Selection of Near-Minimum Time Geometric Paths for Robotic Manipulators

KANG G. SHIN, SENIOR MEMBER, IEEE, AND NEIL D. MCKAY

Abstract—**A number of trajectory planning algorithms exist for calculating the joint positions, velocities, and torques which will drive a robotic manipulator along a given geometric path in minimum time. However, the time depends upon the geometric path, so the traversal time of the path should be considered again for geometric planning. There are algorithms available for finding minimum distance paths, but even when obstacle avoidance is not an issue, minimum (Cartesian) distance is not necessarily equivalent to minimum time.**

In this paper, we have derived a lower bound on the time required to move a manipulator from one point to another, and determined the form of the path which minimizes this lower bound. As numerical examples, we have applied the path solution to the first three joints of the Bendix PACS arm and the Stanford arm. These examples do indeed demonstrate that the derived approximate solutions usually require less time than Cartesian straight-line (minimum-distance) paths and joint-interpolated paths.

I. INTRODUCTION

PRODUCTIVITY increase is the goal of the utmost importance in contemporary automation with programmable robotic manipulators. Driving robotic manipulators as fast as possible, i.e., minimum time control of manipulators, is an important means of achieving this goal. Minimum time control of manipulators generates several interesting but difficult control and planning problems. This paper is intended to treat one such problem, that is, minimum time geometric path planning for manipulators.

Loosely speaking, the problem of minimum time control (MTC) of a manipulator is concerned with the determination of control signals that will drive the manipulator from a given initial configuration to a given final configuration in as short a time as possible, given constraints on the magnitudes of the control signals and constraints on the intermediate configurations of the manipulator, i.e., given that the manipulator must not hit any obstacles. In general, it is extremely difficult, if not impossible, to obtain an exact closed-form solution to the MTC problem due mainly to 1) the nonlinearity and coupling in the manipulator dynamics, and 2) the complexity involved with collision avoidance. One way to sidestep the collision avoidance problem 2) is to assume that the desired geometric or spatial path has been specified *a priori*. As to the difficulty 1), although there are a few suboptimal solutions derived using approximate manipulator dynamics [3], [4], the MTC problem is usually divided into two subproblems, i.e., *trajectory planning* and *trajectory tracking*, each of which is then solved separately. This division of the MTC problem can best be explained by Fig. 1. From a *task planner* we obtain an ordered sequence of points in Cartesian space which represent a collision-free path if we connect them properly (e.g., by spline functions or straight line segments). The *geometric path generator* 1) *transforms* these Cartesian points to the corresponding points in joint space,[1] and 2) using the transformed points in joint space, *generates* a geometric path which is a parameterized curve in joint space. The trajectory planner receives these geometric paths as input and determines a time history of position, velocity, acceleration, and input torques which are then fed to the *trajectory tracker*.

With the division outlined above, we have formulated in [10] the minimum time trajectory planning (MTTP) problem to determine controls which will drive a given manipulator along a specified curve in joint space in minimum time, given constraints on initial and final positions and velocities as well as on control signal magnitudes. Since a geometric path can be described as a parameterized curve, and the geometric path is assumed to be given, trajectory planning is relatively simple. By introducing a single parameter which describes the manipulator's position, the dimensionality of the problem has been reduced considerably. The current state (joint positions and velocities) of the manipulator can be described in terms of the parameter used to describe the geometric path and its time derivative. The MTTP problem is therefore essentially a *two*-dimensional minimum time control problem with some state and input constraints.

More formally, assume that the geometric path is given in the form of a parameterized curve, say

$$q^i = f^i(\lambda), \qquad 0 \le \lambda \le \lambda_{\max} \tag{1.1}$$

where q^i is the position of the ith joint, the initial and final points on the trajectory correspond to the points $\lambda = 0$ and $\lambda = \lambda_{\max}$, respectively, and the functions f^i are continuous and piecewise differentiable. Also assume that the bounds on the actuator torques can be expressed in terms of the state of the system, i.e., the manipulator's speed and position, so that

$$u_i^{\min}(q, \dot{q}) \le u_i \le u_i^{\max}(q, \dot{q}) \tag{1.2}$$

where u_i is an n-dimensional vector of actuator torques/forces; $u_i^{\max}$ and $u_i^{\min}$ are n-dimensional vectors that represent the maximum and minimum torque bounds, respectively, and n is the number of joints that the manipulator has. Note that 1) the torque bounds may in general be functions of joint position and velocity, and 2) the vector inequality (1.2) denotes component-wise inequalities. Given the functions f^i in (1.1), the inequality (1.2), the desired initial and final positions and velocities, and the manipulator dynamic equations to be given in (4.1), the MTTP problem is to find $q(\lambda)$ and $\dot{q}(\lambda)$, and hence the controls $u_i(\lambda)$

Manuscript received February 19, 1985; revised December 3, 1985. This paper is based on a prior submission of October 11, 1984. Paper recommended by Associate Editor, W. J. Book. This work was supported in part by the National Science Foundation under Grant ECS-8409938 and the U.S. Air Force Office of Scientific Research under Contracts F49620-82-C-0089 and F33615-85-C-5105.

K. G. Shin is with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109.

N. D. McKay was with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109. He is now with the Computer Science Department, General Motors Research Laboratories, Warren, MI 48090.

IEEE Log Number 8607826.

[1] This transformation requires the solution of the inverse kinematic problem, which is not, however, a subject of this paper.

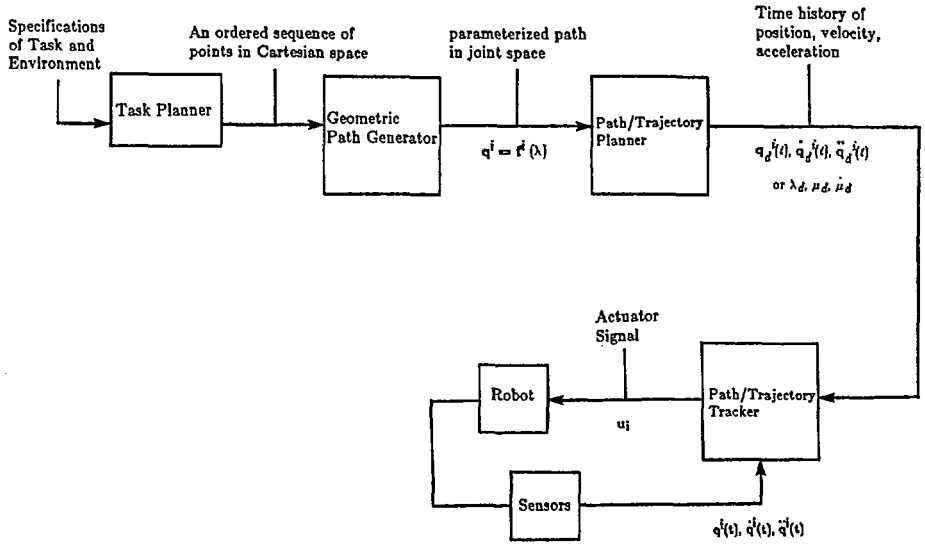


Fig. 1. Functional block diagram of manipulator position control.

which minimize the time T given by

$$T = \int_0^{t_f} 1 \cdot dt = \int_0^{\lambda_{\max}} \frac{dt}{d\lambda}\, d\lambda = \int_0^{\lambda_{\max}} \frac{1}{\mu}\, d\lambda \qquad (1.3)$$

where $\mu \equiv d\lambda/dt$ is the "speed" of the manipulator. See [10] for more detailed descriptions of our solution to the MTTP problem.

In terms of the trajectory planning problem, the geometric path planning problem is the problem of picking the parametric functions f^i. In contrast to the trajectory planning problem, in which the desired solutions can be expressed in terms of the position parameter λ and its first and second time derivatives, the geometric path planning problem requires that a set of functions be chosen from an *infinite*-dimensional space, thereby leading to a more difficult problem to solve.

Some of the earlier trajectory planning techniques restricted the form of the geometric paths to a set of corner points connected by straight lines [5], [7]. The trajectory planner than "rounds off" the corners. But straight lines are not simple motions to produce for most manipulators, and so are not necessarily the best paths to choose. It will be assumed in this paper that a more flexible trajectory planning scheme like the one in [10] will be used.

It should also be noted that the shortest path may not be the minimum time path. In particular, the shortest Cartesian path may have one or more sharp corners,[2] and the manipulator would have to come to a complete stop at these points. This is certainly undesirable in view of the need for minimum time control of manipulators.

In this paper, we will develop a method for determining an approximate minimum time geometric path for our previous trajectory planners described above and in [10]. This is a significant departure from most of the conventional planning methods in which geometric path planning [6] and trajectory planning have been performed *separately* and *independently*. Due to the intimate, synergistic relationship between the two, it is clear that the conventional methods will lead to inefficient solutions, e.g., not truly minimum time path solutions or even infeasible solutions. Specifically, we intend in this paper to remedy this inefficiency by considering the effects of actuator constraints and robot dynamics in both geometric path and trajectory planning.

This paper is organized as follows. Section II describes tensor notation and those aspects of Riemannian geometry which are used in the following sections. In Section III, we state formally the minimum time geometric path planning (MTGPP) problem to be solved in conjunction with trajectory planning. Section IV discusses some interesting dynamic properties of manipulators that are useful for deriving solutions to the MTGPP problem. In Section V, we present 1) an exact solution to the MTGPP problem under certain restricted conditions, 2) a method for finding lower bounds on the traversal time from one point to another, and 3) the paths which result from a) minimizing the traversal time bounds and b) using the velocity bounds derived in [10]. Section VI provides a real feeling for our solutions by presenting some examples which are based on the first three joints of a cylindrical manipulator, called the PACS arm, manufactured by the Bendix Corporation, and a well-known manipulator, the Stanford manipulator. The paper concludes with Section VII.

