Appl. Math. and Comp. Sci., 1997, Vol.7, No.2, 459–472

MOTION GENERATION IN CARTESIAN SPACE FOR INDUSTRIAL ROBOTS WITH ACTUATOR LIMITATIONS

YASMINA BESTAOUI*, PATRICK PLÉDEL*, MAXIME GAUTIER*

Cartesian positions, speeds and accelerations are planned to describe the desired motion of manipulators. We examine the cases of point-to-point and via-points motions, assuming that the reference values of the manipulator are represented by C^2 polynomial trajectories. Taking into account usual kinematical constraints does not make use of full dynamic capabilities of the actuators. We propose to take into account the actuators' limitations on voltages and currents, in order to minimize the arrival time. We present some simulation results and show that the adopted method is of easy use.

1. Introduction

This paper is primarily concerned with motion generation when the end-effector of the robot must follow a prescribed trajectory as a straight line or a circular arc in the Cartesian space. Motion generation consists in computing reference values to be given to the controller. Most preceding studies (Binford *et al.*, 1977; Dombre and Khail, 1988; Morlec, 1992; Paul, 1982; Plédel and Bestaoui, 1995b) refer to joint motion generation while inverse kinematics are used to transform Cartesian points to joint ones. We employ another approach where all parameters are reported in the Cartesian space (Plédel *et al.*, 1996b).

The minimum-time motion generation has been solved in a number of ways, following the usual approach, i.e. taking as feasible limits purely kinematical constraints on the velocity and acceleration (Binford *et al.*, 1977; Dombre and Khail, 1988). Conventional motion generation uses a constant bound on the acceleration (Morlec, 1992). This bound must represent the global least upper bound of all operating accelerations so as to enable the manipulator to move under any operating conditions. This implies that the full capabilities of the manipulator cannot be utilized if the conventional approach is taken. The efficiency of the robotic system can be increased by considering the characteristics of the robot dynamics at the motion generation stage. In (Kahn and Roth, 1971), the classical approach of point-to-point minimum-time control to robot arms has been applied using only a linear approximate model. In (Bessonnet, 1992) a trajectory generation based on an optimal control formulation is

* Institut de Recherche en Cybernétique de Nantes (IRCyN), UMR 6597, 1 rue de la Noë, BP 92101, 44321 Nantes Cedex 03, France, e-mail: Yasmina.Bestaoui@lan.ec-nantes.fr.

presented. Assuming that joint torques are constrained and using the Hamiltonian formulation of the dynamic model, a minimum-time cost criterion was considered. It has been shown that most often the structure of the minimum-time control requires that at least one of the actuators is always in saturation whereas the others adjust their torques so that constraints on motion are not violated while enabling the arm to reach its final desired destination (Bobrow *et al.*, 1985).

Although these results are very important theoretically, practically they are not applicable directly to an industrial robot. From a user's view point, it would be preferable to have a somewhat suboptimal but simpler solution to implement. For this purpose, we have chosen, *a priori*, a polynomial trajectory where coefficients are optimized to get a C^2 minimum-time motion. The position and orientation are considered. The resulting motion for the end-effector position is usually obvious, but the end-effector orientation motion depends on the parameters adopted to describe the orientation.

In this paper, using the computer-algebra system Maple V, we will show that the simple expressions previously obtained (Dombre and Khail, 1988; Morlec, 1992), can be numerically extended taking into account different actuator constraints.

This paper is divided into six sections. While the models and the proposed problem are formulated in Sections 2 and 3, the resolution method is stated in Section 4. Some simulation results are given in Section 5 and, finally, some conclusions are provided in the last section.

2. Models

2.1. Manipulator Model

The manipulator is assumed to be made of rigid links. Its dynamic model depends on q, $\dot{q}$ and $\ddot{q}$, being respectively the joint position, velocity and acceleration:

$$\Gamma = A(q)\ddot{q} + \dot{q}^T B(q)\dot{q} + F(q)\dot{q} + G(q) \tag{1}$$

The vector Γ is the joint input torque, G denotes the gravitational force vector, B stands for the $n \times n \times n$ Coriolis and centrifugal force matrix, F is the viscous friction and A is the $n \times n$ inertial matrix. Coulomb frictions are included in the gravitational force G and we suppose that the friction derivatives with respect to time are equal to zero.

