Chapter 13

A SEMI-INFINITE OPTIMIZATION APPROACH TO OPTIMAL SPLINE TRAJECTORY PLANNING OF MECHANICAL MANIPULATORS

Corrado Guarino Lo Bianco and Aurelio Piazzi
Dipartimento di Ingegneria dell’Informazione, Università di Parma
Parco Area delle Scienze 181A - 43100 Parma - Italy *
guarino@ce.unipr.it, aurelio@ce.unipr.it

Abstract The paper deals with the problem of optimal trajectory planning for rigid links industrial manipulators. According with actual industrial requirements, a technique for planning minimum-time spline trajectories under dynamics and kinematics constraints is proposed. More precisely, the evaluated trajectories, parametrized by means of cubic splines, have to satisfy joint torques and end-effector Cartesian velocities within given bounds. The problem solution is obtained by means of an hybrid genetic/interval algorithm for semi-infinite optimization. This algorithm provides an estimated global minimizer whose feasibility is guaranteed by the use of a deterministic interval procedure, i.e. a routine based on concepts of interval analysis. The proposed approach is tested by planning a 10 via points movement for a two link manipulator.

1. INTRODUCTION.

Motion planning for industrial manipulators can be handled with several approaches depending on the control target. An usual technique solves two different aspects of the control problem separately: the trajectory planning and the design of controllers to track the planned trajectory. This paper is devoted to approach the former problem by means of an hybrid algorithm for generalized global semi-infinite optimization.

Usually the aim is to obtain a movement that satisfies some optimality requirements. For example, in the case of redundant manipulators, the trajectory...
path can be a priori assigned and, by taking advantage of the redundancy, the inverse kinematics can be optimized according to some given objective functions [28, 14, 17].

Under the assumption of assigned path, Bobrow et al. [2] and Shin and McKay [35] proposed to plan minimum time trajectories taking into account constraints on the admissible joint torques/forces. The problem was posed in the same way in both papers but was solved applying different approaches. Path curves were defined by a set of functions parametrized by a single variable (the distance from the origin of the curve to the current position measured along the path). Torque constraints were converted into constraints for the acceleration along the path and, then, into constraints for the velocity. In both papers the authors verified that minimum time motion can be obtained if the acceleration is always kept at its maximum value (positive or negative) compatibly with the constraint on the torque. This consideration has permitted solving the optimization problem by searching proper switching instants for the acceleration. Such switching instants were found by means of different algorithmic approaches in the two papers. Joint torques related to the optimal solution are discontinuous and, at each instant, at least one torque constraint is active.

Under the same assumptions, Shiller and Lu [34] extended the results of [2] and [35]. They converted the original problem into an equivalent reduced problem where torque constraints were replaced with constraints on the maximum acceleration and velocity along the assigned path. Shiller and Lu also verified that, under particular conditions, the assumption made by other authors to use always maximum acceleration to achieve minimum time trajectories can introduce unnecessary chattering into the acceleration itself and can be even unfeasible. Points and segments where this problem arises were denominated by Shiller and Lu as singular points and singular arcs. These authors pointed out that at singular points or arcs, in order to achieve the correct optimal feasible solution, an acceleration smaller than the maximal one is adopted and the velocity is maximized by sliding along the velocity limit curve.

The minimum time trajectory planning problem under specified path and torque constraint has been also addressed by Pfeiffer and Johanni [25]. Their method is similar to that used in [2, 35] but the solution is found by applying a different algorithm based on dynamic programming [1]. Pfeiffer and Johanni indicated one of the major limitation common to all minimum-time approaches: joint torques and forces generated by true minimum-time trajectories are intrinsically discontinuous. They proposed to mitigate this inconvenience by using a mixed performance criterion obtained by combining the usual travelling time with the squares of joint torques and velocities. As a result, in almost all positions of the optimal trajectory no one of the torque constraints is active. Obviously travelling time increases but a benefic smoothing effect on joint torques is achieved.
All the above cited approaches basically exploit the problem structure arriving at highly tailored algorithms. Recently, various authors have suggested to operate in a more general context by using algorithms for semi-infinite optimization. The point of view changes. Typical robotic planning problems are converted into equivalent semi-infinite optimization problems and, then, solved by using proper general purpose algorithms. This is the case, for example, of Haaren-Retagne [11]. He proposed to convert dynamic (joint torques) or kinematic bounds (joint velocities, accelerations and jerks) of a robotic problem into equivalent constraints for a semi-infinite problem whose performance index is given by the total traveling time. The traveling path is supposed to be assigned and parametrized by B-splines. The resulting generalized semi-infinite problem is converted, according with the scheme proposed by Marin [20], into a standard semi-infinite problem. In [11], comparisons with the techniques proposed by Marin [20] and Shin and McKay [35] are also exposed.

A different approach to optimal trajectory planning (usually minimum-time) requires the assignment of prespecified via points. In such a way, the geometric path is not completely defined a priori so that the optimization problem has more degrees of freedom. Number and position of via points may also depend on the obstacles to be avoided. Normally, via points are specified in the Cartesian space and then mapped, via inverse kinematics, into a set of joint space points to be interpolated by suitably chosen smooth functions. An effective approach to minimum-time planning is given by the use of parameter optimization methods. Starting from the paper of Lin et al. [18], where a polyhedron local search technique was used to plan an optimal spline movement under kinematics constraints (velocity, acceleration, and jerk of all the joints were constrained), many efforts have been spent in this direction. For example, the same problem was solved in [27] by devising a global optimization interval algorithm while a local gradient-based procedure for semi-infinite optimization was adopted in [4] to solve a more general problem with torque and joint velocity constraints.

In this paper, the problem of minimum-time trajectory generation is coupled with the fulfillment of two important specifications: limits are imposed on both joint torques and end-effector Cartesian velocities. The first (dynamic) constraint is justified by the limits of the torque exerted by the actuators. The second (kinematic) constraint is introduced to avoid damaging the task of the end-effector tool whose Cartesian velocity, in many practical applications, cannot exceed a given operative maximum. By adopting a cubic spline parametrization for the trajectories, equivalent to that proposed in [18], and a full dynamics model for the manipulator, the optimal trajectory planning problem is converted into a generalized semi-infinite nonlinear optimization problem whose cost index is the total travelling time, while the semi-infinite constraints take into account, without conservativeness, the dynamics and kinematics requirements. The optimization problem is then solved by means of the genetic/interval al-
algorithm presented in [6] [10]. It is a global solver that combines a stochastic optimization technique (a genetic algorithm) to minimize the cost index, with a deterministic optimization routine (an interval procedure) to handle the semi-infinite constraints. This hybrid algorithm permits obtaining an estimate of the global minimizer that is feasible with certainty (torque and Cartesian velocity constraints are always satisfied) and, since it is able to manage directly the generalized problem, does not require to convert it into a standard semi-infinite problem. The same genetic/interval algorithm has been used to cope and solve several classic control engineering problems. For example, under proper hypotheses, it is possible to design optimal robust controllers for plants with structured [6] and/or unstructured uncertainties (H∞ problems) [7, 8, 9] by reformulating the control problems into semi-infinite problems.

The proposed hybrid semi-infinite approach has various advantages with respect to those described in [2] and [35]. First of all it is more flexible. For example, the same algorithm can be used, by simply adding new constraints, to solve problems with further dynamic and kinematic specifications. Assigning via points instead of a fixed path adds degrees of freedom to the optimization problem in such a way that better performances may be achieved. The parametrization chosen for the splines guarantees the continuity of joint torques and forces thus reducing mechanical stresses and possible excitation of unmodelled dynamics.

