

A DISCRETE PATH/TRAJECTORY PLANNER FOR ROBOTIC ARMS

H. H. TAN¹ and R. B. POTTS¹

(Received 2 June 1988; revised 12 December 1988)

Abstract

An interesting and challenging problem in robotics is the off-line determination of the minimum cost path along which an end effector should move from a given initial to a given final state. This paper presents a discrete minimum cost path/trajectory planner which provides a general solution and allows for a range of constraints such as bounds on joint coordinates, joint velocities, joint torques and joint jerks. To demonstrate the practicability and feasibility of the planner, simulation results are presented for the Stanford manipulator using three and then the full six of its degrees of freedom. Simulation runs with two-link planar arms are also presented to enable a comparison with previously published results.

1. Introduction

The off-line planning of the minimum cost, and in particular minimum time, movement for a robotic manipulator has been a problem which has attracted considerable attention for several years. It has proved convenient to distinguish two optimisation problems: the simpler trajectory planning problem in which the optimal motion of the robot end effector *along a given path* has to be determined, and the more difficult and comprehensive path planning problem in which the optimal geometric path together with the motion along it has to be determined. Various successful algorithms have been proposed for trajectory planning for a given path [2], [24], and recently the details of two efficient discrete fixed-path trajectory planners have been described [29], [30].

This paper is concerned with the minimum cost path/trajectory planning problem. In essence this requires the determination of the optimal geometric

¹Applied Mathematics Department, University of Adelaide, S.A.5000, Australia.

© Copyright Australian Mathematical Society 1989, Serial-fee code 0334-2700/89

path for the end effector together with the motion along it, given the initial and final configurations, and subject to additional constraints such as bounds on the magnitudes of joint coordinates, velocities, torques and jerks, as well as obstacle avoidance.

In an early attempt to solve the path/trajectory planning problem, Kahn and Roth [12] considered a three-degrees-of-freedom (DOF) case, with constraints on only the joint torques in addition to the usual terminal state constraints. They concluded, from applying the Pontryagin minimum principle to the complete nonlinear dynamic system, that the minimum time path for this restricted set of constraints requires bang-bang profiles for *all* joint torques in the *regular* case. Possible singular arcs for some joints might be present. Unfortunately, the determination of the switching times necessitated solving a computationally intractable nonlinear two-point boundary value problem and this forced Kahn and Roth to adopt a suboptimal approach by linearising the nonlinear dynamics. It was pointed out by Hollerbach [11], using the time-scaling property of robot dynamics, that the main assumption used to justify the linearisation was fundamentally incorrect. Snyder and Gruver [26] and Turner and Gruver [33] also simplified their nonlinear dynamic models in the design of two fixed-time minimum energy controllers. Lynch [14] simplified the dynamics by moving one joint at a time for his minimum time controller. His solution is not optimal as it takes more time than when all joints are allowed to move simultaneously. Kim and Shin [13] used an averaged dynamics method in their suboptimal weighted time/energy controller; the nonlinear dynamics is still not fully taken in account.

The Pontryagin minimum principle for the full nonlinear dynamics was used by Geering *et al.* [7] for obtaining minimum time paths for three robots, each with two degrees of freedom and a common robot geometry. For given terminal states and joint torque constraints, they used a shooting method and a parameter optimisation technique to solve the two-point boundary value problem, and they presented results for regular bang-bang control as well as for the singular case. Computational intractability prevented them from presenting results for robotic arms with more degrees of freedom.

The methods proposed by Scheinman and Roth [23] and Niv and Auslander [17] also take into account the full dynamics but assume either a bang-coast-bang or bang-bang form of control with a specified number of switching points. These assumptions on the form of joint torques are not generally valid for the true minimum time solution.

Gilbert and Johnson [8] used distance functions between potentially colliding parts to introduce obstacle avoidance in their optimal control approach to solving the problem. The computational cost was reported to be excessive.

The computational difficulty experienced by these control theoretic methods prompted an alternative approach which searches, in some systematic manner, for the minimum time path in a space of paths. The methods using this approach will be called *search-space* methods in this paper. Sahar and Hollerbach [22] claim to present a general solution of the minimum time path planning problem using joint space tessellation and dynamic scaling of robot dynamics [11], with optimal paths chosen by an exhaustive graph search technique. Each of the paths to be searched is assumed to have a trajectory which follows the bang-bang torque profile for a fixed path. Their results are not optimal, featuring only the torque for one joint bang-bang while the other is not saturated throughout the motion. For one of their test cases, we have verified with our discrete planner that the minimum time solution follows the regular case and thus is bang-bang in both torques. Sahar and Hollerbach confined their analysis to a two DOF manipulator and while they concluded that the implementation of their method might be possible for three degrees of freedom, the full six DOF case was ruled out because of the complexity of the search technique. Their claim to have presented a general solution to the problem seems not to be substantiated. Brown [4] also developed a planner based on state space tessellation and a graph search. He presumed a bang-coast-bang solution with fixed switching points. This assumption on the control makes his solutions non-optimal.

Rajan [20] took a different approach by representing the search space as a set of parameterised paths. Cubic splines are chosen for parameterisation on the assumption that the optimal path will be smooth. The algorithm given in [2] is used to find the minimum time trajectory for a chosen parameterised path with a gradient descent technique applied to the parameters of the path. Initially the path is represented by a single spline curve for each joint coordinate, and in the subsequent iterations, this is further subdivided by adding knot points and therefore increasing the number of spline curves and the number of parameters. The reported results only feature the bang-bang profiles true of the fixed-path case. An extension of this method to incorporate constraints on the joint excursion limits and obstacle avoidance was given by Dubowsky *et al.* [5]. They were able to obtain near minimum time paths for manipulators with up to six degrees of freedom.

Shin and McKay [25] developed a different method for determining and approximating minimum time geometric paths. A lower bound on the time required to move a manipulator arm from one point to another was derived. The near minimum time path which minimises this lower bound and the product of the path's length and curvature was shown to be a geodesic in inertia space. Numerical examples were presented, showing that their derived

approximate solutions *usually* required less time than the minimum distance paths and the joint-interpolated paths.

The only formal results deal with the restricted minimum time problem with torque constraints (with constant bounds) and terminal velocity constraints. Ailon and Langholz [1] and Wen and Desrochers [34] studied the existence and structure of time-optimal controls for robotic manipulators. The condition which ensures the existence of time-optimal control was demonstrated by them. Sontag and Sussman [27], [28] provided some formal results on the presence of singular extremals for the same restricted problem and considered some special cases. Their results, although contributing to the understanding of motions in this subproblem, do not apply generally when other cost function and realistic constraints are present. No computational algorithm was proposed.

The aim of this paper is to describe an off-line minimum cost path/trajjectory planner which (i) takes into account the full nonlinear dynamics; (ii) is able to generate the bang-bang torque profiles for the regular case when it is theoretically expected; (iii) is applicable to the full six DOF case; (iv) is able to handle a general set of constraints, including those on the magnitudes of the joint coordinates, joint velocities, joint torques and joint jerks, and can incorporate obstacle avoidance constraints; and (v) incorporates the energy objective.

The discrete planner is based on a discrete dynamic robot model [16] and formulated as a nonlinear program. Work on applying the discrete dynamic robot model to trajectory planning on fixed paths was initiated by Flash and Potts [6] and developed by Tan and Potts [29]. The discrete path/trajjectory planner to be described in this paper is based on an extension to the discrete trajectory planner as presented in [30] which deals with the easier case when the path is already prescribed. A preliminary outline of this path/trajjectory planner is given in [31].

The organisation of the paper is as follows. In Section 2, the mathematical model of the discrete path/trajjectory planner and the solution methods for the nonlinear program (NLP) are presented. In Section 3, simulation results for the minimum time case will be presented for the two DOF planar arms [3] studied in [22], [20], and the Stanford arm [18] using its first three degrees of freedom and then for all six degrees of freedom. Simulation results for the composite time/energy case will be presented in Section 4. Obstacle avoidance with the discrete planner is demonstrated in a case study in Section 5. Discussion and conclusions follow in Section 6.

2. The discrete path/trajectory planner

2.1. The mathematical formulation

In a robotic task, the specification of a geometric path is usually given in hand space and since the discrete path/trajectory planner operates in the joint space, the given endpoints of transfer movement have to be transformed by inverse kinematics to those in the joint space. The optimal joint path obtained by the planner is then transformed using direct kinematics to yield the minimum cost path in hand space.

The geometric path in hand space is discretised into M intervals by introducing $M + 1$ knots. Correspondingly, the joint coordinate profiles to be planned by the planner are also discretised into joint coordinates $q_p(k)$, $k = 0, \dots, M$, $p = 1, \dots, N$ where N is the number of degrees of freedom for the robotic manipulator involved. Hereafter the subscript p should always be taken to represent the range $1, \dots, N$. The time intervals for traversing the intervals along the geometric path (or equivalently the joint path) are assumed to be equal for all intervals. This common time interval is denoted by Δt and its reciprocal, which it is preferable to use in the formulation, by H .

The joint velocities are assumed given by the smoothing formulae

$$\dot{q}_p(k) = -\dot{q}_p(k-1) + 2H\Delta q_p(k-1), \quad k = 1, \dots, M, \quad (1)$$

where

$$\Delta q_p(k) = q_p(k+1) - q_p(k), \quad k = 0, \dots, M-1 \quad (2)$$

give the displacements along the N joint coordinate profiles.

