INT. J. CONTROL, 2002, VOL. 75, NO. 13, 967–980

Minimum-time trajectory planning of mechanical manipulators under dynamic constraints

CORRADO GUARINO LO BIANCO† and AURELIO PIAZZI†*

The paper presents a global optimization approach to the trajectory planning problem of mechanical manipulators. The purpose is to obtain a minimum-time cubic spline trajectory subject to constraints given by limited joint torques and torque derivatives taking into account the non-linear manipulator dynamics. It is shown how, without conservativeness, a semi-infinite optimization problem emerges. Conditions ensuring that the formalized problem admits a solution are given. The estimated global solution can be actually obtained by means of an hybrid genetic/interval algorithm that guarantees the feasibility of the found solution. The methodology is illustrated with numerical details for a two-link planar arm and a PUMA six-link manipulator; for the former, comparisons with an alternative optimization solver are exposed.

1. Introduction

Optimal trajectory planning for non-redundant robotic manipulators is a problem dealt with several different approaches depending on the desired control goal. For example in many applications the manipulator has to run along a predefined path (laser cutting, arc welding, etc.). This is the case proposed by Žlajpah (1996) where a minimum-time movement over an assigned path was planned by considering both dynamic and kinematic requirements. This author uses a parameter phase-plane method originally presented by Bobrow *et al.* (1985) and Shin and McKay (1985).

In another common situation the manipulator is only required to cross a given number of via points: the robot motion, whose corresponding geometric path is not imposed *a priori*, is planned by minimizing an assigned performance index subject to appropriate constraints. Also in this case the total travelling time is the usual cost index. A classical approach to the optimal manipulator planning is the one proposed by Lin *et al.* (1983) where kinematic requirements (i.e. limited joint velocities, accelerations, and jerks) are converted into constraints for the resulting minimum-time optimization problem. A key idea of this approach is the use of cubic polynomials to optimally interpolate, in the joint space, the given sequence of knots. In more recent approaches, dynamic constraints have also been introduced. For example De Luca *et al.* (1991) and Guarino Lo Bianco and Piazzi (1999, 2001 b), still using the cubic spline scheme of Lin *et al.* (1983), deal with similar problems where a minimum-time trajectory is planned including in the constraints the bounds on the joint torques. In all the cases a semi-infinite optimization problem has resulted. For the former case it is solved with an algorithm for local optimization whereas for the latter a global optimization technique is applied. The necessity of limiting the torques arises from the physical limits of the joint actuators and cannot be neglected in actual industrial applications.

It is also known that sudden torque variations (i.e. large torque derivatives) are difficult to track for the actuator control system and introduce undesired mechanical solicitations in the manipulator. Hence the unmodelled dynamics of the manipulator are excited and this complicates the task of the joint controllers and increases the vibrations and noise produced. A partial indirect solution to this problem is obtained by limiting the joint jerk. Indeed the amplitude of the torque derivative mainly depends on the jerk. For this reason Lin *et al.* (1983) and Piazzi and Visioli (1998) proposed to constrain the jerk in their approaches. The common denominator of the two techniques is again given by the cubic spline scheme adopted to parameterize the trajectories but they differ for the chosen algorithmic method. For the latter work, in fact, an interval algorithm guarantees that the global minimum-time trajectory is determined within arbitrarily predefined precision.

This paper extends our previous work (Guarino Lo Bianco and Piazzi, 1999) pursuing a direct approach to limit the torque variations. The performance index is the total travelling time and explicit constraints on the torque and its derivative are taken into account with the non-linear manipulator dynamics. The resulting problem is shown to be equivalent, without conservativeness, to a non-linear semi-infinite optimization problem. This can be solved providing an estimated global minimizer by means of a genetic/interval algorithm (Guarino Lo Bianco and Piazzi, 2001 a, b). A distinguished feature of this solution and the novelty over other approaches (for example, local techniques such as sequential quadratic programming, etc.) is the guaranteed feasibility of the found solution (due to the deterministic interval part

Finally accepted April 2002.
* Author for correspondence. e-mail: aurelio@ce.unipr.it
† Dipartimento di Ingegneria dell'Informazione, Universitá di Parma, Parco Area delle Scienze, 181/A, I-43100 Parma, Italy.

International Journal of Control ISSN 0020–7179 print/ISSN 1366–5820 online © 2002 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170210156161

of the algorithm) and its estimated global quality (due to the stochastic genetic part of the algorithm; cf. § 3).

In § 2 the manipulator minimum-time planning problem is presented. An alternative formulation to the joint cubic spline scheme of Lin *et al.* (1983) is exposed with admissibility proof (Property 1 and Corollary 1). The deduced closed-form expression of the torque derivatives in (14) leads to the formal semi-infinite problem (15)–(17). Conditions for the existence of a feasible solution to problem (15)–(17) are given and demonstrated in § 3. The same section shows how to convert it into an unconstrained minimization problem that can be solved by the hybrid algorithm previously devised by the authors. Section 4 applies the proposed approach to plan minimum-time movements for a two-link planar arm and a PUMA manipulator with six degrees of freedom. For the former case, comparisons with an alternative optimization solver (the Matlab optimizer) are exposed and the advantages of the pursued approach are highlighted. The benefits of introducing the constraint on the torque derivatives are evidenced in the trajectory planning of the PUMA manipulator. Concluding remarks are reported in § 5.

1.1. *Notation*

The set of positive reals is denoted by $\mathbb{R}^+$. Vectors are indicated by means of lower case bold characters (e.g. $\mathbf{q}$) while matrices are indicated by capital bold characters (e.g. $\mathbf{M}$). Norms of vectors and matrices are denoted with $\|\cdot\|$. The absolute value of a vector of n elements is defined as $|\mathbf{q}| := [|q_1| \; |q_2| \; \cdots \; |q_n|]^{\mathrm{T}}$, while the writing $|\mathbf{q}| < |\mathbf{g}|$ indicates that $|q_i| < |g_i|$, $i = 1, 2, \ldots, n$. An homogeneous polynomial is denoted by a capital letter as in $P^{(s)}(\mathbf{h})$ where s designates its total degree.

2. The cubic spline trajectory planning

We want to plan a minimum-time movement for an m link manipulator. The manipulator tool frame has to cross s assigned via points in the cartesian work space. By solving the inverse kinematics these points can be converted into equivalent via points of the joint-space work envelope $\mathcal{Q}$. Let us indicate by $\mathbf{q} := [q_1 \; q_2 \; \cdots \; q_m]^{\mathrm{T}} \in \mathcal{Q} \subset \mathbb{R}^m$ the joint variable vector and let the manipulator movement be parametrized by cubic splines. For this class of interpolating functions, Lin *et al.* (1983) showed that two more free joint displacements have to been added to the assigned via points in order to guarantee continuity for velocities and accelerations. Let us represent the resulting $s+2$ joint via points by the data vectors ($n := s+1$)

$$\mathbf{q}^i := [q_1^i \; q_2^i \; \cdots \; q_m^i]^{\mathrm{T}}, \qquad i = 0, 1, \ldots, n \qquad (1)$$

where $\mathbf{q}^1$ and $\mathbf{q}^{n-1}$ are the free displacements. In (1) the generic component q_k^i represents the displacement of the kth joint at the ith knot. Analogously, the vectors of the joint velocities and accelerations at the ith knot are indicated by $\mathbf{v}^i := [v_1^i \; v_2^i \; \cdots \; v_m^i]^{\mathrm{T}}$ and $\mathbf{a}^i := [a_1^i \; a_2^i \; \cdots \; a_m^i]^{\mathrm{T}}$ respectively. Let us assume that vectors $\mathbf{v}^0, \mathbf{v}^n, \mathbf{a}^0, \mathbf{a}^n$ are assigned, i.e. velocities and accelerations are given at the first and last knot.

The time required to move the manipulator from the $(i-1)$th via point to the ith one is denoted by h_i. The vector containing all these interval times is indicated by $\mathbf{h} := [h_1 \; h_2 \; \cdots \; h_n]^{\mathrm{T}} \in \mathcal{B} := [\nu, +\infty)^n$, where ν is a small positive number which is introduced in order to avoid possible degenerate situations. Thus, the total travelling time to complete the robot movement is given by the sum of all the components of $\mathbf{h}$.

The ith cubic polynomial function for the kth joint displacement is indicated by $p_k^i(t)$ with $t \in [0, h_i]$. Among all possible parameterization for $p_k^i(t)$, the one proposed by Craig (1989) is chosen because it guarantees the continuity of positions and velocities over the knots

$$p_k^i(t) := q_k^{i-1} + v_k^{i-1} t + \left[\frac{3}{h_i^2}(q_k^i - q_k^{i-1}) - \frac{1}{h_i}(v_k^i + 2v_k^{i-1})\right] t^2 + \left[-\frac{2}{h_i^3}(q_k^i - q_k^{i-1}) + \frac{1}{h_i^2}(v_k^i + v_k^{i-1})\right] t^3,$$
$$i = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, m, \quad t \in [0, h_i] \qquad (2)$$

The continuity of the acceleration is imposed for each joint through the relations ($k = 1, 2, \ldots, m$)

$$\left.\begin{aligned} \ddot{p}_k^1(0) &= a_k^0 \\ \ddot{p}_k^2(0) &= \ddot{p}_k^1(h_1) \\ &\vdots \\ \ddot{p}_k^n(0) &= \ddot{p}_k^{n-1}(h_{n-1}) \\ a_k^n &= \ddot{p}_k^n(h_n) \end{aligned}\right\} \qquad (3)$$

By performing the explicit second-order differentiation of the spline function $p_k^i(t)$ reported in (2) we obtain a linear system of $n+1$ independent equations ($k = 1, 2, \ldots, m$) in the unknowns $\{v_k^1, v_k^2, \ldots, v_k^{n-1}, q_k^1, q_k^{n-1}\}$. The first and the last equality of (3) permit expressing q_k^1 and q_k^{n-1} as functions of the sole unknown v_k^1 and v_k^{n-1} respectively, according to the expressions

$$q_k^1 = (6q_k^0 + 4h_1 v_k^0 + 2h_1 v_k^1 + a_k^0 h_1^2)/6 \qquad (4)$$

$$q_k^{n-1} = (6q_k^n - 4h_n v_k^n - 2h_n v_k^{n-1} + a_k^n h_n^2)/6 \qquad (5)$$

By replacing (4) and (5) into the remaining equations of system (3), we obtain a reduced linear system of order $n-1$ that can be compactly written as

$$\mathbf{A}(\mathbf{h})\mathbf{x} = \mathbf{b} \qquad (6)$$

where $\mathbf{x} := [v_k^1 \ v_k^2 \ \dots \ v_k^{n-1}]^{\mathrm{T}}$ is the reduced vector of the unknowns while $\mathbf{A}(\mathbf{h})$ and $\mathbf{b}$ exhibit the structures (see bottom of page)

Note that matrix $\mathbf{A}(\mathbf{h})$ is tridiagonal and only depends on vector $\mathbf{h}$. It is then possible to introduce and prove the following result.

Property 1: *System* (6) *admits an unique solution* $\mathbf{x}$ *for any* $\mathbf{h} \in \mathbb{R}^{+n}$.