II. Tensor Notation

Since many of the equations which follow are written in tensor notation, we briefly introduce tensor notation in this section (see [11] for more details). Tensor notation is much like vector notation except that the symbols used in tensor equations may have subscripts and/or superscripts. A vector is written as a symbol with one index, and a matrix will have two. It is possible in tensor notation to write arrays of three or more dimensions; a three-dimensional array simply has three indexes.

Formally, a tensor is a quantity or set of quantities which obeys certain rules when transformed from one set of curvilinear coordinates to another. The transformation rules are of two types, as indicated by the position of the tensor's indexes. Superscripts indicate that the index is *contravariant* and subscripts indicate that it is *covariant*. Tensors of order two can have two subscripts, two superscripts, or one subscript and one superscript. These are called *covariant, contravariant,* and *mixed tensors of order two.* Since we make no use of coordinate transforms in this work, we omit a description of the tensor coordinate transformation rules.

An important notational tool is the so-called *summation convention*. If an index appears twice in a product of two tensor expression, then the expression is summed from 1 to N over the repeated index, where N is the dimension of the space. Thus, $a_i b^i$ is shorthand for $\sum_{i=1}^{N} a_i b^i$. It is important to note that a given index should not appear more than twice in any term of a tensor equation, and that repeated indexes should appear once as a subscript and once as a superscript. If a repeated index appears,

[2] Even if the Cartesian path does not have such sharp corner points, it does not usually yield a minimum time trajectory, e.g., consider a cylindrical manipulator moving along the shortest Cartesian path around its waist.

for example, twice as a subscript, then the resulting quantity will not in general be a tensor. If, on the other hand, the index appears once as a subscript and once as a superscript, it is easily verified that the expression is a tensor whose character is indicated by the remaining (nonrepeated) indexes. For example, $T_{ij}x^j$ is a covariant tensor of order one; the j indexes "cancel."

In what follows, some use will be made of Riemannian geometry, which is the study of the metrical properties of spaces of an arbitrary number of dimensions. The space is described by its *metric tensor*, which gives the square of the differential line element as a quadratic form in the differentials of the coordinates, i.e., $ds^2 = J_{ij}dq^i dq^j$, where J_{ij} is the metric tensor and the q^i and q^j are the coordinates. J_{ij} may without loss of generality be assumed to be symmetric, i.e., $J_{ij} = J_{ji}$. It also will be assumed throughout this paper that the metric tensor is positive definite, i.e., that $J_{ij}x^i x^j > 0$ for all $x \neq 0$.

The introduction of the metric tensor allows distances and angles to be measured, and allows the computation of norms of vectors. Distances along curves are calculated by integrating the formula for the differential line element ds. The norm of a contravariant vector x^i is given by the formula $\|x^i\|^2 \equiv J_{ij}x^i x^j$.

The angle between two vectors x and y is given by the formula

$$\cos\ \theta = \frac{J_{ij}x^i y^j}{\|x\| \cdot \|y\|}. \tag{2.1}$$

If J_{ij} is positive definite, then it can be shown that the right-hand side of (2.1) is always between $+1$ and -1, so that θ is a real angle. Even if J_{ij} is not positive definite, two vectors x and y are said to be *orthogonal* if and only if $J_{ij}x^i y^j = 0$.

The curves in a Riemannian space corresponding to straight lines in Euclidean space are *geodesics*, or curves of minimum distance. The differential equation which describes these curves can be found from the form of the line element ds using variational techniques. Using such techniques, the differential equation obtained is

$$J_{ij}\frac{d^2q^j}{ds^2} + [jk,\ i]\frac{dq^j}{ds}\frac{dq^k}{ds} = 0. \tag{2.2}$$

The symbol $[jk, i]$ is a Christoffel symbol of the first kind, and is defined by

$$[jk,\ i] \equiv \frac{1}{2}\left(\frac{\partial J_{ij}}{\partial q^k} + \frac{\partial J_{ik}}{\partial q^j} + \frac{\partial J_{jk}}{\partial q^i}\right). \tag{2.3}$$

(For purposes of applying the summation convention this symbol can be thought of as having three subscripts.) It should be noted that these symbols are *not* tensors. For a geometric interpretation of the inertia matrix and the Christoffel symbols, see [1].

Since the metric tensor J_{ij} is positive definite, it is also invertible. The inverse of this matrix is denoted by J^{jk}, and we have the relationship

$$J_{ij}J^{jk} = \delta_i^k = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases}. \tag{2.4}$$

If (2.2) is multiplied by J^{mi} and summed over i we obtain the equation

$$\frac{d^2q^m}{ds^2} + \left\{ \begin{matrix} m \\ jk \end{matrix} \right\} \frac{dq^j}{ds}\frac{dq^k}{ds} = 0. \tag{2.5}$$

The symbol $\left\{ \begin{matrix} m \\ jk \end{matrix} \right\}$ is Christoffel symbol of the second kind, and is defined as $\left\{ \begin{matrix} m \\ jk \end{matrix} \right\} \equiv J^{mi}[jk, i]$. As a special case, consider ordinary Euclidean space with rectangular Cartesian coordinates. Then the metric tensor is just the identity matrix (or Kronecker delta) δ_{ij}, and the Christoffel symbols are zero. Reassuringly, (2.2) and (2.5) then reduce to $d^2q^m/ds^2 = 0$, the differential equations which describe straight lines.

Equation (2.5) is often written $(\delta/\delta s)(dq^m/ds) = 0$, where the operator $\delta/\delta s$ is called the *absolute derivative* with respect to s. The absolute derivative of a contravariant tensor of order one with respect to the scalar ϕ is defined as

$$\frac{\delta T^i}{\delta \phi} \equiv \frac{dT^i}{d\phi} + \left\{ \begin{matrix} i \\ jk \end{matrix} \right\} T^j \frac{dq^k}{d\phi}. \tag{2.6}$$

This derivative consists of the ordinary derivative followed by a term involving a Christoffel symbol. The absolute derivative of a tensor of order two has two terms involving Christoffel symbols, tensors of order three have three additional terms, and so forth. The absolute derivative of an invariant is just the invariant's ordinary derivative. It can be shown that the absolute derivative of a tensor is itself a tensor, unlike the ordinary derivative. However, in many ways the absolute derivative behaves as an ordinary derivative behaves; it obeys the same rules for derivatives of sums and products, and obeys something like the chain rule, i.e., $\delta A^i/\delta\phi = (\delta A^i/\delta\psi)(d\psi/d\phi)$.

A curve in a Riemannian space can be written parametrically like (1.1), i.e., $q^i = f^i(\lambda)$. The derivative of this position vector, $dq^i/d\lambda$, is the *tangent* to the curve. In particular, if λ is the line element s then it is the *unit tangent* to the curve. The absolute derivative of the unit tangent, $(\delta/\delta s)(dq^i/ds)$, is the *curvature vector*. Note that for a geodesic the curvature vector is zero. It can also be shown that, as in ordinary (Euclidean) space, the curvature vector is orthogonal to the unit tangent.

III. Problem Statement

The minimum time trajectory planning algorithms described in [10] give the time history of manipulator's position, velocity, and joint torques required for the minimum time traversal of a *given* geometric path. However, these algorithms give no firm indication of how to pick a geometric path. The chosen geometric path ideally should be that which avoids all obstacles and can be traversed with the minimum time. In conjunction with trajectory planning, the minimum time geometric path planning (MTGPP) problem can be stated as follows.

Problem MTGPP: Given the solution to the minimum time trajectory planning problem, choose the geometric path, or the functions f^i in (1.1), so as to minimize the traversal time.

We will take approaches to this problem which are totally different from the techique described in [12], which makes use of Pontryagin's maximum principle or the method described in [2]. Both these methods are very computationally expensive. First, the minimum time path will be determined for a restricted class of robotic manipulators using some geometric techniques. Second, a method will be proposed for generating approximate minimum time geometric paths for more general manipulators. Note that these solutions are derived for collision-free space motions.

We begin in the next section with a few dynamic properties of manipulators that are needed.

IV. Some Useful Dynamic Properties of Manipulators

In this section we will introduce some properties of manipulators which will prove to be useful later on. Most of these properties relate to the "inertia space" of the manipulator, i.e., that Riemannian space which has the manipulator's *inertia matrix* as its *metric tensor*. The dynamic equations of a manipulator can be derived from Lagrange's equations, and take the form

$$u_i = J_{ij}\dot{v}^j + [jk,\ i]v^j v^k + R_{ij}v^j + g_i \tag{4.1}$$

where u_i is the generalized force/torque applied to the ith joint, v^i is the generalized velocity of the ith joint, J_{ij} is the inertia matrix, R_{ij} is the viscous friction matrix, and g_i is the gravitational force on the ith joint. The summation convention has been used here, and all sums range from 1 to n for an n-jointed manipulator. It should also be noted that J_{ij}, R_{ij}, and g_i may in general be

functions of the generalized coordinates q^i. The symbol $[jk, i]$ is a Christoffel symbol of the first kind, as described in the previous section. Equation (4.1) can be written in terms of absolute derivatives as

$$u_i = J_{ij}\frac{\delta v^j}{\delta t} + R_{ij}v^j + g_i. \tag{4.2}$$

For our purposes, we will define arc length ds by the quadratic form $ds^2 \equiv J_{ij}dq^i dq^j$. Since the kinetic energy of the manipulator is given by $K = 1/2J_{ij}(dq^i/dt)(dq^j/dt)$, it can be seen that the infinitesimal arc ds in this space is related to the kinetic energy of the manipulator by the formula $(ds/dt)^2 = 2K$.

