Reprinted from

ROBOT CONTROL 1988 (SYROCO '88)

Selected Papers from the 2nd IFAC Symposium,
Karlsruhe, FRG, 5–7 October 1988

Edited by
U. REMBOLD
Institut für Prozeßrechentechnik und Robotik,
Universität Karlsruhe, FRG

Published for the
INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL
by
PERGAMON PRESS
OXFORD · NEW YORK · BEIJING · FRANKFURT
SÃO PAULO · SYDNEY · TOKYO · TORONTO

A SENSITIVITY APPROACH TO OPTIMAL SPLINE ROBOT TRAJECTORIES

A. De Luca, L. Lanari, G. Oriolo and F. Nicolò

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Eudossiana 18, 00184 Roma, Italy

Abstract. A robot trajectory planning problem is considered. Using interpolating cubic splines as joint space trajectories, the path is parametrized in terms of the n-1 time intervals between the n knots. A minimum time optimization problem is formulated under maximum dynamic torque and maximum velocity constraints and solved by means of a first order derivative-type algorithm for semi-infinite nonlinear programming. Feasible directions in the parameter space are generated using sensitivity coefficients of the active constraints. Numerical simulations for a two-link Scara robot are reported. This approach can be used also for more general objective functions and including different types of constraints.

Keywords. Robots, Splines, Optimization, Nonlinear programming, Sensitivity analysis, Trajectory planning.

Introduction

The optimal planning of robot trajectories is a relevant problem for improving robotic performance in industrial applications. Difficulties arise in the robot motion planning and control due to the integrated nature of the problem, with interacting geometric, kinematic and dynamic issues.

Essentially, there are two different types of tasks to be tackled by the robot trajectory planner: moving from point to point and moving along a given path. In both cases, a common objective is to minimize the total traveling time. Other general criterions combine minimization of time with fuel or energy consumption.

It has been recognized that inclusion of robot dynamics into the traditionally purely kinematic phase of path planning allows to specify feasible or even optimal reference trajectories to be tracked in real time by the robot controller (Brady, 1982). Methods which explicitly consider bounds on the input torques need to take into account this dynamics in the optimization.

Several techniques have been proposed for solving the point-to-point minimum time problem, using the manipulator full dynamics. They differ in the algorithmic approach followed: an adaptive selection of switching times (Gawronski et al., 1981), a modified gradient-type optimal control algorithm (Weinreb and Bryson, 1985), the multiple-shooting technique for nonlinear TPBVP (Geering et al.,1986), or a dynamic programming scheme (Sahar and Hollerbach, 1986). Issues related to the existence of time-optimal solutions and of singular arcs have been investigated by Ailon and Langholz (1985). All these numerical methods are computationally intensive. Moreover, the introduction of state constraints, like joint limits or maximum velocity bounds, brings in additional complexity.

The time-optimal behavior between the initial and final points is unpredictable and possibly dangerous. Therefore, a sequence of via points is usually assigned, which have to be interpolated in an optimal way. Other robotic tasks explicitly require that the arm follows a given geometric path, typically in minimum time. In these cases, most solution methods are based on a continuous parametrization of the path.

Hollerbach (1984) used constant time scaling on parametric robot trajectories in order to recompute feasible torques for moving on the path. The generalization to time-varying scaling has brought to the successful method of Bobrow et al. (1985) and of Shin and McKay (1985) for finding exact minimum-time motion on the given path. In the phase-plane of scalar position and velocity along the path, an efficient solution algorithm is derived under maximum torque constraints. A closely related technique allowing more general objective functions is due to Pfeiffer and Johanni (1986).

An appealing choice for the parametrization of paths is to use spline functions. These are general purpose piecewise cubic polynomials interpolating a sequence of points, called knots, with smoothness requirements and with a minimum curvature property (de Boor, 1978). In the robotic applications, they have been used to generate minimum-time trajectories under purely kinematic constraints (Lin et al., 1983). Relaxing the continuity of the second derivative (i.e. acceleration), splines are an appropriate tool for on-line trajectory generation (Chand and Doty, 1985). Rajan (1985) used them in an iterative procedure for approximating the point-to-point minimum-time path, under dynamic torque constraints. The same approach is followed by Lin and Chang (1985), but considering the class of splines with continuous acceleration and parametrizing the path in terms of the time intervals between knots. Their minimization algorithm, as well as the one used in (Lin et al., 1983), is the Nelder-Mead flexible polyhedron search; being based only on function evaluations, it is slow and may stop in false local constrained minima.

In this paper, a new algorithm is presented for the optimal timing along a smooth path in the joint space. The work is an outgrowth of (De Luca and Nicolò, 1985). The trajectory is a C^2-spline passing through n knots in the joint space, with boundary conditions on initial and final velocity. The n-1 time intervals between the knots parametrize uniquely the path. A minimum time problem with both maximum joint velocity and torque limits will be considered.

For cubic splines, the set of instants where maximum velocity is achieved is finite and known (Lin et al., 1983). It is enough then to check the feasibility of the velocity in these instants. On the other hand, no specific pattern arises for the computed torques along the spline trajectory: the torque constraint has to be checked for satisfaction in *all* instants along the path. This turns to be a infinite dimensional constraint and the optimization problem has to be formulated as a semi-infinite programming one, with both conventional and functional constraints.

For its solution a slight modification of an efficient algorithm proposed by Gonzaga et al. (1980) will be used. This is a method of feasible directions in the parameter space, in which the functional constraints are treated by a proper discretization. The main advantage of this algorithm is that it belongs to the class of first order methods, i.e. based on gradient information, which are usually faster and safer than those using function evaluations only. Moreover, after a certain number of iterations the current point becomes feasible and remains such until optimality is reached. It is worth noting that many engineering design problems have been solved using this numerical method (Tits et al., 1986).

The algorithm requires the knowledge of the gradients of the constraints which are active at a given iteration. In other words, sensitivity functions of the constraints in terms of parameters' variations are needed. As an intermediate step, the sensitivity of the spline functions has to be computed. Explicit expressions of the sensitivities will be derived; these have their own interest and relevance also for other applications.