Proof: The first and the last equations of system (6) permit writing two biunivocal relations respectively between v_k^1 and v_k^2 and between v_k^{n-1} and v_k^{n-2}

$$v_k^1 = [h_2^2(a_k^0 h_1 + 2v_k^0) + 6h_1(q_k^2 - q_k^0) - h_1^2(h_1 a_k^0 + 4v_k^0) - 2h_1 h_2 v_k^2]/[2(h_1 + h_2)^2] \quad (7)$$

$$v_k^{n-1} = [-h_{n-1}^2(a_k^n h_n - 2v_k^n) + 6h_n(q_k^n - q_k^{n-2}) + h_n^2(h_n a_k^n - 4v_k^n) - 2h_{n-1}h_n v_k^{n-2}]/[2(h_n + h_{n-1})^2] \quad (8)$$

This allows us to reduce the system order. Indeed, using expressions (7) and (8) we eliminate the unknowns v_k^1 and v_k^{n-1} from (6) and the equivalent linear system is obtained

$$\mathbf{A}'(\mathbf{h})\,\mathbf{x}' = \mathbf{b}' \quad (9)$$

where $\mathbf{x}' := [v_k^2 \ v_k^3 \ \dots \ v_k^{n-2}]^{\mathrm{T}}$ and (see bottom of next page)

According to the Hadamard criterion, a square matrix $\mathbf{L} = [l_{ij}]$ $(i, j = 1, 2, \dots, n)$ is non-singular if

$$|l_{ii}| > \sum_{j \neq i} |l_{ij}| \qquad (i = 1, 2, \dots, n) \quad (10)$$

For system (9) the following inequalities hold for $i = 2, 3, \dots, n-4$

$$4(h_{i+1}h_{i+2}^2 + h_{i+1}^2 h_{i+2}) > 2h_{i+1}^2 h_{i+2} + 2h_{i+1}h_{i+2}^2 \qquad \forall h_{i+1}, h_{i+2} \in \mathbb{R}^+$$

Moreover, for the first and the last rows of $\mathbf{A}'(\mathbf{h})$ we have respectively

$$4h_3(h_3 + h_2)(h_1 + h_2) - 2h_3^2 h_1 > 2h_2 h_3(h_1 + h_2) \qquad \forall h_1, h_2, h_3 \in \mathbb{R}^+$$

$$4h_{n-2}(h_{n-1} + h_{n-2})(h_{n-1} + h_n) - 2h_{n-2}^2 h_n > 2h_{n-2}h_{n-1}(h_{n-1} + h_n) \qquad \forall h_{n-2}, h_{n-1}, h_n \in \mathbb{R}^+$$

so that we can conclude that matrix $\mathbf{A}'(\mathbf{h})$ satisfies the Hadamard conditions and system (9) has an unique solution $\mathbf{x}'$ for all $\mathbf{h} \in \mathbb{R}^{+n}$. Finally, by virtue of the two biunivocal relations (7) and (8), it is possible to conclude that system (6) also has an unique solution $\mathbf{x}$ for all $\mathbf{h} \in \mathbb{R}^{+n}$. □

The following corollary is an immediate consequence of Property 1 and of equations (4) and (5).

Corollary 1: *The linear system* (3) *admits a unique solution for any* $\mathbf{h} \in \mathbb{R}^{+n}$.

Matrix $\mathbf{A}'(\mathbf{h})$ is tridiagonal of order $n-3$. Thus, the solution of the associated linear system (9) can be found by means of known efficient algorithms based

$$\mathbf{A}(\mathbf{h}) = \begin{bmatrix} 2(h_1+h_2)^2 & 2h_1h_2 & 0 & & & & \\ 2h_3^2(h_1+h_2) & 4(h_2h_3^2+h_2^2h_3) & 2h_2^2h_3 & 0 & & & \\ 0 & \ddots & \ddots & \ddots & 0 & & \\ & 0 & 2h_ih_{i+1}^2 & 4(h_ih_{i+1}^2+h_i^2h_{i+1}) & 2h_i^2h_{i+1} & 0 & \\ & & 0 & \ddots & \ddots & \ddots & 0 \\ & & & 0 & 2h_{n-2}h_{n-1}^2 & 4(h_{n-2}h_{n-1}^2+h_{n-2}^2h_{n-1}) & 2h_{n-2}^2(h_{n-1}+h_n) \\ & & & & 0 & 2h_{n-1}h_n & 2(h_{n-1}+h_n)^2 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 6h_1(q_k^2 - q_k^0) + h_2^2(a_k^0h_1 + 2v_k^0) - h_1^2(h_1a_k^0 + 4v_k^0) \\ 6h_3^2(q_k^2 - q_k^0) + 6h_2^2(q_k^3 - q_k^2) - h_3^2h_1(h_1a_k^0 + 4v_k^0) \\ \vdots \\ 6h_{i+1}^2(q_k^i - q_k^{i-1}) + 6h_i^2(q_k^{i+1} - q_k^i) \\ \vdots \\ 6h_{n-1}^2(q_k^{n-2} - q_k^{n-3}) + 6h_{n-2}^2(q_k^n - q_k^{n-2}) + h_{n-2}^2h_n(a_k^nh_n - 4v_k^n) \\ 6h_n(q_k^n - q_k^{n-2}) - h_{n-1}^2(a_k^nh_n - 2v_k^n) + h_n^2(a_k^nh_n - 4v_k^n) \end{bmatrix}$$

on triangular decomposition (see, for example, Press *et al.* 1992, pp. 50–51). Once the solution $\mathbf{x}'$ has been evaluated, it is possible to determine the remaining unknowns of the original system (3) (i.e. $q_k^1, q_k^{n-1}, v_k^1, v_k^{n-1}$) by applying (7), (8), (4) and (5). For a discussion on the presented cubic spline trajectory planning see the equivalent derivation of Sciavicco and Siciliano (2000).

Obviously, the solution to system (3) depends on $\mathbf{h}$ so that we can write $v_k^i := v_k^i(\mathbf{h})$ and, by virtue of (2), we can indicate a generic spline as $p_k^i(t;\mathbf{h})$. Hence, the following spline vectorial function can be defined

$$\mathbf{p}^i(t;\mathbf{h}) := [p_1^i(t;\mathbf{h}) \quad p_2^i(t;\mathbf{h}) \quad \cdots \quad p_m^i(t;\mathbf{h})]^{\mathrm{T}} \qquad t \in [0, h_i], \qquad i = 1, 2, \ldots, n \quad (11)$$

When friction can be neglected joint torques can be evaluated by means of the manipulator dynamic equation (Craig 1989, p. 206)

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q})[\dot{\mathbf{q}}\dot{\mathbf{q}}] + \mathbf{D}(\mathbf{q})[\dot{\mathbf{q}}^2] + \mathbf{g}(\mathbf{q}) \qquad (12)$$

where $\boldsymbol{\tau} \in \mathbb{R}^m$ is the vector of generalized torques, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{m\times m}$ is the inertia matrix, $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{m\times m(m-1)/2}$ is a matrix of Coriolis terms, $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{m\times m}$ is a matrix of centrifugal coefficients, $\mathbf{g}(\mathbf{q})$ is the vector of the gravity terms, and $[\dot{\mathbf{q}}\dot{\mathbf{q}}] \in \mathbb{R}^{m(m-1)/2}$ and $[\dot{\mathbf{q}}^2] \in \mathbb{R}^m$ are vectors composed with velocities according to the definitions

$$[\dot{\mathbf{q}}\dot{\mathbf{q}}] := [\dot{q}_1\dot{q}_2 \quad \dot{q}_1\dot{q}_3 \quad \cdots \quad \dot{q}_1\dot{q}_m \quad \dot{q}_2\dot{q}_3 \quad \cdots \quad \dot{q}_2\dot{q}_m \quad \cdots \quad \dot{q}_{(m-1)}\dot{q}_m]^{\mathrm{T}}$$

and

$$[\dot{\mathbf{q}}^2] := [(\dot{q}_1)^2 \quad (\dot{q}_2)^2 \quad \cdots \quad (\dot{q}_m)^2]^{\mathrm{T}}$$

Taking into account (11), the torque vector becomes a time function parameterized by $\mathbf{h}$, so that we define, congruently with (12)

$$\boldsymbol{\tau}^i(t;\mathbf{h}) := \mathbf{M}(\mathbf{p}^i(t;\mathbf{h}))\,\ddot{\mathbf{p}}^i(t;\mathbf{h}) + \mathbf{B}(\mathbf{p}^i(t;\mathbf{h}))[\dot{\mathbf{p}}^i(t;\mathbf{h})\dot{\mathbf{p}}^i(t;\mathbf{h})] + \mathbf{D}(\mathbf{p}^i(t;\mathbf{h}))[\dot{\mathbf{p}}^i(t;\mathbf{h})^2] + \mathbf{g}(\mathbf{p}^i(t;\mathbf{h})) \qquad i = 1, \ldots, n, \qquad t \in [0, h_i] \quad (13)$$

By virtue of the dynamics equation (12) and using the derivation rule, reported in the Appendix, for matrices depending on a vectorial time function, the derivative of the torque vector can then be expressed as

$$\dot{\boldsymbol{\tau}} = \frac{\partial \mathbf{M}}{\partial \mathbf{q}}(I_m \otimes \dot{\mathbf{q}})\ddot{\mathbf{q}} + \mathbf{M}(\mathbf{q})\dddot{\mathbf{q}} + \frac{\partial \mathbf{B}}{\partial \mathbf{q}}(I_{m(m-1)/2} \otimes \dot{\mathbf{q}})[\dot{\mathbf{q}}\dot{\mathbf{q}}] + \mathbf{B}(\mathbf{q})[\ddot{\mathbf{q}}\dot{\mathbf{q}}] + \mathbf{B}(\mathbf{q})[\dot{\mathbf{q}}\ddot{\mathbf{q}}] + \frac{\partial \mathbf{D}}{\partial \mathbf{q}}(I_m \otimes \dot{\mathbf{q}})[\dot{\mathbf{q}}^2] + 2\mathbf{D}(\mathbf{q})\,\mathrm{diag}\,(\dot{\mathbf{q}})\ddot{\mathbf{q}} + \frac{\partial \mathbf{g}}{\partial \mathbf{q}}\dot{\mathbf{q}} \qquad (14)$$

where

$$[\ddot{\mathbf{q}}\dot{\mathbf{q}}] := [\ddot{q}_1\dot{q}_2 \quad \ddot{q}_1\dot{q}_3 \quad \cdots \quad \ddot{q}_1\dot{q}_m \quad \ddot{q}_2\dot{q}_3 \quad \cdots \quad \ddot{q}_2\dot{q}_m \quad \cdots \quad \ddot{q}_{(m-1)}\dot{q}_m]^{\mathrm{T}}$$

$$[\dot{\mathbf{q}}\ddot{\mathbf{q}}] := [\dot{q}_1\ddot{q}_2 \quad \dot{q}_1\ddot{q}_3 \quad \cdots \quad \dot{q}_1\ddot{q}_m \quad \dot{q}_2\ddot{q}_3 \quad \cdots \quad \dot{q}_2\ddot{q}_m \quad \cdots \quad \dot{q}_{(m-1)}\ddot{q}_m]^{\mathrm{T}}$$

