Time-Optimal Trajectories for a Robot Manipulator: A Provably Good Approximation Algorithm

Greg Heinzinger* UC Berkeley

Paul Jacobs† LAAS-CNRS

John Canny‡ UC Berkeley

Brad Paden§ UC Santa Barbara

Abstract

In this paper we present an algorithm for generating near-time-optimal trajectories for an open-kinematic-chain manipulator moving in a cluttered workspace. This is the first algorithm to guarantee bounds on the closeness of an approximation to a time-optimal trajectory. The running time and space required are polynomial in the desired accuracy of the approximation. The user may also specify tolerances by which the trajectories must clear obstacles in the workspace to allow modeling of control errors.

Notation

$\mathbf{R}$	Real numbers.
$\mathbf{Z}$	Integers.
$\lVert\cdot\rVert$	Euclidean norm (2-norm).
ϵ	The slow down factor.
m	Number of manipulator joints.
T	The time discretization.
$\Delta\ddot{\theta}$	The joint acceleration discretization.
N	The number of multi-steps.
Θ_0	The initial configuration.
Θ_f	The final configuration.
$\delta\ddot{\Theta}$	The minimum spacing in acceleration associated with $\tau_{\max}$ and $\tilde{\tau}_{\max}$.
$\tau_{\max}$	The maximum allowable torque for the manipulator.
$\hat{\tau}_{\max}$	The maximum allowable torque for a constrained trajectory.
$\tilde{\tau}_{\max}$	The maximum allowable torque for an unconstrained trajectory.
γ	$\tau_{\max} - \tilde{\tau}_{\max}$.
$G_{\max}$	The maximum joint torques due to gravity.
$D(\Theta, \dot{\Theta})$	The safety margin.
$D_\epsilon(\Theta, \dot{\Theta})$	The tracking error.
L	The vector of joint limits.

*Research supported by by the Berkeley Fellowship and by the David and Lucile Packard Foundation. This author is with the Department of Electrical Engineering and Computer Science and The Electronics Research Laboratory, University of California at Berkeley, Berkeley, CA 94720

†Research supported by the David and Lucile Packard Foundation; by the Defense Advanced Research Projects Agency (DoD), monitored by Space and Naval Warfare Systems Command under Contract N00039-88-C-0292. This author is now at the Laboratoire d'Automatique et d'Analyse des Systems, 7, Avenue du Colonel Roche, 31077 Toulouse Cedex, France. Research supported by Groupement Robots d'Intervention sur Site Planetaire (RISP).

‡Research supported by the David and Lucile Packard Foundation and by PYI grant number IRL-8958577. This author is with the Dept. of Computer Science at the University of California at Berkeley.

§Research supported by the National Science Foundation through Grant no. 8421415. This author is with the Department of Mechanical Engineering and Center for Robotic Systems in Microelectronics, University of California at Santa Barbara, Santa Barbara, CA 93106

1 Introduction

While the piano movers problem has served as a paradigm for motion planning almost from its inception, it has become increasingly clear that the framework is not general enough to model many systems of interest. What is lacking is the ability to impose other restrictions on allowable trajectories besides simply avoiding collisions with obstacles in the environment. Although we can also treat mechanical systems subject to kinematic constraints by extending the formalism to include the notion of configuration space, beyond this point, the essentially geometric formulation breaks down. Once we introduce constraints which are not just based upon geometry, we are forced to deal with the dynamic model of the manipulator, as well as its kinematic model. In this paper, our emphasis is on such a system: we plan paths for a robot manipulator with the addition of bounds on its actuator torques.

Once we include dynamics, we operate under the paradigm of *trajectory planning*, in which we take into account the time history of the curve. We can turn the bounds on the actuator torques into constraints upon allowable trajectories using the dynamic model of the manipulator. What makes the problem difficult is that the allowed accelerations are dependent upon the current state of the manipulator, which is constantly changing.

As we've noted, previous planners were able to accommodate kinematic constraints by searching for a path in configuration space. Because these spaces have natural geometric constructions, these algorithms have tended to use geometric measures of the quality of a path, such as path length. Unfortunately, with a complex system such as a robot manipulator, it is not often true that the minimum arc-length path takes the least amount of time. Because we have adopted a framework which includes a notion of time, we can search for trajectories where the goal is not only to avoid obstacles while traveling between points in the environment but also to utilize the capabilities of the robot to the fullest. We search for motions which are nearly time-optimal given the dynamic constraints imposed by the limitations on the manipulator's actuators.

It seems clear that, in order to be practical, motion planning must take into account the dynamics of the system. But it is not necessary to abandon the ideas of path planning which have been developed so far and adopt only the techniques of control theory, although there is necessarily some relationship between the approaches. Here we present an approximation algorithm for planning time optimal paths based upon the idea of graph search using a discretization of time and joint accelerations which is fine enough to allow provably good approximation. It is in proving the quality of the approximation that we have used ideas from both domains.

The algorithm presented in this paper is the first provably good approximation algorithm solving the problem of finding a time-optimal trajectory between two configurations for an open-kinematic-chain manipulator. Suppose this time-optimal trajectory takes time t_f^*, then the algorithm will generate a trajectory taking time less than $(1+\epsilon)t_f^*$ for any $\epsilon > 0$ supplied by the user. We take the view that finding provably good approximate solutions is as useful, in a practical sense, as finding exact solutions and in some cases, it is provably more tractable.

1.1 Statement of the Problem

We consider the following specific problem in this work. We wish to determine a collision-free trajectory for a m-jointed open-kinematic-chain manipulator with prismatic and revolute joints moving from an initial configuration to a final configuration through a cluttered workspace. We also require safety margins be supplied which specify the clearance that must be maintained between the manipulator and obstacles in its workspace.[1] Finally, the user specifies the maximum ratio between the time needed to traverse the trajectory generated by the algorithm and a time-optimal safe trajectory.

