Bryan J. Martin

Southwest Aerospace
2672 Dow Avenue
Tustin, California 92780, USA
jkmartin@ix.netcom.com

James E. Bobrow

Department of Mechanical and Aerospace Engineering
University of California, Irvine
Irvine, California 92697, USA

Minimum-Effort Motions for Open-Chain Manipulators with Task-Dependent End-Effector Constraints

Abstract

In this article, we examine the solution of minimum-effort optimal control problems for open-chain manipulators. An approximate solution to the optimal control problem is determined by a constrained parameter optimization over a set of B-spline basis functions. We demonstrate that the parameter-optimization formulation of the problem is numerically ill-conditioned, and that it is therefore essential to include analytic, or exact, gradients of the objective function and the constraints in order to guarantee a solution. A recursive expression for these gradients is developed for general serial chains. Constraints on end-effector motions are taken into account using the logarithm of the spatial displacement. Our formulation relies on the use of matrix exponentials for the manipulator kinematics, dynamics, and task constraints. Several examples are presented that demonstrate the power and flexibility of our approach.

1. Introduction

Previous research on the optimal control of robotic systems has been hampered by the complexity of the nonlinear equations of motion, and the difficulty of satisfying the necessary conditions for optimality. In this paper we present an approach that we have successfully applied to a number of challenging problems in the optimal control of open chains. For our approach, like some others, we parameterize the joint trajectories to transform the optimal control problem into a discrete parameter optimization. Unlike other approaches, we determine the analytic gradient of the objective function and the constraints in a manner that can be applied to general open chains. Also unlike others, we show that task-based end-effector constraints can be handled naturally with exponential coordinates (Park and Brocket 1994). Although SE(3) is not globally homeomorphic to $\Re^6$, an advantage of exponential coordinates is that the Jacobian of the map from $\Re^6$ to SE(3) *is* globally nonsingular.

Much early research on optimal control of robot motions used the minimum-time criterion (Kahn and Roth 1971; Bobrow, Dubowsky, and Gibson 1985; Shin and McKay 1985). Several researchers (Bobrow 1988; Dubowsky, Norris, and Shiller 1986) parameterized the geometric path with splines, and used a discrete parameter-optimization approach to find minimum time paths that avoid obstacles. Further work in this direction added energy to the cost function (Shiller 1994) and also actuator dynamics (Pledel and Bestaoui 1995). Geering and colleagues (1986), and later, Bryson and Meier (1990), solved the general time-optimal control problem for a two-link arm following an unspecified path. Both researchers noted the difficulty of applying numerical shooting methods, and resorted to a parameter-optimization approach.

Other performance measures such as effort, energy, and the time derivative of the control torques have been considered in the literature (Chen, Cheng, and Sun 1994; Malladi et al. 1992; Suh and Hollerbach 1987; Hu, Teo, and Lee 1994). These performance measures result in smooth, natural-looking motions that reduce wear and tear on the robot when compared to time-optimal motions (Kenjo and Nagamori 1986). Žefran, Kumar, and Yun (1994) considered a two-arm robot system holding an object, and directly solved the two-point boundary-value problem that arose from the maximum principle (Pontryagin et al. 1962). They also noted the difficulty in obtaining the gradient of the dynamics. In a later work (Žefran and Kumar 1995), they extended the problem to handle unilateral constraints, and used a variational approach to develop a finite-difference solution.

Many researchers have recognized the utility of using a polynomial parameterization of the robot-joint trajectories to

The International Journal of Robotics Research
Vol. 18, No. 2, February 1999, pp. 213-224,

convert the optimal control problem into a discrete, nonlinear parameter optimization. With this approach, one guesses an initial motion that satisfies the constraints, and adjusts the spline parameters in search of an optimum. The constraints are usually obstacles in the workspace, joint limits, and constraints on end-effector motions. One of the main advantages is that even if a nonoptimal path that satisfies the constraints has been found, it is still a valid, or feasible, trajectory. Gilbert and Johnson (1985) used uniform B-splines to parameterize the motion of a body moving in a plane in the presence of obstacles. They demonstrated that minimum energy motions can be found reliably with a large number of path parameters. A subsequent paper (Gilbert and Ong 1994) extended the basic approach to solve path-planning problems. In the work of Lu (1992), the difficulty of obtaining gradients for Gilbert's approach (applied to flight-vehicle trajectory optimization) was discussed, and an approximate scheme was developed. Other researchers (Tabarah, Benhabib, and Fenton 1994; Field and Stepanenko 1996; Ozaki and Lin 1996; Park 1997) have used splines with varying degrees of success. All of the above-mentioned papers discuss the need for exact gradients of the objective function.

An important topic related to our work is kinematic redundancy resolution and path planning for specified end-effector tasks. Several efficient global approaches have been developed for kinematic redundancy resolution (Wang and Chen 1991; Kazerounian and Wang 1988; Kim, Park, and Lee 1994; Martin, Baillieul, and Hollerbach 1989) using a variational approach for their solution. Dynamics was added to the problem for a 3-R planar robot by Ma (1996). Seereeram and Wen (1993, 1995) added obstacle avoidance to the redundancy-resolution problem, where joint rates were considered to be the control variable. All of these approaches require analytic gradients of the cost functional. An interesting aspect of these papers is that each seems to use a different form for the end-effector orientation constraint (Angeles, Rojas, and Lopez-Cajun 1988).

The foundation of our approach is the use of product-of-matrix exponentials (POE) to represent the manipulator kinematics. Although other formulations of the kinematics and dynamics could have been used, the POE-based approach makes it easier to express the problem for general open chains, since joint motions are represented in terms of screw motions, which include either rotational or prismatic joints, with the form of the equations identical in either case (Park, Bobrow, and Ploen 1995). The manipulator dynamics are computed from a modified version of Newton-Euler dynamics, expressed in terms of products of matrix exponentials.

We determine an approximate solution to the optimal control problem by a constrained parameter optimization on a set of B-spline basis functions. We then demonstrate that the standard parameter-optimization formulation of the problem is numerically ill-conditioned. While it is possible to optimize many objective functions by estimating the gradients (Lu 1992), poorly conditioned problems require analytic gradients. These exact gradients allow the optimization algorithm to quickly converge to a local minima that might otherwise not be found. Computation of the gradient of the cost function requires derivatives, or sensitivities, of the equations of motion with respect to the joint variables. These sensitivities have also been developed by other researchers (Balafoutis, Misra, and Patel 1986; Murray and Neuman 1986; Balafoutis and Patel 1991), and could have been used for the present work. However, our formulation provides additional insight into the problem, since we are able to represent end-effector task constraints in exponential coordinates, and obtain analytic expressions for the gradients of these constraints. These constraint sensitivities have not been presented previously in the literature.

2. Minimum-Effort Optimal Control

Consider an n-degree of freedom, possibly redundant, open-chain manipulator. The minimum-effort optimal control problem for this system is

$$\underset{\tau(\cdot)}{\text{minimize}} \qquad J(\tau) = \frac{1}{2}\int_0^{t_f} ||\tau||^2 dt, \tag{1}$$

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

$$\underline{q} \le q(t) \le \overline{q}, \tag{3}$$

$$q(0) = q_0,\ \dot{q}(0) = 0, \tag{4}$$

$$q(t_f) = q_f,\ \dot{q}(t_f) = 0, \tag{5}$$

where eq. (2) is the equation of motion for the open chain with the joint coordinates $q \in \Re^n$ and the joint forces or torques $\tau \in \Re^n$; $M(q)$ is the $n \times n$ inertia matrix; and $h(q, \dot{q})$ is the vector of Coriolis, gravity, and friction terms. Constraints on joint displacements are represented by eq. (3), where $\underline{q}, \overline{q} \in \Re^n$ are assumed to be given for a particular manipulator. While the problem is formulated with the final time t_f fixed, we give an example that shows how our approach will find the optimal final time if it is less than the specified t_f. In addition to the constraints of eqs. (2)–(5), we also consider task-dependent constraints on the end effector. For these constraints, the end effector is required to trace a given path in space, with or without a prescribed orientation. This constraint is represented as k equality constraints of the form

$$\tilde{g}_j(q(t), c(t)) = 0, \quad j = 1 \ldots k, \tag{6}$$

where $c(\cdot)$ is a curve in space that includes both position and orientation information about the desired end-effector position, and we assume that $\tilde{g}_j(q(t), c(t))$ is continuously differentiable. These constraints are specified directly in exponential coordinates, and are discussed in more detail in Section

4. (See Park's [1991]) work for a discussion of exponential coordinates.)