Using the deduced equation (14) and analogously to the definition given in (13), the torque derivative vector, as it results from the planning (11), can be redefined as $\dot{\boldsymbol{\tau}}^i(t;\mathbf{h})$ $i = 1, \ldots, n$, $t \in [0, h_i]$. Hence, a minimum-time movement under torque and torque derivative constraints can be planned by solving the following semi-infinite optimization problem

$$\min_{\mathbf{h}\in\mathcal{B}} \sum_{i=1}^{n} h_i \qquad (15)$$

subject to ($i = 1, 2, \ldots, n$)

$$|\boldsymbol{\tau}^i(t;\mathbf{h})| \le \boldsymbol{\alpha} \qquad \forall t \in [0, h_i] \qquad (16)$$

$$\mathbf{A}'(\mathbf{h}) = \begin{bmatrix} 4h_3(h_3+h_2)(h_1+h_2) - 2h_3^2h_1 & 2h_2h_3(h_1+h_2) & 0 & & \\ \ddots & \ddots & \ddots & & \\ 0 & 2h_{i+1}h_{i+2}^2 & 4(h_{i+1}h_{i+2}^2 + h_{i+1}^2h_{i+2}) & 2h_{i+1}^2h_{i+2} & 0 \\ & \ddots & \ddots & \ddots & \ddots \\ & 0 & 2h_{n-2}h_{n-1}(h_{n-1}+h_n) & & 4h_{n-2}(h_{n-1}+h_{n-2})(h_{n-1}+h_n) - 2h_{n-2}^2h_n \end{bmatrix}$$

$$\mathbf{b}' = \begin{bmatrix} -h_3^2h_1(h_1+h_2)a_k^0 - 2h_3^2(2h_1+h_2)v_k^0 + 6h_3^2(q_k^2 - q_k^0) + 6h_2(h_1+h_2)(q_k^3 - q_k^2) \\ \vdots \\ 6h_{i+2}^2(q_k^{i+1} - q_k^i) + 6h_{i+1}^2(q_k^{i+2} - q_k^{i+1}) \\ \vdots \\ h_{n-2}^2h_n(h_n+h_{n-1})a_k^n - 2h_{n-2}^2(2h_n+h_{n-1})v_k^n + 6h_{n-2}^2(q_k^n - q_k^{n-2}) + 6h_{n-1}(h_n+h_{n-1})(q_k^{n-2} - q_k^{n-3}) \end{bmatrix}$$

$$|\dot{\boldsymbol{\tau}}^i(t;\mathbf{h})| \le \boldsymbol{\beta} \qquad \forall t \in [0, h_i] \tag{17}$$

where $\boldsymbol{\alpha} := [\alpha_1\ \alpha_2\ \ldots\ \alpha_m]^{\mathrm{T}} \in \mathbb{R}^{+m}$ is the vector of the assigned torque bounds while $\boldsymbol{\beta} := [\beta_1\ \beta_2\ \ldots\ \beta_m]^{\mathrm{T}} \in \mathbb{R}^{+m}$ represents the vector of the torque derivative bounds. The global minimizer of problem (15) be indicated by $\mathbf{h}^* \in \mathcal{B}$.

3. Solving the semi-infinite optimization problem

A feasibility result that is useful in addressing problem (15) is proposed with the following property.

Property 2: *Under the assumption of* $\mathbf{v}^0 = \mathbf{v}^n = 0$ *and* $\mathbf{a}^0 = \mathbf{a}^n = 0$ *problem* (15) *has a solution if*

$$\alpha_k > \max_{\mathbf{q}\in\mathcal{Q}} |g_k(\mathbf{q})| \quad \text{and} \quad \beta_k > 0, \qquad k = 1, 2, \ldots, m \tag{18}$$

Proof: Problem (15) admits a solution if there exists a feasible point $\mathbf{h}$ in the set $\mathcal{B}$, i.e. an admissible point $\mathbf{h}$ that satisfies the semi-infinite inequalities (16) and (17). Choose a point $\mathbf{h} \in \mathcal{B}$ defined as $\mathbf{h} := \boldsymbol{\lambda} = [\lambda\ \lambda\ \cdots\ \lambda]^{\mathrm{T}}$ where λ is a positive real parameter. In the following we will show that $\boldsymbol{\lambda}$ is feasible for a sufficient large λ.

By scrutiny of system (6) we have

$$v_k^i(\mathbf{h}) = \frac{P_{ki}^{(3n-6)}(\mathbf{h})}{\det[\mathbf{A}(\mathbf{h})]}, \qquad i = 1, \ldots, n-1, \qquad k = 1, \ldots, m \tag{19}$$

where $\det[\mathbf{A}(\mathbf{h})]$ is a homogeneous polynomial of order $3n-5$ and $P_{ki}^{(3n-6)}(\mathbf{h})$ is a suitable homogeneous polynomial. Hence, it follows that

$$\lim_{\lambda\to\infty} |v_k^i(\boldsymbol{\lambda})| = 0, \qquad i = 1, \ldots, n-1, \qquad k = 1, 2, \ldots, m \tag{20}$$

From (2), joint velocities, accelerations, and jerks are given by ($i = 1, \ldots, n$, $k = 1, 2, \ldots, m$, and $t \in [0, h_i]$)

$$\dot{p}_k^i(t;\mathbf{h}) := v_k^{i-1} + \left[\frac{6}{h_i^2}(q_k^i - q_k^{i-1}) - \frac{2}{h_i}(v_k^i + 2v_k^{i-1})\right]t + \left[-\frac{6}{h_i^3}(q_k^i - q_k^{i-1}) + \frac{3}{h_i^2}(v_k^i + v_k^{i-1})\right]t^2 \tag{21}$$

$$\ddot{p}_k^i(t;\mathbf{h}) := \left[\frac{6}{h_i^2}(q_k^i - q_k^{i-1}) - \frac{2}{h_i}(v_k^i + 2v_k^{i-1})\right] + \left[-\frac{12}{h_i^3}(q_k^i - q_k^{i-1}) + \frac{6}{h_i^2}(v_k^i + v_k^{i-1})\right]t \tag{22}$$

$$\dddot{p}_k^i(t;\mathbf{h}) := \left[-\frac{12}{h_i^3}(q_k^i - q_k^{i-1}) + \frac{6}{h_i^2}(v_k^i + v_k^{i-1})\right] \tag{23}$$

In the sequel, the joint jerk is simply denoted by $\dddot{p}_k^i(\mathbf{h})$ because it is independent from time t.

By virtue of (20) and above expressions (21), (22) and (23) for any given $\varepsilon > 0$ there exists $\lambda = \lambda(\varepsilon) > 0$ such that ($i = 1, \ldots, n$)

$$\|\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\| < \varepsilon \qquad \forall t \in [0, h_i] \tag{24}$$

$$\|\ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\| < \varepsilon \qquad \forall t \in [0, h_i] \tag{25}$$

$$\|\dddot{\mathbf{p}}^i(\boldsymbol{\lambda})\| < \varepsilon \tag{26}$$

Taking into account the boundedness, over $\mathcal{Q}$, of matrices $\mathbf{M}(\mathbf{q})$, $\mathbf{B}(\mathbf{q})$ and $\mathbf{D}(\mathbf{q})$ we have ($i = 1, \ldots, n$ and $t \in [0, h_i]$)

$$\begin{aligned}&\|\mathbf{M}(\mathbf{p}^i(t;\boldsymbol{\lambda}))\ \ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda}) + \mathbf{B}(\mathbf{p}^i(t;\boldsymbol{\lambda}))[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})] + \mathbf{D}(\mathbf{p}^i(t;\boldsymbol{\lambda}))[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})^2]\| \\ &\le \bar{M}\ \|\ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\| + \bar{B}\ \|[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})]\| + \bar{D}\ \|[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})^2]\|\end{aligned} \tag{27}$$

where $\bar{M}$, $\bar{B}$, and $\bar{D}$ are the appropriate real positive bounds for the matrices involved. From (24) and (25) we obtain

$$\begin{aligned}&\bar{M}\ \|\ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\| + \bar{B}\ \|[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})]\| + \bar{D}\ \|[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})^2]\| \\ &< \bar{M}\varepsilon + \bar{B}\,\frac{m(m-1)}{2}\,\varepsilon^2 + \bar{D}\,m\,\varepsilon^2 \\ &= \left(\bar{M} + \bar{B}\,\frac{m(m-1)}{2}\,\varepsilon + \bar{D}m\varepsilon\right)\varepsilon =: \gamma(\varepsilon)\,\varepsilon\end{aligned} \tag{28}$$

From (27) and (28) this inequality follows

$$\|\mathbf{M}(\mathbf{p}^i(t;\boldsymbol{\lambda}))\ \ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda}) + \mathbf{B}(\mathbf{p}^i(t;\boldsymbol{\lambda}))[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})] + \mathbf{D}(\mathbf{p}^i(t;\boldsymbol{\lambda}))[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})^2]\| < \gamma(\varepsilon)\,\varepsilon \tag{29}$$

By considering definition (13) we deduce ($i = 1, \ldots, n$ and $t \in [0, h_i]$)

$$|\boldsymbol{\tau}^i(t;\boldsymbol{\lambda})| < \varepsilon[\gamma(\varepsilon)\ \ \cdots\ \ \gamma(\varepsilon)]^{\mathrm{T}} + |\mathbf{g}(\mathbf{p}^i(t;\boldsymbol{\lambda}))| \tag{30}$$

We can write in turn

$$|\boldsymbol{\tau}^i(t;\boldsymbol{\lambda})| < \varepsilon\begin{bmatrix}\gamma(\varepsilon)\\ \vdots \\ \gamma(\varepsilon)\end{bmatrix} + \begin{bmatrix}\max_{\mathbf{q}\in\mathcal{Q}}|g_1(\mathbf{q})|\\ \vdots \\ \max_{\mathbf{q}\in\mathcal{Q}}|g_m(\mathbf{q})|\end{bmatrix} \tag{31}$$

The hypothesis (18) permits choosing ε such that

$$\gamma(\varepsilon)\varepsilon < \min_{k=1,\ldots,m}\left(\alpha_k - \max_{\mathbf{q}\in\mathcal{Q}}|g_k(\mathbf{q})|\right) \tag{32}$$

Therefore

$$\gamma(\varepsilon)\varepsilon + \max_{\mathbf{q}\in\mathcal{Q}}|g_k(\mathbf{q})| < \alpha_k \qquad k = 1, \ldots, m \tag{33}$$

and, from (31), we finally conclude

$$|\boldsymbol{\tau}^i(t;\boldsymbol{\lambda})| < \boldsymbol{\alpha} \qquad \forall t \in [0, h_i], \qquad i = 1, \ldots, n \tag{34}$$

The boundedness, over $\mathcal{Q}$, of the jacobians $\partial\mathbf{M}/\partial\mathbf{q}$, $\partial\mathbf{B}/\partial\mathbf{q}$, $\partial\mathbf{D}/\partial\mathbf{q}$, and $\partial\mathbf{g}/\partial\mathbf{q}$ implies the existence of real constants satisfying, for any $\mathbf{q} \in \mathcal{Q}$, the inequalities

$$\left\|\frac{\partial\mathbf{M}}{\partial\mathbf{q}}(\mathbf{q})\right\| \leq \widetilde{M}, \qquad \left\|\frac{\partial\mathbf{B}}{\partial\mathbf{q}}(\mathbf{q})\right\| \leq \widetilde{B}$$

$$\left\|\frac{\partial\mathbf{D}}{\partial\mathbf{q}}(\mathbf{q})\right\| \leq \widetilde{D}, \qquad \left\|\frac{\partial\mathbf{g}}{\partial\mathbf{q}}(\mathbf{q})\right\| \leq \widetilde{g}$$