The minimum cost path/trajectory problem can be stated as: *given the initial and final joint coordinates, $q_p(0)$ and $q_p(M)$, and joint velocities, $\dot{q}_p(0)$ and $\dot{q}_p(M)$, determine the joint displacements $\Delta q_p(k)$, $k = 0, \dots, M-1$, and H , so that the total cost is minimised subject to geometrical, kinematical and dynamical constraints.*

The minimum time path problem is formulated as a nonlinear programming problem. The variables of the NLP are the displacements along the N joint coordinate profiles $\Delta q_p(k)$, $k = 0, \dots, M-1$ and the reciprocal time interval H . For the minimum time problem, the objective function is the linear function

$$Z = H, \quad (3)$$

which in order to minimise the total motion time, has to be maximised. When the energy objective is introduced, the resulting composite function is

$$Z = W_t Z_t + W_e Z_e \quad (4)$$

where W_l and W_e are positive weights, $Z_l = M/H$, and Z_e the nonlinear energy cost function.

The constraints to be considered in the paper are those on the magnitudes of the joint coordinates, joint velocities, joint torques and joint jerks.

The constraints on the joint coordinates are given by

$$q_p^-(k) \leq q_p(k) \leq q_p^+(k), \quad k = 1, \dots, M, \quad (5)$$

where the bounds give the limits on the possible range of the joint coordinates, and $q_p^+(M) = q_p^-(M) = q_p(M)$ for the final joint coordinates constraints. These constraints are linear in the variables as $q_p(k) = q_p(0) + \sum_{i=0}^{k-1} \Delta q_p(i)$.

The joint velocities are restricted by the constraints

$$\dot{q}_p^-(k) \leq \dot{q}_p(k) \leq \dot{q}_p^+(k), \quad k = 1, \dots, M, \quad (6)$$

where for the final velocity equality constraints, the lower and upper bounds $\dot{q}_p^-(M)$ and $\dot{q}_p^+(M)$ are both set equal to $\dot{q}_p(M)$. These constraints are non-linear in the variables as evident from (1).

For the discrete dynamic robot model, the discrete joint torques are given as

$$F_p(k) = F_p^*(k) + \Delta V(p; k-1)/\Delta q_p(k-1), \quad (7)$$

$$\begin{aligned} F_p^*(k) = & H \sum_{i=1}^N \Delta \{d_{ip}(k-1)\dot{q}_i(k-1)\} - \frac{1}{2} \sum_{i=1}^{p-1} \frac{\Delta d_{ii}(p; k-1)}{\Delta q_p(k-1)} \dot{q}_i(k-1)\dot{q}_i(k) \\ & - \frac{1}{2} \sum_{i=1}^{p-1} \sum_{j=i+1}^N \frac{\Delta d_{ij}(p; k-1)}{\Delta q_p(k-1)} (\dot{q}_i(k-1)\dot{q}_j(k) + \dot{q}_i(k)\dot{q}_j(k-1)) \\ & + \sigma_p(\dot{q}_p(k) + \dot{q}_p(k-1))/2, \quad k = 1, \dots, M \end{aligned} \quad (8)$$

where $F_p(k)$ is the generalised force (or joint torque as it will be usually called), d_{ij} are the kinetic energy coefficients, V the potential energy, and σ_p a constant frictional coefficient. In (8), the first term corresponds to the inertial forces, the second term to the centripetal and the third to the coriolis forces acting in joint p . Note that in (7) and (8) we use the hybrid discrete evaluations of the multivariate continuous time function $f(q_1, q_2, \dots, q_N)$ with

$$\begin{aligned} \Delta f(p; k) = & f(q_1(k), \dots, q_{p-1}(k), q_p(k+1), q_{p+1}(k+1), \dots, q_N(k+1)) \\ & - f(q_1(k), \dots, q_{p-1}(k), q_p(k), q_{p+1}(k+1), \dots, q_N(k+1)). \end{aligned} \quad (9)$$

The constraints on the joint torques result from saturation limits of the joint motors and are expressed as

$$F_p^-(k) \leq F_p(k) \leq F_p^+(k), \quad k = 1, \dots, M \quad (10)$$

where the bounds may be position and velocity dependent.

Minimum time paths with the joint torque constraints lead to bang-bang profiles and it may be desirable in practice to smooth such abrupt variation in the joint torques to avoid excitation of elastic vibrations in the system or to prevent excessive wear of the gear mechanisms. To include this desirable feature in the formulation, the joint jerk G , defined as the time rate of change of joint torque, is discretised as follows:

$$G_p(k) = H(F_p^*(k+1) - F_p^*(k)), \quad k = 1, \dots, M-1 \quad (11)$$

and

$$G_p(0) = HF_p^*(1), \quad G_p(M) = -HF_p^*(M). \quad (12)$$

The constraints on the joint jerk are taken as

$$G_p^-(k) \leq G_p(k) \leq G_p^+(k), \quad k = 0, \dots, M. \quad (13)$$

The bounds can be selected to achieve different degrees of smoothness at different parts of the motion.

For the running of an NLP algorithm it is necessary to introduce bounds on the problem variables and these are taken as

$$H^- \leq H \leq H^+ \quad (14)$$

and

$$\Delta q_p^-(k) \leq \Delta q_p(k) \leq \Delta q_p^+(k), \quad k = 0, \dots, M-1. \quad (15)$$

In summary, the discrete path/trajectory planner is formulated as the following NLP: maximise the linear objective function (3), or minimise the nonlinear objective function (4), subject to the one linear constraint (14), the NM linear constraints (15), the NM linear constraints (5), the NM nonlinear constraints (6), the NM nonlinear constraints (10), and the $N(M+1)$ nonlinear constraints (13).

The $NM+1$ variables are the NM joint displacement variables $\Delta q_p(k)$ and the reciprocal time interval H .

For a typical case with $N=3$, $M=40$, there are 121 variables, 241 linear constraints and 363 nonlinear constraints.

2.2. Solving the nonlinear program

Before we discuss the various techniques for solving the nonlinear program, we shall examine some of its features.

The Jacobian matrix, giving the derivatives of the constraints with respect to the variables, must be specified for any NLP solver. Most commercially available NLP solvers allow for finite difference approximation of the Jacobian matrix but this is usually computationally expensive. In our case, the Jacobian matrix can easily be obtained using routine partial differentiation either by hand or with a symbolic algebraic language such as REDUCE [10].

An important feature of the Jacobian matrix is that it is regularly sparse. For the single variable H , the matrix elements are nonzero for all constraints except for the joint coordinates, which are independent of H . It is the numerous joint displacement variables which contribute to the sparsity of the Jacobian. For the linear constraints on the joint coordinates and all the nonlinear constraint functions, namely those for the joint velocity, joint torque, joint jerk, the values at the k th interval depend only on the previous k intervals, that is, on $\Delta q_r(l)$, $l = 0, \dots, k-1$, $r = 1, \dots, N$, and therefore

$$\partial q_p(k)/\partial \Delta q_r(l) = \partial \dot{q}_p(k)/\partial \Delta q_r(l) = \partial F_p(k)/\partial \Delta q_r(l) = 0, \quad l \geq k, \quad k = 1, \dots, M \quad (16)$$

and

$$\partial G_p(k)/\partial \Delta q_r(l) = 0, \quad l > k, \quad k = 0, \dots, M. \quad (17)$$

Furthermore, in the case of joint coordinates and joint velocities, we have

$$\partial q_p(k)/\partial \Delta q_r(l) = \partial \dot{q}_p(k)/\partial \Delta q_r(l) = 0, \quad p \neq r. \quad (18)$$

Each sub-block of the Jacobian matrix with respect to the joint displacement variables is therefore either a lower triangular or a zero matrix, with the exception of the joint jerk constraints which yield a near-triangular sub-matrix. For the constraints on the joint coordinates, joint velocities, joint torques and joint jerks, the density of the Jacobian matrix is given by $[3M + NM + 1 + M(M + 1)(N + 1)]/[(4M + 1)(MN + 1)] \times 100\%$. By way of illustration, for $N = 3$ and $M = 20$ the density of the Jacobian matrix is about 36%. It is therefore important that any NLP solver fully utilises sparsity to save storage and floating point operations.

An NLP solver requires an initial solution to start the algorithm in its search for the optimal solution. It is preferable for the initial path/trajectory which is used as input to be as feasible as possible. The specified input path should satisfy the given initial and final configurations and a simple choice is a straight line path in hand space. The chosen initial value of H should not be too large, as a comparatively small Δt corresponds to comparatively fast motion which is likely to cause most constraint functions to exceed their bounds, introducing large infeasibilities in the initial solution.

The nonlinear program for the path/trajectory planning problem can be solved by a number of algorithms such as the successive linear programming (SLP), successive quadratic programming (SQP) or the projected augmented Lagrangian algorithm. A relatively simple implementation of one variant of the SLP algorithm has been used successfully in solving fixed-path trajectory planning problems [29], [30], but this was not acceptable for the path/trajectory planning problem because of the increased nonlinearity. Instead we have used the commercially available packages NPSOL [9] and

MINOS [15]. The former is based on the SQP technique and has proved very reliable in solving the nonlinear program being considered here, but it does not exploit the sparsity of the Jacobian matrix. On the other hand, MINOS allows for sparse matrix storage and has therefore been preferred. For our constrained NLP with linear and nonlinear constraints, MINOS uses the projected augmented Lagrangian algorithm. Both NPSOL and MINOS allow for warm starting a run for solving an NLP and this feature can be utilised for our problem in an efficient manner. A run with a cold start using a coarse discretisation can be performed comparatively quickly to obtain an approximation to the optimal path, and this information, in the form of a basis, can be used to warm start a later run with a finer discretisation using more knots.

3. Simulation results—minimum time case

To demonstrate the practicability and feasibility of the discrete planner, this section presents simulation results, first for two two-link planar arms, and then for the Stanford manipulator with three degrees of freedom and finally six degrees of freedom. For the simulation, the discrete planner was programmed in FORTRAN and run with MINOS under VMS on a VAX-11/785 minicomputer in double precision arithmetic. For all test cases, we used the following bounds on the variables: $H^- = 1Hz$, $H^+ = 200Hz$, $\Delta q_p^+(k) = -\Delta q_p^-(k) = 0.5$ rad or m . Optimality tolerance used for all runs is chosen such that four significant figures in the objective value can be achieved.

