Approximate Kinodynamic Planning Using L_2-norm Dynamic Bounds*

John H. Reif[†] Stephen R. Tate[‡]

Abstract

In this paper we address the issue of kinodynamic motion planning. Given a point that moves with bounded acceleration and velocity, we wish to find the time-optimal trajectory from a start state to a goal state (a state consists of both a position and a velocity). As finding exact optimal solutions to this problem seems very hard, we present a provably good approximation algorithm using the L_2 norm to bound acceleration and velocity. Our results are an extension of the earlier work of Canny, Donald, Reif, and Xavier [1], who present similar results where the dynamics bounds can be examined in each dimension independently (they use the L_∞ norm to bound acceleration and velocity).

Keywords: Motion planning, kinodynamic planning, approximation algorithms, robotics.

---

*Supported in part by DARPA/ARO contract DAAL03-88-K-0195, Air Force Contract AFOSR-87-0386, DARPA/ISTO contract N00014-88-K-0458, and NASA subcontract 550-63 of prime contract NAS5-30428.

†Department of Computer Science, Duke University, Durham, NC 27706

‡Department of Computer Sciences, University of North Texas, P.O. Box 13886, Denton, TX 76203. This work was performed while the author was at Duke University.

1 Introduction

With the increasing use of industrial robots, the associated computational problems such as planning and control are receiving a lot of attention. The basic foundation for motion planning comes from geometric problems, such as finding a path for an object (robot) which avoids a set of obstacles (known as the "Piano Movers' Problem") [12]. Even for the case of the robot being a simple point, finding the *shortest* path through a set of objects can be very difficult in three dimensions, but a fully polynomial approximation algorithm was given by Papadimitriou [10]. Unfortunately, these problems do not take into account the physical limitations of a real robot (for instance, the shortest path between two points will usually involve an instantaneous change in the direction of motion); furthermore, it is much more important to consider a path that takes the shortest *time* rather than covering the shortest distance. With this in mind, the problem of kinodynamic motion planning addresses these real-world issues.

Kinodynamic planning extends kinematic planning (avoiding a set of static obstacles) by including dynamics (or dynamical) constraints, such as dynamics laws (e.g. $\mathbf{f} = m\mathbf{a}$) and dynamics bounds (a maximum allowable acceleration $a_{\max}$ and velocity $v_{\max}$). In addition to simply finding a trajectory between a start state and a goal state (a state consists of both a position and a velocity), it is desirable to find the *optimal* trajectory, i.e., the trajectory that takes the least amount of time. Dynamics bounds are given by bounding the norm of the vectors that represent velocity and acceleration. As finding optimal trajectories is computationally intensive, practical algorithms must focus on *approximately optimal* trajectories; specifically, an approximation algorithm will find a trajectory connecting the start state and goal state that requires time only slightly greater than the time required by the optimal trajectory. Previously, an approximation algorithm was known when the dynamics bounds are stated in terms of the L_∞ norm [1]; however, while such a case is easier to show (due to the independence of the dimensions), it relies on somewhat artificially imposed properties, such as the orientation of the coordinate axes.

In this paper, we present an approximation algorithm that uses the L_2 norm for dynamics bounds; our results parallel those of Canny, Donald, Reif, and Xavier [1], but the proof techniques are very different. In independent work concurrent with the research presented in this paper, Donald and Xavier have also developed an approximation algorithm with dynamics bounds stated in terms of L_2 norms [5].

Optimal kinodynamic planning seems to be very hard in practical situations; the only exact solutions to the optimal kinodynamic planning problem are for one or two dimensions. In fact, in three dimensions (or more) finding a minimum distance path has been shown to be NP-hard [3], and this proof can be used to show that finding the exact solution for kinodynamic planning in ≥ 3 dimensions is NP-hard. However, as with many NP-hard problems, it is possible to find an approximately optimal solution in polynomial time; as we show here, the goodness of the approximation can be bounded by a proven scalar multiple. In other words, if the optimal solution is a robot trajectory that takes time T, then for any given $\epsilon > 0$ we can find a solution that takes time at most $(1+\epsilon)T$ by a search algorithm whose running time is polynomial both in the complexity of the environment and in $\frac{1}{\epsilon}$.

In real life there are additional problems to address (such as external forces) that we do not address in this paper. One additional real-world property that we *do* address is the inability of real robots to navigate accurately at high speeds. To this end, we use the notion of "safe" and "also-safe" trajectories introduced in [1]; basically, this concept uses an affine mapping from speed (i.e., magnitude of velocity) to distance that bounds how close the robot may be to an obstacle. Exact definitions of "safe" and "also-safe" trajectories can be found in section 5. The robot model that we use is simply a point robot with unit mass; non-point robots can be handled easily by "growing" the obstacles to reflect the shape of the robot. It should be noted that the approximation algorithm we present is extremely simple; the complex equations found in this paper are used exclusively for proving the correctness of the algorithm.

1.1 Summary of Previous Work

Much of the previous work in motion planning and related problems was mentioned above. The motion planning problem has been studied from a variety of movement constraints; in addition to the problem of kinodynamic motion planning as defined above, Fortune and Wilfong [6] examine the problem of motion planning where the moving object has a bounded turning radius. This problem was further examined by Jacobs and Canny, who present a polynomial time approximation algorithm for finding such a path [7].

A problem which can be viewed as one dimensional kinodynamic planning (with moving obstacles) is examined by Ó'Dúnlaing [8], who gives an algorithm for the *exact* optimal solution. A recent result of Canny, Rege, and Reif gives a PSPACE algorithm that finds an exact solution for the two-dimensional case [2]; unfortunately, finding exact solutions in higher dimensions with polynomial time algorithms is extremely unlikely as Canny and Reif have shown the shortest path problem (in three or more dimensions) to be NP-hard [3]. Polynomial time approximation algorithms for arbitrary (but fixed) dimensions are examined in a variety of papers [1], [4], [5], [11], [13] (Note, however, that [11] and [13] do not prove bounds on the goodness of their approximation).

2 Preliminaries

2.1 Definitions and Terminology

Before starting the technical material, we will present the definitions and terminology that are used in this paper. All vector variables will be typeset in boldface, to separate them from scalars which are typeset in standard math italics. For example, $\mathbf{v}$ is a vector (of reals), and t is a scalar real. First and second derivatives are denoted by superscripted dots as in standard control theory literature. For example, if $\mathbf{p}(t)$ is a (twice differentiable) function, then $\dot{\mathbf{p}}(t)$ is its first derivative, and $\ddot{\mathbf{p}}(t)$ is its second derivative.

Consider a point traveling through d-dimensional Euclidean space. By a trajectory Γ, we mean both the velocity and position of the path that the point takes. By a point on a trajectory, we mean both the position and velocity at a particular time; for example, the endpoints can be given by $(\mathbf{p_0}, \mathbf{v_0})$ and $(\mathbf{p_1}, \mathbf{v_1})$, where $\mathbf{p_0}$ and $\mathbf{p_1}$ are the starting and ending positions, respectively, and $\mathbf{v_0}$ and $\mathbf{v_1}$ are the starting and ending velocities. If trajectory Γ takes time T, we say that Γ is a time T trajectory. For a subscripted trajectory Γ_r, we denote the position at time t by $\mathbf{p}_r(t)$, the velocity by $\dot{\mathbf{p}}_r(t)$, and the acceleration by $\ddot{\mathbf{p}}_r(t)$. The change (from time 0) in any of these functions is represented by a delta prefix; for example, the change in position is $\Delta\mathbf{p}_r(t) = \mathbf{p}_r(t) - \mathbf{p}_r(0)$. Similar definitions hold for $\Delta\dot{\mathbf{p}}_r(t)$ and $\Delta\ddot{\mathbf{p}}_r(t)$. The environment is a set of polyhedral obstacles in d-dimensional space, where d is considered to be a small constant.