The dynamic equations may now be expressed in terms of the arc length s and the time derivatives of s. We have, since the absolute derivative obeys the chain rule,

$$u_i = J_{ij}\frac{\delta v^j}{\delta s}\frac{ds}{dt} + R_{ij}v^j + g_i. \tag{4.3}$$

Using the relationship $v^j = (dq^j/ds)(ds/dt)$, then

$$u_i = J_{ij}\frac{\delta}{\delta s}\left(p^j\frac{ds}{dt}\right)\frac{ds}{dt} + R_{ij}p^j\frac{ds}{dt} + g_i \tag{4.4}$$

where $p^j \equiv dq^j/ds$ is the unit tangent to the manipulator's path. But, applying the product rule and using the fact that the absolute derivative of a scalar is just the ordinary derivative

$$\frac{\delta}{\delta s}\left(p^j\frac{ds}{dt}\right) = \frac{\delta p^j}{\delta s}\frac{ds}{dt} + p^j\frac{d}{ds}\left(\frac{ds}{dt}\right). \tag{4.5}$$

Plugging this into the dynamic equations (4.4),

$$u_i = J_{ij}\frac{\delta p^j}{\delta s}\left(\frac{ds}{dt}\right)^2 + J_{ij}p^j\frac{d^2s}{dt^2} + R_{ij}p^j\frac{ds}{dt} + g_i \tag{4.6}$$

where we have used the identity $(ds/dt)(d/ds)(ds/dt) \equiv (d/dt)(ds/dt) \equiv d^2s/dt^2$.

It is interesting to consider the form of the last equation. The left-hand side consists of externally applied forces. There are four terms on the right-hand side: a term proportional to the square of the velocity, a term proportional to the acceleration, a viscous friction term which is proportional to the velocity, and a gravitational term which is a function only of position. The first two terms are of particular interest. They are just the Coriolis and tangential acceleration terms, respectively. The Coriolis term is just the (vector) curvature of the path multiplied by the square of the speed, and so has a form analogous to the familiar mv^2/r term encountered in uniform circular motion. The second term, likewise, looks like the classical ma term one sees in one-dimensional Newtonian mechanics. The most important fact to note is that it is clear from this form of the dynamic equations that the Coriolis terms result directly from the curvature of the path in the manipulator's inertia space.

The work W done on the manipulator is

$$W = \int u_i dq^i = \int u_i\frac{dq^i}{ds}\,ds = \int u_i p^i\,ds. \tag{4.7}$$

Plugging in the expression for u_i from (4.7),

$$W = \int\left[J_{ij}p^i\frac{\delta p^j}{\delta s}\frac{ds}{dt} + J_{ij}p^i p^j\frac{d}{ds}\left(\frac{ds}{dt}\right)\right]\frac{ds}{dt}\,ds + \int R_{ij}p^i p^j\frac{ds}{dt}\,ds + \int g_i p^i\,ds. \tag{4.8}$$

Using the facts that the curvature vector $\delta p^j/\delta s$ is orthogonal to the unit tangent p^j and that p^i is a unit vector, i.e., that $J_{ij}p^i p^j = 1$, (4.8) transforms to

$$W = \int\left(\frac{ds}{dt}\right)\frac{d}{ds}\left(\frac{ds}{dt}\right)ds + \int R_{ij}p^i p^j\frac{ds}{dt}\,ds + \int g_i p^i\,ds = \frac{1}{2}\left(\frac{ds}{dt}\right)^2 + \int R_{ij}p^i p^j\frac{ds}{dt}\,ds + \int g_i p^i\,ds. \tag{4.9}$$

The power consumed by the manipulator at any given time is just

$$P \equiv \frac{dW}{dt} = \frac{ds}{dt}\frac{d^2s}{dt^2} + R_{ij}p^i p^j\left(\frac{ds}{st}\right)^2 + g_i p^i\frac{ds}{dt}. \tag{4.10}$$

V. Minimum Time Geometric Path Planning

As was previously pointed out, use of the maximum principle for solving the MTGPP problem is practically impossible. Alternative approaches must be sought. In this section we will develop three methods for generating geometric paths. For the first two we use energy methods to derive a lower bound on path traversal times, and for the third method we use the velocity limits derived in [10].

First it will be shown that geodesics in inertia space, i.e., solutions of the differential equations $\delta/\delta s(dq^i/ds) = 0$, are the optimal solutions to the MTGPP problem under some restricted conditions. Although the conditions required in the special case are not met by realistic manipulators, the proof does provide a simple illustration of the method used here for obtaining lower bounds on traversal time; the curves which minimize the lower bound for the more general case can then be found using essentially the same technique used in the special case, giving an absolute lower bound on the time required to move from one point to another. Then the use of the derived traversal time bounds and the velocity limits derived in [10] are used to find approximations to minimum time paths.

A. A Special Case

It will now be shown that if a manipulator has no friction terms and no gravitational terms and the limitations on the joint torques consist only of limits on the total power supplied to (or taken from) the manipulator, then the minimum time geometric paths are geodesics in inertia space. Formally, we have the following theorem.

Theorem 1: If a manipulator is frictionless and has zero gravitational terms, i.e., $R_{ij} = 0$ and $g_i = 0$ in the dynamic equations (4.1), and the only restrictions on the torques applied to the manipulator arise from constant, symmetric limits on the total power supplied to (or taken from) the manipulator, then the minimum-time geometric path between any two configurations of the manipulator is a *geodesic* in inertia space provided that the initial and final velocities are zero.

Proof: Under the stated conditions the dynamic equations for the manipulator become

$$u_i = J_{ij}\dot{v}^j + [jk,\ i]v^j v^k = J_{ij}\frac{\delta p^j}{\delta s}\left(\frac{ds}{dt}\right)^2 + J_{ij}p^j\frac{d^2s}{dt^2}. \tag{5.1}$$

The total power sunk or sourced by the manipulator is limited by symmetrical constant bounds, i.e., $-P_{\max} \le P \le P_{\max}$. Then by (4.10), applying the constant maximum power gives

$$P = P_{\max} = \frac{ds}{dt}\frac{d^2s}{dt^2} = \mu\frac{d\mu}{dt} \tag{5.2}$$

where $\mu \equiv ds/dt$. (Note that the gravitational and friction terms have been dropped from (4.10) for the special case.) Solving the differential equation (5.2) gives

$$\frac{1}{2}\mu^2 = tP_{\max}, \text{ or } s = \frac{2}{3}\sqrt{2P_{\max}}\,t^{3/2} \tag{5.3}$$

since the manipulator starts at rest.

Obviously, minimizing the traversal time for a given path requires that we maximize the "velocity" ds/dt. This in turn requires that the power P be maximized. Therefore, the maximum distance s which can be traveled in time t is given by (5.3).

Looking now at the end of the curve, we wish to have zero velocity at the end of the motion. Again, since we wish to minimize the traversal time, we want to stop as quickly as possible, which requires that we *drain* energy from the system as fast as possible. Applying the minimum power, we have $\mu(d\mu/dt) = -P_{\max}$. Solving this equation gives

$$\frac{1}{2}\mu^2 = P_{\max}(T-t), \text{ or } s = S - \frac{2}{3}\sqrt{2P_{\max}}\,(T-t)^{3/2} \tag{5.4}$$

where S is the total "length" of the curve which is to be traversed and T is the (unknown) time when the destination point is reached.

At some point in the middle of the curve there must be a switch from acceleration to deceleration. Let the time, distance, and velocity at this point be denoted by t_s, s_s and μ_s, respectively. Then (5.3) and (5.4) must give identical results at the switching point, so we have $\mu_s^2 = 2P_{\max}t_s = 2P_{\max}(T - t_s)$, and

$$s_s = S - \frac{2}{3}\sqrt{2P_{\max}}\,(T-t_s)^{3/2} = \frac{2}{3}\sqrt{2P_{\max}}\,t_s^{3/2}. \tag{5.5}$$

If we eliminate t_s from these equations, we can express T in terms of S. The resulting equation is $T = (9/4P_{\max})^{1/3}S^{2/3}$. The total time T increases monotonically with S, so minimum distance in inertia space is, in this case, equivalent to minimum time. Therefore, the geodesic, being the curve of shortest "distance" between any two points, is the optimal geometric path. ■

The conditions under which this proof of optimality applies, namely that friction and gravity be zero, are not realistic. The proof of optimality depends on the absence of gravitational and friction terms because the presence of such terms makes the power supplied to the manipulator a function of position rather than a function only of the kinetic energy of the manipulator. The requirement that the joint torques/forces be only constrained by a total power limit for the entire manipulator is also unrealistic. However, it is possible in practice to obtain bounds on the total available power for the more general case; in the next subsection, such bounds will be used to find bounds on traversal times.

B. Traversal Time Bounds

In this subsection, we show how energy methods similar to those used in the previous subsection may be used to obtain lower bounds on the time required to move from one point in the robot's workspace to another. We start with (4.10), the formula for the power supplied to the manipulator, namely

$$P_t = \frac{ds}{dt}\frac{d^2s}{dt^2} + \boldsymbol{R}_{ij}\boldsymbol{p}^i\boldsymbol{p}^j\left(\frac{ds}{dt}\right)^2 + \boldsymbol{g}_i\boldsymbol{p}^i\frac{ds}{dt} \tag{5.6a}$$

where P_t is the total power supplied to the robot.