2.2. Actuator Model

Electrical motors are very popular for driving manipulators. We focus on DC motors that are often in use as servo-motors. For a permanent magnet DC motor, the torque Γ is proportional to the armature current I. For a non-redundant multi-degree-of-freedom robot, there are usually as many actuators as the number of degrees of freedom. Then actuator dynamics for the whole robot can be characterized in a matrix form as:

$$\Gamma = K_{\mathrm{em}} I, \quad U = L\frac{\mathrm{d}I}{\mathrm{d}t} + RI + K_{\mathrm{em}}\dot{q} \tag{2}$$

L, R and K_{em} are square regular diagonal matrices representing inductance, resistance and torque constants of the robot actuators, respectively. U is the motor voltage vector. We assume the transmission from the motor to the mechanism to be perfectly rigid, i.e. the transmission does not suffer from backlash or flexibility.

2.3. Actuator Constraints

DC motors are supplied with Pulse-Width Modulation voltage amplifiers. The motors and amplifiers have limited voltages and currents. Thus usual current constraints must be coupled with constraints on the voltage and current slew rate, for every joint $1 \leq j \leq n$:

$$|I_i| \leq I_{\mathrm{max},j} \tag{3}$$

$$|U_j| \leq U_{\mathrm{max},j} \tag{4}$$

$$\left|\frac{\mathrm{d}I}{\mathrm{d}t}\right| \leq \mathrm{d}I_{\mathrm{max},j} \tag{5}$$

$I_{\mathrm{max},j}$ is the maximum armature pulse current avoiding actuator demagnetization, or the amplifier current limit. $U_{\mathrm{max},j}$ is the actuator or amplifier voltage limit and $\mathrm{d}I_{\mathrm{max},j}$ is the amplifier current slew rate limit. If $I_{\mathrm{max},j}$ is greater than the maximum allowable DC permanent current $I^{\mathrm{eff}}_{\mathrm{max},j}$ in continuous operation, the thermal limit must be taken into account to prevent overheating. Power losses are mainly resistance losses so iron and friction losses will be neglected. We will only consider the case of a periodic motion with period t_f. This is common in robotic applications. The power losses are periodic and filtered through a low-pass thermal model whose time constant is very large versus t_f. Then the motor temperature can be easily obtained with the average power which is proportional to the square root of the mean square value of the current. This must be limited by the following relation:

$$\sqrt{\frac{1}{t_f}\int_0^{t_f} I_j^2(t)\,\mathrm{d}t} \leq I^{\mathrm{eff}}_{\mathrm{max},j} \tag{6}$$

Finally, we also have to fulfil a limitation on joint speeds because of mechanical considerations:

$$|\dot{q}_j| \leq qp_{\mathrm{max},j} \tag{7}$$

In general, more restricting relations than (3)–(7) are used. Maximal accelerations and velocities, making approximations for the worst case in (1)–(7) are defined (Dombre and Khail, 1988; Morlec, 1992). The motion is not optimized with respect to real actuators' capacities.

3. Problem Formulation

3.1. Point-to-Point Motion

The desired trajectory must be chosen smooth enough so as not to excite the high-frequency unmodelled dynamics. This is the reason why we shall choose polynomials allowing for the zero speed and acceleration motion at start and end points.

In order to satisfy the previous assumption, we assume that the position and orientation of the robot are represented by using a fifth-degree polynomial interpolation in time between two points X_i and X_f (Plédel *et al.*, 1996a):

$$X(t) = X_i + Dr(t/tf), \quad D = X_f - X_i \tag{8}$$

$$r(t/t_f) = 10(t/t_f)^3 - 15(t/t_f)^4 + 6(t/t_f)^5 \tag{9}$$

For instance, if $X = [x, y, z, \varphi, \theta, \omega]^T$ using Euler angles, the position and orientation are given by (Taylor, 1979):

$$P(t) = \Big[x(t), y(t), z(t)\Big]^T \tag{10}$$

$$A(t) = \mathrm{Rot}\big(Z, \varphi(t)\big)\mathrm{Rot}\big(X, \theta(t)\big)\mathrm{Rot}\big(Z, \omega(t)\big) \tag{11}$$

For our problem, the fifth-degree polynomial is the lowest degree polynomial which satisfies the actuator limitations introduced previously.

Using the Inverse Geometrical Model (IGM), with $x = t/t_f$, gives

$$q(t) = f_{\mathrm{IGM}}\big(X(x)\big) = \breve{q}(x) \tag{12}$$

It is convenient to introduce a normalized time variable x that allows for treating each joint trajectory equation in the same way, with time varying from $x = 0$ (initial time for all joints) to $x = 1$ (final time for all joints). To get a synchronized trajectory, all joints must reach their destination position at the same time determined by the so-called restricting joint.