In §2, the manipulator planning problem is posed and worked out to reformulate it as a generalized semi-infinite optimization problem. A related feasibility result is reported in §3 (Proposition 1). In §4, by means of a penalty method, the semi-infinite problem is converted into an equivalent unconstrained problem in order to apply the genetic/interval algorithm. §5 describes with details the penalty computation via interval analysis: a succinct exposition on inclusion functions is followed by the algorithmic description of an interval procedure and its deterministic convergence is established (Proposition 2). The potential effectiveness of the proposed approach is tested in §6 by planning a 10 via points optimal trajectory for a two link planar manipulator.

Notation.

In the following, vectors are indicated by means of lower case bold characters (e.g. \( \mathbf{q} \)) while matrices are indicated by capital bold characters (e.g. \( \mathbf{M} \)). The absolute value of a vector of \( n \) elements be defined as \( |\mathbf{q}| := [|q_1| \ |q_2| \ \cdots \ |q_n|]^T \), while \( |\mathbf{q}| < |\mathbf{g}| \) means that \( |q_i| < |g_i|, i = 1, 2, \ldots, n \). The euclidean norm of \( \mathbf{q} \) is denoted by \( \|\mathbf{q}\| \). The set of real positive numbers and the set positive integers are denoted with \( \mathbb{R}^+ \) and \( \mathbb{N} \) respectively.
2. CUBIC SPLINE TRAJECTORY PLANNING UNDER TORQUE AND VELOCITY CONSTRAINTS.

Consider an $m$ link robot and denote by $p := [p_1 \ p_2 \ \cdots \ p_m]^T \in \mathbb{P} \subset \mathbb{R}^m$ the joint variable vector belonging to the joint-space work envelope $\mathbb{P}$. Let us assume that $s$ via points have been assigned in the tool-configuration (Cartesian) space. These are mapped, via inverse kinematics problem, into $s$ joint knots of $\mathbb{P}$. It was shown in [18] that, assuming continuity of velocities and accelerations, two free displacement knots must be added in order to exactly interpolate the given via points by cubic spline joint trajectories. The resulting $s+2$ knots can be represented by the data vectors ($n := s+1$)

$$q^i := [q^i_1 \ q^i_2 \ \cdots \ q^i_m]^T, \ i = 0, 1, \ldots, n$$

where $q^i$ and $q^{i-1}$ are the free displacement vectors. In particular, note that the component $q^i_k$ represents the displacement of the $k$-th joint at the $i$-th knot.

The vectors of the joint velocities and accelerations at the $i$-th knot be denoted by $\dot{q}^i := [\dot{q}^i_1 \ \dot{q}^i_2 \ \cdots \ \dot{q}^i_m]^T$ and $\ddot{q}^i := [\ddot{q}^i_1 \ \ddot{q}^i_2 \ \cdots \ \ddot{q}^i_m]^T$ respectively. Velocities and accelerations have to be considered assigned for the first and the last knot, i.e. vectors $\dot{q}^0$, $\ddot{q}^0$, $\dot{q}^n$, $\ddot{q}^n$ are known given data.

Denote by $h := [h_1 \ h_2 \ \cdots \ h_n]^T \in \mathbb{B} := [\nu, +\infty)^n$ a vector of interval times, where $h_i$ is the time required to move all the joints from the $(i-1)$-th via point to the $i$-th one and $\nu$ is a small positive number which is imposed in order to avoid possible degeneracies at the implementation stage. The sum of all the components of $h$ is the total traveling time.

The $i$-th spline function for the $k$-th joint be indicated by $p^i_k(t)$ with time $t \in [0, h_i]$. A convenient parametrization of $p^i_k(t)$ that guarantees the continuity of positions and velocities is the following:

$$p^i_k(t) := q^{i-1}_k + \dot{q}^{i-1}_k t + \frac{3}{h^3_i} (q^i_k - q^{i-1}_k) - \frac{1}{h^2_i} (\dot{q}^i_k + 2 \dot{q}^{i-1}_k) t^2 + \frac{2}{h^3_i} (q^i_k - q^{i-1}_k) + \frac{1}{h^2_i} (\ddot{q}^i_k + \ddot{q}^{i-1}_k) t^3,$$

$i = 1, \ldots, n, \ k = 1, 2, \ldots, m,$

$t \in [0, h_i]. \quad (13.1)$

By imposition of the continuity of the acceleration in the resulting spline trajectory, the following linear system of $n+1$ equations can be written for
each joint \((k = 1, 2, \ldots, m)\)

\[
\begin{aligned}
\ddot{q}_k^1(0) &= \dot{q}_k^0, \\
\ddot{q}_k^2(0) &= \ddot{q}_k^1(h_1), \\
&\vdots \\
\ddot{q}_k^n(0) &= \ddot{q}_k^{n-1}(h_{n-1}), \\
\dot{q}_k^n &= \ddot{q}_k^n(h_n).
\end{aligned}
\]

System (13.2) admits an unique solution, depending on \(h\), in the unknowns \(\{\dot{q}_k^0, \ddot{q}_k^1, \ldots, \dot{q}_k^{n-1}, \ddot{q}_k^n\}\) [18]. Therefore, the cubic spline \(p_k^i(t; h)\) can be redenoted more explicitly as \(p_k^i(t; h)\) and the \(i\)-th spline functions for all the joints can be consequently introduced in vectorial form as:

\[
p^i(t; h) := [p_1^i(t; h) \ p_2^i(t; h) \cdots p_m^i(t; h)]^T, \quad t \in [0, h_i], \ i = 1, 2, \ldots, n.
\]

By neglecting friction, joint torques and forces can be evaluated by means of the manipulator dynamic equation [3, p. 206]

\[
\tau = M(p) \ddot{p} + B(p)[\dot{p}] \dddot{p} + D(p)[\dddot{p}] + g(p),
\]

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

\[
[\dot{p}] := [\dot{p}_1 \ \dot{p}_2 \ \cdots \ \dot{p}_1 \ \dot{p}_2 \ \cdots \ \dot{p}_2 \dot{p}_m \ \cdots \ \dot{p}_1 \dot{p}_m \ \cdots \ \dot{p}_1 \dot{p}_m]^T
\]

and

\[
[\dddot{p}] := [(\dot{p}_1)^2 \ (\dot{p}_2)^2 \cdots (\dot{p}_m)^2]^T.
\]

Taking into account (13.3), the torque vector becomes a time function parametrized by \(h\), so that we define, congruently with (13.4)

\[
\tau^i(t; h) := M(p^i(t; h)) \dddot{p}^i(t; h) + B(p^i(t; h))[\dot{p}] \dddot{p}^i(t; h) + D(p^i(t; h))[\dddot{p}] + g(p^i(t; h)), \quad i = 1, \ldots, n, \ t \in [0, h_i].
\]

An explicit relation between the joint variables and the Cartesian velocities of the end-effector can be expressed by means of the “geometric” Jacobian matrix \(J(p)\) [33] as follows:

\[
\begin{bmatrix}
\dot{v} \\
\omega
\end{bmatrix} = J(p) \ddot{p}
\]

(13.6)
Mechanical manipulators: optimal trajectory planning

where \( \mathbf{v} := [v_x \ v_y \ v_z]^T \) and \( \mathbf{\omega} := [\omega_x \ \omega_y \ \omega_z]^T \) are, respectively, the linear and the rotational velocities vectors of the tool frame affixed to the manipulator end-effector. The joint trajectories are time functions parametrized by \( h \), so that (13.6) can be rewritten as

\[
\begin{bmatrix}
\mathbf{v}^i(t; \mathbf{h}) \\
\mathbf{\omega}^i(t; \mathbf{h})
\end{bmatrix} := J(p^i(t; \mathbf{h}))\dot{p}^i(t; \mathbf{h}),
\]

\[ t \in [0, h_i], \quad i = 1, 2, \ldots, n. \quad (13.7) \]