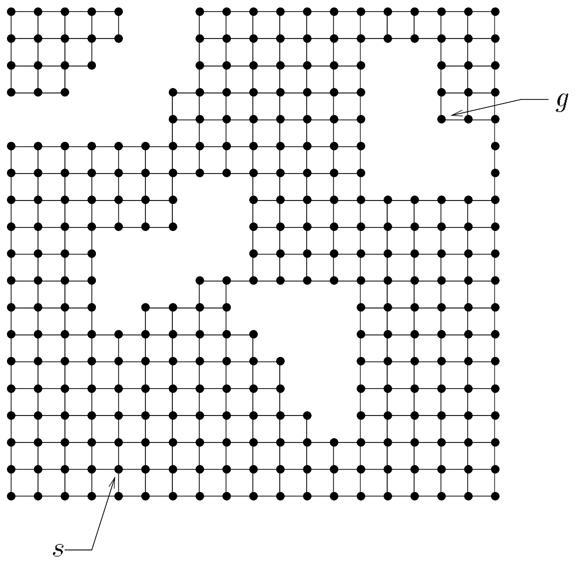

Figure 1: An example grid search problem

The 2-norm of a vector $\mathbf{v}$ is written as $\|\mathbf{v}\|_2$, and the infinity norm is $\|\mathbf{v}\|_\infty$. Hereafter, if we write simply $\|\mathbf{v}\|$ without a subscript, the 2-norm should be understood.

The set of obstacles in the environment is represented by $\mathcal{E}$. All obstacles are polyhedral and require a total of n bits to encode. Furthermore, it is assumed that the space in which the robot may move is bounded by a ball of diameter D.

2.2 Outline of Algorithm and Proof

Consider the following search problem: we are given a subgraph of a d-dimensional grid-graph; in other words, a grid-graph with some vertices missing. There are two distinguished vertices s and g, and we want to know if there is a path from s to g (an example in two dimensions is shown if figure 1). This problem is easy to solve using depth-first search on the graph; a minimum distance path from s to g can be found (if a path exists) by using breadth-first search.

The problems we are interested in for this paper are similar, but involve searching a continuous space. By a *discretization* of the environment with grid-length g, we are referring to a graph constructed from the environment as follows. First, construct a graph with nodes for each point $(i_1 g, i_2 g, ..., i_d g)$ in the environment, where each i_j is an integer; since the environment is bounded by a ball of diameter D, the graph is finite. Edges are added between neighboring vertices to form a grid-graph. Finally, the vertices that lie inside any obstacle are removed from the graph.

The graph of figure 1 is such a graph — the missing parts of the grid correspond to obstacles. Simple reachability problems can be answered using this graph: by making g small enough we can guarantee that there exists a continuous path in the environment if and only if there exists a path on the constructed grid-graph, and a breadth-first search on the grid-graph gives an approximately minimum distance path in the continuous environment. Unfortunately, even this simple reachability problem requires a grid whose size grows exponentially with the algebraic complexity of the environment. We use a variant of this strategy that requires only a polynomial size graph (described fully in section 3) to solve approximate kinodynamic planning.

The proof of the correctness of our algorithm is based on a *tracking theorem* (theorem 4.2). This theorem states that for *any* continuous trajectory Γ_e, there exists a trajectory Γ_a that travels only between grid-points of our discretization and is always close (in both position and velocity) to the continuous trajectory Γ_e. Thus, the *minimum time* continuous trajectory has a corresponding approximating trajectory in the constructed grid, and this approximating trajectory can be found by simple breadth-first search. Since any discovered trajectory between grid-points is also a valid continuous trajectory, we never find an invalid trajectory, and the correctness of the approximation algorithm follows.

The proof of the tracking theorem is rather involved, so we outline it here. First we show that any continuous trajectory can be stretched in time so that it takes slightly longer, but the new trajectory meets a smaller acceleration bound (lemma 4.2). Thus, when approximating the slowed-down continuous trajectory, the additional acceleration available to the approximating trajectory can be used to reach a grid-point that is close to the continuous trajectory. Unfortunately, there may still be some position error build-up while approximating the continuous trajectory, so we alternate phases of approximating with phases of error correction. A slightly modified continuous trajectory that doesn't change velocity during the error correction phase is shown to exist (lemma 4.3), and this trajectory is used in the approximating phases instead of the original one. By making the approximating and error correcting phases short enough, we show that the constructed trajectory is still a good approximation of the original continuous trajectory, which completes the proof of the tracking theorem.

3 Constructing a Grid

For our kinodynamic planning approximation, we build a grid of points in state space, rather than just in the position as outlined above. The approximation proceeds in time steps of length τ as follows: At all times $i\tau$ (i an integer), the velocity that is desired at time $(i+1)\tau$ is chosen from the neighbors of the current state, and the trajectory in the time interval $(i\tau, (i+1)\tau)$ is a linear transition to the desired next velocity (i.e., constant acceleration). Notice that the position at time $i\tau$ and the selected velocity transition completely determine the position at time $(i+1)\tau$. For such a discrete step method, we must show that it is possible to stay reasonably close to an exact path by this method of moving between neighboring grid-points. Note that while we still refer to our discretization as a grid, it is *not* a regular grid-graph in position space — the actual structure is a grid-graph in *velocity space*, along with the positions that correspond to moves on this velocity grid.

Since we want to define a finite grid, at any time step there must be finitely many choices for the change in velocity over the next time interval. If we let $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k$ be these vectors (called *choice vectors*), then for each vector $\mathbf{v}_i$ we can determine θ_i, the smallest angle between $\mathbf{v}_i$ and any other choice vector. Remember that these vectors are actually *change* in velocity vectors, so the velocity at time $(i+1)\tau$ is $\dot{\mathbf{p}}(i\tau) + \mathbf{v}_j$ for the chosen vector $\mathbf{v}_j$. We always include the zero vector ($\mathbf{0}$) in a set of choice vectors to denote that it is possible to stay at the current velocity during a time interval; thus the set of choice vectors referred to above is $V = \{\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$. We now argue that θ_i must vary with ϵ if we bound the 2-norm of the acceleration; this implies that the number of choice vectors must grow as ϵ decreases.

Assume that the angles do not vary with ϵ, and pick a particular non-zero θ_i. Let $\mathbf{v}_m$ be a choice vector that makes angle θ_i with $\mathbf{v}_i$. Consider a continuous path with maximum acceleration at an angle that exactly bisects the angle made by $\mathbf{v}_i$ and $\mathbf{v}_m$; it should be obvious that by making ϵ sufficiently small, the exact path taking time T simply outruns any path made up of choice vectors taking time $(1+\epsilon)T$. In other words, any approximating path will fall farther and farther behind the exact path. In particular, in two (or more) dimensions we can show that there needs to be $\Omega(\frac{1}{\epsilon})$ choice vectors to approximate within an ϵ factor of optimal.

Now we examine how to vary the angle between choice vectors with ϵ. The first method that comes to mind is to simply use maximal acceleration vectors at angles that are evenly spaced (and varying with ϵ); unfortunately, this gives rise to a "grid" that grows exponentially with the number of time steps, and in fact does not even form a finite graph. The method we actually use is to superimpose a square grid on top of this set of choices, and then using parts of this grid with a new neighbor relationship, we have a grid that grows polynomially with the number of time steps. For a small enough square grid, we can track velocities closely; a more formal presentation of this follows.

Definition 3.1 A set of choice vectors $\{\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$ is called δ-dense $(0 < \delta < 1)$ if for *any* non-zero vector $\mathbf{v}$ there exists a non-zero choice vector $\mathbf{v}_i$ such that

$$\frac{\mathbf{v}_i \cdot \mathbf{v}}{\|\mathbf{v}_i\|\|\mathbf{v}\|} \geq \delta.$$

What this means geometrically is that given any vector $\mathbf{v}$, you can always find a choice vector $\mathbf{v}_i$ such that the angle between $\mathbf{v}$ and $\mathbf{v}_i$ is small (less than or equal to $\arccos \delta$).

The easiest way to obtain a δ-dense set of vectors is to space unit vectors evenly with respect to angles. As mentioned above, this is not good enough for our application, so we consider a square grid with small grid length. A set of "almost unit length" (i.e., within one grid length of unit length, but never more than unit length) choice vectors can be constructed using these grid-points while assuring that the set is δ-dense. A set of $(1 - \frac{\epsilon}{4(1+\epsilon)})$-dense choice vectors on a square grid with grid-length $\frac{\epsilon}{4}$ (exactly the conditions required by the following theorem) is illustrated in figure 2 for the specific case of two dimensions and $\epsilon = \frac{1}{2}$. The dots represent the points of the square grid, and the circle is a unit radius circle drawn for reference.

Theorem 3.1 For $0 < \epsilon \leq 1$, let $V = \{\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$ be a set of $(1 - \frac{\epsilon}{4(1+\epsilon)})$-dense choice vectors that are "almost unit length" (as defined above) on a square grid with grid-length $\frac{\epsilon}{4}$. Then for any vector $\mathbf{v}$ with $\|\mathbf{v}\| \leq 1 + \frac{1}{1+\epsilon}$, there is a choice vector $\mathbf{v}_c$ with $\|\mathbf{v} - \mathbf{v}_c\| \leq 1$.

