Article ID: 1000 - 8152(2003)02 - 0185 - 08

Theoretical and experimental research on time-optimal trajectory planning and control of industrial robots

TAN Guan-zheng[1], WANG Yue-chao[2]

(1. Institute of Robotics, Central South University, Hunan Changsha 410083, China;
2. Robotics Laboratory, Chinese Academy of Sciences, Liaoning Shengyang 110016, China)

Abstract: A new method used for time-optimal trajectory planning and control of industrial robots is proposed, which can ensure the motion of a robot's hand along a specified path in Cartesian space has the minimum traveling time under the constraints on the boundary values of joint displacements, velocities, accelerations, and jerks. In this method, the planned joint trajectories are all expressed by a quadratic polynomial plus a cosinoidal function and are continuous not only in displacements, velocities, accelerations but also in jerks. By using the method, a robot's working efficiency can be raised and its life span can be extended. The results of computer simulation and experiment with a Unimate PUMA 560 type robot proves that this method is correct and effective. It provides a better solution to the problem of industrial robot's time-optimal trajectory planning and control under the nonlinear kinematical constraints.

Key words: industrial robot; time-optimal; trajectory planning; trajectory control; quadratic polynomial; cosinoidal function; optimization algorithm

CLC number: TP24 **Document code**: A

工业机器人时间最优轨迹规划及轨迹控制的理论与实验研究

谭冠政[1]，王越超[2]

(1. 中南大学 机器人研究所,湖南 长沙 410083; 2. 中国科学院 机器人学开放研究实验室,辽宁 沈阳 110016)

摘要：提出了一种用于工业机器人时间最优轨迹规划及轨迹控制的新方法，它可以确保在关节位移、速度、加速度以及二阶加速度边界值的约束下，机器人手部沿笛卡尔空间中规定路径运动的时间最短.在这种方法中,所规划的关节轨迹都采用二次多项式加余弦函数的形式，不仅可以保证各关节运动的位移、速度、加速度连续而且还可以保证各关节运动的二阶加速度连续.采用这种方法，既可以提高机器人的工作效率又可以延长机器人的工作寿命.以 PUMA 560 机器人为对象进行了计算机仿真和机器人实验，结果表明这种方法是正确和有效的.它为工业机器人在非线性运动学约束条件下的时间最优轨迹规划及控制问题提供了一种较好的解决方案.

关键词：工业机器人；时间最优；轨迹规划；轨迹控制；二次多项式；余弦函数；最优化算法

1 Introduction

The problem of time-optimal trajectory planning and control for industrial robots means optimizing the joint motion trajectories of a robot by taking the minimum traveling time as the performance index so that the motion of the robot's hand between two points or along a specified path in Cartesian space has the minimum traveling time. Conducting this research is of practical significance for raising the working efficiency of industrial robots.

So far, many researchers have studied this problem and developed many methods to solve it. These methods can be divided roughly into three categories: 1) use the constraints on joint velocities and/or accelerations, and choose optimum joint velocity and/or acceleration profiles[1,2]; 2) use the maximum principle[3,4]; 3) use various optimization algorithms[5,6]. Moreover, some researchers studied this problem using neural networks or genetic algorithms[7].

The research method adopted by the authors of this paper falls under the third category. In this paper, a new and effective method for the time-optimal trajectory pla-

Received date: 2001 - 11 - 13 ; Revised date: 2002 - 08 - 08.

Foundation item: supported by the Foundation of Robotics Laboratory, Chinese Academy of Sciences (RL200002).

nning and control of industrial robots will be proposed, in which each joint trajectory of a industrial robot will be expressed by a quadratic polynomial plus a cosinoidal function and the constraint on joint jerk (the rate of change of acceleration) continuity will be added to this optimization problem. This method can ensure the motion of a robot's hand along a specified path in Cartesian space has the minimum traveling time under the constraints on the boundary values of joint displacements, velocities, accelerations, and jerks. These optimized joint trajectories can prevent a robot's mechanical parts from excessive wear and tear. This method can not only raise a robot's working efficiency but also extend its life span.

