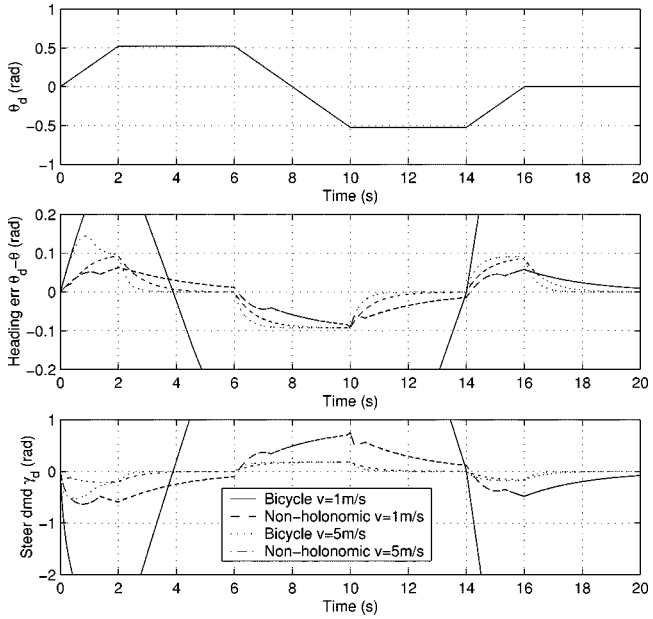

Fig. 7. Simulated steering response for bicycle and nonholonomic models for plant with no rate or saturation limits and $K = 10$ and for two speeds $v = 1, 5$ m/s.

to a nonholonomic constraint. The bicycle and exact models were then compared in the context of a heading control loop where the effect of the zero on error and input demand is quite marked, particularly at low speed. As vehicle speed increases, the kinematics tend toward those of a bicycle. However, at low speed, such as required when turning tight corners, the difference is significant and given the confined environment must be taken into account.

ACKNOWLEDGMENT

The first author would like to thank C. Altafini, KTH Stockholm, for discussions on this topic. Experimental data was provided by AMIRA sponsored project P517 and the CSIRO LHD project team: G. Winstanley, J. Roberts, E. Duff, P. Sikka, L. Overs, S. Wolfe, R. McCasker, and J. Cunningham.

REFERENCES

[1] J. Cunningham, P. Corke, H. Durrant-Whyte, and M. Dalziel, "Automated LHD's and underground haulage trucks," *Australian Journal of Mining*, 1999.

[2] L. Bloomquist and J. P. Huissoon. (1996) Navigation system for autonomous underground mining vehicles. [Online]. Available: http://sail.uwaterloo.ca/ labloomq/workshop/info.html

[3] J. D. Lane, "Automatic steering system for an underground mine haul truck," Master's, Colorado School of Mines, 1992.

[4] G. Eriksson and A. Kitok, "Automatic loading and dumping using vehicle guidance in a Swedish mine," in *International Symposium on Mine Mechanization and Automation*, Colorado, June 1991, pp. 15.33–15.40.

[5] J. Roberts and P. Corke, "Automation of underground truck haulage," in *International Symposium on Mine Mechanization and Automation*, Brisbane, July 1997, pp. b5.23–32.

[6] C. Altafini, "Why to use an articulated vehicle in underground mining operations?," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 1999, pp. 3020–3025.

[7] J. Steele, C. Ganesh, and A. Kleve, "Control and scale model simulation of sensor-guided LHD mining machines," *IEEE Trans. Industry Applications*, vol. 29, Nov 1993.

[8] A. Hemami and V. Polotski, "Path tracking control problem formulation of an lhd loader," *Int. J. Robot. Res.*, vol. 17, pp. 193–198, Feb 1998.

[9] S. Scheding, G. Dissanyake, and E. M. N. H. Durrant-Whyte, "An experiment in autonomous navigation of an underground mining vehicle," *IEEE Trans. Robot. Autom.*, vol. 15, pp. 85–95, Feb 1999.

[10] R. DeSantis, "Modeling and path-tracking for a load-haul-dump mining vehicle," *Journal of Dynamic Systems, Measurement and Control*, vol. 119, pp. 40–47, Mar 1997.

[11] C. Altafini, "A path-tracking criterion for an lhd articulated vehicle," *Int. J. Robotics Research*, vol. 18, pp. 435–441, May 1999.

[12] P. Bolzern, R. DeSantis, A. Locatelli, and S. Togno, "Dynamic model of a two-trailer articulated vehicle subject to nonholonomic constraints," *Robotica*, vol. 14, pp. 445–450, 1996.

[13] A. Hemami and V. Polotski, "Dynamics of a dual-unit articulated vehicle for path tracking control problem formulation," in *Proc. Int. Conf. Control Applications*, Hartford, CT, Oct 1997, pp. 173–176.

Payload Maximization for Open Chained Manipulators: Finding Weightlifting Motions for a Puma 762 Robot

Chia-Yu E. Wang, Wojciech K. Timoszyk, and James E. Bobrow

***Abstract*—Although the dynamic equations of motion of open-chained robot systems are well known, they are seldom taken into account during the planning of motions. In this work, we show that the dynamics of a robot can be used to produce motions that extend the payload capability beyond the limit set by traditional methods. In particular, we develop a point-to-point weightlifting motion planner for open-chained robots. The governing optimal control problem is converted into a direct, SQP parameter optimization in which the gradient is determined analytically. The joint trajectories are defined by B-spline polynomials along with a time-scale factor. The algorithm is applied to a Puma 762 robot, with its physical limitations incorporated into the formulation. The torque limits are formulated as soft constraints added into the objective function while the position and velocity limits are formulated as hard, linear inequality constraints, on the parameters. The solutions obtained with our algorithm extend the robot's payload capability while reducing the joint torques. Interestingly, nearly all the trajectories found pass through singular configurations, where large internal forces from the robot are applied to the payload and little torque is needed from the motors. A video file of the resulting motions can be found at http://www.eng.uci.edu/~chwang/project/puma762.html.**

***Index Terms*—B-spline, dynamic motion planning, open chained manipulator, optimal robot control, optimization.**

I. INTRODUCTION

The majority of current robot motion planning algorithms, such as those found in well-known texts (see, e.g., [10]), ignore the dynamics of the robot during the motion planning phase. In this paper, we show how the dynamics of a manipulator can be used to extend its payload capability beyond the limits established by the manufacturer. The goal of the research was to develop an algorithm that maximizes the robot payload while taking into account realistic constraints such as joint torque limits and velocity bounds. Our solutions also minimize joint torques

Manuscript received August 7, 2000. This paper was recommended for publication by Associate Editor I. Walker and Editor A. De Luca upon evaluation of the reviewers' comments. The work was supported by the National Science Foundation under Grant IIR-9711782. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Detroit, MI, May 1999.

