Proceedings of the 36th
Conference on Decision & Control
San Diego, California USA • December 1997

A Cutting-Plane Algorithm for Minimum-Time Trajectory Planning of Industrial Robots [1]

Aurelio Piazzi¶ Antonio Visioli§

¶ Dipartimento di Ingegneria dell'Informazione
University of Parma - Italy
e-mail: aurelio@ce.unipr.it

§ Dipartimento di Elettronica per l'Automazione
University of Brescia - Italy
e-mail: visioli@bsing.ing.unibs.it

Abstract

The optimal minimum-time trajectory planning of an m-joint industrial robot is proposed by means of a newly devised outer cutting-plane algorithm. By using piecewise cubic polynomials in the joint space, this algorithm provides a global solution to the minimum total time planning problem by adopting an interval subroutine, i.e. a procedure which uses concepts of interval analysis. An example for the 2-joint case with computational results is included.

1 Introduction and Problem Formulation

A practical simplified approach to the optimal control of industrial robots is to split the problem into an optimal trajectory planning followed by feedback trajectory tracking. Specifically a sequence of path points is mapped, via inverse kinematics, into a set joint angles/offsets (knots). These knots are then interpolated with smooth functions to be optimized subject to appropriate constraints for a specific robot application. The resulting joint trajectory forms the input to the robot's feedback control system. In this context Lin *et al.* [1] proposed cubic polinomial functions (splines) for a trajectory planning where the total travelling time is minimized under constraints on joint velocities, accelerations and jerks. They propose an optimization solver which is computationally efficient but only provides local solutions to the constrained optimization problem. The aim to achieve a true global solution in optimal trajectory planning was pursued by Simon [2] that proposed, in a similar context, a stochastic optimization method based on neural networks, in order to obtain a minimum jerk joint trajectory. In a previous paper [3], following the joint space scheme of [1], we proposed a deterministic global optimization approach based on an interval algorithm to obtain a global minimum-time trajectory subject to constraints on joint accelerations and jerks. The constraints on joint jerks are imposed with the aim to limit robot's vibrations and to increase robot's life-span. In this paper we repropone the problem of [3] in the special but important case in which both the initial and final velocities and accelerations are fixed to zero. By doing so we are able to present a novel cutting-plane algorithm which still providing a global solution as in [3] leads to a good computational improvement since the dimension of the problem domain is practically reduced from n to $n-1$. With this new approach, interval analysis is used to reveal if a local minimum, determined with a gradient-based method, is actually a global minimum. An introduction to interval analysis techniques can be found in the book of Moore [4] (the founder of interval analysis) and applications to general purpose global optimization are covered by Ratschek and Rokne [5].

Consider an m-joint robot manipulator with given $n-1$ interspaced points of the tool frame Cartesian path. By appropriate application of the inverse kinematics we obtain $n-1$ sets of joint positions so that for each joint we have a sequence of dispacements to be interpolated by piecewise cubic polynomials. All the initial and final joint velocities and accelerations are zeros. In this context, as is already known [1], for each joint two extra knots with free displacement have to be inserted in second and penultimate positions to assure an overall continuity of position (displacement), velocity, and acceleration. As a conseguence at the k-th joint the displacement sequence be described by $q_{k0}, \ldots, q_{kn}$ $(k = 1, \ldots, m)$ where q_{k1} and $q_{k,n-1}$ are free displacement parameters and all the others are to be considered given data. Joint velocity and acceleration at the i-th knot are respectively denoted by v_{ki} and a_{ki}. Denote by h_i $(i = 1, \ldots, n)$ the elapsed time necessary for the i-th spline $Q_{ki}(t)$ to connect knot $i-1$ to knot i with

