Open access • Journal Article • DOI:10.1109/41.824136

Global minimum-jerk trajectory planning of robot manipulators — Source link

Aurelio Piazzi, Antonio Visioli

Institutions: University of Brescia

Published on: 01 Feb 2000 - IEEE Transactions on Industrial Electronics (IEEE)

Topics: Motion planning and Trajectory

Global Minimum-Jerk Trajectory Planning of Robot Manipulators

Aurelio Piazzi, *Member, IEEE,* and Antonio Visioli, *Member, IEEE*

***Abstract*—A new approach based on interval analysis is developed to find the global minimum-jerk (MJ) trajectory of a robot manipulator within a joint space scheme using cubic splines. MJ trajectories are desirable for their similarity to human joint movements and for their amenability to path tracking and to limit robot vibrations. This makes them attractive choices for robotic applications, in spite of the fact that the manipulator dynamics is not taken into account. Cubic splines are used in a framework that assures overall continuity of velocities and accelerations in the robot movement. The resulting MJ trajectory planning is shown to be a global constrained minimax optimization problem. This is solved by a newly devised algorithm based on interval analysis and proof of convergence with certainty to an arbitrarily good global solution is provided. The proposed planning method is applied to an example regarding a six-joint manipulator and comparisons with an alternative MJ planner are exposed.**

***Index Terms*—Global optimization, interval algorithm, manipulator trajectory planning, minimum jerk.**

I. Introduction

IT IS well known that robot manipulators are highly nonlinear, coupled multivariable systems with nonlinear constraints. For this reason, making an optimal control is a very difficult task and provided methods result in impractically complicated algorithms. An alternative simpler approach is to plan a robot path along which an optimization can be achieved and then to control the robot to track the path. Thus, the optimum control problem is actually divided in two separate steps: optimum path planning for off-line processing, followed by on-line path tracking. In order to reduce the computational effort, path planning is often performed taking into account kinematic constraints and disregarding the robot dynamics. A sequence of path points, whose number is a tradeoff between exactness and computational expense, is usually specified in terms of a desired position and orientation of the tool frame in the Cartesian space. Each of these via points is then mapped into a set of joint angles/offsets (knots) by application of the inverse kinematics. These knots are then interpolated with smooth functions to be optimized subject to constraints accordingly chosen for a specific robot application. Cubic splines are widely used for interpolation since they can assure the continuity of velocity and acceleration [1] and prevent large oscillations of the trajectory which can result with higher order polynomials. In this framework, the total motion time can be minimized subject to constraints on velocities, accelerations, and jerks [1], [2].

If the traveling time is determined by the operative context (e.g., due to time constraints in automated robotic cells), an interesting approach is to minimize the jerk. The importance of minimizing the jerk in trajectory planning is primarily due to the fact that joint position errors decrease when the jerk decreases, as asserted by Kyriakopoulos and Saridis, who proposed the formulation of an analytic solution for minimum-jerk (MJ) point-to-point trajectories [3]. Moreover, MJ joint trajectories are desirable for their similarity to human joints movements and to limit excessive wear on the robot and the excitation of resonances so that the robot life-span is expanded [4]. With this aim, using trigonometric splines, Simon and Isik [5] presented a closed-form solution to minimize the integral of the squared jerk under the suboptimal assumption that the time spacing between the knots is even. For the same problem, Simon [6] devised a stochastic optimization method based on neural networks in order to obtain trajectories that are however numerical rather than analytic functions of time.

In this paper, in order to obtain MJ cubic splines joint trajectories, we first formalized the proposed planning as a global constrained minimax optimization problem. Then, to solve it, we expose a newly devised algorithm, based on interval analysis, which converges within an arbitrary precision to a global solution. This MJ algorithm is logically divided into two parts. The first one is dedicated to construct a new search domain that eliminates the linear constraint (given by the imposed total motion time) so to obtain an equivalent unconstrained minimax optimization problem. The second part solves this unconstrained problem through an exhaustive branch-and-bound procedure where the bounding is performed via inclusion functions. The concept of inclusion function (see Section III) is fundamental in interval analysis, which is a generalization of the "standard" real analysis over the arithmetic of real intervals [7]. Interval analysis has been proved to be a very useful tool for global optimization [8], and it has already been applied to motion planning problems [2], [9].

The paper is organized as follows. In Section II, the problem of the MJ trajectory planning is formalized. Section III introduces interval analysis, while the MJ algorithm with convergence analysis (*Theorem 1*) is exposed in Section IV. An illustrative example, regarding a six-degrees-of-freedom manipulator, is described in Section V and comparisons with the trigonometric spline method of Simon and Isik [5] are exposed. In

Manuscript received January 22, 1998; revised July 15, 1999. Abstract published on the Internet August 20, 1999. This work was supported in part by MURST scientific research funds and by ASI (Italian Space Agency).

A. Piazzi is with the Dipartimento di Ingegneria dell'Informazione, University of Parma, I-43100 Parma, Italy.

A. Visioli is with the Dipartimento di Elettronica per l'Automazione, University of Brescia, I-25123 Brescia, Italy.

Publisher Item Identifier S 0278-0046(00)01040-6.

particular, both planning methods are checked against a full simulation of the PUMA 560 robot dynamics. Concluding remarks are reported in the last section.

II. MJ Trajectory Planning

Given s interspaced points of the tool frame Cartesian path, these are transformed, by the application of the inverse kinematics into s sets of joint displacements (knots) $\{d_{1i}, \cdots, d_{mi}\}$ $(i = 1, \cdots, s)$ where d_{ki} is the displacement of the kth joint of the m-joint robot at knot i. Hence, the sequences of displacements at a given joint k are given by $\{d_{k1}, \cdots, d_{ks}\}$ $(k = 1, \cdots, m)$. Each of these sequences will be exactly interpolated by cubic polynomials which have to assure an overall continuity of position (displacement), velocity and acceleration. In this context, assigning the velocity and acceleration for the first and last knot implies that two extra knots have to be inserted in second and penultimate positions [1]. So, for each kth joint we describe the displacement sequence of the knots as follows: $q_{k0}, \cdots, q_{kn}$ with $n := s + 1$, $q_{k0} := d_{k1}$, $q_{ki} := d_{ki}$ $(i = 2, \cdots, s-1)$, $q_{kn} := d_{ks}$, q_{k1}, and $q_{k,n-1}$ are free displacement parameters. Joint velocity and acceleration at the ith knot are, respectively, denoted by v_{ki} and a_{ki}. Velocities v_{k0}, v_{kn} and accelerations a_{k0}, a_{kn} are assigned data of the problem. Denote by h_i the elapsed time necessary for the ith spline $Q_{ki}(t)$ to connect knot $i-1$ to knot i for $t \in [0,\ h_i]$ (note that h_i is independent of the considered joint k). A convenient parameterization of spline $Q_{ki}(t)$ naturally incorporates continuity of positions and velocities [4]

$$\begin{aligned} Q_{ki}(t) = {} & q_{k,i-1} + v_{k,i-1}t + \left[\frac{3}{h_i^2}(q_{ki} - q_{k,i-1}) \right. \\ & \left. - \frac{1}{h_i}(v_{k,i} + 2v_{k,i-1})\right] t^2 \\ & + \left[-\frac{2}{h_i^3}(q_{ki} - q_{k,i-1}) + \frac{1}{h_i^2}(v_{ki} + v_{k,i-1})\right] t^3 \\ & t \in [0,\ h_i]. \end{aligned} \tag{1}$$

