrt* any direction given by a vector of 1's and -1's, there is a (μ, τ)-extremal acceleration that has a κ_l advantage over $\mathbf{a}$. Thus, the (μ, τ)-extremal shell has a uniform κ_l advantage over $\mathcal{A}'$ for any τ.

The Safe Tracking Lemma from [CDRX] applies to all metrics, so we can follow the development used for the L_∞ case to obtain an upper bound on our choice of τ. Combining this bound with the acceleration-grid-spacing μ yields an upper bound on the search-space size.

Parameterizing τ as a function of β, we get

$$\begin{aligned} \tau_x(\beta) &= \sqrt{\frac{c_0\epsilon^2\beta}{a_{max}(48+2\epsilon)(c_1(1-\epsilon)+\beta)}} \\ \tau_v(\beta) &= \frac{c_0\epsilon}{3a_{max}(c_1(1-\epsilon)+\beta)} \\ \tau(\beta) &= \min(\tau_x(\beta), \tau_v(\beta)). \end{aligned} \tag{41}$$

$\tau(\beta)$ is maximized when $\tau_x(\beta)$ and $\tau_v(\beta)$ are equal. Solving for β to get a positive τ yields

$$\begin{aligned} \beta &= \tfrac{1}{2}\left(-c_1(1-\epsilon) + \sqrt{c_1^2(1-\epsilon)^2 + 4\gamma}\right) \quad \text{where} \\ \gamma &= \frac{c_0(48+2\epsilon)}{9a_{max}} \end{aligned} \tag{42}$$

Substituting this value into (41) yields a maximum choice of τ:

$$\tau \le \frac{2c_0\epsilon}{3a(c_1(1-\epsilon)+\sqrt{c_1^2(1-\epsilon)^2+4\gamma})} \tag{43}$$

We now bound number of (a_{max}, μ, τ)-gridpoints for a Cartesian robot with maximum (L_2) speed v_{max} in a d-dimensional free-space d-cube with side-length l. (Recall that μ is given by (38).) We so this by simply bounding the number of gridpoints reachable under (μ,τ)-bangs. Let $G_2(a_{max}, \tau, v_{max}, l, d)$ denote this bound. Then

$$G_2(a_{max}, \mu, \tau, v_{max}, l, d) < \left(\left(\frac{2v_{max}}{\mu\tau}+1\right)\left(\frac{l}{\mu\tau^2}+1\right)\right)^d \tag{44}$$

We summarize these results for a Cartesian robot obeying L_2 dynamics bounds with the following theorem.

Theorem 6.4 *Given L_2-norm acceleration bounds a_{max} and velocity bounds v_{max}, environment diameter l, and positive scalars ϵ, c_0, and c_1, the (μ,τ)-grid with τ chosen to satisfy (43) has polynomial size. In addition, given obstacles $\mathcal{O}$, start* **S**, *and goal* **G**, *a minimal-time (μ,τ)-grid-bang, δ_v'-safe trajectory Γ_q from* **S** *to* **G** *can be computed in $O(n^2(\frac{a_{max}}{\mu})^{d-1}G_2)$, where G_2 is given by (44). For fixed 2- or 3-dimensional world size and fixed dynamics bounds this trajectory can be computed in time $O(n^2(\frac{1}{\epsilon})^{6d-1})$.*

Proof: The n^2 in the time complexity comes from checking the safety of trajectory segments [CDRX]. The extra $d-1$ in the exponent is due to the number of μ-extremal neighbors a state may have. □

6.4 Obtaining a Polynomial-time Approximation Algorithm for a Robot with Revolute and Prismatic Joints

6.4.1 Outline

We now consider the case of an open-chain robot with d joints that obeys generalized force bounds **f** at all times. Conceptually, this differs from the L_2 case in the dependence of its dynamics bounds on the state of the system. For a given ϵ we show how to choose a grid-spacing $\boldsymbol{\mu}$ and a timestep τ so that we can use the Lemma 5.2 to show that if $\Gamma_r(t)$ obeys generalized force bounds $\mathbf{f}_{max}$ and generalized velocity bound $\mathbf{v}_{max}$, then there exists a $(\boldsymbol{\mu},\tau)$-bang trajectory that tracks the time-rescaled trajectory given by (24) closely enough to be $(1-\epsilon)\delta_v$ safe. Then, by the approach of Section 5, we have the desired algorithm.

We argue that it is possible to find conditions that yield polynomially small $\boldsymbol{\mu}_i$ and τ. This argument is based on the form of tensors in (23) and a numerical analysis theorem, and it is sufficient to obtain a polynomial-time approximation algorithm. To get an explicit bound for our algorithm, we use of dynamics bounds from [HP] to derive of sufficient conditions to apply the Coupled Tracking Lemma. As the conditions we derive share a mathematical basis with [JHCP], the development here is conceptually similar.

Suppose $\Gamma_r(t)$ is the joint-space trajectory of an open-chain robot that obeys generalized force bounds $\mathbf{f}_{max}$. Let $\Gamma'_r(t)$ be the time-rescaled trajectory given by (24). Recall that (25) implies that$\Gamma'_r(t)$ obeys tighter bounds $\mathbf{f}'_{max}$. Assume that the forces needed to hold the robot still in any configuration are bounded by[7] $\frac{5\mathbf{f}_{max}}{9}$. Then

$$\mathbf{f}'_{max\ i} \leq \frac{2\mathbf{f}_{\max\ i}}{2+\epsilon}. \tag{45}$$

Let $\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ denote the set of possible accelerations obeying bounds $\mathbf{f}_{max}$ at state $(\mathbf{p}, \dot{\mathbf{p}})$; let $\mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$ denote the possible accelerations under bounds $\mathbf{f}'_{max}$. Observe that each of these sets is d-dimensional parallelpiped.

Suppose we have τ_0 and (η_{x0}, η_{v0}). Then if Γ_q tracks Γ'_r to tolerance (η_{x0}, η_{v0}), the set $\tilde{\mathcal{A}}^{(n)}_{\Gamma'_r}$ of *constant* accelerations Γ_q can obey for the *entirety* of $\mathcal{I}_n = [n\tau_0, (n+1)\tau_0]$ while respecting $\mathbf{f}_{max}$ is given by

$$\tilde{\mathcal{A}}^{(n)}_{\Gamma'_r} = \bigcap_{\phi \in Tube_n} \mathcal{A}(\phi), \tag{46}$$

where $Tube_n = \mathcal{B}_{(\eta_x, \eta_v)} \oplus \Gamma'_r(\mathcal{I}_n)$,

Suppose we have a grid $\mathcal{G}_{\boldsymbol{\mu}}$ of spacing $\boldsymbol{\mu}$ in acceleration space and that $\boldsymbol{\mu}_i \leq \mu$ for all i. Let $\tilde{\mathcal{H}}^{(n)}_{\Gamma'_r} = \mathcal{G}_{\boldsymbol{\mu}} \cap \tilde{\mathcal{A}}^{(n)}_{\Gamma'_r}$. Let $\mathcal{H}^{(n)}_{\Gamma'_r}$ be the members of $\tilde{\mathcal{H}}^{(n)}_{\Gamma'_r}$ within μ of the boundary of $\tilde{\mathcal{A}}^{(n)}_{\Gamma'_r}$.

For each n, let $\mathcal{A}^{(n)}_{\Gamma'_r}$ be the set of all accelerations functions a trajectory obeying $\mathbf{f}'_{max}$ can follow if it is in state $\Gamma'_r(n\tau_0)$ at time $n\tau_0$. Then, if there is some $\kappa_l > 0$ such that for each n, $\mathcal{H}^{(n)}_{\Gamma'_r}$ has a uniform advantage κ_l over $\mathcal{A}^{(n)}_{\Gamma'_r}$ during the interval $[n\tau_0, (n+1)\tau_0]$, then for a given (η_x, η_v), we can apply Lemma 5.1 to find a τ such that there is a $(\boldsymbol{\mu}, \tau)$-bang trajectory that tracks Γ'_r to tolerance (η_x, η_v). Once we have this τ, we simply choose the individual $\boldsymbol{\mu}_i$; recall that if joint j is revolute, then $\boldsymbol{\mu}_j \tau^2$ must divide 4π.

For a given ϵ, we need show that we can choose $\boldsymbol{\mu}$, η_{x0}, η_{v0}, τ_0, and κ_l for a given (η_x, η_y) so this is true for arbitrary Γ_r. Specifically, we show that appropriate parameters exist so that for an arbitrary Γ_r obeying $\mathbf{f}_{max}$, $\mathcal{H}^{(n)}_{\Gamma'_r}$ has a uniform advantage of at least κ_l over $\mathcal{A}^{(n)}_{\Gamma'_r}$ for all n. (Recall that members of $\mathcal{A}^{(n)}_{\Gamma'_r}$ can be time-dependent.) We now sketch a way to make this choice for a given ϵ; we do not claim this to be the optimal.