The paper is organized as follows. The construction of a spline trajectory will be recalled first, together with the recursive scheme for computing the spline accelerations at the knots. The minimum-time optimization is then formulated as a semi-infinite nonlinear programming problem, and the basic algorithm of Gonzaga et al. (1980) summarized. Next, a sensitivity analysis of the spline and the torque profiles with respect to variations of the path parameters will be presented. A discussion of the numerical results obtained for a two-link Scara robot arm and of the possible extensions of the method completes the paper.

Spline robot trajectories

Let the robotic task be specified in the N-dimensional joint space by a sequence of n joint position values, q_{j1}, q_{j2},..., q_{jn}, for j = 1,...,N, to be assumed at unspecified time instants t_1, t_2,..., t_n, with $t_1 = 0$. The desired initial and final joint velocities, v_{j1} and v_{jn}, are given. Let $h_i = t_{i+1} - t_i$, for i = 1,...,n-1, be the time intervals between the knots and $\mathbf{h} = [h_1, \dots, h_{n-1}]^T$.

For the generic joint j, a trajectory will be constructed by interpolating the data via a C^2 - spline $Q_j(t)$, a piecewise cubic polynomial with continuity up to the second derivative. Each of the n-1 cubics $Q_{ji}(t)$ constituting the spline can be written as (Lin et al., 1983):

$$Q_{ji}(t) = \frac{(t_{i+1}-t)^3}{6h_i}\omega_{ji} + \frac{(t-t_i)^3}{6h_i}\omega_{j,i+1} + \left[\frac{q_{j,i+1}}{h_i} - \frac{h_i\omega_{j,i+1}}{6}\right](t-t_i) + \left[\frac{q_{ji}}{h_i} - \frac{h_i\omega_{ji}}{6}\right](t_{i+1}-t)$$

for $t \in [t_i, t_{i+1}]$. ω_{ji} is the acceleration at the i-th knot of the j-th joint. The continuity of the second derivative follows from

$$\omega_{ji} = \ddot{Q}_{j,i-1}(t_i) = \ddot{Q}_{ji}(t_i), \qquad \text{for } i = 2,\dots,n-1.$$

The interpolating spline function is completely specified when a given timing (i.e. vector **h**) is assigned and the associated knots accelerations ω_{ji} are determined. Let Ω_j be the n-dimensional acceleration vector relative to the j-th joint. Ω_j is found by solving the following tridiagonal linear system of n equations, obtained imposing the two boundary conditions and the n-2 continuity conditions on the first derivative at the internal knots:

$$A\,\Omega_j = b_j \qquad (1)$$

where

$$A = \begin{bmatrix} 2h_1 & h_1 & & & \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & \dots & \dots & \dots & \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ & & & h_{n-1} & 2h_{n-1} \end{bmatrix}$$

$$b_j = \left[6\left(\frac{q_{j2}-q_{j1}}{h_1} - v_{j1}\right) \dots 6\left(\frac{q_{jk}-q_{j,k-1}}{h_{k-1}} - \frac{q_{j,k-1}-q_{j,k-2}}{h_{k-2}}\right) \dots 6\left(v_{jn} - \frac{q_{jn}-q_{j,n-1}}{h_{n-1}}\right)\right]^T.$$

It can be shown that, if $\mathbf{h} > 0$ (i.e. $h_i > 0$, i = 1,...,n-1), the solution is unique and continuous as a function of **h**. Note that the matrix A is the same for all joints. This system can be efficiently solved by a Gauss-type algorithm without pivoting (Stoer and Bulirsch, 1980). The components of the solution Ω_j can be found by forward and backward substitution as:

$$\omega_{jn} = \bar{\omega}_{jn}, \qquad \omega_{ji} = \bar{\omega}_{ji} - K_i\,\omega_{j,i+1}, \quad i = n-1,\dots,1$$

with

$$\bar{\omega}_{j1} = \frac{3}{h_1}\left(\frac{q_{j2}-q_{j1}}{h_1} - v_{j1}\right)$$

$$\bar{\omega}_{ji} = \frac{6\left(\frac{q_{j,i+1}-q_{ji}}{h_i} - \frac{q_{ji}-q_{j,i-1}}{h_{i-1}}\right) - h_{i-1}\bar{\omega}_{j,i-1}}{2(h_i+h_{i-1}) - h_{i-1}K_{i-1}}, \qquad i = 2,\dots,n-1$$

$$\bar{\omega}_{jn} = \frac{6\left(v_{jn} - \frac{q_{jn}-q_{j,n-1}}{h_{n-1}}\right) - h_{n-1}\bar{\omega}_{j,n-1}}{h_{n-1}(2-K_{n-1})}.$$

and

$$K_1 = 0.5, \qquad K_i = \frac{h_i}{2(h_i+h_{i-1}) - h_{i-1}K_{i-1}}, \quad i = 2,\dots,n-1.$$

These expressions will be used later for computing the spline sensitivity to variations in the vector parameter **h**.

As a result, a unique set of N joint spline trajectories $Q_j(t,\mathbf{h})$ is built for a given **h**. Let these splines be organized into a vector $\mathbf{Q}(t,\mathbf{h})$. It should be noted that by changing one or more of the discrete parameters h_i, the whole interpolating spline will be different unless a uniform scaling is performed on *all* the components of **h** (Hollerbach, 1984, Lin et al., 1983).

In order to evaluate the torques needed to travel along the spline trajectory, the dynamics of the robot arm will be used:

$$\mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{e}(\mathbf{q}) = \mathbf{u}$$

q are the generalized joint coordinates, **u** the joint torques, D is the inertia matrix, **c** are Coriolis and centrifugal terms (quadratic in $\dot{\mathbf{q}}$), **e** is the gravitational contribution. The torque u_j arising at joint j along a given spline trajectory $\mathbf{q} = \mathbf{Q}(t,\mathbf{h})$ is computed through the above inverse dynamics as

$$d_j^T[\mathbf{Q}(t,\mathbf{h})]\,\ddot{\mathbf{Q}}(t,\mathbf{h}) + \dot{\mathbf{Q}}(t,\mathbf{h})^T C_j[\mathbf{Q}(t,\mathbf{h})]\,\dot{\mathbf{Q}}(t,\mathbf{h}) + e_j[\mathbf{Q}(t,\mathbf{h})] = u_j(t,\mathbf{h})$$

where d_j is the j-th column of the symmetric matrix D.