The unknown parameters in each spline can be determined imposing the continuity of acceleration, i.e., solving the following system of $n+1$ linear equations $(k = 1, \cdots, m)$:

$$\begin{aligned} \ddot{Q}_{k1}(0) &= a_{k0} \\ \ddot{Q}_{k1}(h_1) &= \ddot{Q}_{k2}(0) \\ &\vdots \\ \ddot{Q}_{k,n-1}(h_{n-1}) &= \ddot{Q}_{kn}(0) \\ \ddot{Q}_{kn}(h_n) &= a_{kn}. \end{aligned} \tag{2}$$

It can be easily seen that, once the h's have been fixed, the above system (2) admits a unique solution for any assigned data set [1]. The total traveling time required to perform the robot task is evidently $\sum_{i=1}^{n} h_i$ and can be fixed based on the velocity and acceleration constraints or the time required to do a certain task in an automated cell. The jerk is evidently constant on each spline and its expression is given by

$$\dddot{Q}_{ki}(t) = -\frac{12}{h_i^3}(q_{ki} - q_{k,i-1}) + \frac{6}{h_i^2}(v_{ki} + v_{k,i-1}). \tag{3}$$

Denoting by $j_{ki}(\boldsymbol{h})$ the jerk $\dddot{Q}_{ki}(t)$ of the ith spline at joint k, where $\boldsymbol{h} = (h_1, \cdots, h_n)$ is the vector of the spline times, the optimal trajectory planning problem with MJ criterion can be posed as the following constrained minimax optimization problem:

$$\min_{\boldsymbol{h} \in \mathbb{R}^{+n}} \max\{|j_{ki}(\boldsymbol{h})|: i = 1, \cdots, n;\ k = 1, \cdots, m\} \tag{4}$$

subject to

$$\sum_{i=1}^{n} h_i = T. \tag{5}$$

Solving the MJ trajectory problem is to find a global minimizer $\boldsymbol{h}^* = (h_1^*, \cdots, h_n^*)$ corresponding to the global minimum $j^* = \max\{|j_{ki}(\boldsymbol{h}^*)|: i = 1, \cdots, n;\ k = 1, \cdots, m\}$. The data set of the above optimization problem (4), (5) can be displayed as follows $(k = 1, \cdots, m)$:

q_{k0}, v_{k0}, a_{k0}	(displacement, velocity, and acceleration at the first knot)
$q_{k2}, \cdots, q_{k,n-2}$	(displacements at the intermediate knots)
q_{kn}, v_{kn}, a_{kn}	(displacement, velocity, and acceleration at the last knot)
T	(total traveling time).

(6)

The global optimization problem (4) and (5) always admits a solution for any generically chosen data set (6) because corresponding functions $j_{ki}(\boldsymbol{h})$ are well-defined rational functions without singularities over $\mathbb{R}^{+n}$. In this context, we make the following assumption.

Assumption 1: It is known a sufficiently small $\nu > 0$ such that

$$\mathbb{R}_{\nu}^{+n} := \{\boldsymbol{h} \in \mathbb{R}^n: \quad h_i \geq \nu, i = 1, \cdots, n\} \tag{7}$$

contains all global minimizers of problem (4) and (5).

Remark 1: From a technical viewpoint, it is not an issue to choose a proper ν. Indeed, in order to make a sound implementation of a given spline trajectory, all its spline times h_i must be significantly greater than the sampling time associated to the actual path update rate [4].

Remark 2: The minimax optimization problem (4) and (5) can also be posed by weighting the relative importance of the jerk in different joints:

$$\min_{\boldsymbol{h} \in \mathbb{R}^{+n}} \max\{w_k |j_{ki}(\boldsymbol{h})|: \quad i = 1, \cdots, n;\ k = 1, \cdots, m\} \tag{8}$$

subject to $\sum_{i=1}^{n} h_i = T$, where $w_k \geq 1$, $k = 1, \cdots, m$ are user-chosen weights. This problem formulation may be useful in practical trajectory planning in order to accomodate actuator specifications especially when many joints are involved (e.g., redundant manipulators). For simplicity, in our subsequent development we still consider problem (4) and (5) because reformulation (8) is trivial and does not change the mathematical peculiarities of the original minimax problem.

III. Interval Analysis

Let $I := \{[a, b]\colon a, b \in \mathbb{R}, a \leq b\}$ denote the set of real intervals. The interval arithmetic defined over I is given by

$$[a, b] + [c, d] := [a + c, b + d] \tag{9}$$

$$[a, b] - [c, d] := [a - d, b - c] \tag{10}$$

$$[a, b] \cdot [c, d] := [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}] \tag{11}$$

$$[a, b]/[c, d] := [a, b] \cdot [1/d, 1/c] \quad \text{if } 0 \notin [c, d]. \tag{12}$$

Regarding the algebraic structure, the mathematical systems $\{I, +\}$ and $\{I, \cdot\}$ are commutative semigroups containing the "units" [0, 0] and [1, 1], respectively. The absolute value of an interval $[a, b]$, i.e., $|[a, b]| := \{|x|\colon x \in [a, b]\}$ can be determined as

$$|[a, b]| = \begin{cases} [a, b], & \text{if } a \geq 0 \\ [-b, -a], & \text{if } b \leq 0 \\ [0, \max\{|a|, |b|\}], & \text{otherwise.} \end{cases} \tag{13}$$

Let $X \in I^n$ be a multidimensional interval (box) of $\mathbb{R}^n$, i.e., $X = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+]$. The midpoint of X be designated by mid(X). Moreover, denote with $I(X) := \{Y \in I^n\colon Y \subseteq X\}$ and $w(X) := \max_{i=1, \cdots, n}\{x_i^+ - x_i^-\}$, respectively, the set of all subboxes of X and the "width," or the measure, of X. Consider $f\colon X \to \mathbb{R}$, a real scalar function $f(x)$ defined over the box X; for any $Y \in I(X)$ define by $\hat{f}(Y) := \{f(x)\colon x \in Y\}$ the range of f over Y.

Definition 1: A function $F\colon I(X) \to I$ is an *inclusion function* of $f(x)$ if

1) $\hat{f}(Y) \subseteq F(Y)\ \forall Y \in I(X)$;
2) $\lim_{w(Y)\to 0} w(F(Y)) = 0$.