Joint velocities and accelerations are given as:

$$\dot{q}(t) = \frac{1}{t_f}\breve{q}_x(x), \quad \ddot{q}(t) = \frac{1}{t_f^2}\breve{q}_{xx}(x) \tag{13}$$

with

$$\breve{q}_x(x) = J^{-1}\big(\breve{q}(x)\big)\frac{\mathrm{d}X}{\mathrm{d}x}, \quad \breve{q}_{xx}(x) = J^{-1}\big(\breve{q}(x)\big)\left[\frac{\mathrm{d}^2X}{\mathrm{d}x^2} - \big\{J_q\big(\breve{q}(x)\big)\breve{q}_x(x)\big\}\breve{q}_x(x)\right]$$

J is the Jacobian matrix which is supposed to be square and regular. The study of singular points is beyond the scope of this paper. So we choose Cartesian trajectories without singularity.

We can see that the joint velocities and accelerations can be written as separate functions of x and t_f. Then we are able to rewrite the constraints (3)–(7) introducing (13) in the dynamic model of the robot (1) and the actuator model (2) (Plédel and Bestaoui, 1995b):

$$\pm qp_{\max} t_f - \breve{q}(x) \le 0 \tag{14}$$

$$\Big[\pm I_{\max} - K_{\mathrm{em}}^{-1}\tilde{Q}(x)\Big] t_f^2 - K_{\mathrm{em}}^{-1}\tilde{F}(x) t_f - K_{\mathrm{em}}^{-1}\tilde{A}(x) \le 0 \tag{15}$$

$$\pm \mathrm{d}I_{\max} t_f^3 - K_{\mathrm{em}}^{-1}\tilde{Q}(x) t_f^2 - K_{\mathrm{em}}^{-1}\tilde{F}(x) t_f - K_{\mathrm{em}}^{-1}\tilde{A}(x) \le 0 \tag{16}$$

$$\Big[\pm U_{\max} - RK_{\mathrm{em}}^{-1}\tilde{Q}(x)\Big] t_f^3 - \hat{C}(x) t_f^2 - \hat{B}(x) t_f - LK_{\mathrm{em}}^{-1}\tilde{A}(x) \le 0 \tag{17}$$

where all matrices are given by:

$$\tilde{A}(x) = A\big(\breve{q}(x)\big)\breve{q}_{xx}(x) + H\big(\breve{q}(x), \breve{q}_x(x)\big)$$

$$\tilde{Q}(x) = Q\big(\breve{q}(x)\big)$$

$$\tilde{F}(x) = F_v \breve{q}_x(x)$$

$$\tilde{A}(x) = A\big(\breve{q}(x)\big)\breve{q}_{xxx}(x) + \frac{\mathrm{d}A}{\mathrm{d}q}\big(\breve{q}(x)\big)\breve{q}_{xx}(x)\breve{q}_x(x) + 2H\big(\breve{q}(x), \breve{q}_x(x)\big)\breve{q}_{xx}(x) + \frac{\mathrm{d}H}{\mathrm{d}q}\big(\breve{q}(x), \breve{q}_x(x)\big)\big[\breve{q}_x(x)\big]^3$$

$$\tilde{F}(x) = F_v \breve{q}_{xx}(x)$$

$$\tilde{Q}(x) = \frac{\mathrm{d}Q}{\mathrm{d}q}\big(\breve{q}(x)\big)\breve{q}_x(x)\tilde{B}(x) = LK_{\mathrm{em}}^{-1}\tilde{F}(x) + RK_{\mathrm{em}}^{-1}\tilde{A}(x)$$

$$\tilde{C}(x) = K_{\mathrm{em}} D\breve{q}_x(x) + LK_{\mathrm{em}}^{-1}\tilde{Q}(x) + RK_{\mathrm{em}}^{-1}\tilde{F}(x)$$

Equations (14)–(17) are related to the constraints (7), (3), (5) and (4), respectively. Analogously, the constraint (6) becomes:

$$\big(I_{\max}^{\mathrm{eff}}\big)^2 t_f^4 - t_f^4\int_0^1 \vec{E}(x)\,\mathrm{d}x - t_f^3\int_0^1 \vec{D}(x)\,\mathrm{d}x - t_f^2\int_0^1 \vec{C}(x)\,\mathrm{d}x - t_f\int_0^1 \vec{B}(x)\,\mathrm{d}x - \int_0^1 \vec{A}(x)\,\mathrm{d}x \le 0 \tag{18}$$