Optimization problem formulation

Assume for simplicity that the given bounds on maximum joint velocity and torque are constant and symmetric:

$$|\dot{Q}_j(t,\mathbf{h})| \le V_j, \qquad |u_j(t,\mathbf{h})| \le U_j, \qquad j = 1,\dots,N.$$

Since the spline velocity is a piecewise quadratic function, its maximum value is attained either at the knots or where acceleration is zero. The first constraint is then satisfied for all instants t of interest if and only if it is satisfied in a prespecified finite set of instants. Velocity has to be checked only at the knot instants t_i, i = 1,...,n, and at intermediate instants $t_{ji}^* \in [t_i, t_{i+1}]$ such that $\ddot{Q}_j(t_{ji}^*,\mathbf{h}) = 0$ (Lin et al., 1983). These intermediate instants exists only when $\omega_{ji}\,\omega_{j,i+1} < 0$ (for joint j) and their expression is $t_{ji}^* = t_i + \omega_{ji} / (\omega_{ji} - \omega_{j,i+1})$. The same reasoning does not apply to the torques. In fact, the maximum may be attained everywhere along the spline trajectory. In principle, the torque constraint has to be evaluated at all path points.

As a result, the minimum time optimization problem can be

formulated in the following way:

$$\min t_n = \sum_{i=1}^{n-1} h_i$$

$$\text{s.t. } |\dot{Q}_j(t_i,\mathbf{h})| - V_j \le 0, \quad i = 1,\dots,n,$$

$$|\dot{Q}_j(t_{ji}^*,\mathbf{h})| - V_j \le 0, \quad i = 1,\dots,n-1,$$

$$\max_{t\in[0,t_n]} \{|u_j(t,\mathbf{h})| - U_j\} \le 0; \qquad \text{for } j = 1,\dots,N.$$

This form of the torque constraints is usually referred to as an infinite-dimensional or functional one. The resulting problem is a semi-infinite nonlinear programming one, with conventional and functional constraints. The expressions of the spline position, velocity and acceleration should be used above. In particular, the velocity at the instant t_{ji}^* is

$$\dot{Q}_{ji}(t_{ji}^*) = -\frac{h_i\,\omega_{ji}\,\omega_{j,i+1}}{2(\omega_{j,i+1}-\omega_{ji})} + \frac{q_{j,i+1}-q_{ji}}{h_i} - \frac{h_i}{6}(\omega_{j,i+1}-\omega_{ji})$$

In order to ensure the existence of a solution to the tridiagonal system (1) defining the interpolating spline, the constraint $\mathbf{h} > \mathbf{0}$ has to be added to the problem formulation. This noncompact constraint can be replaced by a lower bound on the parameter vector $\mathbf{h}$ derived from the velocity limits, namely

$$\mathbf{w} - \mathbf{h} \le \mathbf{0}, \qquad w_i = \max_{j=1,\dots,N}\{\frac{|q_{ji+1}-q_{ji}|}{V_j}\}, \qquad i = 1,\dots,n-1.$$

For efficiency reasons, this simple constraint will be treated separately from the other conventional ones.

A solution algorithm

The optimization problem formulated in the previous section is an instance of the class

$$\min f^0(\mathbf{h}),$$

$$\text{s.t. } g^j(\mathbf{h}) \le 0,\ j = \{1,\dots,p\} = \mathbf{p}, \quad f^j(\mathbf{h}) \le 0,\ j = \{1,\dots,m\} = \mathbf{m}$$

in which

$$f^j(\mathbf{h}) = \max_{\tau\in T} \phi^j(\tau,\mathbf{h})$$

where $\mathbf{h} \in \mathbb{R}^{(n-1)}$ is the vector of design parameters with respect to which we are optimizing, g^j are the conventional constraints, f^j are the functional ones, and T is a closed interval in $\mathbb{R}$. The scalar *continuous* variable τ spans the domain T over which the functional constraints have to be satisfied. The functions involved in the minimum time problem formulation satisfy the following assumptions: $f^0(\mathbf{h})$ and $g^j(\mathbf{h})$ are C^1 - functions; $\phi^j(\tau,\mathbf{h})$ are continuous in both arguments, while the gradient $\nabla_{\mathbf{h}}\phi^j(\tau,\mathbf{h})$ is continuous w.r.t. $\mathbf{h}$.

For the above class of semi-infinite programming problems, Gonzaga et al. (1980) developed an efficient algorithm of solution which is a combined phase I - phase II method of feasible directions with proven convergence properties. There is no need of a feasible starting point, since feasibility is achieved anyway after a finite number of iterations. In the process of recovering feasibility (phase I), the algorithm already considers the objective function that has to be minimized in the feasible region (phase II). To generate a search direction $\mathbf{d}$ in the space of the parameters $\mathbf{h}$, a small quadratic programming subproblem is solved at each iteration. In this QP, directional derivatives of the objective function and of the ε-active (i.e. active or almost) constraints are used. The implementable version of the algorithm requires a proper discretization of the infinite-type constraints: the continuous domain T has to be substituted by a set T_q (of cardinality q) of discrete "instants".