Proof: Let $\mathbf{v}$ be any vector with $\|\mathbf{v}\| \leq 1 + \frac{1}{1+\epsilon}$. Since $V = \{\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$ is a set of

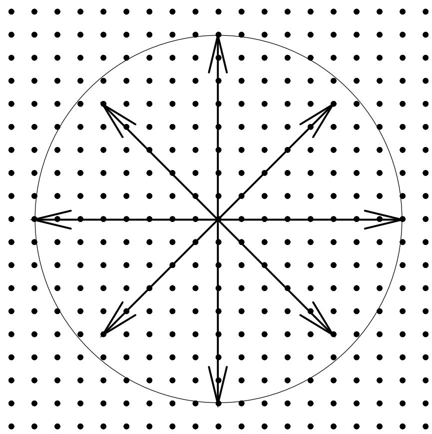

Figure 2: Possible choice vectors in two dimensions for $\epsilon = \frac{1}{2}$.

$\left(1 - \frac{\epsilon}{4(1+\epsilon)}\right)$-dense choice vectors, there exists a $\mathbf{v}_c \in V$ such that

$$\mathbf{v} \cdot \mathbf{v}_c \geq \left(1 - \frac{\epsilon}{4(1+\epsilon)}\right) \|\mathbf{v}\| \|\mathbf{v}_c\|. \tag{1}$$

We are interested in finding $\|\mathbf{v} - \mathbf{v}_c\|$. A simple geometric identity states that

$$\|\mathbf{v} - \mathbf{v}_c\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{v}_c\|^2 - 2\mathbf{v} \cdot \mathbf{v}_c = \|\mathbf{v}\|^2 + \|\mathbf{v}_c\|^2 - 2\|\mathbf{v}\|\|\mathbf{v}_c\| \cos\theta,$$

where θ is the angle between $\mathbf{v}$ and $\mathbf{v}_c$. Fixing $\|\mathbf{v}_c\|$ and θ and viewing the above equation as a polynomial in $\|\mathbf{v}\|$, differentiating with respect to $\|\mathbf{v}\|$ shows that the *minimum* value of $\|\mathbf{v} - \mathbf{v}_c\|^2$ occurs when $\|\mathbf{v}\| = \|\mathbf{v}_c\| \cos\theta$. For all $\|\mathbf{v}\| < \|\mathbf{v}_c\| \cos\theta$, the maximum value for $\|\mathbf{v} - \mathbf{v}_c\|^2$ occurs at the smallest possible value for $\|\mathbf{v}\|$; i.e., at $\|\mathbf{v}\| = 0$. When $\|\mathbf{v}\| = 0$, it is obvious that $\|\mathbf{v} - \mathbf{v}_c\| = \|\mathbf{v}_c\| \leq 1$.

It is also seen that for all $\|\mathbf{v}\| > \|\mathbf{v}_c\| \cos\theta$, the quantity $\|\mathbf{v} - \mathbf{v}_c\|^2$ is monotonically increasing, so the maximum value occurs at the largest allowable value for $\|\mathbf{v}\|$; in other words, when $\|\mathbf{v}\| = 1 + \frac{1}{1+\epsilon}$. Similar arguments show that $\|\mathbf{v} - \mathbf{v}_c\|^2$ is maximized when $\|\mathbf{v}_c\| = 1 - \frac{\epsilon}{4}$ and $\cos\theta = 1 - \frac{\epsilon}{4(1+\epsilon)}$. In other words, for all $\mathbf{v}$ such that $\|\mathbf{v}\| \leq 1 + \frac{1}{1+\epsilon}$, there exists a choice vector $\mathbf{v}_c$ such that

$$\|\mathbf{v} - \mathbf{v}_c\|^2 \leq \left(1 + \frac{1}{1+\epsilon}\right)^2 + \left(1 - \frac{\epsilon}{4}\right)^2 - 2\left(1 + \frac{1}{1+\epsilon}\right)\left(1 - \frac{\epsilon}{4}\right)\left(1 - \frac{\epsilon}{4(1+\epsilon)}\right).$$

Algebraic manipulation reveals that the right side of the above inequality is equivalent to

$$1 - \frac{\epsilon(8 + 3\epsilon - \epsilon^3)}{16(1+\epsilon)^2}.$$

In this form, it is obvious that for all valid ϵ (i.e., all ϵ satisfying $0 < \epsilon \leq 1$), $\|\mathbf{v} - \mathbf{v}_c\|^2 \leq 1$. This completes the proof of the theorem. □

This theorem is used to show that with a certain finite set of choice vectors for the change in velocity, any exact trajectory can be closely tracked using only velocity changes from the set of choice vectors; the direct application of this theorem can be found in the text following lemma 4.3.

To see how trajectories are constructed from a set of choice vectors, let τ denote the length of one discrete time interval. Consider a trajectory with an acceleration bound of a. The most that the velocity can change during one time interval is $a\tau$, so we consider this to be one "unit length"; it is obvious that theorem 3.1 applies using this as one unit, and this fact is made explicit in the following corollary.

Corollary 3.1 For $0 < \epsilon \leq 1$, let $V = \{\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$ be a set of $(1 - \frac{\epsilon}{4(1+\epsilon)})$-dense choice vectors that are "almost $a\tau$ length" on a square grid with grid length $\frac{\epsilon}{4}a\tau$. Then for any vector $\mathbf{v}$ with $\|\mathbf{v}\| \leq \left(1 + \frac{1}{1+\epsilon}\right) a\tau$, there is a choice vector $\mathbf{v}_c$ with $\|\mathbf{v} - \mathbf{v}_c\| \leq a\tau$.

Now consider a trajectory made up of N time intervals. Let $i : \{0, 1, ..., N-1\} \rightarrow \mathbf{Z}^+$ be an indexing function such that at the beginning of time interval t, we decide to use choice vector $\mathbf{v}_{i(t)}$. First, a preliminary lemma shows how the position component of a trajectory is affected by the schedule of choice vectors taken. The proof of the lemma is omitted, but is trivial; simply integrating over the velocity function defined by the indexing function gives the formula in the lemma. Notice that the velocity at any time $k\tau$ is given by $\dot{\mathbf{p}}(0) + \sum_{t=0}^{k-1} \mathbf{v}_{i(t)}$.

Lemma 3.1 If i is an indexing function as above, then the total change in position is given by

$$\Delta\mathbf{p}_a = \dot{\mathbf{p}}(0)N\tau + (N - \frac{1}{2})\tau \sum_{k=0}^{N-1} \mathbf{v}_{i(k)} - \tau \sum_{k=0}^{N-1} k\mathbf{v}_{i(k)}. \quad (2)$$

4 Tracking in the Absence of Obstacles

Before talking about trajectories that avoid obstacles, we must first show how paths can be constructed on our grid. To simplify this, arbitrary trajectories are shown to be easily approximated by a series of moves on the grid, with no obstacles in the environment.

The following lemma is stated in general terms, and will be used in several ways. Applications will be discussed after the proof of the lemma.

Lemma 4.1 Let $f : [0, T] \to \mathbf{R}$ be a continuous real-valued function on the closed interval $[0, T]$. If we know that $f(0) = f_0$, $f(T) = f_0 + \Delta f$, and that $|\frac{df(t)}{dt}| \leq a$ for all $t \in [0, T]$, the following inequalities must hold:

$$f_0 T + \frac{\Delta f T}{2} + \frac{(\Delta f)^2}{4a} - \frac{aT^2}{4} \leq \int_0^T f(t)dt \leq f_0 T + \frac{\Delta f T}{2} - \frac{(\Delta f)^2}{4a} + \frac{aT^2}{4}$$

Proof: First we argue that for any function $f(t)$ satisfying the end-point and derivative constraints of the lemma, the following inequalities must hold for all times t in the interval $[0, T]$.

$$f(t) \leq f_0 + at \tag{3}$$

$$f(t) \leq f_0 + \Delta f + a(T - t) \tag{4}$$

Consider equation (3). If the inequality does not hold, then there exists a time t_1 such that $f(t_1) > f_0 + at_1$, and by the mean value theorem of derivatives there must be some time t_2 in the interval $[0, t_1]$ such that $f'(t_2) = \frac{f(t_1) - f_0}{t_1} > a$. This contradicts our bound on the derivative as stated in the lemma, so cannot be true; therefore, equation (3) must hold. The argument for equation (4) is similar.