with

$$\vec{A}(x) = \Big[\Big(K_{\mathrm{em},j}^{-1}\tilde{A}_j(x)\Big)^2\Big]_{1\le j\le n}$$

$$\vec{B}(x) = \Big[2K_{\mathrm{em},j}^{-1}\tilde{F}_j(x)K_{\mathrm{em},j}^{-1}\tilde{A}_j(x)\Big]_{1\le j\le n}$$

$$\vec{C}(x) = \left[\left(K_{\mathrm{em},j}^{-1}\tilde{F}_j(x)\right)^2 + 2K_{\mathrm{em},j}^{-1}\tilde{Q}_j(x)K_{\mathrm{em},j}^{-1}\tilde{A}_j(x)\right]_{1\le j\le n}$$

$$\vec{D}(x) = \left[2K_{\mathrm{em},}^{-1}\tilde{F}_j(x)K_{\mathrm{em},j}^{-1}\tilde{Q}_j(x)\right]_{1\le j\le n}$$

$$\vec{E}(x) = \left[\left(K_{\mathrm{em},j}^{-1}\tilde{Q}_j(x)\right)^2\right]_{1\le j\le n}$$

When a constraint ((3) to (6)) is saturated, the corresponding relation ((14) to (18)) is equal to zero.

3.2. Via-Points Motion

Now the manipulator has to go through $m+1$ points, with some imposed straight-line trajectories between via-points. Velocities and accelerations at these via-points are different from zero. The motion is supposed to have a continuous acceleration. Start and end-point speeds and accelerations are equal to zero. For this reason, the motion is given by a fifth-degree polynomial with respect to time between each crossing point:

$$X_{j,k}(t) = \sum_{i=0}^{5} a_{i,j,k}\left(\frac{t-t_k}{t_{f,k}}\right)^i$$
$$= \sum_{i=0}^{5} a_{i,j,k}\left(\frac{t-t_k}{t_{k+1-t_k}}\right)^i, \quad 1\le k\le m,\ 1\le j\le 6 \tag{19}$$

where t_k is the arrival time at point $k+1$ $(k=1,\dots,m)$.

But, in order to have expressions which are not explicit functions of the final time $t_{f,k}$, we introduce the following change of variables:

$$\dot{\hat{X}}_k = \dot{X}_k t_{f,k}, \qquad \ddot{\hat{X}}_k = \ddot{X}_k t_{f,k}^2, \qquad \Phi_k = \frac{t_{f,k+1}}{t_{f,k}} \tag{20}$$

The time $t_{f,k}$ is a function of Φ_k and time $t_{f,1}$:

$$t_{f,k+1} = \psi_k t_{f,1}, \quad \psi_k = \prod_{i=1}^{k}\Phi_i \tag{21}$$

Additional assumptions are added in order to obtain a straight-line motion between two adjacent via-points. Two adjacent straight lines are connected by the classical fifth-degree polynomial allowing for position, speed and acceleration continuity. Then, for every polynomial k defined on $t\in[0,\Psi_{k-1}t_{f,1}]$, (19) can be rewritten as follows:

$$\tilde{X}_{k,j}(t) = \sum_{i=0}^{5}\frac{a_{i,k,j}}{\psi_{k-1}^i}\left(\frac{t}{t_{f,1}}\right)^i, \qquad l\le j\le n,\ l\le k\le m \tag{22}$$

with

$$a_{0,k} = X_k, \qquad a_{1,k} = \mathring{X}_k, \qquad a_{2,k} = \frac{1}{2}\mathring{\mathring{X}}_k \tag{23}$$

$$a_{3,k} = 10(X_{k+1} - X_k) - \left(6\mathring{X}_k + 4\frac{\mathring{X}_{k+1}}{\Phi_k}\right) + \frac{1}{2}\left(\frac{\mathring{\mathring{X}}_{k+1}}{\Phi_k^2} - 3\mathring{\mathring{X}}_k\right)$$

$$a_{4,k} = -15(X_{k+1} - X_k) + \left(8\mathring{X}_k + 7\frac{\mathring{X}_{k+1}}{\Phi_k}\right) - \frac{1}{2}\left(2\frac{\mathring{\mathring{X}}_{k+1}}{\Phi_k^2} - 3\mathring{\mathring{X}}_k\right)$$

$$a_{5,k} = 6(X_{k+1} - X_k) - \left(3\mathring{X}_k - 3\frac{\mathring{X}_{k+1}}{\Phi_k}\right) + \frac{1}{2}\left(\frac{\mathring{\mathring{X}}_{k+1}}{\Phi_k^2} - \mathring{\mathring{X}}_k\right)$$

Joint speeds and accelerations can be expressed by (13). Then relations (14)–(18) can be obtained with $t_{f,1}$.