6.4.2 Sets in Acceleration Space

The conditions on $\mathcal{H}^{(n)}_{\Gamma'_r}$ relative to $\mathcal{A}^{(n)}_{\Gamma'_r}$ will be obeyed if certain geometric conditions on $\mathcal{A}(\Gamma_q(t))$ and $\mathcal{A}'(\Gamma'_r(t))$ hold for each time interval $[n\tau_0, (n+1)\tau_0]$.

[7]Alternatively, we could formulate the problem without assumptions about the robot's strength compared to gravity and show how to track Γ_r using a less massive robot or with bounds $(1+\epsilon)^2\mathbf{f}_{max}$.

Basically, if Γ_q tracks Γ'_r closely enough, $\mathcal{A}'(\Gamma'_r(t)) \subset \mathcal{A}(\Gamma_q(t))$ for all t. If τ_0 is sufficiently small, then we would expect that $\mathcal{A}'(\Gamma'_r(I_n)) \subset \tilde{\mathcal{A}}^{(n)}_{\Gamma'_r}$. We make the following observation:

Claim 6.5 *Suppose that $\mathcal{A}'(\Gamma'_r(t)) \subset \mathcal{A}(\Gamma_q(t))$ for all t, and that $\|\mathbf{a}_q - \mathbf{a}'_r\| > 2\sqrt{d}\kappa_l$ for all $\mathbf{a}_q \in \partial\tilde{\mathcal{A}}^{(n)}_{\Gamma'_r}$ and all $\mathbf{a}'_r \in \partial\mathcal{A}'(\Gamma'_r(I_n))$. Then $\mathcal{H}^{(n)}_{\Gamma'_r}$ will have advantage of at least κ_l over $\mathcal{A}^{(n)}_{\Gamma'_r}$.*

Proof: Follows from definitions and basic properties of L_2 distance. □

Recalling the definition of $\partial\tilde{\mathcal{A}}^{(n)}_{\Gamma'_r}$, we see that we can expand this to obtain a condition on $\mathcal{A}(\Gamma_q(t))$ and $\mathcal{A}'(\Gamma'_r(t))$.

Let $f_{min} = \min_i f_{max\ i}$. Recall the dynamics equation for acceleration as a function of generalized force and state

$$\mathbf{a}(\mathbf{p}, \dot{\mathbf{p}}, \mathbf{f}) = \mathbf{M}^{-1}(\mathbf{p})\left\{\mathbf{f} - [\dot{\mathbf{p}}^T\mathbf{C}(\mathbf{p})\dot{\mathbf{p}}] - \mathbf{G}(\mathbf{p})\right\}. \tag{47}$$

Consider the sets $\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ and $\mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$. Observe that each of these sets is d-dimensional parallelpiped, and recall (45). Let λ_{min} and λ_{max} be the minimum and maximum eigenvalues of $\mathbf{M}$ over all positions. Then we claim the following.

Claim 6.6 *The minimum L_2 distance between $\partial\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ and $\mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$ is at least*

$$\alpha = \frac{\epsilon f_{min}}{(2+\epsilon)\lambda_{max}} \tag{48}$$

Proof: A geometric interpretation of this claim is that we can fit d-sphere with diameter α anywhere between the surfaces $\mathcal{S} = \partial\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ and $\mathcal{S}' = \partial\mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$. Consider the surfaces of force-extremals corresponding to $\mathbf{f}_{max}$ and $\mathbf{f}'_{max}$. Recalling (45), we can fit a d-sphere of diameter $\frac{\epsilon f_{min}}{(2+\epsilon)}$ between these two surfaces. Since $\mathbf{M}(\mathbf{p})$ is symmetric positive definite, it has orthogonal eigenvectors; so does $\mathbf{M}^{-1}(\mathbf{p})$. Since λ_{max} bounds the maximum eigenvalue of $\mathbf{M}(\mathbf{p})$ from above, $\frac{1}{\lambda_{max}}$ bounds the minimum eigenvalue of $\mathbf{M}^{-1}(\mathbf{p})$ from below. Therefore, we can fit a d-sphere of diameter $\frac{\epsilon f_{min}}{\lambda_{max}(2+\epsilon)}$ between $\partial\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ and $\mathcal{A}'(\mathbf{p}, \dot{\mathbf{p}})$. The claim follows immediately. □

Assuming global velocity bounds implies global acceleration bounds. [HP] give these bounds in a simple form using purely mathematical methods. We use global acceleration bounds to limit the possible state changes of Γ'_r and Γ_q during one time step, as functions of τ_0; call these perturbations $\gamma'_r(\tau_0)$ and $\gamma_q(\tau_0)$.

We then take (47) and compute the global perturbation bounds

$$\begin{aligned} h_r(\tau_0) &= \max\|\mathbf{a}(\mathbf{p}, \dot{\mathbf{p}}, \mathbf{f}') - \mathbf{a}(\mathbf{p} + \Delta\mathbf{p}, \dot{\mathbf{p}} + \Delta\dot{\mathbf{p}}, \mathbf{f}')\|_2 \\ h_{q0}(\tau_0) &= \max\|\mathbf{a}(\mathbf{p}, \dot{\mathbf{p}}, \mathbf{f}) - \mathbf{a}(\mathbf{p} + \Delta\mathbf{p}, \dot{\mathbf{p}} + \Delta\dot{\mathbf{p}}, \mathbf{f})\|_2 \\ h_{q1}(\eta_x, \eta_v) &= \max\|\mathbf{a}(\mathbf{p}, \dot{\mathbf{p}}. \mathbf{f}) - \mathbf{a}(\mathbf{p} + \Delta\mathbf{p}, \dot{\mathbf{p}} + \Delta\dot{\mathbf{p}}, \mathbf{f})\|_2 \end{aligned} \tag{49}$$

where $(\mathbf{p}, \dot{\mathbf{p}})$ ranges over TC, $f_i \leq f_{max\ i}$ and $f_i' \leq f_{max\ i}'$ for all i, and $(\mathbf{p}-\mathbf{p}', \dot{\mathbf{p}}-\dot{\mathbf{p}}') = (\Delta\mathbf{p}, \Delta\dot{\mathbf{p}})$ ranges over $\gamma_r'(\tau_0)$, $\gamma_q(\tau_0)$, and $\mathcal{B}_{(\eta_{x0},\eta_{v0})}(\mathbf{0})$, respectively.

Consider the set $\mathcal{K}(\mathbf{p}, \dot{\mathbf{p}}, \eta_{x0}, \eta_{v0}, \tau_0) \subset \alpha(\mathbf{p}, \dot{\mathbf{p}})$ that lies between all perturbations of $S_r(\mathbf{p}, \dot{\mathbf{p}})$ by $h_r(\tau_0)$ and all perturbations of $S_q(\mathbf{p}, \dot{\mathbf{p}})$ by $h_{q0}(\tau_0) + h_{q1}(\eta_{x0}, \eta_{v0})$. Suppose we have a choice of μ. If $\mathcal{K}(\mathbf{p}, \dot{\mathbf{p}}, \eta_{x0}, \eta_{v0}, \tau_0)$ is at least $2\mu\sqrt{d}$ "thick" everywhere, then by Claim 6.5 its intersection with the $\boldsymbol{\mu}$-grid forms a sufficient $\mathcal{H}_n'(\Gamma_r)$, with $k_l = \mu$. Thus, the following constraint governs our choice of μ, η_{x0}, η_{v0}, and τ_0:

$$2\mu_i\sqrt{d} + h_r(\tau_0) + h_{q0}(\tau_0) + h_{q1}(\eta_{x0}, \eta_{v0}) \leq \alpha. \tag{50}$$

To meet this constraint we can choose (η_{x0}, η_{v0}), μ, and τ_0 using the constraints:

$$\begin{aligned} \mu &= \tfrac{\alpha}{5\sqrt{d}}; \\ h_r(\tau_0) &< \tfrac{\alpha}{5}; \\ h_{q0}(\tau_0) &< \tfrac{\alpha}{5}; \\ h_{q1}(\eta_{x0}, \eta_{v0}) &< \tfrac{\alpha}{5}. \end{aligned} \tag{51}$$

Thus, to show that we can apply the Coupled Tracking Lemma, we need to show how fast h_r, h_{q0}, and h_{q1} shrink with τ_0, η_{x0}, and η_{v0}. (Recall that we assume global velocity bounds.)