2 Formulation of joint trajectories

In Cartesian space, the hand position and orientation of an industrial robot can be expressed by the following 4×4 homogeneous transformation matrix[6]:

$$\boldsymbol{H}(\mathrm{t}) = \begin{bmatrix} \boldsymbol{n}(t) & \boldsymbol{s}(t) & \boldsymbol{a}(t) & \boldsymbol{p}(t) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where $\boldsymbol{p}$ is the position vector of the hand; $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{a}$ are the unit normal, unit slide, and unit approach vectors, respectively. These vectors are all expressed with respect to the base coordinate frame $O_0 - \boldsymbol{X}_0\boldsymbol{Y}_0\boldsymbol{Z}_0$ of the robot[6].

Assume that m knots are selected from the initial point to the terminal point along a specified path of a robot's hand in Cartesian space, and the times corresponding to these knots are $t_1, t_2, \cdots, t_m$ in order. To construct joint trajectories, these knots $\boldsymbol{H}(t_i), i = 1,2, \cdots, m$, must be first transformed into joint vectors $[q_{11}, q_{21}, \cdots, q_{N1}]$, $[q_{12}, q_{22}, \cdots, q_{N2}], \cdots$, $[q_{1m}, q_{2m}, \cdots, q_{Nm}]$ by use of the inverse kinematical equations of the robot, where q_{ji} is the displacement of joint j at knot i corresponding to $\boldsymbol{H}(t_i)$ and N is the number of joints of the robot. Secondly, a suitable function must be selected for each joint to fit joint displacements $q_{j1}, q_{j2}, \cdots, q_{jm}$ together. In this section, we will adopt a quadratic polynomial plus a cosinoidal function to construct trajectory for each joint. Because the procedure of constructing joint trajectories deals with one joint at a time and joint number j is not necessary to be specified, q_{ji} will be replaced by q_i for simplicity.

Assume that at the initial time $t = t_1$, the joint displacement q_1, velocity v_1, and acceleration a_1 are specified; so as q_m, v_m, and a_m at the terminal time $t = t_m$. In addition, joint displacements q_i at $t = t_i$ for $i = 3,4, \cdots, m-2$ are also specified. However, in order to give enough freedom to solve the optimization problem with the constraints, q_2 and q_{m-1} are not fixed, which are referred to as extra knot. Let $Q_i(t)$ be a quadratic polynomial plus a cosinoidal function defined on the time interval $[t_i, t_{i+1}]$. The aim of trajectory planning is to spline $Q_i(t)$, for $i = 1,2,\cdots, m-1$, together so that the joint displacement, velocity, acceleration, and jerk are continuous on the entire time interval $[t_1, t_m]$; and the motion of the robot's hand along a specified path in Cartesian space has the minimum traveling time under the constraints on boundary values of joint displacements, velocities, accelerations, and jerks.

Because $Q_i(t)$ is a quadratic polynomial plus a cosinoidal function, the second time derivative $\ddot{Q}_i(t)$ must be a constant plus a cosinoidal function. Hence, $\ddot{Q}_i(t)$ can be expressed as:

$$\begin{aligned} &\ddot{Q}_i(t) = k_1 + k_2\cos\left[\frac{\pi}{h_i}(t - t_i)\right], \\ &t \in [t_i, t_{i+1}],\ i = 1,2,\cdots, m-1, \end{aligned} \quad (2)$$

where $h_i = t_{i+1} - t_i$, k_1 and k_2 are two undetermined constants.

Let t in (2) be equal to t_i and t_{i+1} respectively, we can obtain k_1 and k_2. Substituting k_1 and k_2 into (2), we obtain the expression for joint acceleration as follows:

$$\begin{aligned} \ddot{Q}_i(t) = &\frac{1}{2}[\ddot{Q}_i(t_i) + \ddot{Q}_i(t_{i+1})] + \\ &\frac{1}{2}[\ddot{Q}_i(t_i) - \ddot{Q}_i(t_{i+1})]\cos\left[\frac{\pi}{h_i}(t - t_i)\right]. \end{aligned} \quad (3)$$

Integrating $\ddot{Q}_i(t)$ twice and imposing the conditions $Q_i(t_i) = q_i$ and $Q_i(t_{i+1}) = q_{i+1}$, we obtain the expression for joint displacement as follows:

$$Q_i(t) = \left\{q_i + \frac{h_i^2}{2\pi^2}[\ddot{Q}_i(t_i) - \ddot{Q}_i(t_{i+1})]\right\} +$$

$$\left\{\frac{q_{i+1}-q_i}{h_i} - \frac{h_i}{4}[\ddot{Q}_i(t_i) + \ddot{Q}_i(t_{i+1})] - \frac{h_i}{\pi^2}[\ddot{Q}_i(t_i) - \ddot{Q}_i(t_{i+1})]\right\}(t - t_i) +$$
$$\frac{1}{4}[\ddot{Q}_i(t_i) + \ddot{Q}_i(t_{i+1})](t - t_i)^2 -$$
$$\frac{h_i^2}{2\pi^2}[\ddot{Q}_i(t_i) - \ddot{Q}_i(t_{i+1})]\cos\left[\frac{\pi}{h_i}(t - t_i)\right]. \quad (4)$$

