Global minimum-time trajectory planning of mechanical manipulators using interval analysis

AURELIO PIAZZI† and ANTONIO VISIOLI‡

The paper addresses the global minimum-time trajectory planning of an m-joint mechanical manipulator. Using a joint space scheme with given intermediate points to be interpolated by piecewise cubic polynomials, a novel bisecting-plane algorithm is proposed to schedule the times between adjacent knots under velocity, acceleration, and jerk constraints. This algorithm, which is proved to be globally convergent with certainty within an arbitrary precision, uses an interval procedure (a subroutine adopting tools and ideas of interval analysis) in proving that a local minimum is actually a global one. A worked example for the six-joint case is exposed, and computational results of a C++ implementation are included.

1. Introduction

The mechanical manipulator is a highly non-linear dynamic system which has to be subjected to (non-linear) constraints in order to obtain a correct and safe operating functioning. For this reason, making an optimal control for a mechanical manipulator is a very difficult task. With various simplifying assumptions, only suboptimal solutions have appeared in the literature. An incomplete list of relevant works includes Shin and McKay (1986), Tan and Pot (1988), Bobrow (1988), Wu (1994), and a survey, with many additional references, was published by Shiller and Dubowsky (1988). An alternative simpler approach, where the manipulator dynamics is not taken into account, is to divide the problem into two steps: optimal trajectory planning, followed by feedback trajectory tracking. A sequence of path points is usually specified in terms of a desired position and orientation of the tool frame. Each of these path 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 suitably chosen for a specific manipulator application. The resulting joint trajectory forms the input to the robot’s feedback control system.

In this context, Lin et al. (1983) proposed cubic polynomial functions (splines) for a trajectory planning where the total travelling time is minimized under constraints on joint velocities, accelerations, and jerks. Cubic splines are widely used for interpolation since they prevent the large oscillations of the trajectory which can result with higher-order polynomials. The trajectory smoothness was assured by imposing, for the piecewise cubic polynomials, continuity of positions, velocities, and accelerations. They provide an efficient optimization solver based on the iterative generation of flexible polyhedrons in the search space, which can only determine
local solutions to the constrained optimization problem. The aim of achieving a true global solution in optimal trajectory planning was pursued by Simon (1993) who proposed, in a similar context, a stochastic optimization method based on a neural network, in order to obtain a minimum-jerk joint trajectory.

Using the spline joint space scheme of Lin et al. (1983), the authors (Piazzi and Visioli 1997) proposed a deterministic global optimization approach based on an interval algorithm to obtain a minimum-time spline trajectory subject to constraints on joint accelerations and jerks. Jerk constraints are primarily due to the fact that joint position errors increase when the jerk increases, as asserted by Kyriakopoulos and Saridis (1988), and to limit excessive wear on the robot and the excitation of resonances so that the robot life-span is extended (Craig 1989). Maintaining the aim of pursuing a globally optimal planning, this paper significantly extends our previous work by taking into account constraints on joint velocities, and proposes a novel algorithmic method of cutting planes in the search domain. From the viewpoint of computation time, this method leads to a significant improvement over the previous algorithm, since the interval procedure is applied to a few \((n - 1)\) dimensional feasibility problems instead of to a complete \(n\)-dimensional problem. Precisely, interval analysis is used to reveal if a local minimum, determined, for example, with a local
Mechanical manipulators planned by interval analysis

gradient procedure, is actually a global minimum. The techniques of interval analysis, which is a generalization of the 'standard' real analysis over the arithmetic of real intervals, have recently been applied to control theory and motion planning (Walter and Jaulin 1994, Jaulin and Walter 1996, Piazzi and Marro 1996, Malan et al. 1997). An introduction to these techniques can be found in the book by Moore (1979) (the founder of interval analysis), and applications to general purpose global optimization are covered by Ratschek and Rokne (1988) and Hansen (1992).

The paper is organized as follows. In section 2, the problem of the global minimum-time trajectory planning is formalized. It is explained how to transform the pertinent semi-infinite optimization problem (equation (3)) into a finite one (equations (6)–(11)) by means of an elementary quantifier elimination technique. Section 3 gives the relevant properties and the bisecting-plane (BP) algorithm, which is proved to be globally convergent with certainty (Theorem 1), whilst the interval procedure Unfeasibility to be used by the BP algorithm is described in section 4. An illustrative example is presented in section 5. Perspectives and concluding remarks are included in the last section.

1.1. Notation

We denote by \( \mathbb{R} \) and \( \mathbb{N} \) the set of reals and of positive integers, respectively. The closure and the interior of a set \( I \) are denoted by \( \text{clo}(I) \) and \( \text{int}(I) \). A homogeneous polynomial is denoted by a capital letter with its total degree placed between parentheses as a superscript, e.g. \( P^{(\alpha)}(h) \). \( ||h|| \) denotes the Euclidean norm of a vector \( h \) in \( \mathbb{R}^n \), whereas \( |h| \) denotes the vector whose components are the absolute values of the components of \( h \).

2. Minimum-time trajectory planning

Consider a mechanical manipulator with \( m \) revolute or prismatic joints and let \( q_i \), \( i = 1, \ldots, m \), denote the joint variables (as an example see the six-joint robot in figure 1). These are packed together into the joint variable vector \( q = (q_1, \ldots, q_m)^T \) that spans in \( Q \) the joint-space work envelope. Given \( s \) interspaced location points of the tool frame Cartesian path, these are mapped, by the application of the inverse kinematics, into \( s \) point vectors of \( Q \), formally denoted as \( q_i = (q_i, \ldots, q_{i,n})^T \), \( i \in \{0, 2, 3, \ldots, n-2, n\} \), \( n := s + 1 \). With this notation, \( q_k \) is the displacement of the \( k \)th joint at the \( i \)th knot. Hence, the sequences of displacements at a given joint \( k \) are given by \( \{q_k^0, q_k^2, q_k^3, \ldots, q_k^{n-2}, q_k^n\} \) \((k = 1, \ldots, m)\). Each of these sequences will be exactly interpolated by cubic polynomials which have to ensure an overall continuity of position (displacement), velocity, and acceleration (Lin et al. 1983). In this context, assigning the joint velocity and acceleration for the first and last location (at knot 0 and knot \( n \), respectively) implies that two free displacement vectors \( q^1 \) and \( q^{n-1} \) have to be inserted in the second and penultimate locations (at knot 1 and knot \( n-1 \), respectively). By \( \mathbf{v} := (v_1^i, \ldots, v_m^i)^T \) and \( \mathbf{a} := (a_1^i, \ldots, a_m^i)^T \), we denote the vectors of joint velocities and of accelerations at the \( i \)th knot, respectively. In defining the trajectory planning problem, velocity and acceleration vectors \( \mathbf{v}^0, \mathbf{v}^n \) and \( \mathbf{a}^0, \mathbf{a}^n \) must be assigned. Denote by \( h_i \) the elapsed time necessary for the \( i \)th spline \( q_k^i(t) \) to connect knot \( i-1 \) to knot \( i \) for \( t \in [0, h_i] \) at the joint \( k \) (note that \( h_i \) is independent of the considered joint \( k \)). The vectorial function of the \( i \)th splines is defined as \( q^i(t) := (q^i_1(t), \ldots, q^i_m(t))^T \). A convenient parametriza-
tion of spline $q_k^i(t)$ naturally incorporates continuity of positions and velocities (Craig 1989):

$$q_k^i(t) = q_k^i(-1) + v_k^i(-1)t + \left[\frac{3}{h_i^2} (q_k^i - q_k^i(-1)) - \frac{1}{h_i} (v_k^i + 2v_k^i(-1))\right] t^2$$

$$+ \left[-\frac{2}{h_i^3} (q_k^i - q_k^i(-1)) + \frac{1}{h_i^2} (v_k^i + v_k^i(-1))\right] t^3,$$

