Available online at www.sciencedirect.com
ScienceDirect
Robotics and Computer-Integrated Manufacturing 24 (2008) 415–426
Robotics and Computer-Integrated Manufacturing
www.elsevier.com/locate/rcim

A technique for time-jerk optimal planning of robot trajectories

Alessandro Gasparetto*, Vanni Zanotto

Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Università di Udine, Via delle Scienze 208, 33100 Udine, Italy

Received 2 November 2005; received in revised form 16 October 2006; accepted 18 April 2007

Abstract

A technique for optimal trajectory planning of robot manipulators is presented in this paper. In order to get the optimal trajectory, an objective function composed of two terms is minimized: a first term proportional to the total execution time and another one proportional to the integral of the squared jerk (defined as the derivative of the acceleration) along the trajectory. This latter term ensures that the resulting trajectory is smooth enough. The proposed technique enables one to take into account kinematic constraints on the robot motion, expressed as upper bounds on the absolute values of velocity, acceleration and jerk. Moreover, it does not require the total execution time of the trajectory to be set a priori. The algorithm has been tested in simulation yielding good results, also in comparison with those provided by another important trajectory planning technique.

Keywords: Robotics; Trajectory planning; Jerk; Execution time; Optimization; Splines

1. Introduction

The trajectory planning problem is a fundamental one in Robotics. It may be formulated thus: define a temporal motion law along a given geometric path, such as certain requirements set on the trajectory properties are fulfilled. Hence, the aim of trajectory planning is to generate the reference inputs for the control system of the manipulator, in order to be able to execute the motion. The inputs of any trajectory planning algorithm are: the geometric path, the kinematic and dynamic constraints; and the output is the trajectory of the joints (or of the end effector), expressed as a time sequence of position, velocity and acceleration values.

Usually, the geometric path is specified in the operating space, i.e. with reference to the end effector of the robot, because both the task to perform and the obstacles to avoid can be more naturally described in this space. However, the trajectory is normally planned in the joint space of the robot, after a kinematic inversion of the given geometric path has been done. The joint trajectories are then obtained by means of interpolating functions which meet the imposed kinematic and dynamic constraints.

The main advantage of planning a trajectory in the joint space rather than in the operating space is that the control system acts on the manipulator joints rather than on the end effector, so it would be easier to adjust the trajectory according to the design requirements if working in the joint space. Moreover, trajectory planning in the joint space would allow to avoid the problems arising with kinematic singularities and manipulator redundancy.

The main disadvantage of planning the trajectory in the joint space is that, given the planned trajectory in the joint space, the motion actually performed by the robot end effector is not easily foreseeable, due to the non-linearities introduced when transforming the trajectories of the joints into the trajectories of the end effector through direct kinematics.

Apart from the particular strategy adopted, the motion laws generated by the trajectory planner must fulfill the constraints set a priori on the maximum values of the generalized joint torques, and must be such that no mechanical resonance mode is excited. This can be achieved by forcing the trajectory planner to generate *smooth* trajectories, i.e. trajectories with good continuity features: in particular, it would be desirable to obtain trajectories with continuous joint accelerations, so that the absolute value of the *jerk* (i.e. of the derivative of the acceleration) keeps bounded. Limiting the jerk is very

*Corresponding author. Tel.: +39 0432 558257; fax: +39 0432 558251.
E-mail address: gasparetto@uniud.it (A. Gasparetto).

0736-5845/$ - see front matter
doi:10.1016/j.rcim.2007.04.001

important, because high jerk values can wear out the robot structure, and heavily excite its resonance frequencies. Vibrations induced by non-smooth trajectories can damage the robot actuators, and introduce large errors while the robot is performing tasks such as trajectory tracking. Moreover, low-jerk trajectories can be executed more rapidly and accurately.

Most techniques found in the scientific literature on the trajectory planning problem is based on the optimization of some parameter or some objective function. The most significant optimality criteria are:

(A) minimum execution time;
(B) minimum energy (or actuator effort); and
(C) minimum jerk.

Besides the aforementioned approaches, some hybrid optimality criteria have also been proposed (e.g. time–energy optimal trajectory planning).

1.1. Minimum-time trajectory planning

Minimum-time algorithms were the first trajectory planning techniques proposed in the scientific literature because they were tightly linked to the need of increasing the productivity in the industrial sector. The first interesting methods of this kind [1,2] are developed in the so-called *position-velocity phase plane*. The main idea is to use the curvilinear abscissa θ of the path as a parameter, in order to express the dynamic equation of the manipulator in a parametric form. An alternative approach is proposed in [3,4], where dynamic programming techniques are employed. However, the aforementioned techniques generate trajectories with discontinuous values of accelerations and joint torques, because the dynamic models used for trajectory computation assume the robot members as perfectly rigid and neglect the actuator dynamics. This leads to two undesired effects: first, the real actuators of the robot cannot generate discontinuous torques, which causes the joint motion to be always delayed with respect to the reference trajectory. The accuracy in trajectory following is then greatly reduced and the so-called chatter phenomenon may eventually occur, consisting in high frequency vibrations that can damage the manipulator structure. Second, the time-optimal control requires saturation of at least one robot actuator at any time instant, so the controller cannot correct the tracking errors arising from disturbances or modeling errors.

In order to overcome such problems, other approaches [5,6] impose some limits on the *actuator jerks*, defined as the variation rates of the joint torques. In this way, the generated trajectory will not be exactly time-optimal, albeit close to the optimality value; however, the generated trajectories can be effectively implemented and more advanced control strategies can be applied.

In order to generate trajectories with continuous accelerations, a common strategy is to use smooth trajectories, such as the *spline* functions, that have been extensively employed in the scientific literature on both kinematic and dynamic trajectory planning.

The first formalization of the problem of finding the optimal curve interpolating a sequence of nodes in the joint space, obtained through kinematic inversion of a discrete set of points representing position and orientation of the reference frame linked to the end effector of the robot, can be found in [7]. In [8] the same algorithm is presented, but the trajectories are expressed by means of cubic B-splines. In [9] the optimization algorithm proposed in [7] is modified, so that it can deal also with dynamic constraints and with objective functions of a more general type; however, the reported simulations still refer to the trajectory planning problem with kinematic constraints.

In [10] a method is proposed, to compute point-to-point minimum-time trajectories using uniform B-splines. The dynamic model of the manipulator is considered, and the problem of semi-infinite constraint resulting from the algorithm is bypassed by sampling a certain number of points.

All the algorithms based on the optimization of cubic splines mentioned in the foregoing yield a local optimal solution, which may or may not be the global optimum. Some algorithms have been developed, aimed at finding a global optimal solution for the trajectory planning problem for robotic manipulators. Examples are given by [11], where the proposed algorithm is based on the so-called *interval analysis*, and [12] where a hybrid technique using genetic algorithms is put forward.