To briefly state the algorithm, some definitions are needed:

$$f_q^j(\mathbf{h}) = \max_{\tau\in T_q} \phi_q^j(\tau,\mathbf{h})$$

$$\psi_q(\mathbf{h}) = \max\{g^j(\mathbf{h}),\ j\in\mathbf{p};\ f_q^j(\mathbf{h}),\ j\in\mathbf{m}\}$$

Let $\psi_{q,0}(\mathbf{h}) = \max\{0,\psi_q(\mathbf{h})\} \ge 0$ be the value of the most violated constraint. The indices of the ε-most active constraints are then:

$$J_{q,\varepsilon}^f(\mathbf{h}) = \{\, j\in\mathbf{m} \mid f_q^j(\mathbf{h}) \ge \psi_{q,0}(\mathbf{h}) - \varepsilon \,\}$$

$$J_{q,\varepsilon}^g(\mathbf{h}) = \{\, j\in\mathbf{p} \mid g^j(\mathbf{h}) \ge \psi_{q,0}(\mathbf{h}) - \varepsilon \,\}$$

The set of τ's where the j-th torque constraint is ε-most active is:

$$T_{q,\varepsilon}^j(\mathbf{h}) = \{\, \tau\in T_q \mid \phi_q^j(\tau,\mathbf{h}) \ge \psi_{q,0}(\mathbf{h}) - \varepsilon \,\}$$

A suitable search direction $\mathbf{d}$ in the parameter space follows from the solution of the following QP problem:

$$\vartheta_{q,\varepsilon}(\mathbf{h}) = \min_{\mathbf{d}}\{\frac{1}{2}\|\mathbf{d}\|^2 + \max\{<\nabla f^0(\mathbf{h}),\mathbf{d}> - \gamma\psi_{q,0}(\mathbf{h});\ <\nabla g^j(\mathbf{h}),\mathbf{d}>, j\in J_{q,\varepsilon}^g;\ <\nabla_{\mathbf{h}}\phi^j(\tau,\mathbf{h}),\mathbf{d}>, \tau\in T_{q,\varepsilon}^j, j\in J_{q,\varepsilon}^f\}\}$$

The directional derivatives of the objective function, of the conventional and of the functional constraints should be available; the last ones have to be evaluated only at $\tau\in T_q$.

The algorithm consists of the following formal steps:

Algorithm (Gonzaga et al., 1980).

Given an initial $\mathbf{h}^0$, an integer q and coefficients $\gamma \ge 1$, $\varepsilon > 0$, and $\alpha \in (0,1)$, set i = 0.

Step 1 Compute the set of instants $T_{q,\varepsilon}^j(\mathbf{h}^i)$, $j\in\mathbf{m}$.

Step 2 Compute a direction $\mathbf{d}_{q,\varepsilon}(\mathbf{h}^i)$ solving the QP problem and let $\vartheta_{q,\varepsilon}(\mathbf{h}^i)$ be the optimal value.

Step 3 If $\vartheta_{q,\varepsilon}(\mathbf{h}^i) \le -\varepsilon$, go to 5; else, go to 4.

Step 4 If $\varepsilon \le 1/2^q$ and $\psi_{q,0}(\mathbf{h}^i) \le 1/2^q$, set q = q+1 and go to 1; else set $\varepsilon = \varepsilon/2$ and go to 1.

Step 5 Along the search direction $\mathbf{d}_{q,\varepsilon}(\mathbf{h}^i)$:

if $\psi_{q,0}(\mathbf{h}^i) = 0$ (*feasible region*), compute a step $\lambda^i > 0$ s.t.:

$$f^0(\mathbf{h}^i + \lambda^i\mathbf{d}_{q,\varepsilon}(\mathbf{h}^i)) - f^0(\mathbf{h}^i) \le -\alpha\,\varepsilon\,\lambda^i,$$

$$g^j(\mathbf{h}^i + \lambda^i\mathbf{d}_{q,\varepsilon}(\mathbf{h}^i)) \le 0, \quad j\in\mathbf{p},$$

$$f_q^j(\mathbf{h}^i + \lambda^i\mathbf{d}_{q,\varepsilon}(\mathbf{h}^i)) \le 0, \quad j\in\mathbf{m};$$

else if $\psi_{q,0}(\mathbf{h}^i) > 0$ (*unfeasible region*), compute $\lambda^i > 0$ s.t.:

$$\psi_q(\mathbf{h}^i + \lambda^i\mathbf{d}_{q,\varepsilon}(\mathbf{h}^i)) - \psi_q(\mathbf{h}^i) \le -\alpha\,\varepsilon\,\lambda^i.$$

Step 6 Set $\mathbf{h}^{i+1} = \mathbf{h}^i + \lambda^i\mathbf{d}_{q,\varepsilon}(\mathbf{h}^i)$, i = i +1, and go to 1.

In this algorithm, step 4 provides an internal loop that increases the discretization grid and/or modifies the ε-active rule. Step 5 implements an Armijo-type line search rule. An additional test is included at this point in order to satisfy the constraint $\mathbf{w} - \mathbf{h} \le \mathbf{0}$.

In general, the design vector parameter $\mathbf{h}$ and the continuous variable τ are not related each to other. However, in the above minimum time problem they both take on values in the time domain (i.e. $\tau = t$). Moreover, $T = [0, t_n]$ so that the continuous domain of interest is itself a *function* of the chosen $\mathbf{h}$. This requires an adaptive iteration-varying strategy for discretizing T into T_q, as opposed to an uniform one. At iteration k, the following set of mesh points is used:

$$T_q^{(k)} = \{t_{im}^{(k)} \mid t_{im}^{(k)} = t_i^{(k)} + \alpha_m h_i;\ m = 0,1,\dots,q;\ i = 1,\dots,n-1\}$$

where $\alpha_m = m/q$. In this way each subinterval $[t_i, t_{i+1}]$ is uniformly sampled and the knots t_k (where the acceleration is maximum) are always included in the current discretization.

This interdependence between t and $\mathbf{h}$ should be carefully considered also in the evaluation of $\nabla_{\mathbf{h}}\phi^j(t,\mathbf{h})$, the gradient with respect to $\mathbf{h}$ of the torque constraint. For determining the sensitivity of $\phi^j(t,\mathbf{h})$ to changes in $\mathbf{h}$, the values of the constraint before and after a small perturbation of $\mathbf{h}$ should be compared at the *same mesh point* in the discretization.