The concept of inclusion function is central in the development of interval analysis and the use of it characterizes the so-called "interval algorithms," especially in deterministic global optimization [8]. There are a large variety of inclusion functions: natural interval extensions, standard centered forms, meanvalue forms, Taylor forms, Cornelius-Lohner forms, etc. The simplest inclusion function is the natural interval extension, which can be obtained by substituting, in the expression of a given $f(x)$, the requested box argument Y and then by performing the necessary interval computations. For example, given $f(x) = (1 + 4x^2)\cos(3x)$ and rewriting it as $f(x) = (1 + 4|x|^2)\cos(3x)$, its natural interval extension $F(Y)$ with $Y = [-1, 3]$ can be computed as follows: $F(Y) = (1 + 4|[-1, 3]|^2)\cos(3[-1, 3]) = (1 + 4[0, 3]^2)\cos([-3, 9]) = (1 + 4[0, 9])[-1, 1] = (1 + [0, 36])[-1, 1] = [1, 37][-1, 1] = [-37, 37]$.

IV. MJ Algorithm

Introduce the $(n-1)$-dimensional interval

$$\mathrm{A}_\nu := [\nu, T - (n-1)\nu]^{n-1} \subseteq \mathbb{R}^{n-1}$$

and denote by $\tilde{\boldsymbol{h}} := (h_1, \cdots, h_{n-1})$ a vector point in $\mathbb{R}^{n-1}$. Taking into account *Assumption 1*, it can be easily shown that problem (4) and (5) is equivalent to

$$\min_{\tilde{\boldsymbol{h}} \in \mathrm{A}_\nu} \max\{|j_{ki}(h_1, \cdots, h_{n-1}, T - h_1 - \cdots - h_{n-1})|\colon i = 1, \cdots, n;\ k = 1, \cdots, m\} \tag{14}$$

subject to

$$T - h_1 - \cdots - h_{n-1} \geq \nu. \tag{15}$$

Now, define the following "pyramid" domains in $\mathbb{R}^{+(n-1)}$

$$\mathcal{T}_\nu := \{\tilde{\boldsymbol{h}} \in \mathrm{A}_\nu\colon T - h_1 - \cdots - h_{n-1} \geq \nu\} \tag{16}$$

$$\mathcal{T}_{\nu/p_t} := \{\tilde{\boldsymbol{h}} \in \mathrm{A}_\nu\colon T - h_1 - \cdots h_{n-1} \geq \nu/p_t\} \tag{17}$$

where p_t is a positive integer and $\tilde{j}_{ki}(\tilde{\boldsymbol{h}})$ be given by

$$\tilde{j}_{ki}(\tilde{\boldsymbol{h}}) := j_{ki}\left(h_1, \cdots, h_{n-1}, T - \sum_{i=1}^{n-1} h_i\right). \tag{18}$$

Hence, minimax problem (14) and (15) can be succinctly rewritten as

$$\min_{\tilde{\boldsymbol{h}} \in \mathcal{T}_\nu} \max\{|\tilde{j}_{ki}(\tilde{\boldsymbol{h}})|\colon i = 1, \cdots, n;\ k = 1, \cdots, m\}. \tag{19}$$

Remark 3: The proposed reformulation (19) has the merit of reducing the dimensionality of the search domain from n to $n-1$. A global minimizer $\tilde{\boldsymbol{h}}^* \in \mathcal{T}_\nu$ of problem (19) is obviously related to a global minimizer $\boldsymbol{h}^* \in \mathbb{R}_\nu^{+n}$ of problem (4) and (5) by equation $\boldsymbol{h}^* = (\tilde{\boldsymbol{h}}^*, T - h_1^* - \cdots - h_{n-1}^*)$.

The interval algorithm to be devised in the sequel relies on the following result.

Property 1: Let $\tilde{\mathcal{T}}_\nu$ any closed set of $\mathbb{R}^{n-1}$ satisfying the inclusions: $\mathcal{T}_\nu \subseteq \tilde{\mathcal{T}}_\nu \subseteq \mathcal{T}_{\nu/p_t}$. On *Assumption 1*, the constrained minimax optimization problem (4) and (5) is equivalent to

$$\min_{\tilde{\boldsymbol{h}} \in \tilde{\mathcal{T}}_\nu} \max\{|\tilde{j}_{ki}(\tilde{\boldsymbol{h}})|\colon i = 1, \cdots, n;\ k = 1, \cdots, m\}. \tag{20}$$

Proof: By definitions (16) and (17), evidently, $\mathcal{T}_\nu \subseteq \mathcal{T}_{\nu/p_t}$. On the other hand, *Assumption 1* implies that any global minimizer $\boldsymbol{h}^*$ of problem (4) and (5) must have $h_n^* \geq \nu$. Considering that all the points in $\mathcal{T}_{\nu/p_t} - \mathcal{T}_\nu$ satisfy the inequality $T - h_1 - \cdots - h_{n-1} < \nu$, it follows that no global minimizers can be found in $\mathcal{T}_{\nu/p_t} - \mathcal{T}_\nu$. Hence, the search domain of the global optimization problem (19) can be enlarged from $\mathcal{T}_\nu$ to any closed set $\tilde{\mathcal{T}}_\nu \subseteq \mathcal{T}_{\nu/p_t}$ without modifying the solution set of (19). □

The function to be minimized in (20) can be formally introduced as the following nonsmooth function $f(\tilde{\boldsymbol{h}})$:

$$f(\tilde{\boldsymbol{h}}) := \max\{|\tilde{j}_{ki}(\tilde{\boldsymbol{h}})|\colon i = 1, \cdots, n;\ k = 1, \cdots, m\}. \tag{21}$$

The concept of inclusion function, introduced in Section III, can be easily extended to nonsmooth maxfunctions such as $f(\tilde{\boldsymbol{h}})$. Specifically, consider a box $\mathrm{B} \subseteq \mathrm{A}_\nu$.

The value of an inclusion function of $f(\tilde{\boldsymbol{h}})$ computed on B is denoted by

$$F(\mathrm{B}) := [F(\mathrm{B})^-, F(\mathrm{B})^+]. \tag{22}$$

Introduce also $[l_{ki}, u_{ki}] := |\tilde{J}_{ki}(\mathrm{B})|$ where $\tilde{J}_{ki}(\mathrm{B})$ indicates a (standard) inclusion function of $\tilde{j}_{ki}(\tilde{\boldsymbol{h}})$ computed on B. Then,

$$F(\mathrm{B})^- := \max\{l_{ki}\colon i = 1, \cdots, n;\ k = 1, \cdots, m\}, \tag{23}$$

$$F(\mathrm{B})^+ := \max\{u_{ki}\colon i = 1, \cdots, n;\ k = 1, \cdots, m\}; \tag{24}$$

and the obtained inclusion function for the maxfunction $f(\tilde{\boldsymbol{h}})$ still satisfies statements 1) and 2) of *Definition 1*.