We assume that the motors driving the joints are able to generate torques up to a specified bound, where the bounds are independent of the manipulator configuration and joint velocities. We also assume the joint velocities of the manipulator are bounded above.[2]

The general problem of finding a time-optimal safe trajectory can be stated more precisely as follows. Given an open-kinematic-chain manipulator with prismatic and revolute joints, consider the class of trajectories in joint space $\Theta(t)$, which satisfy the following constraints.

1. $\Theta(0) = \Theta_0$, $\dot{\Theta}(0) = 0$
2. $\exists t_f > 0$, such that $\Theta(t_f) = \Theta_f$, $\dot{\Theta}(t_f) = 0$
3. $\forall t \in [0, t_f]$, the minimum distance between the manipulator in configuration $\Theta(t)$ and the workspace obstacles is greater than the user-specified function $D(\Theta, \dot{\Theta})$ which is bounded away from 0.[3]
4. $\forall t \in [0, t_f]$, the torques required to execute $\Theta(t)$ are upper-bounded by $\tau_{i,\max}$ for each joint i.
5. $\forall t \in [0, t_f]$, $|\dot{\theta}_i(t)| \leq \dot{\theta}_{i,\max}$ for each joint i.

The problem is to find a trajectory $\Theta^*(t)$ defined for $t \in [0, t_f^*]$, so that t_f^* is minimal over all trajectories in this class. We call $\Theta^*(t)$ a **time-optimal safe trajectory**.

In order to approximate a time-optimal safe trajectory $\Theta^*(t)$, we introduce some user-specified parameters which determine the closeness of the approximation.

1. User specifies $\epsilon > 0$ so that the time $\hat{t}_f$ to traverse the trajectory approximating $\Theta^*(t)$ satisfies $\hat{t}_f \leq (1+\epsilon)t_f^*$. ϵ is called the **slowdown factor**.
2. User specifies $D_\epsilon(\Theta, \dot{\Theta})$, with D_ϵ bounded away from zero and $D_\epsilon < D$, so that $\forall t \in [0, \hat{t}_f]$ the minimum distance between the robot and the workspace obstacles is greater than $D - D_\epsilon$.[4]

Extensions to this work, in which we relax some of these assumptions, are discussed in Section 5.

[1] A trajectory which respects some safety margins is known as a **safe trajectory**[2].

[2] In practice this restriction may be due to factors such as friction in the joints and limitations on the range of motion of the joints.

[3] A specific example of such a function is $D(\Theta, \dot{\Theta}) = c_0 + c_1\|\dot{\Theta}\|$ as in [2].

[4] Because the form of the functions D and D_ϵ are not crucial to the performance of the algorithm, these functions can be specified in workspace, joint space, or a combination of the two. The important properties are that D and D_ϵ are bounded away from zero and $D > D_\epsilon$.

1.2 Statement of the Results

We have developed the first algorithm that generates a trajectory between two configurations of an open-kinematic-chain manipulator that is a guaranteed approximation to a time-optimal safe trajectory which uses the full robot dynamics and is subject to torque constraints on the actuators. We prove that the trajectory generated by this algorithm lags behind an optimal trajectory by a specified time factor under reasonable conditions. Naturally, there is a trade-off between the computation speed of the algorithm and the closeness of the approximation.

The algorithm is based on the construction of a lattice in the state space[5] of the manipulator and discretizes time and allowable joint accelerations. It involves only a breadth-first search of a dynamically generated search graph to determine the appropriate trajectory. The running time depends on the closeness of the approximation. It is polynomial in $1/\epsilon$ and the complexity of the environment. It should be emphasized that our algorithm will find an approximation requiring only slightly more time than a globally time-optimal-safe trajectory[6], however, it is not guaranteed to be close to a globally time-optimal-safe trajectory in state space.

The resultant trajectory is returned as a sequence of joint accelerations each commanded for a fixed period of time. Joint accelerations are used to specify the trajectory, but we have insured that the trajectory respects the torque bounds.

The majority of the paper is spent determining the discretizations and proving that the guarantee of near-time-optimality can be made. Among other techniques, the proof relies on the time-scaling property of the manipulator dynamics [3], configuration-independent bounds on the manipulator dynamics [4], and multi-dimensional tracking procedures [1].

1.3 Previous Results

The problem of motion planning for manipulators has attracted a great deal of attention from researchers in the robotics community. The reader is directed to [5, 6, 7] for a general review of robot motion planning.

There has been some work which directly addresses the question of finding time-optimal trajectories for manipulators. The biggest drawback of the previous schemes is that they give no estimate of the quality of the approximation. The algorithm we present here generates a trajectory for an open-kinematic-chain manipulator which is *guaranteed* to be an arbitrarily close approximation to a time-optimal safe trajectory.

The approach which we have taken was initially motivated by work done by Canny, Donald, Reif, and Xavier [2]. Their work discussed finding a near-time-optimal safe trajectory for a moving particle in the plane which is subject to uniform L_∞ acceleration bounds on each axis. We have adapted some techniques from the proof architecture used in this paper. They 1) use the concept of safe trajectories; 2) show that a trajectory respecting constraints on acceleration can keep up with an unconstrained trajectory; and 3) reduce the problem to a graph search. Further work on the particle problem yielding an improved running time and simpler proof has been presented[8]. The formulation in [2] also relates to finding near-time-optimal safe trajectories for a cartesian robot, whose dynamics are neither state-dependent nor coupled,[7] as in the more general case which we consider here.

[5] The state space of the manipulator is the set of joint angles and velocities $(\Theta, \dot{\Theta})$.

[6] This is in contrast to optimization techniques which may yield a result which is locally minimal.

[7] Because the dynamics of a cartesian robot are decoupled and uniform, force bounds directly translate into uniform acceleration bounds.

The earliest examples of trajectory planning for manipulators are due to Kahn and Roth [9]. Their technique was based upon linearization of the system dynamics, resulting in a bang-bang optimal solution. The problem remained open since Hollerbach [3] showed that the non-linear terms in the robot dynamics could not be neglected when planning time-optimal trajectories.