Hence, a minimum-time movement under torque and tool velocity constraints can be planned by solving the following semi-infinite optimization problem

\[
\min_{\mathbf{h} \in \mathcal{B}} \sum_{i=1}^{n} h_i
\]

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

\[
|\mathbf{\tau}^i(t; \mathbf{h})| \leq \mathbf{\hat{\tau}} \quad \forall t \in [0, h_i],
\]

\[
\|\mathbf{v}^i(t; \mathbf{h})\| \leq \mathbf{\hat{v}} \quad \forall t \in [0, h_i],
\]

\[
\|\mathbf{\omega}^i(t; \mathbf{h})\| \leq \mathbf{\hat{\omega}} \quad \forall t \in [0, h_i].
\]

(13.9)–(13.11)

where \( \mathbf{\hat{\tau}} := [\mathbf{\hat{\tau}}_1 \ \mathbf{\hat{\tau}}_2 \ \ldots \ \mathbf{\hat{\tau}}_m]^T \in \mathbb{R}^+^m \) is the vector of the imposed torque limits while \( \mathbf{\hat{v}} \in \mathbb{R}^+ \) and \( \mathbf{\hat{\omega}} \in \mathbb{R}^+ \) are the linear and the angular velocity limits respectively. Our aim is to find an estimated global minimizer of problem (13.8) hereinafter denoted by \( \mathbf{h}^* \in \mathcal{B} \).

3. A FEASIBILITY RESULT.

Problem (13.8) will be solved by means of an hybrid algorithm for global optimization (briefly described in the next sections). It is important to verify if a feasible solution exits. For this reason, a partial feasibility result related to problem (13.8) is given in the following.

Proposition 1 Let the initial and final velocities and accelerations be zeros \( (\dot{\mathbf{q}}^0 = \ddot{\mathbf{q}}^0 = 0, \quad \dot{\mathbf{q}}^n = \ddot{\mathbf{q}}^n = 0) \). Then problem (13.8) admits a solution if

\[
\mathbf{\hat{\tau}}_k > \max_{\mathbf{p} \in \mathcal{P}} |g_k(\mathbf{p})|, \quad k = 1, 2, \ldots, m.
\]

(13.12)

where \( g_k(\mathbf{p}) \) denotes the \( k \)-th component of the gravity vector \( \mathbf{g}(\mathbf{p}) \).

Proof: Problem (13.8) 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 (13.9)–(13.11).
First, we consider constraint (13.9). The first and the last equality of (13.2) permit expressing \( q_k^1 \) and \( q_k^{n-1} \) as functions of the sole unknown \( \dot{q}_k^1 \) and \( \dot{q}_k^{n-1} \) respectively, according to the following expressions

\[
q_k^1 = \frac{(6q_k^0 + 4h_1q_k^0 + 2h_1\dot{q}_k^1 + \dot{q}_k^2h_1^2)}{6}, \quad (13.13)
\]

\[
q_k^{n-1} = \frac{(6q_k^0 - 4h_nq_k^n - 2h_n\dot{q}_k^{n-1} + \dot{q}_k^n h_n^2)}{6}. \quad (13.14)
\]

By replacing (13.13) and (13.14) into the remaining equations of system (13.2), we obtain a reduced linear system of order \( n - 1 \) that can be compactly written as

\[
A(h)x = b(h), \quad (13.15)
\]

where \( x := [\dot{q}_k^1\; \dot{q}_k^2\; \cdots\; \dot{q}_k^{n-1}]^T \) is the reduced vector of the unknowns while \( A(h) \in \mathbb{R}^{(n-1) \times (n-1)} \) is a proper tridiagonal matrix that depends only on vector \( h \).

Choose a point \( h \in B \) defined as \( h := \lambda = [\lambda\; \lambda\; \cdots\; \lambda]^T \) where \( \lambda \) is a positive real parameter. In the following we will show that \( \lambda \) is feasible for a sufficiently large \( \lambda \). Consequently with the choice of \( h \) and taking into account that initial velocities and accelerations have been set equal to zero, matrices \( A(h) \) and \( b(h) \) become

\[
A(\lambda) = \\
\begin{bmatrix}
8\lambda^2 & 2\lambda^2 & 0 & & & & 0 \\
4\lambda^3 & 8\lambda^3 & 2\lambda^3 & 0 & & & & \\
0 & 2\lambda^3 & 8\lambda^3 & 2\lambda^3 & 0 & & & \\
& & & & & & & \\
0 & & & & & & & \\
0 & & & & & & & \\
0 & & & & & & & \\
0 & & & & & & &
\end{bmatrix}
\]

\[
b(\lambda) = \\
\begin{bmatrix}
6\lambda(q_k^2 - q_k^0) \\
6\lambda^2(q_k^3 - q_k^0) \\
6\lambda^2(q_k^4 - q_k^0) \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
6\lambda^2(q_k^{n-1} - q_k^{n-2}) \\
6\lambda^2(q_k^{n-2} - q_k^{n-4}) \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
& & & & & & & \\
6\lambda(q_k^n - q_k^{n-2})
\end{bmatrix}
\]
By scrutiny of system (13.15) for the chosen value of \( h \) we have

\[
\dot{\ddot{\lambda}}_k(t; \lambda) = \frac{P_{ki} \lambda^{3n-6}}{\det[A(\lambda)]} = \frac{P_{ki} \lambda^{3n-6}}{C_n \lambda^{3n-5}} = \frac{P_{ki}}{C_n \lambda}, \quad i = 1, \ldots, n-1, \quad k = 1, \ldots, m,
\]

(13.16)

where \( C_n \in \mathbb{N} \) depends only on \( n \) and \( P_{ki} \in \mathbb{R} \). Hence, it follows that

\[
\lim_{\lambda \to \infty} |\dot{\ddot{\lambda}}_k(\lambda)| = 0, \quad i = 1, \ldots, n-1, \quad k = 1, 2, \ldots, m.
\]

(13.17)

From (13.1), joint velocities and accelerations are given by \((i = 1, \ldots, n, \quad k = 1, 2, \ldots, m, \text{ and } t \in [0, \lambda])\)

\[
\dot{\lambda}_i(t; \lambda) = \dot{q}_i(t) + \left[ \frac{6}{\lambda^2} (\dot{q}_i - \ddot{q}_i - 1) - \frac{2}{\lambda} (\ddot{q}_i + 2\dot{q}_i) \right] \quad \text{and} \quad \lambda_i(t; \lambda) = \left[ \frac{12}{\lambda^2} (\dot{q}_i - \ddot{q}_i - 1) + \frac{6}{\lambda^2} (\ddot{q}_i + 2\dot{q}_i) \right] t.
\]

(13.18)

(13.19)

By virtue of (13.17) and above expressions (13.18) and (13.19) for any given \( \varepsilon \in \mathbb{R}^+ \) there exists \( \lambda = \lambda(\varepsilon) \in \mathbb{R}^+ \) such that \((i = 1, \ldots, n)\)

\[
\|\dot{\lambda}_i(t; \lambda)\| < \varepsilon \quad \forall t \in [0, \lambda],
\]

(13.20)

\[
\|\lambda_i(t; \lambda)\| < \varepsilon \quad \forall t \in [0, \lambda].
\]

(13.21)