For our approximation to the solution of the optimal control problem, we assume that the joint coordinates are parameterized by cubic B-splines (De Boor 1978) in all of the examples in this paper. The B-spline curve depends on the blending, or basis, functions $B_i(t)$, and the control points $P = \{p_1, ..., p_m\}$, with $p_i \in \Re^n$. The joint trajectories then have the form $q = q(t, P)$ with

$$q(t, P) = \sum_{i=1}^{m} B_i(t) p_i. \tag{7}$$

The control points p_i of the spline only have a local effect on the curve geometry (Faux and Pratt 1979, ch. 6), so given any t, there will be a maximum of four nonzero $B_i(t)$ in eq. (7) for a cubic spline. In addition, the convex-hull property of B-splines makes them useful for smoothing or approximating data. The fact that $\sum_{i=1}^{m} B_i(t) = 1$ also gives the desirable property that *limits on joint displacements translate directly to limits on the spline parameters* p_i. That is, if there is a joint limit of the form $q \leq \overline{q}$, and one constrains $p_i \leq \overline{q}$, then

$$q = \sum_{i=1}^{m} B_i(t) p_i \leq \sum_{i=1}^{m} B_i(t) \overline{q} = \overline{q}.$$

The parameter optimization equivalent of the original minimum-effort problem is

$$\underset{P}{\text{minimize}} \qquad J(P) = \frac{1}{2} \int_0^{t_f} ||\tau(t, P)||^2 dt, \tag{8}$$

$$\text{subject to} \qquad \underline{q} \leq p_i \leq \overline{q}, \quad i = 1 \ldots m, \tag{9}$$

$$g_j(q(t, P), c(t)) = 0, \quad j = 1 \ldots k. \tag{10}$$

With this approach, $\tau = \tau(t, P)$, because q, $\dot{q}$, and $\ddot{q}$ all are given functions of t and P from eq. (7) and its time derivatives, so τ is an explicit function of the spline parameters through eq. (2). By the proper choice of the spline basis functions at both ends of the joint trajectory, the path-end conditions, eqs. (4) and (5), can be satisfied by keeping the outer two pairs of spline-control points constant.

If one does not consider the end-effector curve-tracing constraint (eq. (10)), we have converted the original problem into a parameter-optimization problem with no nonlinear constraints, and efficient quasi-Newton algorithms can then be used to solve the problem. However, for assured convergence of these algorithms, two conditions must be met: the second derivatives of $J(P)$ must be bounded, and every approximate Hessian (found, for example, from a BFGS update [Luenberger 1989]) used in the quasi-Newton algorithm must remain positive definite with bounded condition number (Gill, Murray, and Wright 1981).

Unfortunately, for problems that are ill-conditioned to begin with, approximate finite-difference gradients can lead to an unbounded condition number of the approximate Hession of $J(P)$, and the algorithm will fail. We experienced this problem firsthand at the beginning of this research: our optimizations would terminate prematurely when we used finite-difference gradients. A more-complete discussion of the problems associated with finite-difference gradients is given by Gill and colleagues (Gill, Murray, and Wright 1981; Gill et al. 1986) and their references. Due to the complexity of the dynamic equations of motion, most nontrivial solutions to optimal control problems for robotic systems presented in the research literature use finite-difference gradient approximations. In the following sections, we give explicit formulas for the gradients of J and g, which can be applied to general open chains.

3. Optimization of Unconstrained Motions

The first problem we consider is a minimum-effort motion with no constraint on the motion of the end effector, so that $k = 0$ in eq. (10). For any given parameter vector P, the joint torques can be computed efficiently with a modified recursive Newton-Euler dynamics algorithm based on the product-of-exponentials (POE) kinematic representation, shown in Figure 1 (Park and Bobrow 1994). The notation used for the algorithm is defined in Appendix A. Aside from its coordinate-free geometric interpretation, an advantage of this POE-based algorithm is that the joint coordinates q_i appear explicitly in the equations in the *same form* for revolute or prismatic joints. Because matrix exponentials are used to represent the joint-to- joint transformations, it is trivial to differentiate them with respect to joint displacements. One can then compute joint torques and their derivatives with the same computer source code for any open-chain robot.

3.1. The Recursive Gradient Formulation

To compute the gradient of the cost functional, we note that

$$\nabla_P J = \int_0^{t_f} \tau^T \cdot (\nabla_P \tau) \, dt. \tag{11}$$

The most significant step for this gradient is computing the derivatives of the joint torques with respect to the path parameters, P. We compute these derivatives analytically by differentiating the recursive dynamics shown in Figure 1. The details of this derivation are provided in Appendix A. The resulting recursive algorithm for the gradient is shown in Figure 2. Alternative approaches have been developed for the *linearization* of the dynamics equations that result in a similar algorithm. Balafoutis and Patel (1991) provide a complexity analysis of a recursive algorithm to compute a linearized model of a manipulator with revolute joints.

- **Initialization**

$$\mathcal{V}_0 = \dot{\mathcal{V}}_0 = \mathcal{W}_{n+1} = 0$$

- **Forward recursion: for $i = 1$ to n do**

$$\begin{aligned} T_{i-1,i} &= M_i e^{S_i q_i} \\ \mathcal{V}_i &= \mathrm{Ad}_{T^{-1}_{i-1,i}}(\mathcal{V}_{i-1}) + S_i \dot{q}_i \\ \dot{\mathcal{V}}_i &= S_i \ddot{q}_i + \mathrm{Ad}_{T^{-1}_{i-1,i}}(\dot{\mathcal{V}}_{i-1}) + \\ &\quad \left[\mathrm{Ad}_{T^{-1}_{i-1,i}}(\mathcal{V}_{i-1}), S_i \dot{q}_i\right] \end{aligned}$$

- **Backward recursion: for $i = n$ to 1 do**

$$\begin{aligned} \mathcal{W}_i &= \mathrm{Ad}^*_{T^{-1}_{i,i+1}}(\mathcal{W}_{i+1}) + J_i \dot{\mathcal{V}}_i - \mathrm{ad}^*_{\mathcal{V}_i}(J_i \mathcal{V}_i) \\ \tau_i &= S_i^T \mathcal{W}_i \end{aligned}$$

Fig. 1. The POE-recursive Newton-Euler inverse-dynamics algorithm.

- **Initialization**

$$\frac{d\mathcal{V}_0}{dp_i} = \frac{d\dot{\mathcal{V}}_0}{dp_i} = \frac{d\mathcal{W}_{n+1}}{dp_i} = 0, \ \forall\, p_i \in P$$

- **Forward recursion: for $i = 1$ to n do**

$$\begin{aligned} \frac{d\mathcal{V}_i}{dp_i} &= \frac{dq_i}{dp_i} \mathrm{ad}_{\mathrm{Ad}_{T^{-1}_{i-1,i}}(\mathcal{V}_{i-1})} S_i + \mathrm{Ad}_{T^{-1}_{i-1,i}} \frac{d\mathcal{V}_{i-1}}{dp_i} + S_i \frac{d\dot{q}}{dp_i} \\ \frac{d\dot{\mathcal{V}}_i}{dp_i} &= \frac{dq_i}{dp_i} \mathrm{ad}_{\mathrm{Ad}_{T^{-1}_{i-1,i}}(\dot{\mathcal{V}}_{i-1})} S_i + \mathrm{Ad}_{T^{-1}_{i-1,i}} \frac{d\dot{\mathcal{V}}_{i-1}}{dp_i} + \\ &\quad \mathrm{ad}_{\frac{d\mathcal{V}_i}{dp_i}} S_i \dot{q}_i + \mathrm{ad}_{\mathcal{V}_i} S_i \frac{d\dot{q}_i}{dp_i} + S_i \frac{d\ddot{q}}{dp_i} \end{aligned}$$