3.1. Two-link planar arms

For comparison purposes, the discrete planner was used in two simulation studies; one for the two-link planar arm considered in [22] and the other for a similar two-link arm considered in [20]. These two references considered the minimum time path/trajectory planning problem with only the terminal velocity and joint torque constraints using constant saturation limits. As mentioned in the introduction, for such a problem, the necessity condition for a minimum time path is that all joint torques are bang-bang for the *regular case* and possibly up to $N - 1$ joint torques are not saturated for the *singular case*.

In the first simulation study, one of the two test cases considered by [22] will be used to demonstrate the *regular* bang-bang torque profiles obtained by our planner, in contrast to their results. In the second, the one test case considered by [20] is used to demonstrate the ability of the discrete planner to handle possible singular arcs in the torque profiles.

The parameters of the two-link arm, as used in [22], are $l_1 = l_2 = 0.5m$, $m_1 = 50kg$, $m_2 = 30kg$, $I_1 = 5kg \cdot m^2$, $I_2 = 3kg \cdot m^2$, where l_i , m_i , I_i are the link length, mass and moment of inertia for link i .

Case 1. The test case to be considered is that of moving the arm in its fully stretched configuration $(q_1, q_2) = (0, 0)$ in radians with the end effector at the position $(x, y) = (1, 0)$ in metres, to the final configuration $(-\pi/3, 2\pi/3)$ with the end effector at $(0.5, 0)$. The initial and final joint velocities are chosen to be zero. The only other constraints considered were bounds on the torques: $F_1^+(k) = -F_1^-(k) = 350N \cdot m$ and $F_2^+(k) = -F_2^-(k) = 100N \cdot m$, $k = 1, \dots, M$.

The minimum time path for this test case was reported in [22] to be the straight line path from the initial to the final configuration in both hand space and joint space. Their joint torque profiles showed *only one* joint torque saturated at any one time.

Using the discrete planner to solve the same problem, we performed two successive runs: 20 intervals with a cold start, then 40 intervals with a warm start from the previous run. The straight line path was chosen as the initial path and $20Hz$ for the initial value of H . The resultant minimum time path is slightly curved in both hand and joint space as shown in Figure 1. The bang-bang profiles for both torques are shown in Figure 2(a) and the joint velocities are shown in Figure 2(b). Note that the switching of torques is not instantaneous because our model is discrete. The main feature of the torque profiles is that both torques are saturated, in sharp contrast to the profiles obtained in [22]. Note also that although we have included in our formulation constraints for the joint coordinates and joint velocities, these were not used in [22]. For the comparative tests, we set the bounds to be sufficiently large so that these constraints were in effect inoperative.

The total motion time for the path obtained is 0.458 s in contrast to 0.525 s given in [22]. A direct comparison between the total motion times is not straightforward because different discretisation schemes are used. However, using the discrete fixed-path trajectory planner [30] to obtain the minimum time trajectory for the straight line hand path given in [22], we obtained a total motion time of 0.509 s. As both the discrete path/trajectory planner and discrete fixed-path trajectory planner are based on the same discretisation scheme, this at least confirms that our path/trajectory planner did produce a path with a shorter total motion time.

The comparison with Rajan's results [20] will be given next. The link parameters, and inertial coefficients are the same as in Case 1 except: $I_1 = 1.04kg \cdot m^2$, $I_2 = 0.625kg \cdot m^2$. Note that gravity is not considered. The torque bounds were chosen to be $\pm 10N \cdot m$ for both joints as in [20].

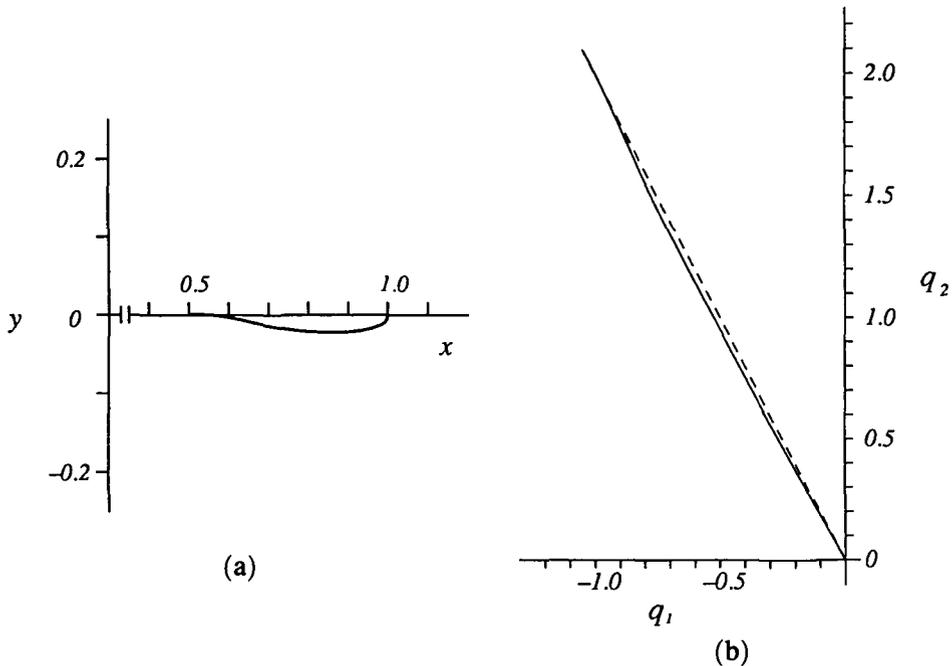


FIGURE 1. Two-link planar arm. Case 1: with joint torques constrained within constant saturation limits. For the point to point motion from $(1, 0)$ to $(0.5, 0)$ in metres in the $x - y$ plane (a) the curved line is the minimum time hand path obtained by the discrete path/trajectory planner, in contrast to the straight line hand path obtained by Sahar and Hollerbach; (b) the slightly curved solid line is the minimum time joint path obtained by the discrete planner compared to the straight dashed line obtained by Sahar and Hollerbach.

Case 2. The test case to be considered is that of moving the arm in its fully stretched configuration $(q_1, q_2) = (0, 0)$, in radians, with the end effector at the position $(x, y) = (1, 0)$, in metres, to the final configuration $(2\pi, 0)$ with the end effector back at $(1, 0)$. The initial and final joint velocities are chosen to be zero. The minimum time path for this test case was reported by Rajan [20] as having a motion time of 4.37 s and his torque profiles showed the joint torque for joint 1 saturated at all time while that for joint 2 is not saturated throughout the motion except for *short* durations.

Using the discrete path/trajectory planner to solve this same problem, we performed two successive runs: 20 intervals with a cold start, then 40 intervals with a warm start from the previous run. The initial path used to start the first run consists of two straight line segments in the joint space, from $(0, 0)$ to (π, π) and (π, π) to $(2\pi, 0)$; equally spaced intervals were taken along these straight line segments. The initial value of H was taken to be 4 Hz. The resultant minimum time path in both hand and joint space are shown in Figure 3. The ‘kick back’ feature of link 1 at the beginning of the optimal

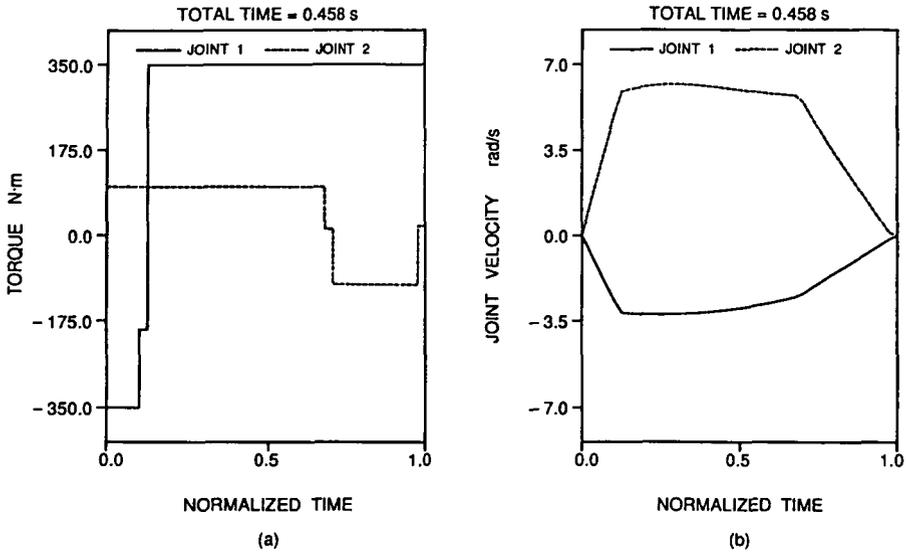


FIGURE 2. Two-link planar arm. Case 1. (a) Profiles of the joint torques for the minimum time path are shown, illustrating bang-bang behaviour. Joint 1 is saturated at the level $-350\text{ N}\cdot\text{m}$ in the normalised time interval $(0, 0.09)$ then at the level $350\text{ N}\cdot\text{m}$ in $(0.10, 1)$. Joint 2 is saturated at $100\text{ N}\cdot\text{m}$ in $(0, 0.67)$ and then at $-100\text{ N}\cdot\text{m}$ in $(0.69, 0.98)$. (b) Joint velocity profiles, showing abrupt changes in the acceleration at the switching times (approximately 0.10 and 0.69).

path, noticed by Rajan, is also evident in our result. The joint torque profiles are shown in Figure 4(a) and the joint velocities are shown in Figure 4(b). It can be seen that joint 1 executes a bang-bang torque profile for the whole motion and joint 2, while bang-bang for most part of the motion, is not saturated in the middle portion. The difference with Rajan's result lies in joint 2. The middle singular arc of joint 2, shown in our result, corresponds to the fact that link 2 is folded back on link 1 and $q_2 = \pi$. This phenomenon can be observed in Figure 3(a). Similar results have been obtained by Geering *et al.* [7] for special case studies on another two-link planar arm with different inertial coefficients and link parameters. It is important to note that because of discretisation and numerical errors unavoidable on finite-precision computers, the variables, $\Delta q_2(k)$, of the NLP were not exactly zero for the observed singular arc; neither is the joint velocity of joint 2 exactly zero (see Figure 4(b)). Again we have included in our simulation study constraints for the joint coordinates and joint velocities. As these were not used by Rajan, we set the bounds on these two constraints to be sufficiently large so that they were in effect inoperative.