Differentiating $Q_i(t)$ once and three times respectively, we obtain the expressions for joint velocity $\dot{Q}_i(t)$ and joint jerk $\dddot{Q}_i(t)$ as follows:

$$\dot{Q}_i(t) = \left\{\frac{q_{i+1}-q_i}{h_i} - \frac{h_i}{4}[\ddot{Q}_i(t_i) + \ddot{Q}_i(t_{i+1})] - \frac{h_i}{\pi^2}[\ddot{Q}_i(t_i) - \ddot{Q}_i(t_{i+1})]\right\} +$$
$$\frac{1}{2}[\ddot{Q}_i(t_i) + \ddot{Q}_i(t_{i+1})](t - t_i) +$$
$$\frac{h_i}{2\pi}[\ddot{Q}_i(t_i) - \ddot{Q}_i(t_{i+1})]\sin\left[\frac{\pi}{h_i}(t - t_i)\right], \quad (5)$$

$$\dddot{Q}_i(t) = -\frac{\pi}{2h_i}[\ddot{Q}_i(t_i) - \ddot{Q}_i(t_{i+1})]\sin\left[\frac{\pi}{h_i}(t - t_i)\right]. \quad (6)$$

Because $\dddot{Q}_i(t)$ is a sinusoidal function, its values at the two ends of $[t_i, t_{i+1}]$ are all equal to zero, which makes the joint jerk be automatically continuous at the juncture of two adjacent time intervals. This is why the variable of the cosinoidal function in (2) is taken the form of $\pi(t - t_i)/h_i$. Because the joint jerk is continuous in the process of motion, the acceleration profile of each joint does not generate zigzag lines. Obviously, this is beneficial to the extension of the life spans of mechanical parts.

According to (3) ~ (6), $Q_i(t)$, $\dot{Q}_i(t)$, $\ddot{Q}_i(t)$, and $\dddot{Q}_i(t)$ can be determined if $\ddot{Q}_i(t_i)$ and $\ddot{Q}_i(t_{i+1})$ are known. The following is the introduction to the solution for $\ddot{Q}_2(t_2), \ddot{Q}_3(t_3), \cdots, \ddot{Q}_{m-1}(t_{m-1})$.

According to the continuity condition for velocities, we obtain $\dot{Q}_{i-1}(t_i) = \dot{Q}_i(t_i), i = 2,3,\cdots,m-1$, which leads to the following equations by use of (3) and (5):

$$\frac{h_{i-1}}{h_i}\ddot{Q}_{i-1}(t_{i-1}) + \left(\frac{\pi^2+4}{\pi^2-4}\right)\left(\frac{h_{i-1}+h_i}{h_i}\right)\ddot{Q}_i(t_i) + \ddot{Q}_i(t_{i+1}) =$$
$$\frac{4\pi^2}{(\pi^2-4)h_i}\left(\frac{q_{i+1}-q_i}{h_i} - \frac{q_i-q_{i-1}}{h_{i-1}}\right), \ i = 2,3,\cdots,m-1. \quad (7)$$

The unspecified joint displacements of the two extra knots can be expressed in terms of initial conditions at the beginning and terminal knots together with $\ddot{Q}_1(t_2)$ and $\ddot{Q}_{m-1}(t_{m-1})$. By use of (3) ~ (5) and letting i be equal to 1 and $m-1$ respectively, we can obtain

$$q_2 = q_1 + v_1h_1 + \frac{\pi^2+4}{4\pi^2}h_1^2a_1 + \frac{\pi^2-4}{4\pi^2}h_1^2\ddot{Q}_1(t_2), \quad (8)$$

$$q_{m-1} = q_m - v_mh_{m-1} + \frac{\pi^2+4}{4\pi^2}h_{m-1}^2a_m + \frac{\pi^2-4}{4\pi^2}h_{m-1}^2\ddot{Q}_{m-1}(t_{m-1}). \quad (9)$$