Sensitivity analysis

For solving the minimum time optimization problem using the previous algorithm the following directional derivatives are needed:

$$\langle \nabla f^0(\mathbf{h}), \mathbf{d} \rangle = \sum_{i=1}^{n-1} d_i$$

$$\langle \nabla g^j(\mathbf{h}), \mathbf{d} \rangle = \pm \sum_{k=1}^{n-1} \frac{\partial \dot{Q}_j(s,\mathbf{h})}{\partial h_k} d_k, \qquad \text{with } s = t_i \text{ or } t_{ji}^*$$

$$\langle \nabla_{\mathbf{h}} \phi^j(s,\mathbf{h}), \mathbf{d} \rangle = \pm \sum_{k=1}^{n-1} \frac{\partial u_j(s,\mathbf{h})}{\partial h_k} d_k, \qquad \text{with } s = t_{im}$$

The ± signs arise from the modulus in the constraints. Note that the quantities above are computed only at the knots t_i or at the intermediate instants t_{ji}^* or, for the functional constraints ϕ^j, at mesh points t_{im}. Using the inverse dynamics, the sensitivity of the robot torques takes the form:

$$\frac{\partial u_j}{\partial h_k} = \sum_{r=1}^{N} \{ \sum_{s=1}^{N} \frac{\partial d_{jr}}{\partial Q_s} \frac{\partial Q_s}{\partial h_k} \ddot{Q}_r + d_{jr} \frac{\partial \ddot{Q}_r}{\partial h_k} + \frac{\partial c_j}{\partial Q_r} \frac{\partial Q_r}{\partial h_k} + \frac{\partial c_j}{\partial \dot{Q}_r} \frac{\partial \dot{Q}_r}{\partial h_k} + \frac{\partial e_j}{\partial Q_r} \frac{\partial Q_r}{\partial h_k} \}$$

As a result, the following quantities are needed:

a) the sensitivity of the dynamic terms in the robot model with respect to variations of the joint coordinates **q** and **q'**, evaluated along the spline (i.e. for $q_j = Q_j$, $q_j' = Q_j'$); this depends on the specific robot arm, but programs are available which compute the sensitivity of the dynamic model in a closed symbolic form (Neuman and Murray, 1984);

b) the sensitivity of the splines Q_j, and of their first and second derivatives Q_j' and Q_j'', with respect to changes of the time intervals h_k; no results seem to have been reported previously on this aspect.

In the following analysis, a preliminary result is needed concerning the sensitivity of the solution to the tridiagonal system (1) (i.e. of the knots accelerations ω_{ji}) w.r.t. variations of **h**, which appears in the system coefficients and in the right-hand-side term. This sensitivity will be denoted by

$$\omega_{ji}^{(k)} = \frac{\partial \omega_{ji}}{\partial h_k}, \qquad k = 1,\ldots,n-1, \quad i = 1,\ldots,n, \quad j = 1,\ldots,N,$$

and the actual expressions are reported in the Appendix.

The spline sensitivity has to be computed at a generic mesh point $t_{im} \in [t_i, t_{i+1}]$ and at the instants t_{ji}^* where maximum velocity may occur. Dropping the joint index j in the i-th cubic Q_{ji}, this can be rewritten at t_{im} as:

$$Q_i(t_{im}) = \frac{h_i^2}{6} [(1-\alpha_m)^3 \omega_i + \alpha_m^3 \omega_{i+1} - \alpha_m \omega_{i+1} - (1-\alpha_m)\omega_i] + \alpha_m q_{i+1} + (1-\alpha_m) q_i.$$

The sensitivity of the spline position at a mesh point is then:

$$\frac{\partial Q_i(t_{im})}{\partial h_k} = \frac{h_i^2}{6} [(1-\alpha_m)^3 \omega_i^{(k)} + \alpha_m^3 \omega_{i+1}^{(k)} - \alpha_m \omega_{i+1}^{(k)} - (1-\alpha_m)\omega_i^{(k)}] +$$

$$+ \delta_{ik} [\frac{h_i}{3} [(1-\alpha_m)^3 \omega_i + \alpha_m^3 \omega_{i+1} - \alpha_m \omega_{i+1} - (1-\alpha_m)\omega_i]].$$

for i = 1,...,n-1, m = 0,1,...,q, k = 1,...,n-1. δ_{ik} is the Kronecker delta. Note that for $\alpha_0 = 0$ and $\alpha_q = 1$, the sensitivity is zero. These values correspond to the instants at the knots, where the spline trajectory should pass in any case (i.e. for all **h**).

In a similar way, the spline velocity becomes at t_{im}:

$$\dot{Q}_i(t_{im}) = \frac{h_i}{2} [\alpha_m^2 \omega_{i+1} - (1-\alpha_m)^2 \omega_i] + \frac{q_{i+1} - q_i}{h_i^2} - \frac{h_i(\omega_{i+1} - \omega_i)}{6}$$

which gives

$$\frac{\partial \dot{Q}_i(t_{im})}{\partial h_k} = \frac{h_i}{2} [\alpha_m^2 \omega_{i+1}^{(k)} - (1-\alpha_m)^2 \omega_i^{(k)}] - \frac{h_i(\omega_{i+1}^{(k)} - \omega_i^{(k)})}{6} +$$

$$+ \delta_{ik} [\frac{\alpha_m^2 \omega_{i+1} - (1-\alpha_m)^2 \omega_i}{2} - \frac{q_{i+1} - q_i}{h_i^2} - \frac{\omega_{i+1} - \omega_i}{6}]$$

Again, this expression goes to zero at the initial and final instants t_1 and t_n. For the spline velocity at the instant t_{ji}^*, where acceleration may go to zero:

$$\dot{Q}_i(t_{ji}^*) = -\frac{h_i \omega_i \omega_{i+1}}{2(\omega_{i+1} - \omega_i)} + \frac{q_{i+1} - q_i}{h_i} - \frac{h_i(\omega_{i+1} - \omega_i)}{6}$$

The sensitivity is:

$$\frac{\partial \dot{Q}_i(t_{ji}^*)}{\partial h_k} = -\frac{h_i(\omega_{i+1}^{(k)} - \omega_i^{(k)})}{6} - \frac{h_i(\omega_{i+1}^2 \omega_i^{(k)} - \omega_i^2 \omega_{i+1}^{(k)})}{2(\omega_{i+1} - \omega_i)^2} +$$

$$- \delta_{ik} [\frac{\omega_{i+1} - \omega_i}{6} + \frac{q_{i+1} - q_i}{h_i^2} + \frac{\omega_i \omega_{i+1}}{\omega_{i+1} - \omega_i}].$$

Finally, for the spline acceleration

$$\ddot{Q}_i = \alpha_m \omega_{i+1} + (1-\alpha_m)\omega_i$$

the sensitivity is:

$$\frac{\partial \ddot{Q}_i(t_{im})}{\partial h_k} = \alpha_m \omega_{i+1}^{(k)} + (1-\alpha_m)\omega_i^{(k)}$$

for i = 1,...,n-1, m = 0,...,q, k = 1,...,n-1. It can be shown that all the above sensitivity expressions satisfy the smoothness assumptions in the region $\mathbf{h} \geq \mathbf{w} > \mathbf{0}$.

