Reliable Computation of Minimum-Time Motions for Manipulators Moving in Obstacle Fields Using a Successive Search for Minimum-Overload Trajectories

Jong-keun Park*
Department of Mechanical and Automation Engineering
Kyungnam University
Masan 631-701, South Korea
e-mail: jkpark@kyungnam.ac.kr

James E. Bobrow
Department of Mechanical and Aerospace Engineering
University of California at Irvine
Irvine, CA 92697, U.S.A.
e-mail: jebobrow@uci.edu

Received 6 September 2002; accepted 31 August 2004

In this paper we present a practical method for finding obstacle-free minimum-time motions for manipulators subject to the limits of velocity-dependent actuator forces. An optimal motion-planning problem is converted into a finite-dimensional nonlinear programming problem by means of parameter optimization with quintic B-splines. We introduce the concept of the minimum-overload trajectory in which the motion time is specified to be faster than the actuators can handle, and the actuator overloads are minimized with the motion time fixed. By successive searches for minimum-overload trajectories, minimum-time motions are determined for three example manipulators without simplifying any of the kinematic, dynamic or geometric properties of the manipulators or the obstacles. In the resultant minimum-time motions, almost all of the joint actuators are close to saturation during the motions.

1. INTRODUCTION

In the past two or three decades, determination of minimum-time motions for manipulators has been an active area of research in robotics because minimum-time motions yield high productivity and efficiency, especially when they are executed repeatedly. This subject is roughly divided into two categories according to the tasks manipulator arms should fulfill. These categories are characterized by motions with or without geometric path constraints.

*To whom all correspondence should be addressed. Contract sponsor: Kyungnam University Research Fund (2002)

Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/rob.200

If the geometric path of the end-effector of a nonredundant manipulator is predetermined, the motion has 1 degree of freedom (DOF) and can be represented by a scalar path variable. In this case, rigorous solutions were obtained subject to constant bounds on the actuator forces.[1,2] Later, studies were extended to cases where the path included a certain singular points on it,[3] or actuator jerks as well as actuator torques were limited within constant bounds,[4] or joint velocities and end-effector velocity were bounded additionally.[5]

Most manipulator tasks, except arc welding, painting or cutting, etc., are essentially motions without geometric path constraints. In this case obstacle avoidance should be considered simultaneously with time minimization. These kinds of manipulator motions have the same DOF as the number of joints and minimum-time motions are more complicated than above. The subject of our study is in this category.

Various kinds of methods have been developed to solve the minimum-time optimal control problem with obstacles. Traditional optimal control theory,[6–9] nonlinear programming,[10–12] dynamic programing,[13] tessellation of joint[14] or configuration space,[15] or a mixture of them[16,17] are the main techniques used. By applying optimal control theory, Pontryagin's maximum principle leads to a two-point boundary value problem. Some researchers have tried to solve these equations directly[6,8] and others have tried to solve them through parameter optimization.[17] Though the theory and the solutions are rigorous, it has been used to solve for motions for 2- or at most 3-link planar manipulators due to the complexity and the nonlinearity of the manipulator dynamics.

Approximation methods, though the solutions cannot be proved to be optimal, have been studied to obtain the solutions for 3 or more DOF spatial manipulators. These approximation methods are roughly divided into two groups according to whether they utilize gradients or not. Most algorithms based on nonlinear programing use gradients to find optimal motions efficiently.[10–12,18,19] For stable convergence, objective functions and constraints must be locally convex and their first derivatives must be continuous. Numerically calculated gradients have been used to find minimum-time motions,[11] but the simulation model was a 3-link spatial manipulator in the presence of relatively simple obstacle model. Later, analytically calculated gradients were used to minimize actuator torques for various multi-body systems[18] or spatial 6-link manipulators.[19] But torque or energy minimizations show more stable convergence properties than minimum-time motions because the motion time is fixed.

Other approximation methods not utilizing gradients are mainly based on (1) approximations in small time-intervals[12,13,17] or (2) discretization (tessellation)[14–16] of joint or configuration space. The former takes less CPU time but may accumulate numerical or modeling errors in the small time-intervals to lower the accuracy of results. The latter assures stable convergence but CPU time may increase exponentially for refinement of tessellation.

Because of the complex dynamics and kinematics of robot manipulators, various assumptions or simplifications were introduced for the purpose of online implementation or simplification of the algorithms. As geometric simplifications, obstacles have been ignored[7,10,13,17,20] or modeled as circles[9,12,14] or as finite surface points[16] and robot links were modeled as lines[9,14,20] or as finite surface points[12] or as ellipsoids.[16] As kinematic simplifications, motions were restricted in a plane[8,9,14,20,21] or the orientation of end-effector was ignored[12,15] or joint velocity profiles[10] or joint acceleration profiles[13,16,22] were prespecified. As dynamic simplifications, manipulator dynamics was ignored subject to the only kinematic constraints.[10,21,22] To the best of our knowledge the minimum-time motions for 6 or more DOF manipulators or multiple robot arms have not yet been obtained without simplifying any of the geometric, kinematic or dynamic models of manipulators or obstacles.

In our study we transform the optimal control problem in function space into a nonlinear programing problem in a finite-dimensional vector space. Joint displacements are represented by linear combinations of finite-term quintic B-splines. If sufficiently many terms are used, with a proportional (not exponential) increase of CPU time, the result will be the exact solution. Using numerically calculated gradients, the coefficients of the splines are obtained optimally.

The novel contribution in this study is the concept of the *minimum-overload trajectory* with total motion time fixed. Minimum-time motion is defined rigorously with this concept and it is found successfully by sequential searches for minimum-overload trajectories. In the minimum-overload searches, the convergence is quite stable because the performance index and all constraints are locally convex and smooth and because the total motion time is fixed.

To compute the minimum-overload trajectory, the total motion time is, at first, specified to be too fast, so that the actuators need more force than they can produce. Then, using an efficient numerical optimization, the actuator overloads are minimized during the motion. Using the information obtained from the minimum-overload trajectory, we predict the motion time of the next minimum-overload search. These successive searches continue until the least time is found when the minimum overload vanishes.

Obstacle information is evaluated with penetration growth distances[23] and its avoidance is achieved by a penalty term included in an augmented performance index. The usefulness of the penetration growth distance will be shown in simulation results. Full geometric, kinematic and dynamic models of a spatial 6-link manipulator and obstacles are considered in this study. The effects of friction are the only variables ignored.

The manipulator dynamics is calculated by the outward and inward iteration method.[24] This iteration method needs the only joint-by-joint recursive calculations and accumulates little error. In addition, it is applicable without knowing the equations of motions explicitly in higher DOF models. On the other hand, in optimal control approaches, the differential equations of motion are solved numerically and this is difficult since errors accumulate for higher DOF models.

In most other studies, constraints on the actuator forces are constant regardless of joint velocities. It is more practical that the bounds of actuator forces are dependent on joint velocities as was done in this study. Example applications to a planar 2-link, a spatial 3-link and a 6-link Puma 560 manipulator in the presence of polyhedral obstacles demonstrate the effectiveness and the numerical stability of this algorithm.

2. MINIMUM-TIME MOTIONS OF CONTINUOUS SYSTEM

The equations of motions for a manipulator arm are

$$\tau = \mathbf{M}(\theta)\alpha + \mathbf{v}(\theta,\omega) + \mathbf{g}(\theta), \tag{1}$$

where τ is generalized actuator forces $(n\times1)$, $\mathbf{M}$ is inertia matrix $(n\times n)$, θ, ω and α are generalized joint displacements, velocities and accelerations $(n\times1)$, $\mathbf{v}$ is centrifugal and coriolis forces $(n\times1)$, and $\mathbf{g}$ is gravitational forces $(n\times1)$.