1.2. Minimum-energy trajectory planning

Planning the robot trajectory using energetic criteria provides several advantages. On one hand, it yields smooth trajectories resulting easier to track and reducing the stresses to the actuators and to the manipulator structure. Moreover, saving energy may be desirable in several applications, such as those with a limited capacity of the energy source (e.g. robots for spatial or submarine exploration).

Examples of energy optimal trajectory planning are provided in [13–15]. In [13,14] point-to-point trajectories with minimal energy are considered, with upper bounds on the amplitude of the control signals and the joint velocities. In [15] a trajectory with motion constraints set on the end effector is optimized. The cost function is given by the time integral of the squared joint torques, and the trajectories are expressed by means of cubic B-spines. The resulting motion minimizes the actuators effort.

Paper [16] deals with time–energy optimal trajectories, i.e. the cost function has a term linked to the execution time and a term expressing the energy spent. The trade-off between the two needs can be adjusted by changing the respective weights.

Other examples of time–energy optimal trajectories are given in [17,18]. In [17] a cubic spline trajectory is considered, subject to kinematic constraints on the maximum values of velocity, acceleration and jerk. In [18] a point-to-point trajectory parametrized by means of cubic B-splines is considered.

1.3. Minimum-jerk trajectory planning

The importance of generating trajectories that do not require abrupt torque variations has already been remarked. In [5,6] this has been obtained by setting upper bounds on the rate of torque variations, but in this way the third-order dynamics of the manipulator must be computed. An indirect method to get the same results is to set an upper limit on the *jerk*, defined as the time derivative of the acceleration of the manipulator joints.

The positive effects induced by a jerk minimization are:

- errors during trajectory tracking are reduced;
- stresses to the actuators and to the structure of the manipulator are reduced;
- excitation of resonance frequencies of the robot is limited; and
- a very coordinated and natural robot motion is yielded.

The last effect appears very interesting; indeed, some studies [19,20] show that the movement of a human arm seems to satisfy an optimization criterion linked to the rate of variation of the joint acceleration. So, it may be inferred that minimum-jerk trajectories are an example of optimization according to physical criteria imitating the capacity of natural coordination of a human being.

In [21] the solutions for two point-to-point minimum-jerk trajectories are analytically obtained; the optimization, obtained through the Pontryagin principle, concerns two objective functions, namely the time integral of the squared jerk and the maximum absolute value of the jerk (*minimax* approach).

An interesting approach is described in [22], where the interpolation is done by means of trigonometric splines, thus ensuring the continuity of the jerk. The interpolation with trigonometric splines features a sound *locality property,* i.e. if the value of some node in the input sequence is changed, only the two splines associated with that node should be computed again, not the whole trajectory: this property can be usefully exploited for performing tasks in real time (e.g. to bypass an obstacle encountered while executing the planned motion).

In [23,24] the so-called *interval analysis* is used to develop an algorithm that globally minimizes the maximum absolute value of the jerk along a trajectory whose execution time is set a priori: hence, an approach of the type *minimax* is used. The trajectories are expressed by means of cubic splines and the intervals between the via-points are computed so that the lowest possible jerk peak is produced. A comparison between the simulation results of the algorithm proposed in [23] and those of the method described in [22] is drawn.

An important remark must be done at the end of this literature overview: in the case of trajectory planning along a given path, all jerk-minimization algorithms that could be found consider an execution time set a priori and do not accept any kinematic constraint as input.

This paper proposes a technique for optimal trajectory planning of robot manipulators. The approach is based on the simultaneous minimization of jerk and execution time, thus allowing to get smooth trajectories but avoiding an excessive duration of trajectory execution.

With respect to most minimum-jerk optimization techniques that could be found in the scientific literature, the method described in this work enables one to define kinematic constraint on the robot motion before running the algorithm. Such constraints are expressed as upper bounds on the absolute values of velocity, acceleration and jerk for all robot joints, so that any physical limitation of the real manipulator can be taken into account when planning its trajectory.

The paper is organized as follows: in Section 2 the optimization problem is formulated, namely the objective function to minimize is defined. Cubic splines are then chosen to define the trajectory which is to solve the optimization problem: the objective function and the kinematic constraints are rewritten accordingly. Section 4 describes the whole algorithm, which is based on an iterative minimization procedure. Section 5 shows the results of some simulation that have been carried out using the same input data as another important algorithm found in the literature, in order to be able to compare the results.

The utility of the proposed method lies in the fact that the user can set a maximum value of the jerk before running the algorithm. In this way, since the vibrations of the structure are directly related to the jerk, setting a prudential value for the jerk ensures that the vibratory phenomena of the resulting motion will be kept under a certain value. In this way, the trajectory generated by our algorithm can be directly input to the robot.

Optimization alternatives, which rely on the intrinsic limitations of the robot's actuators, may not prove effective in reducing vibratory phenomena. For instance, discontinuous values of accelerations, albeit kept below the physical limitations of the servo motors, would cause very high jerk values.

2. Formulation of the optimization problem

As stated in the foregoing, it is desirable that trajectories generated by planning algorithms feature sufficient smoothness properties, so as to avoid to excite mechanical resonance modes of the manipulator structure. This can be achieved by if the acceleration of the planned trajectory is a continuous function, so that the value of the jerk (the derivative of the acceleration) keeps bounded. Limitation of the value of the jerk is an indirect way to bound the variation rate of the joint torques, with no need to keeping into account the dynamic model of the manipulator in the planning algorithm.

Limiting the value of the jerk yields several positive effects such as reduction of the stresses to the actuators and to the robot structure, and a better accuracy in trajectory tracking.

Furthermore, if the value of the jerk is kept low, it becomes possible to design simple robot controllers, which are easier to implement and more robust with respect to external disturbances.

The trajectory planning technique described in this paper assumes that the geometric path, generated a priori by an upper-level path planner, is given in the form of a sequence of via-points in the operating space of the robot, which represent successive positions and orientations of the end effector of the manipulator; obstacle avoidance problems are assumed as already solved by the upper-level path planer and will not be considered here. The proposed technique generates an optimal trajectory for the robot, by associating temporal information to the pre-planned geometric path, according to an optimality criterion based on the minimization of some objective function that will be defined below, and by taking into account any kinematic constraints, expressed by means of upper bounds on velocity, acceleration and jerk.

There are several algorithms where the value of the jerk appears in the objective function. For instance, in [22] the integral of the squared jerk is minimized, while Piazzi and Visioli [23] use a *minimax* approach to minimize the maximum value of the jerk along the trajectory.