- **Backward recursion: for $i = n$ to 1 do**

$$\begin{aligned} \frac{d\mathcal{W}_i}{dp_i} &= J_i \frac{d\dot{\mathcal{V}}_i}{dp_i} + \mathrm{Ad}^*_{T^{-1}_{i,i+1}} \frac{d\mathcal{W}_{i+1}}{dp_i} \\ &\quad - \mathrm{ad}^*_{\frac{d\mathcal{V}_i}{dp_i}} J_i \mathcal{V}_i - \mathrm{ad}^*_{\mathcal{V}_i} J_i \frac{d\mathcal{V}_i}{dp_i} \\ &\quad - \mathrm{ad}^*_{\mathrm{Ad}_{M_{i+1}} S_{i+1} \frac{dq_{i+1}}{dp_i}} \mathrm{Ad}^*_{T^{-1}_{i,i+1}} \mathcal{W}_{i+1} \\ \frac{d\tau_i}{dp_i} &= S_i^T \frac{d\mathcal{W}_i}{dp_i} \end{aligned}$$

Fig. 2. The POE-recursive derivative Newton-Euler inverse dynamics algorithm.

The objective function in eq. (8) and its gradient were integrated numerically, using the trapezoidal rule with a constant number of integration steps. Although we *approximated* the integral in $J(P)$ with a fixed-step-size integrator, the gradients of the approximated integral were *exact*, since the same integration steps were used in the computation of the gradient in eq. (11).

3.2. Examples

Using the above algorithms, we compute minimum-effort motions on planar and spatial open chains using the sequential quadratic programming (SQP) optimizer, NPSOL (Gill et al. 1986). Two examples of open chains are first considered—a two-joint planar arm, and a 5R planar model of a weight lifter. Each example is given 1 sec to perform a point-to-point motion, with the additional requirement that the joint velocities are zero at the endpoints.

The two-joint planar arm was modeled as two thin rods of similar masses and lengths (0.25 m, 0.5 kg), following a path consisting of a uniform, cubic B-spline with seven knots. Figures 3–5 show stop-frame motion at constant time intervals of the original path, the optimized path without gravity, and the optimized path in the presence of gravity. The original path uses the most effort at the beginning of the motion, accelerating against the forces of gravity, whereas the final paths store kinetic energy in a swinging motion and then use that energy to move up toward the final position. The costs for these motions were $J = 3.74$ for the initial motion with gravity, and $J = 1.36$ for the final motion with gravity. The optimization took approximately 40 sec to compute on an SGI Indigo 2 running at 150 MHz.

As a more-interesting example, we examine a 5R planar model of a weight lifter. The person is approximately 5-ft, 10-in (1.9 m) in height, and 80 kg (calves 0.52 m, 14 kg; thighs 0.42 m, 26 kg; torso 0.48 m, 30 kg; upper arm 0.29 m,

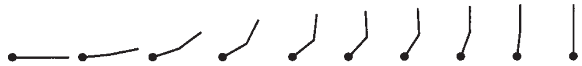

Fig. 3. The 2R planar open chain, original path.

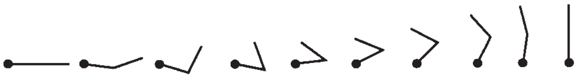

Fig. 4. The 2R planar open chain, no gravity path.

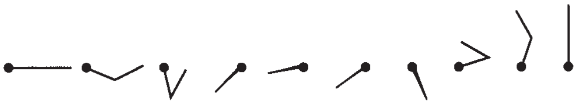

Fig. 5. The 2R planar open chain, gravity path.

6 kg; and forearm 0.33 m, 4 kg), and is lifting a pair of 100-kg weights in a standard gravity field.

The initial, improper, lifting technique is shown in Figure 6, and the final lift is shown in Figure 7. All joints had equal weighting in the cost functional, and the back produced most of the torque load in the original lift. This load was taken up more by the legs in the optimized lift. Figure 8 shows that the integral of the cost function was reduced by an order of magnitude from the original. The major savings in effort came from reducing the torque applied at the beginning of the motion by the back, and the torque applied in the middle of the motion by the shoulders. Of course, the lifter also passes the weights through its knees, which points out the need for adding barrier avoidance to the optimization problem. The path for this problem was a 5-knot, uniform cubic B-spline. The initial and final costs for the weight lifter were 2.25×10^7 and 1.44×10^6, respectively. The optimization took approximately 120 sec to compute.

In this example, *all* the joint parameters were specified in the initial and goal configurations. In the next section, we specify only the desired Cartesian position of the initial and final weight locations, and allow the redundant weight lifter to choose the initial and final values of the joint angles while satisfying the Cartesian constraints. More reduction in total cost is expected for this case, since the lifter has freedom to choose an initial pose to minimize torque requirements.

4. Optimal Robot Motion for Constrained End-Effector Tasks

In this section, we look at the problem of producing minimum-effort motions for tip-based tasks. These tasks consist of trajectories in SE(3) that the robot end effector is required to follow. The end-effector frame can be written from the forward kinematics as the concatenation of the link transformations (Park, Bobrow, and Ploen 1995; Brockett 1984; Murray, Li, and Sastry 1994):

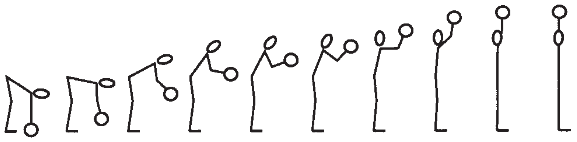

Fig. 6. The 5R planar open chain, original path.

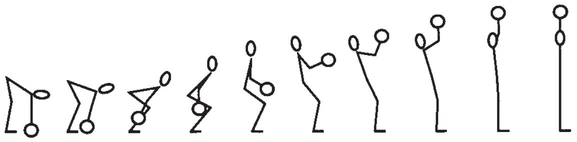

Fig. 7. The 5R planar open chain, optimized path.

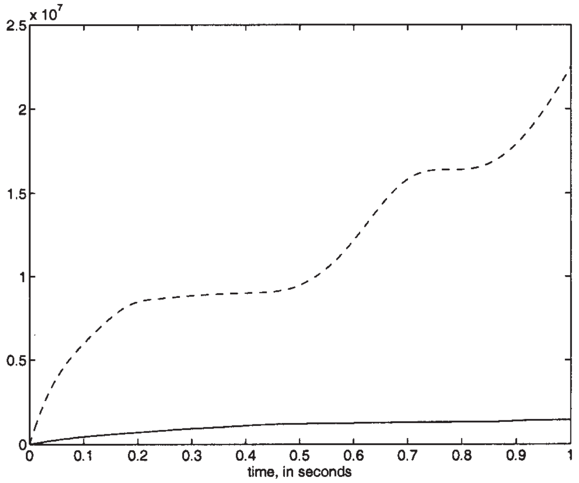

Fig. 8. The integral of the cost function over time for the initial and final paths (5R planar).