Consider (23) and (47). Recalling the derivation of the latter configuration space dynamics equation, say from [AS], we observe that the individual terms of the tensors $\mathbf{M}(\mathbf{p})$, $\mathbf{C}(\mathbf{p})$, and $\mathbf{G}(\mathbf{p})$ are sums of products of components of $\mathbf{p}$ and their sines and cosines (e.g, $\mathbf{p}_i$, $\cos(p_j)$, and $\sin(p_k)$) and configuration independent quantities. Therefore, for sufficiently small variations $\mathbf{h} = (\Delta\mathbf{p}, \Delta\dot{\mathbf{p}})$ in state, changes in the components of $\dot{\mathbf{p}}^T\mathbf{C}(\mathbf{p})\dot{\mathbf{p}}$ and $\mathbf{G}(\mathbf{p})$ are $0(\|\mathbf{h}\|)$. By our assumption of global velocity bounds and the non-singularity of $\mathbf{M}(\mathbf{p})$, changes in $(\mathbf{p}, \dot{\mathbf{p}})$ are linear in τ. Thus, were it not for the $\mathbf{M}^{-1}$ factors in (47), the perturbations h_r, h_{q0}, and h_{q1} would clearly shrink polynomially with τ_0, η_x, and η_y.

In the following section we show how to bound $\|\mathbf{M}^{-1}(\mathbf{p}) - \mathbf{M}^{-1}(\mathbf{p}-\Delta\mathbf{p})\|$ by adding another condition on η_{x0} and τ_0. We can then compute bounds for h_r, h_{q0}, and h_{q1} in terms of τ_0, η_x, and η_y. As [HP] give explicit general bounds that can be applied to the terms on the right-hand side of (23) and their derivatives we will not show the term-by-term $0(\|\mathbf{h}\|)$ argument described above, but rather simply use results from [HP].

6.4.3 Bounding Changes in $\mathbf{M}^{-1}(\mathbf{p})$

Observe that $\|\mathbf{M}(\mathbf{p}) - \mathbf{M}(\mathbf{p}+\Delta\mathbf{p})\|_2$ is polynomially dependent on the components of $\Delta\mathbf{p}$. Let $\mathbf{M}' = \mathbf{M}(\mathbf{p}+\Delta\mathbf{p})$, and $\Delta\mathbf{M} = \mathbf{M} - \mathbf{M}'$. We wish to bound $\|\Delta\mathbf{M}^{-1}\| = \|\mathbf{M}^{-1} - \mathbf{M}'^{-1}\|$. If $\mathbf{M}$ and $\mathbf{M}'$ are inertia matrices, then they are non-singular. Let λ_{min} be the minimum of their minimum eigenvalues.

Now consider the solutions $\mathbf{x}$ and $\mathbf{y}$ to the systems

$$\begin{aligned} \mathbf{Mx} &= \mathbf{b}; \\ \mathbf{M'y} &= \mathbf{b}. \end{aligned} \tag{52}$$

By substitution, $\mathbf{M}(\mathbf{x}-\mathbf{y}) = \Delta\mathbf{My}$, and thus $\|\mathbf{M}(\mathbf{x}-\mathbf{y})\| = \|\Delta\mathbf{My}\|$. We now choose to use the L_2-norm. Since

$$\begin{aligned} \lambda_{min}\|\mathbf{x}-\mathbf{y}\|_2 &\leq \|\mathbf{M}(\mathbf{x}-\mathbf{y})\|_2, \quad \text{and} \\ \tfrac{\|\Delta\mathbf{M}\|_2\|\mathbf{b}\|_2}{\lambda_{min}} &\geq \|\Delta\mathbf{My}\|_2, \end{aligned} \tag{53}$$

it follows that

$$\|\mathbf{x}-\mathbf{y}\|_2 \leq \frac{\|\Delta\mathbf{M}\|_2\|\mathbf{b}\|_2}{\lambda_{min}^2}. \tag{54}$$

Since $\Delta\mathbf{M}$ is arbitrary except for the condition that $\mathbf{M}'$ be non-singular, this implies that

$$\|\Delta\mathbf{M}^{-1}\|_2 \leq \frac{\|\Delta\mathbf{M}\|_2}{\lambda_{min}^2} \tag{55}$$

Therefore, since $\|\Delta\mathbf{M}\|_2$ is $O(\|\Delta\mathbf{p}\|)$, so is $\|\Delta\mathbf{M}^{-1}\|_2$.

6.4.4 Acceleration Bounds and Perturbations

In deriving *explicit* general bounds for the acceleration $\mathbf{a}(\mathbf{p},\dot{\mathbf{p}},\mathbf{f})$ and the perturbations h_r, h_{q0}, and h_{q1}, we will make use of some results of [HP] to bound the right-hand side of (23) and its directional derivative. We now introduce their notation. We will continue to use $\mathbf{M}(\mathbf{p})$ to denote the inertia tensor acting as a matrix, and we use $\mathbf{M}(\mathbf{p})(\cdot,\cdot)$ to denote the tensor itself. For example, the kinetic energy of the system in state $(\mathbf{p},\dot{\mathbf{p}})$ can be expressed as $\frac{1}{2}\dot{\mathbf{p}}^T\mathbf{M}(\mathbf{p})\dot{\mathbf{p}}$ or $\frac{1}{2}\mathbf{M}(\mathbf{p})(\dot{\mathbf{p}},\dot{\mathbf{p}})$.

The n^{th} derivative of $\mathbf{M}(\mathbf{p})$ with respect to $\mathbf{p}$ is defined as follows:

$$D^n\mathbf{M}(\mathbf{p})(\alpha,\beta)(\gamma_1,...,\gamma_n) = \sum_{i,j=1}^{d}\sum_{k_1,...,k_n=1}^{d}\frac{\partial^n\mathbf{M}(\mathbf{p})_{ij}}{\partial\mathbf{p}_{k_1}...\partial\mathbf{p}_{k_n}}\alpha^i\beta^j\gamma_1^{k_1}...\gamma_n^{k_n}. \tag{56}$$

In this formalism, for a given state $(\mathbf{p},\dot{\mathbf{p}})$ and acceleration $\mathbf{a}$, (23) corresponds to

$$\mathbf{f}(\cdot) = \mathbf{M}(\mathbf{p})(\mathbf{a},\cdot) - D\mathbf{M}(\mathbf{p})(\dot{\mathbf{p}},\cdot)(\dot{\mathbf{p}}) + \frac{1}{2}D\mathbf{M}(\dot{\mathbf{p}},\dot{\mathbf{p}})(\cdot) - G(\mathbf{p})(\cdot), \tag{57}$$

with $f_i = \mathbf{f}(\mathbf{e}_i)$, where $\mathbf{e}_i$ is the usual i^th basis vector in $\Re^d$. The second and third terms on the right side are equivalent to the $[\dot{\mathbf{p}}^T\mathbf{C}(\mathbf{p})\dot{\mathbf{p}}]$ term in (23)

We will use four results ((58) through (61)) from [HP]. Let M_i denote the mass of link i, L_i its maximum length from the near joint axis, and $\overline{L_i}$ the greatest distance of its centroid from the first joint axis of the manipulator. Define $\|\mathbf{y}\|_i = \sum_{j=1}^{i}|y_i|$. Then

$$|\mathbf{M}(\mathbf{p})(\mathbf{w}_1, \mathbf{w}_2)| \leq \sum_{i=1}^{d} M_i L_i^2 \|\mathbf{w}_1\|_i \|\mathbf{w}_2\|_i; \tag{58}$$

$$|D^n\mathbf{M}(\mathbf{p})(\mathbf{w}_1, \mathbf{w}_2)(\mathbf{w}_3, ..., \mathbf{w}_{n+2})| \leq 2^d \sum_{i=1}^{d} M_i L_i^2 \|\mathbf{w}_1\|_i ... \|\mathbf{w}_{n+2}\|_i; \tag{59}$$

$$|\mathbf{G}(\mathbf{p})(\mathbf{w}_1)| \leq \sum_{i=1}^{d} g M_i \overline{L}_i \|\mathbf{w}_1\|_i; \tag{60}$$

$$|D^n\mathbf{G}(\mathbf{p})(\mathbf{w}_1, ..., \mathbf{w}_{n+1})| \leq \sum_{i=1}^{d} g M_i \overline{L}_i \|\mathbf{w}_1\|_i ... \|\mathbf{w}_{n+1}\|_i. \tag{61}$$