The authors are with the Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA 92697 USA (e-mail: chwang@eng.uci.edu; wtimoszy@eng.uci.edu; bobrow@eng.uci.edu).

Publisher Item Identifier S 1042-296X(01)04813-3.

for a given load, thereby reducing wear on the robot. This work extends our previous results [11], where effort was minimized for systems with end-effector constraints. The unique aspects of our approach are as follows: 1) the payload is factored directly into the cost function for the motion; 2) the semi-infinite joint velocity bounds are transformed to a finite set of *linear constraints on the parameters*; 3) a time-scale factor and its gradient are incorporated into the formulation; 4) a linear factorization of the dynamic system parameters is developed using the geometric operators of Lie groups and Lie algebras; and 5) to the best of our knowledge, this is the first successful application of minimum-effort optimal control to a weight-lifting problem for a real robot.

We assume that the desired motion can be characterized as the solution to a general optimal control problem of the form $\mathrm{Min}_{\tau(t)}\ J_c = \Phi(q,\dot{q},t_f) + \int_0^{t_f} L(q,\dot{q},\tau,t)dt$ subject to the dynamic equations of motion and bounds on position, velocity, and torque. The specific cost functional that we used for payload maximization will be given in the next section. The final time t_f may be either free or fixed in our formulation. Because of the importance of finding solutions to optimal control problems of the above form, there are several numerical algorithms and commercial software packages available to solve them. All of the competitive approaches incorporate some form of Newton iteration on a parameterized version of the problem. Because the problems are nonlinear, the Newton iteration will converge to a local minimizer. See [3] for a review on progress made in this area since the 1960's. Algorithms used to solve optimal control problems are broadly classified as either indirect methods or direct methods. Indirect methods explicitly solve the optimality conditions stated in terms of the maximum principle [4], the adjoint equations, and the transversality (boundary) conditions. These conditions can be derived using the calculus of variations, where the first variation of the Hamiltonian function is set to zero. A number of multiple shooting methods have been developed to solve the two-point boundary value problem defined by the equations.

Although the indirect approach described above has been used successfully in many applications, it has been replaced by direct methods in recent years. The primary reasons are that the region of convergence is relatively small and that it is difficult to incorporate path inequality constraints. We chose a direct approach for the work reported in this paper. The basic idea is to parameterize the joint trajectories, and to vary the parameters in a nonlinear optimization until a local minimum that satisfies the constraints have been achieved. The joint trajectories used in the motion planning problem can be represented in many ways. Among them are C^2-splines and B-spline polynomials, which have been used by many robotics researchers [5]–[7], [9], [11], [13], [15], [17]. B-spline polynomials provide local control of the joint trajectory and they have the important convex hull property [2]. In this paper, we exploit the convex hull property by transforming the semi-infinite constraints that arise in the problem formulation into finite-dimensional ones. In order to vary the total time taken for the motion, we have added a time-scale factor. This allows us to solve a free-final-time problem in which the derivatives of the joint positions with respect to the time-scale factor can be derived analytically.

By parameterizing the joint trajectories, the optimal control problem becomes a parameter optimization one with the parameters being the time-scale factor and the control vertices of the splines. We add another parameter that represents the payload into the optimization in order to maximize it. Because the Puma 762 has bounded joint velocities, these semi-infinite constraints are included in our formulation. Other researchers have included velocity constraints as penalty functions [5], [13], [15], or have discretized them into a large number of inequality constraints throughout the time interval [6], or have used a variable mesh approach [7] to handle them. As mentioned above, our approach uses the convex hull property of B-splines in order to transform these semi-infinite constraints into a set of linear inequalities in the parameter space. This provides superior numerical stability since the semi-infinite constraints on joint velocity are satisfied exactly with a small number linear constraints on the parameters.

II. Formulation of the Optimization Problem

Consider an n-degree-of-freedom (DOF) open-chained manipulator. The specific cost functional that we are interested in minimizing is

$$\underset{\tau(t),p_w}{\text{Minimize}} \quad J_c = -wp_w + \frac{1}{2}\int_0^{p_t t_f} \tau^{\mathrm{T}} W_e \tau dt \tag{1}$$

$$\text{subject to} \quad H(q)\ddot{q} + h(q,\dot{q}) = \tau \tag{2}$$

$$\underline{\tau}_i \le \tau_i(t) \le \overline{\tau}_i \tag{3}$$

$$\underline{q}_i \le q_i(t) \le \overline{q}_i \tag{4}$$

$$\underline{\dot{q}}_i \le \dot{q}_i(t) \le \overline{\dot{q}}_i \tag{5}$$

$$q(0) = q_0 \qquad \dot{q}(0) = 0 \tag{6}$$

$$q(p_t t_f) = q_f \qquad \dot{q}(p_t t_f) = 0. \tag{7}$$

Equation (2) represents the dynamics for the open-chained manipulator with the joint coordinates $q \in \Re^n$ and the joint forces or torques $\tau \in \Re^n$, where $H(q)$ is the generalized mass matrix and $h(q,\dot{q})$ contains the centrifugal, Coriolis, and gravitational forces and the joint friction. Constraints on the joint torque τ, position (displacement) q and velocity $\dot{q}$ are represented by (3)–(5), respectively, where $\underline{\tau}$, $\overline{\tau}$, $\underline{q}$, $\overline{q}$, $\underline{\dot{q}}$, and $\overline{\dot{q}} \in \Re^n$ are assumed to be given for a particular manipulator. The subscript i in (3)–(5) is the ith component of the corresponding n-vector, so these inequalities represent a total of $6n$ semi-infinite constraint equations of the form $g(q(t),\dot{q}(t),\tau(t)) \le 0$ for $t \in [0, p_t t_f]$.

The cost functional (1) explicitly takes into account the manipulator payload p_w, which is scaled by the factor $w > 0$. The payload is a variable parameter in our formulation. The effect of increasing w places more emphasis on the weight lifted by the manipulator with less emphasis on the actuator effort, which is the term on the right-hand side of (1). Note that increasing p_w decreases the left-hand term in the cost functional, but it simultaneously increases the joint torques required for the motion; so the right-hand term in the cost functional increases. The matrix $W_e > 0$ is the weighting factor that allows us to incorporate the effects of joint coupling and gear ratios in the Puma 762 robot. We obtain a free-final-time problem by fixing t_f and varying p_t as an additional parameter. The parameter p_t scales the nominal trajectory uniformly over the time interval $[0, t_f]$. Note that if t_f is too small, there may be no admissible solution to the optimal control problem since the torque constraints indirectly bound the path traversal time.