Essentially, the first part of the following MJ algorithm aims to obtain all the boxes $\mathrm{B} \subseteq \mathrm{A}_\nu$ whose set union is a domain satisfying the inclusion of *Property 1*. Then, these boxes have to pass to the second part of the algorithm that is directly derived from the basic Hansen's algorithm [8, p. 111].

Input of the MJ Algorithm: the data set (6), the box A_ν, the threshold integer p_t, and the precision parameter ε.

Output of the MJ Algorithm: j^-, j^+: lower and upper bound of j^* satisfying $j^+ - j^- \leq \varepsilon$; the approximate global minimizer $\boldsymbol{h}^a := (\tilde{\boldsymbol{h}}^a, T - \sum_{i=1}^{n-1} h_i^a)$ which satisfies $j^+ = \max\{|j_{ki}(\boldsymbol{h}^a)|\colon i = 1, \cdots, n;\ k = 1, \cdots, m\}$.

```
The MJ Algorithm
First part
  1. Choose a vector point h̃^a ∈ T_ν and set
     j^+ := f(h̃^a).
  2. Initialize list L := {(A_ν, −∞)}.
  3. Denote the first pair of L by (Y, y).
  4. If y > −∞ then go to 10 (terminate first
     part).
  5. Bisect Y on its maximum dimension,
     thus obtaining boxes V1 and V2 and
     denote Vi := [h_i1^−, h_i1^+] × ··· × [h_i,n−1^−, h_i,n−1^+](i
     = 1, 2).
  6. Remove (Y, y) from the top of the list L.
  7. For i = 1, 2 do
   (a) If T − Σ_{l=1}^{n−1} h_il^− < ν then go to (f) (box Vi
       is rejected).
   (b) If T − Σ_{l=1}^{n−1} h_il^+ < ν/p_t then insert (Vi, −∞)
       at the top of the list L and go to (f)
       (box Vi can neither be rejected nor
       used for inclusion function
       evaluation).
   (c) Set v := F(Vi)^−.
   (d) If v > j^+ then go to (f) (box Vi is
       rejected).
   (e) Insert (Vi, v) at the end of the list L.
   (f) End of i-loop.
  8. Go to 3.
Second part
  9. Denote the first pair of the L by
     (Y, y).
 10. Set c := mid(Y) and s := f(c).
 11. If s < j^+ then set j^+ := s and h̃^a := c.
 12. Bisect Y on its maximum dimension,
     thus obtaining boxes V1 and V2 such
     that Y = V1 ∪ V2.
 13. Remove (Y, y) from the top of L.
 14. For i = 1, 2
   (a) Set v := F(Vi)^−.
   (b) Enter the pair (Vi, v) at the end of L.
   (c) End of i-loop.
 15. Set j^− := {the minimum of the second
     elements of all pairs of L} (the
     minimum of all the lower bounds
     associated to the boxes of the list L).
 16. Discard from L all pairs (Z, z) that
     satisfy j^+ < z (midpoint test).
 17. If j^+ − j^− > ε then go to 9.
 18. End.
```

TABLE I
DATA KNOTS IN THE ILLUSTRATIVE EXAMPLE

joint	knot (degrees)					
number	1	2	3	4	5	6
1	-10		60	20		55
2	20		50	120		35
3	15	extra	100	-10	extra	30
4	150	knot	100	40	knot	10
5	30		110	90		70
6	120		60	100		25

TABLE II
RESULTING OPTIMAL SPLINE TIMES

spline	1	2	3	4	5
time (sec.)	0.96	2.40	2.66	2.15	0.93

The following theorem relies on *Property 1* and on the generalization for maxfunctions of Hansen's algorithm.

Theorem 1: For any positive values of ε and p_t, the MJ algorithm converges with certainty and solves the optimization problem (4) and (5) in accordance with the given output definition.

Proof: Denote by $\mathcal{H}^*$ the set of global minimizers contained in A_ν. The first part of the MJ algorithm, based on *Property 1*, is devoted to the algorithmic determination of the domains $\tilde{\mathcal{T}}_\nu^*$ and $\tilde{\mathcal{T}}_\nu$ satisfying the following inclusions:

$$\tilde{\mathcal{T}}_\nu^* \subseteq \tilde{\mathcal{T}}_\nu \tag{25}$$

$$\mathcal{T}_\nu \subseteq \tilde{\mathcal{T}}_\nu \subseteq \mathcal{T}_{\nu/p_t} \tag{26}$$

$$\mathcal{H}^* \subseteq \tilde{\mathcal{T}}_\nu^* \tag{27}$$

and at completion of the first part, the list $\mathcal{L}$ contains all the boxes whose union is $\tilde{\mathcal{T}}_\nu^*$. Indeed, the starting box A_ν is exhaustively processed by a depth-first subdivision strategy for which

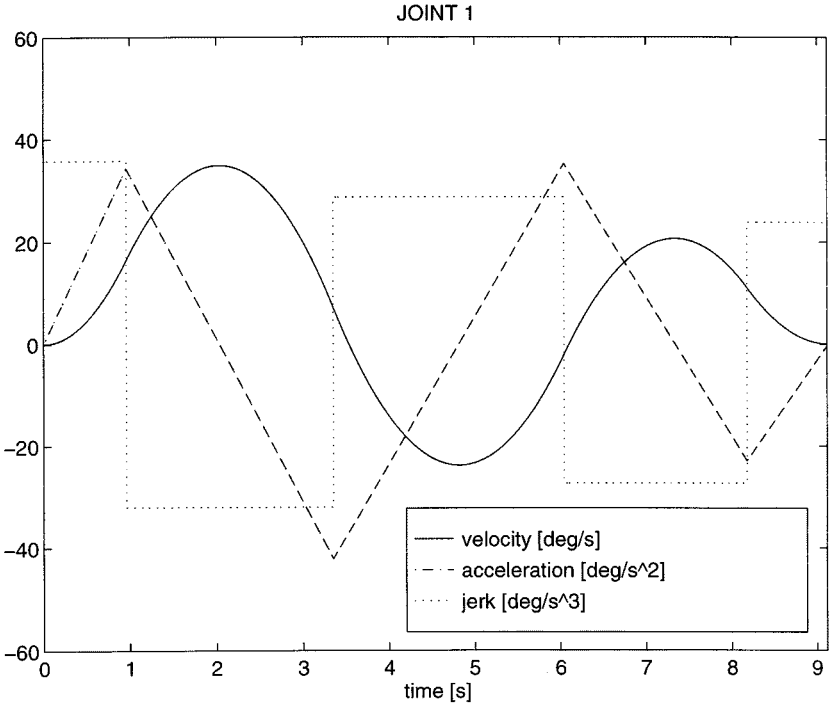

Fig. 1. The resulting trajectory of joint 1 for the example.