Let $i = 2,3,\cdots,m-1$, respectively, $(m-2)$ equations can be obtained using (7). By means of these equations, the following matrix equation can be obtained:

$$B\begin{bmatrix} \ddot{Q}_2(t_2) \\ \ddot{Q}_3(t_3) \\ \vdots \\ \ddot{Q}_{m-1}(t_{m-1}) \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{m-2} \end{bmatrix}, \quad (10)$$

where the matrix B has the following form:

$$B = \begin{bmatrix} b_{11} & b_{12} & 0 & 0 & \cdots & 0 & 0 \\ b_{21} & b_{22} & b_{23} & 0 & \cdots & 0 & 0 \\ 0 & b_{32} & b_{33} & b_{34} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & b_{m-2,m-3} & b_{m-2,m-2} \end{bmatrix}. \quad (11)$$

The elements of the matrix B and $c_1, c_2, \cdots, c_{m-2}$ can be obtained from the left side and right side of (7) respectively. They consist of the given values of q's, v's, a's, and h's.

A program called SUA (solving for unknown accelerations) has been written, the purpose of which is to solve (10) for the unknowns $\ddot{Q}_i(t_i), i = 2,3,\cdots,m-1$.

Property 1 The trajectory problem described above has a unique solution, i.e., the system matrix B of (10) is nonsingular.

The proof of this property is omitted because of the limitation of the paper length.

3 Description of optimization problem

First formulas (3) ~ (6) are modified as follows: a) the joint number j is added to them; b) let $Q_{ji}(t)$ be the joint trajectory equation for joint j between knots i and $i+1$; c) let $A_{ji} = \ddot{Q}_{ji}(t_i)$ and $A_{j,i+1} = \ddot{Q}_{ji}(t_{i+1})$. Secondly, let VC_j, AC_j, and JC_j be the velocity, acceleration, and jerk constraints for joint j.

The optimization problem is then described as follows:

Constraints:

$$\begin{cases} |\dot{Q}_{ji}(t)| \leqslant VC_j, \\ |\ddot{Q}_{ji}(t)| \leqslant AC_j, \\ |\dddot{Q}_{ji}(t)| \leqslant JC_j, \\ j = 1,2,\cdots,N,\ i = 1,2,\cdots,m-1. \end{cases} \tag{12}$$

Objective function: minimize

$$T(X) = h_1 + h_2 + \cdots + h_{m-1} = \sum_{i=1}^{m-1} h_i, \tag{13}$$

where $X = [h_1, h_2, \cdots, h_{m-1}]$ is the vector of time intervals. The above constraints can be expressed in explicit forms as follows.

1) Velocity constraints:

$$|\dot{Q}_{ji}(t)| \leqslant VC_j,\ j = 1,2,\cdots,N,\ i = 1,2,\cdots,m-1. \tag{14}$$

The maximum absolute value of velocity exists at t_i, t_{i+1}, or t_i^*, where t_i^* satisfies $\ddot{Q}_{ji}(t_i^*) = 0$ and is in the interval $[t_i, t_{i+1}]$. The velocity constraints then become

$$\max_{t\in[t_i,t_{i+1}]} |\dot{Q}_{ji}(t)| =$$

$$\max\{|\dot{Q}_{ji}(t_i)|, |\dot{Q}_{ji}(t_{i+1})|, |\dot{Q}_{ji}(t_i^*)|\} \leqslant VC_j,$$
$$i = 1,2,\cdots,m-1,\ j = 1,2,\cdots,N. \tag{15}$$

2) Acceleration constraints:

$$|\ddot{Q}_{ji}(t)| \leqslant AC_j,\ j = 1,2,\cdots,N,\ i = 1,2,\cdots,m-1. \tag{16}$$

Observing (3) and (6), it can be found that the maximum absolute value of acceleration exists at either t_i or t_{i+1} on $[t_i, t_{i+1}]$ and equals $\max\{|A_{ji}|, |A_{j,i+1}|\}$. Hence, the acceleration constraints become

$$\max\{|A_{j1}|, |A_{j2}|, \cdots, |A_{jm}|\} \leqslant AC_j,$$
$$j = 1,2,\cdots,N. \tag{17}$$

3) Jerk constraints:

$$|\dddot{Q}_{ji}(t)| \leqslant JC_j,\ j = 1,2,\cdots,N,\ i = 1,2,\cdots,m-1. \tag{18}$$

Observing (6), it can be found that the maximum absolute value of jerk is a constant $|-\pi(A_{ji} - A_{j,i+1})/(2h_i)|$ on $[t_i, t_{i+1}]$. Consequently, the jerk constraints can be represented by

$$\left|\frac{\pi}{2h_i}(A_{ji} - A_{j,i+1})\right| \leqslant JC_j,$$
$$i = 1,2,\cdots,m-1,\ j = 1,2,\cdots,N. \tag{19}$$

Property 2 The solution is always feasible for the optimization problem with constraints (15), (17), and (19).