Now, we bound the acceleration. Let us define

$$\mathcal{F}(\mathbf{f}, \mathbf{p}, \dot{\mathbf{p}}) = \mathbf{f} - [\dot{\mathbf{p}}^T \mathrm{C}(\mathbf{p})\dot{\mathbf{p}}] - \mathbf{G}(\mathbf{p}). \tag{62}$$

Then we can rewrite (47) as

$$\mathbf{a}(\mathbf{p}, \dot{\mathbf{p}}, \mathbf{f}) = \mathbf{M}^{-1}(\mathrm{p})\mathcal{F}(\mathbf{f}, \mathbf{p}, \dot{\mathbf{p}}). \tag{63}$$

Recalling (47),

$$\begin{aligned} \|\mathbf{a}(\mathbf{p}, \dot{\mathbf{p}}, \mathbf{f})\| &\leq \|\mathbf{M}^{-1}(\mathbf{p})\| \|\mathcal{F}(\mathbf{f}, \mathbf{p}, \dot{\mathbf{p}})\| \\ &\leq \frac{1}{\lambda^M_{min}} \left\| \mathbf{f} - [\dot{\mathbf{p}}^T \mathrm{C}(\mathbf{p})\dot{\mathbf{p}}] - \mathbf{G}(\mathbf{p}) \right\| \\ &\leq \frac{1}{\lambda^M_{min}} \left\{ \|\mathbf{f}\| + \|[\dot{\mathbf{p}}^T \mathrm{C}(\mathbf{p})\dot{\mathbf{p}}]\| + \|\mathbf{G}(\mathbf{p})\| \right\}. \end{aligned} \tag{64}$$

Using (59) and recalling that $\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{d}\|\mathbf{x}\|_2$ for any d-vector $\mathbf{x}$, we obtain

$$\begin{aligned} \|[\dot{\mathbf{p}}^T \mathrm{C}(\mathbf{p})\dot{\mathbf{p}}]\| &\leq \max_{\|\mathbf{x}\|_2 \leq 1} \left\{ |D\mathbf{M}(\mathbf{p})(\dot{\mathbf{p}}, \mathbf{x})(\dot{\mathbf{p}})| + |\tfrac{1}{2} D\mathbf{M}(\mathbf{p})(\dot{\mathbf{p}}, \dot{\mathbf{p}})(\mathbf{x})| \right\} \\ &\leq 2^{d-1} 3 \textstyle\sum_{i=1}^d \sqrt{d} M_i L_i^2 \|\mathbf{v}_{max}\|_i^2 \\ &\leq 2^{d-1} 3 d^{3/2} \|\mathbf{v}_{max}\|^2 \textstyle\sum_{i=1}^d M_i L_i^2. \end{aligned} \tag{65}$$

Using (60)

$$\begin{aligned} \|\mathbf{G}(\mathbf{p})\| &\leq \max_{\|\mathbf{x}\|_2 \leq 1} \textstyle\sum_{i=1}^d g M_i \overline{L}_i \|\mathbf{x}\|_i \\ &\leq g\sqrt{d} \textstyle\sum_{i=1}^d M_i \overline{L}_i. \end{aligned} \tag{66}$$

Thus we have the following global bound for the acceleration:

$$A_{max} \leq \frac{1}{\lambda^M_{min}} \left\{ \|\mathbf{f}_{max}\| + 3d^{3/2}\|\mathbf{v}_{max}\|^2 \sum_{i=1}^{d} M_i L_i^2 + g\sqrt{d} \sum_{i=1}^{d} M_i \overline{L}_i \right\}. \tag{67}$$

We now bound $h_r(\tau_0)$, $h_{q0}(\tau_0)$, and $h_{q1}(\eta_x, \eta_v)$. Recalling the definitions (49) we obtain

$$\begin{aligned}
h_r(\tau_0) &< \tau_0 \max \left\| \left[\frac{\partial \mathbf{M}^{-1}}{\partial \mathbf{p}} \dot{\mathbf{p}} \right] \mathcal{F}(\mathbf{f}, \mathbf{p}, \dot{\mathbf{p}}) + \mathbf{M}^{-1}(\mathbf{p}) \left[\frac{\partial \mathcal{F}}{\partial \mathbf{p}} \dot{\mathbf{p}} + \frac{\partial \mathcal{F}}{\partial \dot{\mathbf{p}}} \mathbf{a} \right] \right\|; && (68) \\
h_{q0}(\tau_0) &\leq \tau_0 \max \left\| \left[\frac{\partial \mathbf{M}^{-1}}{\partial \mathbf{p}} \dot{\mathbf{p}} \right] \mathcal{F}(\mathbf{f}, \mathbf{p}, \dot{\mathbf{p}}) + \mathbf{M}^{-1}(\mathbf{p}) \left[\frac{\partial \mathcal{F}}{\partial \mathbf{p}} \dot{\mathbf{p}} + \frac{\partial \mathcal{F}}{\partial \dot{\mathbf{p}}} \mathbf{a} \right] \right\|; && (69) \\
h_{q1}(\eta_x, \eta_v) &\leq \max \left\| \left[\frac{\partial \mathbf{M}^{-1}}{\partial \mathbf{p}} \Delta \mathbf{p} \right] \mathcal{F}(\mathbf{f}, \mathbf{p}, \dot{\mathbf{p}}) + \mathbf{M}^{-1}(\mathbf{p}) \left[\frac{\partial \mathcal{F}}{\partial \mathbf{p}} \Delta \mathbf{p} + \frac{\partial \mathcal{F}}{\partial \dot{\mathbf{p}}} \Delta \dot{\mathbf{p}} \right] \right\|. && (70)
\end{aligned}$$

Recall (55), which we use to bound $\|\mathbf{M}^{-1}(\mathbf{p}) - \mathbf{M}^{-1}(\mathbf{p} + \Delta\mathbf{p})\|$, and observe that

$$\begin{aligned}
\|\mathbf{M}(\mathbf{p}) - \mathbf{M}(\mathbf{p} + \Delta\mathbf{p})\| &\leq \max_{\|x\|=\|y\|=1} |D\mathbf{M}(\mathbf{p})(\mathbf{x}, \mathbf{y})(\Delta\mathbf{p})| \\
&\leq 2^d d^{3/2} \|\Delta\mathbf{p}\| \textstyle\sum_{i=1}^d M_i L_i^2.
\end{aligned} \tag{71}$$

By substitution, we obtain

$$\|\Delta \mathbf{M}^{-1}\|_2 \leq \frac{2^d d^{3/2} \|\Delta\mathbf{p}\| \sum_{i=1}^d M_i L_i^2}{(\lambda^M_{min})^2}. \tag{72}$$

It follows that

$$\begin{aligned}
\tau_0 \left\| \frac{\partial \mathbf{M}^{-1}}{\partial \mathbf{p}} \dot{\mathbf{p}} \right\| &\leq \frac{2^d d^{3/2} \|\dot{\mathbf{p}}\tau_0\| \sum_{i=1}^d M_i L_i^2}{(\lambda^M_{min})^2} \quad \text{and} && (73) \\
\left\| \frac{\partial \mathbf{M}^{-1}}{\partial \mathbf{p}} \Delta\mathbf{p} \right\| &\leq \frac{2^d d^{3/2} \|\Delta\mathbf{p}\| \sum_{i=1}^d M_i L_i^2}{(\lambda^M_{min})^2}. && (74)
\end{aligned}$$

Applying (59) and (65), we obtain

$$\begin{aligned}
\left\| \tfrac{\partial [\dot{\mathbf{p}}^T \mathbf{C}(\mathbf{p}) \dot{\mathbf{p}}]}{\partial \mathbf{p}} \Delta\mathbf{p} \right\| &\leq \max_{\|\mathbf{x}\| \leq 1} \left\{ |D^2\mathbf{M}(\mathbf{p})(\dot{\mathbf{p}}, \mathbf{x})(\dot{\mathbf{p}}, \Delta\mathbf{p})| + |\tfrac{1}{2} D^2\mathbf{M}(\mathbf{p})(\dot{\mathbf{p}}, \dot{\mathbf{p}})(\mathbf{x}, \Delta\mathbf{p})| \right\} \\
&\leq 2^{d-1} \cdot 3d^2 \|\dot{\mathbf{p}}\|^2 \|\Delta\mathbf{p}\| \textstyle\sum_{i=1}^d M_i L_i^2;
\end{aligned} \tag{75}$$

$$\left\|\frac{\partial[\dot{\mathbf{p}}^T\mathbf{C}(\mathbf{p})\dot{\mathbf{p}}]}{\partial\dot{\mathbf{p}}}\Delta\dot{\mathbf{p}}\right\| \leq \max_{\|\mathbf{x}\|\leq 1}\left\{\begin{array}{l} |D\mathbf{M}(\mathbf{p})(\dot{\mathbf{p}},\mathbf{x})(\Delta\dot{\mathbf{p}})| + |D\mathbf{M}(\mathbf{p})(\Delta\dot{\mathbf{p}},\mathbf{x})(\dot{\mathbf{p}})| \\ +|D\mathbf{M}(\mathbf{p})(\Delta\dot{\mathbf{p}},\dot{\mathbf{p}})(\mathbf{x})| \end{array}\right\}$$

$$\leq 2^{d-1}\cdot 3d^{3/2}\|\dot{\mathbf{p}}\|\|\Delta\dot{\mathbf{p}}\|\textstyle\sum_{i=1}^d M_i L_i^2; \quad \text{and} \tag{76}$$

$$\left\|\frac{\partial\mathbf{G}(\mathbf{p})}{\partial\mathbf{p}}\Delta\mathbf{p}\right\| \leq g\sqrt{d}\textstyle\sum_{i=1}^d M_i\overline{L}_i. \tag{77}$$