To find minimum-time motions, the following performance index is minimized:

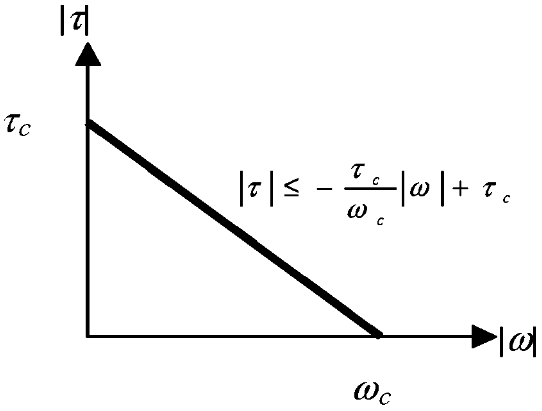


Figure 1. Actuator characteristics of one joint.

$$J = \int_0^T dt = T, \tag{2}$$

where T is the total motion time.

In most practical cases, the limits on the actuator forces are dependent on the joint velocities. In this study, we consider actuator bounds of the form shown in Figure 1. In this case the constraints on the actuators are

$$|\tau| \leq -\mathbf{W}_m|\omega| + \tau_c, \tag{3}$$

where $|\boldsymbol{\tau}|$, $|\boldsymbol{\omega}|$ are absolute values of the elements $(n\times1)$ and $\mathbf{W}_m$ is constant diagonal matrix whose principal diagonal elements are $(\tau_c)_i/(\omega_c)_i$ $(i=1,\ldots,n)$, and τ_c is the actuator limit vector $(n\times1)$ defined in Figure 1.

If we define the *equivalent actuator forces* τ_e as

$$\tau_e = |\tau| + \mathbf{W}_m|\omega| \tag{4}$$

then the actuator constraints (3) become

$$\tau_e \leq \tau_c. \tag{5}$$

To formulate obstacle avoidance constraints, we use Ong and Gilbert's growth function.[23] We briefly review the approach here. Assume there exists a convex *object* $\mathbf{A}$ in a three dimensional workspace. The *object* is defined as a set of all the points inside and on the surface of a rigid body. Let $\mathbf{p}_A$ be an arbitrary point (*seed point*) fixed inside $\mathbf{A}$. Then a *growth model* $\mathbf{G}_A(\sigma)$ is defined as

$$\mathbf{G}_A(\sigma) = \{\mathbf{y}|\mathbf{y} = \mathbf{p}_A + \sigma(\mathbf{x} - \mathbf{p}_A), \mathbf{x}\in\mathbf{A}\}, \tag{6}$$

where σ is a non-negative scalar.

Let there exist another convex object $\mathbf{B}$ in the same workspace and let $\mathbf{G}_B(\sigma)$ be the growth model of $\mathbf{B}$ w.r.t. a seed point $\mathbf{p}_B$ fixed inside $\mathbf{B}$. Then a *growth function* σ^* ($\mathbf{A}$, $\mathbf{B}$) is defined as

$$\sigma^*(\mathbf{A},\mathbf{B}) = \min\{\sigma | \mathbf{G}_A(\sigma) \cap \mathbf{G}_B(\sigma) \neq \varnothing\}. \qquad (7)$$

The growth function can be calculated by linear programing if $\mathbf{A}$ and $\mathbf{B}$ are convex polyhedra. The dimension of this linear programing problem is 4, thus the *active set method*[25] is efficient for such a low dimensional LP problem.

Let there exist m static obstacles in a workspace. We assume that all the obstacle models $\mathbf{O}_1,\ldots,\mathbf{O}_m$ and link models $\mathbf{R}_1(t),\ldots,\mathbf{R}_n(t)$ are convex polyhedra. However, nonconvex models are permissible if they can be decomposed into multiple convex models. If the growth function σ^* of a link model $\mathbf{R}_i$ and an obstacle model $\mathbf{O}_j$ is less than one, one model penetrates into the other and the following *penetration growth distance* D_{ij} indicates the amount of penetration:

$$D_{ij} = (d_i + d_j)\{1 - \sigma^*(\mathbf{R}_i, \mathbf{O}_j)\}_+, \qquad (8)$$

where d_i and d_j are appropriate positive real numbers that represent the actual geometric sizes of $\mathbf{R}_i$ and $\mathbf{O}_j$, respectively, and the plus operator is defined as

$$\{\bullet\}_+ = \begin{cases} \bullet, & \text{if } \bullet \geqslant 0 \\ 0, & \text{if } \bullet < 0' \end{cases} \qquad (9)$$

In general, the penetration growth distance is not equal to the minimum translational distance separating the objects. From the above notation, the obstacle avoidance constraints become

$$\mathbf{D} = \mathbf{0}, \quad \forall t \in [0,T], \qquad (10)$$

where $\mathbf{D}$ is a matrix $(n \times m)$ whose elements are Eq. (8).

Motions at the start and the goal positions are specified as

$$\theta(0) = \theta_s, \quad \theta(T) = \theta_f, \qquad (11)$$

$$\omega(0) = \omega_s, \quad \omega(T) = \omega_f, \qquad (12)$$

$$\alpha(0) = \alpha_s, \quad \alpha(T) = \alpha_f. \qquad (13)$$

The acceleration conditions (13) are given to assure smooth motions at the start and the goal positions.

The minimum-time problem for a continuous system can be stated as

Find $\theta(t)$ which minimizes T subject to (1), (5), (10) and (11)–(13).

This optimal motion-planning problem will be transformed into a finite-dimensional nonlinear programing problem in the following sections.

3. FINITE-DIMENSIONAL JOINT TRAJECTORIES

An infinite number of linearly independent basis functions form a complete set in a function space and this set can represent an arbitrary piece-wise continuous function defined on a closed interval. A finite number of them can express a piece-wise continuous function approximately. Many researchers have used cubic B-splines as the basis functions. In this research, however, quintic B-splines[26] are used.

Both splines play almost the same role when they are used in robot motion planning. The trajectories expressed by quintic B-splines, in spite of more computational burden, have some merits. (1) Accelerations and jerks are continuous third and second order polynomials, (2) they can express various kinds of displacement functions more accurately, (3) they have wider range of feasible directions if used in nonlinear programing and (4) for a given number of basis functions, the optimal performance indices are usually less than that found using cubic B-splines.

The quintic B-spline used in this research is

$$\begin{aligned} B_j(s) = (\tfrac{1}{120})[\{s-(j-3)\}_+^5 - 6\{s-(j-2)\}_+^5 + 15\{s-(j \\ -1)\}_+^5 - 20\{s-j\}_+^5 + 15\{s-(j+1)\}_+^5 - 6\{s-(j \\ +2)\}_+^5 + \{s-(j+3)\}_+^5, \quad s \in [-\infty,\infty], \end{aligned} \qquad (14)$$

where j is an arbitrary integer. The basis function $B_j(s)$ is positive for $j-3 < s < j+3$ and zero otherwise. The nodal values of Eq. (14) and its s-derivatives are listed in Table 1.

If we choose the parameter s-interval $[0, k]$ to express joint trajectories in the time interval $[0, T]$, then $k+5$ splines $B_{-2}(s),\ldots,B_{k+2}(s)$ have nonzero values in $[0, k]$. The joint trajectories are expressed by the linear combinations of the $k+5$ splines as

$$\theta(s) = \mathbf{CB}(s), \qquad (15)$$

$$s = \beta t, \qquad (16)$$

where s is a dummy variable connecting joint vari-

Table I. Nodal values of $B_j(s)$ and its derivatives.

Nodes(s)	$j-3$	$j-2$	$j-1$	j	$j+1$	$j+2$	$J+3$
$B_j(s)$	0	$\frac{1}{120}$	$\frac{26}{120}$	$\frac{66}{120}$	$\frac{26}{120}$	$\frac{1}{120}$	0
$B_j'(s)$	0	$\frac{1}{24}$	$\frac{10}{24}$	0	$-\frac{10}{24}$	$-\frac{1}{24}$	0
$B_j''(s)$	0	$\frac{1}{6}$	$\frac{2}{6}$	$-\frac{6}{6}$	$\frac{2}{6}$	$\frac{1}{6}$	0

ables with time, **C** is a coefficient matrix $(n \times k+5)$, $\mathbf{B}(s)$ is a column vector $(k+5 \times 1)$ whose elements are $B_{-2}(s),\ldots,B_{k+2}(s)$, and a time-scale factor β $(=k/T)$ defines the motion time, thus k remains constant even though the motion time varies.