However, these techniques consider the execution time as known (and set a priori); moreover, it is not possible to set any kind of kinematic constraint on the trajectory, because they are not taken into account.

On the contrary, the algorithm presented in this paper generates an optimal trajectory such as:

- it is not required to set the execution time a priori and
- kinematic constraints on the resulting trajectory (i.e. upper bounds on the values of velocity, acceleration and jerk of the robot joints) can be defined before running the algorithm.

The objective function adopted in proposed technique is given by the sum of two terms having opposite effects:

- the first term is proportional to the execution time and
- the second term is proportional to the integral of the squared jerk.

Of course, reducing the value of the first term of the objective function will lead to trajectories featuring large values of the kinematic quantities (velocity, acceleration and jerk), while reducing the second term will lead to a smoother trajectory.

The trade-off between these two tendencies can be performed by suitably adjusting the weights of the two terms of the objective function: a larger weight of the jerk term will lead to smoother but slower trajectories, while a larger weight of the time term will lead to faster but less smooth trajectories.

Table 1
Meaning of symbols

N	Number of robot joints
k_{T}	Weight of the term proportional to the execution time
vp	Number of via-points
k_{J}	Weight of the term proportional to the jerk
h_i	Time interval between two via-points
t_f	Total execution time of the trajectory
$\dot{q}_j(t)$	Velocity of the jth joint
VC_j	Velocity bound for the jth joint (symmetrical)
$\ddot{q}_j(t)$	Acceleration of the jth joint
WC_j	Acceleration bound for the jth joint (symmetrical)
$\dddot{q}_j(t)$	Jerk of the jth joint
JC_j	Jerk bound for the jth joint (symmetrical)

Hence, the optimal trajectory planning problem can be formulated as

$$\begin{cases} \text{find :} \\ \min \quad k_{\mathrm{T}} N \sum_{i=1}^{vp-1} h_i + k_{\mathrm{J}} \sum_{j=1}^{N} \int_0^{t_f} \left(\dddot{q}_j(t) \right)^2 \mathrm{d}t \\ \text{s.t.} \\ |\dot{q}_j(t)| \leqslant VC_j, \quad j = 1, \ldots, N, \\ |\ddot{q}_j(t)| \leqslant WC_j, \quad j = 1, \ldots, N, \\ |\dddot{q}_j(t)| \leqslant JC_j, \quad j = 1, \ldots, N. \end{cases} \tag{1}$$

The meaning of the symbols appearing in (1) is explained in Table 1.

By solving the optimization problem (1), the vector of the time intervals h_i between any pair of consecutive via-points is computed.

Besides the kinematic constraints appearing in (1), the trajectory solving the above defined optimization problem must also meet the interpolation conditions for all the via-points, as well as the initial and final conditions for velocity, acceleration and jerk.

3. Solving the optimization problem by means of cubic splines

The use of cubic splines in trajectory planning is very common, because the generated trajectories have continuous values of the accelerations. Moreover, unlike higher-order polynomials, cubic splines do not present problems such as excessive oscillations and overshoot between any pair of reference points.

The optimal planning technique proposed in this paper, based on the objective function (1), will be applied to the case of a trajectory composed of cubic splines.

Some facts concerning the spline theory will now be briefly reviewed. Given a sequence of vp points in the operating space, a sequence of vp via-points in the joint space is obtained through kinematic inversion. Every pair of consecutive via-points must now be connected through a cubic spline, so that the complete trajectory minimizes the objective function defined a priori (i.e. (1) for the purposes of this paper).

Two virtual points should be introduced, so that the initial and final conditions for velocity and acceleration can be respected; as is usually done, the virtual points are put at the second and at the second-last position of the sequence.

Now, let $t_1,\ldots,t_n$ $(n = vp+2)$ be the sequence of time instants corresponding to the via-points; let $h_i = t_{i+1}-t_i$ be the time interval between two consecutive via-points and $Q_{j,i}(t)$ be the cubic polynomial for the jth joint defined on the interval $[t_i,\ t_{i+1}]$. Solving the interpolation problem means to find $n-1$ cubic polynomials satisfying the imposed conditions. In order to avoid considering a cumbersome system with $4(n-1)$ unknowns, it can be noticed that the second derivative of $Q_{j,i}(t)$ is a linear function of t in the interval $[t_i\ ,t_{i+1}]$, namely:

$$\ddot{Q}_{j,i}(t) = \frac{t_{i+1}-t}{h_i}\ddot{Q}_{j,i}(t_i) + \frac{t-t_i}{h_i}\ddot{Q}_{j,i}(t_{i+1}), \quad i = 1,\ldots,n-1. \tag{2}$$

By integrating (2) twice and by setting $Q_{j,i}(t_i) = q_{j,i}$ and $Q_{j,i}(t_{i+1}) = q_{j,i+1}$, the following interpolation functions are obtained:

$$\begin{aligned} Q_{j,i}(t_i) = {} & \frac{\ddot{Q}_{j,i}(t_i)}{6h_i}(t_{i+1}-t_i)^3 + \frac{\ddot{Q}_{j,i}(t_{i+1})}{6h_i}(t-t_i)^3 \\ & + \left[\frac{q_{i+1}}{h_i} - \frac{h_i\ddot{Q}_{j,i}(t_{i+1})}{6}\right](t-t_i) \\ & + \left[\frac{q_i}{h_i} - \frac{h_i\ddot{Q}_{j,i}(t_i)}{6}\right](t_{i+1}-t), \quad i = 1,\ldots,n-1. \end{aligned} \tag{3}$$

After some algebra, a linear system with just $n-2$ unknowns (the accelerations at the via-points) is obtained as

$$\mathbf{K}\mathbf{A}_j = \mathbf{B}_j \quad \forall j = 1,\ldots,N. \tag{4}$$

In (4), the coefficient matrix $\mathbf{K}$ (the same for all joints) is non-singular and band-diagonal, while $\mathbf{B}_j$ is a known vector, function of h_i and $q_{j,i}$, different for each joint.

The accelerations at the inner via-points, namely:

$$\mathbf{A}_j = \begin{bmatrix} \alpha_{j,2} \\ \alpha_{j,n-1} \end{bmatrix} \tag{5}$$

with $\ddot{Q}_{j,i} = \alpha_{j,i}$, can then be easily obtained by solving the linear system (4).

In order to apply the present technique, an explicit expression for the objective function and the kinematic constraints (1) must be formulated, considering the fact that the trajectory is made of cubic splines.