A. Joint Trajectory Parameterization with B-Splines

In order to set up the optimal control problem for a numerical solution, the joint trajectories are interpolated by the piecewise $k-1$th order and B-spline polynomials over the knot space of an ordered time sequence.

Let the knot sequence be $0 = t_0 = \cdots = t_{k-1} \le t_k \le \cdots \le t_m \le t_{m+1} = \cdots = t_{m+k} = t_f$ with $m \ge k-1$. The joint trajectories $q \in \Re^n$ are then written as

$$q(t,P) = \sum_{j=0}^{m} p^j B_{j,k}(t) \tag{8}$$

where $P = \{p^0, \ldots, p^m\}$, with $p^j \in \Re^n$, are the control vertices and $B_{j,k}$ is the B-spline basis function with C^{k-2} continuity. The B-spline can be written in a piecewise manner as follows:

$$B_{j,k}(t) = \begin{cases} g_{jl}(t), & \text{if } t_{j+l} \le t < t_{j+l+1} \\ & (l = 0 \sim k-1) \\ 0, & \text{if } t < t_j \text{ or } t \ge t_{j+k} \end{cases} \tag{9}$$

where g_{jl} is a polynomial of order $k-1$. In this paper uniform quintic B-splines ($k = 6$) were used to interpolate the joint trajectories over

the knot sequence defined above with $t_j - t_{j-1} = (tf/(m-k+2))$ for $j = k, k+1, \ldots, m+1$. More discussion on the choice of m is given in the section on optimization results.

The basis function can also be computed by using the following recursive formulation [2].

Algorithm II.1: The recursion for the B-spline is

$$B_{j,k}(t) = \frac{t - t_j}{t_{j+k-1} - t_j} B_{j,k-1}(t) + \frac{t_{j+k} - t}{t_{j+k} - t_{j+1}} B_{j+1,k-1}(t) \tag{10}$$

starting with

$$B_{j,1}(t) = \begin{cases} 1, & \text{if } t_j \le t < t_{j+1} \\ 0, & \text{otherwise.} \end{cases}$$

The derivative of $B_{j,k}$ can be obtained by noting

$$\frac{\partial}{\partial t} B_{j,k}(t) = (k-1)\left[\frac{B_{j,k-1}(t)}{(t_{j+k-1} - t_j)} - \frac{B_{j+1,k-1}(t)}{(t_{j+k} - t_{j+1})}\right]. \tag{11}$$

Hence, the time-derivative of q is

$$\frac{d}{dt} q(t) = \sum_{j=0}^{m} p^j \frac{\partial}{\partial t} B_{j,k}(t) = \sum_{j=0}^{m+1} (k-1) \frac{p^j - p^{j-1}}{t_{j+k-1} - t_j} B_{j,k-1}(t) \tag{12}$$

where p^{-1} and p^{m+1} are defined as zero. The higher-order derivatives of $q(t)$ can be computed following a similar procedure.

Two useful properties for the use of B-splines in our solution to the optimal control problem will now be developed. The first is as follows.

Property II.1: If $\underline{q}_i \le p_i^j \le \overline{q}_i$, then $\underline{q}_i \le q_i(t) \le \overline{q}_i$.

This property is established by using convex hull property of B-splines $B_{j,k}(t) \ge 0$ and $\sum_{j=0}^{m} B_{j,k}(t) = 1$, and noting that for $p_i^j \le \overline{q}_i$

$$q_i(t) = \sum_{j=0}^{m} p_i^j B_{j,k}(t) \le \sum_{j=0}^{m} \overline{q}_i B_{j,k}(t) = \overline{q}_i.$$

The other inequality is obtained similarly by starting with the assumption $\underline{q}_i \le p_i^j$. In a manner similar to the above, and using the relations developed in (12), the joint velocity constraints are determined by the following property.

Property II.2: If $\underline{\dot{q}}_i \le (k-1)(p_i^j - p_i^{j-1})/(t_{j+k-1} - t_j) \le \overline{\dot{q}}_i$, then $\underline{\dot{q}}_i \le \dot{q}_i(t) \le \overline{\dot{q}}_i$.

These two properties are useful for solving the optimal control problem because they show that *semi-infinite dimensional constraints on joint position and velocity can be replaced by linear constraints in the parameter space*.

B. Optimal Control via Direct Parameter Optimization

In order to evaluate the cost function in (1), we see that the joint torques $\tau(t)$ are needed. They are given by (2) since $q(t)$, $\dot{q}(t)$, and $\ddot{q}(t)$ are known from the parameterization of the joint trajectories. We use the $O(n)$ recursive dynamics algorithm in Appendix A to evaluate $\tau(t)$. In order to perform the optimization, an initial feasible trajectory is required. For the problems in this paper, there are no obstacles in the workspace so the initial path planning is trivial. For problems with obstacles, a penalty would need to be added to the cost function and an initial path could be determined from a path planning algorithm (see [10] and [12], for example). We assume that an initial motion has been given, and that it is therefore defined by the parameter set P, with $q = q(t, P)$. In addition, we introduce a time-scale parameter p_t, so that $\hat{t} = p_t t$, $q_s(\hat{t}) \equiv q(\hat{t}/p_t, P)$, and $\dot{q}_s \equiv (dq_s/d\hat{t})$. The optimal control problem (1) is then transformed into the following discrete parameter optimization:

$$\underset{p_w, p_t, P}{\text{Minimize}} \quad J_{cp} = \underbrace{-wp_w + \frac{1}{2}\int_0^{p_t t_f} \tau^{\mathrm{T}} W_e \tau d\hat{t}}_{J_c} + \underbrace{\frac{1}{2}\int_0^{p_t t_f} \phi^{+^{\mathrm{T}}} W_p \phi^+ d\hat{t}}_{J_p} \tag{13}$$

$$\text{subject to} \quad H(q_s)\ddot{q}_s + h(q_s, \dot{q}_s) = \tau \tag{14}$$

$$\underline{q}_i \le p_i^j \le \overline{q}_i \tag{15}$$

$$\underline{\dot{q}}_i \le \frac{(k-1)(p_i^j - p_i^{j-1})}{\hat{t}_{j+k-1} - \hat{t}_j} \le \overline{\dot{q}}_i \tag{16}$$

$$p^0 = p^1 = q_0 \tag{17}$$

$$p^{m-1} = p^m = q_f \tag{18}$$

$$\text{with} \quad \phi_i^+ = \begin{cases} \phi_i, & \text{if } \phi_i > 0 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

$$\phi = \begin{bmatrix} \underline{\tau} - \tau \\ \tau - \overline{\tau} \end{bmatrix} \tag{20}$$

where J_p is a penalty function with a positive weighting factor W_p bounding τ within $[\underline{\tau}\ \overline{\tau}]$. The soft constraints on τ are physically meaningful since, for the Puma 762, there is a thermal switch that disables the motors and applies the brakes when the amplifiers overload. With this hardware, the torques can exceed their limits for short time periods with no adverse consequences. Equations (17) and (18) are required to meet the initial and final conditions (6) and (7), respectively. Therefore, the actual variable parameters are p_w, p_t, and P, excluding the fixed p^0, p^1, p^{m-1}, and p^m; the total number of variable parameters are "$1 + 1 + n(m-3)$."