Hence, considering equation (14) and inequalities (24), (25) and (26) we obtain the following chain of passages ($i = 1, \ldots, n$ and $t \in [0, h_i]$)

$$\begin{aligned}
\|\dot{\boldsymbol{\tau}}^i(t;\boldsymbol{\lambda})\| &\leq \widetilde{M}\|I_m \otimes \dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\|\|\ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\| + \bar{M}\|\dddot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\| \\
&+ \widetilde{B}\|I_{m(m-1)/2} \otimes \dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\|\|[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})]\| \\
&+ \bar{B}\|[\ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})]\| + \bar{B}\|[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda})]\| \\
&+ \widetilde{D}\|I_m \otimes \dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\|\|[\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})^2]\| \\
&+ 2\bar{D}\|\mathrm{diag}(\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda}))\|\|\ddot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\| + \widetilde{g}\|\dot{\mathbf{p}}^i(t;\boldsymbol{\lambda})\| \\
&< \widetilde{M}m^2\varepsilon\varepsilon + \bar{M}\varepsilon + \widetilde{B}\frac{m^2(m-1)}{2}\varepsilon\frac{m(m-1)}{2}\varepsilon^2 \\
&+ \bar{B}\frac{m(m-1)}{2}\varepsilon^2 + \bar{B}\frac{m(m-1)}{2}\varepsilon^2 \\
&+ \widetilde{D}m^2\varepsilon m\varepsilon^2 + 2\bar{D}m\varepsilon\varepsilon + \widetilde{g}\varepsilon =: \delta(\varepsilon)\varepsilon
\end{aligned}$$

It follows that

$$\|\dot{\boldsymbol{\tau}}^i(t;\boldsymbol{\lambda})\| < \delta(\varepsilon)\varepsilon \tag{35}$$

From the hypothesis (18) it is evident that we can choose ε such that

$$\delta(\varepsilon)\,\varepsilon < \min_{k=1,\ldots,m} \beta_k \tag{36}$$

Therefore, for any $t \in [0, h_i]$,

$$|\dot{\tau}_k^i(t;\boldsymbol{\lambda})| < \delta(\varepsilon)\,\varepsilon < \beta_k \qquad i = 1, \ldots, n, \qquad k = 1, \ldots, m$$

and finally

$$|\dot{\boldsymbol{\tau}}^i(t;\boldsymbol{\lambda})| < \boldsymbol{\beta} \qquad \forall t \in [0, h_i], \qquad i = 1, \ldots, n \tag{37}$$

Point $\boldsymbol{\lambda}$ is a feasible point of problem (15) if inequalities (34) and (37) hold simultaneously. This is secured by simply choosing a sufficient small ε that satisfies both (32) and (36). □

By using a penalty method the semi-infinite optimization problem (15) can be converted into the unconstrained problem

$$\min_{\mathbf{h}\in\mathcal{B}}\left\{\sum_{i=1}^{n} h_i + \sum_{i=1}^{n}\sum_{k=1}^{m}\Phi(\sigma_k^i(\mathbf{h})) + \sum_{i=1}^{n}\sum_{k=1}^{m}\Phi(\tilde{\sigma}_k^i(\mathbf{h}))\right\} \tag{38}$$

where ($i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, m$)

$$\sigma_k^i(\mathbf{h}) := \min_{t\in[0,h_i]}\{\alpha_k - |\tau_k^i(t;\mathbf{h})|\} \tag{39}$$

$$\tilde{\sigma}_k^i(\mathbf{h}) := \min_{t\in[0,h_i]}\{\beta_k - |\dot{\tau}_k^i(t;\mathbf{h})|\} \tag{40}$$

while the penalty function $\Phi(\sigma)$ is defined as

$$\Phi(\sigma) := \begin{cases} 0 & \text{if } \sigma \in [0, +\infty) \\ -(N/T)\sigma & \text{if } \sigma \in [-T, 0) \\ N & \text{if } \sigma \in (-\infty, -T) \end{cases} \tag{41}$$

When $T \to 0+$ and $N \to +\infty$ problem (38) is strictly equivalent to (15). For finite values of $N > 0$ and $T > 0$, monotonically better precisions are obtained for larger values of N and smaller values of T.

Problem (38) is finally solved by using the hybrid algorithm developed by the authors Guarino Lo Bianco and Piazzi (2001 a,b). This is a nested algorithm whose inner part evaluate the penalty terms $\Phi(\sigma_k^i(\mathbf{h}))$ and $\Phi(\tilde{\sigma}_k^i(\mathbf{h}))$, while the outer part minimizes the overall cost index (38).

Owing to (39) and (40), the penalty terms have to be evaluated by means of appropriate optimization routines. To this purpose, we devised an interval procedure that can be regarded as a special deterministic algorithm for global optimization. It is based on a branch-and-bound technique and uses operators borrowed by interval analysis, i.e. the mathematics of real intervals (Moore 1979). More details on the applications of interval analysis to algorithms for global optimization can be found in Hansen (1992), Ratschek and Rokne (1988) and Jaulin *et al.* (2001). For the application at hand, the interval procedure is not precisely aimed at finding the global minima (39) and (40) but at computing $\Phi(\sigma_k^i(\mathbf{h}))$ and $\Phi(\tilde{\sigma}_k^i(\mathbf{h}))$ as a whole. In fact, owing to the particular structure of $\Phi(\sigma)$, it is not always necessary to find the exact value of σ ($\sigma_k^i(\mathbf{h})$ or $\tilde{\sigma}_k^i(\mathbf{h})$). For example, when the interval procedure that internally computes improving lower and upper bounds of σ (say l_b and u_b) detects $\sigma \in [0, +\infty)$ (because $l_b \geq 0$) then it can stop the evaluation of σ and can immediately give the correct value of the penalty function (i.e. $\Phi(\sigma) = 0$). Hence the interval procedure exploiting this feature can save in many cases considerable computational time.

Penalty terms, evaluated by the interval procedure, are then passed to the genetic routine to minimize the overall cost index. The adopted genetic algorithm is a variant of the one presented in Menozzi *et al.* (1996) and uses standard operators: selection, mutation and crossover. It is a partially elitistic algorithm because, at each generation, the final population is derived by merging a certain number of individuals drawn from the previous population with the off-spring obtained by means of the genetic operators. The first four best individuals of the final population are further improved by using a local search routine.

The estimated global minimizer $\mathbf{h}^*$ for the optimization problem (38) is given by the individual that strictly satisfy all the semi-infinite constraints and has the best fitness over all the iterated generations. For the problem at hand, this means that the estimated global minimizer satisfies with certainty all the limits imposed on the joint torques and torque derivatives.

Problem (15) is a difficult one because the constraints (16) and (17) are in general non-linear and non-convex with respect to both arguments t and $\mathbf{h}$ and moreover they are of semi-infinite type, i.e. inequalities must hold for any value of t in the continuous interval $[0, h_i]$. In the pertinent mathematical literature, problem (15) is characterized as a generalized semi-infinite problem (see, for example, Reemtsen and Ruckmann 1998) and the known algorithms to solve it can only determine local solutions. It is worth stressing that when a given $\mathbf{h} \in \mathcal{B}$ is offered, checking with certainty its feasibility with respect to constraints (16) and (17) can be solely ascertained by using a deterministic global approach. This explains why, in many cases, using popular optimizers (such as the Matlab optimizer that is based on sequential quadratic programming) for difficult problems, the found 'optimal' solution is not even feasible (i.e. useless from a practical standpoint) and to overcome this flaw many artifices (*ad hoc* discretizations, smoothings, etc.) and repeated trials are necessary to hopefully arrive at an acceptable solution (cf. §4.1). On the contrary, the new idea developed with the proposed approach combining a stochastic global optimization technique, specifically a genetic algorithm, with a deterministic global method, an interval procedure, provides an estimated global solution that is feasible with certainty.

Remark 1: The interval procedure computing the penalty terms $\Phi(\sigma_k^i(\mathbf{h}))$ and $\Phi(\tilde{\sigma}_k^i(\mathbf{h}))$ uses 'inclusion functions' relative to $\tau_k^i(t;\mathbf{h})$ and $\dot{\tau}_k^i(t;\mathbf{h})$ respectively (see, for example, Ratschek and Rokne (1988) or Hansen (1992) for the specific concept of inclusion function). Hence, it is necessary or appropriate the symbolic computation with respect to the arguments t and $\mathbf{h}$ of the torques and their derivatives along each spline segment. On the practical side, this can be efficiently organized using a recursive code scheme. Starting with the iterative solution of the tridiagonal system (9) we use slack variables for the spline expression (2) and its derivatives (21), (22) and (23) and plug them into the dynamics expressions (12) and (14). To effectively evaluate these expressions we use the recursive Newton–Euler formulation (see Sciavicco and Siciliano (2000) that also reports the related computational complexity analysis) for the torque dynamics and the extension of this formulation (deduced with an automatic differentiation approach) for the torque derivative dynamics.

Remark 2: From a computational complexity viewpoint, problem (15) is NP-hard and any algorithm aiming at finding with certainty the exact global solution with arbitrary prespecified precision (i.e. an overall purely deterministic global method) would suffer the so-called curse of dimensionality. Only really trivial problems could be solved by such an algorithm and no significant robotics application, such as those proposed in §4, could be addressed. As known, Goldberg (1989), using a genetic algorithm to search on an n-dimensional space, such as $\mathcal{B}$ in our hybrid approach, alleviates the curse of dimensionality permitting to obtain good estimates of global minimizers even when n is relatively large. A complete description of the proposed genetic/interval algorithm is reported in Guarino Lo Bianco and Piazzi (2001 a). In the same paper, several semi-infinite test problems are formulated and solved with numerical details.

4. Examples

Two examples are exposed in this section. The first one concerns a simple two-link planar arm. It has been mainly proposed to compare the results obtained with the genetic/interval algorithm with those obtained with a standard algorithm for semi-infinite optimization. The second example takes into account an actual manipulator. The dynamics of a Puma 560 manipulator is used to plan an optimal minimum-time trajectory. The purpose is to demonstrate that the proposed approach can be fruitfully used with non-trivial problems.

4.1. *A planar two-link robot arm*

For a two-link ($m = 2$) planar arm with revolute joints, the task is to plan, under given dynamic constraints (limited torque and torque derivative), a minimum-time trajectory whose possible path is sketched in figure 1. More precisely, the distal end of the arm has to cross some via points ($s = 10$ and $n = 11$), given by their cartesian coordinates (see the first two columns of table 1). By solving the inverse kinematics these cartesian via points are converted into the joint via points reported in the last two columns of table 1. Note that the second and the penultimate knots have not been imposed, being associated to the two free joint displacements. The joint variable vector is $\mathbf{q} := [q_1\ q_2]^{\mathrm{T}}$ and belongs to the joint-space work envelope $\mathcal{Q} := [0, \pi/2] \times [-\pi, 0]$. Under the hypothesis of masses concentrated at the distal end of each link the following dynamics equations can be derived (Craig 1989, p. 204).