Numerical simulations

The optimization method has been tested on a two-link planar SCARA-type robot arm. The values of the robot parameters are: $m_1 = m_2 = 1$ Kg, $l_1 = l_2 = 0.5$ m, $J_1 = J_2 = 0.0214$ Kg m^2. The limit values are $V_1 = V_2 = 2$ rad/sec for the velocity at the joints, and $U_1 = 7$ Nm, $U_2 = 2$ Nm for the torques. Details on the dynamic model and on the sensitivity computations can be found in (De Luca et al., 1988). The arm is required to move in minimum time along a smooth trajectory (with continuous acceleration) which interpolates a sequence of six position knots in the joint space:

$$q_{11} = 0 \quad q_{12} = .5 \quad q_{13} = .75 \quad q_{14} = 1 \quad q_{15} = 1.25 \quad q_{16} = 1.5$$

$$q_{21} = 0 \quad q_{22} = -.5 \quad q_{23} = -1 \quad q_{24} = -1.5 \quad q_{25} = -1 \quad q_{26} = .5$$

(in rads). Initial and final velocity are set to zero ($\mathbf{v}_1 = \mathbf{v}_6 = 0$). A program which implements the optimal spline approach has been written in Fortran 77 and runs on a IBM-AT personal computer. At each iteration, the routine E04NAF of the NAG Workstation Library is used for solving the small quadratic programming subproblem that defines the search direction.

In correspondence to the chosen initial design parameters

$$\mathbf{h}^T = [1 \quad .5 \quad .5 \quad .5 \quad .5], \qquad t_n = 3 \text{ sec}$$

the plots of spline positions and velocities, and of the computed torques along this trajectory are shown in Figures 1-3. Velocity and torque at the second joint are both unfeasible, reaching the values of 4 rad/sec and -3 Nm, respectively. After 9 iterations the algorithm recovers full feasibility with

$$\mathbf{h}^T = [.62 \quad .37 \quad .37 \quad .37 \quad 1.97], \qquad t_n = 3.7 \text{ sec}$$

Note that, if the feasibility recovery rule of (Lin and Chang, 1985) were applied to the initial **h**, a higher total traveling time, $t_n = 6$ sec, would have been found. Figures 4-6 refer to the final solution obtained after 40 iterations and a computing time of approximately 4 minutes. The optimal design parameter vector and the total time are:

$$\mathbf{h}^T = [.37 \quad .25 \quad .34 \quad .43 \quad 1.07], \qquad t_n = 2.46 \text{ sec}$$

The plots shows that both torques are saturated at the initial time instant, while the velocity of the second joint saturates twice, around the second and the fourth trajectory knots. The same final solution was obtained by letting the algorithm start from a feasible initial value of **h**.

It is known that the minimum-time torque profile along a parametrized trajectory is such that at least one actuator gives maximum torque at each instant (Bobrow et al., 1985). The reason for the absence of saturated flat tops in the obtained profiles is two-fold. First, the presence of velocity bounds may prevent the torques from reaching their maximum values. Second, the requirement of a continuous acceleration excludes a bang-bang form for the torques.

Conclusions

A new method for minimizing the total traveling time along a spline trajectory under kinematic and dynamic constraints has been presented. Inclusion of both velocity and torque bounds, and the continuity hypothesis on the trajectory acceleration, give rise to new interesting optimal time profiles.

The numerical algorithm used for solution is very robust and efficient, being based on gradient information. The sensitivity of the spline behavior with respect to changes of the time intervals between the knots was computed. The derived expressions can be used also in purely kinematic approaches to robot trajectory planning, and may prove helpful in other applications too.

With little additional complexity, the method can be extended to the optimization of more general objectives under different types of constraints, like velocity dependent bounds on the torques or jerk limits. Finally, nondifferentiable functions may be included following the theoretical developments of the same basic algorithm, as presented in (Polak et al., 1983).

References

Ailon, A. and G. Langholz (1985). On the existence of time-optimal control of mechanical manipulators, *J. Optimization Theory and Appl.*, **46**, 1, 1-21.

Bobrow, J.E., S. Dubowsky, and J.S. Gibson (1985). Time-optimal control of robotic manipulators along specified paths, *Int. J. Robotics Research*, **4**, 3, 3-17.

Brady, M. (1982). Trajectory Planning, in: *Robot Motion: Planning and Control* (M.Brady et al., Eds.), MIT Press, Cambridge, 221-243.

Chand, S. and K.L. Doty (1985). On-line polynomial trajectories for robot manipulators, *Int. J. Robotics Research*, **4**, 2, 38-48.

de Boor, C. (1978). *A Practical Guide to Splines*, Appl. Math. Sciences Vol.27, Springer- Verlag, Berlin.

De Luca, A., L. Lanari, and G. Oriolo (1988). Generation and computation of optimal smooth trajectories for robot arms, DIS Rap.11.88, Università di Roma "La Sapienza".

De Luca, A. and F. Nicolò (1985). Minimum traveling time for robot arm under joints dynamic constraints, *Preprints 1st IFAC Symp. Robot Control (SYROCO'85)*, Barcelona, 361-363.

Gawronski, R., I. Pardyka, A. Pawlowski, and W. Wolski (1981). Time-optimal hierarchical control system of manipulator movement with adaptive on-off control, *Systems Science*, **7**, 2, 167-177.