Taking into account the boundedness, over \( \mathcal{P} \), of matrices \( M(p), B(p), \text{ and } D(p) \) we have \((i = 1, \ldots, n \text{ and } t \in [0, \lambda])\)

\[
\|M(p(t; \lambda))\dot{\lambda}_i(t; \lambda) + B(p(t; \lambda))\lambda_i(t; \lambda)\dot{\lambda}_i(t; \lambda) + D(p(t; \lambda))(\dot{\lambda}_i(t; \lambda))\| \leq \underline{M} \|\dot{\lambda}_i(t; \lambda)\| + \underline{B} \|\dot{\lambda}_i(t; \lambda)\| + \underline{D} \|\dot{\lambda}_i(t; \lambda)\|^2.
\]

(13.22)

where \( \underline{M}, \underline{B}, \text{ and } \underline{D} \) are the appropriate real positive bounds for the matrices involved. From (13.20) and (13.21) we obtain

\[
\underline{M} \|\dot{\lambda}_i(t; \lambda)\| + \underline{B} \|\dot{\lambda}_i(t; \lambda)\| + \underline{D} \|\dot{\lambda}_i(t; \lambda)\|^2 < \underline{M} \varepsilon + \underline{B} \frac{m(m-1)}{2} \varepsilon^2 + \underline{D} m \varepsilon^2
\]

\[
= \left( \frac{\underline{M} + \underline{B} \frac{m(m-1)}{2} \varepsilon + \underline{D} m \varepsilon}{\varepsilon + \gamma} \right) \varepsilon =: \gamma(\varepsilon) \varepsilon.
\]

(13.23)
From (13.22) and (13.23) this inequality follows
\[ \| M(p^i(t; \lambda)) \dot{p}^i(t; \lambda) + B(p^i(t; \lambda)) \ddot{p}^i(t; \lambda) \| + D(p^i(t; \lambda)) \| \dot{p}^i(t; \lambda) \|_2 \| < \gamma(\varepsilon) \varepsilon. \] (13.24)

By considering definition (13.4) we deduce \((i = 1, \ldots, n \text{ and } t \in [0, \lambda])\)
\[ |\tau^i(t; \lambda)| < \varepsilon \| \dot{\gamma}(\varepsilon) \cdots \dot{\gamma}(\varepsilon) |^T + |g(p^i(t; \lambda))|. \] (13.25)

We can write in turn
\[ |\tau^i(t; \lambda)| < \varepsilon \begin{bmatrix} \gamma(\varepsilon) \\ \max_{p \in \mathcal{P}} |g_1(p)| \\ \vdots \\ \gamma(\varepsilon) \\ \max_{p \in \mathcal{P}} |g_m(p)| \end{bmatrix}. \] (13.26)

The hypothesis (13.12) permits choosing \( \varepsilon \) such that
\[ \gamma(\varepsilon) \varepsilon < \min_{k=1, \ldots, m} \left( \dot{\tau}_k - \max_{p \in \mathcal{P}} |g_k(p)| \right). \] (13.27)

Therefore
\[ \gamma(\varepsilon) \varepsilon + \max_{p \in \mathcal{P}} |g_k(p)| < \dot{\tau}_k \quad k = 1, \ldots, m, \] (13.28)
and, from (13.26), we finally conclude that \( \lambda \) is a feasible point for constraint (13.9) since
\[ |\tau^i(t; \lambda)| < \dot{\tau} \quad \forall t \in [0, \lambda], \quad i = 1, \ldots, n. \] (13.29)

Analogously, it is possible to prove that \( \lambda \) is a feasible point also for the other two constraints. We assume the premises made for constraint (13.9) to be valid, so that inequality (13.20) holds for any given \( \varepsilon > 0 \).

The jacobian matrix \( J(p) \) can be divided into two parts to separate the linear and the rotational components of the velocity vector so that (13.7) can be rewritten as \((i = 1, 2, \ldots, n \text{ and } t \in [0, \lambda])\)
\[ \begin{bmatrix} v^i(t; \lambda) \\ \omega^i(t; \lambda) \end{bmatrix} := \begin{bmatrix} J^i(p^i(t; \lambda)) \\ J''^i(p^i(t; \lambda)) \end{bmatrix} \dot{p}^i(t; \lambda). \] (13.30)

For any real robotic manipulator all the elements of the jacobian matrix are bounded for any \( p \in \mathcal{P} \). By virtue of this property, appropriate norm bounds \( \overline{J} \) and \( \overline{J''} \) of matrices \( J^i(p) \) and \( J''^i(p) \) can be found for any \( p \in \mathcal{P} \) so that we can correctly write \((i = 1, 2, \ldots, n \text{ and } t \in [0, \lambda])\)
\[ \| v^i(t; \lambda) \| = \| J^i(p^i(t; \lambda)) \dot{p}^i(t; \lambda) \| \leq \overline{J} \| \dot{p}^i(t; \lambda) \|, \] (13.31)
By applying (13.20), equations (13.30) become

\[ \|v^i(t; \lambda)\| < J_1 \varepsilon \]
\[ \|\omega^i(t; \lambda)\| < J_2 \varepsilon \]

and choosing \( \varepsilon \) such that the following two inequalities hold simultaneously

\[ J_1 \varepsilon \leq \dot{v} \]
\[ J_2 \varepsilon \leq \dot{\omega} \]

it is possible to conclude that \( \lambda \) is feasible because \((i = 1, 2, \ldots, n\) and \( t \in [0, \lambda] \))

\[ \|v^i(t; \lambda)\| < \dot{v} \]
\[ \|\omega^i(t; \lambda)\| < \dot{\omega} . \]

If \( \varepsilon \) satisfies simultaneously inequalities (13.27) and (13.33), the corresponding \( \lambda \) is a feasible point for the constrained problem (13.8). □

4. PROBLEM SOLUTION USING AN HYBRID ALGORITHM.

It is worth noting that (13.8) is formally a generalized semi-infinite problem. In fact, it can be recasted as

\[ \min_{h \in B} f(h) \]
\[ \text{subject to} \]
\[ g_j(r; h) \leq 0 \quad \forall r \in \mathcal{R}_j(h) , j = 1, 2, \ldots, u . \]

where \( g_j(r; h) \) are continuously differentiable constraint functions and \( \mathcal{R}_j(h) \) denotes, in general, a multi-dimensional compact interval that depends on the search parameter \( h \). For the problem at hand, the number of functions \( g_j(r; h) \) is directly correlated to the number of joints and time intervals of the problem. Taking into account that each torque constraint (13.9) must rewritten as \((i = 1, 2, \ldots, n)\)

\[ \tau^i(t; h) \leq \dot{\tau} \quad \forall t \in [0, h_i] , \]
\[ -\tau^i(t; h) \leq \dot{\tau} \quad \forall t \in [0, h_i] , \]

in order to get differentiable functions, the total number of constraints is \( u = n(2m + 2) \).

The notation adopted in (13.35) is useful to show that the proposed hybrid algorithm can solve problems more general than (13.8)–(13.11), i.e. problems with \( r \) being a real vector and \( f(h) \) being a generic nonlinear function that
does not even need to be a continuous function. In the mathematical literature nonlinear semi-infinite optimization has been treated with a variety of approaches, for example generalized gradient procedures, recursive nonlinear programming, etc. Good sources on the subject with extensive bibliography are the book of Polak [29] and the surveys of Hettich and Kortanek [13] and of Reemtsen and Görner [31]. Specific algorithms that can solve (13.35) for the case of mono-dimensional constraints (i.e. \( R_j(h) \) is a compact real interval) that is actually our case for the minimum-time problem (13.8)–(13.11) were presented by Jennings and Teo [15] and Teo et al. [37]. Both algorithms using a constraint transcription method based on an integral representation generate a converging sequence of finite optimization problems. These are equality constraints problems for [15] and penalty-based unconstrained problems in [37] that intentionally rely on standard nonlinear programming, i.e. deterministic local optimization. In this section we sketch a numerical approach to problem (13.35) based on the combined use of stochastic and deterministic global optimization (cf. the last paragraphs of §5 for a comparison with [37]).