According to Property 1, $A_{j2}, A_{j3}, \cdots, A_{j,m-1}$ can be uniquely determined if X is given. But, the constraints on joint velocities, accelerations, and jerks might not be satisfied. In this case, we can follow the method used in Reference [6], i.e., expand all the elements of vector X in the same proportion to bring the unsatisfied velocities, accelerations, and jerks down to the constrained values. The validity of this procedure is explained as follows.

Assuming λ to be a constant greater than 1, if time is lengthened according to $\tau = \lambda t$, then the new time interval becomes $h_i^* = \lambda h_i$ and $[t_i, t_{i+1}]$ becomes $[\tau_i, \tau_{i+1}] = [\tau_i, \tau_i + h_i^*]$. Replacing t_i, t_{i+1}, h_i, $\ddot{Q}_i(t_i)$, $\ddot{Q}_i(t_{i+1})$, $Q_i(t_i)$, $\dot{Q}_i(t_i)$, and $\dddot{Q}_i(t_i)$ in (3) ~ (6) with $\tau_i, \tau_{i+1}, h_i^*, \ddot{Q}_i(\tau_i), \ddot{Q}_i(\tau_{i+1}), Q_i(\tau_i), \dot{Q}_i(\tau_i)$, and $\dddot{Q}_i(\tau_i)$, respectively, and replacing t and h_i in (7) with τ and h_i^* respective, we obtain five new equations. By means of these equations, we can obtain the following relations:

$$\begin{cases} Q_i(\tau) = Q_i(t), \\ \dot{Q}_i(\tau) = \dot{Q}_i(t)/\lambda, \\ \ddot{Q}_i(\tau) = \ddot{Q}_i(t)/\lambda^2, \\ \dddot{Q}_i(\tau) = \dddot{Q}_i(t)/\lambda^3, \end{cases} \tag{20}$$

which indicate that if vector X is replaced by $[\lambda h_1, \lambda h_2, \cdots, \lambda h_{m-1}]$, then the joint displacements will not change but the joint velocities, accelerations, and jerks will be reduced to $1/\lambda$, $1/\lambda^2$, and $1/\lambda^3$ times of their respective values. These changes ensure that the constraints on velocities, accelerations, and jerks can be satisfied. Following Reference [6], λ is called 'the feasibility adjustment factor'. The procedure that converts an infeasible point into a feasible one is called 'the feasible solution converter (FSC)'.

Let

$$\lambda_1 = \max_j\{[\max_i(\max_{t\in[t_i,t_{i+1}]} |\dot{Q}_{ij}(t)|)]/VC_j\}, \quad (21)$$

$$\lambda_2 = \max_j [\max_i |A_{ji}|/AC_j], \quad (22)$$

$$\lambda_3 = \max_j\Big[\max_i \left|\frac{\pi}{2h_i}(A_{ji} - A_{j,i+1})\right|/JC_j\Big], \quad (23)$$

then λ can be determined by the following formula:

$$\lambda = \max(1, \lambda_1, \sqrt{\lambda_2}, \sqrt[3]{\lambda_3}). \quad (24)$$

After λ is determined, we can compute the joint displacements, velocities, accelerations, and jerks by means of the formula set (20).

4 Algorithm structure for solving the optimization problem

The Nelder and Mead's flexible polyhedron search algorithm[8] is used to develop the optimization algorithm. First an initial polyhedron must be formed. Let $q_{j1}, q_{j3}, \cdots, q_{j,m-2}, q_{jm}$ denote the displacement sequence of joint j. The displacements of the two extra knots are temporarily assigned as:

$$q_{j2} = (q_{j1} + q_{j3})/2, \quad (25)$$

$$q_{j,m-1} = (q_{j,m-2} + q_{jm})/2. \quad (26)$$

Thus, the lower bound of the vector of time intervals can be estimated as:

$$X' = [h'_1, h'_2, \cdots, h'_{m-1}] = \left[\max_j \frac{|q_{j2}-q_{j1}|}{VC_j}, \max_j \frac{|q_{j3}-q_{j2}|}{VC_j}, \cdots, \max_j \frac{|q_{jm}-q_{j,m-1}|}{VC_j}\right]. \quad (27)$$