Note that one can incorporate obstacle avoidance with distance constraints into the formulation by adding another soft constraint into (13). This technique for handling obstacles has been used with good results in [12] and [16] and others.

C. Gradient of the Cost Function

In order to minimize J_{cp} in (13), which will produce an approximation to the extremal solution of the original cost functional (1), we use a sequential quadratic programming algorithm. Matlab's constrained optimization function, *constr*, was used for this. Even though B-splines have good numerical properties and are often used for function approximation, in [11], it was demonstrated that optimizations of the form (13) are not numerically well conditioned. Although most previous researchers use approximate finite difference gradients during the minimization process, this can lead to an unbounded condition number of the approximate Hessian of J_{cp}, and the algorithm will fail. For this reason, we derive formulas for the exact gradients of J_{cp}.

First, rewrite the cost function J_{cp} as

$$\begin{aligned} J_{cp} &= -wp_w + \frac{1}{2}\int_0^{p_t t_f} \tau^{\mathrm{T}} W_e \tau + \phi^{+^{\mathrm{T}}} W_p \phi^+ d\hat{t} \\ &= -wp_w + \frac{1}{2}\int_0^{p_t t_f} h(\tau) d\hat{t}. \end{aligned} \tag{21}$$

Then the gradient of the cost function J_{cp} is

$$\nabla J_{cp} = \begin{bmatrix} \dfrac{\partial J_{cp}}{\partial P} & \dfrac{\partial J_{cp}}{\partial p_t} & \dfrac{\partial J_{cp}}{\partial p_w} \end{bmatrix} \tag{22}$$

where

$$\frac{\partial J_{cp}}{\partial p} = \frac{1}{2}\int_0^{p_t t_f} \frac{\partial h}{\partial \tau}\frac{\partial \tau}{\partial p} d\hat{t} \quad (p \in P) \tag{23}$$

$$\frac{\partial J_{cp}}{\partial p_t} = \frac{1}{2}h(\tau(p_t t_f))t_f + \frac{1}{2}\int_0^{p_t t_f} \frac{\partial h}{\partial \tau}\frac{\partial \tau}{\partial p_t} d\hat{t} \tag{24}$$

$$\frac{\partial J_{cp}}{\partial p_w} = -w + \frac{1}{2}\int_0^{p_t t_f} \frac{\partial h}{\partial \tau}\frac{\partial \tau}{\partial p_w} d\hat{t}. \tag{25}$$

The most difficult terms to evaluate in the gradient of the integral are the derivatives of τ with respect to the joint parameters. For a parameterized joint trajectory of the form $q_s = q_s(\hat{t}, P)$, Appendix A gives the derivative of the inverse dynamics with respect to $p \in P$ [11]. In this algorithm, we ignore the friction terms since Coulomb friction is not continuously differentiable.

The last link of the robot holds the payload, which is the parameter p_w in (1). Therefore $J_n = J_n(p_w)$, and the derivative of the inverse dynamics with respect to p_w is described below as follows.

Algorithm II.2: To obtain $(\partial \tau_i / \partial p_w)$, replace J_i in Algorithm IV.1 by

$$J_i = \begin{cases} 0, & \text{if } i = 1 \sim n-1 \\ \dfrac{\partial J_n}{\partial p_w}, & \text{if } i = n. \end{cases}$$

For our application, the payload was modeled as a uniform disc of a fixed radius equal to that of a standard 20-kg (45 lb) plate. In this case, a given value for p_w actually determines the plate thickness, which was assumed to vary continuously. All the inertial parameters for the last link can then be computed easily as functions of p_w.

The other terms needed for the gradient of the cost function are relatively straightforward to compute. However, some care must be taken with the time-scale parameter p_t, since the effect of changing this parameter dramatically influences the cost function. The required derivatives are: $\dot{q}_s = (dq_s/d\hat{t}) = (\partial q/\partial t)(\partial t/\partial \hat{t}) = (1/p_t)\dot{q}$, $\ddot{q}_s = (d^2 q_s/d\hat{t}^2) = (1/p_t^2)\ddot{q}$, and for the gradient, $(\partial q_s(\hat{t})/\partial p_t) = (\partial q/\partial t)(\partial t/\partial p_t) = -(\hat{t}/p_t^2)\dot{q} = -(t/p_t)\dot{q}$, $(\partial \dot{q}_s(\hat{t})/\partial p_t) = -(1/p_t^2)\dot{q} - (t/p_t^2)\ddot{q}$, and $(\partial \ddot{q}_s(\hat{t})/\partial p_t) = -(2/p_t^3)\dot{q} - (t/p_t^3)\ddot{q}$.

III. Optimization Results

In this section, we apply the optimization approach to the Puma 762 robot. The optimization problem formulated in Section II-B was used to improve the Puma payload capability. The manufacturer's specifications state that a maximum load of 20 kg can be lifted at a distance of 25 cm from the center of its wrist. As mentioned previously, the payload was modeled as an uniform disc. The trajectories of the Puma's six joints were parameterized with uniform quintic B-splines.

The robot equations of motion require the mass, inertia, and friction properties. Because the Puma 762 has no documentation for these properties, a system identification technique similar to that employed by [1] was used to find them. The inverse dynamic equations were rewritten using a linear factorization in the unknown parameters of the form $H(q)\ddot{q} + h(q, \dot{q}) = \tau = Y(q, \dot{q}, \ddot{q})a$, where a contains the mass property information, and the matrix Y is computed from the knowledge of q, $\dot{q}$, and $\ddot{q}$. In order to collect the experimental data, the Puma was excited by different sinusoidal trajectories for each joint, and torque and position data were collected. The mass properties were then estimated by using least squares with a QR-decomposition. The detailed description of the system identification procedure can be found in [18]. The resulting solution is given in Table I.

We tested the algorithm for three different cases of 1 DOF, 3 DOFs, and 6 DOFs as discussed below. The goal in all cases was to move the payload from the downward posture, shown in Fig. 1, to the upward posture. However, different numbers of degrees of freedom were used in each case to execute the motion. The cases were as follows.