The general problem of minimum-time motion (PMTM) may be formulated as follows: Minimize t_f subject to the path equations and

$$\begin{cases} t_f \in \mathbb{R}^+, \quad |\dot{q}_j| \le qp_{\max,j}, \quad |I_j| \le I_{\max,j}, \quad |U_j| \le U_{\max,j} \\ \left|\dfrac{\mathrm{d}I_j}{\mathrm{d}t}\right| \le \mathrm{d}I_{\max,j}, \quad \sqrt{\dfrac{1}{t_f}\displaystyle\int_0^{t_f} I_j^2(t)\,\mathrm{d}t} \le I_{\max,j}^{\mathrm{eff}}, \quad 1 \le j \le n \end{cases} \tag{24}$$

The overall problem consists now in determining some variables $(t_{f,1}, \psi_k, \mathring{X}_k, \mathring{\mathring{X}}_k)$ so as to minimize the specified objective function t_f, subject to several equality (path) and inequality constraints (actuators). A proposed resolution method is introduced in the following section.

4. Resolution Method

4.1. Proposed Minimum-Time Approach

Optimization theory gives a solution to the PMTM. It is located on the boundary of the admissible set, i.e. when the robot moves using maximum motor capabilities. The resolution will be organized as follows. First, this problem will be solved assuming that each constraint (14)–(18) is saturated at a given time. Then the largest value of all the computed times will be taken as the predicted arrival time t_f. Let us assume that the robot moves using the maximum motor capabilities.

In what follows, we present a method to calculate t_f in the case of a point-to-point motion under technological constraints (3)–(7), or respectively (14)–(18). First, second, third and fourth-degree polynomial equations may be solved analytically.

First we consider joint speeds limitations (14). The minimal time is obtained when

$$t_{fj/qp}(x) = \frac{\check{q}(x)}{\pm qp_{\max}} \tag{25}$$

Current bounds (3) lead to the second-degree equation in t_f (15). For every joint $(j = 1, n)$, the solution $t_{fj/I}(x)$ of (15) can be approximated, neglecting the gravity and viscous frictions, by

$$t_{fj/I}(x) \approx \sqrt{\tilde{A}(x)/(\pm K_{\mathrm{em}} I_{\max})} \tag{26}$$

Then we may assume that there always exists a real positive root of $t_{fj/I}(x)$.

The candidate times t_f for the constraints on the current derivative (16) and the voltage (17) are also obtained by solving two third-degree equations in t_f. A third-degree polynomial equation may have one or three real roots from which we choose the smallest positive value. Such equations can be solved using the computer-algebra system Maple V. The solutions are called $t_{fj/\mathrm{d}I}(x)$ and $t_{fj/U}(x)$, respectively. The constraint concerning (18) leads to a fourth-degree equation in t_f. Maple V gives all the four solutions. The smallest real positive solution is called $t_{fj/I^{\mathrm{eff}}}$.

4.2. Numerical Implementation

The matrices $\tilde{A}, \ldots, \tilde{F}, \hat{A}, \ldots, \hat{C}$, and $\vec{A}, \ldots, \vec{E}$ are obtained analytically in the first step for a given robot. Then, in point-to-point motions, the numerical implementation consists in choosing the start and end points of the path (8). We then calculate, in the second step, for the given trajectory, the solutions of (14)–(17) for every x belonging to $[0, 1]$ (i.e. with a sufficient discretisation). Besides, the numerical calculus of the coefficient of (18) allows Maple to give all the solutions. The minimum time is the following maximum value:

$$t_f = \max_{1 \le j \le n} \left(\left\{ t_{fj/qp}(x), t_{fj/I}(x), t_{fj/\mathrm{d}I}(x), t_{fj/U}(x) \,\middle|\, x \in [0,1] \right\}, t_{fj/I^{\mathrm{eff}}} \right) \tag{27}$$

In the via-point problem, we calculate in the same way, for every polynomial k, the time $t_{f,1}$. The time t_f is then obtained for the maximum value in order to satisfy the contraints (3)–(7) for the whole motion.