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

$t \in [0, h_i]$. Once the h's have been fixed, the unknown coefficients in each spline $Q_{ki}(t)$ can be uniquely determined by imposing the continuity of displacement, velocity, and acceleration (see [6] for details). The total travelling time required to perform the robot task is evidently $\sum_{i=1}^{n} h_i$ and the optimal trajectory planning problem with minimum-time criterion can be posed as follows:

$$\min_{h \in \mathbb{R}^{+n}} \sum_{i=1}^{n} h_i \tag{1}$$

subject to $(k = 1, \ldots, m)$

$$|\ddot{Q}_{ki}(t)| \le a_{MAX_k} \quad \forall t \in [0, h_i] \;\; i = 1, \ldots, n-1;$$
$$|\dddot{Q}_{ki}(t)| \le j_{MAX_k} \quad \forall t \in [0, h_i] \;\; i = 1, \ldots, n;$$

where $\mathbf{h} = (h_1, \ldots, h_n) \in \mathbb{R}^{+n}$ is the vector of spline times and a_{MAX_k}, j_{MAX_k} denote the maximum absolute values of acceleration and jerk at joint k. Problem (1), which is apparently a semi-infinite optimization problem, can be easily transformed into a finite one by noting that joint acceleration segments are affine functions of time and joint jerk segments are independent of time (constants). Define the constant joint jerk between knot $i+1$ and knot i as $j_{ki}(\mathbf{h})$. The following finite optimization problem is equivalent to (1):

$$\min_{h \in \mathbb{R}^{+n}} \sum_{i=1}^{n} h_i \tag{2}$$

subject to $(k = 1, \ldots, m)$

$$-a_{MAX_k} \le a_{ki}(\mathbf{h}) \le a_{MAX_k} \quad i = 1, \ldots, n-1; \tag{3}$$
$$-j_{MAX_k} \le j_{ki}(\mathbf{h}) \le j_{MAX_k} \quad i = 1, \ldots, n. \tag{4}$$

Note that $a_{ki}(\mathbf{h})$ and $j_{ki}(\mathbf{h})$ are to be considered explicit functions of spline times: they are well defined rational functions over $\mathbb{R}^{+n}$. Solving the minimum-time trajectory problem is to find a global minimizer $\mathbf{h}^* = (h_1^*, \ldots, h_n^*)$ corresponding to the global minimum $T^* = \sum_{i=1}^{n} h_i^*$ of (2)-(4).

Section 2 succinctly describes the main features of the outer cutting-plane (OCP) algorithm. A first application example is illustrated in section 3. Conclusions are included in the last section.

2 The Outer Cutting-Plane Algorithm

Let us introduce the feasible set of the optimization problem (2)-(4) denoted by $\mathcal{F} := \{\mathbf{h} \in \mathbb{R}^{+n} : |a_{ki}(\mathbf{h})| \le a_{MAX_k},\ i = 1, \ldots, n-1,\ |j_{ki}(\mathbf{h})| \le j_{MAX_k},\ i = 1, \ldots, n;\ k = 1, \ldots, m\}$. Given a positive parameter T the intersection of $\mathbb{R}^{+n}$ with the hyperplane $h_1 + \cdots + h_n = T$ be denoted by $\mathcal{T} := \{\mathbf{h} \in \mathbb{R}^{+n} : h_1 + \cdots + h_n = T\}$. The following property can be deduced [6]: If the convex polyhedron $\mathcal{T}$ is not feasible, i.e. $\mathcal{T} \cap \mathcal{F} = \emptyset$, then the negative half space $h_1 + \ldots + h_n < T$ is not feasible, i.e. $\{\mathbf{h} \in \mathbb{R}^{+n} : h_1 + \cdots + h_n < T\} \cap \mathcal{F} = \emptyset$. The role of the convex polyhedron $\mathcal{T}$ is still emphasized by noting that $\mathcal{T}$ is a level curve of the objective function $\sum_{i=1}^{n} h_i$. This has suggested to devise the following cutting-plane algorithm to solve (2)-(4).