We now give the final bounds for $h_r(\tau_0)$, $h_{q0}(\tau_0)$, and $h_{q1}(\eta_{x0},\eta_{v0})$. First, we define

$$\begin{aligned} H &= \textstyle\sum_{i=1}^d M_i L_i^2; \\ \overline{H} &= \textstyle\sum_{i=1}^d gM_i\overline{L}_i; \\ F &= \|\mathbf{f}\| + 2^{d-1}\cdot 3d^{3/2}\|\mathbf{v}_{max}\|^2 H + \sqrt{d}g\overline{H}; \\ F' &= \|\mathbf{f}'\| + 2^{d-1}\cdot 3d^{3/2}\|\mathbf{v}'_{max}\|^2 H + \sqrt{d}g\overline{H}. \end{aligned} \tag{78}$$

In this notation, we can rewrite (67) as

$$A_{max} \leq \frac{F}{\lambda^M_{min}}. \tag{79}$$

Finally,

$$\begin{aligned} h_r(\tau_0) &< \frac{(2^d d^{3/2}\|\mathbf{v}'_{max}\|\overline{H}F' + 2^{d-1}\cdot 3d^2\|\mathbf{v}'_{max}\|^3)\tau_0}{\lambda^M_{min}} \\ &\quad + \frac{2^{d-1}\cdot 3d^{3/2}\|\mathbf{v}'_{max}\|F'\tau_0}{(\lambda^M_{min})^2}; \end{aligned} \tag{80}$$

$$\begin{aligned} h_{q0}(\tau_0) &\leq \frac{(2^d d^{3/2}\|\mathbf{v}_{max}\tau\|HF + 2^{d-1}\cdot 3d^2\|\mathbf{v}_{max}\|^3)\tau_0}{\lambda^M_{min}} \\ &\quad + \frac{2^{d-1}\cdot 3d^{3/2}\|\mathbf{v}_{max}\tau\|F\tau_0}{(\lambda^M_{min})^2}; \end{aligned} \tag{81}$$

$$\begin{aligned} h_{q1}(\eta_{x0},\eta_{v0}) &\leq \left\{\frac{2^d d^{3/2}HF}{(\lambda^M_{min})^2} + \frac{2^{d-1}\cdot 3d^2\|\mathbf{v}_{max}\|H\sqrt{d}g\overline{H}}{\lambda^M_{min}}\right\}\eta_{x0} \\ &\quad + \frac{2^{d-1}\cdot 3d^{3/2}\|\mathbf{v}_{max}\|H}{\lambda^M_{min}}\eta_{v0}^2. \end{aligned} \tag{82}$$

We summarize these bounds by saying that $h_r(\tau_0)$ and $h_{q0}(\tau_0)$ are linear in τ_0, and that $h_{q1}(\eta_{x0}, \eta_{v0})$ is linear in η_{x0} and quadratic inη_{v0}. In other words we can write, with obvious substitutions for C_r, C_{q0}, C_{q1}, and C'_{q1}:

$$\begin{aligned} h_r(\tau_0) &< C_r\tau_0; \\ h_{q0}(\tau_0) &\leq C_{q0}\tau_0; \\ h_{q1}(\eta_x, \eta_v) &\leq C_{q1}\eta_x + C'_{q1}\eta_{v0}^2. \end{aligned} \tag{83}$$

6.4.5 Putting It All Together

Recall that we wish (η_{x0}, η_{v0}), μ, and τ_0 to obey the constraints:

$$\begin{aligned} \mu &= \tfrac{\alpha}{5\sqrt{d}}; \\ h_r(\tau_0) &< \tfrac{\alpha}{5}; \\ h_{q0}(\tau_0) &< \tfrac{\alpha}{5}; \quad \text{and} \\ h_{q1}(\eta_{x0}, \eta_{v0}) &< \tfrac{\alpha}{5}. \end{aligned} \tag{84}$$

Recalling (48) and (83), it is clear that sufficient upper bounds on η_{x0}, μ, and τ_0 are proportional to ϵ and that a sufficient upper bound on η_{v0} is proportional to $\sqrt{\epsilon}$.

Lemma 6.7 *Suppose a revolute- and/or prismatic-jointed robot obeys generalized force bounds $\mathbf{f}_{max}$ and that holding the robot immobile in any configuration never requires more than $\frac{5\mathbf{f}_{max\ i}}{9}$ at any joint i. Then for a given $\epsilon > 0$ there is a constant acceleration grid-spacing $\boldsymbol{\mu}$ such that the following is true: if the robot can follow trajectory Γ_r taking time T_r, then for a given tolerance (η_x, η_v) there is a timestep τ such that*

(A) There is a (μ, τ)-trajectory Γ_q that tracks ϵ-time-rescaled trajectory Γ'_r to tolerance (η_x, η_v). It therefore traverses the path traced by Γ_r in time $T_r(1+\epsilon)$.

(b) A sufficiently small μ is given by (51) and (83).

(c) τ shrinks polynomially with η_x and η_v. Specifically, let $\overline{\eta_x} = \min(\eta_x, \eta_{x0})$ and $\overline{\eta_v} = \min(\eta_v, \eta_{v0})$, let A_{max} be given by (79), and let τ_0 obey (84). Then a sufficient upper bound on τ is given by:

$$\tau \leq \min\left(\sqrt{\frac{f_{min}\overline{\eta_x}\epsilon}{A_{max}(70A_{max}\lambda_{max}\sqrt{d} + \epsilon(35A_{max}\lambda_{max}\sqrt{d} + 2f_{min}))}}, \frac{\overline{\eta_v}}{3A_{max}}, \tau_0\right) \tag{85}$$

Proof: For the given ϵ, obtain η_{x0}, η_{v0}, μ, and τ_0 as described above. By the development in this section, η_{x0}, η_{v0}, and τ_0 satisfy the hypothesis of Lemma 5.2. A timestep τ that obeys (85) satisfies the condition in this lemma (29), and the results follow. □

6.4.6 Final Bounds

Let l be the greater of 2π and the greatest range of a prismatic joint of the manipulator. We now bound the number of gridpoints:

$$G_{oc}(A_{max}, \mu, \tau, v_{max}, l, d) < \left(\left(\frac{v_{max}}{\mu\tau} + 1\right)\left(\frac{l}{\mu\tau^2} + 1\right)\right)^d \tag{86}$$

We would like to choose $\tau \leq \tau_0$, $\eta_x \leq \eta_{x0}$, and $\eta_v \leq \eta_{v0}$ by using lemma 4.1 and (28) as for the L_2 Cartesian robot. As safety is measured in the workspace and not in configuration space, we must scale the bounds for η_x and η_v given by (28). In particular, η_x must be divided by the maximum modulus of the forward kinematic map, and η_v must be divided by $\max_{\mathbf{p},\dot{\mathbf{p}}} \|\mathbf{J}(\mathbf{p})\dot{\mathbf{p}}\|$, where $\mathbf{J}$ is the manipulator Jacobian. Let us denote them by P_{max} and J_{max}. Then we require:

$$\begin{aligned} \eta_v &\leq \frac{c_0\epsilon}{J_{max}(c_1(1-\epsilon)+\beta)} \\ \eta_x &\leq \frac{\beta c_0\epsilon}{P_{max}(c_1(1-\epsilon)+\beta)}. \end{aligned} \tag{87}$$

and choose β to maximize τ as given by (85). That is, β maximizes

$$\min\left(\sqrt{\frac{f_{min}\beta c_0\epsilon^2}{A_{max}P_{max}(c_1(1-\epsilon)+\beta)(70A_{max}\lambda_{max}\sqrt{d}+\epsilon(35A_{max}\lambda_{max}\sqrt{d}+2f_{min}))}}, \frac{c_0\epsilon}{3A_{max}J_{max}(c_1(1-\epsilon)+\beta)}\right) \tag{88}$$

With a little straightforward computation, we find that an upper bound on τ is proportional to ϵ. We then use the previous lemma to obtain the following theorem.