Differentiating Eq. (15) w.r.t. time,

$$\omega(s)=\beta\mathbf{C}\mathbf{B}'(s), \tag{17}$$

$$\alpha(s)=\beta^2\mathbf{C}\mathbf{B}''(s), \tag{18}$$

where the primes $(',")$ mean differentiation w.r.t. s.

The initial and the final motion conditions (11)–(13) can be used to reduce the dimension of the coefficient matrix. By the algebraic manipulation described in the Appendix Section 8.1, the joint trajectories satisfying the initial and the final motion conditions are written as

$$\theta(s)=\mathbf{F}_s(s)+\mathbf{C}_m\mathbf{B}_m(s)+\mathbf{F}_f(s), \tag{19}$$

where the boundary condition splines $\mathbf{F}_s(s)$ and $\mathbf{F}_f(s)$, each dimension is $(n\times 1)$, can be determined from the initial and the final motion conditions. $\mathbf{C}_m$ is *the reduced coefficient matrix* $(n\times k-1)$ and $\mathbf{B}_m(s)$ is *the reduced B-spline basis functions* $(k-1\times 1)$.

The velocities and the accelerations are

$$\omega=\beta[\mathbf{F}_s'(s)+\mathbf{C}_m\mathbf{B}_m'(s)+\mathbf{F}_f'(s)], \tag{20}$$

$$\alpha=\beta^2[\mathbf{F}_s''(s)+\mathbf{C}_m\mathbf{B}_m''(s)+\mathbf{F}_f''(s)]. \tag{21}$$

Equation (19) together with (16) means that arbitrary joint trajectories subject to the initial and the final motion conditions are represented approximately by a point in an $n(k-1)$-dimensional linear vector space defined by the reduced coefficient matrix $\mathbf{C}_m$.

4. MINIMUM-OVERLOAD TRAJECTORIES

4.1. Performance Index

The basic approach used in our study requires that we iteratively find minimum-overload trajectories with the motion time fixed. In this section the actuator constraints (5) are no longer treated as constraints, but as the factors to define the following *overload of trajectories*:

$$J_c=\frac{1}{T}\int_0^T \|\mathbf{p}\|_2^2 dt, \tag{22}$$

where $\|\cdot\|_2$ means Euclidean norm and $\mathbf{p}$ is an actuator overload vector whose elements are

$$(\mathbf{p})_i=\left\{\frac{(\tau_e)_i}{(\tau_c)_i}-1\right\}_+, \quad i=1,\ldots,n, \tag{23}$$

where the plus operator is defined as Eq. (9).

To obtain obstacle-free minimum-overload trajectories, we treat the obstacle avoidance constraints as the penalty term in nonlinear programing, thus the augmented performance index becomes

$$J=J_c+w_oJ_o, \tag{24}$$

where w_o is a sufficiently large weighting coefficient, and J_o is defined as

$$J_o=\frac{1}{T}\int_0^T \sum_{i=1}^{n}\sum_{j=1}^{m} D_{ij}^2 dt. \tag{25}$$

The obstacle-free minimum-overload trajectories with the motion time fixed can be stated as

Find $\mathbf{C}_m \in \Re^{n(k-1)}$) which minimizes (24) subject to (1) and (19)–(21).

4.2. Optimization Using a Quasi-Newton Method

By assigning arbitrary specific values to $\mathbf{C}_m$, we can calculate the objective functional (24) and along successive search directions determined by the BFGS algorithm,[27] we find the optimal $\mathbf{C}_m$. The details are described as follows.

The s-interval [0, k] is divided into l equal subintervals; Δs and Δt are k/l and T/l, respectively. All the following variables are calculated at every $l+1$ nodal points. Using the initial and the final motion conditions (11)–(13) together with appropriately chosen constants k and T, we calculate (A20), (A22), (A23) and their s-derivatives. Assigning arbitrary initial values to $\mathbf{C}_m$, the joint trajectories (19)–(21) are calculated. Applying the manipulator dynamics (1) which is calculated by the outward and inward iteration method,[24] the actuator forces and the equivalent forces (4) are calculated. The penetration growth distance (8) between every link and obstacle is obtained by the active set method.[25]

Using all of the above data at the $l+1$ nodal points, together with an appropriately chosen w_o, the objective functional (24) is calculated with the trapezoidal area integral formula. The gradients are then calculated by the following central difference method:

$$(\mathbf{G})_{ij} \cong \frac{J((\mathbf{C}_m)_{ij}+\delta)-J((\mathbf{C}_m)_{ij}-\delta)}{2\delta}, \quad i=1,\ldots,n, j=1,\ldots,k-1. \tag{26}$$

The B-spline basis functions have some merit at this point. In other words, since the B-spline function is nonzero in a small interval, the increments of the objective functional in the small interval are the only data needed to calculate the corresponding component of the gradient. This gradient $(n\times k-1)$ should be rearranged to a vector $(n(k-1)\times 1)$ to calculate search directions using the BFGS method.[27] A line search is then performed using the golden section search method.[27]

The convergence criterion is

$$(J_{i-1}^{*}-J_i^{*})/J_{i-1}^{*} \leq \varepsilon_J, \tag{27}$$

where J_i^{*} is the minimum value of (24) along the ith line search. We can also check that the gradient is nearly zero at the optimal point.

4.3. Global Search in an Obstacle Field

The numerical optimization described above can be modified to search for the global minimum time path in a given obstacle field. The modification involves repeating the search after changing the position of the seed points within the obstacles defined in Eq. (6). Since the growth function is defined as Eq. (7), the path is updated during the optimization so that the Euclidean distances between the seed points of the link models and the obstacles increase. If we locate the seed point of an obstacle model in an inner corner of the model, the links tend to avoid the obstacle by turning around the opposite vertex. By changing the locations of seed points, we can find all the local minima and can choose the global one that has the least minimum overload. Although we have no proof that this method will always produce the global minimum time path, it has for all of the examples we have solved, including those discussed in this work.

4.4. Other Computational Issues

Another merit of this algorithm is that initial motions may be arbitrary regardless of obstacles. For fast convergence, however, it is helpful to minimize the following error norm.

Find an initial $\mathbf{C}_m$ that minimizes

$$J_{ini}=\int_0^T \|\theta-\theta_p\|_2^2 dt, \tag{28}$$

where θ is defined by Eq. (19) and θ_p is a fifth degree polynomial satisfying the initial and the final motion conditions (11)–(13), (see the Appendix Section 8.2). This minimization process is not a dynamic problem but only a kinematic one and takes little CPU time.

The parameters d_i and d_j in Eq. (8) are the radii of the minimum circumscribed spheres of the two models. The constants k and l are set to 15 and 75, respectively. k has an effect on the accuracy of the joint trajectory and l on the accuracy of the trapezoidal area integral. δ and ε_J are assigned to 10^{-7} and 10^{-12}, respectively, with the computations done in double precision.

5. MINIMUM-TIME MOTIONS

If the models of a manipulator and obstacles are defined and the actuator characteristics are specified, the minimum overload J_c^* in (22) is a function of motion time T, thus we can define the minimum time as follows.

Definition (Minimum time): The minimum time $T^* \equiv \min\{T | J_c^*(T)=0\}$.

Theorem: For an obstacle-free point-to-point manipulator motion, if $T<T^*$, then the minimum overload $J_c^*(T)>0$. If $T \geqslant T^*$, then $J_c^*(T)=0$.

Proof: For $T<T^*$, and the fact that $J_c(T)$ in (22) is non-negative, the above definition implies that $J_c^*(T)>0$. For $T \geqslant T^*$, and the fact that $J_c^*(T)$ in (22) is a monotonically decreasing function of T, $J_c^*(T^*)=0$, and must remain zero for all $T \geqslant T^*$.

5.1. Motion Times of Subsequent Searches

The theorem shows that as we increase the motion time T, starting from a time less than T^*, at some point we will achieve T^* if such a time exists. Thus, a simple line search can be used to find T^*. However, we achieved superior performance with the following heuristic algorithm.