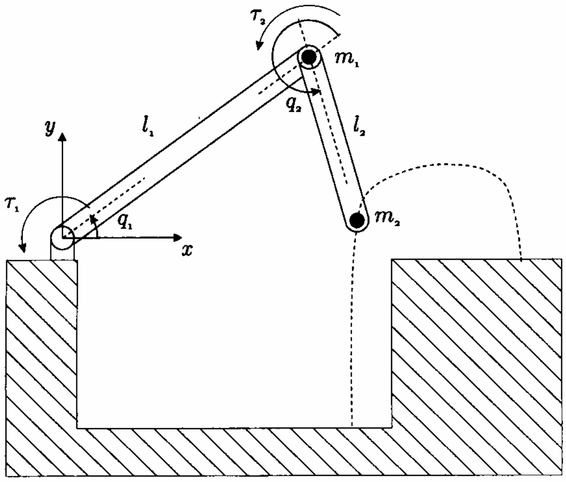

Figure 1. Schematic representation of the two-link planar arm.

$$\tau_1 = m_2 l_2^2(\ddot{q}_1 + \ddot{q}_2) + m_2 l_1 l_2 \cos(q_2)(2\ddot{q}_1 + \ddot{q}_2) + (m_1 + m_2) l_1^2 \ddot{q}_1 - m_2 l_1 l_2 \sin(q_2)\dot{q}_2^2 - 2m_2 l_1 l_2 \sin(q_2)\dot{q}_1\dot{q}_2 + m_2 l_2 g \cos(q_1 + q_2) + (m_1 + m_2) l_1 g \cos(q_1)$$

$$\tau_2 = m_2 l_1 l_2 \cos(q_2)\ddot{q}_1 + m_2 l_1 l_2 \sin(q_2)\dot{q}_1^2 + m_2 l_2 g \cos(q_1 + q_2) + m_2 l_2^2(\ddot{q}_1 + \ddot{q}_2)$$

where m_1 and m_2 are the links masses, g is the gravity acceleration, and l_1, l_2 are the links lengths. For the example at hand the following values have been adopted: $g = 9.8\,\mathrm{m/s^2}$, $l_1 = 1.0\,\mathrm{m}$, $l_2 = 0.5\,\mathrm{m}$, $m_1 = 15.0\,\mathrm{kg}$ and $m_2 = 7.0\,\mathrm{kg}$. By direct derivation of the above dynamics equations, the following expressions for the torque derivatives are obtained

$$\dot{\tau}_1 = m_2 l_2^2(\dddot{q}_1 + \dddot{q}_2) + (m_1 + m_2) l_1^2 \dddot{q}_1 + m_2 l_1 l_2 \cos(q_2)\,[2\dddot{q}_1 + \dddot{q}_2 - \dot{q}_2^3 - 2\dot{q}_2^2\dot{q}_1] - m_2 l_1 l_2 \sin(q_2)[4\ddot{q}_1\dot{q}_2 + 2\dot{q}_1\ddot{q}_2 + 3\ddot{q}_2\dot{q}_2] - l_1 g(m_1 + m_2)\sin(q_1)\dot{q}_1 - m_2 l_2 g \sin(q_1 + q_2)\,(\dot{q}_1 + \dot{q}_2)$$

$$\dot{\tau}_2 = m_2 l_2^2(\dddot{q}_1 + \dddot{q}_2) + m_2 l_1 l_2 \cos(q_2)[\dddot{q}_1 + \dot{q}_2\dot{q}_1^2] - m_2 l_2 g(\dot{q}_1 + \dot{q}_2)\sin(q_1 + q_2) + m_2 l_1 l_2 \sin(q_2)[2\dot{q}_1\ddot{q}_1 - \ddot{q}_1\dot{q}_2]$$

The vector of the interval times that parameterizes the cubic spline trajectory is $\mathbf{h} := [h_1\ h_2\ \cdots\ h_{11}]^{\mathrm{T}} \in \mathcal{B} := [0.02, 10.0]^{11}$ (it has been fixed $\nu = 0.02\,\mathrm{s}$). Consider the arm at rest in the initial and final positions (i.e. $\mathbf{v}^0 = \mathbf{v}^{11} = 0$; $\mathbf{a}^0 = \mathbf{a}^{11} = 0$) and by virtue of Property 2, problem (15) admits a solution if $\boldsymbol{\alpha} > [\max_{\mathbf{q}\in Q}\{g_1(\mathbf{q})\}\ \ \max_{\mathbf{q}\in Q}\{g_2(\mathbf{q})\}]^{\mathrm{T}} = [249.9\ \ 34.3]^{\mathrm{T}}\,\mathrm{Nm}$ and $\boldsymbol{\beta} > [0\ \ 0]^{\mathrm{T}}$. Then, a minimum-time trajectory planning is sought by fixing $\boldsymbol{\alpha} = [260\ \ 50]^{\mathrm{T}}\,\mathrm{Nm}$, and $\boldsymbol{\beta} = [300\ \ 200]^{\mathrm{T}}\,\mathrm{Nm\,s^{-1}}$. The optimization problem is solved by using two different algorithms: the genetic/interval algorithm developed by the authors and the algorithm for semi-infinite optimization provided by the Matlab Optimization Toolbox (Grace 1994). The results, given by the estimated minimizers, are reported in table 2. A single result is proposed for the genetic/interval algorithm owing to its capability to converge to an estimated global minimizer (see the first column of table 2) whereas three different results are shown for the Matlab algorithm. Indeed, the Matlab algorithm for semi-infinite optimization, as well as almost all the standard algorithms, normally converges to local solutions depending on the chosen starting point $\mathbf{h}_0$. Defined $\mathbf{h}_0 := [\eta\ \eta\ \cdots\ \eta]^{\mathrm{T}}$, the three solutions found by the Matlab algorithm refer to the following starting points: $\eta = 10$ (the far end of $\mathcal{B}$), $\eta = 5$ (the middle point of $\mathcal{B}$) and $\eta = 1$. The total travelling times for the four cases

$x^0 = 1.00$	$y^0 = -0.500$	$q_1^0 = 0.0000$	$q_2^0 = -1.5708$
$x^2 = 1.00$	$y^2 = -0.375$	$q_1^2 = 0.1253$	$q_2^2 = -1.6804$
$x^3 = 1.00$	$y^3 = -0.250$	$q_1^3 = 0.2517$	$q_2^3 = -1.7594$
$x^4 = 1.00$	$y^4 = -0.125$	$q_1^4 = 0.3789$	$q_2^4 = -1.8074$
$x^5 = 1.00$	$y^5 = 0.000$	$q_1^5 = 0.5054$	$q_2^5 = -1.8235$
$x^6 = 1.05$	$y^6 = 0.100$	$q_1^6 = 0.5837$	$q_2^6 = -1.7087$
$x^7 = 1.15$	$y^7 = 0.200$	$q_1^7 = 0.6119$	$q_2^7 = -1.4581$
$x^8 = 1.30$	$y^8 = 0.100$	$q_1^8 = 0.4263$	$q_2^8 = -1.1040$
$x^9 = 1.30$	$y^9 = 0.050$	$q_1^9 = 0.3903$	$q_2^9 = -1.1124$
$x^{11} = 1.30$	$y^{11} = 0.000$	$q_1^{11} = 0.3526$	$q_2^{11} = -1.1152$

Table 1. End effector via points expressed in meters and equivalent joint via points expressed in radians for the two-link arm.

	Genetic/ interval	Matlab $\eta = 10$	Matlab $\eta = 5$	Matlab $\eta = 1$
h_1^*	0.14525	0.62483	0.14838	0.14793
h_2^*	0.27951	0.02000	0.27568	0.27765
h_3^*	0.15158	0.17366	0.15686	0.15097
h_4^*	0.13267	0.13859	0.15783	0.12982
h_5^*	0.14022	0.14271	0.21708	0.13686
h_6^*	0.12443	0.12536	0.16241	0.12110
h_7^*	0.17323	0.17501	0.19621	0.16223
h_8^*	0.43928	0.44309	0.42958	0.44926
h_9^*	0.10151	0.10670	0.10108	0.10107
h_{10}^*	0.19062	0.19678	0.19138	0.19182
h_{11}^*	0.11185	0.10888	0.10942	0.10934
$\sum_{i=1}^{n} h_i^*$	1.9902	2.2558	2.1459	2.0105

Table 2. Comparison of solutions (in seconds) for the two-link arm.

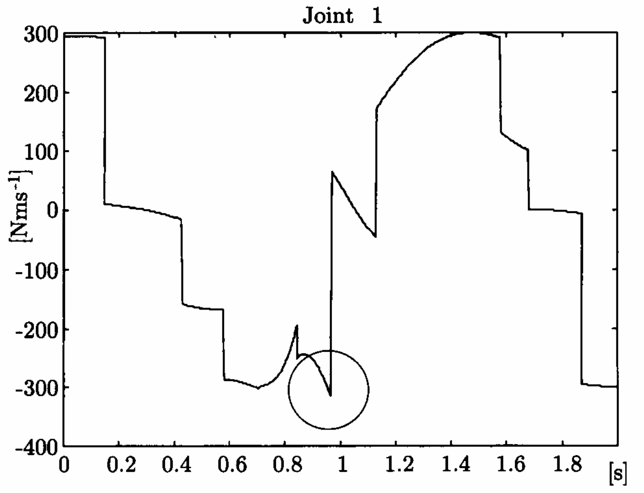

Figure 2. The 'optimal' derivative profile for the first joint of the two-link arm evaluated by the Matlab Optimization Toolbox. The semi-infinite constraint is violated inside the area evidenced by the circle.

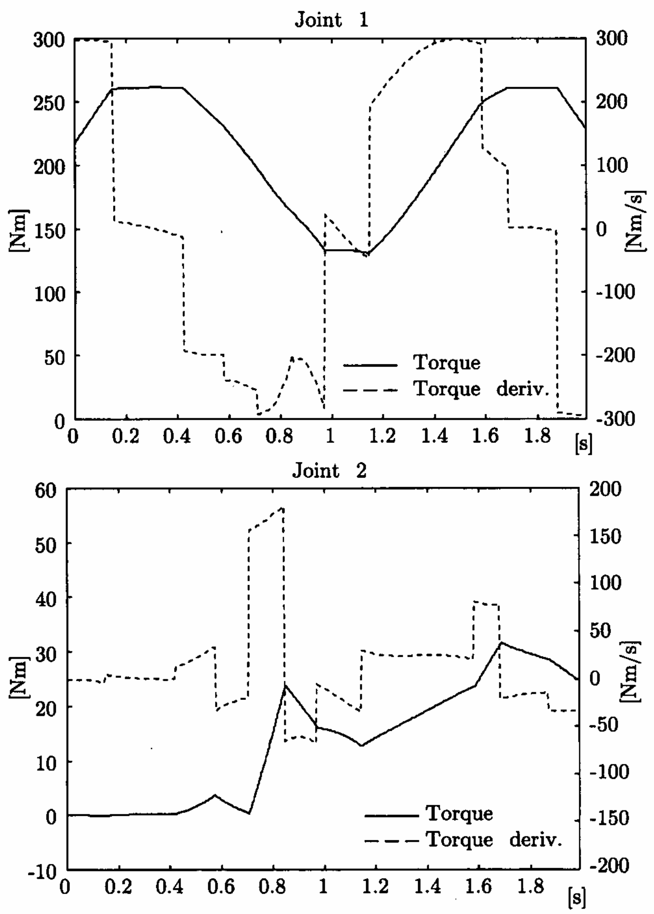

Figure 3. Estimated optimal torque and torque derivative profiles found by the genetic/interval algorithm for the two-link arm.

are reported in the last row of table 2. The total travelling times obtained with $\eta = 10$, $\eta = 5$ and $\eta = 1$ are respectively 13.3%, 7.8% and 1.0% greater than the corresponding minimum-time obtained by means of the genetic/interval algorithm. Many tries can be experimented by running the Matlab optimizer with other different starting points. For example, when η is chosen too small, such as $\eta = 0.02$, the Matlab algorithm stops by warning that it is unable to find feasible solutions.