Theorem 6.8 *Given generalized force bounds $\mathbf{f}_{max}$ and velocity bounds v_{max}, environment diameter l, manipulator constants $\mathcal{M}$, and positive scalars ϵ, c_0, and c_1, the $(\boldsymbol{\mu}, \tau)$-grid with μ and τ chosen to satisfy (85) and $\boldsymbol{\mu}_i \leq \mu$ has polynomial size. In addition, given obstacles $\mathcal{O}$, start $\mathbf{S}$, and goal $\mathbf{G}$, a minimal-time $(\boldsymbol{\mu}, \tau)$-grid-bang, δ'_v-safe trajectory Γ_q from $\mathbf{S}$ to $\mathbf{G}$ can be computed in $O(p(n)(\frac{A_{max}}{\boldsymbol{\mu}_{min}})^{d-1}G)$, where G is given by (86) and $p(n)$ is the time complexity of checking the δ'_v-safety of a $(\boldsymbol{\mu}, \tau)$-bang trajectory segment. For fixed 2- or 3-dimensional world size and fixed dynamics bounds this trajectory can be computed in time $O(p(n)(\frac{1}{\epsilon})^{6d-1})$.*

Proof: Given the dynamics bounds and ϵ, choose μ according to Lemma 6.7 μ only has to shrink linearly in ϵ. Choose (η_v, η_x) according to (87) and (88). Choose τ according to (85) and the lemma. Then choose $\boldsymbol{\mu}$ such that $\boldsymbol{\mu}_i \leq \mu$ for all i and such that $\boldsymbol{\mu}_j\tau^2$ divides 4π for all revolute joints j.

It follows from the lemma that for any δ_v-safe trajectory Γ_r obeying the dynamics bounds and taking time T_r, there will be a $(\boldsymbol{\mu}, \tau)$-bang trajectory Γ_q that tracks the

ϵ-time-rescaled trajectory Γ'_r closely enough so that Γ_q is $(1-\epsilon)\delta_v$-safe and ϵ-close when $t=0$ and $t=(1+\epsilon)T_r$. By the choice of $\boldsymbol{\mu}$, the TC-grid induced by $\boldsymbol{\mu}$, τ, and $(\mathbf{p}_q(0), \dot{\mathbf{p}}_q(0))$ will have $O(\mu^{-3d}\tau^{-3d})$ nodes with out-degree $0(\mu^{-3d+1})$. Thus, the search will consider $O(\mu^{-3d+1}\tau^{-3d})$, or $O((\frac{1}{\epsilon})^{6d-1})$, trajectory segments. □

7 Implementation and Experiments For the L_∞ Cartesian Robot

7.1 Brief Description

To get some idea for how our algorithm for Cartesian robots obeying L_∞ dynamics bounds might behave in practice, we have completed a preliminary Common Lisp implementation for a point robot in two dimensions obeying L_∞ dynamics bounds. This planner, which we have run on a Symbolics Lisp machine and a Sun 4/260 , is the first implementation of any algorithm that generates provably good, provably near-optimal kinodynamic plans for problems in more than one dimension.

Our planner is a simple implementation of the algorithm and uses no search-pruning heuristics. It basically does a breadth-first graph search of the TC-grid, computing a state's (vertex's) neighbors only when it is on the search frontier. A bit array is used to record whether a state has been reached, and each vertex found keeps a pointer to its "parent". Currently, the planner implements collision-avoidance and an approximation of δ'_v-safety. Each trajectory segment allowed is collision-free and δ'_v-safe at its beginning and end. True δ'_v-safety would be easy to implement and would reduce the size of the search space slightly.

Two examples of a solution found by different versions of our planner are shown in fig. 14. The start position is at lower left, and the start velocity is in the positive y direction; the goal, at the upper right, has velocity in the negative y direction. The small circles indicate the position component of the planned trajectory at each time step. The "lines" or "spikes" attached to each circle indicate the velocity at that position. In this problem, the world is 3.5 by 1.9 units; $v_{max}=0.12$; $a_{max}=0.1$; $c_0=0.3$; $c_1=0.1$; and $\epsilon=0.8051948$. τ was chosen as 0.4, resulting of a TC-grid of approximately 800,000 states. The implementation required approximately 32 hours solve the problem on a Sun 4/260.

7.2 Discussion

Despite the slowness of the our current implementation, we do not believe that the algorithm is inherently impractical. In a practical implementation we would seek to exploit parallelism at both the upper (graph-search) and lower (safety-checking) levels of our algorithm. At a low level, we observe that checking the safety of any particular (a_{max}, τ)-bang can be done in parallel, since detecting a possible collision with one obstacle

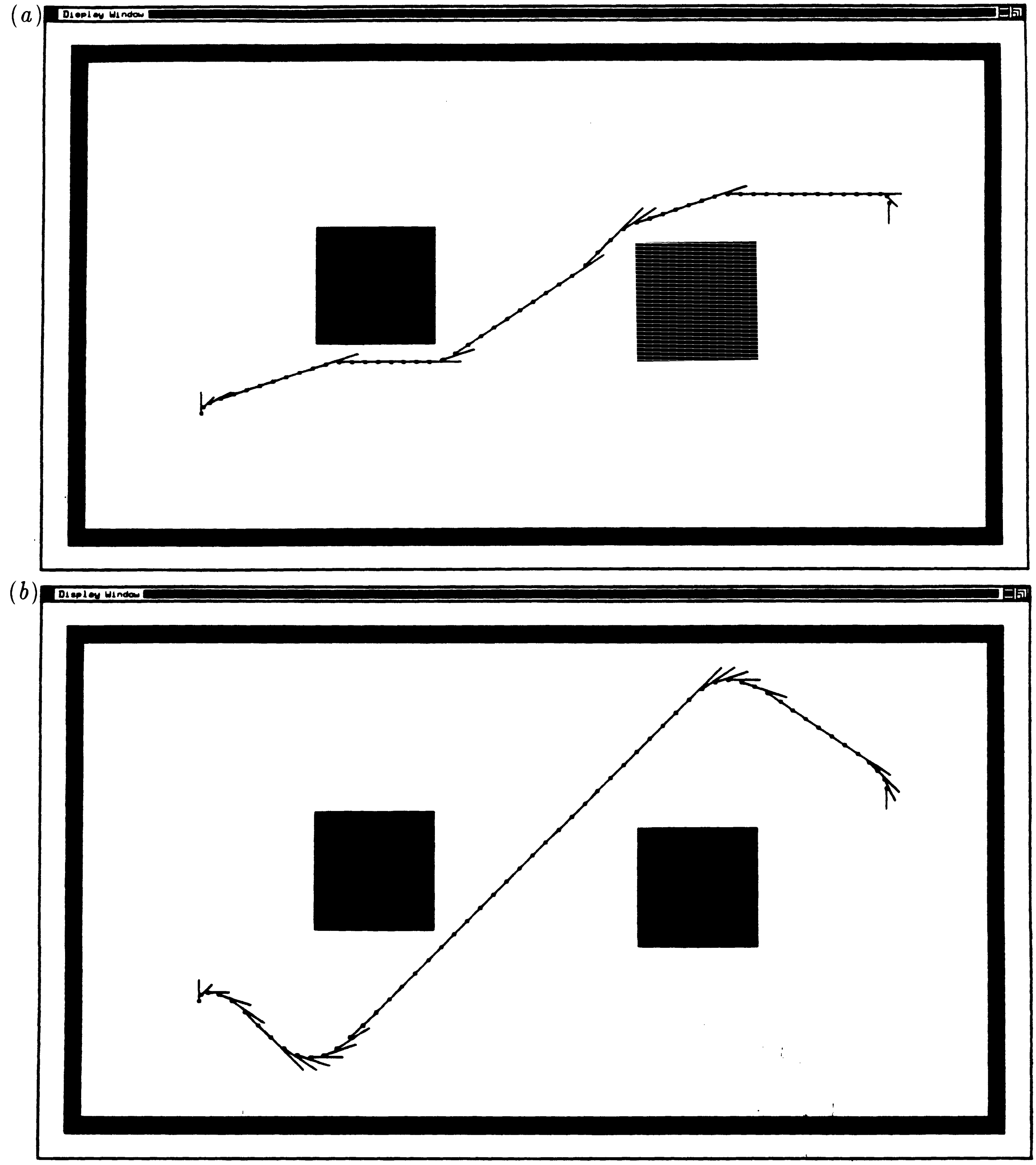


Figure 14: Near-optimal trajectories that were found by two versions of our implementation. (The "spikes" indicate the velocities). If we choose to minimize (89) as a secondary performance measure, the implementation returns the solution in (a), which is less "curvy" than the solution in (b).

face is completely independent of checking for collisions with other faces. We have found that the present implementation spends most of its time checking the safety of (a_{max}, τ)-bangs, so we would expect the speedup to be significant. At a higher level, since the TC-grid is a low-degree directed graph, it can be explored using parallel search techniques. Since safety-checking is an "inner loop", to estimate the total speedup we would multiply the effects of parallelization at the two levels.