As for the velocity constraints, let us assume they are constant and symmetric, so that

$$|\dot{Q}_{j,i}(t)| \leqslant VC_j \quad \forall j = 1,\ldots,N\ \forall i = 1,\ldots,n-1. \tag{6}$$

These constraints are of the semi-infinite type, i.e. they must hold for any value of the continuous variable t. It would be much easier to deal with them if they were put into a finite form; this can be done by noticing that the velocity is parabolic in the interval $[t_i,\ t_{i+1}]$, so that it reaches its maximum value either at one of the interval ends, or at some intermediate instant $t^*_{j,i}$ when the acceleration becomes zero, namely:

$$t^*_{j,i} = t_i + \frac{h_i\alpha_{j,i}}{\alpha_{j,i}-\alpha_{j,i+1}} \tag{7}$$

and the value of the velocity at that instant $t^*_{j,i}$ is given by

$$\dot{Q}^*_{j,i} = -\frac{h_i}{2(\alpha_{j,i+1}-\alpha_{j,i})}\alpha_{j,i}\alpha_{j,i+1} + \frac{q_{j,i+1}-q_{j,i}}{h_i} - \frac{h_i}{6}(\alpha_{j,i+1}-\alpha_{j,i}). \tag{8}$$

The values of velocity at the interval ends are

$$\dot{Q}_{j,i}(t_i) = -\frac{\alpha_{j,i}}{2}h_i + \frac{q_{j,i+1}-q_{j,i}}{h_i} + \frac{\alpha_{j,i}-\alpha_{j,i+1}}{6}h_i,$$

$$\dot{Q}_{j,i}(t_{i+1}) = \frac{\alpha_{j,i+1}}{2}h_i + \frac{q_{j,i+1}-q_{j,i}}{h_i} + \frac{\alpha_{j,i}-\alpha_{j,i+1}}{6}h_i c. \tag{9}$$

Hence, the semi-infinite constraint (6) can be expressed much more conveniently as

$$\begin{aligned} & \max\{|\dot{Q}_{j,i}(t_i)|, |\dot{Q}_{j,i}(t_{i+1})|, |\dot{Q}^*_{j,i}|\} \leqslant VC_j, \\ & j = 1,\ldots,N,\ \ i = 1,\ldots,n-1 \end{aligned} \tag{10}$$

which represent a finite number of relations that must hold for the planned trajectory.

The kinematic constraints on accelerations and jerk can be similarly reduced to a finite form. The semi-infinite constraints on the acceleration:

$$|\ddot{Q}_{j,i}(t)| \leqslant WC_j \quad \forall j = 1,\ldots,N\ \forall i = 1,\ldots,n-1 \tag{11}$$

can be expressed in a finite form, by noticing that the acceleration is linear in the interval $[t_i,\ t_{i+1}]$, so that it must reach its maximum value at one of the interval ends, thus yielding:

$$\max\{|\alpha_{j,1}|,\ldots,|\alpha_{j,n}|\} \leqslant WC_j \quad \forall j = 1,\ldots,N. \tag{12}$$

As for the jerk, the semi-infinite constraint is given by

$$|\dddot{Q}_{j,i}(t)| \leqslant JC_j \quad \forall j = 1,\ldots,N\ \forall i = 1,\ldots,n-1. \tag{13}$$

Since the jerk $\dddot{Q}_{j,i}(t) = (\alpha_{j,i+1}-\alpha_{j,i}/h_i)$ is constant on any interval $[t_i,\ t_{i+1}]$, (13) becomes

$$\left|\frac{\alpha_{j,i+1}-\alpha_{j,i}}{h_i}\right| \leqslant JC_j \quad \forall j = 1,\ldots,N\ \forall i = 1,\ldots,n-1. \tag{14}$$

Moreover, the trajectory intervals h_i $(h_i > 0\ \forall i = 1,\ldots,n-1)$ which are the true optimization parameters, must be considered. They are subject to a lower bound, since in any interval the condition $|q_{j,i+1}-q_{j,i}/h_i| \leqslant VC_j$ must hold. Hence, the intervals h_i must be such as

$$h_i > w_i = \max_{j=1,\ldots,N}\left\{\frac{|q_{j,i+1}-q_{j,i}|}{VC_j}\right\} > 0. \tag{15}$$

In other words, for each trajectory interval, the N time values w_i obtained by running the trajectory at the highest allowed speed should be computed; the minimum value of the interval is then given by the largest computed value.

Finally, it only remains to determine the expression of the objective function for a trajectory made of cubic splines. Since the jerk is piecewise constant for a cubic trajectory, the expression of the integral of its square jerk is very simple. So, the objective function (1) can be formulated as

$$\mathrm{FOBJ} = k_T \sum_{i=1}^{n-1} h_i + k_J \sum_{j=1}^{N} \sum_{i=1}^{n-1} \left(\frac{(\alpha_{j,i+1} - \alpha_{j,i})^2}{h_i} \right). \tag{16}$$

The values of the weights k_T and k_J can be chosen in order to adjust the trajectory and to find a suitable trade-off between the need for a quick execution and the need for a smooth trajectory. By choosing $k_T = 0$, a minimum-jerk trajectory is found, while setting $k_J = 0$ enables one to obtain a minimum-time trajectory.

4. Execution of the algorithm

The trajectory planning algorithm described in the previous section can then be run in whatever simulation environment, by solving the minimization problem formulated in (16) through some dedicated software routine (usually, based on iteration).

The steps for running the algorithm are summarized below:

(1) Starting from a given path in the operating space, a kinematic inversion is applied, so as to get a sequence of via-points in the joint space.
(2) The numeric values of kinematic constraints are set, on the basis of the structural constraints of the manipulator, and of design considerations.
(3) A suitable initial solution to start the iterative optimization algorithm is chosen.
(4) The expression of the objective function (16) and of the kinematic constraints (10), (12), (14) and (15) are input to the optimization algorithm.
(5) The solution of the optimization problem is then obtained using sequential quadratic programming techniques (for instance, the *fmincon* function of MatLab™).

As for all iterative routines, a crucial point is the choice of a suitable initial solution, because a wrong choice would affect the execution time and even the final result of the algorithm.