Later researchers who have addressed the specific problem of time-optimal planning can be grouped into categories according to their method of attack. One focus of research has been to determine the time-optimal control for a manipulator moving along a given path. Bobrow, Dubowsky and Gibson [10] and Shin and McKay [11] have independently developed methods which determine the optimal controls. Bobrow [12] has utilized these techniques along with optimization routines to find the minimum time trajectory among a class of paths. However, none of these researchers give bounds on the closeness of their approximations. Furthermore, optimization algorithms must be given an initial path, and no discussion is presented as to the effect which this choice can have on the approximation generated. Paden, Mees, and Fisher [13] have also made use of optimization techniques. Their paper works in essentially the same manner as [12] though they include a method of automatically selecting an initial path.

Hollerbach [3] has shown that torque requirements are decreased if a given trajectory is time-scaled. Along with Sahar [14], he used this fact as the basis of a grid-based search for a time-optimal trajectory. No bounds are given in this work on the sub-optimality of the trajectory found. Furthermore, no proof is given that the torque bounds are satisfied over the entire trajectory chosen. The time-scaling property of the dynamics is an important fact, and is crucial to the proof of our algorithm. Sontag and Sussman [15] present facts about the exact solution to the problem of time-optimal planning. While no algorithm is given, this paper gives an indication of the complexity of the exact approach.

Other work that is crucial to the proofs we present is the recent development of bounds on the dynamics of open-kinematic-chain manipulators by Heinzinger and Paden [4]. We use these to derive quantitative bounds on the change in the manipulator dynamics in a neighborhood of a given state.

The method of finding *primitive trajectories* which are concatenated to form more complex objects was first proposed by Paul [16] for following cartesian space paths with a robot manipulator. The concept of *path tolerance* is introduced to cope with the approximations inherent in using these primitives. Binford [17], as far back as 1971, and Finkel [18] present the idea that a planned trajectory should lie in a "tolerance region" whose center is the ideal trajectory.

Fundamental work in trajectory planning with dynamic constraints is presented in the paper by Ó'Dúnlaing [19]. He introduces the idea of planning for a particle with constraints on its acceleration. Unfortunately, the analysis there is restricted to the case that the particle moves in one-dimension. As mentioned before, Canny et al. [2] address the problem of a particle in multiple dimensions.

2 An Algorithm to Generate Time-Optimal Paths

In this section we outline our algorithm. The algorithm is based on discretizing time and joint accelerations, dynamically generating a finite graph, and searching this graph.

Before generating the graph, the user-specified values ϵ and D_ϵ, together with manipulator parameters, are used to determine

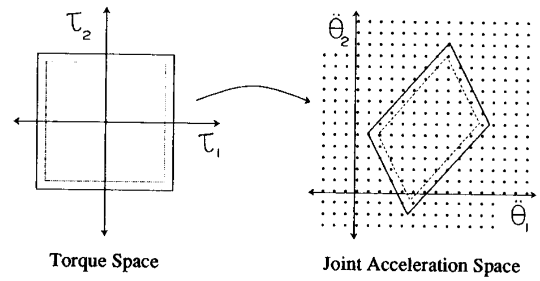


Figure 1: Torque bounds mapped to acceleration space by the inverse dynamics.

the following parameters of the algorithm: the length of the discrete time intervals T, the discretization spacing of the joint accelerations $\Delta\ddot{\theta}$, the number of multi-steps N,[8] and the reduced torque bound $\hat{\tau}_{i,\max} < \tau_{i,\max}$. In [20] and [1] we present a complete description of the algorithm and give the details necessary to compute T, N, $\Delta\ddot{\theta}$, and $\hat{\tau}_{\max}$.

The graph is formed in the following manner. Suppose that at time $t = \frac{(kN+j)T}{N}$ we consider the state $(\Theta(\frac{(kN+j)T}{N}), \dot{\Theta}(\frac{(kN+j)T}{N}))$, where $j < N$ and k and j are non-negative integers. We note that the set of states reachable at time $t = \frac{(kN+j+1)T}{N}$, by following trajectories for which the joint accelerations are fixed, defines a hyperplane in state space. Because we further restrict the joint accelerations to be some multiple of $\Delta\ddot{\theta}$, the reachable states are points lying on this hyperplane. By repeating this process, we form a lattice in state space, with with basis vectors $(\Delta\ddot{\theta}\left(\frac{T}{N}\right)^2, 0)$ and $(\frac{1}{2}\Delta\ddot{\theta}\left(\frac{T}{N}\right)^2, \Delta\ddot{\theta}\left(\frac{T}{N}\right))$. Furthermore, we have a notion of adjacency between states in the lattice defined by the existence of a constant discretized acceleration trajectory passing from one to another.

Up to this point, we have not taken the torque restrictions and obstacles into account. We can think of these as eliminating some of the adjacencies between states. First, because there are bounds on the allowable torques, not all discretized accelerations are allowed. Unfortunately, the allowable accelerations are defined by the torque bounds mapped through the inverse dynamics of the manipulator, and the inverse dynamics are state-dependent. To handle this problem we define a lower torque bound $\hat{\tau}_{\max}$, and we consider only the set of discrete joint accelerations allowed by this reduced torque bound $\hat{\tau}_{\max}$ given the dynamics for the state $(\Theta(kT), \dot{\Theta}(kT))$.[9] In Figure 1 we illustrate this for a two-link manipulator, the allowable accelerations being the points inside the inner parallelogram.

We also remove those adjacencies for which the robot passes closer to an obstacle than allowed by the safety margins $D - D_\epsilon$. Taking all of these restrictions into account, we can define the points in state space which can be reached from a given state in the lattice.

Because the transitions between states all take the same amount of time ($\frac{T}{N}$), a breadth-first search is sufficient to find the minimal time path following these restricted trajectories.