By defining
\[
\sigma_j(h):=\max_{r \in R_j(h)} \{ b_j(r;h) \},
\]
problem (13.35) can be converted into an unconstrained problem by using the penalty method
\[
\min_{h \in B} \left\{ f(h) + \sum_{j=1}^n \Phi(\sigma_j(h)) \right\},
\]  
(13.36)
where the penalty function \( \Phi(\sigma) \) is defined as
\[
\Phi(\sigma) := \begin{cases} 
0 & \text{if } \sigma \in (-\infty,0] \\
M - M(T - \sigma)^2/T^2 & \text{if } \sigma \in (0,T] \\
M & \text{if } \sigma \in (T, +\infty)
\end{cases}.
\]  
(13.37)
When \( T \to 0^+ \) and \( M \to +\infty \) problem (13.36) is strictly equivalent to (13.35). For finite values of \( M > 0 \) and \( T > 0 \), monotonically better precisions are obtained for larger values of \( M \) and smaller values of \( T \).

The equivalent optimization problem (13.36) is solved by means of an hybrid genetic/interval algorithm originally proposed in [6, 10]. An interval procedure is used to evaluate the penalty terms of (13.36), while a genetic algorithm is used to find the estimated global minimizer \( h^* \).

The interval procedure is a special branch-and-bound algorithm based on the principles of interval analysis (an extension of the standard real analysis over the arithmetics of real intervals [24]). Broadly speaking, interval procedures (algorithms) are deterministic routines that can be applied to linear or nonlinear optimization problems, assuring global convergence within an arbitrarily prespecified precision [12] [30].

With the aim of improving the computational efficiency, our interval procedure does not determine the exact value of \( \sigma_j(h) \), but is tailored to directly
compute $\Phi(\sigma_j(\mathbf{h}))$ with the help of special accelerating devices to speed up the convergence. This developed procedure can be considered as a generalization of the interval positivity test presented in [26].

According to (13.36), penalty terms evaluated by the interval algorithm, are added to $f(\mathbf{h})$ to obtain the overall objective function. The resulting unconstrained optimization problem is solved with a genetic algorithm, i.e. a stochastic technique that can deal with a variety of optimization tasks [5, 22]. This choice is justified by the need of a relatively fast procedure, again with the aim of limiting the computational time. A partially elitistic algorithm has been implemented so that, at each generation, the final population is composed by individuals randomly drawn from the previous generation mixed with individuals of the so called offspring population. The genetic algorithm uses a two-phases technique to approach the feasibility region [32]: until the feasible region is reached the objective function is exclusively given by the penalty terms (i.e. the cost index is ignored), later the whole objective function is considered. To increase the convergence rate, procedures for the local improvement of the best individual of the current population have been introduced. The elaborated genetic algorithm is a variant of the one presented in [21] and is described with details in [10].

The estimated global minimizer for the optimization problem (13.36) 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 on the tool Cartesian velocities.

5. **PENALTY COMPUTATION VIA INTERVAL ANALYSIS.**

Penalty terms for the equivalent unconstrained problem are computed with an interval procedure. The interval algorithm evaluate directly $\Phi(\sigma(\mathbf{h}))$ (for simplicity in this section the subscript $j$ is dropped). It is an improvement of the procedure proposed in [10]: the program has been modified to accelerate the convergence toward the solution. In the following, the improved algorithm is exposed and its convergence properties are investigated. A generic semi-infinite constraint function $g(\mathbf{r}; \mathbf{h})$ is considered.

Define the set of real intervals as $I := \{ [a, b] : a, b \in \mathbb{R}, a \leq b \}$. In the following $\mathcal{R}, \mathcal{D} \subseteq \mathbb{R}^m$ are used to denote finite multidimensional real intervals or “boxes”: for example, $\mathcal{D} := [d_1, \bar{d}_1] \times [d_2, \bar{d}_2] \times \ldots \times [d_m, \bar{d}_m]$. A midpoint of a box $\mathcal{D}$ is a vector whose $i$-th component is given by $(d_i + \bar{d}_i)/2$; it individuates the “center” of a box and will be indicated as $\text{mid}(\mathcal{D}) \in \mathcal{D}$. It is usual to indicate the “width” of $\mathcal{D}$ as the widest edge of the box: $w(\mathcal{D}) := \max_{i=1, \ldots, m} \{ \bar{d}_i - d_i \}$. 
For each time the interval procedure is invoked by the genetic algorithm \( h \) acts as a fixed parameter. For this reason the semi-infinite constraint function can be denoted in a compact way as \( g_h(r) \equiv g(r; h) \) and the search box is indicated by \( \mathcal{R}_h \equiv \mathcal{R}(h) \). The image set of \( \mathcal{R}_h \) under function \( g_h(r) \) is denoted by
\[
\mathcal{g}_h(\mathcal{R}_h) := \{ y \in \mathbb{R} : y = g_h(r), r \in \mathcal{R}_h \}.
\]
The continuity of \( g_h(r) \) implies that \( g_h(\mathcal{R}_h) \in I \). The global maximum value of \( g_h \) over \( \mathcal{R}_h \) be denoted by \( g_h^* := \sigma(h) = \max_{r \in \mathcal{R}_h} \{ g_h(r) \} \); \( g_h^* \) is the upper endpoint of \( g_h(\mathcal{R}_h) \). Moreover \( \mathcal{R}_h^* := \{ r^* \in \mathcal{R}_h : g_h(r^*) = g_h^* \} \) denotes the set of global maximizers which may have cardinality greater than one. Lower and upper bounds of \( g_h^* \) are denoted by \( l_b \) and \( u_b \) respectively. \( K_{\text{pre}} \in \mathbb{N} \) denotes the precision factor to be used by the termination test of the interval procedure.

5.1 INCLUSION FUNCTIONS.

An inclusion function with respect to \( g_h \) is an interval-valued function \( G_h : \{ D : D \subseteq \mathcal{R}_h \} \rightarrow I \) satisfying:
\[
\mathcal{g}_h(D) \subseteq G_h(D) \quad \forall D \subseteq \mathcal{R}_h.
\]
Once an inclusion function is known, an upper bound of the global maximum of \( g_h \) over \( D \subseteq \mathcal{R}_h \), denoted by \( \text{ub}(g_h, D) \), can be easily determined as the upper endpoint of \( G_h(D) \). Interval analysis is a straightforward tool to get a variety of inclusion functions. The simplest of these is the so-called natural interval extension. Roughly speaking, it is obtained by evaluating a given form of \( g_h \) with the substitution of the usual arithmetic with the interval arithmetic.

This is summarized as follows:
\[
[a, b] + [c, d] = [a + c, b + d] \\
[a, b] - [c, d] = [a - d, b - c] \\
[a, b] \cdot [c, d] = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}] \\
[a, b] / [c, d] = [\min\{ac, ad, bc, bd\} / d, \max\{ac, ad, bc, bd\} / c] \quad \text{if} \quad 0 \notin [c, d]
\]
As an example consider \( g_h(r) = r_1^2 + 5r_1r_2 + r_2 - r_3^2 \) and \( D = [-1, 1] \times [2, 4] \). Hence, by using the natural interval extension we have \( G_h(D) = [-1, 1]^2 + 5 \cdot [-1, 1] \cdot [2, 4] + [2, 4] - [-1, 1]^3 = [0, 1] + 5 \cdot [-4, 4] + [2, 4] - [-1, 1] = [-19, 26] \).