- *1 DOF:* the fourth joint is the only joint with freedom to move.
- *3 DOF:* the last three joints (wrist) are set free while the others are fixed.

TABLE I
Mass Properties of PUMA 762 Obtained by System Identification (in the SI System of Units)

	I_{xx}	I_{xy}	I_{xz}	I_{yy}	I_{yz}	I_{zz}
Link 1	0.00	0.00	0.00	4.63	0.00	0.00
Link 2	−4.42	2.40	9.39	0.00	−1.52	13.98
Link 3	0.00	2.19	−2.03	11.99	−3.31	0.00
Link 4	0.00	0.36	0.20	0.19	0.12	−0.78
Link 5	0.00	0.22	0.06	0.00	−0.44	0.04
Link 6	0.40	0.03	−0.19	0.08	0.08	0.23
	mr_x	mr_y	mr_z	m	f_c	f_v
Link 1	0.00	0.00	0.00	50.00	0.3000	0.0025
Link 2	−73.35	5.64	0.00	180.00	0.4800	0.0007
Link 3	−0.07	−6.63	0.00	90.00	0.3900	0.0011
Link 4	0.38	6.73	0.19	5.00	0.0800	0.0012
Link 5	−0.22	0.87	0.00	6.00	0.1100	0.0010
Link 6	−0.01	0.00	0.04	3.00	0.0800	0.0018

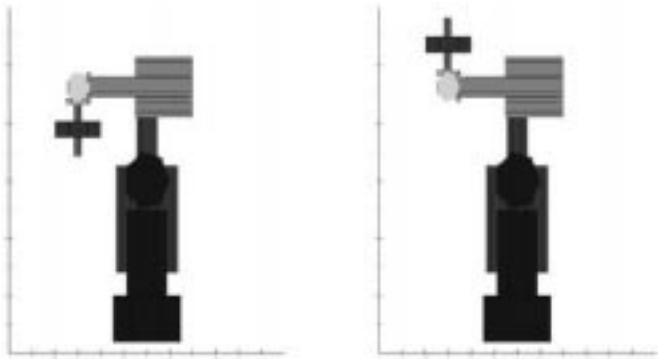

Fig. 1. The (a) initial and (b) final configurations of the Puma.

TABLE II
Optimal Results for Three Cases

Case	Initial J_c	Final J_c	Final Time	Payload
1 DOF	−24.2	−42.5	7.8 *sec*	29.4 *kg*
3 DOF	−4.15	−21.1	3.1 *sec*	46.0 *kg*
6 DOF	−181.0	−600.1	11.9 *sec*	63.2 *kg*

- *6 DOF:* all six joints are free to move.

In each case, the optimization procedure was initialized with $p_w = 20$ kg, $p_t = 1$, $t_f = 20$ s, and a prescribed path which moves the fourth joint from the initial configuration to the final one, shown in Fig. 1, on a smooth trajectory without any oscillation. The other joints were initially parameterized to be constant functions of time during the motion—they did not move initially. The solutions to the optimization problem for the three cases are shown in Table II and Figs. 2–4. Table II shows the initial and final values of the cost function, along with the path traversal time and the payload for all three cases. Note for the 1-DOF and the 3-DOF cases, the motors that were held fixed needed to exert torque on the robot during the motion, but this cost was not included in the evaluation of J_{cp}.

For the 1-DOF case, Fig. 2 shows that energy is pumped into joint 4 through a swinging motion. The amplitude of the swing increases until the velocity of joint 4 reaches its limit. At this point, the arm continues to move until the final upward posture is reached. A comparison of the joint velocity and applied torque shows that both have the same sign throughout most of the time interval. This is expected since energy must be added to the system in order to swing up the weight. One could conjecture that minimal effort motions could thus be constructed by using the natural oscillations of the system and applying a

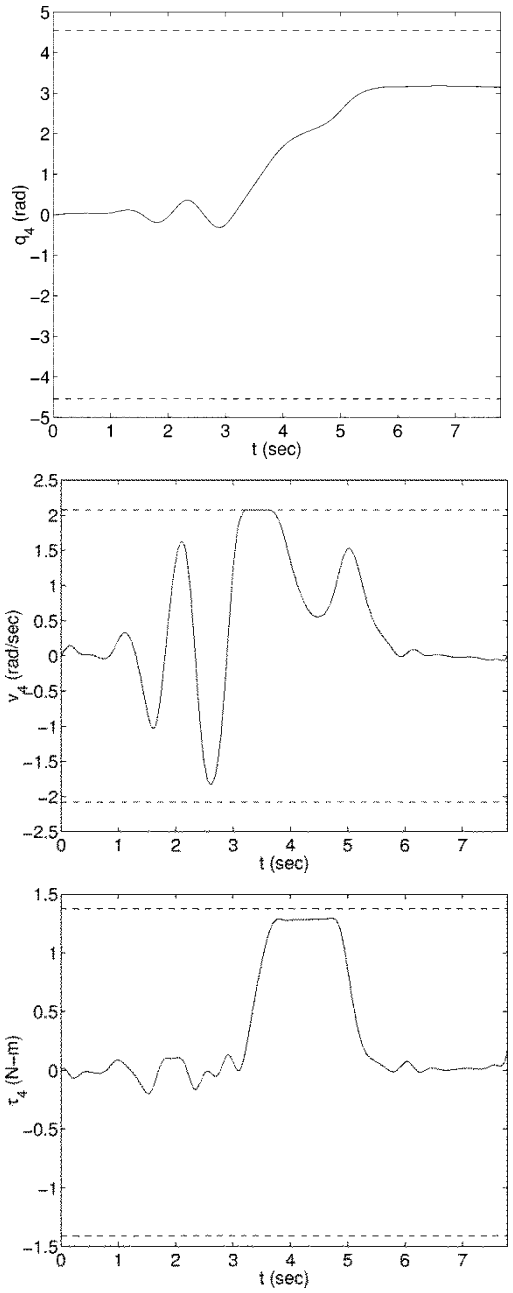

Fig. 2. The motion, speed, and torque of the fourth joint in the 1-DOF case (limits: dotted lines).

small torque in phase with the velocity (i.e., switching the torque when the velocity changes sign). However the real physical constraints, such as the velocity bounds, would make it difficult to plan motions in this manner that satisfy the boundary conditions on the motion. Note that semi-infinite velocity bound is an active constraint for almost an entire second during the motion.