For a single joint a small portion of the graph is shown in

[8]The discretization of time and the restrictions on maximum torque force us to subdivide each interval $[kT, (k+1)T]$ into N subintervals, which we call **multi-steps**. The need for multi-steps is illustrated in Figure 4.

[9]We could recompute the dynamics at the end of each interval of time $\frac{T}{N}$, but we will show that this is not necessary. We only compute the set of allowable accelerations at $t = kT$ and use this set for the N steps of length T/N in the interval $[kT, (k+1)T]$.

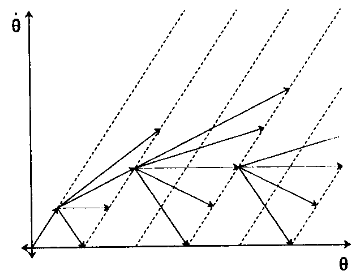


Figure 2: Graph in State Space

Figure 2.

3 Formal Statement of the Result

In this section, we present a precise and formal statement of the central result of this paper. We have listed below all of the assumptions made in the construction of our algorithm and the proof of the result. Some of these assumptions are intrinsic to our formulation of the problem, others deal with technical difficulties in the proof, and two are fundamental and underlie the entire structure of the approximation technique. A discussion of the implications of these assumptions has been presented in [20].

3.1 Assumptions

We provide a list of the assumptions used in proving our results. For clarity we gather all of the assumptions in one place.

We assume that:

1. The manipulator is an m-jointed open-kinematic-chain with prismatic and revolute joints.
2. The torque bounds are given by $\tau_{\max}$.
3. The safety margin and tracking error are such that $D(\Theta,\dot{\Theta}) > D_\epsilon(\Theta,\dot{\Theta}) > 0$, and both are bounded away from zero.
4. The kinematics and mass distribution of the manipulator are known, thus we can calculate the dynamics bounds in [4].
5. There is a lower bound on the minimum singular value of the inertia tensor.
6. The actuator torques can overcome gravity in all configurations of the manipulator (i.e. $\tau_{i,\max} > G_{i,\max}$). This restricts us from considering all open-kinematic-chain manipulators, but it is a reasonable assumption for practical robots.
7. The joint velocities are upper bounded by $\dot{\Theta}_{\max}$.
8. $\dot{\Theta}(0) = \mathbf{0}$ and $\dot{\Theta}(t_f) = \mathbf{0}$. [10]

3.2 Formal Result

Letting t_f^* be the minimal time for a trajectory satisfying the conditions of Section 1.1, we are prepared to state the main result of the paper.

Proposition 1 *Given any $\epsilon > 0$, $D(\Theta,\dot{\Theta})$, $D_\epsilon(\Theta,\dot{\Theta})$, Θ_0, Θ_f, $\tau_{\max}$, and $\dot{\Theta}_{\max}$ with Assumptions 1-8 satisfied, there exists choices of T, $\Delta\ddot{\theta}$, $\frac{1}{N}$, and $\hat{\tau}_{\max}$ that depend polynomially on the input parameters such that our Algorithm will produce a trajectory $\hat{\Theta}(t)$ for which*

1. *$\hat{\Theta}(0) = \Theta_0$, $\hat{\Theta}(\hat{t}_f) = \Theta_f$, $\dot{\hat{\Theta}}(0) = 0$, $\dot{\hat{\Theta}}(\hat{t}_f) = 0$*
2. *$\forall t \in [0,\hat{t}_f]$, the minimum distance between the manipulator in the configuration $\hat{\Theta}(t)$ and the obstacles is greater than $D(\hat{\Theta},\dot{\hat{\Theta}}) - D_\epsilon(\hat{\Theta},\dot{\hat{\Theta}})$.*
3. *$\forall t \in [0,\hat{t}_f]$, the torques required by the actuators are upper-bounded by $\tau_{\max}$.*
4. *$\forall t \in [0,\hat{t}_f]$, $|\dot{\theta}_i(t)| \leq \dot{\theta}_{i,\max}$ for each joint i.*
5. *The time $\hat{t}_f$ to traverse $\hat{\Theta}(t)$ satisfies $\hat{t}_f \leq (1+\epsilon)t_f^*$.*

[10] This assumption can be relaxed if we are willing to allow errors, proportional to $1 - \frac{1}{1+\epsilon} = \frac{\epsilon}{1+\epsilon}$, in matching the final velocity.

4 Proof of the Correctness of the Algorithm

In this section we present an outline of the proof of Proposition 1. The major steps necessary to demonstrate that we can choose T, $\Delta\ddot{\theta}$, $\frac{1}{N}$, and $\hat{\tau}_{\max}$ to depend polynomially on ϵ while guaranteeing the performance of the algorithm are discussed. Details of these steps, which are long and complicated, are presented in [1].

Before proceeding we define the two types of trajectories we will be comparing:

unconstrained trajectory a trajectory $\Theta(t)$ defined over $[0,\tilde{t}_f]$ such that $|\tau_i(t)| \leq \tilde{\tau}_{i,\max}$, $\Theta(0) = \Theta_0$, $\Theta(\tilde{t}_f) = \Theta_f$, $|\dot{\theta}_i(t)| \leq \dot{\theta}_{i,\max}$, and respecting the safety margin $D(\Theta,\dot{\Theta})$. For these trajectories $\ddot{\Theta}(t)$ is allowed to vary with only the constraint that the torque bounds be respected.

constrained trajectory a trajectory $\Theta(t)$ defined over $[0,\hat{t}_f]$ such that $|\tau_i(t)| \leq \tau_{i,\max}$, $\Theta(0) = \Theta_0$, $\Theta(\hat{t}_f) = \Theta_f$, $|\dot{\theta}_i(t)| \leq \dot{\theta}_{i,\max}$, respecting the safety margin $D(\Theta,\dot{\Theta}) - D_\epsilon(\Theta,\dot{\Theta})$, and having $\ddot{\Theta}$ constrained. $\ddot{\Theta}$ is constrained as follows:

1. $\ddot{\Theta}(t)$ is piecewise constant on $[kT,(k+1)T]$ changing only at increments of $\frac{1}{N}T$.
2. for $t \in [kT,(k+1)T]$ $\ddot{\Theta}(t) \in S_{\hat{\tau}}(kT)$ where

$$S_{\hat{\tau}}(kT) \stackrel{\text{def}}{=} \left\{ y \in \mathbf{R}^m : \left| \left(M(\Theta(kT))(y) + J(\Theta(kT),\dot{\Theta}(kT)) \right)_i \right| \leq \hat{\tau}_{i,\max} \right\}$$

(i.e. the set of joint accelerations that respect the torque bound $\hat{\tau}_{\max}$ at the state $(\Theta(kT),\dot{\Theta}(kT))$)

3. $\ddot{\Theta}(t) \in \mathbf{Z}^m \Delta\ddot{\theta}$

where $\tilde{\tau}_{i,\max} < \hat{\tau}_{i,\max} < \tau_{i,\max}$ will be specified later.

We use the superscript asterisk to denote the minimal time for a trajectory which satisfies certain conditions. Thus $\tilde{t}_f^*$ refers to the minimal unconstrained trajectory which respects $\tilde{\tau}_{\max}$, t_f^* to the minimal unconstrained trajectory which respects $\tau_{\max}$, and $\hat{t}_f^*$ to the minimal constrained trajectory which respects $\hat{\tau}_{\max}$.

First we show by time-scaling that we can choose $\tilde{\tau}_{i,\max} < \tau_{i,\max}$ such that $\tilde{t}_f^* \leq (1+\epsilon)t_f^*$. Next we show that we can choose $\hat{\tau}_{\max}$, $\Delta\ddot{\theta}$, N, and T to guarantee that given any unconstrained trajectory respecting the torque bounds $\tilde{\tau}_{\max}$ there exists a constrained trajectory respecting $\tau_{\max}$ such that $\hat{t}_f \leq \tilde{t}_f$ (i.e. the constrained trajectory is at least as fast as the unconstrained trajectory). In particular, we can follow a time-optimal unconstrained trajectory so we have that

$$\hat{t}_f^* \leq \tilde{t}_f^* \leq (1+\epsilon)t_f^* \quad ,$$

thus we have a constrained trajectory which approximates a time-optimal trajectory.

The outline of the proof is similar to that used in [2]. However, in significant ways our proof structure is different and the individual steps more difficult. We consider general safety margins; we use a discretized set of accelerations, with the allowable ones depending on the state; and we consider a general manipulator and not just a particle. The state-dependence of the dynamics substantially increases the complexity of the argument showing that the constrained trajectory can *track*[11] the unconstrained trajectory, and the coupling forces us to deal with a multi-dimensional problem.

We divide the proof into three parts with corresponding sections. First we examine time-scaling to determine $\tilde{\tau}_{\max}$ and $\hat{\tau}_{\max}$; next we reduce the dynamics to be locally constant; and finally we decouple the dynamics and track along these decoupled axes.

4.1 Time-Scaling

Time-scaling arguments [3] as detailed in [1], imply that if we choose

$$\tilde{\tau}_{\max} = \frac{1}{(1+\epsilon)^2}(\tau_{\max} - G_{\max}) + G_{\max},$$

where $G_{\max}$ is the maximum gravitational torques, then $\tilde{t}_f^* \leq (1+\epsilon)t_f^*$. Further, we define

$$\begin{aligned} \gamma &= \tau_{\max} - \tilde{\tau}_{\max} \\ &= \left(1 - \frac{1}{(1+\epsilon)^2}\right)(\tau_{\max} - G_{\max}). \end{aligned}$$

We set

$$\hat{\tau}_{\max} = \tilde{\tau}_{\max} + \frac{4}{5}\gamma,$$

where $\hat{\tau}_{\max}$ is the torque constraint which the constrained trajectory must satisfy at the beginning of each interval of time $\frac{T}{N}$. We are left to show that the constrained trajectory can keep up with the unconstrained trajectory.

4.2 Reduction to Locally Constant Dynamics

The set of joint accelerations which respect the torque constraints are given by the dynamics equations for the manipulator:

$$\ddot{\Theta} = M^{-1}(\Theta)\left(\tau - J(\Theta,\dot{\Theta})\right) \quad \text{with} \quad \tau \in \mathbf{R}^m : |\tau_i| \leq \tau_{i,\max},$$

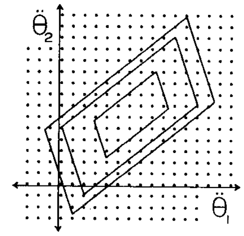
where $J(\Theta,\dot{\Theta})$ combines the coriolis, centripetal, and gravity forces. The mapping of the allowable set of torques is illustrated in Figure 1 for a two-link manipulator. For the unconstrained trajectory the set of allowable accelerations is continually changing since the dynamics depend on the state. This makes it more difficult to show that the constrained trajectory can keep up with the unconstrained trajectory. However, we use results in [4] describing bounds on the dynamics and on the derivatives of the dynamics. This enables us to bound how the sets of allowable joint accelerations change as the state is changed.

Showing that the constrained trajectory can track the unconstrained trajectory is simpler if we can consider the dynamics to be locally constant. To do this there are two problems to address. First we must show that a time interval may be chosen such that the constrained trajectory (which has piecewise constant accelerations) doesn't violate the torque bounds $\tau_{\max}$ at any time. Second we need to show that over the interval $[kT,(k+1)T]$ we can choose an acceleration discretization such that the constrained trajectory maintains an acceleration advantage over the unconstrained trajectory. The allowable accelerations at $t = kT$ for a 2-link manipulator are depicted in Figure 3. The outer ($\tau_{\max}$) and inner ($\tilde{\tau}_{\max}$) parallelepipeds should be thought of as changing as the state varies. The middle one ($\hat{\tau}_{\max}$) is fixed for the interval $[kT,(k+1)T]$; it is used to determine the constrained trajectory's allowable joint accelerations over the interval.