To obtain T_{j+1} for the next minimum-overload search, we must consider the most overloaded joint at the jth minimum-overload trajectories.

First, compute the *saturation rates* as

$$\eta_\tau = \left\| \frac{1}{T} \int_0^T \mathbf{r}_\tau dt \right\|_\infty , \tag{29}$$

$$\eta_\omega = \left\| \frac{1}{T} \int_0^T \mathbf{r}_\omega dt \right\|_\infty , \tag{30}$$

where $\|\cdot\|_\infty$ of a vector is defined as the maximum of absolutes of all elements. The elements of the vectors $\mathbf{r}_\tau$ and $\mathbf{r}_\omega$ are

$$(\mathbf{r}_\tau)_i = (|\tau|)_i / (\tau_c)_i , \quad i=1,\ldots,n, \tag{31}$$

$$(\mathbf{r}_\omega)_i = (|\omega|)_i / (\omega_c)_i , \quad i=1,\ldots,n. \tag{32}$$

Next, compute the *overload rate* η_o at the same trajectories as

$$\eta_o = \|\mathbf{r}_o\|_\infty , \tag{33}$$

where the elements of the vector $\mathbf{r}_o$ are

$$(\mathbf{r}_o)_i = \frac{1}{(\mathbf{T}_+)_i} \int_0^T \frac{(\tau_e)_i}{(\tau_c)_i} d(\mathbf{t}_+)_i , \quad i=1,\ldots,n, \tag{34}$$

where $(\mathbf{t}_+)_i$ denotes the time when ith actuator is overloaded, thus

$$(\mathbf{t}_+)_i = \begin{cases} t, & \text{if } (\tau_e)_i > (\tau_c)_i , \\ 0, & \text{if } (\tau_e)_i \leqslant (\tau_c)_i , \end{cases} \quad i=1,\ldots,n \tag{35}$$

and the vector $\mathbf{T}_+$ denotes as follows:

$$\mathbf{T}_+ = \int_0^T d\mathbf{t}_+ . \tag{36}$$

In a numerical implementation, Eq. (34) can be approximated by the simple mean value of $(\tau_e)_i / (\tau_c)_i$ at the time-nodes when ith actuator is overloaded.

Finally T_{j+1} is obtained by the following formula:

$$T_{j+1} = T_j \left\{ c_R \left(\frac{\eta_\tau}{\eta_\tau + \eta_\omega} \eta_o^{1/2} + \frac{\eta_\omega}{\eta_\tau + \eta_\omega} \eta_o - 1 \right) + 1 \right\}, \tag{37}$$

where c_R is a coefficient governing the speed of convergence, that is, if it is near 1, the search converges fast but may be unstable and, if it is near 0, the search converges slowly but is stable. In this study 0.7 or 0.8 is used for c_R.

In order to explain the heuristic used for Eq. (37), note that if the motion time of jth search is much smaller than expected minimum time, η_τ is much more than η_ω on the jth minimum-overload trajectories, thus the actuator forces make a much greater contribution to the overload rate η_o than the joint velocities. On the other hand, if the motion time is near the expected minimum time, η_τ is almost the same as η_ω, thus the actuator forces and the joint velocities make almost equal contributions.

For a linear motion of a particle traveling a fixed distance in constant acceleration, the force is inversely proportional to the motion time squared and the velocity is inversely proportional to the motion time. Extending this reasoning, if the overload rate η_o is influenced by the actuator forces only, T_{j+1} should be increased to $T_j(\eta_o)^{1/2}$. On the other hand, if it is influenced by the joint velocities only, T_{j+1} should be increased to $T_j \eta_o$. Thus Eq. (37) is probably the most reasonable way to determine T_{j+1}. However, Eq. (37) is not the only way but the problem is convergence speed.

5.2. Other Computational Issues

To improve the numerical stability of the algorithm, the weighting coefficient w_o in (24) is updated at each subsequent search according to the rule

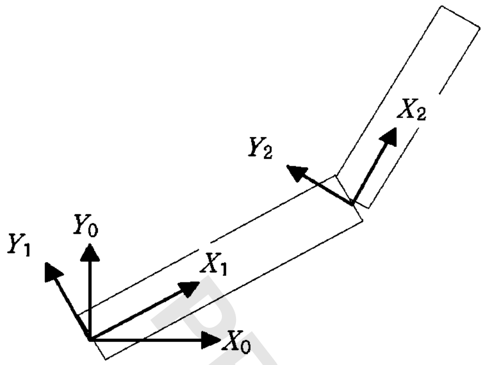
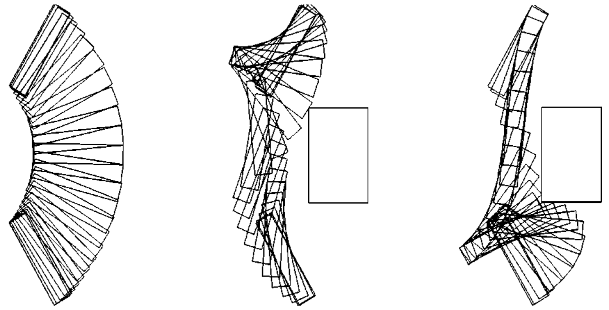


Figure 2. Planar 2-link manipulator.



(a) ignoring obstacle (b) local minimum (A) (c) local minimum (B)

Figure 3. Minimum overload trajectories of last link in example 1.

$$(w_o)_{j+1}=(w_o)_j\sqrt{\frac{(J_o^{ini})_{j+1}/\varepsilon_o}{(J_c^{ini})_{j+1}/\varepsilon_c}}, \quad j=1,\ldots, \qquad (38)$$

where $(J_o^{ini})_{j+1}$ and $(J_c^{ini})_{j+1}$ are those obtained from the initial motions of the $j+1$th search. These initial motions are not the one described in Section 4.4 but the jth minimum-overload trajectories after the motion time has been increased by Eq. (37). A large value for w_o ensures that there is no collision with obstacles, whereas a small value yields faster convergence of the optimization. The tolerances ε_o and ε_c are values of J_o^* and J_c^*, respectively, used to accept a minimum-overload trajectory as an obstacle-free minimum-time motion. Thus, the convergence criteria for obstacle-free minimum-time motions are

$$J_o^* \leq \varepsilon_o, \quad J_c^* \leq \varepsilon_c. \qquad (39)$$

These convergence criteria are tested at the end of every minimum-overload search. In the numerical simulations, 10^{-7} and 10^{-5} are used as ε_o and ε_c, respectively.

6. SIMULATIONS

6.1. Example 1 (Planar 2-Link Manipulator)

The first example is a simple planar 2-link arm shown in Figure 2, where two joints are revolute pairs around their z-axes. The masses, lengths and cross-sections of the first and second links are 25 and 15 kg, 0.8 and 0.6 m, and (0.15, 0.15) and (0.12, 0.12) m, respectively. Gravity is acting in $-Y_0$ direction. $\tau_c=[530\ 90]^T$ Nm, $\omega_c=[6\ 6]^T$ rad/s. The dimensions (l_x, l_y, l_z) and the geometric center of one hexahedral obstacle are (0.4, 0.6, 0.4) and (1.2, 0, 0) m in base coordinates. The manipulator moves from $(-30°, -30°)$ to $(30°, 30°)$ in joint space. The velocities and the accelerations at two end points are zero.

As mentioned in Section 4.3, we performed the global search for the minimum-overload trajectories by changing the location of the seed point of the obstacle. Our algorithm produced two local minima as shown in Figure 3, where the motion of link 2 is shown. When we located the seed point in the right lower corner of the obstacle, the motion converged always to the local minimum (A) regardless of the total motion time. When it was located in the right upper corner and the total motion time is around 0.65 s, the motion converged to the local minimum (B). When ignoring the obstacle, the resultant motion is very simple [Figure 3(a)]. However, if we lowered τ_c, the motion resembles the local minimum (A). The minimum times are 0.69, 0.99 and 1.16 s in Figures 3(a), 3(b) and 3(c), respectively. Figure 4 shows the actuator torques (solid lines) and the equivalent torques (dotted lines) during a minimum-time motion, where joint 2 is almost saturated during the entire motion, and joint 1 is saturated in the latter half.