A suitable initial solution can be found by considering the lower bound of the intervals h_i and making a *time scaling* of the trajectory. Namely, let t be the (independent) time variable, and $q(t)$, $v(t)$, $a(t)$, $j(t)$ the trajectory and its derivatives. If a scaled time variable τ is defined as $\tau = \Lambda t$, the following holds:

$$\begin{cases} v(\tau) = \dfrac{v(t)}{\Lambda}, \\ a(\tau) = \dfrac{a(t)}{\Lambda^2}, \\ j(\tau) = \dfrac{j(t)}{\Lambda^3}. \end{cases} \tag{17}$$

Now, let $\mathbf{H}_{\mathrm{lb}}$ be the vector of the $n-1$ lower bounds of the optimization variables. By substituting $\mathbf{H}_{\mathrm{lb}}$ into system (4) the N trajectories of the joints corresponding to $\mathbf{H}_{\mathrm{lb}}$ are obtained. A set of coefficient $\{\Lambda_1\ \Lambda_2\ \Lambda_3\}$ can then be computed by considering the maximum values of velocity, acceleration and jerk for every joint and every time interval:

$$\begin{aligned} \Lambda_1 &= \max_{j=1,\ldots,N} \left\{ \max_{t\in[t_i,t_{i+1}],i=1,\ldots,n-1} \left\{ \frac{|\dot{Q}_{j,i}(t)|}{VC_j} \right\} \right\}, \\ \Lambda_2 &= \max_{j=1,\ldots,N} \left\{ \max_{t\in[t_i,t_{i+1}],i=1,\ldots,n-1} \left\{ \frac{|\ddot{Q}_{j,i}(t)|}{WC_j} \right\} \right\} \\ &= \max_{j=1,\ldots,N} \left\{ \max_{t\in[t_i,t_{i+1}],i=1,\ldots,n-1} \left\{ \frac{|\alpha_{j,i}|}{WC_j} \right\} \right\}, \\ \Lambda_3 &= \max_{j=1,\ldots,N} \left\{ \max_{t\in[t_i,t_{i+1}],i=1,\ldots,n-1} \left\{ \frac{|\dddot{Q}_{j,i}(t)|}{JC_j} \right\} \right\} \\ &= \max_{j=1,\ldots,N} \left\{ \max_{t\in[t_i,t_{i+1}],i=1,\ldots,n-1} \left\{ \frac{|\alpha_{j,i+1} - \alpha_{j,i}/h_i|}{JC_j} \right\} \right\}. \end{aligned} \tag{18}$$

The scaling factor Λ can now be chosen as

$$\Lambda = \max\{1\ \ \Lambda_1\ \ \Lambda_2^{1/2}\ \ \Lambda_3^{1/3}\} \tag{19}$$

Table 2
Input data for trajectory planning

Joint	Via-points (deg)					
	1	2	3	4	5	6
1	−10	Virtual	60	20	Virtual	55
2	20		50	120		35
3	15		100	−10		30
4	150		100	40		10
5	30		110	90		70
6	120		60	100		25

Table 3
Kinematic limits of the joints

Joint	Velocity (deg/s)	Acceleration (deg/s^2)	Jerk (deg/s^3)
1	100	60	60
2	95	60	66
3	100	75	85
4	150	70	70
5	130	90	75
6	110	80	70

which leads to the following initial solution:

$$\mathbf{H}_0 = \Lambda\mathbf{H}_{\text{lb}}. \tag{20}$$

5. Simulation results and comparison

The algorithm described in this paper has been tested in simulation for a 6-joint robot. The input data has been taken to be the same as in [23], in order to be able to make a comparison with the results obtained from the algorithm proposed by Piazzi and Visioli.

The input data are reported in the tables below. Table 2 contains the values of the via-point of the trajectory, while Table 3 contains the values of the kinematic limits of the joints.

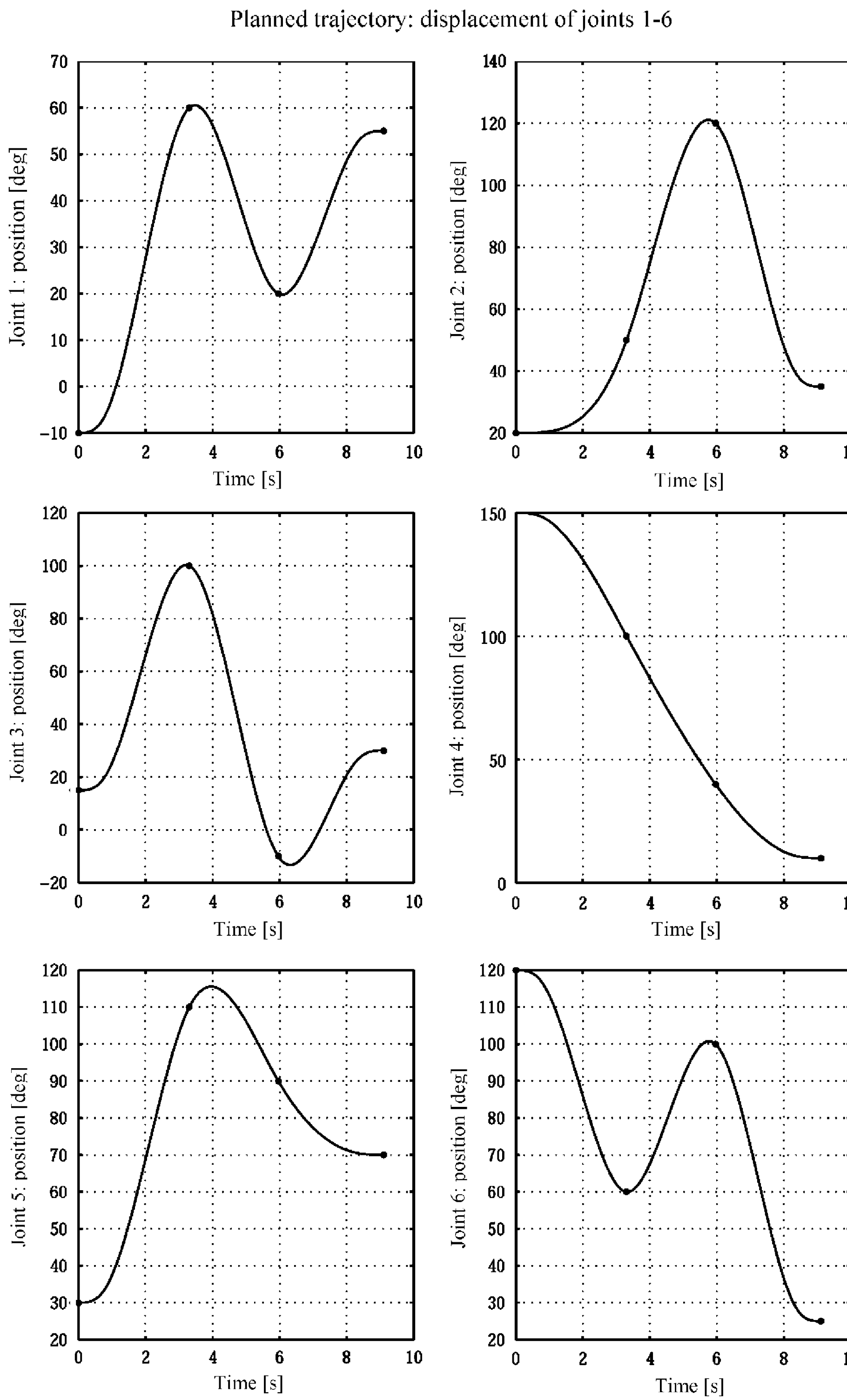

Fig. 1. Planned trajectory for a 6-joint robot.