Before proceeding we see that the minimum distance between the outer and inner parallelepipeds, $\delta\ddot{\Theta}$, is bounded by

$$\delta\ddot{\Theta} \geq \sigma_{min}(M^{-1}(\Theta)) \min_i |\gamma_i|. \tag{4.1}$$

[11] By *track* we mean stay close in both position and velocity for all time.



Joint Acceleration Space

Figure 3: Allowable Joint Accelerations at $t = kT$ (the outer box is associated with $\tau_{\max}$, the middle with $\hat{\tau}_{\max}$, and the inner with $\tilde{\tau}_{\max}$ – the difference has been exaggerated for illustration)

which is the minimum difference in acceleration between a torque associated with $\tau_{\max}$ and a torque associated with $\tilde{\tau}_{\max}$.[12]

To insure that a constrained trajectory does not violate the torque bounds ($\tau_{\max}$) it is equivalent to insure that the outer box ($\tau_{\max}$), over the interval $[kT,(k+1)T]$, doesn't move enough to intersect the middle box ($\hat{\tau}_{\max}$). By Lemma A.1 we find that in time T the box moves at most $f(T)$. Thus we require T to be such that $f(T) \leq \frac{1}{5}\delta\ddot{\Theta}$. Lemma A.1 implies that we can choose $\frac{1}{T} \in O(\frac{1}{\epsilon})$ and satisfy this constraint.

Next we want to insure that the constrained trajectory maintains an advantage in acceleration over the unconstrained trajectory. Equivalently we need to show that the allowable accelerations for the unconstrained trajectory over $[kT,(k+1)T]$ remain bounded away from the maximum allowable accelerations for the constrained trajectory at $t = kT$. In Figure 3 this corresponds to the keeping the inner box bounded away from the middle box. There are two factors which contribute to changes in the allowable accelerations for the unconstrained trajectory 1) as time progresses the state changes and 2) at each time $t = kT$ there is some state error between the unconstrained and constrained trajectories. For the first factor we choose T as above, so that the change in accelerations is less than $\frac{1}{5}\delta\ddot{\Theta}$. For the second we calculate bounds on the initial state error at each step to guarantee that the associated acceleration difference is no more than $\frac{1}{5}\delta\ddot{\Theta}$. Lemma A.2 states that for an initial state error $(\Delta\Theta, \Delta\dot{\Theta})$ the change in acceleration is less than $g(\Delta\Theta, \Delta\dot{\Theta})$, thus we require $(\Delta\Theta, \Delta\dot{\Theta})$ to be such that $g(\Delta\Theta, \Delta\dot{\Theta}) \leq \frac{1}{5}\delta\ddot{\Theta}$. In [1] we show that at the end of each step the position error is less than $\frac{\sqrt{m}}{2}\Delta\ddot{\theta}\left(\frac{T}{N}\right)^2$ and the velocity error is less than $\frac{\sqrt{m}}{2}\Delta\ddot{\theta}\left(\frac{T}{N}\right)$. Thus if we require $\Delta\ddot{\theta}$, T, N to be such that $\frac{\sqrt{m}}{2}\Delta\ddot{\theta}\left(\frac{T}{N}\right)^2 \leq \Delta\Theta$ and $\frac{\sqrt{m}}{2}\Delta\ddot{\theta}\left(\frac{T}{N}\right) \leq \Delta\dot{\Theta}$ we will satisfy the desired condition.

Finally we require the acceleration discretization be such that $\sqrt{m}\Delta\ddot{\theta} \leq \frac{1}{5}\delta\ddot{\Theta}$. Equation 4.1 implies that this constraint can be satisfied if we choose $\frac{1}{\Delta\ddot{\theta}} \in O(\frac{1}{\epsilon})$. This guarantees the existence of discretized accelerations below those associated with $\hat{\tau}_{\max} = \tilde{\tau}_{\max} + \frac{4}{5}\gamma$ and more than $\frac{1}{5}\delta\ddot{\Theta}$ above those that the unconstrained trajectory may have. Thus we have reduced the problem to one where the constrained trajectory has more acceleration than the unconstrained trajectory over the interval $[kT,(k+1)T]$. In addition the amount of extra acceleration can be shown to be at least a factor A larger, where

$$A \geq \min_i \frac{\tilde{\tau}_{i,\max} + \frac{3}{5}\gamma_i}{\tilde{\tau}_{i,\max} + \frac{2}{5}\gamma_i} - 1$$

[12] By associated with $\tau_{\max}$ we mean that for at least one i, $|\tau_i| = \tau_{i,\max}$.

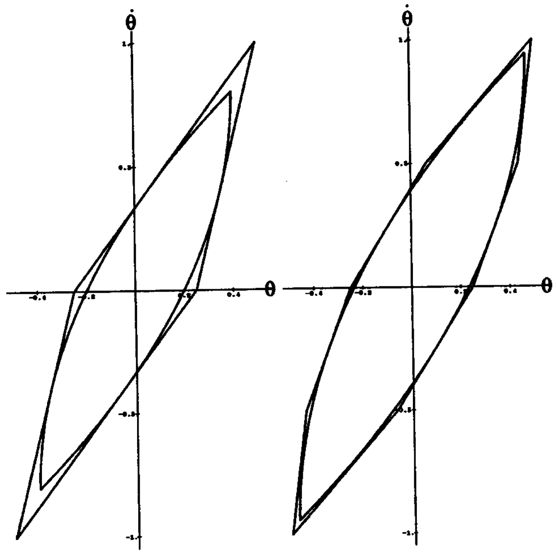


Figure 4: Reachable Locus of Points in State Space

$$\geq \frac{1}{5}\frac{2\epsilon+\epsilon^2}{(1+\epsilon)^2}\min_i\left(\frac{\tau_{i,\max}-G_{i,\max}}{\tau_{i,\max}}\right). \qquad (4.2)$$

Therefore, for the interval $[kT,(k+1)T]$ we can consider the dynamics to be locally constant with the constrained trajectory having an acceleration edge over the unconstrained trajectory by a factor of A.