$t \in [0, h_i]$

The unknown parameters \(\{v_k^1, \ldots, v_k^{n-1}, q_k^1, q_k^{n-1}\}\) in each spline set can be determined by imposing the continuity of acceleration, i.e. solving the following system of \(n+1\) linear equations \((k = 1, \ldots, m)\):

\[
\begin{align*}
\ddot{q}_k^1(0) &= a_k^0 \\
\ddot{q}_k^1(h_1) &= \ddot{q}_k^2(0) \\
& \vdots \\
\ddot{q}_k^{n-1}(h_{n-1}) &= \ddot{q}_k^n(0) \\
\ddot{q}_k^n(h_n) &= a_k^n
\end{align*}
\] (2)

It can easily be seen that once the \(h\)'s have been fixed, the system in equations (2) admits a unique solution for any assigned set \(\{q_k^0, q_k^2, \ldots, q_k^{n-2}, q_k^v, v_k, a_k^0, a_k^n\}\) (Lin et al. 1983). The total travelling time required to perform the robotic task is evidently \(\sum_{i=1}^n h_i\), and the (globally) optimal trajectory planning problem with minimum-time criterion can be posed as follows:

\[
\min_{\mathbf{h} \in \mathbb{R}^n_+} \sum_{i=1}^n h_i
\] (3)

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

\[
\begin{align*}
|\dddot{q}_i(t)| &\leq \mathbf{j}_M \quad \forall t \in [0, h_i] \\
|\dddot{q}_i(t)| &\leq \mathbf{a}_M \quad \forall t \in [0, h_i] \\
|\dddot{q}_i(t)| &\leq \mathbf{v}_M \quad \forall t \in [0, h_i]
\end{align*}
\]

where \(\mathbf{h} := (h_1, \ldots, h_n)^T \in \mathbb{R}^n_+\) is the vector of spline times, and \(\mathbf{j}_M := (j_1^M, \ldots, j_M^m)^T, \mathbf{a}_M := (a_1^M, \ldots, a_M^m)^T\) and \(\mathbf{v}_M := (v_1^M, \ldots, v_M^m)^T\) denote the vectors of bound values for the jerk, acceleration, and velocity, respectively. The problem in equation (3), which is a semi-infinite optimization problem, can be transformed into a finite one by taking into account the following arguments. First of all, note that joint jerk \(\dddot{q}_i(t)\) is independent of time and denote it by \(\mathbf{j}(\mathbf{h}) := (j_1^M, \ldots, j_M^m)^T\) (whose components are rational functions of vector argument \(\mathbf{h}\)). Hence, the first inequality \(|\dddot{q}_i(t)| \leq \mathbf{j}_M, \forall t \in [0, h_i]\) is simply substituted by

\[-\mathbf{j}_M \leq \mathbf{j}(\mathbf{h}) \leq \mathbf{j}_M\]

Also, joint acceleration functions \(\dddot{q}_i(t)\) are affine functions of time, so that the semi-infinite inequality \(|\dddot{q}_i(t)| \leq \mathbf{a}_M, \forall t \in [0, h_i]\) is equivalent to

\[-\mathbf{a}_M \leq a_i^{-1}(\mathbf{h}) \leq \mathbf{a}_M\]

\[-\mathbf{a}_M \leq a_i'(\mathbf{h}) \leq \mathbf{a}_M\]
where \( \mathbf{a}^i(\mathbf{h}) \), a vector of rational functions, explicitly indicates its dependence on vector \( \mathbf{h} \). The last vector inequality constraint of equation (3) has to be pondered more carefully. By considering that \( q_k(t) \) is a quadratic function of time, the semi-infinite inequality \( |\dot{q}_k(t)| \leq v^k, \forall t \in [0,h_i] \) can be equivalently substituted by the following set of inequalities:

\[
\begin{align*}
-v_M^k & \leq v_{k}^{-1}(\mathbf{h}) \leq v_M^k \\
-v_M^k & \leq v_{k}^i(\mathbf{h}) \leq v_M^k \\
 v_{k}^{i+}(\mathbf{h}) & \leq v_M^k \quad \text{if} \quad a_{k}^{i-1}(\mathbf{h}) > 0 \text{ and } a_{k}^i(\mathbf{h}) < 0 \\
 v_{k}^{-}(\mathbf{h}) & \leq v_M^k \quad \text{if} \quad a_{k}^{i-1}(\mathbf{h}) < 0 \text{ and } a_{k}^i(\mathbf{h}) > 0
\end{align*}
\]

where

\[
\begin{align*}
v_{k}^{i+}(\mathbf{h}) & := \dot{q}_k \left( \frac{a_{k}^{i-1}(\mathbf{h})}{a_{k}^{i-1}(\mathbf{h}) - a_{k}^i(\mathbf{h})} \right) \\
v_{k}^{-}(\mathbf{h}) & := \dot{q}_k \left( \frac{-a_{k}^{i-1}(\mathbf{h})}{a_{k}^i(\mathbf{h}) - a_{k}^{i-1}(\mathbf{h})} \right)
\end{align*}
\]

The introduced \( v_{k}^{i+}(\mathbf{h}) \), \( v_{k}^{-}(\mathbf{h}) \) are the global maximum and minimum of the velocity over \( [0,h_i] \), respectively, provided that the endpoint accelerations \( a_{k}^{i-1}(\mathbf{h}) \), \( a_{k}^i(\mathbf{h}) \) have discordant signs. Note that \( \mathbf{j}(\mathbf{h}) \), \( \mathbf{a}(\mathbf{h}) \), \( \mathbf{v}(\mathbf{h}) \), and \( \mathbf{v}^{i+}(\mathbf{h}) := (v_{k}^{i+}(\mathbf{h}), \ldots, v_{m}^{i+}(\mathbf{h}))^T \), \( \mathbf{v}^{-}(\mathbf{h}) := (v_{k}^{-}(\mathbf{h}), \ldots, v_{m}^{-}(\mathbf{h}))^T \) are given as explicit vector functions of spline times; they are well-defined vector rational functions over \( \mathbb{R}^{+n} \), to be determined by solving the tridiagonal linear systems in equation (2). Finally, we obtain this equivalent finite optimization problem:

\[
\min_{\mathbf{h} \in \mathbb{R}^{+n}} \sum_{i=1}^{n} h_i
\]

subject to

\[
\begin{align*}
-\mathbf{j}_M & \leq \mathbf{j}(\mathbf{h}) \leq \mathbf{j}_M \quad i = 1, \ldots, n \\
-\mathbf{a}_M & \leq \mathbf{a}(\mathbf{h}) \leq \mathbf{a}_M \quad i = 1, \ldots, n - 1 \\
-\mathbf{v}_M & \leq \mathbf{v}(\mathbf{h}) \leq \mathbf{v}_M \quad i = 1, \ldots, n - 1 \\
v^{i+}(\mathbf{h}) & \leq \mathbf{v}_M \quad \text{if} \quad \mathbf{a}^{i-1}(\mathbf{h}) > 0 \text{ and } \mathbf{a}^i(\mathbf{h}) < 0 \quad i = 1, \ldots, n \\
v^{-}(\mathbf{h}) & \leq \mathbf{v}_M \quad \text{if} \quad \mathbf{a}^{i-1}(\mathbf{h}) < 0 \text{ and } \mathbf{a}^i(\mathbf{h}) > 0 \quad i = 1, \ldots, n
\end{align*}
\]

Remark 1: All the vector inequalities in equations (7–11) have to be interpreted componentwise. In particular, the conditioned inequality in equation (10), for a given \( i \), is equivalent to

\[
\{v_{k}^{i+}(\mathbf{h}) \leq v_{M_k} \text{ if } a_{k}^{i-1}(\mathbf{h}) > 0 \text{ and } a_{k}^i(\mathbf{h}) < 0; k = 1, \ldots, m\}
\]
Note that if \( a_k^{-1}(h)a_k'(h) \geq 0 \) the corresponding conditioned inequality has to be considered satisfied regardless of the actual value of \( v_k'(h) \). Analogous considerations can apply to the constraints in equation (11).