1) boxes completely contained in $A_\nu - \mathcal{T}_\nu$ are rejected [see 7(a)];
2) boxes not completely contained in $\mathcal{T}_{\nu/p_t}$ are not used for inclusion function evaluation and put at the top of list $\mathcal{L}$ for subsequent bisection;
3) boxes completely contained in $\mathcal{T}_{\nu/p_t}$ are used for the inclusion function evaluation of step 7(c) and saved at the end of the list $\mathcal{L}$. They are not further bisected.

If directive 7(d) is omitted, then, at the end of the first part of the algorithm, we obtain the domain $\tilde{\mathcal{T}}_\nu$, satisfying the inclusions (26), as the union of all the boxes of list $\mathcal{L}$. By virtue of statement 1) in *Definition 1*, the presence of instruction 7(d) causes the rejection of boxes of $\tilde{\mathcal{T}}_\nu$ which surely do not contain global minimizers. Hence, at termination of the first part, we actually obtain $\tilde{\mathcal{T}}_\nu^*$ satisfying (25) and (27).

The second part of the algorithm is devoted to determining, within the prescribed precision ε, a global minimum of $f(\tilde{\boldsymbol{h}})$ over $\tilde{\mathcal{T}}_\nu^*$ (or $\tilde{\mathcal{T}}_\nu$ that is exactly the same problem). It is basically a branch-and-bound procedure where the branching is performed by bisecting a box which has been longest in $\mathcal{L}$ and the bounding is made via inclusion function evaluations. At the ith iteration, the list $\mathcal{L}$ is described by

$$\mathcal{L}_i := \{(Z_{i1}, z_{i1}), \cdots, (Z_{i,n_i}, z_{i,n_i})\}$$

where n_i is the list length. By virtue of steps 10, 11, and 15, we have

$$j_i^- \le j^* \le j_i^+ \qquad \text{for all } i \in \mathbb{N} \tag{28}$$

with j_i^-, j_i^+ denoting the values of variables j^-, j^+ at the closing of the ith iteration. Moreover,

$$\bigcup_{k=1}^{n_i} Z_{i,k} \supseteq \mathcal{H}^* \qquad \text{for all } i \in \mathbb{N} \tag{29}$$

because the midpoint test at step 16 only discards boxes not containing global minimizers. The bisection strategy and the list ordering by age issued by instructions 9, 12, and 14(b) imply that, for any given $\xi > 0$, there exists a (finite) $i^* \in \mathbb{N}$ for which

$$w(Z_{i^*,k}) < \xi, \qquad k = 1, \cdots, n_{i^*}. \tag{30}$$

Define by f_k^* the global minimum of $f(\tilde{\boldsymbol{h}})$ over $Z_{i^*,k}$. Considering the properties 1) and 2) of any inclusion function (*Definition 1*) and that, moreover, the limit in 2) holds uniformly whenever boxes span in a bounded domain, it follows that

$$f_k^* - z_{i^*,k} < \varepsilon/2, \qquad k = 1, \cdots, n_{i^*} \tag{31}$$

for a sufficiently small ξ [see steps 14(a) and (b)]. Taking into account the continuity of $f(\tilde{\boldsymbol{h}})$ over the bounded domain $\tilde{\mathcal{T}}_\nu^*$, it holds

$$f(\tilde{\boldsymbol{h}}) - f_k^* < \varepsilon/2 \quad \forall \tilde{\boldsymbol{h}} \in Z_{i^*,k}, \qquad k = 1, \cdots, n_{i^*} \tag{32}$$

for a sufficiently small ξ. Combine inequalities (31) and (32) to obtain

$$f(\tilde{\boldsymbol{h}}) - z_{i^*,k} < \varepsilon \quad \forall \tilde{\boldsymbol{h}} \in Z_{i^*,k}, \qquad k = 1, \cdots, n_{i^*}. \tag{33}$$

As $j_{i^*}^- = \min\{z_{i^*,k}\colon k = 1, \cdots, n_{i^*}\}$ (cf. step 15) and defining k^* to get $j_{i^*}^- = z_{i^*,k^*}$ it follows that

$$f(\tilde{\boldsymbol{h}}) - j_{i^*}^- < \varepsilon \quad \forall \tilde{\boldsymbol{h}} \in Z_{i^*,k^*}$$

that implies

$$\max_{\tilde{\boldsymbol{h}} \in Z_{i^*,k^*}} \{f(\tilde{\boldsymbol{h}})\} < \varepsilon + j_{i^*}^-. \tag{34}$$

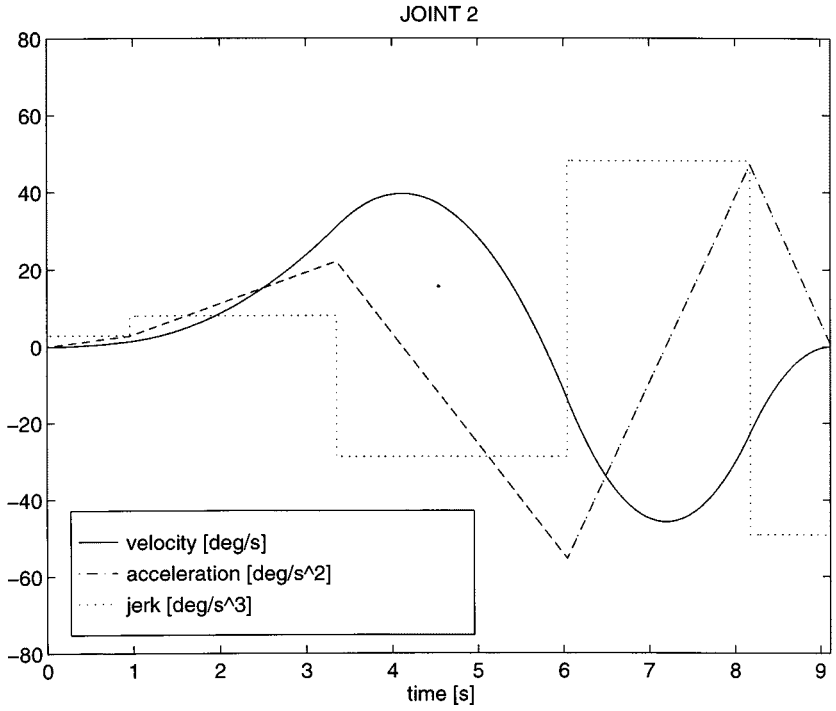

Fig. 2. The resulting trajectory of joint 2 for the example.