If we write $P_k = (ds/dt)(d^2s/dt^2)$, $P_f = \boldsymbol{R}_{ij}\boldsymbol{p}^i\boldsymbol{p}^j(ds/dt)^2$, and $P_g = \boldsymbol{g}_i\boldsymbol{p}^i ds/dt$, then we have

$$P_k = P_t - P_f - P_g. \tag{5.6b}$$

We will find bounds on P_k by finding bounds on P_t, P_f, and P_g.

Before computing these bounds, we need the following result.

Lemma 1: The 2-norm of the vector $\boldsymbol{p}$ is bounded by

$$\frac{1}{\sqrt{\lambda_{\max}(\boldsymbol{J})}} = \frac{1}{f_M(\boldsymbol{J})} \le \|\boldsymbol{p}\|_2 \le \frac{1}{f_m(\boldsymbol{J})} = \frac{1}{\sqrt{\lambda_{\min}(\boldsymbol{J})}} \tag{5.7}$$

where $f_m(\boldsymbol{J}) \equiv \min_{p\neq 0}\sqrt{\boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p}/\boldsymbol{p}^T\boldsymbol{p}}$, $f_M(\boldsymbol{J}) \equiv \max_{p\neq 0}\sqrt{\boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p}/\boldsymbol{p}^T\boldsymbol{p}}$ (the quadratic form $\boldsymbol{J}_{ij}\boldsymbol{p}^i\boldsymbol{p}^j$ has been written in vector form as $\boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p}$), $\lambda_{\min}(\boldsymbol{J})$ is the smallest eigenvalue of $\boldsymbol{J}$ for *all* positions $\boldsymbol{q}$, and $\lambda_{\max}(\boldsymbol{J})$ is defined similarly.

Proof: Since $\boldsymbol{p}^i$ is a unit vector in inertia space, we have $\boldsymbol{J}_{ij}\boldsymbol{p}^i\boldsymbol{p}^j = \boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p} = 1$. Then we have

$$1 = \boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p} = \frac{\boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p}}{\boldsymbol{p}^T\boldsymbol{p}}\,\boldsymbol{p}^T\boldsymbol{p} \ge \boldsymbol{p}^T\boldsymbol{p}\,\min_{p\neq 0}\left[\frac{\boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p}}{\boldsymbol{p}^T\boldsymbol{p}}\right] = f_m^2(\boldsymbol{J})\|\boldsymbol{p}\|_2^2.$$

Likewise, we have

$$1 = \boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p} = \frac{\boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p}}{\boldsymbol{p}^T\boldsymbol{p}}\,\boldsymbol{p}^T\boldsymbol{p} \le \boldsymbol{p}^T\boldsymbol{p}\,\max_{p\neq 0}\left[\frac{\boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p}}{\boldsymbol{p}^T\boldsymbol{p}}\right] = f_M^2(\boldsymbol{J})\|\boldsymbol{p}\|_2^2.$$

Since $\boldsymbol{J}$ is positive definite and symmetric, its eigenvectors span R^n. Expanding $\boldsymbol{p}$ in terms of the eigenvectors of $\boldsymbol{J}$ shows that $f_m^2(\boldsymbol{J}) = \lambda_{\min}(\boldsymbol{J})$ and $f_M^2(\boldsymbol{J}) = \lambda_{\max}(\boldsymbol{J})$, which, combined with the above inequalities, proves the lemma. ■

We now compute bounds on the total applied power P_t. It will be assumed here that bounds on P_t arise from constant bounds on the joint torques and from constant bounds on the total applied power.[3] Then, we have

Lemma 2: If $ds/dt > 0$, then $\max\{P_{\min}, -\zeta(ds/dt)\} \le P_t \le \min\{P_{\max}, \zeta(ds/dt)\}$ where $P_{\min}$ and $P_{\max}$ are the minimum and maximum powers that can be supplied to the robot, $\zeta = \|\boldsymbol{u}_i^{\max}\|_2/\sqrt{\lambda_{\min}(\boldsymbol{J})}$, and $\|\boldsymbol{u}_i^{\max}\|_2$ is the maximum 2-norm of the torque vector.

Proof: By definition, we have $P_t = \boldsymbol{u}_i\boldsymbol{p}^i(ds/dt)$. The component of the torque in the direction of motion, $\boldsymbol{u}_i\boldsymbol{p}^i$, can be bounded by

$$|\boldsymbol{u}_i\boldsymbol{p}^i| \le \|\boldsymbol{u}_i^{\max}\|_2\|\boldsymbol{p}^i\|_2 \le \frac{\|\boldsymbol{u}_i^{\max}\|_2}{\sqrt{\lambda_{\min}(\boldsymbol{J})}}. \tag{5.8}$$

We, therefore, have

$$P_t \ge \max\left\{P_{\min}, \frac{-\|\boldsymbol{u}_i^{\max}\|_2}{\sqrt{\lambda_{\min}(\boldsymbol{J})}}\frac{ds}{dt}\right\} = \max\left\{P_{\min}, -\zeta\frac{ds}{dt}\right\} \tag{5.9a}$$

and

$$P_t \le \min\left\{P_{\max}, \frac{\|\boldsymbol{u}_i^{\max}\|_2}{\sqrt{\lambda_{\min}(\boldsymbol{J})}}\frac{ds}{dt}\right\} = \min\left\{P_{\min}, \zeta\frac{ds}{dt}\right\}. \tag{5.9b}$$

■

Lemma 3: P_f is bounded by $\phi\mu^2 \le P_f \le \phi'\mu^2$ where $\phi = \lambda_{\min}(\boldsymbol{R})/\lambda_{\max}(\boldsymbol{J})$ and $\phi' = \lambda_{\max}(\boldsymbol{R})/\lambda_{\min}(\boldsymbol{J})$.

Proof: By an argument similar to that used to derive bounds on $\|\boldsymbol{p}\|_2$, we have $\lambda_{\min}(\boldsymbol{R})\|\boldsymbol{p}\|_2^2 \le \boldsymbol{p}^T\boldsymbol{R}\boldsymbol{p} \le \lambda_{\max}(\boldsymbol{R})\|\boldsymbol{p}\|_2^2$ so that, by Lemma 1,

$$\frac{\lambda_{\min}(\boldsymbol{R})}{\lambda_{\max}(\boldsymbol{J})} \le \boldsymbol{p}^T\boldsymbol{R}\boldsymbol{p} \le \frac{\lambda_{\max}(\boldsymbol{R})}{\lambda_{\min}(\boldsymbol{J})}. \tag{5.10}$$

Multiplying by μ^2 gives the desired result. ■

Lemma 4: The gravitational energy contribution P_g is bounded by $-\psi(ds/dt) \le P_g \le \psi(ds/dt)$ where $\psi = \|\boldsymbol{g}_i\|_2/\sqrt{\lambda_{\min}(\boldsymbol{J})}$.

Proof: We have

$$|\boldsymbol{g}_i\boldsymbol{p}^i| \le \|\boldsymbol{g}_i\|_2\|\boldsymbol{p}^i\|_2 \le \frac{\|\boldsymbol{g}_i\|_2}{\sqrt{\lambda_{\min}(\boldsymbol{J})}} \tag{5.11}$$

by Lemma 1. Multiplying by ds/dt proves the lemma. ■

Using bounds derived in Lemmas 2–4, we are now in a position to obtain bounds on P_k.

[3] Bounds of this form can easily be found from motor saturation torques and from limits on power supply currents and voltages.

Lemma 5: If we define $\mu \equiv (ds/dt) > 0$,

$$\max\{P_{\min}, -\zeta\mu\} - \phi'\mu^2 - \psi\mu \le P_k \le \min\{P_{\max}, \zeta\mu\} - \phi\mu^2 + \psi\mu.$$

Proving this lemma is just a matter of plugging the bounds obtained in Lemmas 2–4 into (5.6b).

We can now determine maximum velocities, as was done in the previous subsection. We have the following theorem.

Theorem 2: If the initial and final velocities are zero, then

$$\mu \le \min\left\{\mu_m, \frac{\psi+\zeta}{\phi}(1-e^{-\phi t}), \frac{\psi+\zeta}{\phi'}(e^{\phi'(T-t)}-1)\right\}$$

where T is the traversal time of the path and $\mu_m = (\psi + \sqrt{\psi^2 + 4\phi P_{\max}})/2\phi$.