**Remark 2:** The novelty of the above optimization problem with respect to standard finite optimization is the presence of the ‘conditional’ constraints in equations (10) and (11) which depend on logical connectives. From a geometrical viewpoint this novelty is a formal one because the boundary of the feasible set in \( \mathbb{R}^{+n} \) is given again by piecewise algebraic surfaces as in the case of standard finite optimization.

The complete data set of the optimization problem in equation (6)–(11) can be displayed as follows:

\[
\begin{align*}
q^0, v^0, a^0 & \quad \text{(displacement, velocity, and acceleration vectors at the knot 0)} \\
q^2, q^3, \ldots, q^{n-2} & \quad \text{(vectors of displacements at the intermediate knots)} \\
q^n, v^n, a^n & \quad \text{(displacement, velocity, and acceleration vectors at the knot n)} \\
v_M, a_M, j_M & \quad \text{(vectors of absolute bounds for velocity, acceleration, and jerk)}
\end{align*}
\]

The actual values in the data set (12) are arbitrary with the sole compatibility requirement given by \(|v^0| \leq v_M, |v^n| \leq v_M\) and \(|a^0| \leq a_M, |a^n| \leq a_M\). In our framework, solving the minimum-time trajectory problem implies finding a global minimizer \( h^* = (h_1^*, \ldots, h_n^*) \) corresponding to the global minimum \( \tau^* = \sum_{i=1}^n h_i^* \) of problem in equations (6)–(11).

### 3. The bisecting-plane algorithm

The following definition introduces the feasible set \( \mathcal{F} \) of equations (6–11) which is not empty for any given data set (12).

**Definition 1:** The feasible set in \( \mathbb{R}^{+n} \) is denoted by \( \mathcal{F} := \{ h \in \mathbb{R}^{+n} \text{ such that } h \text{ satisfies equations (7)–(11)} \}\).

**Assumption 1:** Assume that the values of the initial and final velocity and acceleration are fixed to zero \( (v^0 = v^n = 0, a^0 = a^n = 0) \).

By virtue of Assumption 1, a strong result regarding the structure of \( \mathcal{F} \) is given by the following property.

**Property 1** (conical parametrization of \( \mathcal{F} \)): Denote by \( \mathcal{E} := \{ h \in \mathbb{R}^{+n} : \| h \| = 1 \} \) the set of all the unit-vectors in \( \mathbb{R}^{+n} \). Then there exists a scalar function \( r : \mathcal{E} \to \mathbb{R}^+, e \to r(e) \) such that \( \mathcal{F} = \{ h \in \mathbb{R}^{+n} : h = \lambda r(e)e, \lambda \geq 1, e \in \mathcal{E} \} \).

**Proof:** For notational simplicity the proof is provided with \( m = 1 \) for the general \( m \)-joint case which is immediately attainable. In this case, all the vectors of equations (7–11) are indeed scalar functions, so that \( \mathcal{F} \) becomes

\[
\mathcal{F} = \{ h \in \mathbb{R}^{+n} : \| a'(h) \| \leq a_M, \quad |v'(h)| \leq v_M, \quad i = 1, \ldots, n-1; \\
\| f'(h) \| \leq j_M, \quad v'(h) \leq v_M \text{ if } \{ a^{i-1}(h) > 0 \land a'(h) < 0 \}, \\
v'(h) \geq -v_M \text{ if } \{ a^{i-1}(h) < 0 \land a'(h) > 0 \}, \quad i = 1, \ldots, n \} \quad (13)
\]
By scrutiny of equations (1) and (2) the structures of functions $v'(\mathbf{h})$, $d'(\mathbf{h})$, and $j'(\mathbf{h})$ can be deduced:

$$v'(\mathbf{h}) = \frac{d_1 P_{il}^{(12n-6)}(\mathbf{h}) + \cdots + d_s P_{ls}^{(12n-6)}(\mathbf{h})}{h_{i+1} D^{(3n-3)}(\mathbf{h})}$$  \hspace{2cm} (14)

$$d'(\mathbf{h}) = \frac{d_1 Q_{il}^{(3n-3)}(\mathbf{h}) + \cdots + d_s Q_{ls}^{(3n-3)}(\mathbf{h})}{h_{i+1}^2 D^{(3n-3)}(\mathbf{h})}$$  \hspace{2cm} (15)

$$j'(\mathbf{h}) = \frac{d_1 R_{il}^{(3n-3)}(\mathbf{h}) + \cdots + d_s R_{ls}^{(3n-3)}(\mathbf{h})}{h_{i}^3 D^{(3n-3)}(\mathbf{h})}$$  \hspace{2cm} (16)

where $D^{(3n-3)}(\mathbf{h})$ is the determinant of the coefficient matrix of equation (2) and $P_{il}^{(3n-3)}(\mathbf{h}), \ldots, R_{ls}^{(3n-3)}(\mathbf{h})$ are other homogeneous polynomials, all with degree equal to $3(n-1)$. Moreover, taking into account the definitions in equations (4) and (5) and equations (14) and (15), we deduce

$$v^+(\mathbf{h}) = \frac{E^{(12n-6)}(\mathbf{h}; \mathbf{d})}{h_i D^{(3n-3)}(\mathbf{h}) E^{(9n-3)}(\mathbf{h}; \mathbf{d})}$$  \hspace{2cm} (17)

$$v^-(\mathbf{h}) = \frac{H^{(12n-6)}(\mathbf{h}; \mathbf{d})}{h_i D^{(3n-3)}(\mathbf{h}) G^{(9n-3)}(\mathbf{h}; \mathbf{d})}$$  \hspace{2cm} (18)

where $\mathbf{d} = (d_1, \ldots, d_s)$ and $E^{(9n-3)}(\mathbf{h}; \mathbf{d})$, $F^{(12n-6)}(\mathbf{h}; \mathbf{d})$, $G^{(9n-3)}(\mathbf{h}; \mathbf{d})$, and $H^{(12n-6)}(\mathbf{h}; \mathbf{d})$ are homogeneous $\mathbf{h}$-polynomials with coefficients depending on $\mathbf{d}$. For every $\mathbf{e} \in \mathcal{E}$ define $\Gamma(\mathbf{e}) := \{\gamma \in \mathbb{R}^n_+ : \gamma \mathbf{e} \in \mathcal{F}\}$ and

$$r(\mathbf{e}) := \min_{\gamma \in \Gamma(\mathbf{e})} \{\gamma\}$$  \hspace{2cm} (19)

Function $r(\mathbf{e})$ is well defined because $\mathcal{F}$ is a closed set and $\Gamma(\mathbf{e})$ is not empty for any given $\mathbf{e}$. Indeed, by virtue of the properties of homogeneous polynomials, we infer that

$$v'(\gamma \mathbf{e}) = \frac{1}{\gamma} v'(\mathbf{e}), \quad d'(\gamma \mathbf{e}) = \frac{1}{\gamma^3} d'(\mathbf{e}), \quad j'(\gamma \mathbf{e}) = \frac{1}{\gamma^3} j'(\mathbf{e})$$  \hspace{2cm} (20)

and

$$v^+(\gamma \mathbf{e}) = \frac{1}{\gamma} v^+(\mathbf{e}), \quad v^-(\gamma \mathbf{e}) = \frac{1}{\gamma} v^-(\mathbf{e})$$  \hspace{2cm} (21)