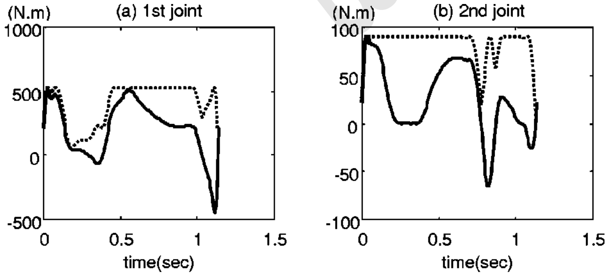


Figure 4. Actuator torques (solid lines) and equivalent torques (dotted lines) of minimum-time motions in local minimum trajectory (B).

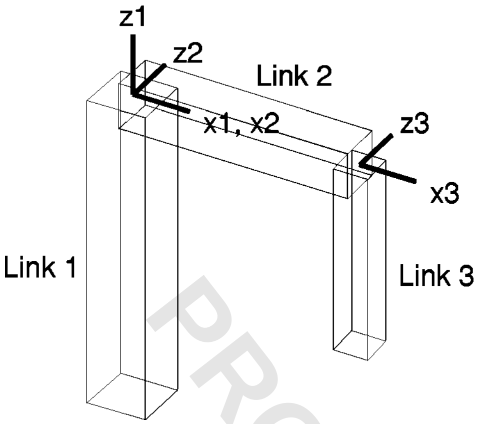


Figure 5. Spatial 3-link manipulator.

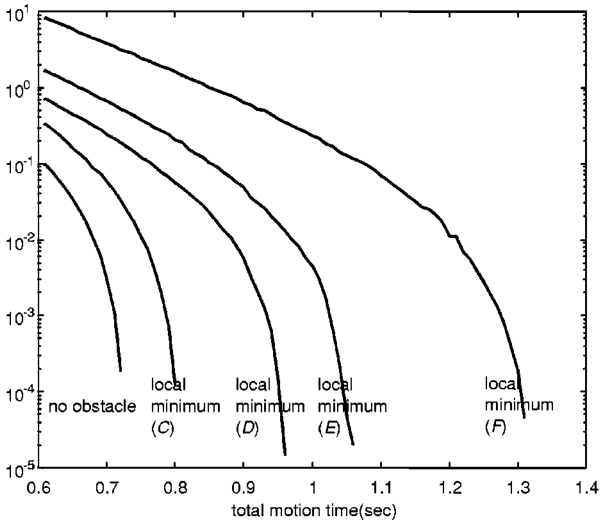


Figure 6. Minimum overloads in example 2.

6.2. Example 2 (Spatial 3-Link Manipulator)

The model is a 3-link arm shown in Figure 5, where all joints are revolute pairs around their z-axes and it is the configurations of zero-displacements. Base coordinates are the same as the first link coordinates. The specifications are listed in Table 2, where gravity is acting in the $-z_0$ direction and τ_c is about twice the static actuator torques necessary to endure gravity in fully stretched configuration. The dimensions of two hexahedral obstacles are both (0.4, 0.4, 0.5) m and the centers are (0.76 −0.47 −0.25) and (0.76 0.47 0.25) m in base coordinates. Their orientations are equal to base coordinates. The seed points of all links are at their geometric centers, but those of all obstacles are located in the corners in order to find various local paths. The manipulator moves from (−60°, 30°, −60°) to (60°, −30°, −120°) in joint space. The velocities and the accelerations at two end points are zero.

We repeated the global search for the minimum-overload trajectories, mentioned in Section 4.3, starting from $T = 0.6$ s and increasing it by 0.01 s. The results in Figure 6 show that the minimum overloads decrease monotonically until they vanish at the minimum times. The convergence was therefore quite stable. This figure also demonstrates that we can find the global optimum trajectory by adjusting the seed points of the obstacles. To show the global search clearly, we aligned the obstacles exactly in the way of the initial motion. We have found four local minima shown in Figure 7 and we can see that the local minimum (C) is the global one whose minimum overload is the least among the four. The minimum times are 0.728 s when ignoring the obstacles, and 0.807, 0.961, 1.08 and 1.32 s in the local minima (C)–(F), respectively. In the case of no obstacles, not shown in Figure

Table II. Specifications of example 2 (SI units).

Links	Masses	l_x, l_y, l_z (local coordinates)	τ_c	ω_c
First	100	0.2, 0.2, 1.0	1000	6
Second	50	0.8, 0.15, 0.15	1000	6
Third	30	0.12, 0.6, 0.12	200	6

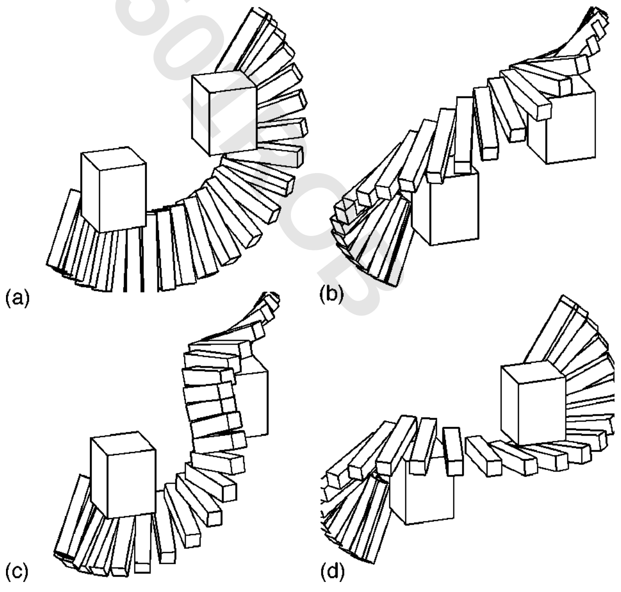


Figure 7. Local minimum trajectories of last link in example 2.

Table III. Convergence data of minimum-time search in example 2.

Iteration no.	Total motion times	Line searches performed	Minimum overloads (J_c^*)	Minimum penalty terms (J_o^*)
1	0.2000	98	156.6	1.338E-4
2	0.7184	87	3.517E-2	1.078E-9
3	0.7817	90	1.606E-3	1.631E-6
4	0.7974	96	2.064E-4	3.958E-7
5	0.8036	104	3.245E-5	5.334E-8
6	0.8060	73	7.373E-6	2.072E-8

7, the manipulator turned around z_1-axis with link 3 bent downward to reduce the moment of inertia about that axis and the moment arm of gravity about z_2- and z_3-axes.

As the total motion time approaches the minimum time, the minimum overload reduces steeply and disappears suddenly at the minimum time. It is a delicate problem to find the exact minimum time efficiently. Table 3 shows convergence data of the minimum-time search mentioned in Section 5. From an initial motion time 0.2 s, only six iterations were performed. At the end of each iteration, the total motion time and w_o were updated using Eqs. (37) and (38), respectively. In all local minima, six or seven iterations were performed to find the trajectories that satisfied the convergence criteria (39).

In the optimal motions, the manipulator touches slightly the surfaces of the obstacles and this may be considered as imperfect obstacle avoidance. The minimum clearance to assure safe avoidance must be added to the actual geometric sizes of obstacles. Figure 8 shows the saturation of the actuators. In the case of no obstacle, joint 2 [dotted line in Figure 8(a)] is not fully saturated. On the other hand, all joints are almost saturated in Figure 8(b) to avoid the obstacles by the local minimum (F).

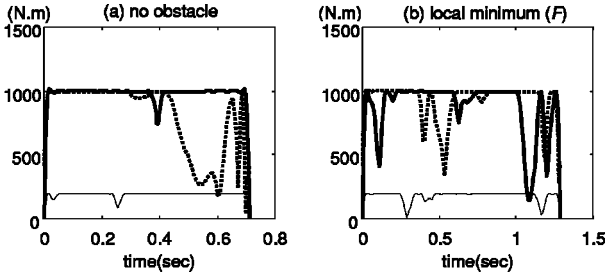


Figure 8. Equivalent torques of minimum-time motions; thick, dotted and thin lines mean first, second, and third joints, respectively.