$$T_{0,tip} = M_1 e^{S_1 q_1} M_2 e^{S_2 q_2} \cdots M_n e^{S_n q_n} M_{tip}, \tag{12}$$

where M_i is a constant-matrix transformation between link frames, S_i is the joint screw written in the ith link frame, and q_i is the joint variable. Recalling that the joint positions are defined by $q = q(t, P)$, assume for the following discussion that the end-effector frame is written in the form

$$T_{0,tip}(t, P) = \begin{bmatrix} \Theta(t, P) & \mathbf{b}(t, P) \\ 0 & 1 \end{bmatrix}, \tag{13}$$

where $T_{0,tip} \in$ SE(3) is a 4×4 homogeneous transformation, $\Theta \in$ SO(3) is a 3×3 rotation matrix, and $\mathbf{b} \in \Re^3$ is a position vector. The desired end-effector frame location is represented similarly,

$$T_d(t, P_c) = \begin{bmatrix} \Theta_d(t, P_c) & \mathbf{b}_d(t, P_c) \\ 0 & 1 \end{bmatrix}, \tag{14}$$

where P_c is a set of B-spline parameters that define the desired end-effector position and orientation, $\mathbf{b}_d$ and Θ_d. Using the relation $[w_d] = \log \theta_d$, so that $\Theta_d = e^{[\omega_d(t, P_c)]}$, it is apparent that one parameterization of the desired orientations of the end effector in SO(3) can be obtained by specifying a smooth curve for $\omega_d(t, P_c) \in \Re^3$. The desired end-effector motions on SE(3) can then be parameterized with cubic B-spline control points, P_c. Three curves were used to define the desired translational motion of the end-effector $\mathbf{b}_d(t, P_c)$, and three curves were used to define the desired rotational motion of the end-effector $\omega_d(t, P_c)$. For general motions on SE(3), the six independent closure-constraint equations are:

$$g(t, P, P_c) = \begin{bmatrix} \log(\Theta_d^T(t, P_c)\Theta(t, P)) \\ \mathbf{b}(t, P) - \mathbf{b}_d(t, P_c) \end{bmatrix} = 0. \tag{15}$$

Note that if $\Theta = \Theta_d$ in eq. (15), then $\log(I) = 0$, so the constraints on orientation would be satisfied. For $\Theta \neq \Theta_d$, a smooth-orientation error term is generated that represents a screw motion between the two orientations.

One difficulty with eq. (15) is that it does not represent a single spatial constraint, but instead is a continuous constraint over all $t \in [0, t_f]$. Several algorithms have been developed to optimize objective functions with these *semi-infinite* constraints. The optimizers use the constraint minima, maxima, and a few interior points to characterize critical information about the constraints. Some early examples were developed by Hettich (1979, 1986) and by Polak (Polak and Tits 1979; Polak and He 1991, 1992). Recently, work by Panier and Tits (1989) has been implemented in the Matlab optimization toolbox as “semiinf.m.”

Rather than apply the relatively complicated algorithms for handling the semi-infinite constraints, we chose to enforce the constraints at a finite number of points $t_k \in [0, t_f]$, and rely on the continuity of the B-splines to ensure that the constraints are approximately satisfied between the t_k. We enforced the constraints at $2m$ points (m is the number of control points) uniformly spaced in $[0, t_f]$ for the examples given in this research. The software package NPSOL solves the SQP problem by choosing the iterates so that they move along a linear approximation to the nonlinear-constraint surfaces. Hence, it is essential to provide exact gradients of the constraints of eq. (15) to form the linear approximation with accuracy.

4.1. Gradients of the Constraints

Consider the position portion of the constraints from eq. (15):

$$g_p = \mathbf{b} - \mathbf{b}_d = 0, \tag{16}$$

where the subscript in g_p denotes position.

The gradient of these constraints is

$$\nabla_P \, g_p(t, P, P_c) = \nabla_P \, (\mathbf{b} - \mathbf{b}_d) = J_b(t, P)(\nabla_P \, q), \tag{17}$$

where $J_b(t, P)$ is the bottom half of the manipulator Jacobian, as defined by

$$\begin{pmatrix} \omega \\ v \end{pmatrix} = J\dot{q} = \begin{bmatrix} J_t \\ J_b \end{bmatrix} \dot{q}. \tag{18}$$

The Jacobian can be easily obtained from the following procedure (Murray, Li, and Sastry 1994). Let

$$T_{i,j} = M_{i+1} e^{S_{i+1} q_{i+1}} \cdots M_j e^{S_j q_j},$$

then

$$J = \dot{T}\, T^{-1} = \begin{bmatrix} Ad_{T_{0,1}} S_1 & Ad_{T_{0,2}} S_2 & \cdots & Ad_{T_{0,n}} S_n \end{bmatrix}, \tag{19}$$

where the columns of J are elements of SE(3) arranged as six vectors.

Next we examine the orientation constraints for partially and fully constrained orientations.

4.2. Relaxed Orientation Constraints

The orientation portion of the constraint defined in eq. (15) provides three independent equations that fully constrain the orientation of the end effector. If one relaxes the full orientation constraint to allow for cases where the direction of only one axis of the end effector is specified and the remainder are free, these equations must be altered. This type of constraint is most useful when the end effector is required to remain normal to a curve or surface during the motion. If we assume that the end-effector z-axis is required to align with a desired $\hat{z}$-axis, but is free to rotate around it, then the constraint is equivalent to the equation

$$\Theta = \Theta_d \, e^{[\hat{z}]\alpha}, \tag{20}$$

where α is any scalar and $\hat{z}$ is a unit vector. Multiplying by Θ_d^T and taking the matrix logarithm yields

$$\log(\Theta_d^T \Theta) = [\hat{z}\alpha] = [(0, 0, \alpha)], \tag{21}$$

so this constraint is equivalent to setting the first two terms of ω_d to zero. This approach can be applied to any direction in the end-effector frame by adding an additional constant transformation to the end effector that aligns the z-axis with the desired direction.

4.3. Orientation-Constraint Gradient

To compute gradients of the orientation constraints in eq. (15), it is necessary to compute the gradient of the matrix log function. Assume that $[h] = \log(\Theta_d^T \Theta_{0,tip}) \in$ so(3) and $\Theta_{0,tip} = \Theta_{M_1} e^{\omega_1 q_1} \Theta_{M_2} e^{\omega_2 q_2} \cdots \Theta_{M_n} e^{\omega_n q_n} \Theta_{M_{tip}} \in$ SO(3), with $q_i = q_i(t, P)$. In Appendix B, it is shown that

$$\nabla_q \, h = Q_h^{-1} \begin{bmatrix} \mathrm{Ad}_{\Theta_d^T \Theta_{0,1}} \omega_1 & \cdots & \mathrm{Ad}_{\Theta_d^T \Theta_{0,n}} \omega_n \end{bmatrix}, \tag{22}$$

where

$$Q_h = I + \frac{1 - \cos h}{h^2}[h] + \frac{h - \sin h}{h^3}[h]^2. \tag{23}$$

4.4. The Optimization Algorithm

We now summarize the optimization algorithm using the equations derived previously:

Algorithm MinEffort.

Given an open-chain robot **R**, a Cartesian path $(\mathbf{b}_d(t, P_c), \omega_d(t, P_c))$, and the constraint equation $g(t, P, Pc)$, compute the minimum-effort robot motion as follows:

Initialization.

1. At the initial time, $t = 0$, find the joint displacements that minimize $g^T g$. That is, produce a local solution $q^*(0)$ to the inverse-kinematics problem. Define $q_0 = q^*(0)$.

2. Produce a set of points q_i corresponding to uniformly spaced t_i, which satisfies the equation $g(t_i, q_i, P_c) = 0$, using the following iterative process starting with $i = 1$:
   (a) using the minimum-norm pseudoinverse Jacobian and the initial condition $q(0) = q_{i-1}$, integrate the equation $\dot{q} = J^{+} \frac{d}{dt} \begin{pmatrix} \mathbf{b}_d(t, P_c) \\ \omega_d(t, P_c) \end{pmatrix}$ until either a singularity is reached, or $t = t_{i+1}$;
   (b) set $i = i + 1$; and
   (c) if $g(t_i, q_i, P_c) \neq 0$, solve the minimization problem in step 1 for $q_i = q(t_i)$, starting from q_{i-1}.
3. Construct the cubic interpolating B-spline from the joint displacements (q_i, t_i).

Step ObjConst.

1. Compute the objective function $J = \frac{1}{2} \int_{t_0}^{t_1} \tau^T \tau \, dt$, where τ is computed using the recursive Newton-Euler equations from Figure 1.
2. Compute the k-constraint functions $g(t_i, P, P_c)$.

Step ObjConstGrad.