Proof: Consider two cases. In the first case, let P_k be limited by the joint torque bounds. Then we have $-\zeta\mu \le P_t \le \zeta\mu$, so that

$$-\phi'\mu^2 - (\psi+\zeta)\mu \le \mu\frac{d\mu}{dt} \le -\phi\mu^2 + (\psi+\zeta)\mu. \tag{5.12}$$

But then, since we are considering positive values of μ,

$$-\phi'\mu - (\psi+\zeta) \le \frac{d\mu}{dt} \le -\phi\mu + (\psi+\zeta). \tag{5.13}$$

We must have zero velocity at the beginning and end of the path. If the (as yet unknown) traversal time is T, then

$$\mu \le \frac{\psi+\zeta}{\phi}(1-e^{-\phi t}) \tag{5.14a}$$

and

$$\mu \le \frac{\psi+\zeta}{\phi'}(e^{\phi'(T-t)}-1). \tag{5.14b}$$

In the second case, limits are imposed by the total power limits, i.e., $P_{\min} \le P_t \le P_{\max}$. Then we have

$$P_{\min} - \phi'\mu^2 - \psi\mu \le \mu\frac{d\mu}{dt} \le P_{\max} - \phi\mu^2 + \psi\mu. \tag{5.15}$$

Again, we are only considering positive values of μ; since in general $P_{\min} < 0$, the lower bound in this inequality will always be less than zero. Therefore, the roots of the lower bound occur for negative values of μ, and we cannot place an upper bound on μ. On the other hand, the upper bound has a positive root. Since we are starting at $\mu = 0$, that value of μ for which $d\mu/dt$ goes to zero cannot be exceeded, and we have

$$\mu \le \mu_m = \frac{\psi + \sqrt{\psi^2 + 4\phi P_{\max}}}{2\phi}. \tag{5.16}$$

Since inequalities (5.14) and (5.16) must all be met, the theorem follows. ■

To find lower bounds on traversal times, consider a manipulator, call it the *super-manipulator*, for which the constraints on joint torques are such that (5.14a), (5.14b), and (5.16) apply. Then the super-manipulator has limits only on the 2-norm of the tangential component of the torque vector and on the total kinetic energy. Since these constraints apply for the original manipulator, the old manipulator's realizable torques are a subset of the super-manipulator's, so that the super-manipulator can do anything that the original manipulator can do. Thus, any path can be traversed by the super-manipulator at least as quickly as the old manipulator could traverse it. Finding the minimum traversal time for the super-manipulator therefore gives a lower bound on the traversal time for the original manipulator.

Finding the lower bound on the traversal time T for the super-manipulator is simple. It is just a matter of finding a value of T such that the area under the velocity versus time curve is equal to the geodesic distance S between the initial and final points. Formally, we have the following.

Theorem 3: Let the times t_1 and t_2 be given by

$$t_1 = \frac{1}{\phi}\log\left[\frac{\psi+\zeta}{\psi+\zeta-\phi\mu_m}\right] \quad \text{and} \quad t_2 = T - \frac{1}{\phi'}\log\left[\frac{\psi+\zeta+\phi'\mu_m}{\psi+\zeta}\right]. \tag{5.17}$$

If t_1 and t_2 are both real and $t_1 \le t_2$, then the minimum traversal time T for the super-manipulator can be found by solving the equation

$$S = \frac{\mu_m}{\phi'} - \frac{\mu_m}{\phi} + \mu_m T - \frac{\mu_m}{\phi'}\cdot\log\left[\frac{\psi+\zeta+\phi'\mu_m}{\psi+\zeta}\right] - \frac{\mu_m}{\phi}\log\left[\frac{\psi+\zeta}{\psi+\zeta-\phi\mu_m}\right]. \tag{5.18}$$

If t_1 is not real or $t_1 > t_2$, then T can be found by solving the simultaneous equations

$$\frac{\psi+\zeta}{\phi}(1-e^{-\phi t_s}) = \frac{\psi+\zeta}{\phi'}(e^{\phi'(T-t_s)}-1) \tag{5.19a}$$

$$\frac{\psi+\zeta}{\phi}\left[t_s - \frac{1}{\phi}(1-e^{-\phi t_s})\right] = S - \frac{\psi+\zeta}{\phi'}\left[\frac{1}{\phi'}(e^{\phi'(T-t_s)}-1) - (T-t_s)\right]. \tag{5.19b}$$

Proof: Finding the distance traveled, the area under the velocity versus time curve, requires that we consider the two cases described above, which correspond to 1) the case in which the velocity limit μ_m is reached, and 2) the case where it is not. Case 1) is relatively simple. First, we need to know the points where the curves described in (5.14a) and (5.14b) reach the limiting velocity μ_m. These times may be obtained from (5.14a) and (5.14b) by setting $\mu = \mu_m$ and solving for t. These times are just t_1 for (5.14a) and t_2 for (5.14b). Then the area under the velocity versus time curve will be given by

$$\begin{aligned} S &= \int_0^{t_1}\frac{\psi+\zeta}{\phi}(1-e^{-\phi t})\,dt + \int_{t_1}^{t_2}\mu_m\,dt \\ &\quad + \int_{t_2}^{T}\frac{\psi+\zeta}{\phi'}(e^{\phi'(T-t)}-1)\,dt \\ &= \frac{\mu_m}{\phi'} - \frac{\mu_m}{\phi} + \mu_m T - \frac{\mu_m}{\phi'}\log\left[\frac{\psi+\zeta+\phi'\mu_m}{\psi+\zeta}\right] \\ &\quad - \frac{\mu_m}{\phi}\log\left[\frac{\psi+\zeta}{\psi+\zeta-\phi\mu_m}\right]. \end{aligned} \tag{5.20}$$

Since S is linear in T, determining T is easy.

In case 2), we have a single switching time t_s. We may match positions and velocities as was done in the special case in the previous subsection, giving the equations

$$\mu_s = \frac{\psi+\zeta}{\phi}(1-e^{-\phi t_s}) = \frac{\psi+\zeta}{\phi'}(e^{\phi'(T-t_s)}-1) \tag{5.21a}$$

$$s_s = \int_0^{t_s} \frac{\psi+\zeta}{\phi}(1-e^{-\phi t})\,dt$$

$$= S - \int_{t_s}^{T} \frac{\psi+\zeta}{\phi'}(e^{\phi'(T-t)}-1)\,dt$$

$$= \frac{\psi+\zeta}{\phi}\left[t_s - \frac{1}{\phi}(1-e^{-\phi t_s})\right]$$

$$= S - \frac{\psi+\zeta}{\phi'}\left[\frac{1}{\phi'}(e^{\phi'(T-t_s)}-1) - (T-t_s)\right]. \quad (5.21b)$$

These are just the equations given in the statement of the theorem. ■

Unfortunately, (5.21) cannot be solved for t_s in closed form. However, we can still use these equations to prove that T increases monotonically with S, and thus prove that the optimal path for the super-manipulator is a geodesic. This being known, the geodesic distance S between the initial and final points can be calculated, and (5.21b) can be solved numerically.

Theorem 4: The minimum traversal time T for the super-manipulator increases monotonically with the geodesic length S of the traversed path.

Proof: To prove that T increases monotonically with S, we will show that $dS/dT > 0$. If case 1) of Theorem 3 holds, then the result is obvious. Case 2) is slightly more complicated. First, we differentiate (5.21a) and (5.21b) with respect to the switching time t_s, giving

$$e^{-\phi t_s} = e^{\phi'(T-t_s)}\left(\frac{dT}{dt_s} - 1\right) \quad (5.22a)$$

$$\frac{\psi+\zeta}{\phi}(1-e^{-\phi t_s}) = \frac{dS}{dt_s} - \frac{\psi+\zeta}{\phi'}\left(\frac{dT}{dt_s}-1\right)(e^{\phi'(T-t_s)}-1). \quad (5.22b)$$

Solving (5.22a) for dT/dt_s, plugging into (5.22b), solving (5.22b) for dS/dt_s, and dividing dS/dt_s by dT/dt_s gives

$$\frac{dS}{dT} = \frac{\psi+\zeta}{\phi\phi'(1+e^{-\phi t_s}e^{-\phi'(T-t_s)})} \cdot [\phi'(1-e^{-\phi t_s}) + \phi e^{-\phi t_s}(1-e^{-\phi'(T-t_s)})] \quad (5.23)$$

which is greater than zero. ■

If the actual lower bound is required, then (5.21a) may be solved for T. To do this, we may make use of (5.17) and (5.18). The maximum velocity μ_m can be varied in these equations until $t_1 = t_2 = t_s$; then this value of t_s can be used in (5.21a), which can be solved for T.

C. Approximate Minimum Time Paths

In this subsection we consider two methods for generating geometric paths which are approximately minimum time. The first method uses the traversal time bounds derived in the previous section and the second method uses the velocity bounds derived in [10].

First, consider the lower bounds on traversal time. If these bounds are good estimates of the actual traversal times, then minimizing the lower bound should approximately minimize the traversal time. Since the lower bound increases monotonically with the geodesic length S of the traversed curve, geodesics (minimum-length curves) must minimize the lower bound. The "near optimal" paths may then be determined by solving the differential equations for a geodesic, namely $(\delta/\delta s)(dq^i/ds) = 0$. This method of generating near-minimum time paths can be applied to most practical robots; however, it places no penalties on forces which are orthogonal to the traversed path, so that path curvature is not penalized. If any further constraints are applied which force the introduction of curvature terms, then ignoring the magnitude of the curvature terms could make the lower bound a poor estimate of the actual traversal time, causing a poor choice of path. The minimization of lower bounds leads to the selection of shortest-distance paths in inertia space, which could have corners at which the manipulator must come to a complete stop. Path segments of high curvature also slow the manipulator down. Thus, it is necessary to strike a compromise between curves of shortest distance and curves of smallest curvature. (This naturally leads to the second method of generating near-minimum time geometric paths.)

In order to reach such a compromise, we choose as an objective function the product of the length of the curve and some measure of the total curvature. This, of course, requires some quantitative measure of both curvature and distance in an n-dimensional space where n is the number of manipulator joints. One obvious measure of total curvature is the reciprocal of the maximum velocity, as computed in [10]. If the path is expressed in terms of an arbitrary parameter λ, then the expression

$$\int_0^{\lambda_{\max}} \frac{d\lambda}{\mu_{\max}(\lambda)} \quad (5.24)$$

would appear to be a good choice, where $\mu \equiv \dot{\lambda}$ and $\mu_{\max}(\lambda)$ is the velocity limit at position λ. This expression is *independent* of the parameterization chosen, and increases both as the length of the curve increases and as the curvature increases.