The total motion time for the path we obtained is 4.28 s in contrast to 4.37 s by Rajan. This comparison on the motion time is not conclusive

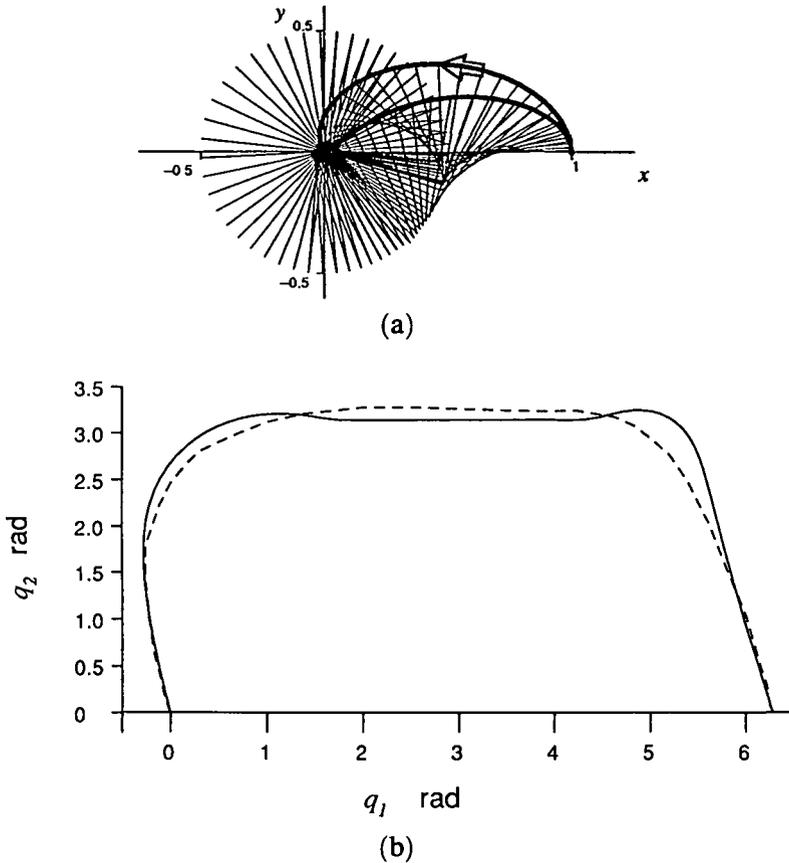


FIGURE 3. Two-link planar arm. Case 2: with joint torques constrained within constant saturation limits. For the point to point motion from $(1, 0)$ back to $(1, 0)$ in metres in the $x - y$ plane, (a) the thicker line shows the minimum time path obtained by the discrete path/trajectory planner while the sequence of thin lines shows the movement of the two links; (b) the solid line is the minimum time joint path obtained by the discrete planner compared to the dashed line obtained by Rajan.

as different discretisation schemes were used and the relevant information for Rajan's optimal path was not available to enable us to plan a fixed-path trajectory.

3.2. Stanford manipulator—using first three joints

The six DOF Stanford manipulator has been studied extensively by a number of researchers. It has five revolute joints and one prismatic joint (the third), all driven by DC-motors. The kinematic model, link parameters and actuator inertias of the arm used in this study were taken from [18]. The bounds on the various constraints to be used on this arm are shown in Table

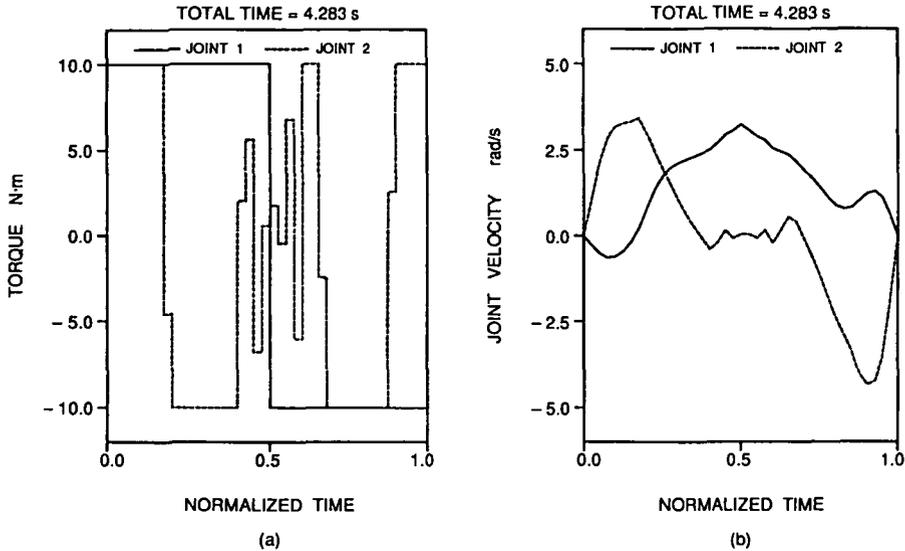


FIGURE 4. Two-link planar arm. Case 2. (a) Profiles of the joint torques for the minimum time path are shown, illustrating bang-bang behaviour. Joint 1 is saturated at the level $10N \cdot m$ in the normalised time interval $(0, 0.5)$ then at the level $-10N \cdot m$ in $(0.5, 1)$. Joint 2 is saturated at $10N \cdot m$ in $(0, 0.17)$, at $-10N \cdot m$ in $(0.19, 0.42)$, at $10N \cdot m$ in $(0.60, 0.65)$, at $-10N \cdot m$ in $(0.67, 0.87)$ and at $10N \cdot m$ in $(0.89, 1)$. (b) Joint velocity profiles, showing abrupt changes in the acceleration at the switching times.

TABLE 1. Bounds for four types of constraints used in the simulation for the Stanford manipulator.

Description	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
$q_p^- \text{ rad}$	-2.7925	-3.8920	0.1m	-3.9180	-1.7453	-4.6425
$q_p^+ \text{ rad}$	2.7925	2.7504	1.2m	3.9670	1.7453	4.6425
$\dot{q}_p^+ = -\dot{q}_p^- \text{ rad/s}$	10	10	5 m/s	10	10	10
$F_p^+ = -F_p^- N \cdot m$	120	140	100N	3	5	1
$G_p^+ = -G_p^- N \cdot m/s$	500	500	500N/s	12	15	4

1, and although they are not obtained from the actual robot itself they are adequate for our simulation studies.

The inertial coefficients d_{ij} and the potential energy V are central to the discrete dynamic model, and expressions for them were obtained using the computation procedure given in [18] with the aid of the symbolic algebraic manipulation language REDUCE.

We first consider simulation runs with only the first three joints, which serve to position the end effector in its workspace. The constant saturation torque limits listed in Table 1 are used for ease of illustration, although

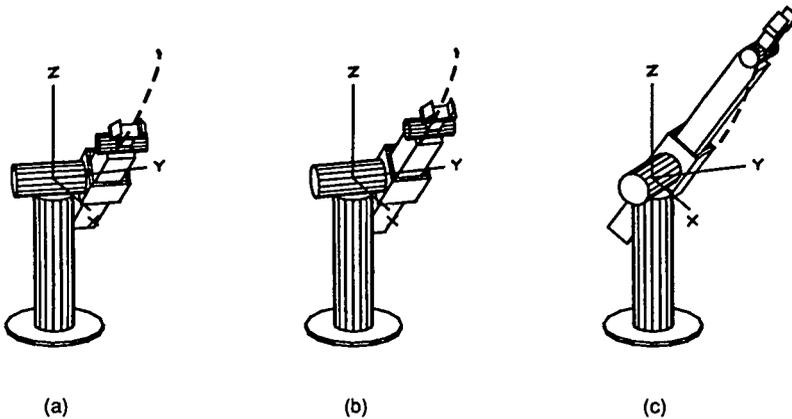


FIGURE 5. Stanford manipulator using first three joints. Case 3: with constraints on joint coordinates, joint velocities and joint torques (force for joint 3). The perspective view of the minimum time path obtained by the discrete path/trajectory planner: (a) initial, (b) intermediate, (c) final position. The dashed line shows the path in the cartesian space. The base coordinate frame is shown.

variable torque bounds can be just as easily handled by the planner. The initial and final joint velocities are again taken to be zero for all joints. The path to be determined is to start from the position $(x, y, z) = (0.3, 0.1, 0.2)$ in metres and end at $(0.1, 0.3, 0.4)$. The initial path to be input to the planner is taken as the straight line path between these two points. The initial value of H for the first run for all following cases was taken as $20Hz$.

Case 3. With constraints on the joint coordinates, joint velocities and joint torques, three runs were performed using 10, 20 and 40 intervals. The first run was cold started and the next two warm started. A perspective view of the minimum time path in hand space for 40 intervals is illustrated in Figure 5. The bang-bang torque profiles are shown in Figure 6(a). The joint velocity profiles, shown in Figure 6(b), indicate that none of the velocity constraints are binding.

Case 4. Same as Case 3 but the velocity bounds for joint 1 are reduced to $\dot{q}_1^+(k) = -\dot{q}_1^-(k) = 7rad/s$ to illustrate the effect of binding velocity constraints. For this modified problem, the resulting torque profiles and velocity profiles are shown in Figure 7(a) and 7(b). The torques for joint 1 are no longer bang-bang throughout the motion as the joint velocity for joint 1 becomes saturated for the middle part of the motion.