1. Compute the objective gradient $\nabla_P J = \int_{t_0}^{t_1} \tau^T \nabla_P \tau \, dt$, where $\nabla_P \tau$ is computed using the recursive Newton-Euler derivative formula in Figure 2.
2. Compute the constraint gradient from eqs. (22) and (23)

4.5. Examples

The weight lifter problem was solved again without specification of the initial joint angles, but with the addition of a Cartesian constraint on the initial position of the weights. For this case, at the initial time $t = 0$, the end-effector constraint had the same form as in eq. (15). There was no constraint on the end-effector motion at instants between the initial and final times. Figure 9 shows the new 200-kg, minimum-effort, 1-sec motion, with a cost of 9.25×10^5 as compared to the 14.4×10^5 cost from the previous optimal solution shown in Figure 7. This looks a lot like the Olympic lift, the “snatch.” The only difference is that in the snatch the lifter pauses with the weight over his head before lifting with the legs.

Figures 10–12 demonstrate another interesting feature of our approach to solving the optimal control problem. If the final time t_f specified is *longer* than that needed for the minimum-effort motion, the algorithm finds the t_f needed for the free final-time problem. That is, the optimal motions will come to rest before they reach the specified final time.

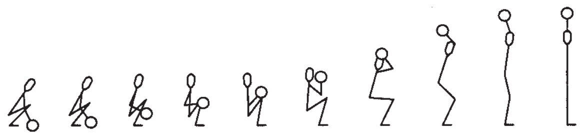

Fig. 9. A 200-kg lift with initial repositioning.

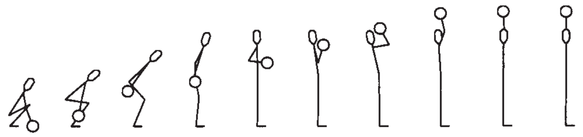

Fig. 10. A 40-kg lift, free end-time path with initial position reconfiguration, $t_f = 2.5$ sec, $J = 5.15 \times 10^4$.

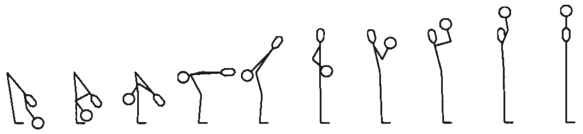

Fig. 11. A 40-kg lift, local minimum with $t_f = 2.0$ sec, $J = 1.76 \times 10^4$.

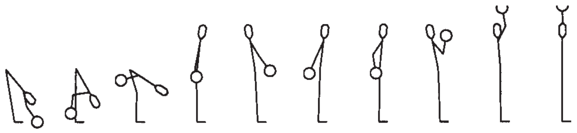

Fig. 12. A 40-kg lift, local minimum with $t_f = 3.5$ sec, $J = 1.77 \times 10^4$.

Figure 10 shows a 40-kg minimum-effort lift for the free end-time problem. In this case, a 4-sec initial motion is specified, and the final motion is completed in 2.5 sec with a cost of 5.15×10^4. Figure 11 shows another local minimum solution to this problem, which completed in 2.0 sec with a cost of 1.76×10^4. The last example in Figure 12 is a similar local minimum, but there is an extra oscillation in the motion. The motion time for this case was 3.5 sec, with a cost of 1.77×10^4.

Almost all of the examples in this work had high condition numbers for the approximate Hessian generated during the nonlinear parameter optimization. This means that the optimization would be difficult to solve without exact gradients. As a typical example, Figure 13 shows the condition number of the approximate Hessian matrix generated during the optimization process for the problem of Figure 10, versus the iteration number. The same plot shows the value of the objective function, J. One reason for the poor conditioning in these problems is that some regions during the motion require large torques, such as the acceleration and deceleration phases, while other regions require little or no torques. The B-spline parameters in the high-torque regions have much greater effect on J than those in the low-torque regions.

Figures 14–16 show the original and optimized motions for a Mitsubishi PA-10 robot tracing a planar path. The original motivation for this problem was that the arm is to be mounted to an undersea vehicle and used for the inspection of welds. Because the entire vehicle is powered electrically, it is essential to conserve energy. The energy consumed by the robot is given approximately by the cost function of eq. (8), since torque is proportional to current. This robot has seven revolute joints, and is tracing the curve shown with a specified orientation with respect to the path. The original motion shown in Figure 14 had a cost of $J = 5.02 \times 10^3$, and the optimized motion shown in Figure 15 had a cost of 4.61×10^3. If one allows the base position to be a parameter for the optimization, the same operation could be performed as shown in Figure 16 for a cost of 1.15×10^3. An interesting feature of the last solution found is that the base of the robot was placed in a location that required very little torque from the robot's base joints, which are the most difficult to move.

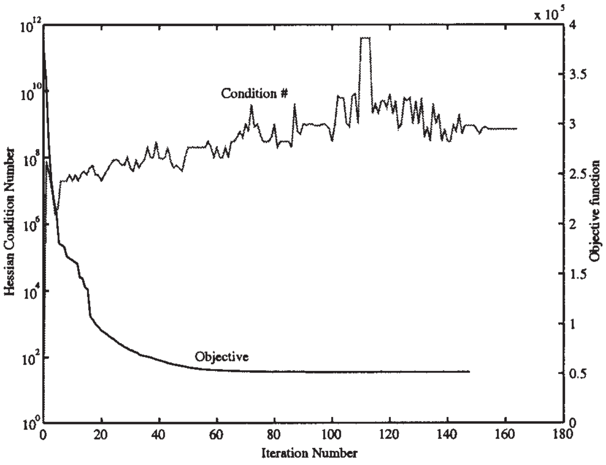

Fig. 13. Condition number of Hessian and cost function versus iteration number for the problem in Figure 10.

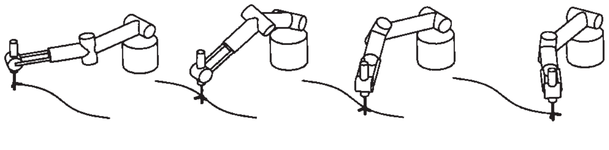

Fig. 14. The Mitsubishi PA-10 7-DOF arm tracing a path.

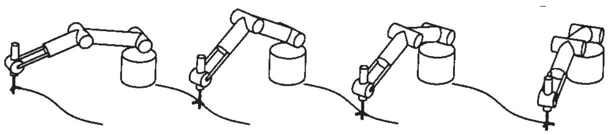

Fig. 15. The Mitsubishi PA-10 7-DOF arm tracing a path with optimized effort.

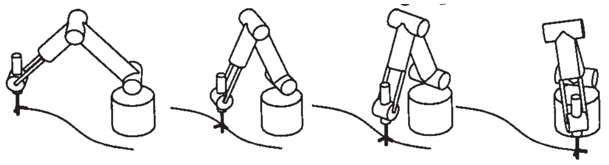

Fig. 16. The Mitsubishi PA-10 7-DOF arm tracing a path with optimal base location.

5. Conclusion

By using B-splines to reformulate the minimum-effort optimal control problem as a discrete parameter-optimization problem, and by using the matrix exponential formulation of the kinematics and dynamics of open-chain mechanisms, we have developed a new approach to motion optimization of open chains with end-effector constraints. We have shown that to reliably obtain local minimizers of the objective function, exact gradients of the dynamics and the constraints are needed, owing to the inherent ill-conditioning of the problem. We have developed efficient, recursive, analytic expressions for these gradients of the dynamics of general open chains. Our examples have reliably produced results for minimum-effort motions of open chains with complex nonlinear dynamics, time-varying constraints, and redundant degrees of freedom. The new motion-optimization software provides a powerful tool for the analysis and design of articulated systems.