When this trajectory was experimentally tested on the Puma, we first attempted to move the arm along the initial trajectory with no swings, but with the larger payload. This indeed caused the amplifier for joint 4 to overload and automatically shut down, with the brakes applied. We then tested the optimal swinging trajectory, and found that only about 25 kg, rather than the predicted 29.4 kg, could actually be lifted without causing the amplifier to overload. We believe that the difference is due to imprecise knowledge of the actual bounds for

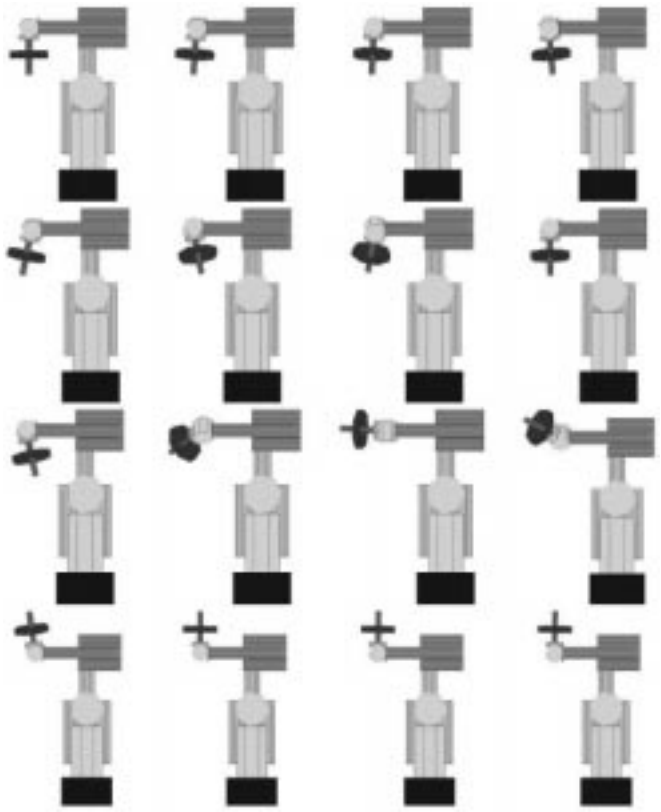

Fig. 3. The motion of the wrist in the 3-DOF case.

torque and velocity. The torque limits $\tau_{\max}$ were found experimentally from static measurements of the stall torque for each motor, when $\dot{q} = 0$. In order to account for nonzero $\dot{q}$, we made $\overline{\tau} = .75\,\tau_{\max}$. In addition, we also made $\overline{\dot{q}} = .8\dot{q}_{\max}$, where $\dot{q}_{\max}$ is the velocity limits specified in the Puma manuals. Other, more complicated bounds could easily have been used in our algorithm if they were known.

The motion shown in Fig. 3 is the 3-DOF case where the entire wrist is allowed to move. Although it is not obvious from the figure (the actual motions can be seen in the video proceedings of the 1999 ICRA or at the website listed in the abstract), the wrist moves in a spherical oscillation using all three joints until the payload axis has lifted 90° from the starting configuration. At this point, the wrist is in a singular configuration that requires no torque from the motors of the wrist to keep the weight suspended. Then the payload continues moving upwards until it reaches the desired final configuration. Using all three joints in this manner, the payload was increased to 46 kg.

The final motion, shown in Fig. 4, is the 6-DOF case where all the joints were allowed to move. In this case, the actual motion (rather than the time-sequence) is very dramatic since the potential energy of the payload is first transformed into internal kinetic energy of the robot and then, through some interesting maneuvering, it arrives at its desired final configuration. The payload actually spins significantly about its axis during the motion, using gyroscopic effects to assist in the weightlifting. For this case, the payload was increased over the previous cases to 63 kg, or more than triple the manufacturer's specifications.

A. Discussion

In order to reduce the motor torques, the gear ratios and the gear interaction between the motors and the joints must be taken into account

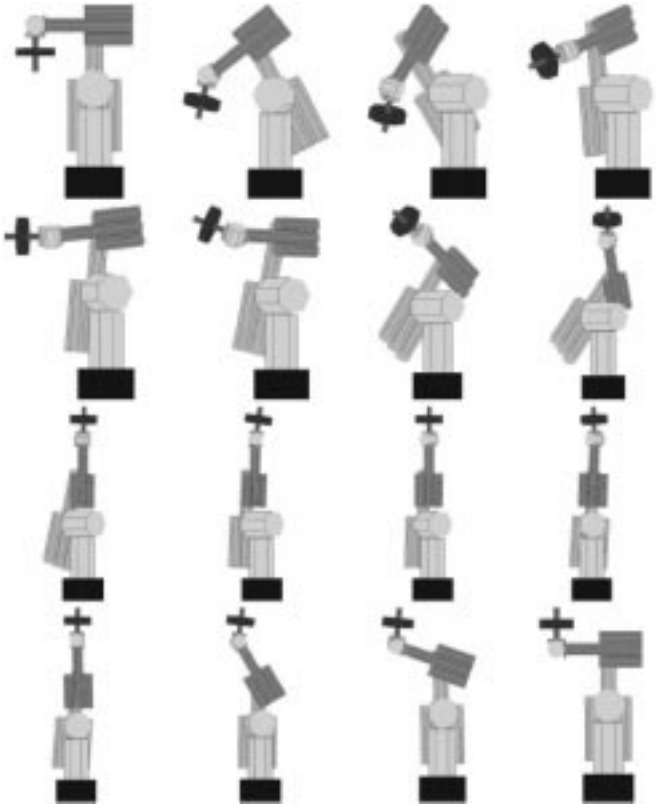

Fig. 4. The motion for the 6-DOF case.

in the problem formulation. The linear relation between the motor displacements and joint displacements are represented as a 6×6 matrix B such that $\dot{q} = B\dot{q}_m$, where $\dot{q}_m$ is the motor shaft velocity. Applying virtual work, we can describe the motor torque τ_m in terms of the joint torque τ as follows:

$$\tau^{\mathrm{T}}\dot{q} = \tau_m^{\mathrm{T}}\dot{q}_m \Rightarrow \tau_m = B^{\mathrm{T}}\tau. \tag{26}$$

In addition, in the Puma, the maximum motor torques for the first three joints is about five times those for the wrist joints. Therefore, the weighting factors W_e and W_p are set as

$$\begin{aligned} W_e &= c_e W_b \quad \text{and} \quad W_p = c_p \begin{bmatrix} W_b & 0 \\ 0 & W_b \end{bmatrix} \text{ with} \\ W_b &= B \begin{bmatrix} \frac{1}{5} \cdot 1_{3\times 3} & 0 \\ 0 & 1_{3\times 3} \end{bmatrix} B^{\mathrm{T}} \end{aligned} \tag{27}$$

where c_e and c_p are scalar weighting factors, 1 is an identity matrix, and (1/5) is replaced by 0 in the 1-DOF and 3-DOF cases.