In order to use (5.24), the value of the maximum velocity $\mu_{\max}(\lambda)$ is required. In [10] we have derived this bound in terms of the manipulator's torque bounds and its dynamic equations. The set of admissible accelerations $\dot{\mu}$ is given by a set of inequalities of the form

$$u_i^{\min} \le M_i\dot{\mu} + Q_i\mu^2 + R_i\mu + S_i \le u_i^{\max} \quad (5.25)$$

where $M_i \equiv J_{ij}(df^j/d\lambda)$, $Q_i \equiv J_{ij}(d^2f^j/d\lambda^2) + [jk, i](df^j/d\lambda)(df^k/d\lambda)$, $R_i \equiv \boldsymbol{R}_{ij}(df^j/d\lambda)$, and $S_i \equiv \boldsymbol{g}_i$. For a given position λ and velocity μ, these inequalities give a range of accelerations $\dot{\mu}$, and so may be thought of as assigning upper and lower acceleration bounds to each point (λ, μ) in the phase plane. Since these inequalities must hold for all joints of the manipulator, the acceleration must fall between the greatest of the lower acceleration bounds and the least of the upper bounds. When one of the upper acceleration bounds is smaller than one of the lower acceleration bounds for some phase joint (λ, μ), there are no accelerations which will keep the manipulator on the desired path. Thus, the acceleration bounds generate restrictions on the velocities at the phase points which can be encountered during a traversal of the path. These relationships can be thought of as assigning velocity limits to a given position λ.

Now consider a frictionless manipulator, i.e., one for which the quantities R_i are zero. Also assume that at every point on the path the manipulator is capable of stopping and holding its position. Then we have $u_i^{\min} \le S_i \le u_i^{\max}$ at all points on the path. (This will hereafter be referred to as the "strong manipulator assumption.") If the parameter λ is defined to be the arc length s in inertia space, then Q_i is just the inertia matrix J_{ij} multiplied by the curvature vector $\delta p^j/\delta s$.

If the path chosen is a geodesic, then the curvature vector is zero, and hence $Q_i \equiv 0$. Then the inequality (5.25) reduces to $u_i^{\min} \le M_i\dot{\mu} + S_i \le u_i^{\max}$, which is independent of the velocity μ, and by the strong manipulator assumption is satisfied identically for $\dot{\mu} = 0$. But if the bounds on $\dot{\mu}$ are independent of μ, there can be no velocity limits; in other words; $\mu_{\max}(\lambda) = \infty$, so that the integrand of (5.24) is zero. Thus, in this case the optimal solution coincides with that obtained from minimizing traversal times.

It may appear at first that the geodesic, since it maximizes velocity bounds, must be the true minimum time path. However, as shown in [10] the manipulator must meet acceleration as well as

velocity constraints. It would then be expected that along the optimal geometric path the maximum acceleration would be maximized during an accelerating portion of the path and the minimum acceleration would be minimized during a decelerating portion. This does not happen along a geodesic, but a similar phenomenon occurs: the acceleration bounds "spread out." To see this, note that velocity limits occur because the acceleration bounds become very close. Since the velocity bounds have been eliminated by choosing a geodesic as the path, the acceleration bounds must never get close. Hence, maximizing velocity bounds also gives a large range of accelerations to choose from. This would lead one to expect that geodesics are good, if not optimal, choices for geometric paths.

In summary, we have two criteria for selecting near-minimum time geometric paths for a manipulator, one based on the minimization of a lower bound on the manipulator's traversal time and the other based on the minimization of the product of the path's length and its curvature. In either case, when the near-optimal path is determined, the path is found to be a geodesic in inertia space. This geodesic can be constructed by solving a set of differential equations and applying appropriate boundary conditions.

One comment on calculating the geodesics is in order. Finding the geodesic which passes through two points is a two-point boundary value problem. Solving the problem using Pontryagin's maximum principle also gives rise to a two-point boundary value problem, so one may legitimately ask what the advantage of using geodesics is. The important difference is that the boundary conditions for finding the geodesic are easier to apply than are the boundary conditions which arise when using the maximum principle. The geodesics may be found using shooting methods, which involve guessing the initial direction of the geodesic, and refining the guess until the curve comes sufficiently close to the desired endpoint.

VI. Examples

To demonstrate the utility of the solutions described above, the traversal times for various geometric path have been calculated, using the method of [10], for the Bendix PACS arm and for the Stanford manipulator. The PACS arm is cylindrical in configuration, and is driven by fixed-field DC motors. Only the dynamics of the first three joints are considered here (see Fig. 2). We construct three paths, a straight line, a geodesic, and a joint-interpolated curve. (The joint-interpolated curve has the form $q^i = q_s^i + \lambda(q_f^i - q_s^i)$, where $0 \le \lambda \le 1$ and q_s^i and q_f^i are the points at which the curve starts and finishes.)

Both construction of geodesics and trajectory planning require that the dynamic equations (4.1) of the robot be known. In particular, the inertia matrix and Coriolis coefficients are needed in order to construct geodesics. For the first three joints of the manipulator, the inertia matrix takes the form

$$J_{ij} = \begin{bmatrix} J_t - Kr + M_t r^2 & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & M_z \end{bmatrix} \tag{6.1}$$

where $q^1 = \theta$, $q^2 = r$, and $q^3 = z$. The constants M_t and M_z are the masses which the r and z axes must move. J_t is the moment of inertia around the θ axis when r is zero. The K term is present because the center of mass of the structure for the r joint does not coincide with the θ axis when r is zero. The values of J_t, K, M_t, and M_z are given in [9], along with friction coefficients and actuator characteristics. The nonzero Christoffel symbols of the first kind (Coriolis coefficients), found by differentiating J_{ij}, are

$$[12,\ 1] = [21,\ 1] = M_t r - \frac{K}{2} \tag{6.2a}$$

$$[11,\ 2] = \frac{K}{2} - M_t r. \tag{6.2b}$$

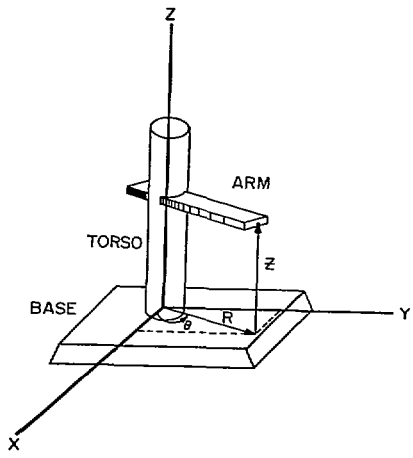


Fig. 2. Schematic diagram of the first three joints of the PACS robot.

The geodesics are solutions of the equations $0 = J_{ij}d^2q^j/ds^2 + [jk, i](dq^j/ds)(dq^k/ds)$. Plugging (6.1)–(6.2b) into this equation gives the equations of the geodesics as

$$0 = (J_t - Kr + M_t r^2)\frac{d^2\theta}{ds^2} + (2M_t r - K)\frac{dr}{ds}\frac{d\theta}{ds} \tag{6.3a}$$

$$0 = M_t \frac{d^2r}{ds^2} + \left(\frac{K}{2} - M_t r\right)\left(\frac{d\theta}{ds}\right)^2 \tag{6.3b}$$

$$0 = M_z \frac{d^2z}{ds^2}. \tag{6.3c}$$

In addition, we have the normality condition

$$(J_t - Kr + M_t r^2)\left(\frac{d\theta}{ds}\right)^2 + M_t\left(\frac{dr}{ds}\right)^2 + M_z\left(\frac{dz}{ds}\right)^2 = 1. \tag{6.4}$$

It can be shown that the differential equations (6.3a)–(6.3c) can be solved in terms of elliptic integrals. In practice, however, it is simpler to solve them numerically.

The gravitational terms for this manipulator are particularly simple; the gravitational forces on the r and θ joints are zero, and the force on the z joint is $M_z g$.

Trajectory planning also requires knowledge of the robot's actuator characteristics. To determine actuator characteristics, consider a circuit consisting of the series connection of a voltage source, a resistance R^m, an inductance L, and an ideal motor, i.e., a device which generates a torque proportional to the current passing through it. The voltage source is the power supply, the resistance is the sum of the voltage source resistance and the motor winding resistance, and the inductance is the inductance of the motor windings.

It will be assumed here that the inductance L can be neglected. This frequently is the case for DC motors, since the electrical time constant of such systems is generally much shorter than the mechanical time constant. Given that the torque τ is proportional to the current, i.e., $\tau = k_m I$, it can be shown from conservation of power that the voltage V_m across the ideal motor is just $k_m\omega$, where ω is angular velocity. Since, if the motor is not in saturation, $\tau = k_m I$ and $I = (V_s - V_m)/R^m = (V_s - k_m\omega)/R^m$ where V^s is the source voltage, we can solve for torque in terms of voltage and angular velocity, giving $\tau = (k_m/R^m)V_s - (k_m^2/R^m)\omega$. Assuming the power supply has constant voltage limits of $V^{\min}$ and $V^{\max}$, this gives torque limits of

$$\frac{k^m}{R^m}V^{\min} - \frac{(k^m)^2}{R^m}\omega \le \tau \le \frac{k^m}{R^m}V^{\max} - \frac{(k^m)^2}{R^m}\omega. \tag{6.5}$$

In addition, at some point the motor saturates, with the result that

increasing the current through the motor has no effect on the torque. This yields two more (constant) torque limits, so we also require that $-\tau^{sat} \leq \tau \leq \tau^{sat}$. Taking the gear ratio k^g into account, this gives torque limits of

$$u_i^{min} = \max\left(-\frac{\tau_i^{sat}}{k_i^g}, \frac{k_i^m}{R_i^m k_i^g} V_i^{min} - \frac{(k_i^m)^2}{R_i^m (k_i^g)^2} \frac{dq^i}{d\lambda} \mu\right) \quad (6.6a)$$

and

$$u_i^{max} = \min\left(\frac{\tau_i^{sat}}{k_i^g}, \frac{k_i^m}{R_i^m k_i^g} V_i^{max} - \frac{(k_i^m)^2}{R_i^m (k_i^g)^2} \frac{dq^i}{d\lambda} \mu\right). \quad (6.6b)$$

A trajectory planner for this robot was written in the C programming language and run under the UNIX[4] operating system on a VAX-11/780.[5] The trajectory planner was used to generate trajectories for a straight line, a geodesic, and a joint interpolated curve, each curve connecting the same endpoints. Phase plane plots (plots of the speed μ versus position λ), plots of position q^i versus time, and plots of motor voltage versus time are shown in Figs. 3(a)–5(c). Fig. 3(a)–(c) is for the straight line, Fig. 4(a)–(c) is for the joint interpolated curve, and Fig. 5(a)–(c) is for the geodesic. The traversal times for these paths are 1.782, 1.796, and 1.588 s, respectively, showing that the geodesic does indeed have the shortest traversal time.