Some tests have been done to check the actual feasibility of the solutions. The solution found by the genetic/interval algorithm is indeed feasible while all the solutions found by the Matlab algorithm are actually unfeasible. Sometimes the constraints are only slightly violated but sometimes a consistent error is detected. For example, the solution corresponding to $\eta = 1$ (see figure 2) produces a torque derivative minimum on the first joint equal to $-316.04\,\mathrm{Nm/s}$, i.e. the constraint violation is equal to a 5.3% excess. What's going on? Simply the Matlab algorithm evaluates the semi-infinite constraint functions on a grid of values chosen over the time intervals $[0, h_i]$, $i = 1, 2, \ldots, n$. The grid thickness is chosen on the basis of heuristic rules. Hence the feasibility of the minimizer is not guaranteed with certainty and, on the contrary, the probability to violate the imposed limits is high when the involved functions are not continuous. This is the case for the torque derivative functions.

The optimal torque and torque derivative profiles evaluated by the genetic/interval algorithm are plotted in figure 3. These plots makes evident that, at joint 1, both the torque and torque derivative constraints are active. The resulting path in Cartesian plane is shown in figure 4 where the crosses denote the assigned via points.

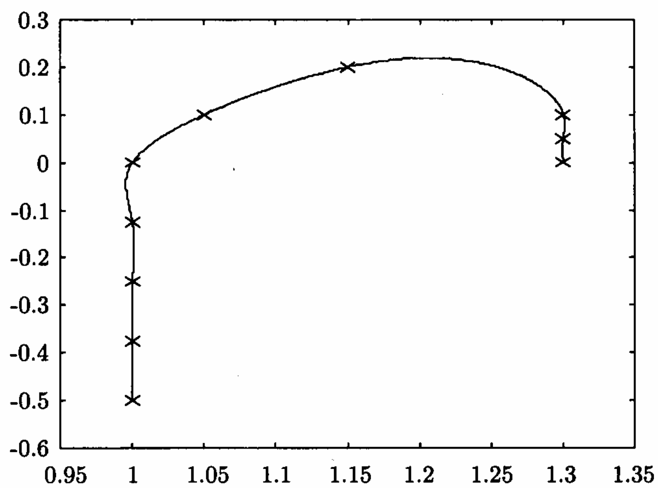

Figure 4. Estimated optimal path in Cartesian space found by the genetic/interval algorithm.

$q_1^0 = -0.1745$	$q_2^0 = 0.3491$	$q_3^0 = 0.2618$	$q_4^0 = 2.6180$	$q_5^0 = 0.5236$	$q_6^0 = 2.0944$
$q_1^2 = 1.0472$	$q_2^2 = 0.8727$	$q_3^2 = 1.7453$	$q_4^2 = 1.7453$	$q_5^2 = 1.9199$	$q_6^2 = 1.0472$
$q_1^3 = 0.3491$	$q_2^3 = 2.0944$	$q_3^3 = -0.1745$	$q_4^3 = 0.6981$	$q_5^3 = 1.5708$	$q_6^3 = 1.7453$
$q_1^5 = 0.9599$	$q_2^5 = 0.6109$	$q_3^5 = 0.5236$	$q_4^5 = 0.1745$	$q_5^5 = 1.2217$	$q_6^5 = 0.4363$

Table 3. The given joint via points (in radians) for the PUMA 560 manipulator.

4.2. *A six-link Puma 560 manipulator*

The Puma 560 manipulator is a well known six-link industrial robot ($m = 6$). Again we want to plan under given dynamic constraints a minimum-time trajectory crossing the through points reported in table 3 ($s = 4$ and $n = 5$). The joint variable vector is $\mathbf{q} := [q_1\ q_2\ q_3\ q_4\ q_5\ q_6]^{\mathrm{T}}$ and belongs to the joint-space work envelope $\mathcal{Q} := [-\pi, \pi]^6$. The reference frames are affixed to the links according to the modified Denavit–Hartenberg representation. No external forces or torques are acting on the end effector. The manipulator kinematic parameters are obtained from Armstrong *et al.* (1986) and listed in table 4. From the same paper we also derived the dynamic parameters reported in table 5 (link masses and centers of gravity) and table 6 (moments of inertia for the links). Owing to the complexity of the dynamic equations the genetic/interval algorithm evaluates both torques and torques derivative by means of a recursive procedure. Torques are evaluated by means of the recursive Newton–Euler formulation and new recursive relations have been developed for the computation of their derivatives (see Remark 1 in § 3). Moreover, these recursive computational procedures have been further improved exploiting the specific PUMA 560 dynamics in a manner similar to that presented in Corke (1996).

The vector of the interval times that parameterizes the cubic spline trajectory is

$$\mathbf{h} := [h_1\ h_2\ \cdots\ h_5]^{\mathrm{T}} \in \mathcal{B} := [0.02, 10.0]^5$$

Consider the manipulator at rest in the initial and final positions (i.e. $\mathbf{v}^0 = \mathbf{v}^5 = 0; \mathbf{a}^0 = \mathbf{a}^5 = 0$). As a consequence, by virtue of Property 2, problem (15) admits a solution if

Joint	α_{i-1}	A_{i-1}	D_i
1	0	0	0
2	$-\pi/2$	0	243.5
3	0	431.8	−93.4
4	$\pi/2$	−20.3	433.1
5	$-\pi/2$	0	0
6	$\pi/2$	0	0

Table 4. Kinematic parameters for the Puma 560 manipulator (angles are expressed in radians and lengths in mm).

Joint	m	x	y	z
1	—	—	—	—
2	17.40	0.068	0.006	−0.016
3	4.80	0.0	−0.070	0.014
4	0.82	0.0	0.0	−0.019
5	0.35	0.0	0.0	0.0
6	0.09	0.0	0.0	0.032

Table 5. Link masses (kg) and coordinates of the centers of gravity (m) for the Puma 560 manipulator.

Joint	I_{xx}	I_{yy}	I_{zz}
1	—	—	0.35
2	0.130	0.524	0.539
3	0.066	0.0125	0.086
4	1.80e−3	1.80e−3	1.30e−3
5	0.30e−3	0.30e−3	0.40e−3
6	0.15e−3	0.15e−3	0.04e−3

Table 6. Link moments of inertia (kg m^2) for the PUMA 560 manipulator.

$$\boldsymbol{\alpha} > [\max_{\mathbf{q}\in\mathcal{Q}} \{|g_1(\mathbf{q})|\}\ \max_{\mathbf{q}\in\mathcal{Q}} \{|g_2(\mathbf{q})|\}\ \ldots \max_{\mathbf{q}\in\mathcal{Q}} \{|g_6(\mathbf{q})|\}]^{\mathrm{T}}$$
$$= [0.000\ 45.8195\ 8.5565\ 0.0283\ 0.0283\ 0.000]^{\mathrm{T}}\,\mathrm{Nm}$$

and $\boldsymbol{\beta} > [0\ 0\ 0\ 0\ 0\ 0]^{\mathrm{T}}$. Two cases have been investigated to highlight the effects deriving from the use of the torque derivative constraints:

(1) $\boldsymbol{\alpha} = [44.8\ 77.6\ 41.6\ 8.0\ 8.0\ 8.0]^{\mathrm{T}}\,\mathrm{Nm}$, $\boldsymbol{\beta} = [\infty\ \infty\ \infty\ \infty\ \infty\ \infty]^{\mathrm{T}}\,\mathrm{Nm\,s^{-1}}$ (torque constraints only);

(2) $\boldsymbol{\alpha} = [44.8\ 77.6\ 41.6\ 8.0\ 8.0\ 8.0]^{\mathrm{T}}\,\mathrm{Nm}$, $\boldsymbol{\beta} = [200\ 150\ 70\ 10\ 10\ 10]^{\mathrm{T}}\,\mathrm{Nm\,s^{-1}}$ (torque and torque derivative constraints).

The torque limits have been chosen by taking into account the maximum motor performance limits proposed in Corke (1996); more precisely $\boldsymbol{\alpha}$ is 80% of the limits proposed in that paper. The results, given by the two estimated global minimizers, are reported in table 7.

Case 1	Case 2
$h_1^* = 0.07551\,\mathrm{s}$	$h_1^* = 0.19520\,\mathrm{s}$
$h_2^* = 0.78565\,\mathrm{s}$	$h_2^* = 0.66491\,\mathrm{s}$
$h_3^* = 0.32411\,\mathrm{s}$	$h_3^* = 0.64081\,\mathrm{s}$
$h_4^* = 0.25160\,\mathrm{s}$	$h_4^* = 0.48053\,\mathrm{s}$
$h_5^* = 0.10890\,\mathrm{s}$	$h_5^* = 0.23559\,\mathrm{s}$

Table 7. The estimated global minimizer $\mathbf{h}^*$ for case 1 and case 2 (PUMA 560 manipulator).

For case 1, the minimum travelling time is $\sum_{i=1}^{5} h_i^* = 1.5458\,\mathrm{s}$. The upper plots of figures from 5 to 10 show, at the optimal solution $\mathbf{h}^*$, both the torque and the torque derivative profiles for all the joints. The constraint on the torque is active for joints 1 and 2: the torque limit α_1 is reached three times by joint 1 and joint 2 touches α_2 twice. It is also evident that the torque variations are probably unacceptably large. The torque derivative peaks are typically detected at the beginning or at the end of the trajectory or when the motion

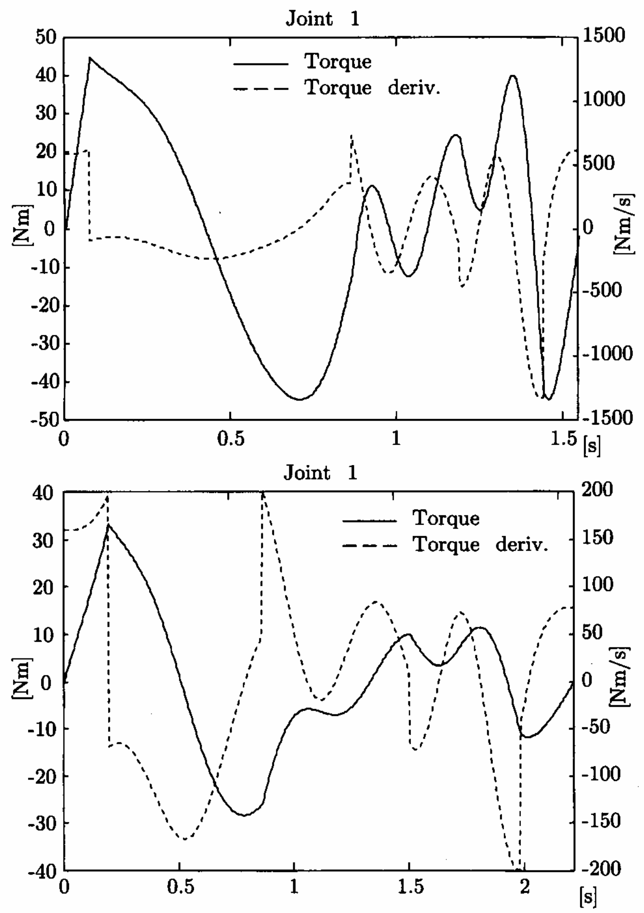

Figure 5. Optimal torque and torque derivative profiles for the first joint; the upper plots refer to case 1 and the lower ones to case 2.