The obtained value depends on the variables $\dot{X}_k$, $\ddot{X}_k$ and Φ_k. Those variables are optimized using the NPSOL software based on a Sequential Quadratic Programming method (SQP). NPSOL is Fortran software designed to minimize a non-linear function subject to a set of constraints. SQP belongs to the class of projected Lagrangian methods. This class includes algorithms that contain a sequence of linearly constrained subproblems based on the Lagrangian method. The idea of linearizing non-linear constraints occurs in many algorithms for non-linearly constrained optimization problems, including the reduced-gradient type methods. The subproblems involve the minimization of a general non-linear function subject to linear equality constraints and can be solved by using an appropriate technique.

For such a method, we need an initial estimate. It is obtained by considering the point-to-point motion between two adjacent points.

The exact solution of this optimal problem, without assuming that trajectories are polynomial, is very difficult to obtain, except for some very particular robots. Many methods have been presented to give an approximate solution to this problem. The well-known phase-plane method was presented in (Plédel and Bestaoui, 1995a). Bobrow *et al.* (1985) used an algorithm to determine the time-optimal trajectory of a manipulator along a specified path given dynamic constraints on torques and velocities. But, even for the joint space, the computation of this trajectory is too slow to be suitable for on-line generation but may be used for pre-planning trajectories and as a basis to compare suboptimal trajectories. A discretization method should be used to make the resolution possible (Plédel and Bestaoui, 1995a). However, this leads to a large-scale optimization problem. The implementation of this optimal method is time-consuming and not very easy from the user's point of view because of the computation time and because of the memory space needed to store the reference values of various joints.

For joint-space trajectories (Plédel and Bestaoui, 1995b; Plédel *et al.*, 1996a), we found that the fifth-degree polynomial is a good choice to approximate quasi bang-bang solutions. In our method, for on-line use, given the via-points, we need to store only the polynomial parameters to retrieve the full trajectory.

5. Numerical Examples

5.1. Robot Characteristics

The methods described in the previous sections are used to elaborate the robot's trajectory. We performed numerical simulations with a two-degree-of-freedom SCARA robot (parameters are those of the prototype robot of our laboratory) whose arm lengths are 0.5 m and 0.3 m. The values of the actuators limitations are:

$$I_{\max} = [11.53, 7.29]\,\mathrm{A}, \qquad U_{\max} = [40.0, 26.3]\,\mathrm{V}$$

$$\mathrm{d}I_{\max} = [104, 104]\,\mathrm{A/s}, \qquad I^{\mathrm{eff}}_{\max} = [12.0, 10.0]\,\mathrm{A}$$

$$qp_{\max} = [7.0, 21.0]\,\mathrm{rad/s}$$

5.2. Simulation Results

Example 1. We present an example of a point-to-point motion for constraints (3)–(7). The trajectory is a straight line (Fig. 1) between two points represented using a fifth-degree polynomial interpolation (9), see Table 1.

Table 1. Straight-line trajectory of Example 1.

	Cartesian position
start point	$[0.6, -0.4]$ m
end point	$[-0.2, 0.6]$ m

In Fig. 1, the disc represents the reach space. With our new formulation (27), involving actuator constraints, the value of the final time is $t_{fa1} = 6.4\,\text{s}$.

Figure 2 presents the current of the first joint, for which the bound is reached.

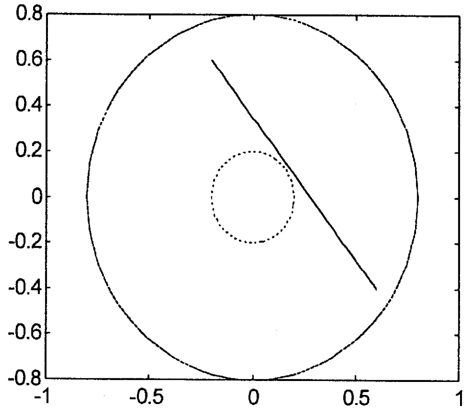

Fig. 1. Trajectory in Example 1.

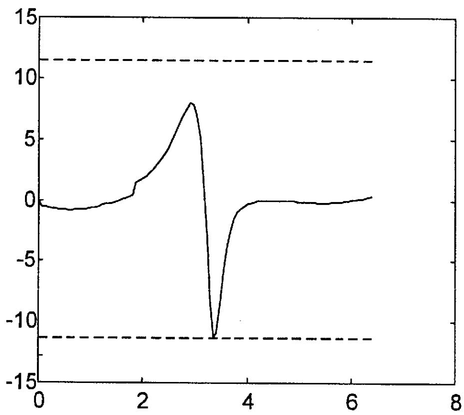

Fig. 2. Current of the first joint in Example 1.