When the robot is driven along a given path in minimum time, one or more of the actuators will be driven to its limit. For the straight line path, the r joint is driven at its maximum or minimum voltage except for two short intervals when the θ joint is saturated. For the joint interpolated curve, the r joint motor voltage is always the limiting factor. For the geodesic, the r joint is saturated most of the time, but both the θ and z joints are driven to their limits at one time or another. The geodesic seems to distribute the workload more *evenly* among the joints than the other two curves do.

Table I presents traversal times for the three types of paths of the PACS arm with various initial and final configurations. The geodesic again shows comparatively small traversal times except for the sixth row, where only z and y coordinates in the initial and final configurations are switched. In such a case, only the θ joint needs to move, since values of both the r $(=\sqrt{x^2 + y^2})$ and z joints do not change on this path. That is, there is not a sufficient number of joints to distribute the workload.

The Stanford manipulator has been extensively studied by a number of researchers and consists of five revolute and one prismatic joints. See [8] for a detailed description of its kinematics and dynamics. Using the dynamic equations in [8] and the data in Table II, we have computed the traversal times for the three types of paths for the Stanford manipulator with various initial and final configurations. The results are given in Table III, where the geodesic again exhibits small traversal times compared to the other two types of paths.

VII. Conclusions

Two methods (excluding the special case) have been proposed for finding geometric paths which allow a robotic manipulator to move from one point to another in minimum time or approximately minimum time; if obstacle avoidance is not a consideration, both methods yield the same result. While these methods do not directly address the problem of obstacle avoidance, they do demonstrate that the problem of choosing minimum time paths is not simple and, in particular, they show that minimum time is not in general equivalent to minimum Cartesian distance.

Equation (5.24) provides a means of evaluating the "goodness" of any given path without actually calculating the path's traversal time. If several paths can be found which avoid

[4] UNIX is a trademark of AT&T Bell Laboratories.
[5] VAX is a trademark of Digital Equipment Corporation.

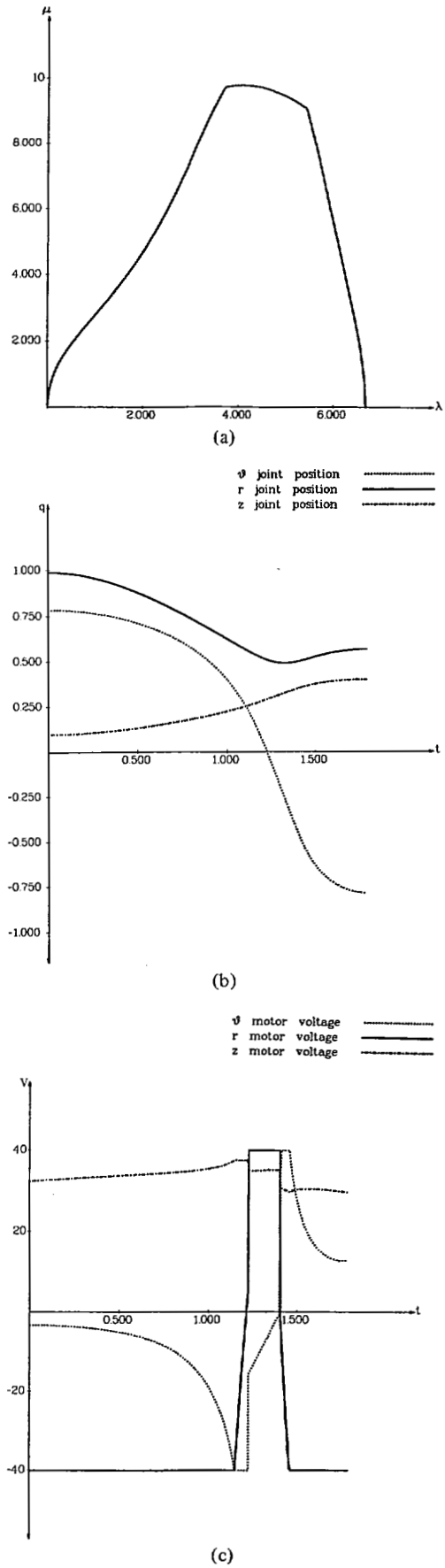


Fig. 3. (a) Phase plane plot for straight line. (b) Joint position versus time for straight line. (c) Motor voltages versus time for straight line.

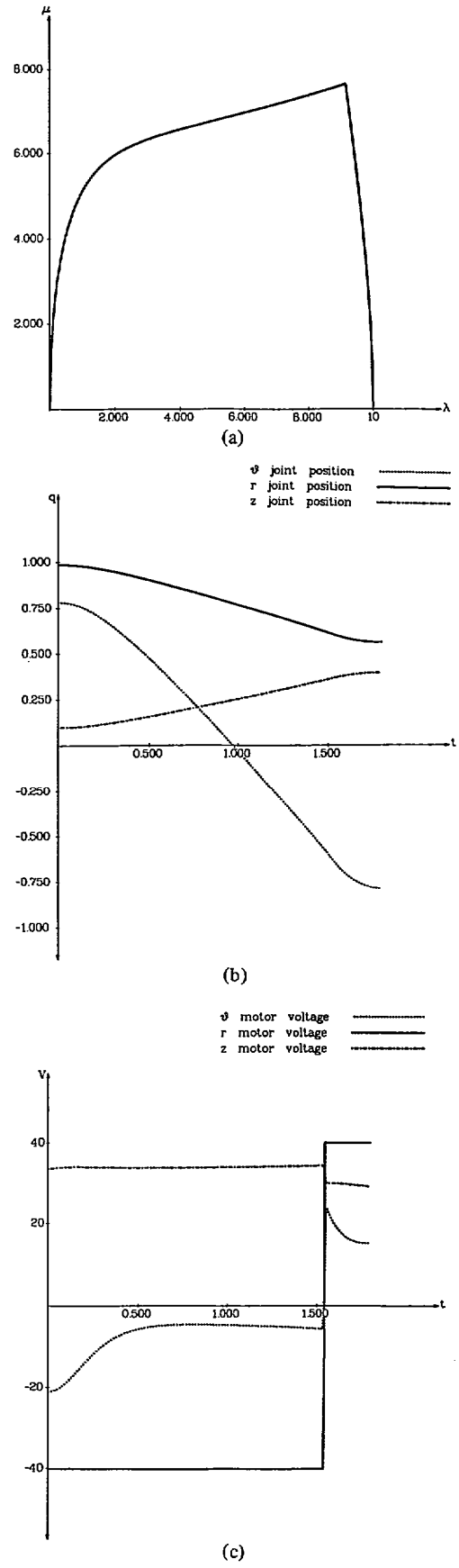


Fig. 4. (a) Phase plane plot for joint-interpolated path. (b) Joint position versus time for joint-interpolated path. (c) Motor voltages versus time for joint-interpolated path.

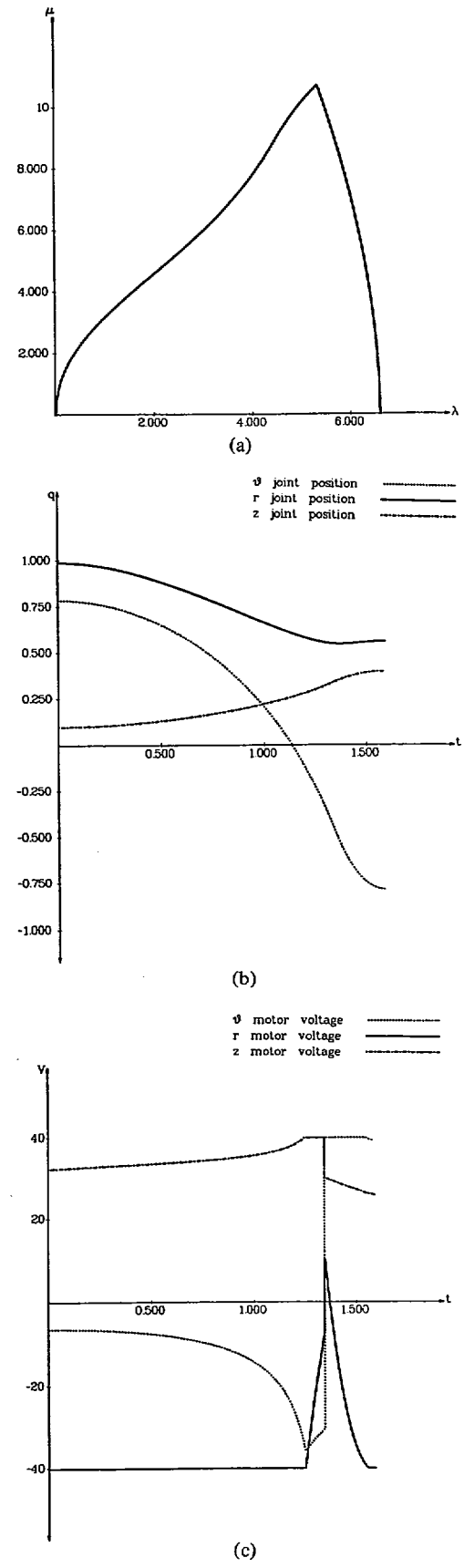


Fig. 5. (a) Phase plane plot for geodesic. (b) Joint position versus time for geodesic. (c) Motor voltages versus time for geodesic.