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

Lin, C.S. and P.R. Chang (1985). Approximate optimum paths of robot manipulators under realistic physical constraints, *2nd IEEE Int. Conf. Robotics and Automation*, St.Louis, 737-742.

Lin, C.S., P.R. Chang, and J.Y.S. Luh (1983). Formulation and optimization of cubic polynomial joint trajectories for industrial robots, *IEEE Trans. Automatic Control*, **AC-28**, 12, 1067-1073.

Gonzaga, C., E. Polak, and R. Trahan (1980). An improved algorithm for optimization problems with functional inequality constraints, *IEEE Trans. Automatic Control*, **AC-25**, 1, 49-54.

Hollerbach, J.M. (1984). Dynamic scaling of manipulator trajectories, *ASME J.Dynamic Syst., Meas., and Control*, **106**, 102-106.

Neuman, C.P. and J.J. Murray (1984). Linearization and sensitivity functions of dynamic robot models, *IEEE Trans. Systems, Man, and Cybernetics*, **SMC-14**, 6, 805-818.

Pfeiffer, F. and R. Johanni (1986). A concept for manipulator trajectory planning, *IEEE J. Robotics and Automation*, **RA-3**, 2, 115-123.

Polak, E., D.Q. Mayne, and Y. Wardi (1983). On the extension of constrained optimization algorithms from differentiable to nondifferentiable problems, *SIAM J. Control and Optimization*, **21**, 2, 179-203.

Rajan, V.T. (1985). Minimum time trajectory planning, *2nd IEEE Int. Conf. on Robotics and Automation*, St.Louis, 759-764.

Sahar, G.S. and J.M. Hollerbach (1986) Planning of minimum-time trajectories for robot arms, *Int. J. Robotics Research*, **5**, 3, 90-100.

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

Stoer, J. and R. Bulirsch (1980). *Introduction to Numerical Analysis*, Springer Verlag, Berlin.

Tits, A.L., W.T. Nye, and A.L. Sangiovanni-Vincentelli (1986). Enhanced methods of feasible directions for engineering design problems,*J. Optimization Theory and Appl.*,**51**,3,475-504.

Weinreb, A. and A.E. Bryson (1985). Optimal control of systems with hard control bounds, *IEEE Trans. Automatic Control*, **AC-30**, 11, 1135-1138.

Appendix

The sensitivity of the spline accelerations ω_{ji} at the knots is computed here. For i = 1,...,n, ω_{ji} are the components of the solution to the tridiagonal linear system (1). The recursive form for obtaining such a solution will be used in the following. For compactness, the index j relative to the considered joint is dropped. The sensitivity functions

$$\omega_i^{(k)} := \frac{\partial \omega_i}{\partial h_k} \qquad \text{for } i = 1,\ldots,n;\ k = 1,\ldots,n-1$$

are iteratively derived as:

$$\omega_n^{(k)} = \bar{\omega}_n^{(k)} \qquad k = 1,\ldots,n-1;$$

$$\omega_i^{(k)} = \bar{\omega}_i^{(k)} - K_i \omega_{i+1}^{(k)} - \frac{\partial K_i}{\partial h_k} \omega_{i+1}, \quad i = n-1,\ldots,1;\ k = 1,\ldots,n-1.$$

The overbarred expressions are obtained conversely from:

$$\bar{\omega}_1^{(1)} = \frac{3v_1}{h_1^2} - \frac{6(q_2 - q_1)}{h_1^3}, \qquad \bar{\omega}_1^{(k)} = 0, \quad k = 2,\ldots,n-1;$$

$$\bar{\omega}_i^{(k)} = \frac{h_{i-1}\left[\frac{\partial K_{i-1}}{\partial h_k}\bar{\omega}_i - \bar{\omega}_{i-1}^{(k)}\right]}{2(h_i + h_{i-1}) - h_{i-1}K_{i-1}}, \qquad k = 1,\ldots,i-2,$$

$$\bar{\omega}_i^{(i-1)} = \frac{\frac{6(q_i - q_{i-1})}{h_{i-1}^2} - \bar{\omega}_{i-1} - h_{i-1}\omega_{i-1}^{(i-1)} + \bar{\omega}_i\left(K_{i-1} + h_{i-1}\frac{\partial K_{i-1}}{\partial h_{i-1}} - 2\right)}{2(h_i + h_{i-1}) - h_{i-1}K_{i-1}},$$

$$\bar{\omega}_i^{(i)} = \frac{\frac{6(q_i - q_{i+1})}{h_i^2} - 2\bar{\omega}_i}{2(h_i + h_{i-1}) - h_{i-1}K_{i-1}}, \qquad \bar{\omega}_i^{(k)} = 0, \quad k = i+1,\ldots,n-1;$$

(for i = 2,...,n-1)

$$\bar{\omega}_n^{(k)} = \frac{\frac{\partial K_{n-1}}{\partial h_k}\bar{\omega}_n + \bar{\omega}_{n-1}^{(k)}}{2 - K_{n-1}}, \qquad k = 1,\ldots,n-2,$$

$$\bar{\omega}_n^{(n-1)} = \frac{\dfrac{6(q_n - q_{n-1})}{h_{n-1}^2} - \bar{\omega}_{n-1} - h_{n-1}\bar{\omega}_{n-1}^{(n-1)} - \bar{\omega}_n\left(2 - K_{n-1} - h_{n-1}\dfrac{\partial K_{n-1}}{\partial h_{n-1}}\right)}{h_{n-1}(2 - K_{n-1})}.$$

To complete the computation, the terms $\partial K_i/\partial h_k$ are needed:

$$\frac{\partial K_1}{\partial h_k} = 0, \qquad k = 1,\ldots,n-1;$$

$$\frac{\partial K_i}{\partial h_k} = \frac{h_i h_{i-1}\dfrac{\partial K_{i-1}}{\partial h_k}}{[2(h_i+h_{i-1}) - h_{i-1}K_{i-1}]^2}, \qquad k = 1,\ldots,i-2,$$

$$\frac{\partial K_i}{\partial h_{i-1}} = \frac{h_i\left(K_{i-1}+h_{i-1}\dfrac{\partial K_{i-1}}{\partial h_{i-1}} - 2\right)}{[2(h_i+h_{i-1}) - h_{i-1}K_{i-1}]^2}, \qquad \frac{\partial K_i}{\partial h_i} = \frac{h_{i-1}(2 - K_{i-1})}{[2(h_i+h_{i-1}) - h_{i-1}K_{i-1}]^2},$$

$$\frac{\partial K_i}{\partial h_k} = 0, \qquad k = i+1,\ldots,n-1; \quad \text{for } i = 2,\ldots,n-1.$$

Use of these equations in the proper order gives the required sensitivities $\omega_i^{(k)}$.