Appendix A: Notation and Derivation of the Recursive Derivative

Standard mathematical notation applies whenever possible in this text. In particular, the gradient operator ∇_x , when applied to a scalar, results in a row vector the dimension of x:

$$\nabla_x \tau_i = \begin{bmatrix} \frac{\partial \tau_i}{\partial x_1} & \frac{\partial \tau_i}{\partial x_2} & \cdots & \frac{\partial \tau_i}{\partial x_m} \end{bmatrix}, \tag{24}$$

and the gradient of the vector-valued function is a matrix:

$$\nabla_x \tau = \begin{bmatrix} \frac{\partial \tau_1}{\partial x_1} & \frac{\partial \tau_1}{\partial x_2} & \cdots & \frac{\partial \tau_1}{\partial x_m} \\ \frac{\partial \tau_2}{\partial x_1} & \frac{\partial \tau_2}{\partial x_2} & \cdots & \frac{\partial \tau_2}{\partial x_m} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \tau_n}{\partial x_1} & \frac{\partial \tau_n}{\partial x_2} & \cdots & \frac{\partial \tau_n}{\partial x_m} \end{bmatrix}. \tag{25}$$

For the algorithm in Figure 1, V_i is the six-dimensional generalized velocity of link i, expressed in the coordinates of link i, $S_i \in$ se(3), and

$$J_i = \left[\begin{array}{cc} I_i - m_i[r_i]^2 & m_i[r_i] \\ -m_i[r_i] & m_i \cdot \mathbf{1} \end{array} \right] \qquad (26)$$

is the 6×6 "inertia" matrix, where r_i is a vector from the origin of the link-i frame to the center of mass of link i. A vector variable in brackets ($[r]$ where $r \in \Re^n$) is taken as the skew-symmetric matrix formed by using the vector to represent the free parameters of the matrix.

Notation and Operators

The *special Euclidean group*, or SE(3), is a Lie group corresponding to the homogeneous transformations on $\Re^3$, so that for $G \in$ SE(3),

$$G = \left[\begin{array}{cc} \Theta & \mathbf{b} \\ 0 & 1 \end{array} \right]. \qquad (27)$$

The Lie algebra associated with SE(3) is referred to as se(3), where $g \in$ se(3) has the form:

$$g = \left[\begin{array}{cc} [\omega] & v \\ 0 & 0 \end{array} \right]. \qquad (28)$$

The spatial rotations represent a subgroup of SE(3), and are referred to as the *special orthogonal group* SO(3), and its associated algebra so(3).

We refer to elements of the Lie group by capital letters, G or H typically (or T, where it matches the accepted use for homogeneous transformations), and the associated member of the Lie algebra by the matching lowercase letters. An element of the Lie group can be used as a linear mapping on the Lie algebra, and this is called the *adjoint map* on SE(3). It is commonly denoted by Ad, and is given by

$$\mathrm{Ad}_G(h) = G\ h\ G^{-1}, \qquad (29)$$

for $G \in$ SE(3), $h \in$ se(3), where h is in the 4×4 matrix form of eq. (28). Similarly, the Lie algebra can be used as a linear mapping on itself, denoted by ad, which is also called the *Lie bracket*. The Lie bracket has the form

$$\mathrm{ad}_g(h) = [g, h] = gh - hg, \qquad (30)$$

for $g,\ h \in$ se(3).

The dual adjoint is a mapping on the space of wrenches, where the dual value g is denoted by g^*. Where a screw has the form of eq. (28), the dual screw $g^* = (w, f)$ has the 4×4 matrix representation

$$g^* = \left[\begin{array}{cc} [f] & w \\ 0 & 0 \end{array} \right].$$

The adjoint and dual-adjoint mappings may be represented as either 4×4 matrices or as operations on the vector form of se(3) and se$^*(3)$, which have the following forms:

$$\begin{aligned}
\mathrm{Ad}_G(h) &= \left[\begin{array}{cc} \Theta & 0 \\ [b]\Theta & \Theta \end{array} \right] \left[\begin{array}{c} \omega_h \\ v_h \end{array} \right], \\
\mathrm{Ad}^*_G(h^*) &= \left[\begin{array}{cc} \Theta^T & \Theta^T\,[b]^T \\ 0 & \Theta^T \end{array} \right] \left[\begin{array}{c} \omega_{h^*} \\ v_{h^*} \end{array} \right], \\
\mathrm{ad}_g(h) &= \left[\begin{array}{cc} [\omega_g] & 0 \\ [v_g] & [\omega_g] \end{array} \right] \left[\begin{array}{c} \omega_h \\ v_h \end{array} \right], \\
\mathrm{ad}^*_g(h^*) &= \left[\begin{array}{cc} [\omega_g]^T & [v_g]^T \\ 0 & [\omega_g]^T \end{array} \right] \left[\begin{array}{c} \omega_{h^*} \\ v_{h^*} \end{array} \right],
\end{aligned}$$

where $G = (\Theta, b)$, $g = (\omega_g, v_g)$, and $h = (\omega_h, v_h)$.

The matrix exponential and logarithm both admit a closed-form expression as mappings between SE(3) and se(3). The matrix exponential (Park 1991) can be computed thus:

$$\exp \left[\begin{array}{cc} [\omega] & v \\ 0 & 0 \end{array} \right] = \left[\begin{array}{cc} \exp([\omega]) & Av \\ 0 & 1 \end{array} \right], \qquad (31)$$

where

$$\begin{aligned}
\phi &= ||\omega||, \\
A &= I + \frac{1 - \cos\phi}{\phi^2}\,[\omega] + \frac{\phi - \sin\phi}{\phi^3}\,[\omega]^2, \\
\exp([\omega]) &= I + \frac{\sin\phi}{\phi}\,[\omega] + \frac{1 - \cos\phi}{\phi^2}\,[\omega]^2.
\end{aligned}$$

The closed-form expression for the matrix logarithm is

$$\log \left[\begin{array}{cc} \Theta & b \\ 0 & 1 \end{array} \right] = \left[\begin{array}{cc} [\omega] & A^{-1}b \\ 0 & 0 \end{array} \right], \qquad (32)$$

where

$$\begin{aligned}
[\omega] &= \frac{\phi}{2\sin\phi}\left(\Theta - \Theta^T\right), \\
A^{-1} &= I - \tfrac{1}{2} \cdot [\omega] + [\omega]^2 \cdot \frac{2\ \sin\phi - \phi(1 + \cos\phi)}{2\phi^2 \sin\phi},
\end{aligned}$$

and ϕ satisfies $1 - 2\cos\phi = Tr(\Theta)$, $|\phi| < \pi$. (Also, $\phi^2 = ||\omega||^2$.)

Computing the Recursive Gradient

To compute derivatives of the recursive formulation, we must be able to compute derivatives of the matrix exponential and of the adjoint maps: ad, Ad, and Ad* (Martin and Bobrow 1995). If we assume $g = S_g q_g$ and $h = S_h q_h$ are two screws associated with rigid-body motions, where $S_g, S_h \in$ se(3) and q_g, q_h are scalars linearly dependent on the parameter P, then $G = e^g$ and $H = e^h$ are the homogeneous transformations associated with those motions, and their derivatives have the form $\frac{d}{dp_i}G = S_g G \frac{dq_g}{dp_i}$. The derivatives of the adjoint mappings on curves defined by eq. (7) are determined as follows.

Recall the definition of the adjoint map,

$$\mathrm{Ad}_G\ h = G\, h\, G^{-1},$$

which may also be written as

$$\mathrm{Ad}_G\ h = e^g h\, e^{-g}$$

for some $g \in$ se(3), and similarly for the dual-adjoint map,

$$\mathrm{Ad}^*_G\ ; h^* = G^{-1} h^*\, G = e^{-g} h^*\, e^g.$$

For the following, it is assumed that $g = S_g x_g$, $G = e^g$, $h = S_h x_h$, and $H = e^h$, where x_g and x_h are linear functions of the p_i (see the work of Gilbert and Johnson (1985)), and implicit functions of t of unknown degree.