So that for a sufficiently high $\gamma$, surely $\gamma \mathbf{e}$ belongs to $\mathcal{F}$. Define the set $\mathcal{F}_c$ as

$$\mathcal{F}_c := \{\mathbf{h} \in \mathbb{R}^n_+ : \mathbf{h} = \lambda r(\mathbf{e}) \mathbf{e}, \quad \lambda \geq 1, \mathbf{e} \in \mathcal{E}\}$$  \hspace{2cm} (22)

First verify that $\mathcal{F} \subseteq \mathcal{F}_c$, i.e. for any $\mathbf{h} \in \mathcal{F}$ then $\mathbf{h} \in \mathcal{F}_c$. Define $\mathbf{e} := \mathbf{h}/||\mathbf{h}||$ and $\lambda := ||\mathbf{h}||/r(\mathbf{e})$. Clearly the following identity holds: $\mathbf{h} = \lambda r(\mathbf{e}) \mathbf{e}$. Since $r(\mathbf{e}) \leq ||\mathbf{h}||$ by equation (19), we imply $\lambda \geq 1$. The proof is completed by showing that $\mathcal{F}_c \subseteq \mathcal{F}$. Consider $\mathbf{h} \in \mathcal{F}_c$, i.e. $\mathbf{h} = \lambda r(\mathbf{e}) \mathbf{e}$ with $\lambda \geq 1$ and $\mathbf{e} \in \mathcal{E}$. By virtue of equation (19), the vector $r(\mathbf{e}) \mathbf{e}$ belongs to $\mathcal{F}$. The derived equations (20) and (21) imply that

$$|v'(\lambda r(\mathbf{e}) \mathbf{e})| \leq |v'(r(\mathbf{e}) \mathbf{e})|, \quad |d'(\lambda r(\mathbf{e}) \mathbf{e})| \leq |d'(r(\mathbf{e}) \mathbf{e})|, \quad |j'(\lambda r(\mathbf{e}) \mathbf{e})| \leq |j'(r(\mathbf{e}) \mathbf{e})|,$$

∀$\lambda \geq 1$, and
This means $\lambda r(\mathbf{e}) \mathbf{e} \in \mathcal{F}, \forall \lambda \geq 1$. 

Remark 3: The conical parametrization of $\mathcal{F}$ provided does not imply that $\mathcal{F}$ is convex, but that $\mathcal{F}$ is certainly simply connected.

Given a positive parameter $\tau$ and the associated hyperplane $h_1 + \cdots + h_n = \tau$, consider the intersection of this hyperplane with $\mathbb{R}^n_+:

$$T := \{ \mathbf{h} \in \mathbb{R}^n_+ : h_1 + \cdots + h_n = \tau \}$$

$T$ is the (open) convex polyhedron defined by vertexes $(\tau, 0, \ldots, 0)^T$, $(0, 0, \ldots, \tau)^T$. A strong result which characterizes $T$ as a special separating set in the search domain $\mathbb{R}^n_+$ is the following.

Property 2: If the convex polyhedron $T$ is not feasible, i.e. $T \cap \mathcal{F} = \emptyset$, then the negative half space $h_1 + \cdots + h_n < \tau$ is not feasible either. Formally, it holds that $\{ \mathbf{h} \in \mathbb{R}^n_+ : h_1 + \cdots + h_n < \tau \} \cap \mathcal{F} = \emptyset$.

Proof: This property is an immediate consequence of Property 1. Indeed suppose the absurdity that there exists a feasible point $\mathbf{h} \in \{ \mathbf{h} \in \mathbb{R}^n_+ : h_1 + \cdots + h_n < \tau \}$. Define $\tilde{\mathbf{e}} := \mathbf{h}/\|\mathbf{h}\|$ and $\tilde{\tau} = h_1 + \cdots + h_n < \tau$. By virtue of Property 1 there exists $\lambda > 1$ such that $\mathbf{h} = \lambda r(\tilde{\mathbf{e}}) \tilde{\mathbf{e}}$. Determine the intersection of the line ray $\mathbf{h} = \lambda r(\tilde{\mathbf{e}}) \tilde{\mathbf{e}}$ with the hyperplane $h_1 + \cdots + h_n = \tau$: $\lambda r(\tilde{\mathbf{e}}) \tilde{e}_1 + \cdots + \lambda r(\tilde{\mathbf{e}}) \tilde{e}_n = \tau$. It follows that $\lambda = \tilde{\tau}/(r(\tilde{\mathbf{e}}) \tilde{e}_1 + \cdots + r(\tilde{\mathbf{e}}) \tilde{e}_n)$. Since $\tilde{\lambda} = \tilde{\tau}/(r(\tilde{\mathbf{e}}) \tilde{e}_1 + \cdots + r(\tilde{\mathbf{e}}) \tilde{e}_n)$, it follows that $\lambda > \tilde{\lambda}$, which in turn implies that $\lambda r(\tilde{\mathbf{e}}) \tilde{\mathbf{e}} \in T$ is feasible. This is the contradiction which ends the proof.

The special role played by the convex polyhedron $T$, as expressed by Property 2, is enforced considering that $T$ is also a level curve of the objective function $\sum_{i=1}^n h_i$. These facts suggest devising the following bisecting-plane (BP) algorithm to solve the problem in equations (6)--(11).

Input of the BP algorithm
1. the data set (12);
2. the precision parameter $\varepsilon$ ($> 0$);
3. the threshold parameter $\eta$ ($> 0$) to be used by procedure Unfeasibility.

Output of the BP algorithm
1. $\tau^+, \tau^-$ respectively upper and lower bound of $\tau^*$ satisfying $\tau^+ - \tau^- \leq \varepsilon$;
2. $\mathbf{h}_\tau = (h_{f_1}, \ldots, h_{f_n})^T \in \mathcal{F}$ approximate global minimizer satisfying $\sum_{i=1}^n h_{f_i} = \tau^+$;

The BP algorithm
1. $\tau^- := 0$.
2. Determine a feasible point $\mathbf{h}_\tau \in \mathcal{F}$.
3. To improve the current feasible point $\mathbf{h}_\tau$ apply a local procedure to find (if possible) a better local minimizer $\mathbf{h}_\tau \in \mathcal{F}$ and $\tau^+ := \sum_{i=1}^n h_{f_i}$.
4. $\tau := \tau^+ - \varepsilon$. 

$$|v^+(\lambda r(\mathbf{e}) \mathbf{e})| \leq |v^+(r(\mathbf{e}) \mathbf{e})|, \quad |v^-(\lambda r(\mathbf{e}) \mathbf{e})| \leq |v^-(r(\mathbf{e}) \mathbf{e})|, \quad \forall \lambda \geq 1.$$
(5) Apply procedure \textit{Unfeasibility}(\eta, \tau, \mathbf{h}_f, \text{answer}).

(6) If \text{answer} = 'Critical' apply procedure \textit{Criticalness}(\eta, \tau, \mathbf{h}_f, \text{answer}).

(7) If \text{answer} = 'Unfeasible' then \tau^- := \tau and terminate.

(8) \tau^+ := \sum_{i=1}^n h_{f_i}.

(9) If \tau^+ - \tau^- \leq \varepsilon then terminate.

(10) \tau := (\tau^+ + \tau^-)/2.

(11) Apply procedure \textit{Unfeasibility}(\eta, \tau, \mathbf{h}_f, \text{answer}).

(12) If \text{answer} = 'Critical' apply procedure \textit{Criticalness}(\eta, \tau, \mathbf{h}_f, \text{answer}).

(13) If \text{answer} = 'Unfeasible' then \tau^- := \tau and go to 9.

(14) Go to 3.

(15) End.