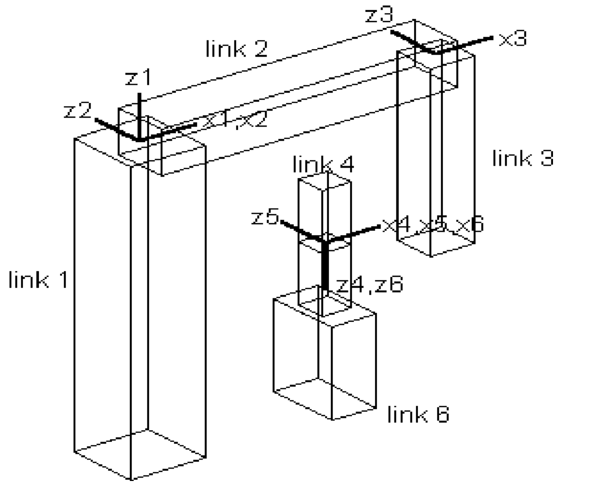


Figure 9. Spatial 6-link manipulator.

6.3. Example 3 (Spatial 6-Link Manipulator)

Figure 9 shows a configuration of a PUMA 560 type manipulator at zero-displacement. All joints are revolute pairs around their z-axes. The base coordinates are the same as the first link coordinates at zero-displacement. Link 4 is connected to link 3. The link parameters in Denavit–Hartenberg notation and the specifications are listed in Tables 4 and 5, respectively, where gravity is acting in $-z_0$ direction and τ_c is about twice the static actuator torques necessary to endure gravity in fully stretched configurations. The mass of the last link includes that of the tool and is heavier than link 5. The dimensions of one hexahedral obstacle are (1.2, 2.0, 1.2) m and the center is (1.2, 0.0, 0.0) m in base coordinates. The orientations of the obstacle are equal to base coordinates. The manipu-

Table IV. Link parameters of example 3 (SI units).

Links	α_{i-1}	a_{i-1}	d_i	θ_i
First	0	0	0	θ_1
Second	−90°	0	0	θ_2
Third	0	0.8	0	θ_3
Fourth	−90°	0	0.8	θ_4
Fifth	90°	0	0	θ_5
Sixth	−90°	0	0	θ_6

Table V. Specifications of example 3 (SI units).

Links	Mass	l_x, l_y, l_z (in link coordinates)	τ_c	ω_c
First	100	0.2, 0.2, 1.0	1500	6
Second	50	0.8, 0.15, 0.15	1500	6
Third	30	0.12, 0.6, 0.12	500	6
Fourth	5	0.08, 0.08, 0.2	75	6
Fifth	5	0.08, 0.2, 0.08	75	6
Sixth	10	0.12, 0.2, 0.3	5	6

lator moves from (20°, 60°, −120°, 0°, −30°, 0°) to (−20°, −60°, −60°, 0°, 30°, 0°) in joint space. The velocities and the accelerations at the start and the goal positions are all zero.

Similar to Figure 6 in example 2, we repeated the global search at every time step. The results are shown in Figure 10. When ignoring the obstacle, the minimum overload decreases gradually until it vanishes at the minimum time 0.900 s. We can see that the convergence is quite stable and it converged to only one optimal trajectory regardless of the total motion time. It is shown in Figure 11, where four frames are 1st, 6th, 11th and 16th ones among totally 16 equal time-interval frames. We can see in this motion that the manipulator turns around z_1-axis with the last link bent downward to reduce the moment of inertia about that axis and the moment arm of gravity about z_2- and z_3-axes. We also observe that the manipulator maximizes the joint coupling effect by the overactions of the underloaded joints to reduce the torques of the overloaded joints. Here, joint 1 is underloaded and joints 2 and 3 are overloaded. (We can also see this in Figure 16.)

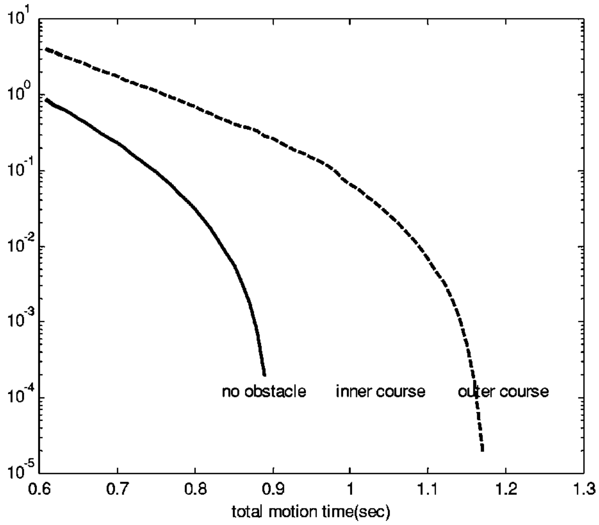


Figure 10. Minimum overloads in example 3.

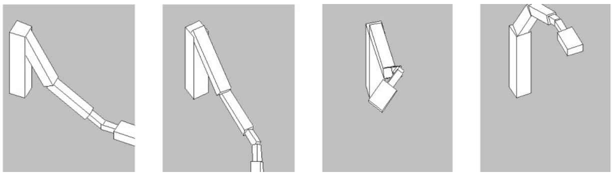


Figure 11. Minimum time motion ignoring the obstacle.

By adjusting the location of seed point of the obstacle, we have found two obstacle-free courses, namely, the inner and the outer courses as shown in Figures 12, 13 and 14, 15, respectively, where, each four frames are 1st, 6th, 11th and 16th ones among totally 16 equal time-interval frames. The convergence properties in Figure 10 were not as stable as those in Figure 6. They converged to different local minima, with the main difference being the bending direction of joint 3.

In the inner course, the local minimum shown in Figure 12 has less minimum overload than Figure 13. We can see this fact in Figure 10, where the dotted curve seems to be composed of several segments of two parallel curves. The difference in the minimum overloads becomes smaller as the total motion time approaches the minimum time. Both the minimum times are equally 1.12 s. In the outer course, the local minimum shown in Figure 14 has less minimum overload than Figure 15. The difference is quite small in Figure 10, but it becomes larger as the total motion time approaches the minimum time. The minimum times of Figures 14 and 15 are 1.17 s and 1.21 s, respectively. Figure 16 shows the saturation state of the actuators. Joints 2, 3 and 5 are overloaded in the initial motions, where maximum equivalent torques are twice over the actuator limits. After the optimization, the motion time is increased from 0.8 to 0.900 s in the case of no obstacle and increased to 1.12 s to avoid the obstacle. Moreover, all the joints are close to saturation during the minimum-time motions. This is consistent with Pontryagin's maximum principle, since a

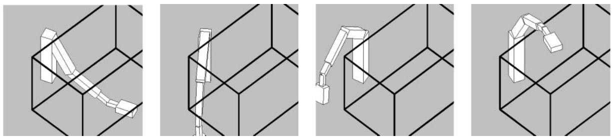


Figure 12. Local minimum trajectory (G) in inner course.

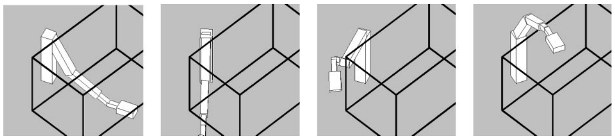


Figure 13. Local minimum trajectory (H) in inner course.

necessary condition for the minimum-time motion (assuming no singular arcs) is that all the joints be in saturation during the motions.

7. CONCLUSIONS

In this paper, we present a practical and reliable method for finding minimum-time motions for manipulators moving in an obstacle field subject to the limits of velocity-dependent actuator forces. Arbitrary point-to-point manipulator motions are represented by a point in a finite dimensional vector space parametrized by quintic B-splines. The novel idea in this work is the concept of the minimum-overload trajectory, in which the actuator-overloads achieve their minimum values with the motion time fixed. The minimum-time motion is defined rigorously with this concept and it is obtained by a successive search for the minimum-overload trajectory.