Writing out the term-by-term derivative for the adjoint mapping,

$$\begin{aligned}\frac{d}{dp_i}\mathrm{Ad}_G\ h \\ &= \tfrac{dx_g}{dp_i} S_g e^g h e^{-g} + e^g \tfrac{dx_h}{dp_i} S_h e^{-g} - e^g h e^{-g} \tfrac{dx_g}{dp_i} S_g, &(33)\\ &= e^g S_h e^{-g} \tfrac{dx_h}{dp_i} + \left(S_g e^g h e^{-g} - e^g h e^{-g} S_g\right) \tfrac{dx_g}{dp_i}, &(34)\end{aligned}$$

where the identity $g e^g = e^g g$ has been used. Note that the first term is simply the adjoint map again, and the second term looks like the Lie bracket performed on S_g and $e^g h e^{-g}$, which leads to:

LEMMA 1.

$$\frac{d}{dp_i}\mathrm{Ad}_G\ h = \mathrm{ad}_{S_g}\ \mathrm{Ad}_G\ h \left(\frac{dq_g}{dp_i}\right) + \mathrm{Ad}_G\ S_h \left(\frac{dq_h}{dp_i}\right). \tag{35}$$

We can similarly derive the relation for the Ad* operator:

LEMMA 2.

$$\frac{d}{dp_i}\mathrm{Ad}^*_G\ h^* = \mathrm{ad}_{\mathrm{Ad}^*_G h}\ S_h \left(\frac{dq_g}{dp_i}\right) + \mathrm{Ad}^*_G\ S_h \left(\frac{dq_h}{dp_i}\right). \tag{36}$$

Note that these two identities still hold true if we replace $G = e^g$ with $e^g M$ ($M \in$ SE(3) and constant), but not if we replace it with $M e^g$. This leads to another relation:

LEMMA 3. For $G = M e^g$,

$$\frac{d}{dp_i}\mathrm{Ad}_G\ h = \mathrm{ad}_{\mathrm{Ad}_{M_i} S_g}\ \mathrm{Ad}_G\ h \left(\frac{dq_g}{dp_i}\right) + \mathrm{Ad}_G\ S_h \left(\frac{dh}{dp_i}\right). \tag{37}$$

Finally, the derivatives for the ad and ad* operators are:

$$\frac{d}{dp_i}\mathrm{ad}_g\ h = \mathrm{ad}_{S_g}\ h \left(\frac{dq_g}{dp_i}\right) + \mathrm{ad}_g\ S_h \left(\frac{dq_h}{dp_i}\right), \tag{38}$$

$$\frac{d}{dp_i}\mathrm{ad}^*_g\ h^* = \mathrm{ad}^*_{S_g}\ h \left(\frac{dq_g}{dp_i}\right) + \mathrm{ad}^*_g\ S_h \left(\frac{dq_h}{dp_i}\right). \tag{39}$$

The preceding identities were used to derive the recursive derivative shown in Figure 2 from the recursive-torque computation in Figure 1.

Appendix B: Derivation of the Orientation-Constraint Gradient

To compute gradients of the orientation-constraint equations, we must first compute the gradient of the matrix log function for our particular homogeneous transform. This result was derived by Park and Bobrow (1995), and is stated here in a slightly different form which is best suited for our computation:

LEMMA 4. Given a constraint equation of the form $[g] = \log e^{[h]} = 0 \in$ so(3), where $e^{[h]} = \Theta_d^T \Theta_{0,tip} \in$ SO(3), $\Theta_{0,tip} = \Theta_{M_1} e^{\omega_1 q_1} \Theta_{M_2} e^{\omega_2 q_2} \cdots \Theta_n e^{\omega_n q_n} \Theta_{M_{tip}} \in$ SO(3), ω_i is from $S_i = (\omega_i, \mathbf{b}_i)$, $q_i = q_i(t, P)$, Θ_{M_i} is the 3×3 rotation matrix associated with M_i, and $\Theta_{i,i+1} = \Theta_{M_i} e^{\omega_i q_i}$, the gradient of g satisfies:

$$\nabla_q g = Q_h^{-1} \left[\ \mathrm{Ad}_{\Theta_d^T \Theta_{0,1}} \omega_1 \quad \cdots \quad \mathrm{Ad}_{\Theta_d^T \Theta_{0,n}} \omega_n\ \right], \tag{40}$$

where

$$Q_h = I + \frac{1 - cos||h||}{||h||^2}[h] + \frac{||h|| - sin||h||}{||h||^3}[h]^2. \tag{41}$$

Proof. Given $e^h = \Theta_d^T \Theta_{M_1} e^{\omega_1 q_1} \cdots \Theta_{M_n} e^{S_n \omega_n} \Theta_{M_{tip}}$, we take the derivative with respect to a joint variable, q_i:

$$\begin{aligned}\left(\frac{\partial}{\partial q_i} e^h\right) e^{-h} \\ &= \left(\Theta_d^T \cdots \Theta_{M_i} e^{\omega_i q_i} \omega_i \Theta_{i+1} \cdots \Theta_{M_n} e^{\omega_n q_n} \Theta_{M_{tip}}\right) * \\ &\quad \left(\Theta_{M_{tip}}^T e^{-\omega_n q_n} \Theta_{M_n}^T \cdots \Theta_{M_1}^T e^{-\omega_1 q_1} \Theta_d\right), &(42)\\ &= \Theta_d^T \Theta_{M_1} e^{\omega_1 q_1} \ldots \Theta_{M_i} e^{\omega_i q_i} \omega_i e^{-\omega_i q_i} \Theta_{M_i}^T \cdots \\ &\quad \Theta_{M_1}^T e^{-\omega_1 q_1} \Theta_d, &(43)\\ &= \mathrm{Ad}_{\Theta_d^T \Theta_{0,i}} \omega_i. &(44)\end{aligned}$$

The left side of the previous equation is in the form of a theorem derived by Park (1991), which states that for $X = e^{A(t)} \in G$,

$$\dot{X} X^{-1} = \int_0^1 e^{A(t)s} \dot{A}(t) e^{-A(t)s} ds.$$

In this case, $X \in$ SO(3), and the derivative is not with respect to time but to some parameter q. Recalling that $[\omega(q)] = \frac{d\Theta}{dq}\Theta^T$, and noting that the integrand is an adjoint map on so(3) and therefore can be rewritten in vector form, the theorem can be rewritten as:

$$\omega(q) = \int_0^1 e^{[\omega(q)]s} \frac{d\omega(q)}{dq} ds = Q_\omega \frac{d\omega(p)}{dq}, \tag{45}$$

where $Q_\omega = \int_0^1 e^{[\omega]s} ds$. This matrix can be found by integrating the closed-form equation for the matrix exponential of ω.

Equating the right side of eq. (44) to the right side of eq. (45) and replacing the generic variable ω with h,

$$Q_h \frac{dh}{dq_i} = \mathrm{Ad}_{\Theta_d^T \Theta_{0,i}} \omega_i .$$

Inverting Q_h yields the equation for a single column of the gradient:

$$\frac{dh}{dq_i} = Q_h^{-1} \mathrm{Ad}_{\Theta_d^T \Theta_{0,i}} \omega_i .$$

The single equation may be expanded to a matrix equation containing all of the columns of the gradient:

$$\begin{aligned} \nabla_q h &= \left[\begin{array}{cccc} \frac{dh}{dq_1} & \frac{dh}{dq_2} & \cdots & \frac{dh}{dq_n} \end{array} \right], \\ &= \left[\begin{array}{ccc} Q_h^{-1} \mathrm{Ad}_{\Theta_d^T \Theta_{0,1}} \omega_1 & \cdots & Q_h^{-1} \mathrm{Ad}_{\Theta_d^T \Theta_{0,n}} \omega_n \end{array} \right], \\ &= Q_h^{-1} \left[\begin{array}{ccc} \mathrm{Ad}_{\Theta_d^T \Theta_{0,1}} \omega_1 & \cdots & \mathrm{Ad}_{\Theta_d^T \Theta_{0,n}} \omega_n \end{array} \right]. \end{aligned}$$

□

References

Angeles, J., Rojas, A., and Lopez-Cajun, C. S. 1988. Trajectory planning in robotics continuous-path applications. *IEEE J. Robot. Automat.* 4(4):380–385.

Balafoutis, C., Misra, P., and Patel, R. 1986. Recursive evaluators of linearized robot dynamic models. *IEEE J. Robot. Automat.* 2(3):146–155.

Balafoutis, C., and Patel, R. 1991. *Dynamic Analysis of Robot Manipulators. A Cartesian Tensor Approach.* Boston, MA: Kluwer Academic.

Bobrow, J. E. 1988. Optimal robot plant planning using the minimum-time criteriron. *IEEE J. Robot. Automat.* 4(4):443–450.

Bobrow, J. E., Dubowsky, S., and Gibson, J. S. 1985. Time-optimal control of robotic manipulators along specified paths. *Intl. J. Robot. Res.*

Bobrow, J. E., and Park, F. C. Forthcoming. On computing exact gradients for rigid-body guidance using screw parameters.