At Step 2, this algorithm requires a feasible point of \mathcal{F} to be found. The simplest method to accomplish this is to choose any point \mathbf{h} \in \mathbb{R}^n. If \mathbf{h} is not feasible, then scale this point \mathbf{h} with a sufficiently high factor \lambda to obtain \mathbf{h}_f = \lambda \mathbf{h}. By virtue of Property 1, \mathbf{h}_f has to be feasible. A local optimization procedure applied to equations (6)–(11) at Step 3 is used to improve the current upper bound of \tau^*.

To speed up the BP algorithm it is opportune to use a good local optimizer (such as, for example, the Matlab optimizer for constrained optimization (Grace 1994)). The core of the algorithm is the procedure \textit{Unfeasibility} which requires that positive real values are already assigned to its first two arguments. The aim of this procedure is to prove that the convex polyhedron \mathcal{T} is completely unfeasible; in this case the output is given by \text{answer} = 'Unfeasible'. If it is not possible to prove that \mathcal{T} \cap \mathcal{F} = \emptyset then two cases emerge:

1. a feasible point \mathbf{h}_f has been found: then the procedure stops with \text{answer} = 'Feasible';
2. no conclusion can be obtained (critical case): then the procedure has to halt with \text{answer} = 'Critical'.

The critical case of procedure \textit{Unfeasibility} can appear when \mathcal{T} is completely unfeasible but very close to the feasibility region \mathcal{F}, or when \mathcal{T} is unfeasible with the exception of isolated feasible points or of very tiny feasible regions embedded in \mathcal{T}. In order to guarantee the overall global convergence of the BP algorithm, procedure \textit{Unfeasibility} has to satisfy these properties (cf. subsection 4.2):

\textbf{Property 3:} For any given unfeasible convex polyhedron \mathcal{T}, i.e. \mathcal{T} \cap \mathcal{F} = \emptyset, there exists a sufficiently small \eta > 0 for which procedure \textit{Unfeasibility}(\eta, \tau, \mathbf{h}_f, \text{answer}) converges within a finite number of steps with \text{answer} = 'Unfeasible'.

\textbf{Property 4:} For any given convex polyhedron \mathcal{T} containing a non-empty feasible subset \mathcal{T} \cap \mathcal{F}, which is robust as an \((n-1)\) dimensional set, i.e. \text{clo}(\text{int}(\mathcal{T} \cap \mathcal{F})) = \mathcal{T} \cap \mathcal{F} (Luenberger, p. 369), there exists a sufficiently small \eta > 0 for which procedure \textit{Unfeasibility}(\eta, \tau, \mathbf{h}_f, \text{answer}) converges within a finite number of steps with \text{answer} = 'Feasible'.

The next section shows how to construct a procedure satisfying Properties 3 and 4 by using interval analysis techniques coupled with an exhaustive global search upon \mathcal{T}.  

Mechanical manipulators planned by interval analysis
The presentation of procedure \textit{Criticalness} completes the exposition of the BP algorithm.

\textbf{Procedure Criticalness}

1. \( p := 2.\)
2. For \( i = 1, \ldots, p - 1 \)
   \( \alpha : = \frac{\tau + i/p \cdot (\varepsilon/2)}{\eta} = \eta/2. \)
   \( (b) \) Apply procedure \textit{Unfeasibility} \((\eta, \xi, h_f, \text{answer}). \)
   \( (c) \) If \( \text{answer} = \text{‘Unfeasible’} \) then \( \tau : = \xi \) and return.
   \( (d) \) If \( \text{answer} = \text{‘Feasible’} \) then return.
3. \( p := p + 1 \) and go to 2.
4. End.

The aim of procedure \textit{Criticalness} lies in resolving with certainty the critical case which may emerge at Steps 5 and 11 of the BP algorithm.

\textbf{Theorem 1:} For any positive values of \( \varepsilon \) and \( \eta, \) the BP algorithm converges with certainty and solves the optimization problem in equations (6)--(11) in accordance with the given output definition.

\textbf{Proof:} First we prove the correct behaviour of procedure \textit{Criticalness} to be used at Steps 6 and 12 whenever \( \text{answer} = \text{‘Critical’} \) (critical case). This procedure has to converge in a finite number of iterations at Step 2(c) with \( \text{answer} = \text{‘Unfeasible’} \) and \( \xi \in (\tau, \tau + \varepsilon/2) \) or at Step 2(d) with \( \text{answer} = \text{‘Feasible’} \) and a new, better feasible point \( h_f. \) Introduce the set
\[
I : = \{\xi \in (\tau, \tau + \varepsilon/2) : T \cap F = \emptyset \text{ or } \text{clo}(\text{int}(T \cap F)) = T \cap F\}
\] (23)
Since the feasible set \( F \) is defined through rational function inequalities (7)--(11), the set \( I \) is given by the same interval \((\tau, \tau + \varepsilon/2)\) with the exception of a finite number of values:
\[
I = (\tau, \tau + \varepsilon/2) - \{\xi^{(1)}, \ldots, \xi^{(u)}\}
\] (24)
for some \( u \in \mathbb{N}. \) Indeed a value \( \xi \in (\tau, \tau + \varepsilon/2) \) does not belong to \( I \) if and only if it corresponds to a local tangency of the boundary of \( F, \) given by a piecewise algebraic surface, with the hyperplane \( h_1 + \cdots + h_n = \tau. \) Therefore, the cyclic iterations of \( \xi \)-values generated by procedure \textit{Criticalness} necessarily determines \( \xi^* \in I \) for some \( p^* \) and \( i^* \) positive integers. Thus, the same values \( \xi^* \) are generated indefinitely many times in correspondence with the sequences \( k_i = ki^* \) and \( p_k = kp^*, \) \( k \in \mathbb{N} \) with values of the threshold parameter \( \eta \) becoming smaller and smaller. By virtue of Properties 3 and 4, this proves that procedure \textit{Criticalness} halts at Step 2(c) or 2(d) within a finite number of iterations.

The global convergence of the BP algorithm is now proved. The key result to be used is Property 2. Indeed this result demonstrates that if procedure \textit{Unfeasibility}(\( \eta, \tau, h_f, \text{answer} \)) converges with \( \text{answer} = \text{‘Unfeasible’}, \) then \( \tau \) is a lower bound of \( \tau^*, \) i.e. \( \tau \leq \tau^* . \) This validates the assignments of Steps 7 and 13. In particular, if at Step 7 \( \text{answer} = \text{‘Unfeasible’}, \) then we have found a lower bound \( \tau^- \) which is equal to \( \tau^+ - \varepsilon, \) or belongs to \((\tau^+ - \varepsilon, \tau^+ - \varepsilon/2)\) in cases where the previous step was performed by procedure \textit{Criticalness}. Hence inequality
\( \tau^+ - \tau^- \leq \varepsilon \) holds and the algorithm halts with the requested output. On the other hand, if the algorithm does not halt at Step 7, it necessarily does halt at Step 9. This comes from the bisection mechanism issued at Steps 10, 13, and 14 which generates a sequence of converging lower and upper bounds \( \tau^- \) and \( \tau^+ \).

4. The interval procedure

The BP algorithm presented in the previous section provides correct results regardless of the algorithmic method used to realize the procedure \( \text{Unfeasibility}(g, \tau, h, f, \text{answer}) \). While it is conceivable to adopt general global optimization methods such as, for example, constrained Lipschitz optimization (Thach and Tuy 1987) it is more viable to solve the posed unfeasibility problem over \( T \) by using interval analysis techniques. First a brief introduction to interval analysis is given, and then the devised interval procedure follows.