Case 5. The bang-bang profiles may not be desirable and a smooth motion might be preferred. To illustrate the ability of the discrete planner to achieve

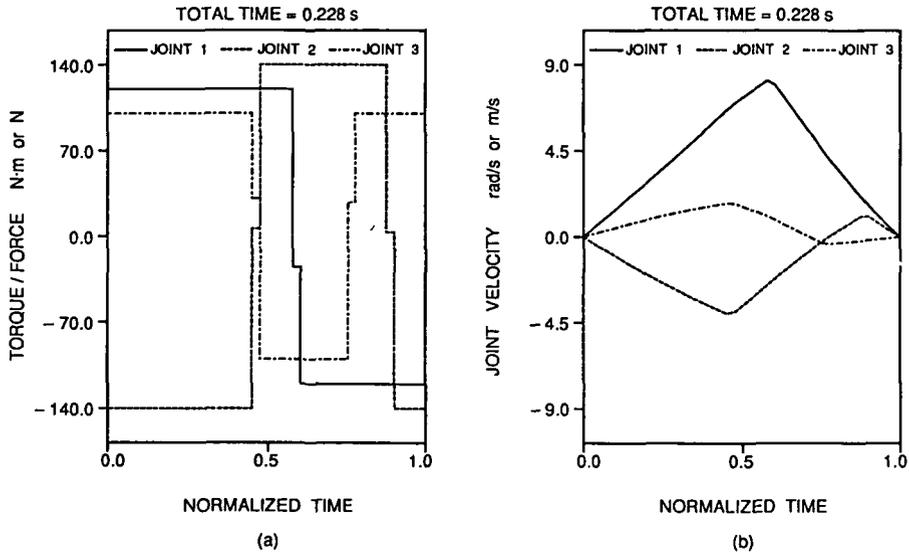


FIGURE 6. Stanford manipulator using the first 3 joints. Case 3. (a) Profiles of the joint torques for the minimum time path are shown, illustrating bang-bang behaviour. Joint 1 is saturated at the level $120N \cdot m$ in the normalised time interval $(0, 0.57)$, then at the level $-120N \cdot m$ in $(0.60, 1)$. Joint 2 is saturated at the level $-140N \cdot m$ in $(0, 0.45)$, then at $140N \cdot m$ in $(0.47, 0.87)$ and finally at $-140N \cdot m$ in $(0.89, 1)$. Joint 3 is saturated at $100N$ in $(0, 0.45)$, then at $-100N$ in $(0.47, 0.75)$ and finally at $100N$ in $(0.77, 1)$. (b) The joint velocity profiles, none of which are saturated. The sharp change in accelerations at the switching times of the torques are shown for the respective joints.

this, we include the jerk constraints in our previous test case. For this modified problem, the resulting 'smooth' torque profiles and velocity profiles are shown in Figure 8. This smoother motion has been obtained at the expense of an increase in motion time. A noteworthy point is that inclusion of the joint jerk constraints has led to bang-bang profiles of the joint jerks rather than of the joint torques.

3.3. Stanford manipulator—using all six joints

The last three degrees of freedom of the arm are now included to demonstrate the ability of the planner to handle the full six DOF problem. In this case, the minimum time path in hand space involves changes in orientation of the end effector as well as in its position. For the test path, the end effector is to move from the start position $(x, y, z) = (0.2, 0.4, 0.4)$ to the end position $(0.5, 0.1, 0)$ with the initial and final orientations of the end effector both set to the Euler angles $(\phi, \theta, \psi) = (90^\circ, -10^\circ, 90^\circ)$.

Case 6. With constraints on all six joint coordinates, joint velocities and joint torques, two runs were performed using 10 and 20 intervals. The initial path

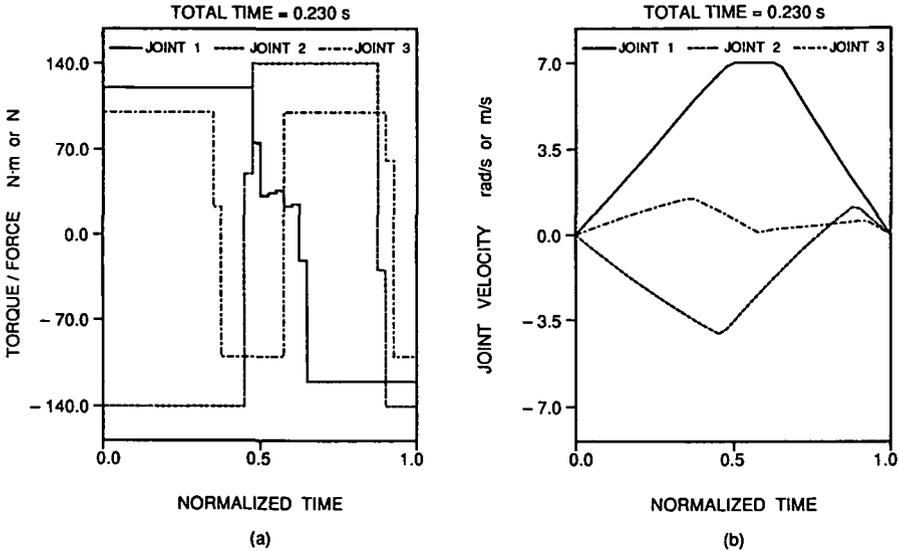


FIGURE 7. Stanford manipulator using the first 3 joints. Case 4: with constraints on joint coordinates, joint velocities and joint torques (force for joint 3). (a) Profiles of the joint torques for the minimum time path are shown. Joint 1 is saturated at the level $120N \cdot m$ in $(0, 0.47)$, then at $-120N \cdot m$ in $(0.65, 1)$. Joint 2 is saturated at $-140N \cdot m$ in $(0, 0.45)$ then at $140N \cdot m$ in $(0.47, 0.86)$ and finally at $-140N \cdot m$ in $(0.89, 1)$. Joint 3 is saturated at $100N$ in $(0, 0.35)$, at $-100N$ in $(0.37, 0.57)$, at $100N$ in $(0.57, 0.90)$ and finally at $-100N$ in $(0.92, 1)$. (b) The joint velocity profiles, with joint 1 saturated at $7.0rad/s$ in the interval $(0.49, 0.64)$.

used was a straight line in position space and an unchanging orientation in the orientation space. The first run was again cold started and the second warm started. The resulting torque profiles and velocity profiles are shown in Figures 9 and 10. For the minimum time path, the orientation of the end effector changed throughout the duration of the motion, performing a 'loop' in the orientation space.

Case 7. Same as in Case 6 except with the addition of joint jerk constraints. The same initial solution as in Case 6 was used and two runs were performed using 10 and 20 intervals. The resulting 'smooth' torque profiles and velocity profiles are shown in Figures 11 and 12. As in the three DOF case, smooth motion was achieved at the expense of an increase in the total motion time; bang-bang profiles of the joint jerks were obtained.

4. Simulation results—minimum time/energy case

To demonstrate the ability of the discrete path/trajectory planner to incorporate the energy cost function in the search for a minimum cost path,

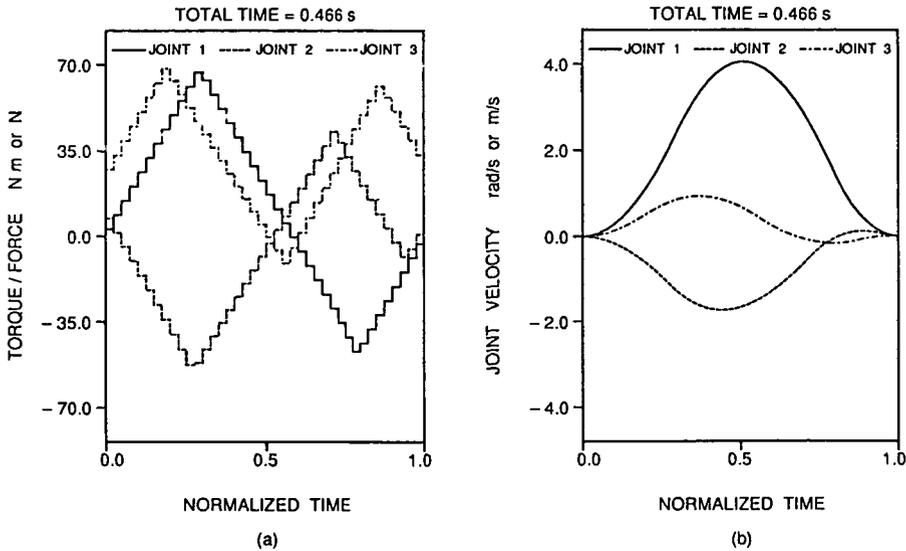


FIGURE 8. Stanford manipulator using the first 3 joints. Case 5: with constraints on joint coordinates, joint velocities, joint torques (force for joint 3) and joint jerks. For the minimum time path obtained, (a) smooth profiles of joint torques, and (b) smooth profiles of joint velocities corresponding to smooth motion, are shown. None of the joint torques nor joint velocities are saturated.

the test case involving only the first three joints of the Stanford manipulator (Case 3 in Section 3) will be used here. The energy cost function includes the electrical energy consumption and frictional energy loss functions. For the discrete planner, we use the following discrete forms:

$$Z_{ee} = H^{-1} \sum_{k=1}^M \sum_{p=1}^N R_p F_p(k)^2 \tag{19}$$

and

$$Z_{fe} = H^{-1} \sum_{k=1}^M \sum_{p=1}^N \frac{1}{2} \sigma_p \dot{q}_p(k)^2 \tag{20}$$

where R_p is a coefficient related to the actuator characteristics. These two functions are quadratic in torques and velocities, and are therefore nonlinear in the variables. The minimum time/energy cost function used is

$$Z = W_t Z_t + W_{ee} Z_{ee} + W_{fe} Z_{fe}, \tag{21}$$

where $Z_t = M/H$.