As already stated in the foregoing, the technique proposed by Piazzi and Visioli [23] takes the total execution time as given, while the algorithm described in this paper outputs the total execution time as a result; its value depending on the weights k_T and k_J appearing in the expression of the cost function (1). So, in order to be able to compare the results yielded by the two algorithms, the values of the weights k_T and k_J have been adjusted so that the execution time of the two algorithms would be the same (namely, 9.1 s).

The results of the simulations are reported in Figs. 1–4, showing the trajectories of the six joints and their derivatives (velocity, acceleration and jerk, respectively).

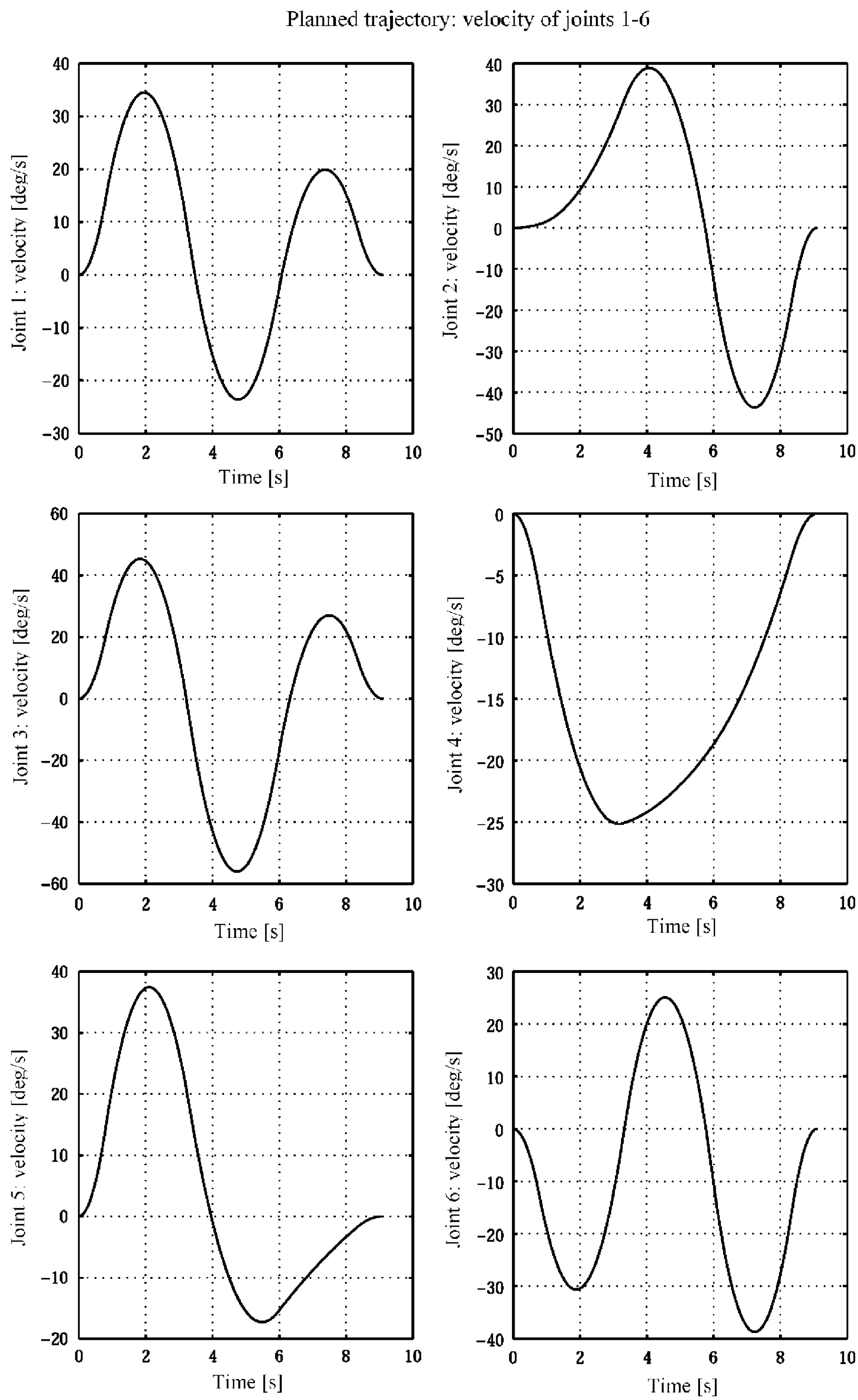

Fig. 2. Velocity of the six joints for the planned trajectory.

Table 4 reports the mean values of velocity, acceleration and jerk for all joints, compared with those found in [23]. The technique presented in this work yields lower mean values for velocity and acceleration, as well as for jerk (except for joint 5 whose value is slightly higher).

The reported results show the effectiveness of the algorithm proposed in performing an optimization of the trajectories. The comparison with a well-known trajectory planning algorithm proves that the kinematic values of the generated trajectories are good and kept under control.