4.1. Interval analysis

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

\[
[a, b] + [c, d] := [a + c, b + d]
\]

\[
[a, b] - [c, d] := [a - d, b - c]
\]

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

\[
[a, b] / [c, d] := [a, b] \cdot [1 / d, 1 / c] \text{ if } 0 \notin [c, d]
\]

Regarding the algebraic structure, the mathematical systems \( \{I, +\} \) and \( \{I, \cdot\} \) are commutative semigroups containing the ‘units’ \([0, 0]\) and \([1, 1]\), respectively. It is noteworthy to cite the subdistributivity property, if \( A, B, C \in I \), then

\[
A(B + C) \subseteq AB + AC
\]

and the monotonic inclusion, if \( A, B, C, D \in I \) with \( A \subseteq C \) and \( B \subseteq D \) then

\[
A * B \subseteq C * D
\]

with * denoting any of the arithmetic operators \( +, -, \cdot \) and \( / \) (the additional hypothesis for the case \( / \) is \( 0 \notin D \)). Let \( X \in I^n \) be a multidimensional interval (box) of \( \mathbb{R}^n \), i.e. \( X = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+] \). Denote by \( I(X) := \{ Y \in I^n : Y \subseteq X \} \) and \( w(X) := \max_{x_i = 1, \ldots, n} \{ x_i^+ - x_i^- \} \) respectively the set of all subboxes of \( X \) and the ‘width’, or the measure, of \( X \). Consider \( f : X \rightarrow \mathbb{R} \), a real scalar function \( f(x) \) defined over the box \( X \); for any \( Y \in I(X) \) define by \( f(Y) := \{ f(x) : x \in Y \} \) the range of \( f \) over \( Y \).

**Definition 2:** A function \( F : I(X) \rightarrow I \) is an inclusion function of \( f(x) \) if

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

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 (Ratschek and Rokne 1988, Hansen 1992). There are a
large variety of inclusion functions: natural interval extensions, standard centred forms, Taylor forms, Cornelius–Lohner forms, etc. (Ratschek and Rokne 1984). The simplest inclusion function is the natural interval extension, which can be obtained by substituting, the requested box argument \( Y \) in the expression of a given \( f(x) \), and then performing the necessary interval computations. For example, given \( f(x) = (1 + 3x^2) \sin(7x) \), its natural interval extension \( F(Y) \) with \( Y = [-1, 2] \) can be computed as follows:

\[
F(Y) = (1 + 3[-1, 2]) \sin((-7, 14)) = (1 + [0, 12][-1, 1]) = [1, 13][-1, 1] = [-13, 13].
\]

In the following, scalar functions are denoted by lower-case letters, whereas inclusion functions are denoted by corresponding upper-case letters (e.g. \( f(x) \) and \( F(X) \)).

4.2. The procedure Unfeasibility

The following assumption is considered to guarantee that the problem in equations (6)-(11) admits a global solution (it avoids degenerate situations such as \( q^0 = q^n = q^i = \text{constant, } i = 2, \ldots, n - 2 \) for which obviously there is no solution to equations (6)-(11)).

Assumption 2: A sufficiently small \( \nu > 0 \) is known such that

\[
\mathbb{R}^{\nu} := \{ h \in \mathbb{R}^n : h_i \geq \nu, i = 1, \ldots, n \} \supseteq \mathcal{F}
\]

To simplify the subsequent development, write all the constraints in equations (7)-(11) by introducing the truth-value functions \( b_i : \mathbb{R}^{\nu} \rightarrow \{ \text{true, false} \} \), \( h \rightarrow b_i(h) \) defined as follows:

\[
b_i(h) := \{-c_i \leq c_i(h) \land c_i(h) \leq \bar{c}_i \}, \quad i = 1, \ldots, (3n-2)m
\]

\[
b_i(h) := \{ y_i(h)z_i(h) \geq 0 \} \lor \{ y_i(h) > 0 \land z_i(h) < 0 \land c_{u_i}(h) \leq \bar{c}_i \}
\]

\[
\lor \{ y_i(h) < 0 \land z_i(h) > 0 \land -\bar{c}_i \leq c_{d_i}(h) \}
\]

\[
i = (3n-2)m + 1, \ldots, (4n-2)m
\]

where the introduced functions can be unambiguously identified from equations (7)-(11). By virtue of Assumption 2, the convex polyhedron \( T \) can be substituted by \( T_\nu \):

\[
T_\nu := \{ h \in \mathbb{R}^n : h_1 + \cdots + h_n = \tau, h_i \geq \nu, i = 1, \ldots, n \}
\]

The procedure Unfeasibility aims to prove one of these statements:

(1) For any \( h \in T_\nu \) there exists an integer \( i_h \) (which depends on \( h \)) such that \( b_{i_h}(h) = \text{false}; \) output argument answer := ‘Unfeasible’.

(2) There exists \( h_f \in T_\nu \) such that \( b_i(h_f) = \text{true}, i = 1, \ldots, (4n-2)m; \) output arguments answer := ‘Feasible’ and \( h_f \).

If it is not possible to establish (1) or (2), then

(3) the output argument answer is set to ‘Critical’ (critical case).

The following notation is introduced. Denote a vector point in \( \mathbb{R}^{+(n-1)} \) by \( h := (h_1, \ldots, h_{n-1})^T \) and introduce:
The above functions with domain in \( \mathbb{R}^{+(n-1)} \) are simply obtained by the corresponding constraints functions evaluated over the hyperplane \( h_1 + \cdots + h_n = \tau \). The multidimensional interval to be processed by the interval procedure is given by

\[
A_v := [v, \tau - (n-1)v]^{n-1} \subseteq \mathbb{R}^{n-1}
\]

A generic box \( H \subseteq \mathbb{R}^{n-1} \) can explicitly be denoted by \( H := [h_1^-, h_1^+] \times \cdots \times [h_{n-1}^-, h_{n-1}^+] \). Let the projection of \( H \) onto the hyperplane \( h_1 + \cdots + h_n = \tau \) be denoted by

\[
\mathcal{P}(H) := \left\{ h \in \mathbb{R}^n : h = \left( h_1, h_2, \ldots, h_{n-1}, \tau - \sum_{i=1}^{n-1} h_i \right)^T, \; \tilde{h} \in H \right\}
\]

This projection has a useful property.

**Property 5:** Assume \( H \subseteq A_v \); the following statements hold

\[
\mathcal{P}(A_v) \supseteq T_\nu
\]

\[
\mathcal{P}(H) \subseteq T_\nu \text{ if and only if } \tau - \sum_{i=1}^{n-1} h_i^+ \geq \nu
\]

\[
\mathcal{P}(H) \cap T_\nu = \emptyset \text{ if and only if } \tau - \sum_{i=1}^{n-1} h_i^- < \nu
\]

\[
\mathcal{P}(H) \subseteq T_\nu \text{ if and only if } \tau - \sum_{i=1}^{n-1} h_i^+ > 0
\]

Proof of the above property is omitted because of its immediate derivability.

**Procedure Unfeasibility**

1. Insert \( A_v \) into the initialized stack \( S \); set \( \text{crit} := 0 \).
2. If the stack \( S \) is empty then
   
   (a) If \( \text{crit} = 0 \) then \( \text{answer} := \text{'Unfeasible'} \) else \( \text{answer} := \text{'Critical'} \).
(b) Return.

(3) From $S$ unstack into $H$.

(4) If $\tau - \sum_{i=1}^{n-1} h_i^- < \nu$ then go to 2 (box $H$ is rejected).

(5) If $\tau - \sum_{i=1}^{n-1} h_i^+ < \nu/p_t$ then go to 11.

(6) $h_f := (h_1^-, \ldots, h_{n-1}^-, \tau - \sum_{i=1}^{n-1} h_i^-)^T$; $\text{count} := 0$.