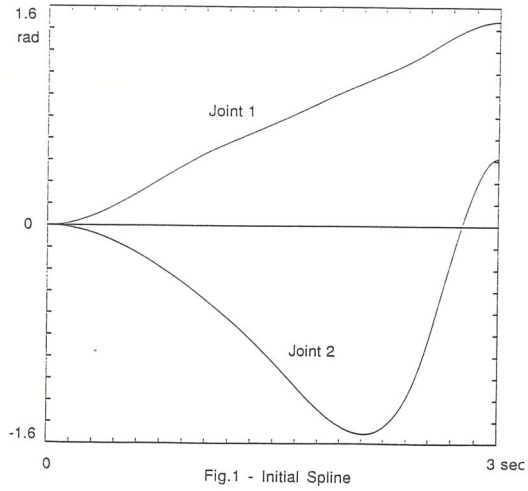

Fig.1 - Initial Spline

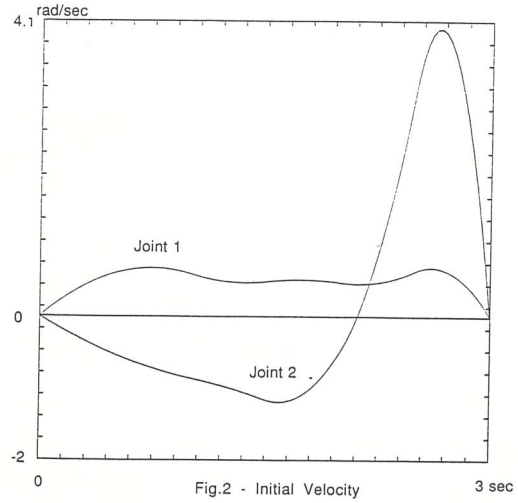

Fig.2 - Initial Velocity

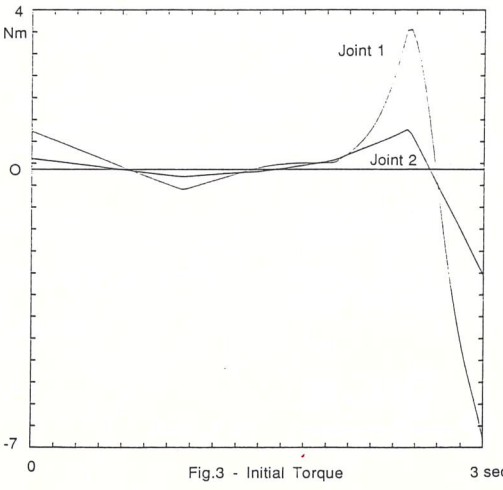

Fig.3 - Initial Torque

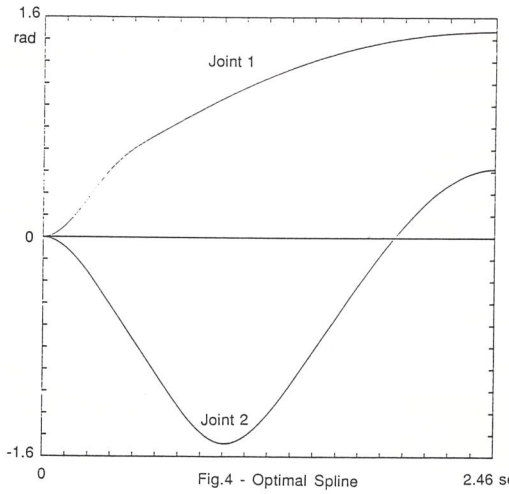

Fig.4 - Optimal Spline

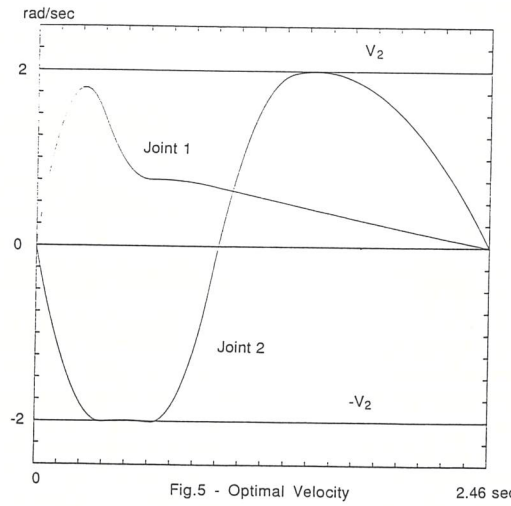

Fig.5 - Optimal Velocity

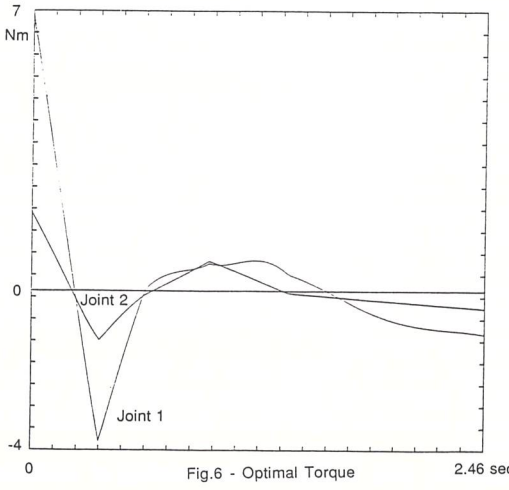

Fig.6 - Optimal Torque