4.3 Multi-Axis Tracking

To complete the proof we need to show that there exists a constrained trajectory which tracks the unconstrained trajectory. Given the results of the previous section we consider the simpler problem of an unconstrained and constrained particle where the constrained particle has a factor A more acceleration. We can do this since we have reduced the dynamics to being constant over each time interval $[kT,(k+1)T]$.

We proceed by performing a transformation to a new set of coordinates (the decoupled coordinates) in which the acceleration bounds are decoupled. The decoupling transformation used is $M(\Theta(kT))(\cdot)+J(\Theta(kT),\dot{\Theta}(kT))$. Thus we have reduced the problem to tracking a particle along a single axis given that the constrained particle has a factor A more acceleration.

Actually, we show that not only can we track the particle, but that the constrained particle matches the final position and velocity of the unconstrained particle within errors caused by discretization of the accelerations. If the constrained particle's accelerations are not discretized it can match the final position and velocity of the unconstrained particle given suitable constraints on the initial errors between the two. Equivalently, the region in state space that the constrained particle (with continuous accelerations) can reach is larger than the region that the unconstrained particle can reach. Discretization of the accelerations places a lattice on the reachable state space and the nearest lattice point is chosen.

The need for multistep trajectories is illustrated in Figure 4. In this figure, we have assumed that the unconstrained and constrained trajectories start at the same state. The inner lens-shaped regions contain the set of states reachable by the unconstrained particle after a time T with a fixed upper bound on its acceleration. The outer polygonal regions contain those states which can be reached when the accelerations can be changed only at certain fractions of T, but which have an higher upper bound on acceleration. If the constrained particle is allowed a factor $A=\frac{1}{4}$ extra acceleration, then only one change in acceleration (at time $\frac{T}{2}$) is sufficient to allow the constrained particle to reach any state reachable by the unconstrained particle. This is illustrated by the figure on the left. However, if $A=\frac{1}{16}$ it is necessary to use a 4-step trajectory as illustrated by the right figure. Detailed analysis in [1] shows that the number of multi-steps N is such that $N \geq \sqrt{\frac{2}{A}}$. Combining this with Equation 4.2, implies that we can choose $N \in O(\frac{1}{\sqrt{\epsilon}})$.

When we discretize the accelerations for the constrained particle we need to insure that the state errors at the end of each time interval $[kT,(k+1)T]$, when projected into the decoupled coordinates, are small enough that the constrained particle's reachable set still covers the reachable set of the unconstrained trajectory in the next interval. This is accomplished by choosing $\Delta\ddot{\theta}$ appropriately small, as specified in Lemma A.3, and implies that we can choose $\frac{1}{\Delta\ddot{\theta}} \in O(\frac{1}{\sqrt{\epsilon}})$.

Finally, we guarantee that the worst case deviation between the constrained and unconstrained trajectories is less than D_ϵ. Thus when we are tracking the optimal unconstrained trajectory, which avoids obstacles by D, the constrained trajectory will avoid the obstacles by at least $D-D_\epsilon$. Lemma A.4 implies that to guarantee this we can choose $\frac{1}{T} \in O(\frac{1}{\sqrt{D_\epsilon}})$.

4.4 Complexity of the Algorithm

In [1] we show that the running time of our algorithm is

$$O\left(\left[\frac{\|\dot{\Theta}_{\max}\|_\infty \|L\|_\infty \|\tau_{\max}\|_\infty N^3}{\Delta\ddot{\theta}^3\, T^3\, \sigma_{min}(M)}\right]^m\right),$$

and in terms of the variable $\frac{1}{\epsilon}$ the algorithm has a running time $O\left(\left(\frac{1}{\epsilon}\right)^{7.5m}\right)$.

5 Extensions to the Algorithm

It is possible to relax some of the assumptions we have made to yield an algorithm which is more general in scope. The extensions which are discussed in this section are straight-forward. We have indicated the general method of solution and consider the remaining work involved in the implementation to be computational in nature.

State Dependent Torque Bounds If the torque bounds were expressed as a function of the manipulator state, under reasonable conditions, another factor allowing the torque bounds to be considered locally constant could be included. The analysis would follow that of the reduction to locally constant dynamics. The rest of our proof would remain unchanged.

State Dependent Determination of Multi-Steps The number of times at which the joint accelerations changes is determined by worst-case bounds. These in turn effect the spacing of the lattice which is generated in state space. In order to spend as little time as possible in computing the time-optimal safe trajectory we would like the lattice to be as sparse as possible. We note from [1] that the number of steps taken is dependent upon the extra acceleration that the constrained trajectory has. In all but the the worst case, the trajectory which follows the lattice will have a much larger advantage in acceleration than was used to compute the number of steps. If we spend the time to compute how much advantage it actually has, then we can choose a smaller number of steps. This directly translates into a search graph with less links, and hence leads to a reduced computation time.

References

[1] P. E. Jacobs, G. Heinzinger, J. Canny, and B. Paden, "Planning guaranteed near-time-optimal trajectories for a manipulator in a cluttered workspace," Tech. Rep. RAMP 89-15, University of California, Berkeley, Engineering Systems Research Center, September 1989.

[2] J. Canny, B. Donald, J. Reif, and P. Xavier, "On the complexity of kinodynamic planning," in 29^{th} *Symposium on the Foundations of Computer Science*, (White Plains, NY.), pp. 306–316, October 1988.

[3] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," in *Proceedings of the ACC*, (San Francisco, CA), pp. 752–756, The American Automatic Control Council, June 1983.

[4] G. Heinzinger and B. Paden, "Bounds on robot dynamics," in *IEEE Conference on Robotics and Automation*, (Scottsdale, AZ), pp. 1744–1751, May 1989.