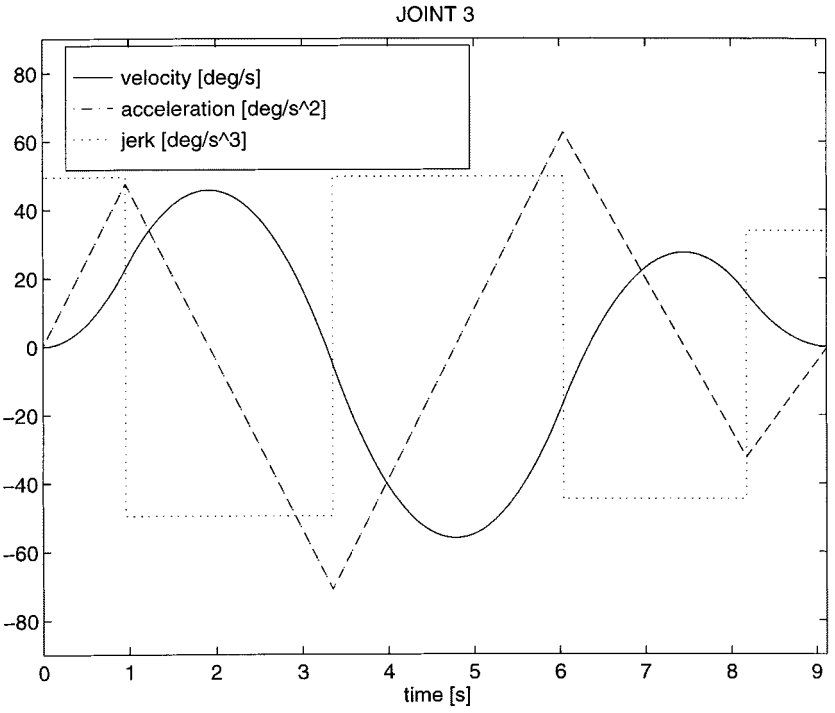

Fig. 3. The resulting trajectory of joint 3 for the example.

By virtue of steps 10–12, we have

$$j_{i^*}^{+} \leq \max_{\tilde{\boldsymbol{h}} \in \mathrm{Z}_{i^*,k}} \{f(\tilde{\boldsymbol{h}})\}, \qquad k = 1, \cdots, n_{i^*}. \tag{35}$$

This arises from the function evaluation at step 10 made with mid(Y) which is a point belonging to both V_1 and V_2 (on the boundary). Eventually, from inequalities (34) and (35) we infer $j_{i^*}^{+} - j_{i^*}^{-} < \varepsilon$. Hence, in a finite number of iterations, the algorithm necessarily halts at step 17 with the required global minimizer $\tilde{\boldsymbol{h}}^a$. □

It is worth stressing that the convergence proof does not depend on the number of global minimum points which is not known in general.

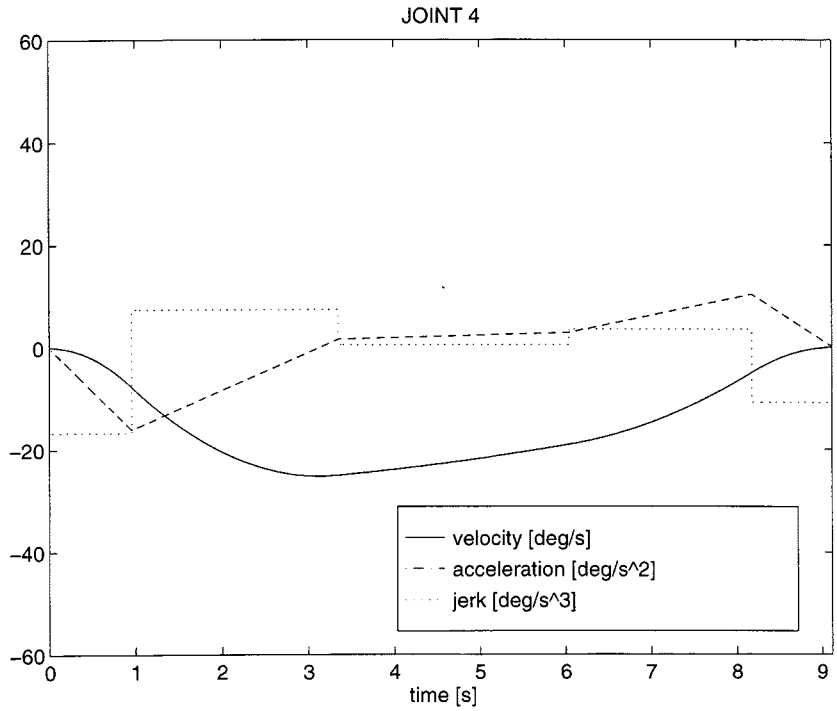

Fig. 4. The resulting trajectory of joint 4 for the example.

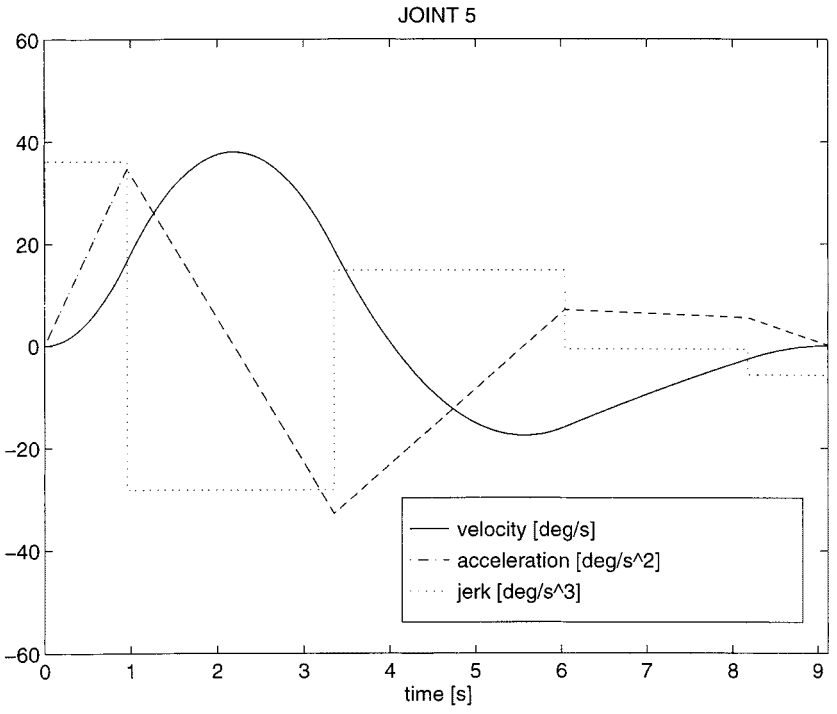

Fig. 5. The resulting trajectory of joint 5 for the example.