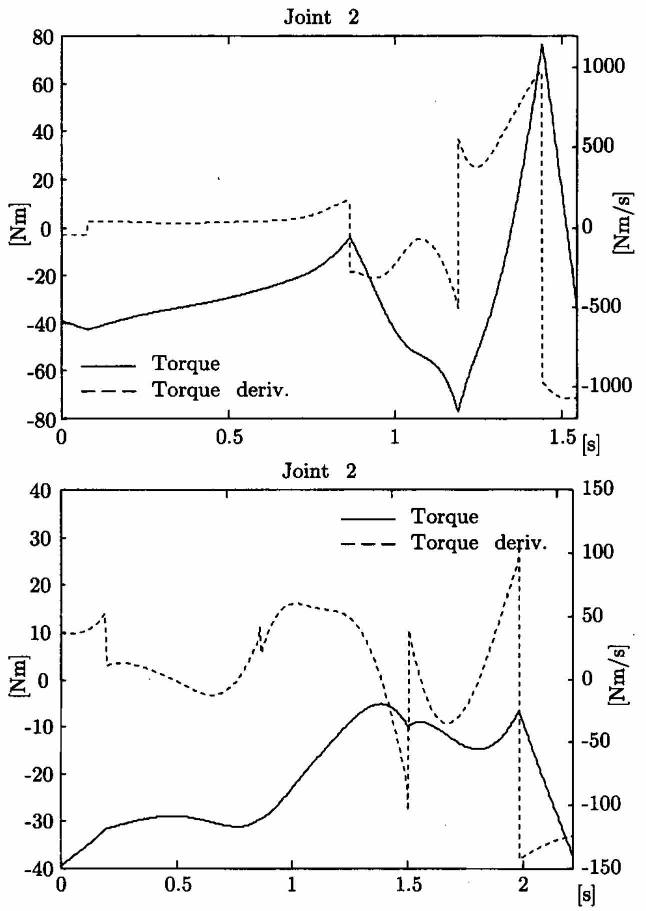

Figure 6. Optimal torque and torque derivative profiles for the second joint; the upper plots refer to case 1 and the lower ones to case 2.

direction changes. These large torque derivatives can be dampened by adding the new constraints of case 2. The resulting optimal torque and torque derivative profiles are shown into the lower plots of figures 5–10. In this case, the torque limit $\boldsymbol{\alpha}$ is never reached but the derivative constraints are active for both joint 1 and 2. Overall, the optimal trajectory planning for the case 2 appears better with respect to case 1 because the moderate torque variations reduce the mechanical solicitations on the manipulator and the actuator control system can more easily track the precomputed torques. The price to be paid is an increase of the minimum travelling time: $\sum_{i=1}^{5} h_i^* = 2.2170\,\mathrm{s}$ for case 2. In spite of the complexity of the problem involved (60 semi-infinite constraints are evaluated at each iteration, all of them composed by long non-linear expressions; cf. Corke 1996) and the use of a standard personal computer based on a Pentium III 733 MHz processor, a reasonable computation time has been necessary to get the estimated global minimizers: 11h 6m for case 1 and 16h 40m for case 2.

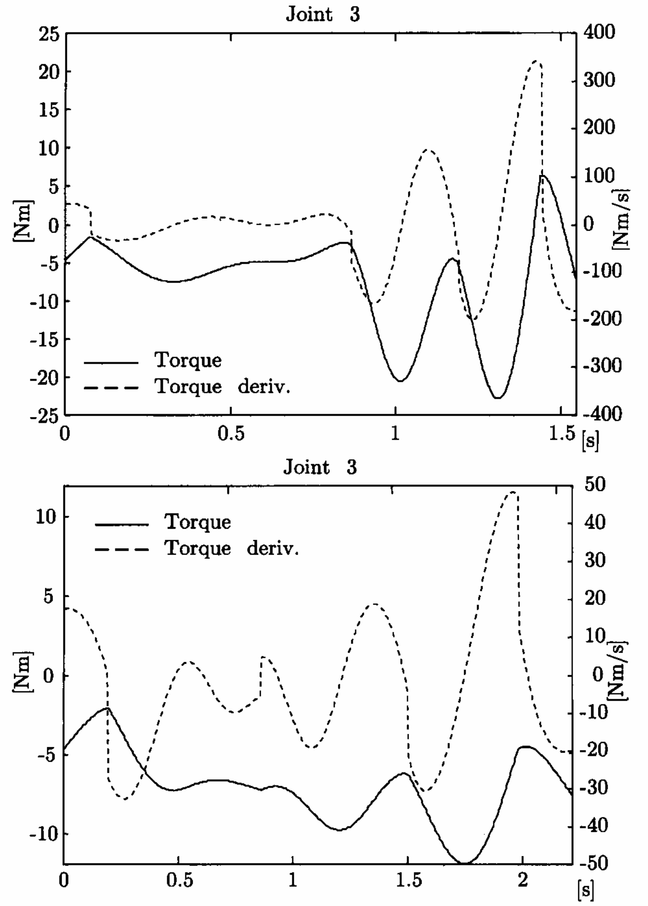

Figure 7. Optimal torque and torque derivative profiles for the third joint; the upper plots refer to case 1 and the lower ones to case 2.

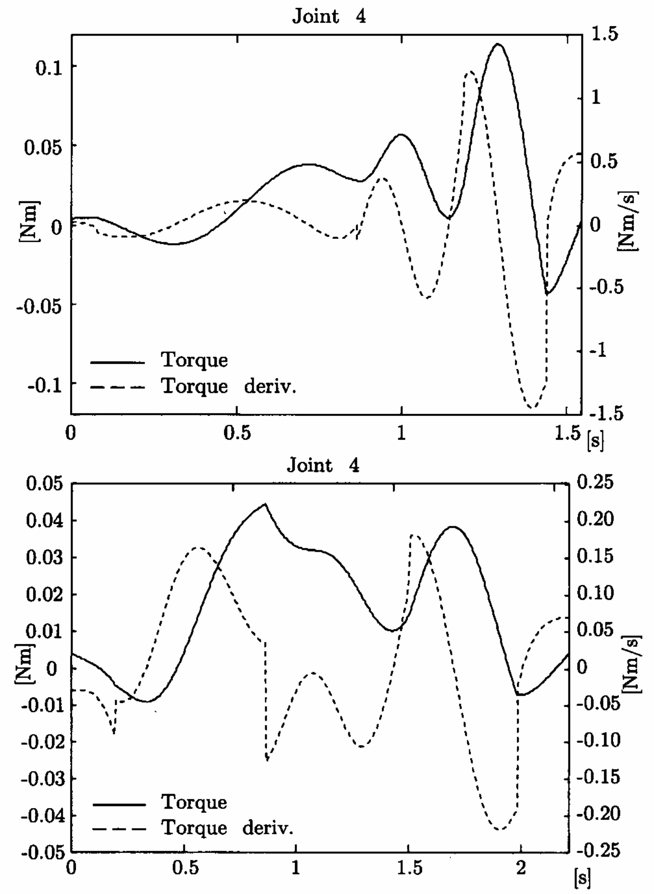

Figure 8. Optimal torque and torque derivative profiles for the fourth joint; the upper plots refer to case 1 and the lower ones to case 2.

5. Conclusions

A new method for the minimum-time planning of manipulator trajectories has been proposed. The planning uses cubic splines that guarantee continuity of velocities and accelerations and exactly interpolate given via points. An hybrid genetic/interval algorithm is adopted for solving a resulting global minimum-time problem that explicitly takes into account the dynamic constraints arising by limiting the torques and the torque derivatives. Two examples, reported with numerical details, indicate the advantages deriving from using trajectories with limited joint torque variations and demonstrate the effectiveness of the proposed global approach when compared with standard optimization algorithms. For example, the guaranteed feasibility of the estimated global solution is a definite advantage of the hybrid algorithm (cf. §4.1) whereas alternative standard optimizers cannot establish feasibility with certainty. The proposed method appears particularly suitable for the trajectory planning of manipulators used in automated factories.

As a cautionary remark, we may add that the resulting minimum-time planning relies on the manipulator dynamics (12) and this does not comprise various effects possibly affecting the actual dynamics, such as friction, actuator dynamics, inaccurate parameters, wear, etc. Nevertheless, the proposed planning may be very useful for the torque command control of real manipulators (Craig 1989, p. 341) especially when it is compared with trajectory planners that do not even consider to limit the torque variations. Moreover, the approach devised in this paper as relying on an efficient hybrid global optimizer could be extended to deal with a more accurate modelling of the manipulator dynamics, for example to address trajectory planning problems where the mechanical dynamics is tightly coupled with the actuator dynamics.

Acknowledgements

This work was partially supported by MURST scientific research funds and ASI (Italian Space Agency).

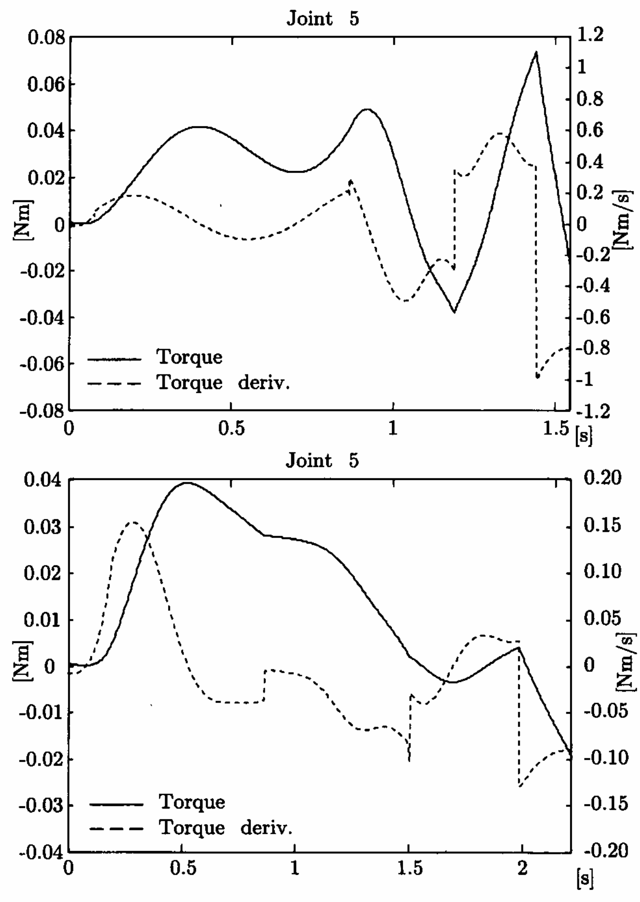

Figure 9. Optimal torque and torque derivative profiles for the fifth joint; the upper plots refer to case 1 and the lower ones to case 2.