As we stated above, we implemented the algorithm to see how it might *behave* in practice. The first aspect of behavior is the algorithm's output. As there are no prior provably good approximation algorithms for 2D optimal kinodynamic planning, *ad hoc* methods had to be used to prove the near-optimality of trajectories generated by other algorithms. Before our implementation, there was no way to study what provably good kinodynamic solutions look like without first having to solve problems by hand. We hope that experiments with the implementation will help us gain further insight into optimal kinodynamic planning.

One way in which we have experimented with the implementation is in the use of a secondary performance measure. In general, there may be many minimal-time (a_{max}, τ)-bang trajectories from s to $\mathbf{g}$, and the algorithm as described in previous sections might find any one of these. These trajectories can differ from each other in many properties, such as homotopy class, maximum curvature, and average speed. Since the algorithm returns only one trajectory, we would like to have a secondary optimization criterion that yields trajectories that are in some sense "good". One quantity of interest is

$$\int_0^T \|\mathbf{a}(t)\|_1 \, dt. \tag{89}$$

Minimizing this quantity as the secondary optimization criterion appears to produce trajectories with minimal chatter. (Chatter is a potential problem with "bang-bang" trajectories.) It also seems to select trajectories that are "globally the least curvy". (Compare the trajectories in fig. 14.)

To minimize (89) we associate a unit "fuel cost" with axial bangs and a higher cost with diagonal bangs. Let E_a denote the total fuel cost of a trajectory Γ_a. The total cost of a trajectory combines time with fuel consumption by lexicographic ordering: $\mathcal{C}(\Gamma_a) < \mathcal{C}(\Gamma_b)$ if (a) $T_a < T_b$, or (b) $T_a = T_b$ and $E_a < E_b$. At the start of any timestep n, the implemented algorithm knows a min-fuel min-time (a_{max}, τ)-bang trajectory to each state it has found. A min-fuel min-time (a_{max}, τ)-bang trajectory to a state $\mathbf{y}$ in the search frontier for the stage is found by extending a previous min-fuel min-time trajectory with an (a_{max}, τ)-bang to reach $\mathbf{y}$ with minimal total fuel cost. Thus, for a given problem, our implementation will return minimal-time (a_{max}, τ)-bang trajectory with the lowest fuel-cost .

We can, in general, incorporate a second cost function L using lexicographic ordering when the following is true of L: if $\Gamma = \Gamma_1 * \Gamma_2$. then $L(\Gamma) = L(\Gamma_1) + L(\Gamma_2)$, where "$*$" denotes trajectory concatenation. This follows from the mapping of trajectory segments to graph edges. However, we will only be guaranteed to find the (a_{max}, τ)*-bang trajectory*

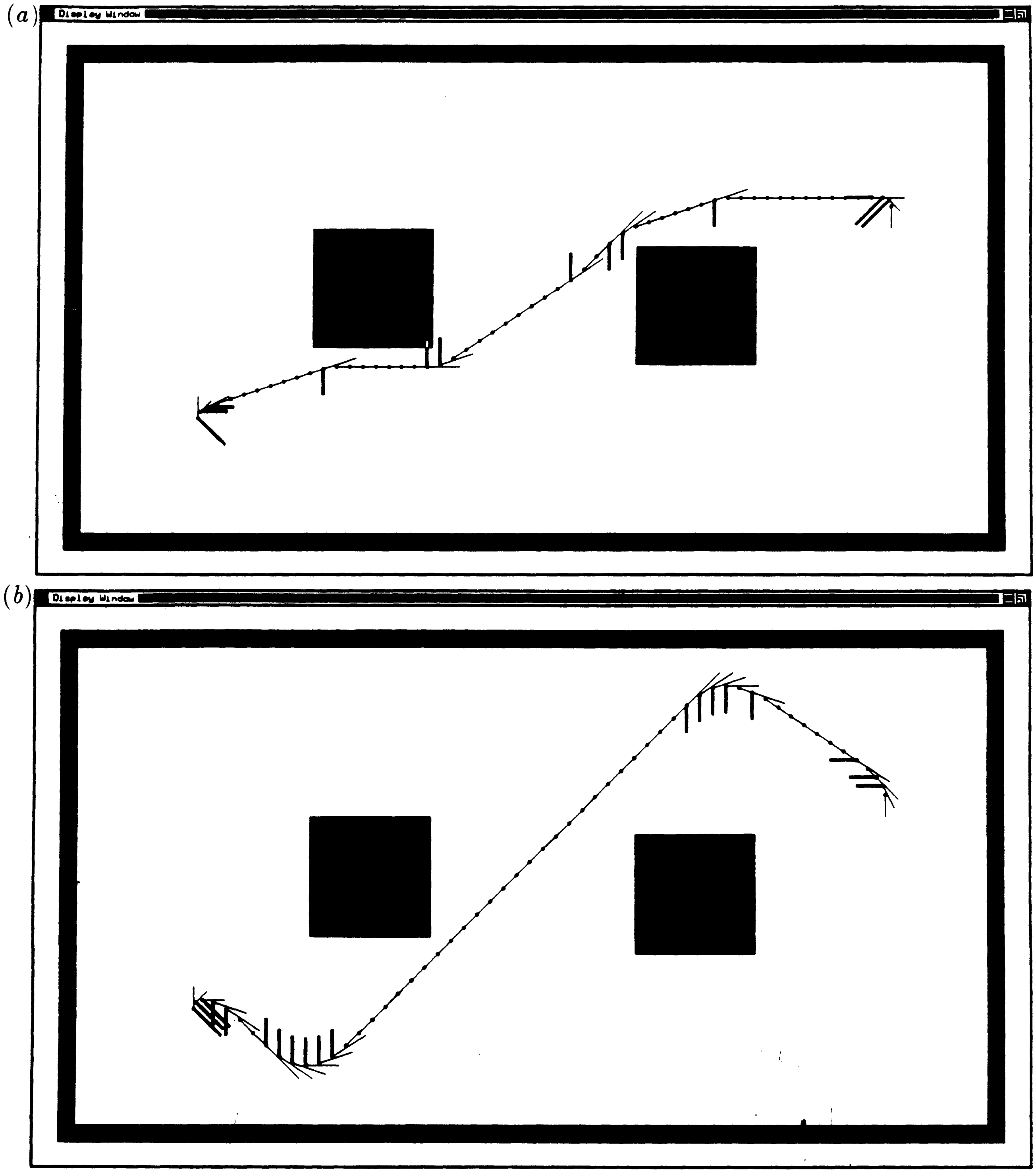


Figure 15: This figure shows the commanded accelerations for the solutions in fig. 14. *E.g.*, the commanded acceleration for the first time-step of both solutions is a bang in the $+x$ direction. The "diagonal" accelerations correspond to simultaneous ∞-norm bangs in the x and $-y$ or $-x$ and $-y$ directions. Thus, this figure represents the "plan" of commanded forces.

Γ_q with minimal L.

Let $\Gamma_{L_{opt}}$ be the time-optimal δ_v-safe trajectory with minimal L-cost. We would like to compare $L(\Gamma_q)$ to $L(\Gamma_{L_{opt}})$, but the lexicographic ordering technique does not imply any particular relationship. To prove for some predicate $\mathcal{R}$ that $\mathcal{R}(\Gamma_q, \Gamma_{L_{opt}})$, we need to extend the tracking lemma to state that if $\Gamma_{L_{opt}}$ is an L-minimal time-minimal trajectory, then there is a trajectory Γ in $\mathcal{T}_{s,g}$ such that $\mathcal{R}(\Gamma_q, \Gamma_{L_{opt}})$.

8 Conclusions

In this paper we obtain provably good approximation algorithms for kinodynamic planning that extend the results of [CDRX] in two ways.