(7) For $i = 1, \ldots, (3n-2)m$ do
   (a) If $b_i(h_f) = \text{'true'}$ then $\text{count} := \text{count} + 1$ and go to (c).
   (b) If $\mathcal{C}_i(H; \tau) \cap [-\bar{c}_i, \bar{c}_i] = \emptyset$ then go to 2.
   (c) end of $i$-loop.

(8) For $i = (3n-2)m + 1, \ldots, (4n-2)m$ do
   (a) If $b_i(h_f) = \text{'true'}$ then $\text{count} := \text{count} + 1$ and go to (d).
   (b) Denote by $[y^-, y^+], [z^-, z^+], [c_u^-, c_u^+], [c_d^-, c_d^+]$ the interval values of $\bar{Y}_i(H; \tau), \bar{Z}_i(H; \tau), \mathcal{C}_u(H; \tau), \mathcal{C}_d(H; \tau)$, respectively.
   (c) If $\{y^- > 0 \land z^+ < 0 \land \bar{c}_i < c_u^+\} \lor \{y^+ < 0 \land z^- > 0 \land c_d^+ < -\bar{c}_i\}$ then go to 2.
   (d) end of $i$-loop.

(9) If $\text{count} = (4n-2)m$ then $\text{answer} := \text{'Feasible'}$ and return.

(10) If $w(H) < \eta$ then $\text{crit} := 1$ and go to 2.

(11) Bisect, thus obtaining boxes $V_1$ and $V_2$, the box $H$ on its maximum dimension.

(12) $H \leftarrow V_1$ and push $V_2$ into $S$.

(13) Go to 4.

(14) End.

The variable arguments to be passed to the procedure are $\eta$, the threshold parameter, and $\tau$, the positive constant associated with the hyperplane $h_1 + \cdots + h_n = \tau$. The arguments to be returned are $\text{answer}$, according to Statements (1), (2) and (3), and $h_f \in \mathcal{F}$ for the case $\text{answer} = \text{'Feasible'}$.

Remark 4: At Step 5, $p_t$ is any fixed integer such that $p_t \geq 2$. This parameter implicitly defines the outer approximation of $T_v$ by means of the set union of all generated $\mathcal{P}(H)$ for which it has to hold $\mathcal{P}(H) \subseteq T$ (cf. statement (39) of Property 5). In other words, all the components of $h \in \mathcal{P}(H)$ have to be strictly positive in order to correctly process $\mathcal{P}(H)$.

Theorem 2: The procedure Unfeasibility always converges. It satisfies Statements (1)–(3) and Properties 3 and 4.

Proof: First it is proved that procedure Unfeasibility always converges, satisfying Statements (1)–(3). The procedure is based on an exhaustive search over the projection $\mathcal{P}(A_v)$ which contains $T_v$ by virtue of equation (36). At Step 4, the current box $H$ is rejected if $H$ satisfies inequality (38): $\mathcal{P}(H)$ is completely external to convex polyhedron $T_v$. At Step 5 it is necessary to bisect $H$ further when condition $\tau - \sum_{i=1}^{n-1} h_i^+ < \nu/p_t$ is satisfied; otherwise the processed projection $\mathcal{P}(H)$ would have points not belonging to $T_v$, or even with a negative last component. Obviously the mechanism issued at Steps 6, 7(a), 8(a) and 9 guarantees that when $\text{answer}$ is set to ‘Feasible’ a feasible point in $T_v$ has been found ($h_f$). If the state-
ment at Step 7(b) or Step 8(c) is true, then by virtue of the definition of inclusion functions, for any \( h \in \mathcal{P}(H) \Rightarrow b_i(h) = \text{false} \Rightarrow h \notin \mathcal{F} \). This dictates that \( H \) is discarded and the next instruction is at Step 2. When the stack \( S \) is empty and \( \text{crit} = 0 \), it means that all the area \( T_V \) has been examined and proved to be unfeasible: then \( \text{answer} := \text{'Unfeasible'} \) and the procedure terminates. If the interval procedure does not converge at Step 2 with \( \text{crit} = 0 \) so that \( \text{answer} = \text{'Unfeasible'} \) or at Step 9 with \( \text{answer} = \text{'Feasible'} \), then it necessarily converges at Step 2 with \( \text{crit} = 1 \) so that \( \text{answer} = \text{'Critical'} \). This can be ascertained by examination of the exhaustive procedure implemented: boxes are inserted and extracted at the top of the stack \( S \) according to a 'depth-first' strategy. Indeed if the procedure does not terminate at Step 2 with \( \text{crit} = 0 \) or at Step 9, the given starting box is bisected over and over again until the current box \( H \) is arbitrarily small (cf. Step 11). Hence, condition \( w(H) < \eta \) is eventually verified for any given positive value of threshold \( \eta \) and variable \( \text{crit} \) is set to 1. Subsequently, by virtue of the exhaustive search over \( A_v \), if the condition at step 9 is never verified, the procedure necessarily clears out all the boxes of stack \( S \) and terminates with \( \text{answer} = \text{'Critical'} \).

Now it will be proved that the exposed procedure satisfies Property 3. Suppose that \( T \) be unfeasible. Then for any \( h \in T \) there exists an integer \( i(h) \) which depends on \( h \) such that \( b_i(h) = \text{false} \). Now introduce the function \( \delta : T \to \mathbb{R}^+, \ h \to \delta(h) \) defined as follows:

If \( i(h) \in \{1, \ldots, (3n-2)m\} \) then
\[
\delta(h) := |c_i(h)| - \bar{c}_i(h)
\]
else (it holds \( i(h) \in \{(3n-2)m + 1, \ldots, (4n-2)m\} \))

If \( \{y_i(h)(h) > 0 \land z_i(h)(h) < 0 \land c_u(h)(h) > \bar{c}_i(h)\} \)
then
\[
\delta(h) := \min\{y_i(h)(h), -z_i(h)(h), c_u(h)(h) - \bar{c}_i(h)\}
\]
else
\[
\delta(h) := \min\{-y_i(h)(h), z_i(h)(h), -\bar{c}_i(h) - c_d(h)(h)\}
\]
endif.
endif.

Define \( \delta^* := \min_{h \in T} \delta(h) \) which is a positive value. Hence for any \( h \in T \) there exists a violated constraint, with the appropriate index \( i \), such that
\[
c_i(h) \geq \bar{c}_i + \delta^*
\]
or
\[
c_i(h) \leq -\bar{c}_i - \delta^*
\]
or
\[
y_i(h) \geq \delta^* \land z_i(h) \leq -\delta^* \land c_u(h) \geq \bar{c}_i + \delta^*
\]
or
\[
y_i(h) \leq -\delta^* \land z_i(h) \geq \delta^* \land -\bar{c}_i - \delta^* \geq c_d(h)
\]

By considering the continuity over \( \mathbb{R}^{n+} \) of all the functions involved in the constraints, we infer there exists a small \( \xi_1 > 0 \) and a constraint with index \( i \) such that for any box \( H \subseteq A_v \) with \( w(H) < \xi_1 \land \mathcal{P}(H) \subseteq T \), one of the following statements is satisfied:
\[ \min_{h \in P(H)} \{ c_i(h) \} \geq \bar{c}_i + \delta^*/2 \quad (40) \]
or
\[ \max_{h \in P(H)} \{ c_i(h) \} \leq -\bar{c}_i - \delta^*/2 \quad (41) \]
or
\[ \min_{h \in P(H)} \{ y_i(h) \} \geq \delta^*/2 \land \max_{h \in P(H)} \{ z_i(h) \} \leq -\delta^*/2 \land \min_{h \in P(H)} \{ c_u(h) \} \geq \bar{c}_i + \delta^*/2 \quad (42) \]
or
\[ \max_{h \in P(H)} \{ y_i(h) \} \leq -\delta^*/2 \land \min_{h \in P(H)} \{ z_i(h) \} \geq \delta^*/2 \land -\bar{c}_i - \delta^*/2 \geq \max_{h \in P(H)} \{ c_d(h) \} \quad (43) \]