For a range of W_{ee} and with $W_{fe} = 10^{-5}$ and $W_t = 1$, minimum time/energy paths were obtained with the discrete planner for a 40-interval case warm-started from three runs with 10, 20 and 40 intervals. The values of

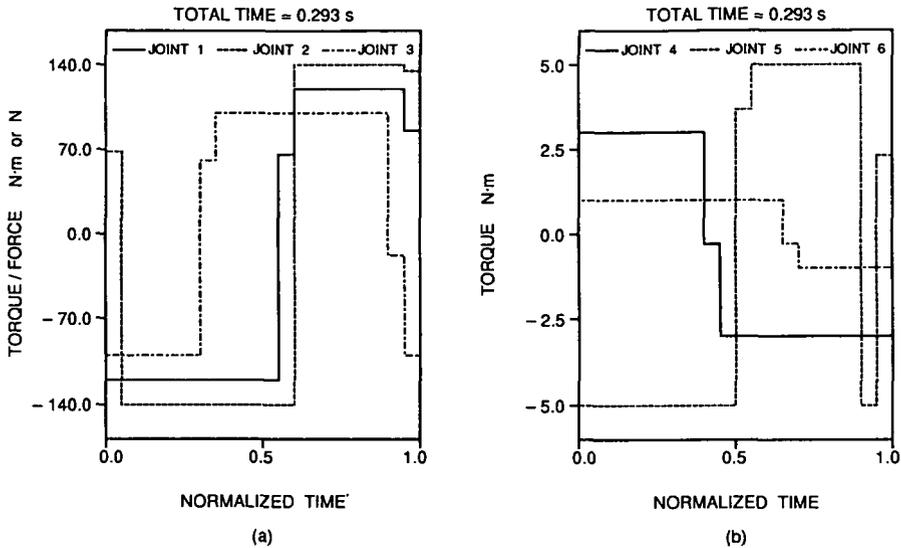


FIGURE 9. Stanford manipulator using all 6 joints. Case 6: with constraints on joint coordinates, joint velocities, joint torques (force for joint 3). For the minimum time path, bang-bang profiles of joint torques are shown for (a) first three degrees of freedom: joint 1 is saturated at the level $-120N \cdot m$ in the interval $(0, 0.55)$ and at $120N \cdot m$ in $(0.60, 0.95)$; joint 2 is saturated at $-140N \cdot m$ in $(0.05, 0.60)$ and at $140N \cdot m$ in $(0.60, 0.95)$; joint 3 is saturated at $-100N$ in $(0, 0.30)$, then at $100N$ in $(0.35, 0.90)$ and at $-100N$ in $(0.94, 1)$; (b) last three degrees of freedom: joint 4 is saturated at $3N \cdot m$ in $(0, 0.40)$ and at $-3N \cdot m$ at $(0.45, 1)$; joint 5 is saturated at $-5N \cdot m$ in $(0, 0.50)$, at $5N \cdot m$ in $(0.55, 0.90)$ and at $-5N \cdot m$ in $(0.90, 0.95)$; joint 6 is saturated at $1N \cdot m$ in $(0, 0.65)$ and at $-1N \cdot m$ in $(0.70, 1)$.

the energy function and the total motion time are plotted against the weight W_{ee} in Figure 13. It shows the expected trend that with increasing weight on the energy consumption, the energy decreases and the total motion time increases. A noteworthy feature is that near the minimum time solution, there is a relatively large drop in the energy consumption corresponding to a small increase in the total motion time. Therefore it might be more desirable to use one of these near minimum time trajectories rather than the minimum time trajectory.

The effect on the joint torque profiles with increasing weights W_{ee} proceeds as follows: starting from the bang-bang profiles for all three joints for the minimum time case, joint 3 is the first to lose its total bang-bang profile, followed by joint 2 and finally joint 1.

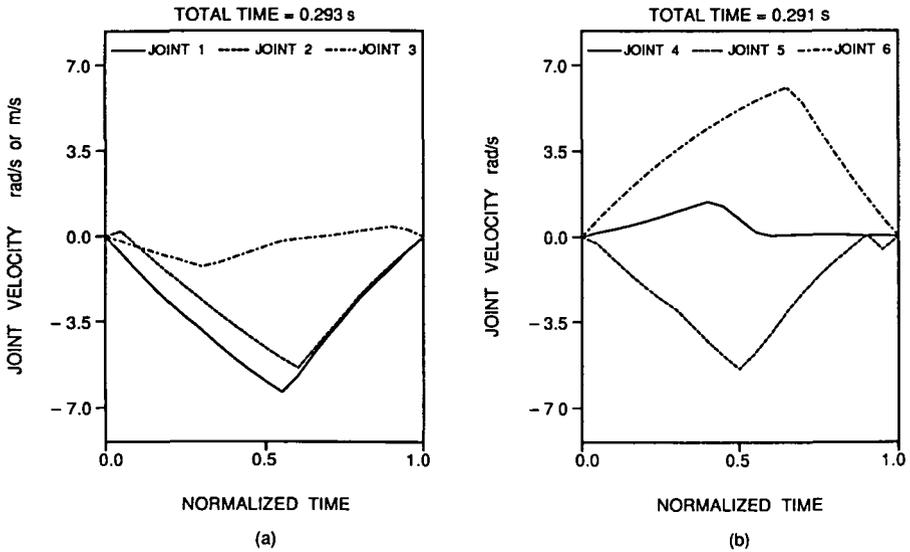


FIGURE 10. Stanford manipulator using all 6 joints. Case 6. For the minimum time path, profiles of joint velocities are shown for (a) first three degrees of freedom (b) last three degrees of freedom.

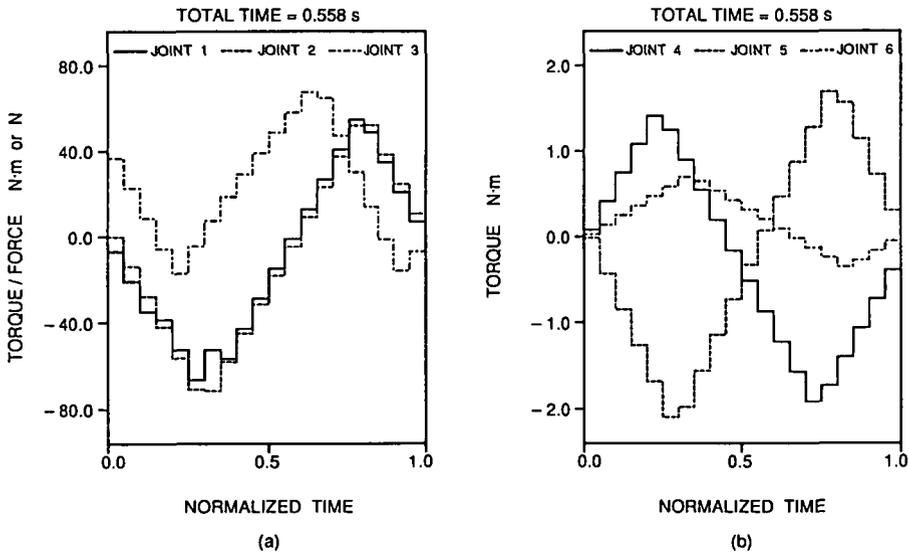


FIGURE 11. Stanford manipulator using all 6 joints. Case 7: with constraints on joint coordinates, joint velocities, joint torques (force for joint 3) and joint jerks. For the minimum time path, smooth profiles of joint torques are shown for (a) first three degrees of freedom (b) last three degrees of freedom.

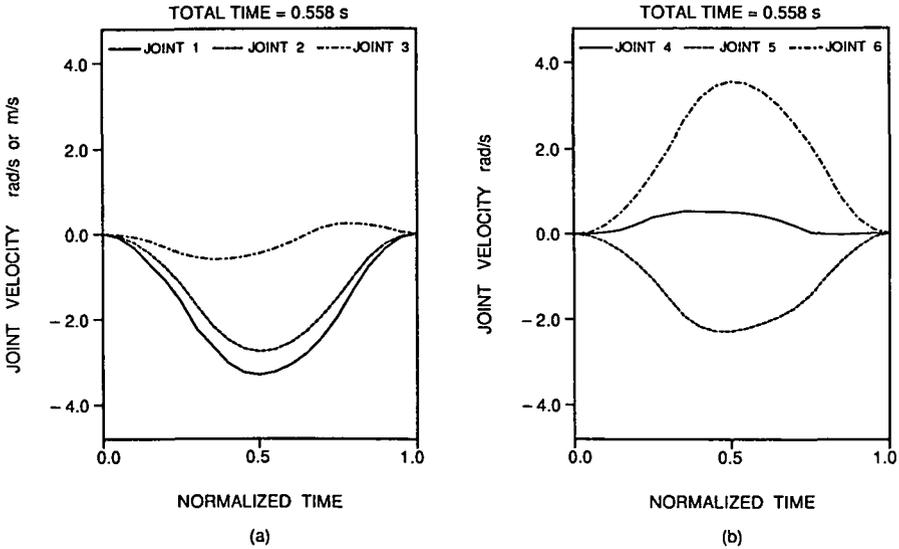


FIGURE 12. Stanford manipulator using all 6 joints. Case 7. For the minimum time path, smooth profiles of joint velocities are shown for (a) first three degrees of freedom; (b) last three degrees of freedom.