First, we reanalyze the [CDRX] algorithm for Cartesian robots obeying L_∞ dynamics bound to (a) tighten the complexity bound from $O\left(n^2 \log n \left(\frac{1}{\epsilon}\right)^{6d}\right)$ to $O\left(n^2 \log n \left(\frac{1}{\epsilon}\right)^{3d}\right)$ and (b) show that the algorithm is trivially modified to find a trajectory taking at most time T_{opt} (the optimal time) instead of time $T_{opt}(1+\epsilon)$. Second, we extend [CDRX] to get provably good approximation algorithms for kinodynamic for robots with coupled dynamics bounds — in particular, Cartesian robots obeying L_2 bounds and d-link, 3D robots with revolute and prismatic joints and full rigid body dynamics. For these robots obtain a general $O\left(n^2(\log n)(\frac{1}{\epsilon})^{6d-1}\right)$ algorithm. where n is the geometric complexity.

To obtain these new results, we use new techniques with a game-theoretic flavor to prove several new tracking lemmas. An improved one-dimensional tracking lemma (4.2) leads directly to the improved L_∞ result. In turn, the intuition behind its proof is the basis for the proofs of the Coupled Tracking and Inductive Coupled Tracking Lemmas (5.1 and 5.2) for robots with coupled dynamics bounds. Using these two lemmas, we obtain polynomial-time approximation algorithms for Cartesian robots obeying L_2 dynamics bounds and open kinematic chain manipulators with revolute and prismatic joints.

The improved results for Cartesian robots with L_∞ dynamics bounds indicate that a practical implementation may eventually be reasonable. We describe a preliminary implementation and report the results of some simple experiments. This is the first implementation of a polynomial-time, provably good approximation algorithm for kinodynamic planning.

There are many directions for future research, among these:

1. We conjecture that the game-theoretic proofs of the Coupled Tracking and Inductive Coupled Tracking Lemmas can be adapted to show that we can track an adversary's trajectory as long as the *convex hull* of our allowable accelerations has a κ_l advantage over his allowable accelerations. This would imply a provably good polynomial-time approximation algorithm for kinodynamic planning using approximately bang-bang controls.

2. We would like a provably good polynomial-time approximation algorithm for kinodynamic planning that uses trajectories that are truly extremal rather than merely

approximately extremal. At least some of the trajectories found by such an algorithm would be optimal (for δ_v' safety). (See Section 5.3).

3. For lemmas 5.1 and 5.2 to be applicable (*e.g.* to get polynomial-time approximation algorithms for other classes of robots), it is necessary for $\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ to have dimension d. For many robot systems with non-holonomic constraints, such as wheeled mobile robots [Odo], this is not so. A tracking lemma (similar to Lemma 5.2) for such robots would allow us to extend the general [CDRX] approach to them.

4. The complexity of our algorithms can probably be improved. Other search algorithms, such as A*, may be employed in place of a breadth-first search.

5. Precise lower bounds for kinodynamic planning should be established (especially in the 2D case).

6. It is possible that the lower bound for the complexity of approximation algorithms that find *near-optimal* solutions might be make them impractical for on-line motion planning, where planning time plus trajectory time is the critical quantity rather than trajectory time alone. If this is true, in situations where an ϵ-approximation scheme would not be appropriate an algorithm that quickly finds a trajectory that is safe and close to the start and goal but not nearly time-optimal might be acceptable, *provided that we can bound the time of that trajectory reasonably in terms of the optimal trajectory time.* Problems with such a character exist in graph theory: for the NP-hard traveling salesman problem (with triangle inequality) on a graph with m edges and n vertices, we can find a minimum-spanning-tree-based solution that costs no more than twice that of the optimal solution in time $O(m \log\log n)$ [GJ, CT, T].

   Thus, we forsee a family of problems related to approximately optimal kinodynamic planning but that might be more useful investigating the on-line problem. We propose a framework in which algorithmic complexity instead of closeness to optimalitiy would be chosen *a priori*; we would bound trajectory time as a function of optimal trajectory time, robot properties, and environment complexity. We would similarly bound the observance of the start and goal states and the safety function. With a suitable set of such algorithms available, we would hope to build a kinodynamic motion planner whose performance, given a fixed time for both planning and execution, would degrade gracefully with the complexity of the environment, the size of the configuration space, and the multiplicative inverse of the safety function.

7. 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 "holes"), 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.

Kinodynamic planning represents a new direction in algorithmic motion planning. 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. There is a great deal of interesting heuristic and experimental work to be done, in reducing such algorithms to practice. Finally, since the problem has an optimization flavor, the algorithms and proof techniques draw on several branches of computer science and robotics.

A Guide to Notation

It was necessary to use a rather large number of symbols in this paper; we have tried to keep our notation consistent and "intuitive". The following is a partial glossary of symbols.

C the configuration space of a particular robot

TC the tangent bundle of this space

d the number of dimensions of the configuration space

$\mathcal{O}$ the encoding of the obstacle set

$\mathbf{p}(t)$ a path; $\mathbf{p} : Time \to C$

$(\mathbf{p}, \dot{\mathbf{p}})$ a trajectory state; $\mathbf{p}$ is position and $\dot{\mathbf{p}}$ is velocity

Γ a trajectory; $\Gamma : Time \to TC$

(η_x, η_v) a tracking tolerance

$\mathbf{a}$ acceleration

$\mathbf{f}$ generalized force vector

$\mathbf{f}_{max}$ a vector of generalized force bounds; $|\mathbf{f}_i| \leq \mathbf{f}_{max\ i}$

$\mathcal{A}(\mathbf{p}, \dot{\mathbf{p}})$ the set of all instantaneous accelerations possible at state $(\mathbf{p}, \dot{\mathbf{p}})$ obeying dynamics bounds

$\mathcal{A}(t)$ a function that gives the set of all instantaneous accelerations possible at time t; i.e., $\mathcal{A} : Time \to \{X : X \text{ is a set of accelerations}\}$

$\mathcal{A}'$ either of the two previous, but obeying tighter dynamics bounds

κ_l a minimal acceleration advantage

κ_h a maximum acceleration advantage; this is sometimes "involuntary"

μ a grid-spacing

$\boldsymbol{\mu}$ a vector of grid-spacings; $\boldsymbol{\mu}_i$ is the spacing in dimension i

τ a timestep size

$\mathcal{G}_{\boldsymbol{\mu}}$ a grid with spacing $\boldsymbol{\mu}$; by this we mean the set of points at the interstices of the ruled grid.

$\mathcal{H}$ a set of accelerations, usually the set of possible accelerations for a tracking trajectory; similar to $\mathcal{A}$

$\mathcal{H}^{(n)}$ the set of constant accelerations a (tracking) trajectory may follow for the duration of the n^{th} timestep $[n\tau, (n+1)\tau]$

$\oplus$ set sum

Subscripts r and q denote trajectories.
Subscript i denotes the i^{th} coordinate axial direction.
Subscript n denotes a timestep.

B References

[AS] Asada, H., and Slotine, J-J. E., *Robot Analysis and Control,* John Wiley & Sons, New York (1986).

[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).

[Ca] Canny, J., *Collision Detection for Moving Polyhedra,* IEEE Trans. on Pattern Analysis and Machine Intelligence, 8, 2, pp. 200 - 209 (1986).

[CDRX] Canny, J., Donald, B., Reif J., and Xavier, P., *On The Complexity of Kinodynamic Planning,* Proc. 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,* Proc. FOCS, Los Angeles (1987).

[CT] Cheriton, D., and Tarjan, R. E... *Finding Minimum Spanning Trees,* SIAM Journal of Computing, 5 (1976). pp. 724 - 742 (1976).

[Don] Donald, B., *A Search Algorithm for Motion Planning with Six Degrees of Freedom,* Artificial Intelligence, **31** (3) (1987).

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

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

[GJ] Garey, M., and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H. Freeman and Company, New York, NY (1979).

[HP] Heinzinger, G., and Paden. B., *Bounds on Robot Dynamics,* Proc. IEEE Int. Conf. On Robotics and Automation. Scottsdale, AZ (1989).

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

[JC] Jacobs, P., and Canny. J.. *Planning Smooth Paths for Mobile Robots,* Proc. IEEE Int'l. Conf. on Robotics and Automation, Scottsdale, AZ (1989).

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

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

[Odo] O'Donnell, M., *Personal communication,* (1989).

[Odu] Ó'Dúnlaing, C., *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,* Proc. IEEE Int'l. 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,* Proc. IEEE Int'l. 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).

[T] Tarjan, R. E., *Data Structures and Network Algorithms,* SIAM, Philadelphia. PA (1983).

[W] Wilfong, G., *Motion Planning for an Autonomous Vehicle,* Proc. IEEE Int'l. 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.