Since any function that satisfies the constraints of the lemma must satisfy both upper bounds of equations (3) and (4), it must satisfy the least of the two at any particular time. Let $g_1(t) = f_0 + at$ and $g_2(t) = f_0 + \Delta f + a(T - t)$, and define $g(t) = \min\{g_1(t), g_2(t)\}$. A simple check of $g(t)$ shows that it satisfies the constraints of the lemma, and by the above argument must be the point-wise maximum of all valid functions.

Since $g(t)$ is the point-wise maximum of all valid functions, the definite integral of $g(t)$ over the interval $[0, T]$ must also be greater than that of any other valid function. Actually calculating this integral gives the upper bound stated in the lemma. The proof of the lower bound is similar. □

The most immediate and obvious result is stated in the following corollary.

Corollary 4.1 If we let Γ be a one dimensional time T trajectory from starting state $(p(0), \dot{p}(0))$ to goal state $(p(T), \dot{p}(T))$ that obeys acceleration bound a, then we can say that

$$p(T) \leq p(0) + \dot{p}(0)T + \frac{\Delta\dot{p}(T)T}{2} - \frac{(\Delta\dot{p}(T))^2}{4a} + \frac{aT^2}{4}$$

and

$$p(T) \geq p(0) + \dot{p}(0)T + \frac{\Delta\dot{p}(T)T}{2} + \frac{(\Delta\dot{p}(T))^2}{4a} - \frac{aT^2}{4}.$$

Further uses of lemma 4.1 will occur when we bound the norm of the integral of vector functions.

The following lemma explains how we can reduce the acceleration bound of a trajectory and still meet the same endpoints. This occurs with a corresponding increase in the time required by the trajectory. Henceforth, assume that whenever ϵ is mentioned, it satisfies $0 < \epsilon \leq 1$.

Lemma 4.2 Given a time T trajectory Γ_r from $(\mathbf{p}_r(0), \mathbf{0})$ to $(\mathbf{p}_r(T), \mathbf{0})$ with acceleration bound a, then there exists a trajectory Γ_q with acceleration bound $\frac{a}{(1+\epsilon)^2}$ and the same endpoints, but takes time $(1+\epsilon)T$.

Proof: Simply let $\ddot{\mathbf{p}}_q(t) = \ddot{\mathbf{p}}_r(\frac{t}{1+\epsilon})/(1+\epsilon)^2$ with $\dot{\mathbf{p}}_q(0) = 0$ and $\mathbf{p}_q(0) = \mathbf{p}_r(0)$. The verification that the ending conditions are met is now a simple calculus problem, and the details are omitted. □

The problem we must now overcome is that given the endpoints of a trajectory, in general we know very little about what happens between the endpoints. The next lemma is designed to solve this problem. Example trajectories as constructed by the lemma are shown in figures 3 and 4. These examples are one dimensional trajectories, and the horizontal axis represents time.

Lemma 4.3 If we let $c = \frac{\sqrt{9+8\epsilon}-1}{2(1+\epsilon)}$ (note that $c < 1$ for all valid ϵ, and $c \to 1$ as $\epsilon \to 0$), then given an arbitrary time T trajectory Γ_r with acceleration bound $\frac{a}{(1+\epsilon)^2}$, there exists a time T

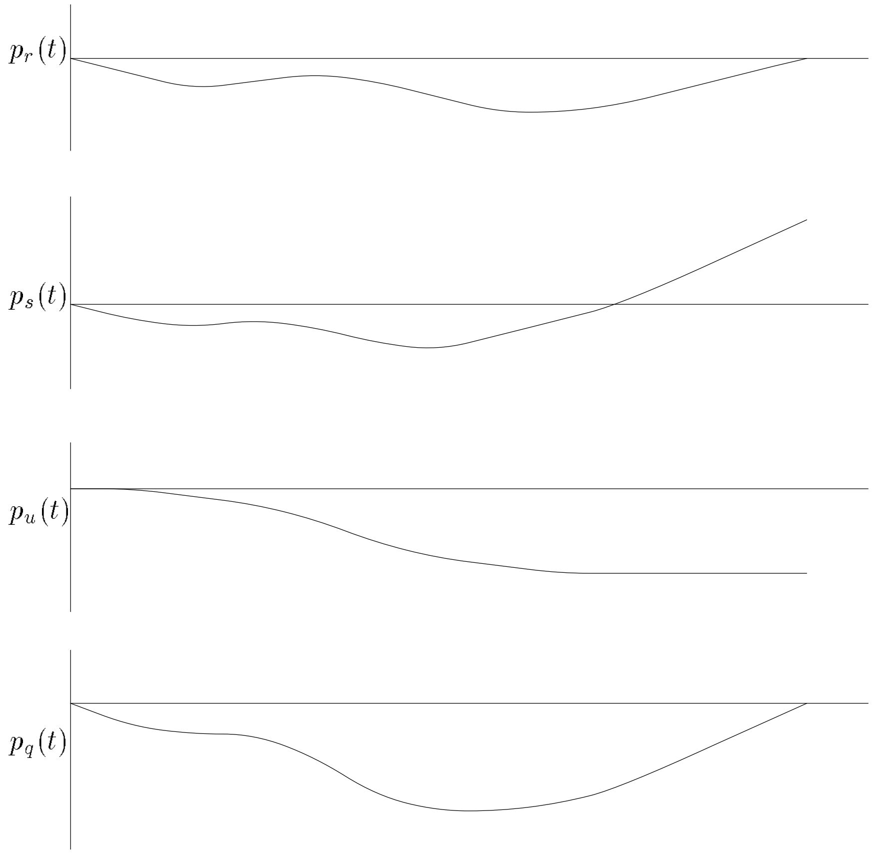

Figure 3: Position graphs for trajectories in lemma 4.3.

trajectory Γ_q which has the same endpoints but does not change velocity for the last time interval of length $(1-c)T$. Furthermore, Γ_q meets acceleration bound $\frac{a}{1+\epsilon}$.

PROOF: We define a temporary trajectory Γ_s by specifying that $\mathbf{p}_s(0) = \mathbf{p}_r(0)$, and then defining the velocity to be a "time-compressed" version of $\dot{\mathbf{p}}_r(t)$. More specifically,

$$\dot{\mathbf{p}}_s(t) = \begin{cases} \dot{\mathbf{p}}_r(\frac{t}{c}) & \text{, for } 0 \le t \le cT \\ \dot{\mathbf{p}}_r(T) & \text{, for } cT < t \le T \end{cases}$$

It is easy to see that $\mathbf{p}_s(T) = (1-c)\mathbf{p}_r(0) + c\mathbf{p}_r(T) + (1-c)T\dot{\mathbf{p}}_r(T)$, and that the velocity at both endpoints of Γ_s is the same as the corresponding velocities of Γ_r. Now we define another

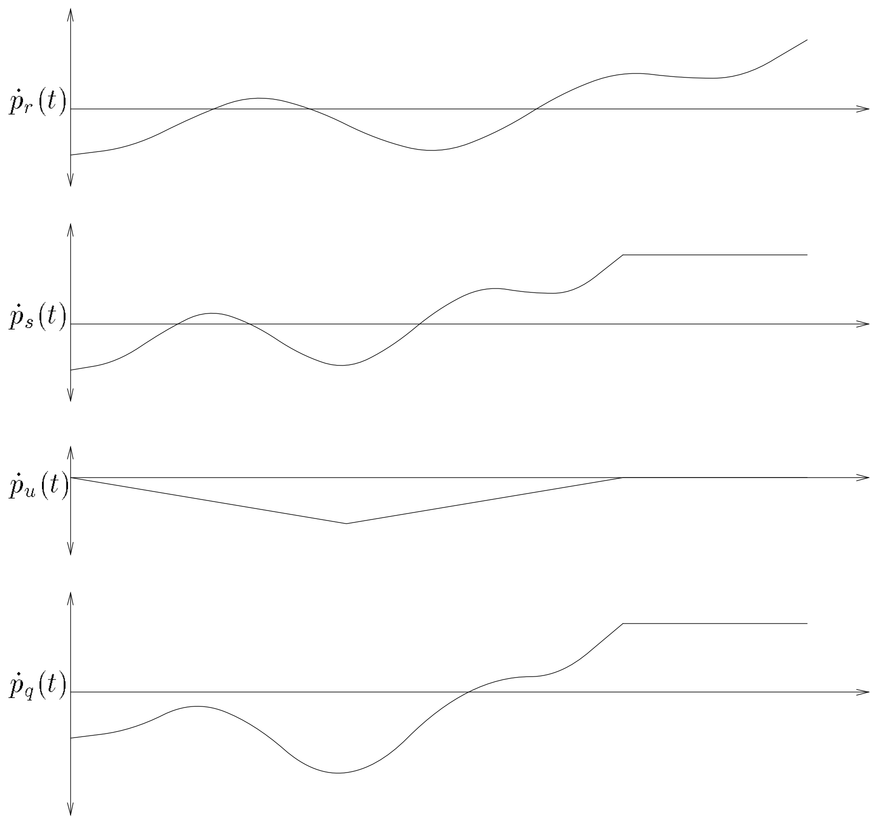

Figure 4: Velocity graphs for trajectories from lemma 4.3.