In the optimization problem, it is necessary to choose appropriate weighting factors w, c_e, and c_p. Ideally one would make the joint torque penalty term c_p as large as possible in order to bound the joint torques within their limits. However, as is well known with penalty function methods, the optimization problem becomes ill-conditioned and will not converge if c_p becomes too large. In addition, we expect that a small value for w and a large value for c_e will yield a solution with a small payload, while a large w and small c_e will yield a large payload. This was consistent with our optimization results. Typical values used for our results were $c_e = 1$ and $c_p = 10000$. The value used for w was 30, 10, and 200 for the 1-DOF, 3-DOF, and 6-DOF cases, respectively.

All of the results obtained are *local* minimizers of the nonlinear cost function. It is desirable to obtain *globally* minimizing trajectories. We started the algorithm with different initial paths in a search for other minimizing trajectories. We found several other interesting local minimizers for the 6-DOF case, which can be seen in the video proceedings of the 1999 ICRA. The highest payload was achieved with the motion shown in Fig. 4. In all the cases, we first solved the optimization with a low number of spline parameters $m = 10$ in (8). We then increased m to 15, and used the solution for $m = 10$ as the starting point for a new optimization. In most cases, the trajectory did not change significantly. If it did, we further increased m. The final value for m for the 1-DOF case was $m = 25$, and for the 6-DOF solution shown, the final value was $m = 15$. Note that for the 6-DOF case, $m = 15$ means that there were a total of $6(m-3)+2 = 74$ variable parameters used in the optimization. This includes the time scale parameter and the payload parameter. The computation time for the optimization ranged from several minutes to about an hour on a PII-300 PC. The time depended on the number of parameters used, and on the termination tolerance of the optimization.

As mentioned previously, there is no exact mathematical model that gives the peak torque versus velocity relation for the motor amplifiers of the Puma. Although we restricted the velocity limits to 80% of those specified in the Puma documents, and the torque bounds to 75% of their static bounds, we found several motions that caused the amplifiers to overload when tested on the real system. It was also interesting that higher payloads could be achieved when the Puma was cold than could be obtained after it had been in operation for some time-probably due to the thermal overload switches in the amplifiers.

Finally, it is interesting to note that the resulting motions from our algorithm routinely swing through singular configurations. In these configurations, the actuator effort is reduced since loads are supported by the robot structure. Note that humans also move to singular configurations to support large loads. However, it is not known what, if any, the cost function is that they are minimizing. There is some clinical evidence that humans may minimize jerk or $\dot{\tau}$ [8].

IV. Conclusions

We developed a weightlifting motion planner for robots which combines maximization of the payload with minimization of the effort. Our solutions demonstrated that optimal motions routinely swing through singular configurations. As is often done by human weightlifters, our solutions also often used a pumping motion to assist the lift. The approach can be applied to any manipulation task where payloads need to exceed the manufacturers' limits—the main requirement is that the robot can support the weight statically in the initial and final configurations.

In our approach, the optimal control problem is reformulated as a parameter optimization with variable control vertices for the B-splines, a time-scale factor, and an unknown payload. The differentiability of the B-splines together with the limited support property provide the ability to calculate derivatives with respect to the control vertices and the time-scale factor. In addition, based on the convex hull property of the B-splines, the semi-infinite constraints on the joint position and velocity can be transformed to linear inequalities in the parameter space.

Appendix A
Recursive Inverse Dynamics

The inverse dynamics algorithm in this paper is defined using notation from differential geometry (see [14] for more detail). The *special Euclidean group*, or $SE(3)$, is a Lie group corresponding to the homogeneous transformations on $\Re^3$. Given $\Theta \in SO(3)$ and $b \in \Re^3$, for $G = (\Theta, b) \in SE(3)$, we write it as

$$G = \begin{bmatrix} \Theta & b \\ 0 & 1 \end{bmatrix}. \tag{28}$$

The Lie algebra associated with $SE(3)$ is referred as $se(3)$, where $g \in se(3)$ can be written as a 6×1 vector

$$g = \begin{bmatrix} \omega_g \\ v_g \end{bmatrix} \tag{29}$$

where ω_g and v_g are the rotational and translational components of the motion. The adjoint mapping on $SE(3)$ is written as

$$Ad_G h = \begin{bmatrix} \Theta & 0 \\ [b]\Theta & \Theta \end{bmatrix} \begin{bmatrix} \omega_h \\ v_h \end{bmatrix} \tag{30}$$

where the matrix $[b]$ (where $b \in \Re^3$) is taken as the skew-symmetric matrix formed from the elements of b, and $h = \left[\begin{smallmatrix} \omega_h \\ v_h \end{smallmatrix}\right] \in se(3)$. The adjoint mapping on $se(3)$, or the Lie bracket, is written as

$$ad_g h = \begin{bmatrix} [\omega_g] & 0 \\ [v_g] & [\omega_g] \end{bmatrix} \begin{bmatrix} \omega_h \\ v_h \end{bmatrix}. \tag{31}$$

The corresponding dual adjoint mappings are

$$Ad_G^* h^* = \begin{bmatrix} \Theta^{\mathrm{T}} & \Theta^{\mathrm{T}}[b]^{\mathrm{T}} \\ 0 & \Theta^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} m_h \\ f_h \end{bmatrix} \tag{32}$$

and

$$ad_g^* h^* = \begin{bmatrix} [\omega_g]^{\mathrm{T}} & [v_g]^{\tau} \\ 0 & [\omega_g]^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} m_h \\ f_h \end{bmatrix} \tag{33}$$

where $h^* = \left[\begin{smallmatrix} m_h \\ f_h \end{smallmatrix}\right]$.

Algorithm IV.1: (Park *et al.* [14]) The recursive formulation of the inverse dynamics is as follows.

- Initialization

$$V_0, \dot{V}_0, F_{n+1}.$$

- Forward recursion: $i = 1$ to n

$$V_i = Ad_{T_{i-1,i}^{-1}} V_{i-1} + S_i \dot{q}_i$$
$$\dot{V}_i = Ad_{T_{i-1,i}^{-1}} \dot{V}_{i-1} + S_i \ddot{q}_i + ad_{V_i} S_i \dot{q}_i.$$

- Backward recursion: $i = n$ to 1

$$F_i = Ad^*_{T_{i,i+1}^{-1}} F_{i+1} + J_i \dot{V}_i - ad^*_{V_i} J_i V_i$$
$$\tau_i = S_i^{\mathrm{T}} F_i + f_{ci} \operatorname{sgn}(\dot{q}_i) + f_{vi} \dot{q}_i$$

where the index i represents the ith link frame counted out from the base frame ($i = 0$). The joint screw, spatial velocity, spatial acceleration, and spatial force are written as S_i, V_i, $\dot{V}_i$, and $F_i \in se(3)$, respectively. V_0 and $\dot{V}_0$ represent the spatial velocity and acceleration of the base, while F_{n+1} represents the external spatial force. $T_{i-1,i} \in SE(3)$ denotes the transformation from the link frame $i-1$ and link frame i. The joint velocity, acceleration, force, Coulomb friction, and viscous friction are written as $\dot{q}_i$, $\ddot{q}_i$, τ_i, f_{ci}, and $f_{vi} \in \Re^n$, respectively. Matrix J is the 6×6 spatial inertia matrix

$$J_i = \begin{bmatrix} I_i - m_i[r_i]^2 & m_i[r_i] \\ -m_i[r_i] & m_i \cdot 1 \end{bmatrix} \tag{34}$$

where m_i and I_i are the mass and inertia of the ith link, respectively, r_i is a vector from the origin of the ith link frame to its center of mass, and 1 is an identity matrix.