Brockett, R. W. 1984. Robotic manipulators and the product of exponentials formula. *Mathematical Theory of Networks and Systems*, pp. 120–129.

Bryson, Jr., A. E., and Meier, E. B. 1990. Efficient algorithm for time-optimal control of a two-link manipulator. *J. Guidance Control Dyn.* 13(5):859–866.

Chen, T.-H., Cheug, F.-T., and Sun, Y.-Y. 1994. Torque-optimization schemes for kinematically redundant manipulators. *J. Robot. Sys.* 11(4):257–269.

De Boor, C. 1978. *A Practical Guide to Splines.* New York: Springer-Verlag.

Dubowsky, S., Norris, M. A., and Shiller, Z. 1986. Time-optimal trajectory planning for robotic manipulators with obstacle avoidance: A CAD approach. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.*, vol. 3. Los Alamitos, CA: IEEE, pp. 1906–1912.

Faux, I. D., and Pratt, M. J. 1979. *Computational Geometry for Design and Manufacture: Mathematics and Its Applications*, 4th ed. Wiley.

Field, G., and Stepanenko, Y. 1996. Iterative dynamic programming: An approach to minimum-energy trajectory planning for robotic manipulators. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 2755–2760.

Geering, H., Guzzella, L., Hepner, S., and Onder, C. 1986. Time-optimal motions of robots in assembly tasks. *IEEE Trans. Automatic Control* 31(6):512–518.

Gilbert, E. G., and Johnson, D. W. 1985. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE J. Robot. Automat.* 1:21–30.

Gilbert, E. G., and Ong, C. J. 1994. Robot path planning with penetration growth distance. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Los Alamitos, CA: IEEE Computer Society Press, pp. 2146–2152.

Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H. 1986. *User's Guide for NPSOL (Version 4.0).* Systems Optimization Laboratory, Department of Operations Research, Stanford University.

Gill, P. E., Murray, W., and Wright, M. H. 1981. *Practical Optimization.* Academic Press.

Hettich, R. 1979. A comparison of some numerical methods for semi-infinite programming. *Semi-Infinite Programming: Proceedings of a Workshop*, pp. 112–125.

Hettich, R. 1986. An implementation of a discretization method for semi-infinite programming. In *Mathematical Programming*, pp. 354–361.

Hu, B., Teo, C., and Lee, H. 1994. Relation between global velocity and local torque optimization of redundant manipulators. *J. Robot. Sys.* 11(4):271–279.

Kahn, M. E., and Roth, B. 1971. The near-minimum-time control of an open-loop articulated kinematic chain. *Trans. ASME* 93(3):164–172.

Kazerounian, K., and Wang, Z. 1988. Global versus local optimization in redundancy resolution of robotic manipulators. *Intl. J. Robot. Res.* 7(6):3–12.

Kenjo, T., and Nagamori, S. 1986. *Permanent-Magnet and Brushless DC Motors.* Oxford, UK: Clarendon Press.

Kim, S.-W., Park, K.-B.,. and Lee, J.-J. 1994. Redundancy resolution of robot manipulators using optimal kinematic control. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Los Alamitos, CA: IEEE, pp. 683–688.

Lu, P. 1992. Use of approximate gradients in trajectory optimization. *J. Guidance Control Dyn.* 15(5):1299–1301.

Luenberger, D. G. 1989. *Linear and Nonlinear Programming,* 2nd ed. Reading, MA: Addison-Wesley.

Ma, S. 1996. A stabilized local torque optimization technique for redundant manipulators. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 2791–2795.

Malladi, S. R., Mulder, W. C., Valvanis, K. P., and Zhang, Y. 1992. Behavior of a minimum-effort control algorithm for a multijointed robotic arm. *Proc. of the IEEE Intl. Symp. on Intell. Control.* Los Alamitos, CA: IEEE, pp. 34–41.

Martin, B. J., and Bobrow, J. E. 1995. Determination of minimum-effort motions for general open chains. *Proc. of the 1995 IEEE Intl. Conf. on Robot. and Automat.*, vol. 1. Washington, DC: IEEE, pp. 1160–1165.

Martin, D. P., Baillieul, J., and Hollerbach, J. M. 1989. Resolution of kinematic redundancy using optimization techniques. *IEEE Trans. Robot. Automat.* 5(4):529–533.

Murray, J., and Neuman, C. 1986. Linearization and sensitivity models of the Newton-Euler dynamic robot model. *ASME J. Dyn. Sys. Meas. Control* 108(3):272–276.

Murray, R. M., Li, Z., and Sastry, S. S. 1994. *A Mathematical Introduction to Robotic Manipulation.* Boca Raton, FL: CRC Press.

Ozaki, H., and Lin, C. 1996. Optimal B-spline joint trajectory generation for collision-free movements of a manipulator under dynamic constraints. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 3592–3597.

Panier, E. R., and Tits, A. L. 1989. A globally convergent algorithm with adaptively refined discretization for semi-infinite optimization problems arising in engineering design. *IEEE Trans. Automatic Control* 34(8):903–908.

Park, F. C. 1991. The optimal kinematic design of mechanisms. Ph.D. thesis, Division of Applied Sciences, Harvard University.

Park, F. C., and Bobrow, J. E. 1994. A recursive algorithm for robot dynamics using Lie groups. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Los Alamitos, CA: IEEE Computer Society Press, pp. 1535–1540.

Park, F. C., and Bobrow, J. E. 1995. Geometric optimization algorithms for robot kinematic design. *J. Robot. Sys.* 12(6):453–463.

Park, F. C., Bobrow, J. E., and Ploen, S. R. 1995. A Lie group formulation of robot dynamics. *Intl. J. Robot. Res.* 14(6):609–618.

Park, F. C., and Brockett, R. W. 1994. Kinematic dexterity of robotic mechanisms. *Intl. J. Robot. Res.* 13(1):1–15.

Park, J.-K. 1997. Minimum-effort motions for manipulators with obstacles. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE.

Pledel, P., and Bestaoui, Y. 1995. Actuator constraints in optimal motion planning of manipulators. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 2427–2432.

Polak, E., and He, L. 1991. Unified steerable phase I-phase II method of feasible direction for semi-infinite optimization. *J. Optimization Theory Applications* 69(1):83–107.

Polak, E., and He, L. 1992. Rare preserving discretization strategies for Semi-infinite programming and control. *Siam J. Control Optimization* 30(3):548–572.

Polak, E., and Tits, A. L. 1979. Combined phase I-phase II methods of feasible directions. *Math. Programming* 17:61–73.

Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F. 1962. *The Mathematical Theory of Optimal Processes.* Interscience.

Seereeram, S., and Wen, J. T. 1993. A global approach to path planning for redundant manipulators. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.*, vol. 2. Los Alamitos, CA: IEEE, pp. 283–288.

Seereeram, S., and Wen, J. T. 1995. A global approach to path planning for redundant manipulators. *IEEE Trans. Robot. Automat.* 11(1):152–160.

Shiller, Z. 1994. Time-energy optimal control of articulated systems with geometric path constraints. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Los Alamitos, CA: IEEE.

Shin, K. G., and McKay, N. D. 1985. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Trans. Automatic Control* 30(6):531–541.

Suh, K. C., and Hollerbach, J. M. 1987. Local versus global torque optimization of redundant manipulators. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.*, vol. 2. Los Alamitos, CA: IEEE, pp. 619–624.

Tabarah, E., Benhabib, B., and Fenton, R. G. 1994. Optimal motion coordination of two robots—a polynomial parameterization approach to trajectory resolution. *J. Robot. Sys.* 11(7):615–629.

Wang, L. T., and Chen, C. C. 1991. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Trans. Robot. Automat.* 7(4):489–499.

Žefran, M., and Kumar, V. 1995. Optimal control of systems with unilateral constraisits. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 2695–2700.

Žefran, M., Kumar, V., and Yun, X. 1994. Optimal trajectories and force distribution for cooperating arms. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Los Alamitos, CA: IEEE, pp. 874–879.