auxiliary trajectory Γ_u by setting the initial position to zero and letting

$$\dot{\mathbf{p}}_u(t) = \begin{cases} \mathbf{k}t & \text{for } 0 \leq t \leq \frac{cT}{2} \\ \mathbf{k}(cT - t) & \text{for } \frac{cT}{2} < t \leq cT \\ 0 & \text{for } cT < t \leq T \end{cases}$$

where $\mathbf{k}$ is the constant vector $\frac{4(1-c)}{(cT)^2}[\Delta\mathbf{p}_r(T) - \dot{\mathbf{p}}_r(T)T]$. In other words, Γ_u is a bang-bang trajectory, used for correction of Γ_s. We can bound $\|\mathbf{k}\|$:

$$\|\mathbf{k}\| = \frac{4(1-c)}{(cT)^2}[\|\Delta\mathbf{p}_r(T) - \dot{\mathbf{p}}_r(T)T\|] = \frac{4(1-c)}{(cT)^2}\|\int_0^T [\Delta\dot{\mathbf{p}}_r(t) - \Delta\dot{\mathbf{p}}_r(T)]dt\|$$

$$\leq \frac{4(1-c)}{(cT)^2}\int_0^T \|\Delta\dot{\mathbf{p}}_r(t) - \Delta\dot{\mathbf{p}}_r(T)\|dt$$

Since $\frac{\mathrm{d}\|\Delta\dot{\mathbf{p}}_r(t)-\Delta\dot{\mathbf{p}}_r(T)\|}{\mathrm{d}t} \leq \frac{a}{(1+\epsilon)^2}$, we can apply lemma 4.1 to get

$$\|\mathbf{k}\| \leq \frac{4(1-c)}{(cT)^2}\left[\frac{\|\Delta\dot{\mathbf{p}}_r(T)\|T}{2} - \frac{\|\Delta\dot{\mathbf{p}}_r(T)\|^2(1+\epsilon)^2}{4a} + \frac{aT^2}{4(1+\epsilon)^2}\right]$$

Maximizing the part in brackets (and noticing that $\|\Delta\dot{\mathbf{p}}_r(T)\| \leq \frac{aT}{(1+\epsilon)^2}$), we get

$$\|\mathbf{k}\| \leq \frac{2(1-c)a}{c^2(1+\epsilon)^2}.$$

Now we can define the trajectory Γ_q by $\dot{\mathbf{p}}_q(t) = \dot{\mathbf{p}}_s(t)+\dot{\mathbf{p}}_u(t)$, and $\mathbf{p}_q(0) = \mathbf{p}_r(0)$. Notice that by the above definitions, $\dot{\mathbf{p}}_q(0) = \dot{\mathbf{p}}_r(0)$ and $\dot{\mathbf{p}}_q(T) = \dot{\mathbf{p}}_r(T)$. To verify that the ending position of Γ_q is the same as the ending position of Γ_r, notice that $\mathbf{p}_u(T) = (1-c)\left[\Delta\mathbf{p}_r(T) - T\dot{\mathbf{p}}_r(T)\right]$, and adding this to $\mathbf{p}_s(T)$ shown above, the resulting simplified expression shows that indeed, $\mathbf{p}_q(T) = \mathbf{p}_r(T)$.

To calculate the acceleration bound of Γ_q, notice that

$$\|\ddot{\mathbf{p}}_q(t)\| = \|\ddot{\mathbf{p}}_s(t) + \ddot{\mathbf{p}}_u(t)\| \leq \|\ddot{\mathbf{p}}_s(t)\| + \|\ddot{\mathbf{p}}_u(t)\| \leq \|\ddot{\mathbf{p}}_s(t)\| + \|\mathbf{k}\|.$$

Using the previously calculated bound for $\|\mathbf{k}\|$ and noticing that $\|\ddot{\mathbf{p}}_s(t)\| = \frac{\|\ddot{\mathbf{p}}_r(t/c)\|}{c} \leq \frac{a}{c(1+\epsilon)^2}$, we see that $\|\ddot{\mathbf{p}}_q(t)\| \leq \frac{2-c}{c^2}\frac{a}{(1+\epsilon)^2}$. Substituting $c = \frac{\sqrt{9+8\epsilon}-1}{2(1+\epsilon)}$, we find that $\frac{2-c}{c^2} = 1+\epsilon$, so $\|\ddot{\mathbf{p}}_q(t)\| \leq \frac{a}{1+\epsilon}$. □

Now we examine how closely we can track a trajectory constructed as in lemma 4.3. First we consider tracking only the velocity; staying close to the desired velocity keeps the position within a tolerable error, and the last part of the interval (the last time interval of length $(1-c)T$ which is called the *adjustment interval*) is used to correct the position while causing no net change in velocity.

The first step is to divide the time T interval into a series of discrete intervals, each of length τ. For the current velocity, consider a set of choice vectors as described in corollary 3.1 with the unit distance being $a\tau$. Assuming that the approximation is within $a\tau$ of the desired velocity at the beginning of an interval, and since the desired trajectory obeys acceleration bound $\frac{a}{1+\epsilon}$, the exact velocity at the end of the interval will be no more than $(1+\frac{1}{1+\epsilon})a\tau$ away from the original approximation. Now using the result of corollary 3.1, we can pick a choice vector that results in a final approximation velocity within $a\tau$ of the desired velocity.

From the above argument, it should be obvious that if our approximation velocity initially starts within $a\tau$ of the desired velocity, then at every time step the approximation velocity can be kept within $a\tau$ of the desired velocity. This is what we mean by being able to closely track the velocity of the given trajectory; now we examine how much the position may be in error from blindly following only the velocity of the given trajectory.

First, a better estimate of how closely the velocity is tracked is needed. Theorem 3.1 says that at the times $i\tau$ (i an integer), the velocity of the approximating trajectory is within $a\tau$ of the velocity of the given trajectory, but what happens between these time instances? A maximizing argument (very similar to that used in the proof of lemma 4.1) shows that at *all* time instances the error is no more than $\frac{3}{2}a\tau$.

Letting Γ_e and Γ_a denote the exact and approximating trajectories, respectively, the error in position displacement can be bounded by

$$\|\int_0^T \dot{\mathbf{p}}_e(t)dt - \int_0^T \dot{\mathbf{p}}_a(t)dt\| = \|\int_0^T [\dot{\mathbf{p}}_e(t) - \dot{\mathbf{p}}_a(t)]\,dt\| \leq \int_0^T \|\dot{\mathbf{p}}_e(t) - \dot{\mathbf{p}}_a(t)\|dt$$

$$\leq \int_0^T \frac{3}{2}a\tau dt = \frac{3}{2}a\tau T$$

Since the time T interval is divided into length τ time segments, let N be the number of such segments (so $T = N\tau$); therefore, over the entire time T interval, the error in displacement is no more than $\frac{3}{2}aN\tau^2$.

Since the given trajectory we are tracking is a trajectory constructed as in lemma 4.3, the velocity does not change for the last $(1-c)T$ time in the time T interval (the approximating velocity as constructed above stays constant in this last time also), this last time can be used to correct the error in position with no net change to the velocity. To show how this is done more explicitly, a few preliminary lemmas are needed.

The next lemma is a purely combinatorial fact, but needs to be established to see how much error can be corrected in the adjustment interval.