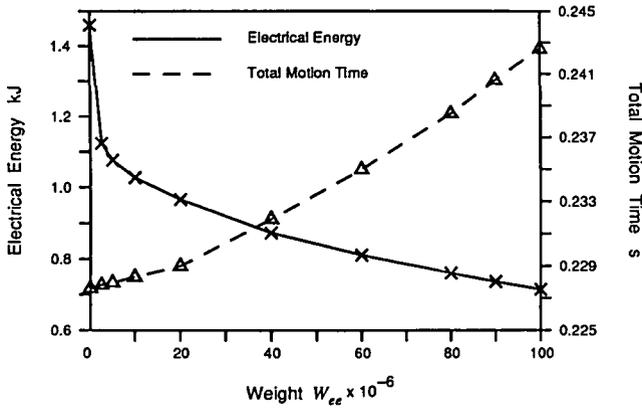


FIGURE 13. For ten minimum time/energy path/trajectories obtained for the Stanford manipulator, Case 3, the electrical energy in kilojoules and the total motion time in seconds are plotted against the electrical energy weight factor W_{ee} .

5. Obstacle avoidance—a case study

In this section, a possible method of incorporating obstacle avoidance into the discrete path/trajectory planner is discussed and demonstrated with an example involving a two-link planar arm.

Stationary workspace obstacles are conveniently described in hand coordinates, but since the discrete planner operates in joint space, we apply the inverse kinematics transformation to obtain their corresponding forbidden regions in the joint space. Obstacle modelling or approximating techniques can then be applied to the forbidden regions to obtain functions describing their boundary curves. If there are a total of R forbidden regions, and $Q_i(q_1, \dots, q_N)$, $i = 1, \dots, R$, denote their boundary functions, then avoidance of the corresponding workspace obstacles are assumed to be achieved by finding the set of joint coordinates, q_1, \dots, q_N , satisfying

$$Q_i(q_1, \dots, q_N) \geq 0, \quad i = 1, \dots, R. \quad (22)$$

If it is assumed that the starting and endpoints are collision-free, then these R constraints must be satisfied at each of the $M - 1$ knots along the path, $q_p(k)$, $k = 1, \dots, M - 1$ and hence we have a total of $R(M - 1)$ constraints introduced into the NLP of the discrete planner.

To illustrate this method of incorporating obstacle avoidance, we consider again Case 1 in Section 3. The link movements of the arm for this minimum time path are shown in Figure 14(a). A stationary rectangular obstacle is introduced in the workspace, as indicated in Figure 14(a), so that a collision would take place if the links executed the minimum time path.

With the use of a sweeping volume method [21] and assuming that the links have zero width (alternatively if the links have finite width, one can shrink the links to straight lines by 'growing' the obstacle, as done in [3]), the forbidden region in the joint space was found to be that illustrated in Figure 14(b). It is a simple closed region, and an approximate function describing its boundary curve is

$$Q(q_1, q_2) = 8(q_2 - P(q_1))^2 + (q_1 + 0.15)^2 - 1.02, \quad (23)$$

where $P(q_1)$ is a polynomial function fitted to the curve passing through the middle of the region. The $N(M - 1)$ constraints, corresponding to avoiding the forbidden region, are added to the NLP of the discrete planner. Two straight line segments in the joint space, $(0, 0)$ to $(-0.3, 1.3)$ and $(-0.3, 1.3)$ to $(-\pi/3, 2\pi/3)$, which do not penetrate the forbidden region, were used as the initial path. Two successive runs were performed: a 20-interval run followed by a 40-interval run. The minimum time path in the joint space is shown in Figure 14(b) and the collision-free hand path shown in Figure 14(c). The total motion time has been increased to 0.763 s from 0.458 s obtained in Case 1. In achieving the minimum time path/trajectories, the joint path just touches the boundary of the forbidden region in joint space for some time and correspondingly, link 2 just touches the top left corner of the rectangular obstacle in hand space.

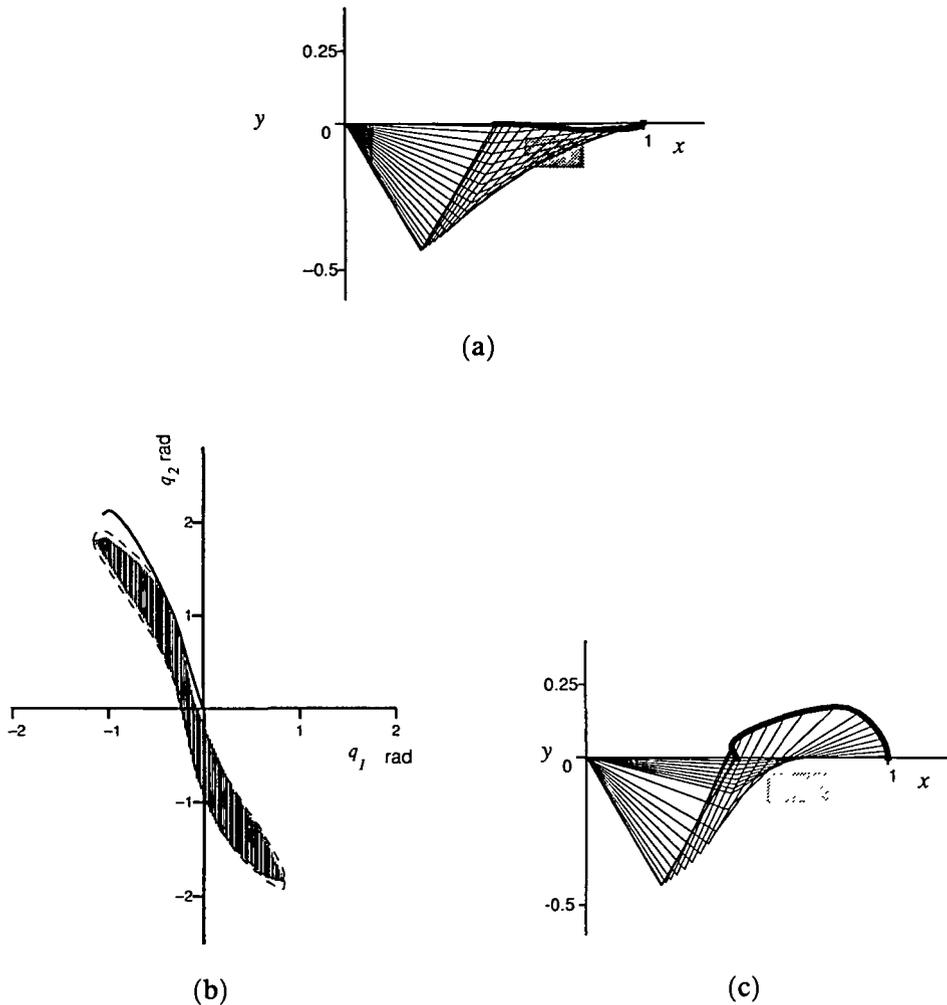


FIGURE 14. Obstacle avoidance for a two-link arm, (a) the minimum time path as obtained in Case 1, showing link movements and the rectangular obstacle to be avoided; (b) the vertically shaded forbidden region in joint space is shown with the approximating boundary curve (dashed); the collision-free minimum time joint path is shown lying outside the forbidden region; and (c) the collision-free minimum time path as obtained by the discrete planner.

6. Discussion and conclusion

In order to examine the applicability and efficiency of the off-line discrete planner proposed in this paper, several features of its computer implementation will be discussed.

6.1. Size of problem

The discrete path/trajectory planner is based on an NLP formulation and its implementation can be limited by the size of problem that an NLP package can handle and the amount of storage available on a particular computer.

If m is the total number of constraints, both linear and nonlinear, then for MINOS, the package we have used, m must not exceed 32767. If all four types of constraints discussed are used, the total number of constraints in the planner is $N(4M + 1)$. If the number of degrees of freedom is $N = 6$ then the upper bound on the number of discrete intervals M is about 1360 which is far beyond what would be needed.

It is storage requirements which impose a practical limitation. The amount of work space storage required by MINOS is roughly $100m$ words, and one "word" is the relevant storage unit for the floating-point arithmetic being used. Our simulation study was run on a VAX-11/785 with double precision, requiring approximately $800m$ bytes. Approximately 300K bytes are also needed for the MINOS subroutines and buffer space for several files. In order to compute the Jacobian matrix for the NLP, further storage is required for the dynamical coefficients, the potential energy terms and their derivatives with respect to the variables, but sparsity of the matrix can be exploited. In a multi-user environment on a VAX-11/785, the compilation of the planner has been restricted by the page file quota imposed on a process. For $N = 6$, the planner is currently compilable for $M \leq 22$, and for $N = 3$, $M \leq 86$.

Our current implementation of the discrete planner restricts its use for problems with higher degrees of freedom to shorter paths, although it should be pointed out that the bounds on M are obtained on the assumption that all four types of constraints are considered, and all dynamical coefficients are dependent on all the joint variables. Storage savings are possible if: i) the independency of dynamical coefficients on certain joint variables for a particular robot is used to avoid storing zero derivative elements; ii) ways are devised to compress storage for Jacobians of joint coordinates and joint velocities for which elements are all '1' and ' $\pm 2H$ ' respectively; iii) the Jacobian elements for the joint jerks are computed from those of the joint torques and hence storage is not needed for them; iv) internal files are used to store some of the array variables; v) runs with smaller M are used to determine any non-binding constraints which can be eliminated when M is increased.

6.2. Computational statistics

Although for off-line planning the time for computer implementation may not be critical, nevertheless it should not be excessive. The computer CPU times for the test cases considered in Section 3 are listed in Table 2 together with other statistics.

TABLE 2.

For the seven test cases considered in Section 3, the number of intervals M along the path, the degrees of freedom N , the number of variables, the total number of constraints, the size of the time interval Δt and the CPU times taken are shown.