The two-rotational-joint inverse geometrical model admits two solutions. Then the same trajectory has been tested with another configuration of the robot arms. The duration of the motion is then $t_{fa3} = 6.6\,\text{s}$. We give the current of the first joint as a function of time in Fig. 3.

Usual approaches consider bounds on the Cartesian speeds and accelerations (Morlec, 1992), or refer to constraints on torques and joint speeds (Bessonet, 1992). In the first case, the values of the bounds are not easy to define. In the second case, the values are chosen to realize a trade-off beetwen the torque and joint speed available. In both cases, those values are lower than the ones obtained when considering constraints (3)–(7).

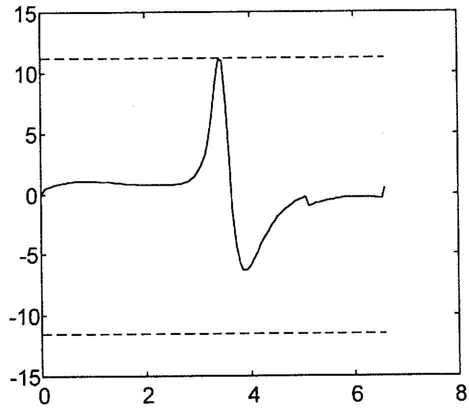

Fig. 3. Current of the first joint in Example 1 (for another configuration).

Example 2. Our formulation is also applicable to a via-point trajectory using eqn. (22). The points shown in Table 2 are defined in the Cartesian space.

Table 2. Points defining the trajectory in Example 2.

	1	2	3	4	5	6
x [m]	0.6	0.5	0.2	−0.2	−0.4	−0.1
y [m]	0.0	0.3	0.6	0.6	0.4	0.2

The time obtained for a point-to-point motion between these crossing points is $t_{fa3} = 4.9\,\mathrm{s}$. For a via-point motion with straight lines between points 1 and 2, points 3 and 4 and points 5 and 6, the time is $t_{fa3} = 3.52\,\mathrm{s}$ (Fig. 4).

For the resulting motion, the current of the first axis reaches its bound (Fig. 5). The current of the second axis is adjusted in order to follow the prescribed path (Fig. 6).

We show that all the constraints are satisfied. Then the motion is admissible. The optimal-control approach (Bessonnet, 1992) will certainly lead to shorter times. However, such approaches only consider constraints more restrictive and more unrealistic than (3)–(7). Including those constraints imposes a great cost for calculation (Paul, 1982), involving no possibility for on-line computation.

Compared with the maximal velocity and acceleration problem, even though the computation time for our formulation is longer and depends on the discretisation adopted, it leads to good results. A special fixed motion has often to be repeated thousands of times. In such cases, the generation of smooth trajectories, which can be performed in minimum time, becomes interesting even at the price of longer off-line computation times (2 min). On-line computation times, involving a few parameters (9) or (21), remain short.

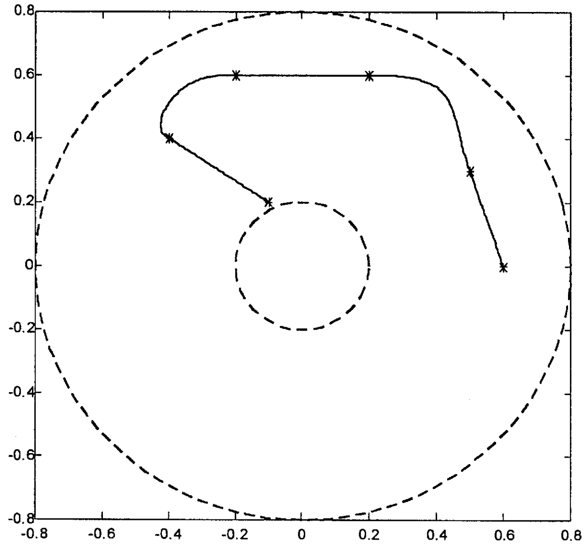

Fig. 4. Trajectory of Example 2.

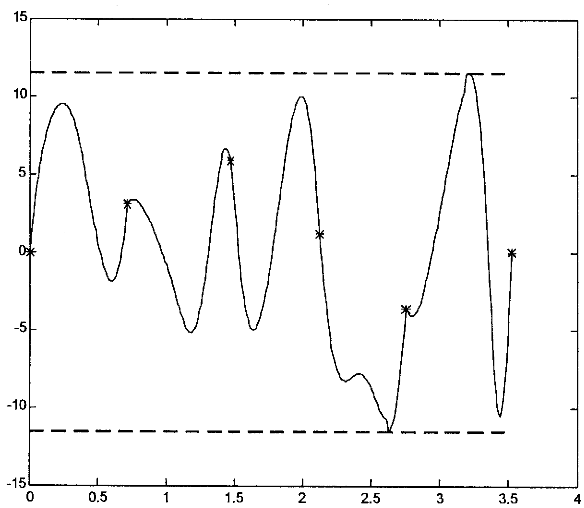

Fig. 5. Current of the first joint in Example 2.