Lemma 4.4 If M is an even integer ≥ 2, we define the sets

$$S_M = \{(a_1, a_2, ..., a_M) | a_k \in \{-1, 0, 1\} \text{ for } 1 \leq k \leq M, \text{ and } \sum_{k=1}^{M} a_k = 0\}$$

$$T_M = \{\sum_{k=1}^{M} k a_k | (a_1, a_2, ..., a_M) \in S_M\}.$$

Then the set T_M is simply $\{-\left(\frac{M}{2}\right)^2, -\left(\frac{M}{2}\right)^2+1, ..., -1, 0, 1, ..., \left(\frac{M}{2}\right)^2-1, \left(\frac{M}{2}\right)^2\}$.

Proof: The proof is by induction. For the base case, we will enumerate S_2 and T_2. It should be obvious that $S_2 = \{(1,-1),(0,0),(-1,1)\}$, and from this it is easy to construct $T_2 = \{-1,0,1\}$. This agrees with the lemma, and the base case has been proved.

For the induction, assume that the lemma holds for $M-2$, and we will prove that this implies that the lemma is true for M. We will construct a set $S'_M = \{(a_1, a_2, ..., a_M) | (a_1, a_M) \in S_2$, and $(a_2, a_3, ..., a_{M-1}) \in S_{M-2}\}$ and a set $T'_M = \{\sum_{k=1}^{M} k a_k | (a_1, a_2, ..., a_M) \in S'_M\}$. Clearly, $S'_M \subseteq S_M$ and $T'_M \subseteq T_M$.

We make the following observation: for all $(a_1, a_2, ..., a_M) \in S'_M$,

$$\sum_{k=1}^{M} k a_k = a_1 + M a_M + \sum_{k=2}^{M-1} k a_k = a_1 + M a_M + \sum_{k=1}^{M-2} (k+1) a_{k+1}$$

$$= a_1 + M a_M + \sum_{k=1}^{M-2} k a_{k+1} + \sum_{k=1}^{M-2} a_{k+1}$$

Since $(a_2, a_3, ..., a_{M-1}) \in S_{M-2}$, we know that $\sum_{k=1}^{M-2} a_{k+1} = 0$. Furthermore, since the image of $\sum_{k=1}^{M-2} k a_{k+1}$ over S'_M is T_{M-2} (by the induction hypothesis), we can use the definition of T'_M and this observation to see that

$$T'_M = \{M-1+e | e \in T_{M-2}\} \bigcup \{e | e \in T_{M-2}\} \bigcup \{1-M+e | e \in T_{M-2}\}$$

It is easy to see that if $M - 1 + \min\{T_{M-2}\} \leq \max\{T_{M-2}\} + 1$, then $\{n | n \in T'_M$ and $n \geq 0\} = \{0, 1, ..., M-1+\max\{T_{M-2}\}\}$. Using the inductive hypothesis for $\min\{T_{M-2}\}$ and $\max\{T_{M-2}\}$, we see that this is indeed true for all $M \geq 2$. A similar argument holds for the negative half of T'_M. Noticing that

$$M - 1 + \max\{T_{M-2}\} = M - 1 + \left(\frac{M-2}{2}\right)^2 = M - 1 + \left(\frac{M}{2}\right)^2 - M + 1 = \left(\frac{M}{2}\right)^2,$$

we see that $T'_M = \{-\left(\frac{M}{2}\right)^2, -\left(\frac{M}{2}\right)^2+1, ..., -1, 0, 1, ..., \left(\frac{M}{2}\right)^2-1, \left(\frac{M}{2}\right)^2\} \subseteq T_M$.

To see that the inclusion also goes the other way, observe that the maximum value of T_M occurs when $a_1 = a_2 = \cdots = a_{M/2} = -1$ and $a_{M/2+1} = a_{M/2+2} = \cdots = a_M = 1$, so $\max\{T_M\} = \left(\frac{M}{2}\right)^2$. Similarly, it can be shown that $\min\{T_M\} = -\left(\frac{M}{2}\right)^2$. The proof of the lemma is now complete. $\square$

This lemma easily applies to give a result about the adjustment interval.

Lemma 4.5 Let M be a positive multiple of $2d$ (d a positive integer), and let $\mathbf{e}$ be any d-dimensional vector with $\|\mathbf{e}\| \leq \left(\frac{M}{2d}\right)^2$. Define the set

$$A = \{(a_1, a_2, ..., a_d) | a_i \in \{-1, 0, 1\} \text{ for some } 1 \leq i \leq d, \text{ and } a_j = 0 \text{ for all } j \neq i\},$$

so $\|\mathbf{a}\| = 1$ or $\|\mathbf{a}\| = 0$ for all $\mathbf{a} \in A$. Then there exists a sequence $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_M$ where each $\mathbf{v}_i \in A$ such that $\sum_{k=1}^{M} \mathbf{v}_k = \mathbf{0}$ and $\sum_{k=1}^{M} k\mathbf{v}_k = \mathbf{m}$, where $\|\mathbf{e} - \mathbf{m}\| \leq \frac{\sqrt{d}}{2}$.

PROOF: Let $\mathbf{e} = (e_1, e_2, ..., e_d)$ and define

$$A_1 = \{(a_1, a_2, ..., a_d) | a_1 \in \{-1, 0, 1\}, \text{ and } a_i = 0 \text{ for all } 2 \leq i \leq d\},$$

and

$$S_1 = \{(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{M/d}) | \mathbf{v}_i \in A_1 \text{ for } i = 1, 2, ..., \frac{M}{d} \text{ and } \sum_{k=1}^{M/d} \mathbf{v}_k = \mathbf{0}\}.$$

For any real number r with $|r| \leq \left(\frac{M}{2d}\right)^2$, we can pick an integer m_1 such that $|r - m_1| \leq \frac{1}{2}$ and $-\left(\frac{M}{2d}\right)^2 \leq m_1 \leq \left(\frac{M}{2d}\right)^2$. By lemma 4.4, there exists a sequence $(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{M/d}) \in S_1$ such that $\sum_{k=0}^{M/d} k\mathbf{v}_k = (m_1, 0, ..., 0)$ — there are $d - 1$ zeros following m_1.

Since $|e_i| \leq \left(\frac{M}{2d}\right)^2$ for $i = 1, 2, ..., d$, this error correction can be repeated for each dimension, so there exists a sequence of M vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_M$ from A such that $\sum_{k=1}^{M} \mathbf{v}_k = \mathbf{0}$ and $\sum_{k=1}^{M} k\mathbf{v}_k = (m_1, m_2, ..., m_d) = \mathbf{m}$, where $|m_i - e_i| \leq \frac{1}{2}$ for $i = 1, 2, ..., d$. Therefore,

$$\|\mathbf{m} - \mathbf{e}\| \leq \sqrt{(m_1 - e_1)^2 + (m_2 - e_2)^2 + \cdots + (m_d - e_d)^2} \leq \sqrt{d\left(\frac{1}{2}\right)^2} \leq \frac{\sqrt{d}}{2}.$$

$\square$

We use a sequence of choice vectors constructed as in lemma 4.5 to correct the position during the adjustment interval. As in corollary 3.1 we use $a\tau$ as one "unit length", and set $M = (1-c)N$. From lemma 4.3 and lemma 4.5 it can be seen that making adjustments during the adjustment interval as in lemma 4.5 keeps the final velocity the same, and the first two terms of equation (2) remain the same, but the last term can be adjusted by $\pm\left(\frac{(1-c)N}{2d}\right)^2 a\tau^2$. Thus as long as this possible adjustment is greater than the possible error, we can adjust the final position to within $\frac{\sqrt{d}}{2}a\tau^2$ of the exact trajectory, while the final velocity is within $a\tau$ of the exact trajectory.[1] This is summed up in the following theorem.

Theorem 4.1 If we set $N = \left\lceil \frac{6d^2}{(1-c)^2} \right\rceil$ (where c is from lemma 4.3) and $\tau = \frac{T}{N}$, then given any time T trajectory Γ_e that meets acceleration bound $\frac{a}{(1+\epsilon)^2}$, there is a trajectory Γ_a that uses only the velocity choice vectors (meeting acceleration bound a) with

$$\|\mathbf{p}_e(T) - \mathbf{p}_a(T)\| \le \frac{\sqrt{d}}{2}a\tau^2$$

$$\|\dot{\mathbf{p}}_e(T) - \dot{\mathbf{p}}_a(T)\| \le a\tau$$

Furthermore, we then have $N = O(d^2\left(\frac{1}{\epsilon}\right)^2)$.