[5] J. M. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, *Robot Motion: Planning and Control*. Cambridge MA: The MIT Press, 1982.

[6] J. F. Canny, *The Complexity of Robot Motion Planning*. Cambridge MA: The MIT Press, 1988.

[7] S. Whitesides, "Computational geometry and motion planning," in *Computational Geometry* (G. Toussaint, ed.), pp. 377–427, North-Holland, 1985.

[8] B. Donald and P. Xavier, "A provably good approximation algorithm for optimal-time trajectory planning," in *1989 IEEE International Conference on Robotics and Automation*, (Scottsdale), pp. 958 – 963, May 1989.

[9] M. Kahn and B. Roth, "The near-minimum-time control of open-loop articulated kinematic chains," *Journal of Dynamic Systems, Measurement and Control*, vol. 93, pp. 164–172, 1971.

[10] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "On the optimal control of robotic manipulators with actuator constraints," in *Proceedings of the ACC*, (San Francisco, CA), pp. 782–787, The American Automatic Control Council, June 1983.

[11] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. AC-30, pp. 531–541, June 1985.

[12] J. E. Bobrow, "Optimal robot path planning using the minimum-time criterion," *IEEE Journal of Robots and Automation*, vol. 4, pp. 443–450, August 1988.

[13] B. Paden, A. Mees, and M. Fisher, "Path planning using a jacobian-based freespace generation algorithm," in *IEEE Conference on Robotics and Automation*, (Scottsdale, AZ), pp. 1732–1737, May 1989.

[14] G. Sahar and J. M. Hollerbach, "Planning of minimum-time trajectories for robot arms," Tech. Rep. A.I. Memo No. 804, MIT, November 1984.

[15] E. D. Sontag and H. J. Sussmann, "Remarks on the time-optimal control of two-link manipulators," in *Proc. of the 24th Conference on Decision and Control*, (Ft. Lauderdale FL), pp. 1646–1652, IEEE, December 1985.

[16] R. P. Paul, "Manipulator path control," in *Proceedings of the IEEE International Conference on Cybernetics and Society*, (New York), pp. 147–152, September 1975.

[17] T. O. Binford, "Visual perception by computer," in *Proceedings of IEEE Conference on Systems and Control*, 1971.

[18] R. A. Finkel, "Constructing and debugging manipulator programs," Tech. Rep. AIM 284, Stanford University, August 1976.

[19] C. Ó'Dúnlaing, "Motion planning with inertial constraints," Robotics Report 73, New York University, July 1986.

[20] P. E. Jacobs, G. Heinzinger, J. Canny, and B. Paden, "Planning guaranteed near-time-optimal trajectories for a manipulator in a cluttered workspace," in *Proceedings of the International Workshop on Sensorial Integration for Industrial Robots: Architectures and Applications*, (Zaragoza, Spain), November 1989.

A Technical Lemmas

In this appendix we state various technical lemmas used in the proof of our algorithm. For more details and for proofs of the following the reader is referred to [1]. We assume we are given the dynamical parameters $\sigma_{\min}(M(\Theta))$, $\bar{M}$, and $\bar{G}$.[13]

Lemma A.1 *For a constant torque the change in acceleration,* $\|\ddot{\Theta} - \ddot{\Theta}_0\|$, *over a time interval* T *is less than* $f(T)$, *where*

$$f(T) = \frac{2\sqrt{m}\bar{M}}{\sigma_{\min}(M(\Theta))}\left(\begin{array}{c} \frac{1}{2}\|\ddot{\Theta}_{\max}\|_1^2 T^2 + \|\ddot{\Theta}_{\max}\|_1\|\dot{\Theta}_{\max}\|_1 T \\ + \frac{3}{2}\|\dot{\Theta}_{\max}\|_1^2\|\ddot{\Theta}_{\max}\|_1 T^2 + 3\|\dot{\Theta}_{\max}\|_1^3 T \\ + 3\|\dot{\Theta}_{\max}\|_1\|\ddot{\Theta}_{\max}\|_1 T \end{array}\right) + \frac{\sqrt{m}\bar{G}}{\sigma_{\min}(M(\Theta))}\left(\frac{1}{2}\|\ddot{\Theta}_{\max}\|_1 T^2 + \|\dot{\Theta}_{\max}\|_1 T\right).$$

Lemma A.2 *For a constant torque the change in acceleration,* $\|\ddot{\Theta} - \ddot{\Theta}_0\|$, *due to a state error* $(\Delta\Theta, \Delta\dot{\Theta})$ *is less than* $g(\Delta\Theta, \Delta\dot{\Theta})$, *where*

$$g(\Delta\Theta, \Delta\dot{\Theta}) = \frac{\sqrt{m}}{\sigma_{\min}(M(\Theta))}\left[\begin{array}{l} 2\bar{M}\left(\begin{array}{c} \|\ddot{\Theta}_{\max}\|_1\Delta\Theta + 3\|\dot{\Theta}_{\max}\|_1^2\Delta\Theta \\ + 3\|\dot{\Theta}_{\max}\|_1\Delta\dot{\Theta} \end{array}\right) \\ + \bar{G}\Delta\Theta \end{array}\right]$$

Lemma A.3 *For the discretization error to be sufficiently small we require*

$$\Delta\ddot{\theta} \leq \frac{\min(\frac{1}{4}, \frac{\sqrt{4}}{3\sqrt{2}})}{\sqrt{m}\sigma_{\max}(M(\Theta))} \min_i \tau_{i,\max}. \tag{A.3}$$

Lemma A.4 *To guarantee the maximum deviation is sufficiently small we require that*

$$T \leq \sqrt{\frac{24}{5}\frac{1}{\max_i \tau_{i,\max}}\sigma_{\min}(M(\Theta))\dot{D}_\epsilon}. \tag{A.4}$$

[13] See [1] on how to define and calculate $\bar{M}$ and $\bar{G}$.