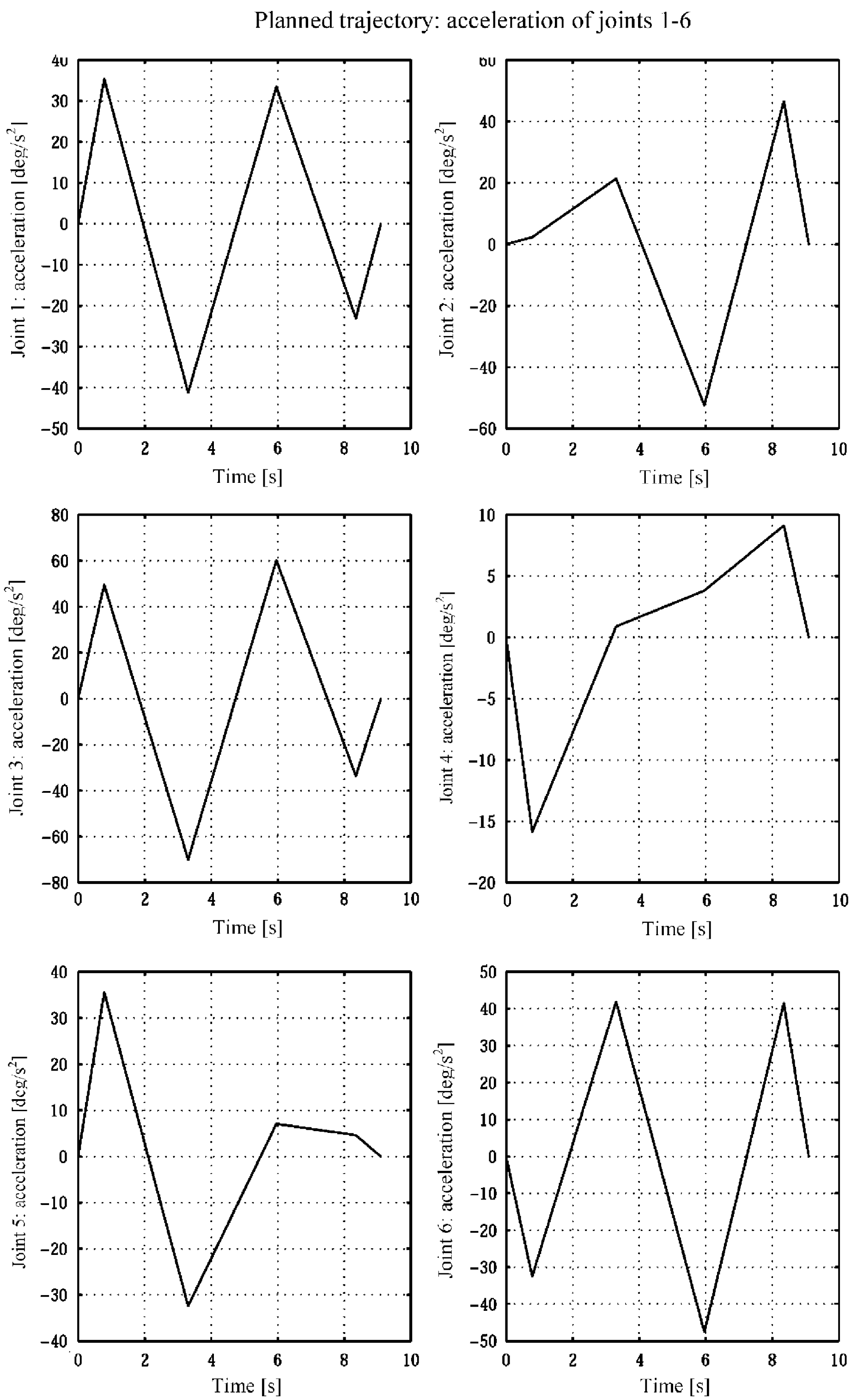

Fig. 3. Acceleration of the six joints for the planned trajectory.

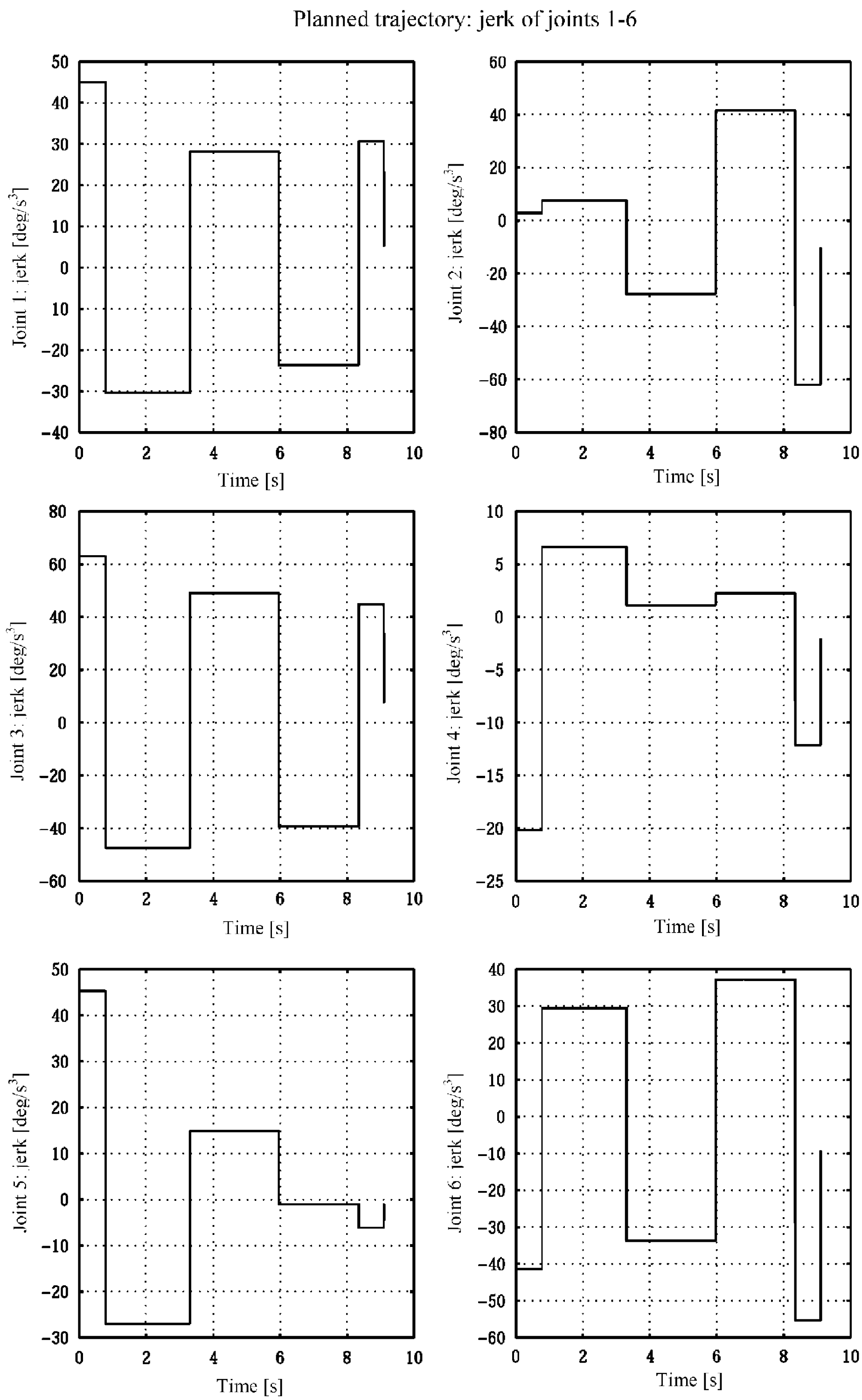

Fig. 4. Jerk of the six joints for the planned trajectory.