Moreover, by virtue of the limit property of inclusion functions, there always exists a small \( \xi_2 > 0 \) such that for any box \( H \subseteq A_\nu \) with \( \omega(H) < \xi_2 \), we simultaneously have (see notation introduced at Step 8(b) and \([c^-, c^+] := \tilde{C}_i(H; \tau)\))

\[ \min_{h \in P(H)} \{ c_i(h) \} - c^- < \delta^*/2 \quad (44) \]
\[ c^+ - \max_{h \in P(H)} \{ c_i(h) \} < \delta^*/2 \quad (45) \]
\[ \min_{h \in P(H)} \{ y_i(h) \} - y^- < \delta^*/2 \quad (46) \]
\[ y^+ - \max_{h \in P(H)} \{ y_i(h) \} < \delta^*/2 \quad (47) \]
\[ \min_{h \in P(H)} \{ z_i(h) \} - z^- < \delta^*/2 \quad (48) \]
\[ z^+ - \max_{h \in P(H)} \{ z_i(h) \} < \delta^*/2 \quad (49) \]
\[ \min_{h \in P(H)} \{ c_u(h) \} - c^- < \delta^*/2 \quad (50) \]
\[ c^+_d - \max_{h \in P(H)} \{ c_d(h) \} < \delta^*/2 \quad (51) \]

Therefore inequalities (40) and (44) imply that
\[ c^- > \bar{c}_i \quad (52) \]
or, considering inequalities (41) and (45),
\[ c^+ < -\bar{c}_i \quad (53) \]
or, considering inequalities (42), (46), (49), and (50),
\[ y^- > 0 \land z^+ < 0 \land \bar{c}_i < c^-_u \quad (54) \]
or, from inequalities (43), (47), (48), and (51),
\[ y^+ < 0 \land z^- > 0 \land c^+_u < -\bar{c}_i \quad (55) \]

This permits the conclusion that by choosing a positive \( \eta < \min\{\xi_1, \xi_2\} \), every sufficiently small box generated by the procedure is eventually discarded by the activation of conditions (52) or (53) at Step 7(b), or conditions (54) or (55) at Step 8(c). Therefore, the exhaustive nature of the procedure implies that the stack \( S \) finally
becomes empty within a finite number of iterations without modifying the zero value of the 'flag' variable crit. This concludes the proof of Property 3.

Finally, Property 4 is verified. Procedure Unfeasibility discards boxes with projection on $T$ when they are proved to be unfeasible (cf. Steps 7(b) and 8(c)) or when they have widths smaller than $\eta$, (cf. Step 10). This guarantees convergence within a finite number of iterations. Suppose that $T \cap \mathcal{F}$ is not empty and is as robust as an $(n - 1)$-dimensional set. Hence, for a sufficiently small $\eta$ the procedure necessarily generates a box $H$ such that $\mathcal{P}(H) \subseteq T \cap \mathcal{F}$. Indeed for any box $H$ with $\mathcal{P}(H) \not\subseteq T \cap \mathcal{F}$ it must happen that: (i) it is discarded by proving that $\mathcal{P}(H)$ is not feasible, or (ii) it is discarded by setting crit $= 1$, or (iii) it is bisected over and over until all the regions of $H$ are discarded through (i) and (ii) or a subbox of $H$ is found with its projection completely contained in $T \cap \mathcal{F}$. As a consequence

$$\mathcal{P}(H) \subseteq T \cap \mathcal{F} \Rightarrow h_f := \left(h_1^n, \ldots, h_{n-1}^n, \tau - \sum_{i=1}^{n-1} h_i^n \right) \in \mathcal{F}$$

so that the condition at Step 10 is verified and the procedure terminates with answer $= \text{'Feasible'}. \square$

Remark 5: To find a value for $\nu$ to be used in the procedure Unfeasibility, in practice we can choose any sufficiently small positive value compatible with computation precision, for example $\nu = \varepsilon$.

5. An illustrative example

The previous bisecting-plane algorithm has been implemented in C++ language exploiting the PROFIL/BIAS libraries made by Knüppel (1993 a, b) and using the Matlab optimization toolbox as local optimizer (Grace 1994). As an illustrative example we considered the case of a trajectory composed of five splines of a six degrees-of-freedom robot manipulator, like the one shown in figure 1, in which the maximum limits of velocity, acceleration, and jerk are the same for each joint and fixed as $v_M^k = 50$ degrees/s, $a_M^k = 50$ degrees/s$^2$ and $j_M^k = 60$ degrees/s$^3$, $k = 1, \ldots, 6$. The starting, intermediate, and final joint positions are indicated in table 1. We set the precision parameter $\varepsilon = 0.01$ and the other parameters $\eta$ and $\nu$ to 0.001 and 0.01, respectively. The BP algorithm has been initialized with the starting feasible point $h_f = (15,15,15,15,15,15)^T$. The optimal travelling time obtained is 12.53 s, with corresponding spline times as listed in table 2. The plots of the velocities,
accelerations, and jerks of the six joints are shown in figures 2–7. Note how the constraints are active for joint 1 (jerk in the first spline and velocity in the third), joint 5 (velocity in the fourth spline, jerk in the last one) and joint 6 (jerk in the first spline). It is worth stressing that the Matlab optimizer, starting from the feasible point \((15, 15, 15, 15, 15)^T\), converges to the feasible local minimum with a total time 15.32 s, which is not the required global minimum (see the corresponding spline times in table 3). Even starting from better initial points, such as \(h_f = (4, 4, 4, 4, 4)^T\), the Matlab optimizer converges to the same result. This definite improvement (of 18%) of the global minimum over the local minimum emphasizes the potential importance

---

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline time (s)</td>
<td>0.80</td>
<td>2.10</td>
<td>4.62</td>
<td>4.45</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Table 2. The optimal spline times.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline time (s)</td>
<td>0.83</td>
<td>2.00</td>
<td>4.72</td>
<td>7.37</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Table 3. The spline times obtained with the Matlab optimizer.

![Figure 2. The optimal trajectory of joint 1 for the example.](image)
Mechanical manipulators planned by interval analysis

Figure 3. The optimal trajectory of joint 2 for the example.

Figure 4. The optimal trajectory of joint 3 for the example.
Figure 5. The optimal trajectory of joint 4 for the example.

Figure 6. The optimal trajectory of joint 5 for the example.
Mechanical manipulators planned by interval analysis

in an automated industrial environment and in space robotics of searching for the global minimum time.

6. Conclusions

A global optimizer, using an interval procedure, was devised for the optimal minimum-time trajectory generation of an m-joint manipulator. The joint space scheme adopted employs cubic splines to ensure continuity of displacements, velocities, and accelerations under constraints on velocities, accelerations, and jerks. The key result, that has led to proof of the global convergence of the proposed algorithmic method of cutting planes in the unfeasible domain, is the conical parametrization of the feasible set $F$ (Property 1). This was derived under the assumption of zero velocities and accelerations in the first and last knot. If we assume as a conjecture that $F$ is connected in the general case (i.e., arbitrary given velocities and accelerations are assigned in the extreme knots), then the BP algorithm would provide a global minimum-time solution again. This conjecture is an open question which needs further investigation.

Even though the minimum-time problem addressed is evidently NP-hard from the computational complexity viewpoint, the proposed BP algorithm can be of practical use in the planning of mechanical manipulators in highly automated environments.
Acknowledgments

Partial support for this research has been provided by MURST scientific research funds and by the ASI (Italian Space Agency).

References