Proof: By lemma 4.3, we can construct a trajectory Γ_s with the same endpoints as Γ_e, takes time T, meets acceleration bound $\frac{a}{1+\epsilon}$, and has constant velocity on the interval $[cT, T]$. As was remarked following the proof of lemma 4.3, trajectory Γ_s can be tracked on our grid (producing a grid trajectory Γ_t) such that the grid trajectory also takes time T, meets acceleration bound a, and has constant velocity on $[cT, T]$. Furthermore, it was shown that the error of this approximation can be bounded as

$$\|\mathbf{p}_e(T) - \mathbf{p}_t(T)\| \le \frac{3}{2}Na\tau^2.$$

$$\|\dot{\mathbf{p}}_e(T) - \dot{\mathbf{p}}_t(T)\| \le a\tau$$

The interval $[cT, T]$ is used to remove the error from the position (with no net change in velocity) — the relationship between lemma 4.5 and the displacement of a grid trajectory is

[1] We have implicitly assumed that positive and negative unit length choice vectors for each coordinate axis exist in our set of choice vectors. This assumption is not too great, as adding these vectors only increases the size of our set of choice vectors by $2d$. Furthermore, these vectors obviously exist on our superimposed square grid.

obvious from equation (2). By lemma 4.5, the error of at most $\frac{3}{2}Na\tau^2$ can be reduced to $\frac{\sqrt{d}}{2}a\tau^2$ in $(1-c)N$ steps as long as this error is less than the possible adjustment: $\left[\frac{(1-c)N}{2d}\right]^2 a\tau^2$. In other words, the error bounds in the theorem are met if

$$\frac{3}{2}N \leq \left[\frac{(1-c)N}{2d}\right]^2.$$

This condition is met for all $N \geq \frac{6d^2}{(1-c)^2}$, so in particular is met for $N = \left\lceil \frac{6d^2}{(1-c)^2} \right\rceil$, and the error bounds have been proved.

Due to the odd form of c, the asymptotic growth of N is not clear. Consider $\frac{1}{1-c}$; by definition this is simply (for $\epsilon \leq 1$)

$$\frac{1}{1-c} = \frac{2(1+\epsilon)}{3+2\epsilon-\sqrt{9+8\epsilon}} \leq \frac{4}{3+2\epsilon-\sqrt{9+8\epsilon}}.$$

The growth rate (as $\frac{1}{\epsilon} \to \infty$) can be compared with that of $\frac{1}{\epsilon}$ by taking the limit of the ratio

$$\lim_{\frac{1}{\epsilon}\to\infty} \frac{\frac{1}{3+2\epsilon-\sqrt{9+8\epsilon}}}{\frac{1}{\epsilon}} = \lim_{\epsilon\to 0} \frac{\epsilon}{3+2\epsilon-\sqrt{9+8\epsilon}}.$$

The numerator and denominator of this limit both go to 0, so using L'Hôpital's rule, the limit is equal to

$$\lim_{\epsilon\to 0} \frac{1}{2-4(9+8\epsilon)^{-1/2}} = \frac{1}{2-\frac{4}{3}} = \frac{3}{2}.$$

In other words, $\frac{1}{1-c} = \Theta(\frac{1}{\epsilon})$. It follows that

$$N = \left\lceil \frac{6d^2}{(1-c)^2} \right\rceil = O(d^2\left(\frac{1}{\epsilon}\right)^2).$$

□

Now we turn attention to tracking within a certain tolerance. By tracking within tolerance (η_x, η_v), we mean that given an exact trajectory Γ_e and an approximating trajectory Γ_a, at all times t, both of the following inequalities hold.

$$\|\mathbf{p}_e(t) - \mathbf{p}_a(t)\| \leq \eta_x \tag{5}$$

$$\|\dot{\mathbf{p}}_e(t) - \dot{\mathbf{p}}_a(t)\| \leq \eta_v \tag{6}$$

The way we satisfy this is to divide the entire trajectory into a number of intervals, each of which meet the endpoint conditions of theorem 4.1. By making the length of such intervals sufficiently small, we can insure that equations (5) and (6) are satisfied.

For any two time T trajectories Γ_e and Γ_a satisfying the endpoint constraints of theorem 4.1, it is easy to see that the approximating velocity can never be farther than $aT + a\tau = a\tau(N+1)$ from the exact velocity; therefore, to satisfy condition (6) we only need to insure that $a\tau(N+1) \leq \eta_v$, or $\tau \leq \frac{\eta_v}{a(N+1)}$.

Guaranteeing that the position tolerance is obtained is also easy. An easy proof using lemma 4.1 shows that at all times the position can never be farther off than $\frac{(N(N+2)+\sqrt{d})a\tau^2}{2}$, so to satisfy condition (5) we need to insure that $\tau^2 \leq \frac{2\eta_x}{a(N(N+2)+\sqrt{d})}$. Both tolerance conditions can be satisfied if

$$\tau \leq \min\left(\sqrt{\frac{2\eta_x}{a(N(N+2)+\sqrt{d})}}, \frac{\eta_v}{a(N+1)}\right). \tag{7}$$

Using the bound for N and noting that we want to control the growth of $\frac{1}{\tau}$, it is interesting to note that the above formula guarantees that we can track within tolerance (η_x, η_v) with $\frac{1}{\tau} = O(\frac{ad^2}{\epsilon^2}\max(\sqrt{\frac{1}{\eta_x}}, \frac{1}{\eta_v}))$ (in other words, polynomial in a, $\frac{1}{\epsilon}$, d, $\frac{1}{\eta_x}$, and $\frac{1}{\eta_v}$).

The above discussion can be summed up in the following tracking theorem.

Theorem 4.2 Given any time T trajectory Γ_e from $(\mathbf{p}_e(0), \mathbf{0})$ to $(\mathbf{p}_e(T), \mathbf{0})$ that meets acceleration bound a, there exists a time $(1+\epsilon)T$ trajectory Γ_a on a grid constructed as described in corollary 3.1 that also meets acceleration bound a and satisfies

$$\|\mathbf{p}_e(t) - \mathbf{p}_a((1+\epsilon)t)\| \leq \eta_x$$

$$\|\dot{\mathbf{p}}_e(t) - \dot{\mathbf{p}}_a((1+\epsilon)t)\| \leq \eta_v,$$

for any given tolerance (η_x, η_v). Furthermore, the time spacing τ of the grid can be made to meet

$$\frac{1}{\tau} = O\left(ad^2\left(\frac{1}{\epsilon}\right)^2 \max\left(\sqrt{\frac{1}{\eta_x}}, \frac{1}{\eta_v}\right)\right).$$

Proof: Consider the trajectory Γ_e slowed down as by lemma 4.2. This new trajectory joins the same endpoints, takes time $(1+\epsilon)T$, and meets acceleration bound $\frac{a}{(1+\epsilon)^2}$. From the given ϵ and the number of dimensions d, we can calculate N as in theorem 4.1 and τ as in equation (7). Now consider the time required by the slowed down trajectory to be divided into segments, each of the form $[iN\tau, (i+1)N\tau]$. Each segment meets all of the requirements to be tracked as described in the text preceding this theorem, so the result is exactly as stated in the theorem. $\square$

5 Tracking with Obstacles

As stated in the introduction, we are actually interested in finding paths that avoid a given set of obstacles. The concepts of "safe" and "also-safe" trajectories reflect the real-world physical property that robots cannot navigate accurately at high speeds; the terms were introduced in section 1, and are restated here in a more formal setting.

Definition 5.1 Let $\delta(c_1, c_0) : \mathbf{R} \rightarrow \mathbf{R}$ be an affine function that maps real numbers to real numbers by $\delta(c_1, c_0)(x) = c_1 x + c_0$ (it will map velocity magnitudes to distance magnitudes); when there is no ambiguity about the values of c_1 and c_0 or the particular values are unimportant, this function is written as simply δ. A trajectory Γ_r is considered $\delta(c_1, c_0)$-*safe* (or just *safe*) if at all times t during the trajectory, the norm of the distance vector to any object is at least $\delta(||\dot{\mathbf{p}}_r(t)||)$. An approximating trajectory Γ_q (approximating with accuracy ϵ) is called "also-safe" if at all times t during the trajectory, the norm of the distance vector to any object is at least $(1-\epsilon)\delta(||\dot{\mathbf{p}}_q(t)||)$.

The notion of safe and also-safe trajectories comes from [1], and a more general version of the following theorem can be found in their paper (as lemma 3.3). Note that in the following proof, the only property of the norm that we use is the triangle inequality, so the theorem is true for *all* norms, not just the L_2 norm.