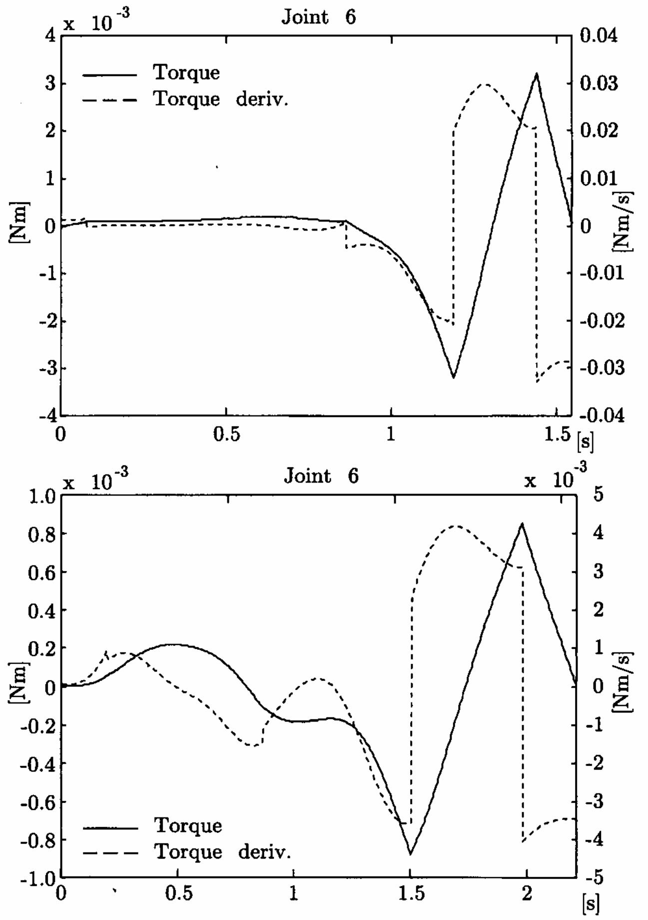

Figure 10. Optimal torque and torque derivative profiles for the sixth joint; the upper plots refer to case 1 and the lower ones to case 2.

Appendix

Let be given the vector function $\mathbf{x}(t) := [x_1(t) \;\cdots\; x_m(t)]^{\mathrm{T}} \in \mathbb{R}^m$ and the matrix function $\mathbf{A}(\mathbf{x}(t)) \in \mathbb{R}^{p\times q}$ depending on $\mathbf{x}(t)$. Then, the derivation rule holds

$$\frac{\mathrm{d}}{\mathrm{d}t}[\mathbf{A}(\mathbf{x}(t))] = \frac{\partial \mathbf{A}(\mathbf{x}(t))}{\partial \mathbf{x}}(I_q \otimes \dot{\mathbf{x}}(t)) \qquad (42)$$

where $\otimes$ indicates the Kronecker product (Brewer 1978) and I_q is an identity matrix of order q. The partial derivative of matrix $\mathbf{A}(\mathbf{x})$ with respect to vector $\mathbf{x}$ be defined as

$$\frac{\partial \mathbf{A}}{\partial \mathbf{x}} := \begin{bmatrix} \dfrac{\partial A_{11}}{\partial \mathbf{x}} & \dfrac{\partial A_{12}}{\partial \mathbf{x}} & \cdots & \dfrac{\partial A_{1q}}{\partial \mathbf{x}} \\ \dfrac{\partial A_{21}}{\partial \mathbf{x}} & \dfrac{\partial A_{22}}{\partial \mathbf{x}} & \cdots & \dfrac{\partial A_{2q}}{\partial \mathbf{x}} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial A_{p1}}{\partial \mathbf{x}} & \dfrac{\partial A_{p2}}{\partial \mathbf{x}} & \cdots & \dfrac{\partial A_{pq}}{\partial \mathbf{x}} \end{bmatrix} \qquad (43)$$

where

$$\frac{\partial A_{ij}}{\partial \mathbf{x}} := \begin{bmatrix} \dfrac{\partial A_{ij}}{\partial x_1} & \dfrac{\partial A_{ij}}{\partial x_2} & \cdots & \dfrac{\partial A_{ij}}{\partial x_m} \end{bmatrix} \qquad (44)$$

Proof: The time derivative of matrix $\mathbf{A}(\mathbf{x}(t))$ can be explicitly expressed by

$$\frac{\mathrm{d}}{\mathrm{d}t}[\mathbf{A}(\mathbf{x}(t))] = \begin{bmatrix} \dfrac{\mathrm{d}}{\mathrm{d}t}A_{11}(\mathbf{x}(t)) & \dfrac{\mathrm{d}}{\mathrm{d}t}A_{12}(\mathbf{x}(t)) & \cdots & \dfrac{\mathrm{d}}{\mathrm{d}t}A_{1q}(\mathbf{x}(t)) \\ \dfrac{\mathrm{d}}{\mathrm{d}t}A_{21}(\mathbf{x}(t)) & \dfrac{\mathrm{d}}{\mathrm{d}t}A_{22}(\mathbf{x}(t)) & \cdots & \dfrac{\mathrm{d}}{\mathrm{d}t}A_{2q}(\mathbf{x}(t)) \\ \vdots & \vdots & & \vdots \\ \dfrac{\mathrm{d}}{\mathrm{d}t}A_{p1}(\mathbf{x}(t)) & \dfrac{\mathrm{d}}{\mathrm{d}t}A_{p2}(\mathbf{x}(t)) & \cdots & \dfrac{\mathrm{d}}{\mathrm{d}t}A_{pq}(\mathbf{x}(t)) \end{bmatrix} \qquad (45)$$

Matrix (45) has order $p \times q$ and is composed by scalar terms. For each of them the time derivative is given by the relation

$$\frac{\mathrm{d}}{\mathrm{d}t}A_{ij}(\mathbf{x}(t)) = \frac{\partial A_{ij}(\mathbf{x}(t))}{\partial x_1}\dot{x}_1(t) + \cdots + \frac{\partial A_{ij}(\mathbf{x}(t))}{\partial x_m}\dot{x}_m(t) = \frac{\partial A_{ij}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) \quad (46)$$

By using (46) it is possible to write (45) in a more compact form. In fact, we have

$$\frac{d}{dt}[\mathbf{A}(\mathbf{x}(t))] = \begin{bmatrix} \frac{\partial A_{11}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) & \frac{\partial A_{12}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) & \cdots & \frac{\partial A_{1q}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) \\ \frac{\partial A_{21}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) & \frac{\partial A_{22}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) & \cdots & \frac{\partial A_{2q}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) \\ \vdots & \vdots & & \vdots \\ \frac{\partial A_{p1}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) & \frac{\partial A_{p2}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) & \cdots & \frac{\partial A_{pq}(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial A_{11}(\mathbf{x}(t))}{\partial \mathbf{x}} & \cdots & \frac{\partial A_{1q}(\mathbf{x}(t))}{\partial \mathbf{x}} \\ \vdots & & \vdots \\ \frac{\partial A_{p1}(\mathbf{x}(t))}{\partial \mathbf{x}} & \cdots & \frac{\partial A_{pq}(\mathbf{x}(t))}{\partial \mathbf{x}} \end{bmatrix} \times \begin{bmatrix} \dot{\mathbf{x}}(t) & 0 & \cdots & \cdots & 0 \\ 0 & \dot{\mathbf{x}}(t) & 0 & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & 0 & \dot{\mathbf{x}}(t) & 0 \\ 0 & \cdots & \cdots & 0 & \dot{\mathbf{x}}(t) \end{bmatrix}$$

$$= \frac{\partial \mathbf{A}(\mathbf{x}(t))}{\partial \mathbf{x}}(I_q \otimes \dot{\mathbf{x}}(t)) \qquad \square$$

References

ARMSTRONG, B., KHATIB, O., and BURDICK, J., 1986, The explicit dynamic model and inertial parameters of the puma 560 arm. *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, Washington, USA, pp. 510–518.

BOBROW, J. E., DUBOWSKY, S., and GIBSON, J. S., 1985, Time-optimal control of robotic manipulators along specified paths. *International Journal of Robotics Research*, **4**, 554–561.

BREWER, J. W., 1978, Kronocker products and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems*, **25**, 772–781.

CORKE, P. I., 1996, An automated symbolic and numeric procedure for manipulator rigid-body dynamic significance analysis and simplication. *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, USA, pp. 1018–1023.

CRAIG, J. J., 1989, *Introduction to Robotics: Mechanics and Control* (Reading, MA: Addison-Wesley).

DE LUCA, A., LANARI, L., and ORIOLO, G., 1991, A sensitivity approach to optimal spline robot trajectories. *Automatica*, **27**, 535–539.

GOLDBERG, D., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Reading, MA: Addison-Wesley).

GRACE, A., 1994, *Optimization Toolbox User's Guide* (Reading, MA: The Mathworks, Inc.).

GUARINO LO BIANCO, C., and PIAZZI, A., 1999, A genetic/interval aproach to optimal trajectory planning of industrial robots under torque constraints. *Proceedings of the 1999 European Control Conference*, Karlsruhe, Germany.

GUARINO LO BIANCO, C., and PIAZZI, A., 2001a, A hybrid algorithm for infinitely constrained optimization. *International Journal of Systems Science*, **32**, 91–102.

GUARINO LO BIANCO, C., and PIAZZI, A., 2001b, A semi-infinite optimization approach to optimal spine trajectory planning of mechanical manipulators. In M. Goberna and M. López (eds) *Semi-Infinite Programming: Recent Advances*, Vol. 57, *Nonconvex Optimization and its Applications* (Dordrecht, The Netherlands, Kluwer Academic), Chapter 13, pp. 271–297.

HANSEN, E., 1992, *Global Optimization Using Interval Analysis* (New York: Marcel Dekker).

JAULIN, L., KIEFFER, M., DIDRIT, O., and WALTER, E., 2001, *Applied Interval Analysis* (London: Springer).

LIN, C.-S., CHANG, P.-R., and LUH, J., 1983, Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Transactions on Automatic Control*, **28**, 1066–1074.

MENOZZI, R., PIAZZI, A., and CONTINI, F., 1996, Small-signal modeling for microwave FET linear circuits based on a genetic algorithm. *IEEE Transactions on Circuits and Systems, Part I: Fundamental Theory and Applications*, **43**, 839–847.

MOORE, R., 1979, *Methods and Applications of Interval Analysis* (Philadelphia, PA: SIAM Press).

PIAZZI, A., and VISIOLI, A., 1998, Global minimum-time trajectory planning of mechanical manipulators using interval analysis. *International Journal of Control*, **71**, 631–652.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., and FLANNERY, B. P., 1992, *Numerical Recipes in C*, 2nd edn (Cambridge, UK: Cambridge University Press).

RATSCHEK, H., and ROKNE, J., 1988, *New Computer Methods for Global Optimization* (Chichester, UK: Ellis Horwood Limited).

REEMTSEN, R., and RUCKMANN, J.-J. (eds), 1998, *Semi-Infinite Programming, Nonconvex Optimization and Its Applications* (Dordrecht, The Netherlands: Kluwer Academic).

SCIAVICCO, L., and SICILIANO, B., 2000, *Modelling and Control of Robot Manipulators*, Advanced Textbooks in Control and Signal Processing (Berlin, Germany: Springer-Verlag).

SHIN, K. G., and MCKAY, N. D., 1985, Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, **30**, 531–541.

ŽLAJPAH, L., 1996, On time optimal path control of manipulators with bounded joint velocities and torques. *Proceedings of the 1996 International Conference on Robotics and Automation*, Minneapolis, Minnesota, pp. 1572–1577.