TABLE I
TRAVERSAL TIMES FOR THREE TYPES OF PATHS OF PACS ARM WITH VARIOUS ENDPOINTS

Endpoints		Curve Type		
From	To	Line	Geodesic	Joint Interpolated
(0.7,0.7,0.1)	(0.4,-0.4,0.4)	1.782	1.588	1.798
(-0.5,0.8,0.1)	(0.5,0.3,0.1)	1.758	1.533	1.716
(0.2,0.2,0.6)	(0.8,0.8,0.2)	1.927	1.919	1.927
(0.8,0.8,0.2)	(0.2,0.2,0.8)	1.927	1.921	1.927
(0.2,-0.2,0.2)	(0.7,0.7,0.4)	1.758	1.600	1.802
(0.6,0.2,0.6)	(0.2,0.6,0.2)	0.900	0.804	0.657
(0.8,0.6,0.5)	(0.1,0.7,0.7)	1.131	1.127	1.231
(0.5,-0.2,0.5)	(0.1,0.2,0.3)	1.340	1.223	1.322
(0.4,0.4,0.8)	(0.7,0.1,0.4)	0.756	0.727	0.781

TABLE II
DYNAMIC COEFFICIENTS AND ACTUATOR CHARACTERISTICS FOR STANFORD ARM

Parameter	Description	Value
$\tau_{\theta_1}^{sat}$	Saturation torque of θ_1 motor	0.918 Nt.-M.
$\tau_{\theta_2}^{sat}$	Saturation torque of θ_2 motor	3.036 Nt.-M.
$\tau_{d_3}^{sat}$	Saturation torque of d_3 motor	0.494 Nt.-M.
$V_{\theta_1}^{min}$	Lower voltage limit for θ_1 joint	-40 v.
$V_{\theta_2}^{min}$	Lower voltage limit for θ_2 joint	-40 v.
$V_{d_3}^{min}$	Lower voltage limit for d_3 joint	-40 v.
$V_{\theta_1}^{max}$	Upper voltage limit for θ_1 joint	40 v.
$V_{\theta_2}^{max}$	Upper voltage limit for θ_2 joint	40 v.
$V_{d_3}^{max}$	Upper voltage limit for d_3 joint	40 v.
$k_{\theta_1}^{g}$	Gear ratio for θ_1 drive	0.01
$k_{\theta_2}^{g}$	Gear ratio for θ_2 drive	0.01
$k_{d_3}^{g}$	Gear ratio for d_3 drive	0.005 Meters/radian
$k_{\theta_1}^{m}$	Motor constant for θ_1 joint	0.02 Nt.-M./amp
$k_{\theta_2}^{m}$	Motor constant for θ_2 joint	0.06 Nt.-M./amp
$k_{d_3}^{m}$	Motor constant for d_3 joint	0.01 Nt.-M./amp
$R_{\theta_1}^{m}$	Motor and power supply resistance, θ_1 joint	1 Ω
$R_{\theta_2}^{m}$	Motor and power supply resistance, θ_2 joint	1 Ω
$R_{d_3}^{m}$	Motor and power supply resistance, d_3 joint	1 Ω
k_{θ_1}	Friction coefficient of θ_1 joint	41.875 Kg.-$M.^2$/sec./rad.
k_{θ_2}	Friction coefficient of θ_2 joint	113.0 Kg.-$M.^2$/sec./rad.
k_{d_3}	Friction coefficient of d_3 joint	49.0 Kg./sec.

TABLE III
TRAVERSAL TIMES FOR THREE TYPES OF PATHS OF STANFORD ARM WITH VARIOUS ENDPOINTS

Endpoints		Curve Type		
From	To	Line	Geodesic	Joint Interpolated
(0.7,0.7,0.1)	(0.4,-0.4,0.4)	1.203	1.099	1.090
(0.5,0.5,0.1)	(0.5,-0.5,0.1)	1.343	1.011	1.030
(0.0,0.5,0.1)	(0.8,0.5,0.6)	2.204	2.199	2.209
(0.0,0.5,0.1)	(0.8,0.8,0.6)	2.733	2.728	2.737
(-0.2,0.4,0.1)	(0.2,0.2,0.6)	1.251	0.755	0.944
(0.9,0.1,0.1)	(0.1,0.7,0.8)	1.383	0.952	0.834
(0.2,0.3,0.3)	(0.6,0.6,0.6)	2.092	2.075	2.083
(0.8,0.8,0.2)	(0.2,0.2,0.8)	0.708	0.660	0.732
(0.2,0.6,0.8)	(0.5,0.2,0.2)	0.645	0.625	0.632

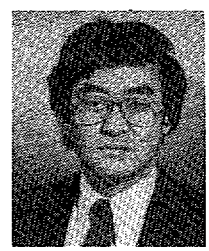
collisions with obstacles, then each one can be evaluated and the best one chosen on the basis of formula (5.24). This presumes that some method can be developed for generating collision-free paths quickly. It also presumes that at least some of the paths generated by the algorithm are reasonably close to the optimal path. But since minimization of the product of curvature and distance gives paths with short traversal times, some guidelines for generating paths are now available.

ACKNOWLEDGMENT

The authors are grateful to R. Throne of the University of Michigan for the computer simulation of the Stanford manipulator in Section VI.

REFERENCES

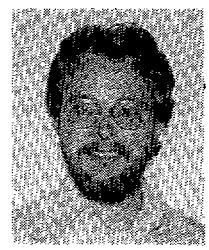
[1] H. Asada, "A geometrical representation of manipulator dynamics and its application," *ASME J. Dynam. Syst., Measurement Contr.*, vol. 105, pp. 131–135, Sept. 1983.
[2] E. G. Gilbert and D. W. Johnson, "Distance functions and their application to robot path planning in the presence of obstacles," *IEEE J. Robotics Automat.*, vol. RA-1, pp. 21–30, Mar. 1985.
[3] M. E. Kahn and B. Roth, "The near-minimum-time control of open-loop articulated kinematic chains," *ASME J. Dynam. Syst., Measurement, Contr.*, pp. 164–172, Sept. 1971.
[4] B. K. Kim and K. G. Shin, "Suboptimal control of industrial manipulators with a weighted minimum time-fuel criterion," *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 1–10, Jan. 1985.
[5] B. K. Kim and K. G. Shin, "Minimum-time path planning for robot arms and their dynamics," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 213–223, Mar./Apr. 1985.
[6] T. Lozano-Perez, "Spatial planning: A configuration space approach," M.I.T. Artificial Intelligence Lab., Cambridge, MA, A.I. Memo 605, Dec. 1980.
[7] J. Y. S. Luh and C. S. Lin, "Optimum path planning for mechanical manipulators," *ASME J. Dynam. Syst., Measurement, Contr.*, vol. 102, pp. 142–151, June 1981.
[8] R. P. C. Paul, *Robot Manipulators: Mathematics, Programming, and Control.* Cambridge, MA: M.I.T. Press, 1981.
[9] K. G. Shin and N. D. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," this issue, pp. 491–500.
[10] ——, "Minimum-time control of a robotic manipulator with geometric path constraints," *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 531–541, June 1985.
[11] J. L. Synge and A. Schild, *Tensor Calculus.* New York: Dover, 1978.
[12] A. Weinreb and A. E. Bryson, "Optimal control of systems with hard control bounds," in *Proc. 1985 Amer. Contr. Conf.*, pp. 1248–1252, June 1985.



Kang G. Shin (S'75–M'78–SM'83) received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1970, and both the M.S. and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, in 1976 and 1978, respectively.

From 1978 to 1982 he was an Assistant Professor at Rensselaer Polytechnic Institute, Troy, NY. He was also a Visiting Scientist at the U.S. Air Force Flight Dynamics Laboratory in the Summer of 1979 and at Bell Laboratories, Holmdel, NJ, in the Summer of 1980, where his work was concerned with distributed airborne computing and cache memory architecture, respectively. Since September 1982, he has been with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, where he is currently an Associate Professor. His current teaching and research interests are in the areas of distributed real-time and fault-tolerant computing, computer architecture, and robot control, planning, and programming.

Dr. Shin is a member of Sigma Xi, Phi Kappa Phi, and the Association for Computing Machinery.



Neil D. McKay was born in Albany, NY, on May 5, 1958. He received the B.S. degree in electrical engineering from the University of Rochester, Rochester, NY, in 1980, the M.S. degree in 1982 from Rensselaer Polytechnic Institute, Troy, NY, and the Ph.D. degree from the University of Michigan, Ann Arbor, in 1985.

Currently, he is with the Computer Science Department at General Motors Research Laboratories, Warren, MI.