Algorithm IV.2: (Martin and Bobrow [11]) The recursive formulation of the derivative of the inverse dynamics is as follows.

- Initialization

$$\frac{\partial V_0}{\partial p}, \frac{\partial \dot{V}_0}{\partial p}, \frac{\partial F_{n+1}}{\partial p}.$$

- Forward recursion: $i = 1$ to n

$$\frac{\partial V_i}{\partial p} = Ad_{T_{i-1,i}^{-1}} \frac{\partial V_{i-1}}{\partial p} + S_i \frac{\partial \dot{q}_{si}}{\partial p} + ad_{V_i} S_i \frac{\partial q_{si}}{\partial p}$$
$$\frac{\partial \dot{V}_i}{\partial p} = ad_{Ad_{T_{i-1,i}^{-1}} \dot{V}_{i-1}} S_i \frac{\partial q_{si}}{\partial p} + Ad_{T_{i-1,i}^{-1}} \frac{\partial \dot{V}_{i-1}}{\partial p} + S_i \frac{\partial \ddot{q}_{si}}{\partial p} + ad_{(\partial V_i/\partial p)} S_i \dot{q}_{si} + ad_{V_i} S_i \frac{\partial \dot{q}_{si}}{\partial p}.$$

- Backward recursion: $i = n$ to 1

$$\frac{\partial F_i}{\partial p} = Ad^*_{T_{i,i+1}^{-1}} \frac{\partial F_{i+1}}{\partial p} + J_i \frac{\partial \dot{V}_i}{\partial p} - ad^*_{(\partial V_i/\partial p)} J_i V_i - ad^*_{V_i} J_i \frac{\partial V_i}{\partial p} - ad^*_{Ad_{T_{i,i+1}} S_{i+1} (\partial q_{si+1}/\partial p)} Ad^*_{T_{i,i+1}^{-1}} F_{i+1}$$
$$\frac{\partial \tau_i}{\partial p} = S_i^{\mathrm{T}} \frac{\partial F_i}{\partial p}.$$

ACKNOWLEDGMENT

The algorithms developed in this work stemmed from collaborations with Prof. F. C. Park at Seoul National University, Dr. S. Ploen, and Dr. B. Martin at the Jet Propulsion Laboratory. The authors also thank the anonymous reviews for their helpful comments.

REFERENCES

[1] C. H. An, C. G. Atkeson, and J. M. Hollerbach, "Estimation of inertial parameters of rigid body links of manipulators," in *Proc. IEEE Conf. Decision and Control*, vol. 2, 1985, pp. 990–995.

[2] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* San Mateo, CA: Morgan Kaufmann, 1987.

[3] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *J. Guidance, Contr., Dynam.*, vol. 21, no. 2, pp. 193–207, 1999.

[4] A. E. Bryson and Y. C. Ho, *Applied Optimal Control.* New York: Wiley, 1995.

[5] B. Cao, G. I. Dodds, and G. W. Irwin, "Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots," *IEE Proc.-Contr. Theory Applicat.*, vol. 144, no. 5, pp. 467–475, 1997.

[6] Y.-C. Chen, "Solving robot trajectory Planning problems with uniform cubic B-splines," *Optimal Contr. Applicat. Methods*, vol. 12, no. 4, pp. 247–262, 1991.

[7] A. De Luca, L. Lanari, and G. Oriolo, "A Sensitivity Approach to Optimal Spline Robot Trajectories," *Automatica*, vol. 27, no. 3, pp. 535–539, 1991.

[8] J. P. Desai, M. Zefran, and V. Kumar, "Two-arm manipulation tasks with friction-assisted grasping," *Adv. Robot.*, vol. 12, no. 5, pp. 485–507, 1999.

[9] G. Field and Y. Stepanenko, "Iterative dynamic programming: an approach to minimum energy trajectory planning for robotic manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1996, pp. 2755–2760.

[10] J.-C. Latombe, *Robot Motion Planning.* Boston, MA: Kluwer, 1991.

[11] B. J. Martin and J. E. Bobrow, "Minimum effort motions for open chain manipulators with task-dependent end-effector constraints," *Int. J. Robot. Res.*, vol. 18, no. 2, pp. 213–224, 1999.

[12] C. J. Ong and E. G. Gilbert, "Robot path planning with penetration growth distance," *J. Robot. Syst.*, vol. 15, no. 2, pp. 57–74, Feb 1998.

[13] H. Ozaki and C.-J. Lin, "Optimal B-spline joint trajectory generation for collision-free movements of a manipulator under dynamic constraints," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1996, pp. 3592–3597.

[14] F. C. Park, J. E. Bobrow, and S. R. Ploen, "A Lie group formulation of robot dynamics," *Int. J. Robot. Res.*, vol. 14, no. 6, pp. 609–618, 1995.

[15] J.-K. Park and J. E. Bobrow, "Minimum-time trajectory planning for a robot manipulator amid obstacles," in *Proc. Fourth Japan–France Congress and Second Asia–Europe Congress on Mechatronics*, vol. 1, 1999, pp. 369–374.

[16] Z. Shiller and S. Dubowsky, "On computing the global time-optimal motions of robotic manipulators in the presence of obstacles," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 785–797, Dec. 1991.

[17] S. E. Thompson and R. V. Patel, "Formulation of joint trajectories for industrial robots using B-splines," *IEEE Trans. Ind. Electron.*, vol. 34, no. 2, pp. 192–199, 1987.

[18] C.-Y. E. Wang, "Optimal Path Generation for Robots," Ph.D. dissertation, Univ. California, Irvine, 2001.

[19] C.-Y. E. Wang, W. K. Timoszyk, and J. E. Bobrow, "Weightlifting motion planning for a Puma 762 robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, 1999, pp. 480–485.