Input of the OCP algorithm: (i) the data set given by $q_{k0}, q_{k2}, q_{k3}, \ldots, q_{k,n-2}, q_{kn}$ and a_{MAX_k}, j_{MAX_k} $k = 1, \ldots, m$; (ii) the precision parameter $\varepsilon > 0$; (iii) the threshold parameter $\eta > 0$.

Output of the OCP algorithm: (i) T^-, T^+ respectively lower and upper bound of T^* with $T^+ - T^- \le \varepsilon$; (ii) $\mathbf{h}_f = (h_{f_1}, \ldots, h_{f_n}) \in \mathcal{F}$ which is the approximate global minimizer satisfying $\sum_{i=1}^{n} h_{f_i} = T^*$.

The OCP algorithm

1. Determine a feasible point $\mathbf{h}_f \in \mathcal{F}$.
2. Apply a local gradient-based procedure to find a local minimizer $\mathbf{h}_f \in \mathcal{F}$ with $T^+ := \sum_{i=1}^{n} h_{f_i}$.
3. $T := T^+ - \varepsilon$
4. Apply procedure *Unfeasibility*$(\eta, T, \mathbf{h}_f, u_0)$.
5. If u_0 = "UnF" then $T^- := T$ and terminate.
6. If u_0 = "F" go to 2.
7. If u_0 = "Cr" apply procedure *Criticalness* and go to 5.
8. End.

At step 1 the above algorithm requires to find a feasible point of $\mathcal{F}$. The simplest method to accomplish it 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 λ to obtain $\mathbf{h}_f = \lambda \mathbf{h}$. To speed up the OCP algorithm is crucial to use a good local optimizer (at step 2) such as, for example, the *generalized reduced gradient method* [7, page 348]. Indeed if step 2 is omitted then the global convergence of the OCP algorithm still holds but the price would be a very slow convergence rate (depended on ε). The core of the algorithm is the interval procedure *Unfeasibility* whose aim is to prove that the convex polyhedron $\mathcal{T}$ is completely unfeasible: output u_0 = "UnF". If it is not possible to prove that $\mathcal{T} \cap \mathcal{F} = \emptyset$ then two cases emerge: (a) a feasible point $\mathbf{h}_f$ has been found: output u_0 = "F"; (b) no conclusion can be obtained (critical case): then the procedure has to halt with u_0 = "Cr". The critical case of procedure *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 $\mathcal{T}$. Roughly speaking, the sensitivity of this procedure to the critical cases is inversely proportional to η which is the "threshold width" of multidimensional intervals processed by *Unfeasibility*. Indeed this procedure solves the feasibility/unfeasibility problem over $\mathcal{T}$ with the above given specifications by using interval analysis techniques [5]. Specifically an exhaustive global search over $\mathcal{T}$ performed via a *depth first* strategy with the use of interval

joint number	knot (degrees)				
	1	2	3	4	5
1	40	extra	120	extra	-30
2	75	knot	-10	knot	-120

Table 1: Knot angles of the example.

spline	1	2	3	4
time (sec.)	0.67	2.81	3.87	0.71

Table 2: The result of the program.

inclusion functions of $a_{ki}(\mathbf{h})$ and $j_{ki}(\mathbf{h})$ is adopted by *Unfeasibility*. The aim of procedure *Criticalness* lies in resolving with certainty the critical case which may emerge at step 7 (u_0 = "Cr"). It is based on a repetitive application of *Unfeasibility*$(\eta, \xi, \mathbf{h}_f, u_0)$ with cyclic values $\xi \in (T, T + \varepsilon/2)$ and with the threshold parameter η halved each time of application ($\eta \leftarrow \eta/2$) until the output is necessarily u_0 = "F" or u_0 = "UnF". A complete convergence analysis of the overall cutting-plane algorithm is reported in [6].