Other noteworthy inclusion functions are the “meanvalue forms” and “Taylor forms”; both forms belong to the class of “centered forms” introduced by [23]. For the purpose of the interval procedure to follow, the optimal meanvalue form of Baumann has been chosen as inclusion function for its sharp bounds and moderate computational burden. An introduction to inclusion functions with details on the Baumann form can be found in [30].
5.2 THE INTERVAL PROCEDURE.

By making use of operators borrowed from interval analysis, an algorithm for the evaluation of the penalty function $\Phi(\sigma(h))$ has been developed. It is an improved version of the algorithm appeared in [10]. More precisely, a more versatile management of the List is obtained and the algorithm is strongly tailored to the direct evaluation of $\Phi(\sigma(h))$.

1. Locally maximize the function $g_h(r)$ with starting point $\text{mid}(R_h)$ to obtain $\tilde{r}$ and set $l_q := g_h(\tilde{r})$. [$\tilde{r} \in R_h$ and $g_h(\tilde{r}) \geq g_h(\text{mid}(R_h))$]
2. Set $u_b = +\infty$.
3. If $u_b(g_h, R_h)) > 0$ initialize the list List inserting the pair $(R_h, u_b(g_h, R_h))$.
4. If $l_q \geq T$ then set $\Phi(\sigma(h)) := M$ and terminate.
5. If List is empty then set $\Phi(\sigma(h)) := 0$ and terminate.
7. If $(u_b - l_q) \leq T/K_{pre}$ then set $\Phi(\sigma(h)) := \Phi ((l_q + u_b)/2)$ and terminate.
8. The first pairs of List is popped out and its box, by halving on the largest edge, is split into boxes $D_1$ and $D_2$. 
9. For $i := 1, 2$ do
   If $g_h(\text{mid}(D_i)) > l_q$ then locally maximize the function $g_h(r)$ with starting point $\text{mid}(D_i)$ to obtain $\tilde{r}$ and set $l_q := g_h(\tilde{r})$. [$\tilde{r} \in R$ and $g_h(\tilde{r}) \geq g_h(\text{mid}(D_i))]$.
10. For $i := 1, 2$ do
    If $\text{ub}(g_h, D_i) \geq l_q$ and $\text{ub}(g_h, D_i) > 0$ then insert pair $(D_i, \text{ub}(g_h, D_i))$ in List, in such a way that the second elements of the pairs be placed in decreasing order.
11. Discard from List, without perturbing the decreasing order, any pair such that its second element is smaller than $l_q$.
13. End.

The exposed procedure is based on the branch-and-bound principle where the bounding is done via inclusion functions and the branching is done by splitting
the box which has the largest upper bound. In such a way it emerges, at the
core of the procedure, an interval algorithm which can compute $\Phi(\sigma(h))$
with arbitrary precision.

At steps 1 and 9 a local maximization is simply performed with the steepest
ascent method [19, see p. 214]. This accelerates the procedure convergence
because it helps discarding portions of $\mathcal{R}_h$ not containing global maximizers
(cf. step 11).

5.3 CONVERGENCE ANALYSIS

The considered assumption of $g_h(r)$ being continuously differentiable on
the compact $\mathcal{R}_h$ implies that $g_h(r)$ is Lipshitzian as well as continuous on $\mathcal{R}_h$.
The following properties are essential to establish the main result of this section
(Proposition 2).

**Property 1** For any meanvalue form the following limit holds uniformly for
$D \subseteq \mathcal{R}_h$:

$$\lim_{w(D) \rightarrow 0} \text{ub}(g_h, D) = \max_{r \in D} \{g_h(r)\}.$$  \hspace{1cm} (13.38)

**Property 2** The following limit holds uniformly for $D \subseteq \mathcal{R}_h$:

$$\lim_{w(D) \rightarrow 0} g_h(\text{mid}(D)) = \max_{r \in D} \{g_h(r)\}.$$  \hspace{1cm} (13.39)

Property 1 is a well known result in the interval analysis literature, cf. for
example [30], and Property 2 is an obvious consequence of the continuity of
function $g_h(r)$ over $\mathcal{R}_h$. In the following, considering the interval procedure,
we denote by $l_b^{(i)}$ and $u_b^{(i)}$ the values of variables $l_b$ and $u_b$ at step 4 of the $i$-th
iteration.

**Lemma 1** At any stage of iterations, the above interval procedure guarantees
that $g_h^* \in [l_b^{(i)}, u_b^{(i)}]$.

**Proof:** By virtue of steps 1 and 9, $l_b^{(i)}$ is the maximum of all the $g_h$ function
values computed till the $i$-th iteration. Obviously, the global maximum of $g_h$
must be greater than or equal to the current $l_b$ so that inequality $g_h^* \geq l_b^{(i)}$ is
ensured.
At the \(i\)-th iteration the list \(\text{List}\) be composed as

\[
\text{List} = \left\{ (D_1^{(i)}, \text{ub}(g_h, D_1^{(i)})), (D_2^{(i)}, \text{ub}(g_h, D_2^{(i)})), \ldots, (D_h^{(i)}, \text{ub}(g_h, D_h^{(i)})) \right\}.
\]

Steps 3 and 10 guarantee that, at any iteration \(i\), the \(\text{List}\) is composed by boxes such that \(\text{ub}(g_h, D_j^{(i)}) > 0\), \(j = 1, 2, \ldots, h\) where \(h\) indicates the current list length. By considering also steps 2 and 6 we conclude that \(u_b^{(i)} > 0\). Two situations could arise depending on the sign of the global maximum of \(g^*_h\). Let suppose that \(g^*_h > 0\). In this case, all the global maximizers are contained into boxes of \(\text{List}\) at any stage of iterations. In fact, there are only two conditions that permit discarding a generic box \(D\): \(\text{ub}(g_h, D) \leq 0\) or \(\text{ub}(g_h, D) \leq l_b^{(i)}\) (see steps 3, 10 and 11). In both instances we deduce \(\text{ub}(g_h, D) < g^*_h\) so that \(D\) does not contain global maximizers. All boxes are inserted into the \(\text{List}\) such that their related upper bounds are placed with decreasing order (cf. steps 10 and 11). Hence

\[
\text{ub}(g_h, D_1^{(i)}) \geq \text{ub}(g_h, D_2^{(i)}) \geq \text{ub}(g_h, D_3^{(i)}) \geq \ldots \geq \text{ub}(g_h, D_h^{(i)}) , \quad (13.40)
\]

and \(u_b^{(i)} = \text{ub}(g_h, D_1^{(i)})\) owing to step 6. Evidently, the upper bound ordering implies that \(u_b^{(i)} \geq g^*_h\) with certainty.

Now let suppose that \(g^*_h \leq 0\). In this case, being always verified that \(u_b^{(i)} > 0\), it is straightforward to conclude that \(u_b^{(i)} > g^*_h\).

\(\square\)

**Proposition 2** For any \(T \in \mathbb{R}^+\) and \(K_{\text{pre}} \in \mathbb{N}\) the above interval procedure converges with certainty and computes \(\Phi(\sigma(h))\) with arbitrarily good precision.

**Proof:** The proof of Proposition 1 is divided into two parts: first we prove that, if the algorithm converges, it returns the correct value of \(\Phi(\sigma(h))\) with arbitrarily good precision; secondly, it is shown that convergence is guaranteed.

First part:

The interval procedure can stop owing to one of the three termination criteria (steps 4, 5, and 7).

If the algorithm stops because of step 4, Lemma 1 permits asserting that \(g^*_h \geq T\) so that, according with (13.37), the exact value of \(M\) for \(\Phi(\sigma(h))\) is provided.