6. Conclusion and future work

A methodology for optimal trajectory planning of robotic manipulators has been described in this paper. The technique is based on minimization of an objective function that takes into account both the execution time and the integral of the squared jerk along the whole trajectory. Unlike many other trajectory planning techniques, the proposed method does not take the execution time as given a priori and takes into account kinematic constraints on the robot motion, expressed as upper bounds on the absolute values of velocity, acceleration and jerk for all robot joints.

Table 4
Mean kinematic values resulting from the optimization procedure

Algorithm	Joint					
	1	2	3	4	5	6
Gasparetto–Zanotto						
V_{med}	16.10	20.57	26.61	15.38	14.40	19.40
A_{med}	17.15	18.15	28.23	5.53	12.03	20.76
J_{med}	29.24	26.45	46.85	5.48	16.48	35.90
Piazzi–Visioli [23]						
V_{med}	16.08	20.69	26.42	15.38	14.46	19.50
A_{med}	17.45	18.80	28.52	5.55	12.18	21.39
J_{med}	29.73	27.68	47.35	5.75	16.38	36.97

The algorithm has been applied considering to compose the trajectory by means of cubic splines connecting pairs of consecutive via-points. The expressions for the objective function and the kinematic constraints have been formulated in this case.

Finally, the algorithm has been run in simulation, taking as input data those found in the work by Piazzi and Visioli [23]. Comparison of the results with those provided in [23] has shown that the effectiveness of the algorithm is effective in performing an optimal trajectory planning.

Future work will be devoted to apply the present technique to trajectories of different kinds (like B-splines, trigonometric splines, etc.), so as to evaluate the applicability, the algorithm and its results.

Moreover, application of the algorithm to a real robotic structure is expected in the next future, according to the following procedure:

- the kinematic and dynamic characteristics of the robot will be input to the algorithm,
- the algorithm will be running so as to generate the desired minimum jerk and time trajectory, and
- the generated trajectory will then be input to the robot for execution.

In principle, the planned trajectory will be compatible with the robot controller, because all the limitations on the kinematics and dynamics of the robot are taken into account before running the trajectory planning algorithm, thus the resulting trajectory will be smooth enough.

The possibility to set maximum bounds for the jerk before executing the trajectory allows for a large stability and safety margin. Since the proposed algorithm ensures that the jerk of the resulting trajectory will be kept under the limit values set a priori by the user, the resulting vibrations will also be kept low, provided that a suitable safety coefficient is chosen.

Acknowledgments

The authors would like to thank Mr. Claudio Mantovani for his important contribution to this work. This work was supported in part by the Italian Ministry of Instruction and University, in the framework of the PRIN 2004 project.

References

[1] Bobrow JE, Dubowsky S, Gibson JS. Time-optimal control of robotic manipulators along specified paths. Issue Series Title: Int J Robot Res 1985;4(3):554–61.
[2] Shin KG, McKay ND. Minimum-time control of robotic manipulators with geometric path constraints. IEEE Trans Automat Control 1985;30(6):531–41.
[3] Shin KG, McKay ND. A Dynamic programming approach to trajectory planning of robotic manipulators. IEEE Trans Automat Control 1986;31(6):491–500.
[4] Balkan T. A dynamic programming approach to optimal control of robotic manipulators. Mech Res Commun 1998;25(2):225–30.
[5] Constantinescu D, Croft EA. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. Issue Series Title: J Robot Syst 2000;17(5):233–49.
[6] Constantinescu D. Smooth time optimal trajectory planning for industrial manipulators. PhD Thesis, The University of British Columbia; 1998. Available at: 〈http://batman.mech.ubc.ca/~ial/publication〉.
[7] Lin CS, Chang PR, Luh JYS. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. IEEE Trans Automat Control 1983;28(12):1066–73.
[8] Wang CH, Horng JG. Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic B-Spline functions. IEEE Trans Automat Control 1990;35(35):573–7.
[9] Jamhour E, André PJ. Planning smooth trajectories along parametric paths. Math Comput Simulation 1996;41:615–26.
[10] Chen YC. Solving robot trajectory planning problems with uniform cubic B-splines. Optimal Control Appl Methods 1991;12:247–62.
[11] Piazzi A, Visioli A. Global minimum-time trajectory planning of mechanical manipulators using interval analysis. Int J Control 1998;71(4):631–52.
[12] Guarino Lo Bianco C, Piazzi A. A semi-infinite optimization approach to optimal spline trajectory planning of mechanical manipulators. In: Goberna e MA, López MA, editors. Semi-infinite programming: recent advances. Berlin: Springer; 2001. p. 271–97.
[13] Von Stryk O, Schlemmer M. Optimal control of the industrial robot Manutec r3. In: Bulirsch R, Kraft D, editors. Computational optimal control. International series of numerical mathematics, vol. 115. Basel: Birkhäuser; 1994. p. 367–82.
[14] Field G, Stepanenko Y. Iterative dynamic programming: an approach to minimum energy trajectory planning for robotic manipulators. Proceedings of the IEEE international conference on robotics and automation 1996:2755–60.
[15] Martin BJ, Bobrow JE. Minimum effort motions for open chain manipulators with task-dependent end-effector constraints. Int J Robot Res 1999;18(2):213–24.
[16] Shiller Z. Time–energy optimal control of articulated systems with geometric path constraints. Trans ASME J Dynam Syst Meas Control 1996;118:139–43.
[17] Saramago SFP, Steffen Jr V. Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system. Mech Mach Theory 1998;33(7):883–94.
[18] Saramago SFP, Steffen Jr V. Optimal trajectory planning of robot manipulators in the presence of moving obstacles. Mech Mach Theory 2000;35:1079–94.
[19] Bobrow JE, Martin B, Sohl G, Wang EC, Park FC, Kim J. Optimal robot motion for physical criteria. J Robot Syst 2001;18(12):785–95.
[20] Simon D. The application of neural networks to optimal robot trajectory planning. Robot Autonomous Syst 1993;11:23–34.
[21] Kyriakopoulos KJ, Saridis GN. Minimum jerk path generation. In: Proceedings of the 1988 IEEE international conference on robotics and automation, Philadelphia; 1988. p. 364–9.

[22] Simon D, Isik C. A trigonometric trajectory generator for robotic arms. Int J Control 1993;57(3):505–17.
[23] Piazzi A, Visioli A. Global minimum-jerk trajectory planning of robot manipulators. IEEE Trans Ind Electron 2000;47(1):140–9.
[24] Piazzi A, Visioli A. An interval algorithm for minimum-jerk trajectory planning of robot manipulators. In: Proceedings of the 36th conference on decision and control, San Diego, CA, December 1997. p. 1924–7.