V. An Illustrative Example

The previous interval algorithm has been implemented in C++ language exploiting the PROFIL/BIAS libraries made by Knüppel [10]. As an illustrative example, we considered the case of a trajectory composed of five splines of a six-degrees-of-freedom robot manipulator. The joint displacements are indicated in Table I. We set the total traveling time T to 9.1 s and the other parameters ε and p_t to 0.01 and 10, respectively, while ν has been fixed to 0.1. Initial and final velocities and

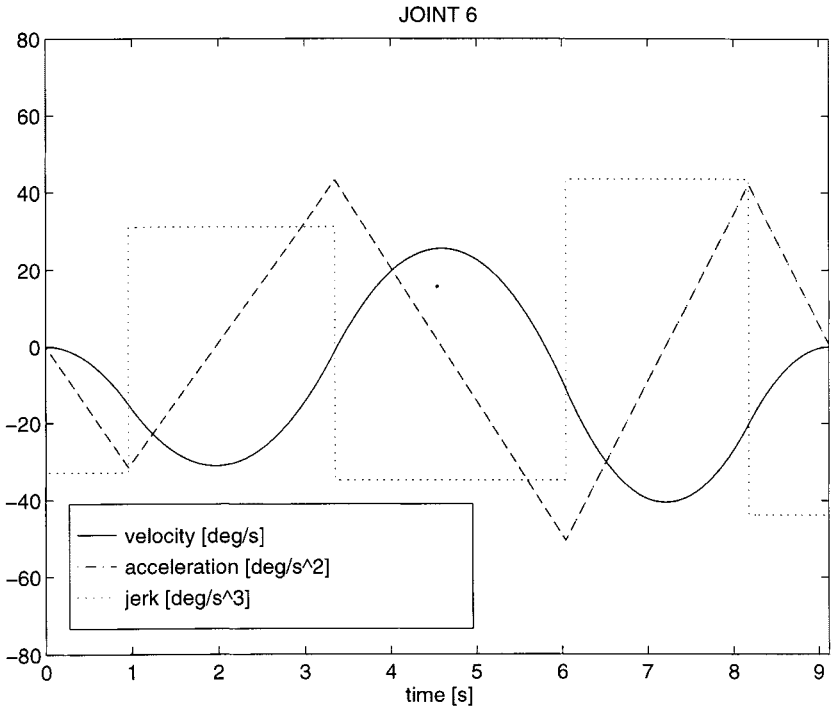

Fig. 6. The resulting trajectory of joint 6 for the example.

TABLE III
COMPARISON BETWEEN MJ ALGORITHM (CUBIC SPLINES) AND SIMON AND ISIK'S PROCEDURE (TRIGONOMETRIC SPLINES)

joint	max jerk (deg/s)		max torque (Nm)		max torque variation (Nm/s)	
number	cubic	trigonometric	cubic	trigonometric	cubic	trigonometric
1	35.74	54.79	2.75	2.44	2.81	8.93
2	49.35	59.22	32.42	34.29	34.06	109.8
3	49.35	80.80	8.93	8.99	4.40	16.65
4	16.56	25.99	0.0472	0.0595	0.0549	0.245
5	36.05	49.02	0.104	0.107	0.110	0.441
6	43.73	61.40	0.172	0.148	0.155	0.631

accelerations are set to zero, as it is in the ordinary cases. In constructing the inclusion function of the maxfunction $f(\tilde{\boldsymbol{h}})$, we adopted the Baumann meanvalue form [8, p. 42]. The resulting global minimax jerk j^+ is 49.35°/s^3 and the optimal spline times are reported in Table II. The plots of velocities, accelerations, and jerks of the six joints are reported in Figs. 1–6. It can be noticed that the maximum value of the jerk is attained by joint 2 in the penultimate and last spline and by joint 3 in the first, second, and third spline of the trajectory.

The same example can be approached with the MJ planning procedure of Simon and Isik [5]. They use trigonometric splines that are continuous functions up to third derivative with free velocity, acceleration, and jerk values at the endpoints of the spline segments. By solving a finite set of algebraic linear systems, these free values are used to minimize the integral of the squared jerk over the total motion time which is equally divided into three spline times of 3.033 s. The trajectories obtained with the MJ algorithm (cubic splines) and with the Simon and Isik's procedure have been compared. In particular, using the Matlab Robotics Toolbox made by Corke [11], both trajectories have been used to compute the joint torque and the derivative of the joint torque for the PUMA 560 robot dynamics. For each joint, Table III reports the maximum values of jerk, torque, and torque variation (derivative). It appears that the overall maximum jerk for Simon and Isik's procedure is 80.80°/s^3 which is much greater than the corresponding 49.35°/s^3 obtained with the MJ algorithm. Moreover, for each joint, the maximum jerk is smaller than the maximum jerk associated with the trigonometric splines. The same property holds comparing the maximum torque variations in the two last columns of Table III. In particular, the maximum torque variations for the trigonometric splines are 3.17–4.46 times greater than those for the cubic splines. By examining the maximum torques, we reveal that, for joints 1 and 6, the cubic spline maximum torque is

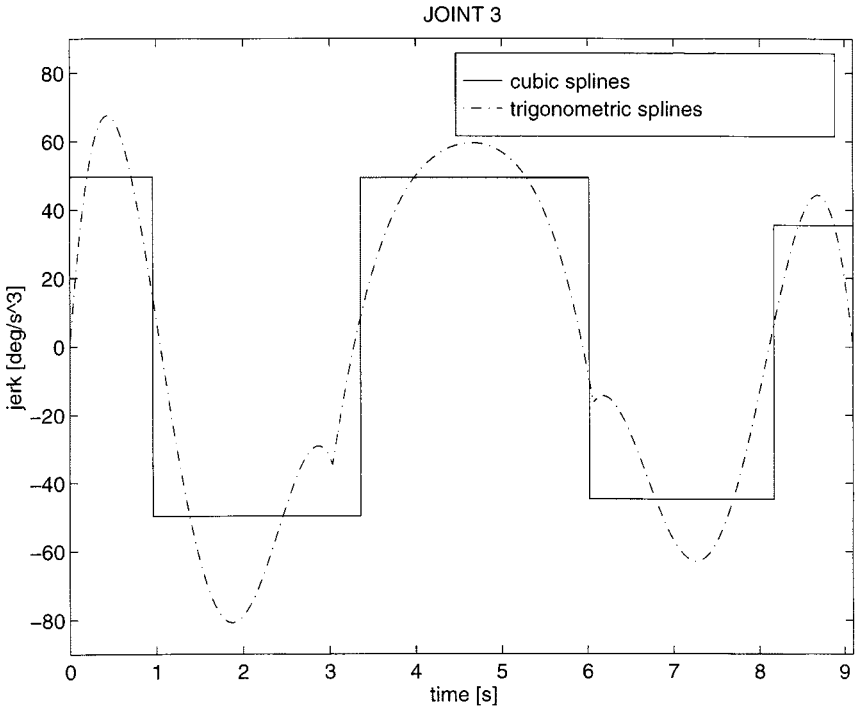

Fig. 7. Jerk functions for joint 3 with MJ algorithm and Simon and Isik's procedure.