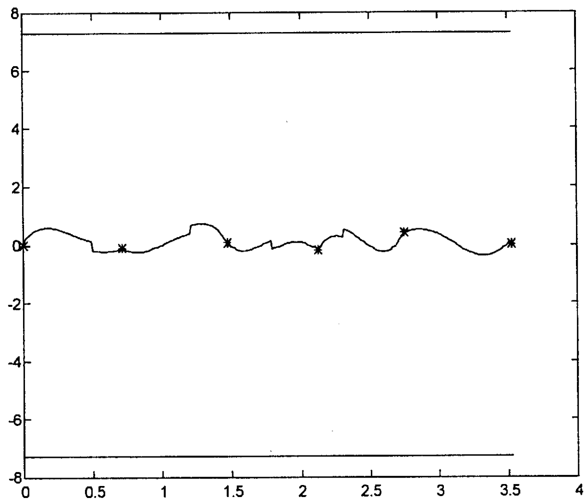

Fig. 6. Current of the second joint in Example 2.

6. Conclusions

When specifying a trajectory, the physical limits of the system must be taken into account. It is common to model these limits as constant maximum values for the acceleration and velocity. The trajectory goes from the initial to the final position with initial and final velocities equal to zero, subject to limits on the speed and acceleration. These considerations mean that even for joint level trajectories, any assumptions about fixed acceleration limits must be based on the worst case. This results in motions that are usually slower than necessary or, otherwise, the actuators may be unable to follow the requested trajectory. A more realistic assumption is that the amount of voltage and current a motor may generate is limited.

The proposed motion generation algorithm uses the solution of polynomial equations in t_f to find the predicted arrival time. Besides, the polynomial interpolation with only a few parameters, allows us to generate easily the path on-line.

Although we have considered DC motors, other actuators such as synchronous machines present the same constraints on both the current and voltage.

References

Bessonnet G. (1992): *Optimisation dynamique des mouvements point à point de robots manipulateurs.* — Doctorat d'Etat Thesis, Univ. of Poitiers, France.

Binford T.O. *et al.* (1977): *Discussion of trajectory calculation methods*, In: Exploratory study of computer integrated assembly systems. — Stanford Art. Int. Lab., Progress Report, Memo AIM-285.4, Stanford.

Bobrow J.E., Dubowsky S. and Gibson J.S. (1985): *Time optimal control of robotic manipulators along specified paths.* — Int. J. Robot. Res., Vol.4, No.3, pp.3–17.

Dombre E. and Khalil W. (1988): *Modélisation et commande des robots.* — Edition Hermès, Paris.

Kahn M. and Roth B. (1971): *The near minimum time control of an open-loop articulated kinematic chains.* — Trans. ASME J. Dyn. Syst. Meas. Contr., Vol.93, No.3, pp.164–172.

Morlec C. (1992): *Contribution au developpement d'un calculateur B-spline intégral pour commande numérique à hautes performances: intégration du module de trajectographie.* — Ph.D. Thesis, Besançon.

Paul R. (1982): *Robot Manipulators.* — Cambridge, MA: MIT Press.

Plédel P. and Bestaoui Y. (1995a): *Actuator constraints on optimal motion planning of manipulators.* — Proc. IEEE Int. Conf. *Robotics and Automation*, Nagoya, pp.2427–2432.

Plédel P. and Bestaoui Y. (1995b): *Actuator constraints in point to point motion planning of manipulators.* — Proc. IEEE Conf. *Decision and Control*, USA, New Orleans, pp.1009–1010.

Plédel P., Bestaoui Y. and Gautier M. (1996a): *Polynomial motion generation of manipulators with technological constraints.* — Proc. IEEE Int. Conf. *Robotics and Automation*, Minneapolis, pp.448–453.

Plédel P., Bestaoui Y. and Gautier M. (1996b): *A motion generation method of industrial robots in Cartesian space.* — Proc. IEEE Int. Conf. *Decision and Control*, Kobe, pp.863–868.

Taylor R.H. (1979): *Planning and execution of straight line manipulator trajectories.* — IBM J. Res. Develop. Vol.23, No.4, pp.424–436.