Theorem 5.1 Let $\delta(c_1, c_0)$ be a safety function as described in definition 5.1. A trajectory Γ_a (found as described in theorem 4.2) that tracks a safe exact trajectory Γ_e with tolerances

$$\eta_x = \eta_v = \frac{\epsilon c_0}{(1-\epsilon)c_1 + 1}$$

will be also-safe.

Proof: For any time t, we define the "safe ball" about Γ_e to be the set of points within distance $\delta(\dot{\mathbf{p}}_e(t))$ of the point $\mathbf{p}_e(t)$. Similarly, the "also-safe ball" about Γ_a at time $(1+\epsilon)t$ is the set of points within distance $(1-\epsilon)\delta(\dot{\mathbf{p}}_a((1+\epsilon)t))$ of the point $\mathbf{p}_a((1+\epsilon)t)$. It is only necessary to show that the also-safe ball around Γ_a lies entirely within the safe ball about Γ_e at all times. After showing this, it is clear that the also-safe ball around Γ_a is free of obstacles (since the safe ball around Γ_e is free of obstacles); in other words, Γ_a is also-safe.

To show that the also-safe ball for Γ_a lies within the safe ball for Γ_e, consider any point $\mathbf{q}$ in the also-safe ball about $\mathbf{p}_a((1+\epsilon)t)$ — we wish to prove that $\mathbf{q}$ lies within the safe ball about $\mathbf{p}_e(t)$, which is true if and only if $\|\mathbf{q}-\mathbf{p}_e(t)\| \leq \delta(\|\dot{\mathbf{p}}_e(t)\|)$. Of course,

$$\|\mathbf{q}-\mathbf{p}_e(t)\| \leq \|\mathbf{q}-\mathbf{p}_a((1+\epsilon)t)\| + \|\mathbf{p}_a((1+\epsilon)t)-\mathbf{p}_e(t)\|. \tag{8}$$

We can bound the first term on the right hand side by using the fact that $\mathbf{q}$ is within the also-safe ball of $\mathbf{p}_a((1+\epsilon)t)$ (so $\|\mathbf{q}-\mathbf{p}_a((1+\epsilon)t)\| \leq (1-\epsilon)\delta(\|\dot{\mathbf{p}}_a((1+\epsilon)t)\|)$), and then write this in terms of $\dot{\mathbf{p}}_e(t)$ and η_v. The final result is that

$$\|\mathbf{q}-\mathbf{p}_a((1+\epsilon)t)\| \leq (1-\epsilon)\delta(\|\dot{\mathbf{p}}_e(t)\|+\eta_v).$$

The second term on the right hand side of equation (8) is easily upper bounded by η_x (by the very definition of η_x), so

$$\|\mathbf{q}-\mathbf{p}_e(t)\| \leq (1-\epsilon)\delta(\|\dot{\mathbf{p}}_e(t)\|+\eta_v)+\eta_x$$

Substituting the values of η_x and η_v found in the statement of the theorem, it is easily shown that

$$(1-\epsilon)\delta(\|\dot{\mathbf{p}}_e(t)\|+\eta_v)+\eta_x \leq \delta(\|\dot{\mathbf{p}}_e(t)\|),$$

so $\mathbf{q}$ must lie in the safe ball around $\mathbf{p}_e(t)$. Since this is true for all points $\mathbf{q}$ in the also-safe ball of Γ_a, the also-safe ball of Γ_a must lie entirely within the safe ball of Γ_e. □

Combining this with the other results gives the following corollary (our main result).

Corollary 5.1 Given acceleration bounds a, obstacles $\mathcal{E}$, and positive reals $\epsilon \leq 1$, c_0, and c_1, for any $\delta(c_1, c_0)$-safe trajectory taking time T, there exists a time spacing τ with

$$\frac{1}{\tau} = O\left(\frac{c_1}{c_0}ad^2\left(\frac{1}{\epsilon}\right)^3\right),$$

a grid constructed from choice vectors (as described in section 3), and a $(1-\epsilon)\delta$-safe approximating trajectory Γ_a between grid-points that takes time at most $(1+\epsilon)T$. Furthermore, this results in an approximation algorithm that is fully polynomial in the combinatorial and algebraic complexity of the environment, and pseudopolynomial in the kinodynamic bounds.

PROOF: The existence proof of the $(1-\epsilon)\delta$-safe approximating trajectory follows from the results and discussion above. From the derivation of the bound on τ, it follows that a rational grid size can be chosen where the grid length can be represented with a number of bits that is polynomial in the lengths of the input parameters. It follows that the results of the other simple intermediate calculations will also have polynomially many bits. As the grid is searched, it is reasonably simple to check if the current state (a point on the grid) violates safety margins with the obstacles — simply find the closest obstacle boundary point to the point being tested, then check to see if that distance violates the safety function at the current velocity (the state gives the velocity at the point). Verifying that safety constraints are not violated *between* grid-points is a simple extension [1]. This operation is fully polynomial in the geometric complexity of the obstacles $\mathcal{E}$.

The size of the search space is exactly the number of possible states. Considering how fast the grid of section 3 grows, it is clear that the number of possible velocity vectors in the search space is bounded by $\left(\frac{4v_{\max}}{\epsilon a_{\max}\tau}\right)^d$. From the diameter D of the space and equation (2), it should be clear that the number of possible positions is bounded by $\left(\frac{4D}{\epsilon a_{\max}\tau^2}\right)^d$. Combining these quantities, the number of states is $O\left(\left[\frac{v_{\max}D}{\epsilon^2 a_{\max}^2\tau^3}\right]^d\right)$; in other words, since $\frac{1}{\tau}$ is polynomial in the dynamics bounds, the total number of grid-points is polynomial in the dynamics bounds (but not in their *lengths* — hence the search algorithm is only pseudopolynomial).

Since the grid size is polynomial in the kinodynamic bounds, and the complexity of checking the validity of each grid-point is polynomial in the geometric complexity, the complexity results claimed in the theorem are verified. □

6 Conclusion

We have shown that while the (exact) optimal kinodynamic planning problem may be computationally difficult, it is possible to approximate the optimal path with our simple algorithm — simply construct a grid as explained in section 3 and perform a search on this grid to find a path from the start state to the goal state. The main result of this paper is that if the grid is constructed within certain parameters (see corollary 5.1, equation (7), etc.), then for any safe optimal path there exists an also-safe grid path that is within a $(1+\epsilon)$ factor of optimal. The size of the grid is polynomial in the input size, in $\frac{1}{\epsilon}$, and in the dynamics bounds, so the result is a polynomial approximation algorithm for kinodynamic planning (where dynamics bounds are expressed in terms of maximum 2-norm for acceleration).

References

[1] J. Canny, B. Donald, J. Reif, and P. Xavier. "On The Complexity of Kinodynamic Planning," *29th FOCS*, 1988, pp. 306-316.

[2] J. Canny, A. Rege, and J. Reif. "An Exact Algorithm for Kinodynamic Planning in the Plane," Tech. Report.

[3] J. Canny, and J. Reif. "New Lower Bound Techniques for Robot Motion Planning," *28th FOCS*, 1987, pp. 49-60.

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

[5] B. Donald and P. Xavier. "Near-Optimal Kinodynamic Planning for Robots With Coupled Dynamics Bounds," *IEEE Int. Symp. on Intelligent Controls*, 1989.

[6] S. Fortune and G. Wilfong. "Planning Constrained Motion," *20th STOC*, 1988, pp. 445-457.

[7] P. Jacobs and J. Canny. "Planning Smooth Paths for Mobile Robots," *IEEE Int. Conf. on Robotics and Automation*, 1989.

[8] C. Ó'Dúnlaing, "Motion Planning with Inertial Constraints," *Algorithmica*, Vol. 2(4), 1987, pp. 431-475.

[9] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, NY, 1982.

[10] C. Papadimitriou. "An Algorithm for Shortest Path Motion in Three Dimensions," *Info. Proc. Letters*, Vol. 20, 1985, pp. 259-263.

[11] G. Sahar and J. Hollerbach. "Planning of Minimum-Time Trajectories for Robot Arms," *IEEE Int. Conf. on Robotics and Automation*, 1985.

[12] J. T. Schwartz and M. Sharir. "On the Piano Movers' Problem: I. The Case of a Rigid Polygonal Body Moving Amidst Polygonal Barriers," *Comm. Pure and Appl. Math.*, Vol. 36, 1983, pp. 345-398.

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