3 An example

The OCP algorithm has been implemented in C++ exploiting the PROFIL libraries made by Olaf Knüppel [8] and GRG2 as local optimizer [9]. As an example we considered the case of a trajectory composed by four splines of a two degrees-of-freedom SCARA-like robot arm in which the maximum limits of acceleration and jerk are the same for each joint and fixed as $a_{MAX_k} = 50$ degrees/s^2 and $j_{MAX_k} = 60$ degrees/s^3, $k = 1, 2$. The starting, intermediate and final positions are indicated in Table 1. We set $\varepsilon = 0.01$ and $\eta = 0.001$. The result is reported in Table 2. Plots of velocities, accelerations, and jerks are given with Figure 1,2.

4 Conclusions

A global outer cutting-plane algorithm has been proposed for the optimal minimum-time trajectory generation of an m-joint robot. Even though the adopted joint space scheme is a very crude simplification for the general optimal robot planning problem, the presented OCP algorithm can be useful in the practical programming of industrial robots or mechanical manipulators.

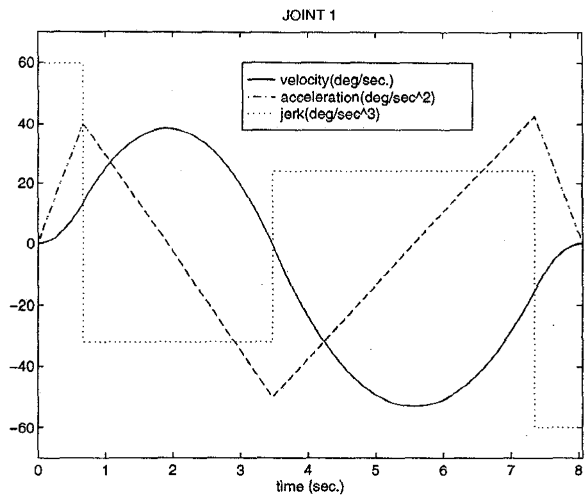

Figure 1: The optimum trajectory of joint 1.

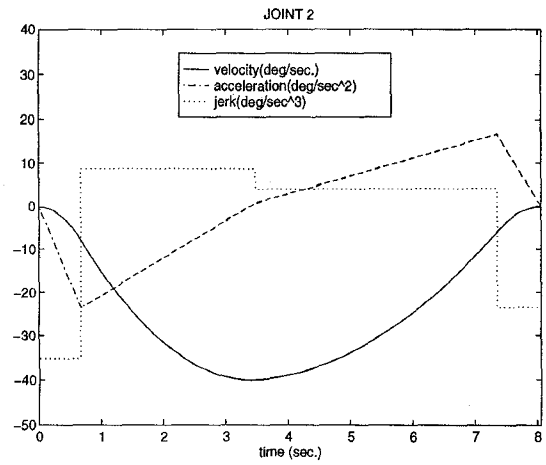

Figure 2: The optimum trajectory of joint 2.

References

[1] C.-S. Lin, P.-R. Chang, and J.Y.S. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Trans. Automatic Control*, vol. AC-28, no. 12, pp. 1066–1074, 1983.

[2] D. Simon, "The application of neural networks to optimal robot trajectory planning," *Robotics and Autonomous Systems*, vol. 11, pp. 23–34, 1993.

[3] A. Piazzi and A. Visioli, "A global optimization approach to trajectory planning for industrial robots," IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, September 1997, Accepted for presentation.

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

[5] H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization*, Ellis Horwood Limited, Chichester, UK, 1988.

[6] A. Piazzi and A. Visioli, "Global minimum-time trajectory planning of industrial robots," Tech. Rep. TSC01/97, Dipartimento di Ingegneria dell'Informazione, Università di Parma (Italy), August 1997.

[7] D.G. Luenberger, *Linear and Nonlinear Programming*, Addison–Wesley, Reading, Massachusetts, second edition, 1989.

[8] O. Knüppel, "PROFIL — programmer's runtime optimized fast interval library," Tech. Rep. 93.4, Technische Universität Hamburg-Harburg (Germany), July 1993.

[9] "GRG2 user's guide," Winward Technologies, Inc. and Optimal Methods, Inc., June 1995.