In the resultant minimum-time motions of planar 2-link, spatial 3-link and 6-link manipulators we have the following: (1) Consistent with Pontryagin's maximum principle, almost all actuators are close to saturation during the motion. (2) The manipulator maximizes the joint coupling effect by the overactions of underloaded joints to reduce the torques on the overloaded joints. (3) The manipulator turns around the vertical z_1-axis with the last link bent in order to reduce the moment of inertia about that axis and torques due to gravity.

8. APPENDIX

8.1. Reduced Coefficient Matrix

Using (15), (17), (18) in Section 3, the initial and the final motion conditions, (11)–(13), become

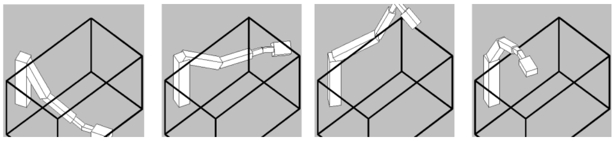


Figure 14. Local minimum trajectory (I) in outer course.

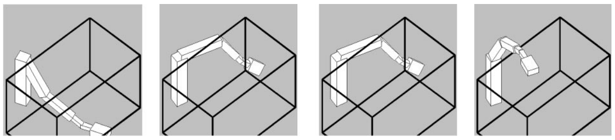


Figure 15. Local minimum trajectory (J) in outer course.

$$[\theta_s \quad \omega_s/\beta \quad \alpha_s/\beta^2 \quad \theta_f \quad \omega_f/\beta \quad \alpha_f/\beta^2] = \mathbf{C}[\mathbf{B}(0) \quad \mathbf{B}'(0) \quad \mathbf{B}''(0) \quad \mathbf{B}(k) \quad \mathbf{B}'(k) \quad \mathbf{B}''(k)]. \tag{A1}$$

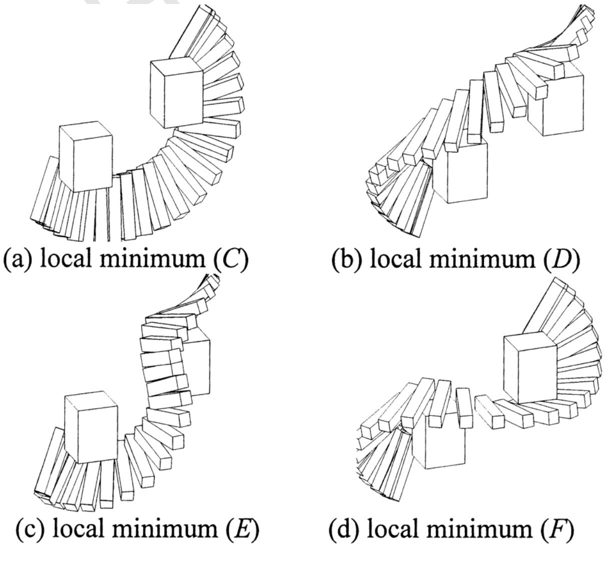
Using the nodal values of Table 1 in Section 3:

$$\mathbf{B}(0) = (\tfrac{1}{120})[1 \quad 26 \quad 66 \quad 26 \quad 1 \quad 0 \quad \ldots \quad 0]^T, \tag{A2}$$

$$\mathbf{B}'(0) = (\tfrac{1}{24})[-1 \quad -10 \quad 0 \quad 10 \quad 1 \quad 0 \quad \ldots \quad 0]^T, \tag{A3}$$

$$\mathbf{B}''(0) = (\tfrac{1}{6})[1 \quad 2 \quad -6 \quad 2 \quad 1 \quad 0 \quad \ldots \quad 0]^T, \tag{A4}$$

$$\mathbf{B}(k) = (\tfrac{1}{120})[0 \quad \ldots \quad 0 \quad 1 \quad 26 \quad 66 \quad 26 \quad 1]^T, \tag{A5}$$



(a) local minimum (C)

(b) local minimum (D)

(c) local minimum (E)

(d) local minimum (F)

Figure 16. Equivalent torques in example 3; solid lines are minimum-time motions in local inimum (G) and dotted lines are initial motions when total motion time is 0.8 s.

$$\mathbf{B}'(k)=(\tfrac{1}{24})[0 \quad \dots \quad 0 \quad -1 \quad -10 \quad 0 \quad 10 \quad 1]^T, \tag{A6}$$

$$\mathbf{B}''(k)=(\tfrac{1}{6})[0 \quad \dots \quad 0 \quad 1 \quad 2 \quad -6 \quad 2 \quad 1]^T. \tag{A7}$$

Substituting (A2)–(A7) for the right side of (A1),

$$[\mathbf{A}_s \quad \mathbf{A}_f]=[\mathbf{C}_s \quad \mathbf{C}_r \quad \mathbf{C}_f]\begin{bmatrix} \mathbf{B}_c & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_c \end{bmatrix}, \tag{A8}$$

where $\mathbf{C}_s$, $\mathbf{C}_f$ are the first and the last five columns ($n\times 5$) of matrix $\mathbf{C}$, respectively, $\mathbf{C}_r$ is the remaining columns ($n\times k-5$), and

$$\mathbf{A}_s=[\theta_s \quad \omega_s/\beta \quad \alpha_s/\beta^2], \tag{A9}$$

$$\mathbf{A}_f=[\theta_f \quad \omega_f/\beta \quad \alpha_f/\beta^2], \tag{A10}$$

$$\mathbf{B}_c=\begin{bmatrix} \frac{1}{120} & - & \frac{1}{6} \\ \frac{13}{60} & -\frac{5}{12} & \frac{1}{3} \\ \frac{11}{20} & 0 & -1 \\ \frac{13}{60} & \frac{5}{12} & \frac{1}{3} \\ \frac{1}{120} & \frac{1}{24} & \frac{1}{6} \end{bmatrix}. \tag{A11}$$

By matrix partition, (A8) becomes

$$\mathbf{A}_s=\mathbf{C}_s\mathbf{B}_c, \tag{A12}$$

$$\mathbf{A}_f=\mathbf{C}_f\mathbf{B}_c. \tag{A13}$$

Equations (A12) and (A13) are partitioned again as

$$\mathbf{A}_s=\mathbf{C}_{s1}\mathbf{B}_{c1}+\mathbf{C}_{s2}\mathbf{B}_{c2}, \tag{A14}$$

$$\mathbf{A}_f=\mathbf{C}_{f1}\mathbf{B}_{c3}+\mathbf{C}_{f2}\mathbf{B}_{c4}, \tag{A15}$$

where $\mathbf{C}_{s1}$, $\mathbf{C}_{s2}$ are the first three and the last two columns of $\mathbf{C}_s$; $\mathbf{C}_{f1}$, $\mathbf{C}_{f2}$ are the first two and the last three columns of $\mathbf{C}_f$, respectively; $\mathbf{B}_{c1}$, $\mathbf{B}_{c2}$ are the first three and the last two rows of $\mathbf{B}_c$; and $\mathbf{B}_{c3}$, $\mathbf{B}_{c4}$ are the first two and the last three rows of $\mathbf{B}_c$, respectively.

Since $\mathbf{B}_{c1}$, $\mathbf{B}_{c4}$ are nonsingular, we can rewrite (A14) and (A15) as

$$\mathbf{C}_{s1}=\mathbf{A}_s\mathbf{B}_{c1}^{-1}-\mathbf{C}_{s2}\mathbf{B}_{c2}\mathbf{B}_{c1}^{-1}, \tag{A16}$$

$$\mathbf{C}_{f2}=\mathbf{A}_f\mathbf{B}_{c4}^{-1}-\mathbf{C}_{f1}\mathbf{B}_{c3}\mathbf{B}_{c4}^{-1}. \tag{A17}$$

By matrix partition, (15) in Section 3 can be written as

$$\theta(s)=\mathbf{C}_{s1}\mathbf{B}_{s1}(s)+\mathbf{C}_{s2}\mathbf{B}_{s2}(s)+\mathbf{C}_r\mathbf{B}_r(s)+\mathbf{C}_{f1}\mathbf{B}_{f1}(s) + \mathbf{C}_{f2}\mathbf{B}_{f2}(s), \tag{A18}$$

where $\mathbf{B}_{s1}(s)$ (3×1), $\mathbf{B}_{s2}(s)$ (2×1), $\mathbf{B}_r(s)$ $(k-5\times 1)$, $\mathbf{B}_{f1}(s)$ (2×1) and $\mathbf{B}_{f2}(s)$ (3×1) are the corresponding row partitions of $\mathbf{B}(s)$.