Case	M	N	#Variables	#Constraints	Δt (ms)	CPU times
Case 1	20	2	41	161	22.9	32 s
	40	2	81	321	11.5	1min 6 s
Case 2	20	2	41	161	214.5	2min 2 s
	40	2	81	321	107.1	4min 20 s
Case 3	10	3	31	121	22.8	19 s
	20	3	61	241	11.4	42 s
	40	3	121	481	5.7	2min 42 s
Case 4	10	3	31	121	23.2	18 s
	20	3	61	241	11.5	43 s
	40	3	121	481	5.8	2min 52 s
Case 5	10	3	31	154	46.5	20 s
	20	3	61	304	23.3	53 s
	40	3	121	604	11.6	3min 3 s
Case 6	10	6	61	241	29.4	3min 13 s
	20	6	121	481	14.7	7min 42 s
Case 7	10	6	61	307	55.7	4min 18 s
	20	6	121	607	27.9	9min 27 s

As expected, Cases 6 and 7 with six degrees of freedom are most demanding on computation. The CPU times for the other cases with two and three degrees of freedom are much less. For most runs using the time objective and with a cold start, the number of iterations (each involving a linear sub-problem) is in the range 5–7 and is less with a warm start. The convergence is fairly rapid, and numerically the order of convergence is estimated to be roughly quadratic. This good convergence feature is shown in all runs which have tight constraint sets, that is, as many constraints binding as there are variables in the NLP. Convergence is slower with the singular torque case (as in Rajan's test case) and when the energy objective is introduced; more iterations are needed in the initial stage of the NLP algorithms to get near the optimum before superlinear or quadratic convergence to the optimum can be achieved. Typically, the computation times increased by a factor from three to five.

One interesting point to note is that the total motion time, the objective function of the NLP, does not depend sensitively on the choice of the path near the optimum. As evident from Table 2, a doubling in the value of M results in almost exactly a halving of the optimal value of Δt . There is of course little point in striving too hard to achieve the exact optimum of a

nonlinear program formulation of a model of a robot but it is reassuring that in the vicinity of the optimal path there is a family of near optimal paths for which the total motion times are almost the same.

A useful test of the accuracy of the computations was carried out by using the three algorithms SLP, NPSOL and MINOS for selected problems. The results obtained were identical. As a further test, the discrete planner was run for various classical calculus of variations problems for which the exact solutions are known. For the optical problems of reflection and refraction, and for the brachistochrone problem and for a minimum surface area problem, the correct results were obtained [32].

A strict comparison of the computing time efficiency with various existing methods is difficult. The control-theoretic methods reported long computational time [7], [17] though details were not published. The graph-search method of Sahar and Hollerbach reported minutes to hours of CPU time on a Lisp machine even for their simple two DOF manipulator. The other search-space methods using parameterised paths [20], [5] would be more time efficient; and it is indicated without details, by Dubowsky et al. [5], that their method takes about 5–10 minutes of CPU time on a Micro VAX. Our discrete path/trajectory planner takes about 15 minutes on a VAX-11/785 for a full six DOF manipulator, 20 intervals and constraints on joint coordinates, joint velocities, joint torques and joint jerks. It has to be noted that in using the discrete planner, far more path-related variables are being perturbed in the solution process than the number of parameters related to, say, four spline spans in the search-space methods using parameterised paths. It is expected that more effort in developing an in-house but robust NLP solver, rather than the sophisticated but general purpose MINOS package, could better exploit the structure of the NLP and improve the time and space efficiency of the discrete planner.

6.3. Conclusion

The minimum cost discrete path/trajectory planner presented in this paper has been shown to be generally applicable to manipulators with up to six degrees of freedom. It is able to give the theoretically expected bang-bang profiles (for regular control) and can accommodate constraints on joint coordinates, joint velocities, joint torques and joint jerks. The CPU times on a minicomputer are quite acceptable except for longer paths for six degrees of freedom, but improvement here is possible with better use of computer storage. Both minimum time and weighted minimum time/energy objectives have been considered and other appropriately chosen objectives can be incorporated.

Two extensions of the discrete path/trajectory planner are being investigated. The first is to the practical but less complex problem for six degrees of freedom when the minimum time paths are to be planned with the orientation of the end effector fixed during the motion. A composite approach is being used with the discrete path/trajectory planner formulation for the first three degrees of freedom and the discrete fixed-path trajectory planner formulation for the final three degrees of freedom. The second extension is to develop further the simple technique used to model forbidden regions in joint space and apply it to the more general case of avoiding three-dimensional stationary or moving obstacles.

Acknowledgement

This work has been supported by a grant awarded under the Australian Research Grants Scheme.

References

- [1] A. Ailon and G. Langholz, "On the existence of time-optimal control of mechanical manipulators", *J. Opt. Theory Appl.* **46** (1985) 1–21.
- [2] J. E. Bobrow, S. Dubowsky and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths", *Int. J. Robotics Res.* **4** (1985) 3–17.
- [3] M. Brady et al. (eds), *Robot motion: planning and control* (MIT Press, Cambridge MA, 1982).
- [4] M. L. Brown, "Optimal robot planning via state space networks", M. S. Thesis, Princeton University, 1984.
- [5] S. Dubowsky, M. A. Norris and Z. Shiller, "Time optimal trajectory planning for robotic manipulators with obstacle avoidance: a CAD approach", *Proc. IEEE Conf. Robotics Autom.* (1986) 1906–1912.
- [6] T. Flash and R. B. Potts, "Discrete trajectory planning", *Int. J. Robotics Res.* **7** (1988) 48–57.
- [7] H. P. Geering et al., "Time-optimal motions of robots in assembly tasks", *IEEE Trans. Autom. Contr.* **31** (1986) 512–518.
- [8] E. G. Gilbert and D. W. Johnson, "Distance functions and their applications to robot path planning in the presence of obstacles", *IEEE J. Robotics Autom.* **1** (1985) 21–30.
- [9] P. E. Gill et al., "User's Guide for NPSOL (Version 4.0)", Stanford University, Department of Operations Research, Report SOL-86-2 (1986).
- [10] A. C. Hearn, *REDUCE User's Manual, Version 3* (Rand Corp, California, 1984).
- [11] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories", *ASME J. Dyn. Meas., Contr.* **106** (1984) 102–106.
- [12] M. E. Kahn and B. Roth, "The near-minimum-time control of open-loop articulated kinematic chain", *J. Dyn. Sys., Meas., Contr.* **93** (1971) 164–172.
- [13] B. K. Kim and K. G. Shin, "Suboptimal control of industrial manipulators with a weighted time-fuel criterion", *IEEE Trans. Autom. Contr.* **30** (1985) 1–10.

- [14] P. M. Lynch, "Minimum-time, sequential axis operation of a cylindrical, two-axis manipulator", *Proc. Joint Autom. Contr. Conf.* **1** (1981) paper WP-2A.
- [15] B. A. Murtagh and M. A. Saunders, "MINOS 5.1 User's Guide", Stanford University, Department of Operations Research, Report SOL 83-20R (1987).
- [16] C. P. Neuman and V. D. Tourassis, "Discrete dynamic robot models", *IEEE Trans. Sys. Man Cyber.* **15** (1985) 193–204.
- [17] M. Niv and D. M. Auslander, "Optimal control of a robot with obstacles", *Proc. Amer. Contr. Conf.* (1984) 280–287.
- [18] R. P. Paul, *Robot manipulator—mathematics, programming and control* (MIT Press, Cambridge, 1981).
- [19] R. B. Potts, "Discrete Lagrange equations", *Bull. Austral. Math. Soc.* **3** (1988) 227–233.
- [20] V. T. Rajan, "Minimum time trajectory planning", *Proc. IEEE Conf. Robotics Autom.* (1985) 759–764.
- [21] W. E. Red and H. Truong-Cao, "The configuration space approach to robot path planning", *Proc. Amer. Contr. Conf.* (1984) 288–295.
- [22] G. Sahar and J. M. Hollerbach, "Planning of minimum-time trajectories for robot arms", *Int. J. Robotics Res.* **5** (1986) 90–100.
- [23] V. Scheinman and B. Roth, "On the optimal selection and placement of manipulators", *Proc. RoManSy: 5th CISM-IFTOMM Symp.* (1984) 39–46.
- [24] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints", *IEEE Trans. Autom. Contr.* **30** (1985) 531–541.
- [25] K. G. Shin and N. D. McKay, "Selection of near-minimum time geometric paths for robotic manipulators", *Proc. Amer. Contr. Conf.* (1985) 346–355.
- [26] W. E. Snyder and W. A. Gruver, "Microprocessor implementation of optimal control for a robotic manipulator system", *Proc. IEEE Conf. Dec. Contr.* (1979) 839–841.
- [27] E. D. Sontag and H. J. Sussman, "Remarks on the time-optimal control of two link manipulators", *Proc. IEEE Conf. Dec. Contr.* (1985) 1643–1652.
- [28] E. D. Sontag and H. J. Sussman, "Time-optimal control of manipulator", *Proc. IEEE Conf. Robotics Autom.* (1986) 1692–1697.
- [29] H. H. Tan and R. B. Potts, "Minimum time trajectory planner for the discrete dynamic robot model with dynamic constraints", *IEEE J. Robotics Autom.* **4** (1988) 174–185.
- [30] H. H. Tan and R. B. Potts, "A discrete trajectory planner for robotic arms with six degrees of freedom", University of Adelaide, Dept. of Applied Maths. Research Report UAAM-87-5 (1987).
- [31] H. H. Tan and R. B. Potts, "Minimum time robot motions", University of Adelaide, Dept. of Applied Maths. Research Report UAAM-87-2 (1987).
- [32] H. H. Tan and R. B. Potts, "A discrete calculus of variations algorithm", *Bull. Austral. Math. Soc.* **38** (1988) 365–371.
- [33] T. L. Turner and W. A. Gruver, "A viable suboptimal controller for robotic manipulators", *Proc. IEEE Conf. Dec. Contr.* (1980) 83–87.
- [34] J. Wen and A. Desrochers, "Existence of the time optimal control for robotic manipulators", *Proc. Amer. Contr. Conf.* (1986) 109–113.