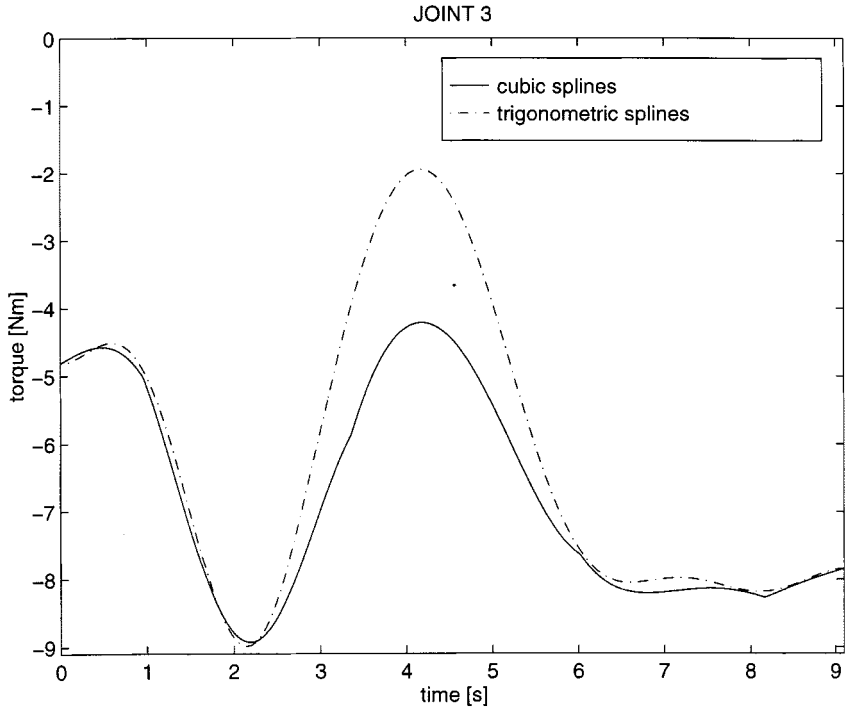

Fig. 8. Torque functions for joint 3 with MJ algorithm and Simon and Isik's procedure (PUMA 560 dynamics).

greater than the trigonometric one, whereas for the other joints 2–5 the contrary holds. For joint 3, that reaches the maximum jerk value for both the cubic and trigonometric splines, Figs. 7 and 8 show the jerk and torque functions planned with the MJ algorithm and Simon and Isik's procedure.

Remark 4: Minimizing the maximum jerk in joint space (i.e., minimizing the jerk uniformly over the total motion time) has a beneficial effect in reducing the actuator and mechanical strain. This is due to the fact that, considering the manipulator dynamics, the derivative of vector torque depends on the typically dominant term of the inertia matrix multiplied by the vector joint jerk [4]. Therefore, minimaximizing the joint jerk, which is the aim of the MJ algorithm, helps in reducing the torque variations which in turn reduces the actuator and mechanical strain.

Remark 5: For the presented example, the trajectory planning based on the MJ algorithm appears superior to the one of Simon and Isik from the viewpoint of minimizing actuator and mechanical strain and joint wear. On the other hand, the use of trigonometric splines has the advantage of permitting to easily alter the planned trajectory in mid-course if necessary in order, for example, to avoid unexpected obstacles in real-time environments. Hence, the MJ algorithm planner is particularly suitable to be applied in automated robotic cells where the total motion time is fixed by the cell scheduler which ensures that velocity and acceleration constraints be satisfied.

VI. Conclusions

In this paper, the global MJ joint-space trajectory planning of an m-joint robot manipulator has been presented. Cubic splines have been employed because their simplicity does not preclude the possibility to exactly interpolate given knots, assuring the continuity of velocities and accelerations; moreover, they do not exhibit large overshoot displacements. The problem can be reformulated as a global constrained minimax optimization problem for which a solution can be obtained through the presented MJ algorithm. This approach can be successfully exploited in a variety of real automation settings. Specifically, interesting applications can be found in the off-line programming of robot manipulators in automated plants with scheduled time constraints and in operating aerospace environments. In fact, once the total traveling time of the motion has been fixed, minimizing the jerk is desirable because it reduces the actuator and mechanical strain and the joint wear. This implies that trajectory tracking performances by the robot control system are improved and there is also a positive effect on expanding the robot lifespan.

References

[1] C.-S. Lin, P.-R. Chang, and J. Y. S. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Trans. Automat. Contr.*, vol. AC-23, pp. 1066–1074, Dec. 1983.

[2] A. Piazzi and A. Visioli, "Global minimum-time trajectory planning of mechanical manipulators using interval analysis," *Int. J. Contr.*, vol. 71, no. 4, pp. 631–652, Nov. 1998.

[3] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, Philadelphia, PA, 1988, pp. 364–369.

[4] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed. New York: Addison-Wesley, 1989.

[5] D. Simon and C. Isik, "A trigonometric trajectory generator for robotic arms," *Int. J. Contr.*, vol. 57, no. 3, pp. 505–517, 1993.

[6] D. Simon, "The application of neural networks to optimal robot trajectory planning," *Robot. Auton. Syst.*, vol. 11, no. 1, pp. 23–34, 1993.

[7] R. E. Moore, *Methods and Applications of Interval Analysis*. Philadelphia, PA: SIAM Press, 1979.

[8] H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization*. Chichester, U.K.: Ellis Harwood, 1988.

[9] L. Jaulin and E. Walter, "Guaranteed tuning, with application to robust control and motion planning," *Automatica*, vol. 32, no. 8, pp. 1217–1221, 1996.

[10] O. Knüppel, "PROFIL—Programer's runtime optimized fast interval library," Technische Univ. Hamburg-Harburg, Hamburg, Germany, Tech. Rep., July 1993.

[11] P. I. Corke, "A robotics toolbox for Matlab," *IEEE Robot. Automat. Mag.*, vol. 3, pp. 24–32, Mar. 1996.

Aurelio Piazzi (M'92) received the Laurea degree in nuclear engineering and the Ph.D. degree in system engineering from the University of Bologna, Bologna, Italy, in 1982 and 1987, respectively.

From 1990 to 1992, he was a Research Associate in System Theory, D.E.I.S., University of Bologna. Since November 1992, he has been an Associate Professor of Automatic Control, Dipartimento di Ingegneria dell'Informazione, University of Parma, Parma, Italy. His main research interests are in system and control theory and related engineering applications. His current research activities focus on noncausal regulation and on methods of global optimization applied to the analysis and design of robust control systems.

Dr. Piazzi is a member of the International Federation for Automatic Control and the Society for Industrial and Applied Mathematics.

Antonio Visioli (S'96–M'00) received the Laurea degree in electronic engineering from the University of Parma, Parma, Italy, and the Ph.D. degree in applied mechanics from the University of Brescia, Brescia, Italy, in 1995 and 1999, respectively.

Currently, he is an Assistant Professor of Automatic Control in the Department of Electronics for Automation, University of Brescia. His research interests include industrial robot control and trajectory planning, system-inversion-based control, and PID control.