Substituting (A16) and (A17) for (A18) and rearranging yields

$$\theta(s)=\mathbf{F}_s(s)+\mathbf{C}_m\mathbf{B}_m(s)+\mathbf{F}_f(s), \tag{A19}$$

where

$$\mathbf{F}_s(s)=\mathbf{A}_s\mathbf{B}_{c1}^{-1}\mathbf{B}_{s1}(s), \tag{A20}$$

$$\mathbf{C}_m=[\mathbf{C}_{s2} \quad \mathbf{C}_r \quad \mathbf{C}_{f1}], \tag{A21}$$

$$\mathbf{B}_m(s)=\left\{ \begin{matrix} \mathbf{B}_{s2}(s)-\mathbf{B}_{c2}\mathbf{B}_{c1}^{-1}\mathbf{B}_{s1}(s) \\ \mathbf{B}_r(s) \\ \mathbf{B}_{f1}(s)-\mathbf{B}_{c3}\mathbf{B}_{c4}^{-1}\mathbf{B}_{f2}(s) \end{matrix} \right\} \tag{A22}$$

$$\mathbf{F}_f(s)=\mathbf{A}_f\mathbf{B}_{c4}^{-1}\mathbf{B}_{f2}(s), \tag{A23}$$

where the boundary condition splines, $\mathbf{F}_s(s)$, $\mathbf{F}_f(s)$, whose dimensions are $(n\times 1)$, can be calculated from the initial and the final motion conditions. $\mathbf{C}_m$ is the *reduced coefficient matrix* $(n\times k-1)$ and $\mathbf{B}_m(s)$ is the *reduced B-spline bases functions* $(k-1\times 1)$.

8.2. Initial Fifth Degree Polynomials

$$\theta_p(s)=\mathbf{a}s^5+\mathbf{b}s^4+\mathbf{c}s^3+\mathbf{d}s^2+\mathbf{e}s+\mathbf{f}, \tag{A24}$$

$$\mathbf{a}=\frac{\alpha_f-\alpha_s}{2\beta^2k^3}-\frac{3\omega_f+3\omega_s}{\beta k^4}+\frac{6\theta_f-6\theta_s}{k^5}, \tag{A25}$$

$$\mathbf{b}=-\frac{2\alpha_f-3\alpha_s}{2\beta^2k^2}+\frac{7\omega_f+8\omega_s}{\beta k^3}-\frac{15\theta_f-15\theta_s}{k^4}, \tag{A26}$$

$$\mathbf{c}=\frac{\alpha_f-3\alpha_s}{2\beta^2k}-\frac{4\omega_f+6\omega_s}{\beta k^2}+\frac{10\theta_f-10\theta_s}{k^3}, \tag{A27}$$

$$\mathbf{d}=\frac{\alpha_s}{2\beta^2}, \tag{A28}$$

$$\mathbf{e}=\frac{\omega_s}{\beta}, \tag{A29}$$

$$\mathbf{f}=\theta_s. \tag{A30}$$

REFERENCES

1. J.E. Bobrow, S. Dubowsky, and J.S. Gibson, Time-optimal control of robotic manipulators along specified paths, Int J Robot Res 4 (1985), 3–17.
2. K.G. Shin and N.D. McKay, Minimum-time control of robotic manipulators with geometric path constraints, IEEE Trans Autom Control AC-30 (1985), 531–541.
3. Z. Shiller, On singular time-optimal control along specified paths, IEEE Trans Robot Autom 10 (1994), 561–566.
4. D. Constantinescu and E.A. Croft, Smooth and time-optimal trajectory planning for industrial manipulators along specified paths, J Robot Syst 17 (2000), 233–249.
5. H.H. Tan and R.B. Potts, A discrete trajectory planner for robotic arms with six degrees of freedom, IEEE Trans Robot Autom 5 (1989), 681–690.
6. A.E. Bryson, Jr. and E.B. Meier, Efficient algorithm for time-optimal control of a two-link manipulator, J Guid Control Dyn 13 (1990), 859–866.
7. G. Bessonnet and J.P. Lallemand, On the optimization of robotic manipulator trajectories with bounded joint actuators or joint kinetic loads considered as control variables, Trans ASME J Dyn Syst Measure Control 116 (1994), 819–826.
8. A.M. Formalsky, The time-optimal control of the bending of a plane two-link mechanism, J Appl Math Mech 60 (1996), 243–251.
9. M. Galicki, The planning of robotic optimal motions in the presence of obstacles, Int J Robot Res 17 (1998), 248–259.
10. R.G. Fenton, B. Benhabib, and A.A. Goldenberg, Optimal point-to-point motion control of robots with redundant degrees of freedom, Trans ASME J Eng Ind 108 (1986), 120–126.
11. J.E. Bobrow, Optimal robot path planning using the minimum-time criterion, IEEE J Robot Autom 4 (1988), 443–450.
12. S.K. Singh and M.C. Leu, Manipulator motion planning in the presence of obstacles and dynamic constraints, Int J Robot Res 10 (1991), 171–186.
13. M.K. Jouaneh, D.A. Dornfeld, and M. Tomizuka, Trajectory planning for coordinated motion of a robot and a positioning table: part 2, IEEE Trans Robot Auto 6 (1990), 746–759.
14. H. Ozaki and C-j. Lin, Optimal B-spline joint trajectory generation for collision-free movements of a manipulator under dynamic constraints, IEEE Int Conf on Robotics Automation, 1996, pp. 3592–3597.
15. Z. Shiller and S. Dubowsky, On computing the global time-optimal motions of robotic manipulators in the presence of obstacles, IEEE Trans Robot Autom 7 (1991), 785–797.
16. M. Schlemmer and G. Gruebel, Real-time collision-free trajectory optimization of robot manipulators via semi-infinite parameter optimization, Int J Robot Res 17 (1998), 1013–1021.
17. C.W.J. Hol, L.G. Willigenburg, E.J. Henten, and G. Straten, A new optimization algorithm for singular and non-singular digital time-optimal control of robots, in IEEE Int Conf on Robotics and Automation 2001, pp. 1136–1141.
18. J.E. Bobrow, B. Martin, G. Sohl, E.C. Wang, F.C. Park, and J. Kim, Optimal robot motions for physical criteria, J Robot Syst 18 (2001), 785–795.
19. C-Y.E. Wang, W.K. Timoszyk, and J. E. Bobrow, Payload maximization for open chained manipulators: finding weightlifting motions for a Puma 762 robot, IEEE Trans Robot Autom 17 (2001), 218–224.
20. J. Lee, Dynamic programming approach to near minimum-time trajectory planning for two robots, IEEE trans Robot Autom 11 (1995), 160–164.
21. K.G. Shin and Q. Zheng, Minimum-time collision-free trajectory planning for dual-robot systems, IEEE Trans Robot Autom 8 (1992), 641–644.
22. B. Cao, G.I. Dodds, and G.W. Irwin, A practical approach to near time-optimal inspection-task-sequence planning for two cooperative industrial robot arms, Int J Robot Res 17 (1998), 858–867.
23. C.J. Ong and E.G. Gilbert, Growth distances: new measures for object separation and penetration, IEEE Trans Robot Autom 12 (1996), 888–903.
24. J.J. Craig, Introduction to robotics; mechanics and control, Addison-Wesley, Reading, MA, 1986.
25. M.J. Best and K. Ritter, Linear programming: active set analysis and computer programs, Prentice–Hall, Englewood Cliffs, NJ, 1985.
26. P.M. Prenter, Splines and variational methods, J Wiley, New York, 1975.
27. R. Fletcher, Practical method of optimization, 2nd ed., J Wiley, New York, 1987.