If the algorithm stops because of step 5, the given output is correct only if \(g^*_h \leq 0\). This can indeed be proved. If the \(\text{List}\) is empty, all the boxes have been eliminated. A generic box \(D\) can be discarded on account of the following conditions: (1) \(\text{ub}(g_h, D) \leq 0\), or (2) \(\text{ub}(g_h, D) < l_b^{(i)}\). Let indicate by \(I_1\) and \(I_2\) the union of all the boxes discarded because of inequality (1) and inequality
(2) respectively. Since the List is empty we have \( R_h = I_1 \cup I_2 \). By virtue of Lemma 1 and condition (2) we can immediately conclude that set \( I_2 \) cannot include any global maximizer, so that they are all within \( I_1 \). On the basis of condition (1), the global maximum of \( g_h \) over \( I_1 \) is not positive so that \( g^*_h \leq 0 \).

Finally, if the algorithm stops because of step 7 and taking into account Lemma 1, the global maximum \( g^*_h \) is estimated as the midpoint between \( l_b^{(i)} \) and \( u_b^{(i)} \). In this case, the final distance between \( l_b^{(i)} \) and \( u_b^{(i)} \) can be set arbitrarily small because \( T \) and \( K_{pre} \) can be freely selected. Remembering that \( g^*_h \in [l_b^{(i)}, u_b^{(i)}] \), we conclude that \( g^*_h \) and, consequently, \( \Phi(\sigma(h)) \) can be evaluated with arbitrarily good precision.

**Second part:**

Now we demonstrate that if the interval procedure does not halt at step 4 or 5 then, necessarily, it halts at step 7. The branching mechanism issued at Step 8, in absence of the exit tests 4, 5, and 7, permits writing

\[
\lim_{i \to \infty} w(D_1^{(i)}) = 0 .
\]  

(13.41)

Property 1 implies that for any given \( \varepsilon > 0 \) there exists \( \delta_u > 0 \) such that for any \( D_1^{(i)} \) satisfying \( w(D_1^{(i)}) < \delta_u \) it follows

\[
u_b(g_h, D_1^{(i)}) - \max_{r \in D_1^{(i)}} \{ g_h(r) \} < \varepsilon
\]

or, equivalently, by virtue of step 6

\[
u_b^{(i)} - \varepsilon < \max_{r \in D_1^{(i)}} \{ g_h(r) \} .
\]  

(13.42)

On the other hand, it is possible to deduce from Property 2 that, for any given \( \varepsilon > 0 \), there exists \( \delta_m > 0 \) such that for any \( D_1^{(i)} \) satisfying \( w(D_1^{(i)}) < \delta_m \) it follows

\[
\max_{r \in D_1^{(i)}} \{ g_h(r) \} - g_h \left( \text{mid}(D_1^{(i)}) \right) < \varepsilon ,
\]

\[
\max_{r \in D_1^{(i)}} \{ g_h(r) \} < \varepsilon + g_h \left( \text{mid}(D_1^{(i)}) \right) .
\]  

(13.43)

By considering that function \( g_h(r) \) is evaluated at the midpoint of every box inserted in the List (cf. steps 1 and 9), it is clearly verified \( g_h \left( \text{mid}(D_1^{(i)}) \right) \leq l_b^{(i)} \). Then from inequality (13.43) descends

\[
\max_{r \in D_1^{(i)}} \{ g_h(r) \} < \varepsilon + l_b^{(i)} .
\]  

(13.44)
Clearly limit (13.41) implies that \( w(D_1^{(i)}) \) can become arbitrarily small for sufficiently large values of \( i \). For this reason, it is possible to find an \( i^* \in \mathbb{N} \) such that for any \( i \geq i^* : w(D_1^{(i)}) < \min \{ \delta_u, \delta_m \} \). Finally, from (13.42) and (13.44), it is obtained
\[
 u_b^{(i)} - l_b^{(i)} < 2\varepsilon \quad \forall i \geq i^*
\]
and, imposing \( \varepsilon = T/(2K_{pre}) \), the above inequality becomes
\[
 u_b^{(i)} - l_b^{(i)} < T/K_{pre} \quad \forall i \geq i^*. \quad (13.45)
\]
The interval procedure halts at step 7 not exceeding iteration \( i^* \) by virtue of statement (13.45).

It is worth comparing the proposed genetic/interval algorithm with the penalty-based constraint transcription algorithm of Teo, Rehbock, and Jennings [37]; see more details on the related enforced smoothing constraint transcription in the book of Teo et al. [36] that also reports a general treatment of semi-infinite control problems.

Both approaches use a penalty method to transform the original semi-infinite constrained problems into finite unconstrained problems (cf. the second paragraph at the beginning of the §4). The unconstrained objective function for the Teo-Rehbock-Jennings (TRJ) algorithm is smooth (i.e. the associated gradient is continuous) by virtue of the devised smoothing technique whereas the unconstrained function (13.36) is not smooth. On the other hand, the TRJ algorithm needs smoothness in the unconstrained problem because it is then solved with a gradient method whereas the proposed algorithm using a genetic algorithm does not necessitate to compute the gradient.

The genetic/interval algorithm can be intrinsically used for generalized problems where \( \mathcal{R}_j(h) \) is a multi-dimensional box that can explicitly depend on \( h \) while on the contrary the TRJ algorithm only deals with mono-dimensional constraints. Moreover, the TRJ algorithm nominally requires a real interval \( \mathcal{R}_j(h) \) not depending on \( h \) so that some artifices should be applied in order to extend the approach to generalized problems.

A crucial aspect of the comparison regards the feasibility of the solutions provided by the algorithms. The genetic/interval algorithm guarantees the feasibility of the found minimizer because it relies on a deterministic global method (the interval procedure) to compute the penalty terms. Applying the TRJ algorithm, feasibility can not ensured with certainty because the penalty computation uses a quadrature discretization in evaluating the related integral term. As a consequence in case of harsh constraints containing positive spikes feasibility could be erroneously claimed.

From a computational viewpoint the TRJ algorithm provides definitely better computation times than the genetic/interval algorithm. This is basically due to
the reasons: (i) the efficiency of the gradient method of TRJ algorithm to reach a local solution compared to the relatively heaviness of the genetic algorithm to reach an estimated global solution; (ii) the fast computation of the penalty terms using the quadrature points in the TRJ algorithm compared to the slower branch-and-bound interval procedure of the proposed algorithm.

6. AN EXAMPLE.

The example proposed concerns a two-link \((m = 2)\) mechanical arm with revolute joints. The problem is the minimum-time planning under torque and velocities constraints of a trajectory whose Cartesian path is, for example, schematically shown in Fig. 13.1.

The distal end of the second link is required to exactly cross some given via points \((s = 10, \text{so that } n = 11)\) on the Cartesian path. These via points are expressed as Cartesian coordinates of the arm base frame (first two columns in Tab. 13.1). By solving the inverse kinematics problem the Cartesian via points are converted into joint via points and reported in the last two columns of Tab. 13.1. Note that the second and the penultimate knot points have not been imposed, being associated to the two free joint displacements. The joint variable vector is \(p := [p_1 \ p_2]^T \in \mathcal{P} := [0, \pi/2] \times [-\pi, 0]\). Closed form dynamic equations were derived in [3, p. 204] under the hypothesis of masses concentrated at the distal end of each link

\[
\tau_1 := m_2 l_2^2 (\ddot{p}_1 + \ddot{p}_2) + m_2 l_1 l_2 \cos(p_2)(2\ddot{p}_1 + \dddot{p}_2) +
\]

\[
\tau_2 := m_1 l_1 (\ddot{q}_1 + \dddot{q}_2) + m_1 l_1 l_2 \cos(q_2)^2(2\ddot{q}_1 + \dddot{q}_2) +
\]

\[
m_2 l_2 \dddot{p}_1 - m_1 \dddot{q}_1 = 0
\]

\[
m_2 l_2 \dddot{p}_2 - m_1 \dddot{q}_2 = 0
\]
Table 13.1  End-effector via points expressed in meters and equivalent joint via points expressed in radians.

<table>
<thead>
<tr>
<th>x0</th>
<th>y0</th>
<th>q0 1</th>
<th>q0 2</th>
<th>q0 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>-0.500</td>
<td>0.0000</td>
<td>1.5708</td>
<td></td>
</tr>
<tr>
<td>x1</td>
<td>y1</td>
<td>q1 1</td>
<td>q1 2</td>
<td>q1 3</td>
</tr>
<tr>
<td>1.00</td>
<td>0.375</td>
<td>0.1253</td>
<td>1.6804</td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>y2</td>
<td>q2 1</td>
<td>q2 2</td>
<td>q2 3</td>
</tr>
<tr>
<td>1.00</td>
<td>0.250</td>
<td>0.2517</td>
<td>1.7594</td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>y3</td>
<td>q3 1</td>
<td>q3 2</td>
<td>q3 3</td>
</tr>
<tr>
<td>1.00</td>
<td>0.125</td>
<td>0.3789</td>
<td>1.8074</td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td>y4</td>
<td>q4 1</td>
<td>q4 2</td>
<td>q4 3</td>
</tr>
<tr>
<td>1.00</td>
<td>0.000</td>
<td>0.5054</td>
<td>1.8235</td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td>y5</td>
<td>q5 1</td>
<td>q5 2</td>
<td>q5 3</td>
</tr>
<tr>
<td>1.05</td>
<td>0.375</td>
<td>0.1253</td>
<td>1.7087</td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td>y6</td>
<td>q6 1</td>
<td>q6 2</td>
<td>q6 3</td>
</tr>
<tr>
<td>1.15</td>
<td>0.200</td>
<td>0.6119</td>
<td>1.4581</td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td>y7</td>
<td>q7 1</td>
<td>q7 2</td>
<td>q7 3</td>
</tr>
<tr>
<td>1.30</td>
<td>0.200</td>
<td>0.6119</td>
<td>1.1040</td>
<td></td>
</tr>
<tr>
<td>x8</td>
<td>y8</td>
<td>q8 1</td>
<td>q8 2</td>
<td>q8 3</td>
</tr>
<tr>
<td>1.30</td>
<td>0.050</td>
<td>0.3903</td>
<td>1.1124</td>
<td></td>
</tr>
<tr>
<td>x9</td>
<td>y9</td>
<td>q9 1</td>
<td>q9 2</td>
<td>q9 3</td>
</tr>
<tr>
<td>1.30</td>
<td>0.000</td>
<td>0.3526</td>
<td>1.1152</td>
<td></td>
</tr>
</tbody>
</table>

Table 13.2  Parameters of the two-link arm.

<table>
<thead>
<tr>
<th>l1</th>
<th>l2</th>
<th>m1</th>
<th>m2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 m</td>
<td>0.5 m</td>
<td>15.0 kg</td>
<td>7.0 kg</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
(m_1 + m_2)l_1^2\dot{p}_1 &= m_2l_1l_2\sin(p_2)\dot{p}_2^2 - 2m_2l_1l_2\sin(p_2)\dot{p}_1\dot{p}_2 + m_2l_2g\cos(p_1 + p_2) + (m_1 + m_2)l_1g\cos(p_1), \\
\tau_2 &= m_2l_1l_2\cos(p_2)\dot{p}_1 + m_2l_1l_2\sin(p_2)\dot{p}_1^2 + m_2l_2g\cos(p_1 + p_2) + m_2l_2^2(\dot{p}_1 + \dot{p}_2),
\end{align*}
\]  

(13.46)

where \(m_1\) and \(m_2\) are the link masses, \(g\) is the gravity acceleration, and \(l_1\) and \(l_2\) are the link lengths. The arm parameters are listed in Tab. 13.2. Analogously, closed form equations were derived for the linear and angular velocities of the end-effector [3, p. 169]

\[
\|v\| := \sqrt{l_1^2\dot{p}_1^2 + l_2^2(\dot{p}_1 + \dot{p}_2)^2 + 2l_1l_2\dot{p}_1(\dot{p}_1 + \dot{p}_2)\cos(p_2)},
\]  

(13.47)

\[
\omega_z := \dot{p}_1 + \dot{p}_2.
\]  

(13.48)

Here, the vector of the interval times is defined as \(h := [h_1 \ h_2 \ \cdots \ h_{11}]^T \in B := [0.02, 1.0]^{11}\) (it has been fixed \(\nu = 0.02\) sec). Moreover, we consider the arm at rest in the initial and final positions (i.e. \(\dot{q}_0 = \dot{q}^{11} = 0; \ddot{q}_0 = \ddot{q}^{11} = 0\)). By virtue of Proposition 1, problem (13.8) admits a feasible solution if \(\hat{\tau} > \)
With the aim of planning the optimal trajectory with moderate actuator exertion, the torque limit vector is set to $\hat{\tau} = [260, 50]^T$ Nm, which is only slightly larger than its minimum admissible value.

The minimizer evaluated by the genetic/interval algorithm is given in Tab. 13.3. The estimated global minimum travelling time is $\sum_{i=1}^{11} h_i^* = 2.05009$ sec. Fig. 13.2 shows that, at the optimal solution, the joint 1 torque constraint is active into a wide time segment. Moreover, also the constraint on the maximum linear velocity modulus is active (see Fig. 13.3). Finally, the planned optimal trajectory in the Cartesian plane is shown in Fig. 13.4 where crosses indicates the assigned via points.

The genetic/interval algorithm has been coded in C++ and uses at the lower level the efficient PROFIL library [16].

7. CONCLUSIONS.

In this paper a new method has been presented for the minimum-time trajectory planning of mechanical manipulators. The method, using a joint space scheme with cubic splines, takes into account Cartesian velocity constraints and torque constraints by inclusion of a full manipulator dynamic model. The resulting problem is shown to be a semi-infinite nonlinear optimization problem for which an estimate of the global solution can be obtained by means of a genetic/interval algorithm. This estimate solution is guaranteed to be feasible due to the deterministic interval procedure used by the hybrid algorithm.

First computational results highlight the effectiveness of the method and suggest to apply and extent it in broader robotic planning contexts.
**Figure 13.3** Angular and Cartesian velocities of the end-effector.

**Table 13.3** Estimated global minimizer $h^*$ of the planned optimal trajectory.

<table>
<thead>
<tr>
<th>$h^*_1$</th>
<th>0.02000 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h^*_2$</td>
<td>0.36429 s</td>
</tr>
<tr>
<td>$h^*_3$</td>
<td>0.18419 s</td>
</tr>
<tr>
<td>$h^*_4$</td>
<td>0.18386 s</td>
</tr>
<tr>
<td>$h^*_5$</td>
<td>0.18423 s</td>
</tr>
<tr>
<td>$h^*_6$</td>
<td>0.16735 s</td>
</tr>
<tr>
<td>$h^*_7$</td>
<td>0.22310 s</td>
</tr>
<tr>
<td>$h^*_8$</td>
<td>0.36539 s</td>
</tr>
<tr>
<td>$h^*_9$</td>
<td>0.09945 s</td>
</tr>
<tr>
<td>$h^*_{10}$</td>
<td>0.23818 s</td>
</tr>
<tr>
<td>$h^*_{11}$</td>
<td>0.02005 s</td>
</tr>
</tbody>
</table>

### 8. ACKNOWLEDGEMENTS

This work was partially supported by MURST and ASI (Italian Space Agency)

### References


Figure 13.4  Optimal trajectory in the Cartesian space. Crosses indicate the assigned via points