Because the vector X is $(m-1)$ dimensional, the flexible polyhedron should consist of m vertices. The first vertex X_1^0 is selected as X' if X' is feasible or selected as the feasible vertex converted from X' by FSC. The remaining vertices $X_2^0, X_3^0, \cdots, X_m^0$ can be determined as follows. First the regular polyhedron is computed as follows:

$$\begin{cases} X'_2 = X_1^0 + [d_1, d_2, d_2, \cdots, d_2], \\ X'_3 = X_1^0 + [d_2, d_1, d_2, \cdots, d_2], \\ \quad\vdots \\ X'_m = X_1^0 + [d_2, d_2, d_2, \cdots, d_1], \end{cases} \quad (28)$$

where

$$\begin{cases} d_1 = \dfrac{D}{(m-1)\sqrt{2}}(\sqrt{m} + m - 2), \\ d_2 = \dfrac{D}{(m-1)\sqrt{2}}(\sqrt{m} - 1), \end{cases} \quad (29)$$

and D is the distance between arbitrary two vertices of the regular polyhedron. Let $X_1^0 = [h_1^0, h_2^0, \cdots, h_{m-1}^0]$. According to the suggestion of Nelder and Mead, D is computed as follows:

$$D = 10\min\{\frac{0.2}{m-1}\sum_{i=1}^{m-1}(h_i^0 - h'_i), (h_1^0 - h'_1), (h_2^0 - h'_2), \cdots, (h_{m-1}^0 - h'_{m-1})\}. \quad (30)$$

Then, X_i^0 is selected as X'_i if X'_i is feasible or X_i^0 is selected as the feasible vertex converted from X'_i by FSC.

Among the m vertices of initial polyhedron, let X_g, X_l, and X_s have the greatest, second greatest, and least objective function, respectively. Let X_{m+1} be the centroid of vertices except X_g. X_{m+1} is calculated as:

$$X_{m+1} = \frac{1}{m-1}[(\sum_{i=1}^{m} X_i^0) - X_g]. \quad (31)$$

The purpose of flexible polyhedron search algorithm is to select a better feasible vertex, which corresponds to a smaller value of the objective function, to replace the worst vertex X_g. Operations for searching a better vertex include reflection, expansion, contraction, and reduction. They are defined as follows.

1) Reflection: Reflect X_g through the centroid by computing

$$X_{m+2} = X_{m+1} + \alpha(X_{m+1} - X_g), \quad (32)$$

where $\alpha > 0$ is the reflection coefficient. According to Reference [6], α is calculated as

$$\alpha = \begin{cases} 1, \text{ if } 2h_i^{m+1} - h_i^g > 0 \text{ for all } i \\ \delta_1 \cdot \left\{ \max\limits_{\substack{\text{for those } i \\ \text{with } 2h_i^{m+1}-h_i^g \le 0}} [\dfrac{h_i^{m+1}}{h_i^g - h_i^{m+1}}] \right\}, \\ \quad \text{if } 2h_i^{m+1} - h_i^g \le 0 \text{ for some } i, \end{cases} \quad (33)$$

where $0 < \delta_1 < 1$ is selected to keep X_{m+2} away from the boundary where at least one element of X_{m+2} is zero.

2) Expansion: Expand the vector $(X_{m+2} - X_{m+1})$ by computing

$$X_{m+3} = X_{m+1} + \gamma(X_{m+2} - X_{m+1}), \quad (34)$$

where $\gamma > 1$ is the expansion coefficient. According to Reference [6], γ is calculated as

$$\gamma = \begin{cases} 2, \text{ if } 2h_i^{m+2} - h_i^{m+1} > 0 \text{ for all } i, \\ \delta_2 \cdot \left\{ \max\limits_{\substack{\text{for those } i \\ \text{with } 2h_i^{m+2}-h_i^{m+1} \leqslant 0}} \left[\dfrac{h_i^{m+1}}{h_i^{m+1} - h_i^{m+2}} \right] \right\}, \\ \quad \text{if } 2h_i^{m+2} - h_i^{m+1} \leqslant 0 \text{ for some } i, \end{cases} \quad (35)$$

where $0 < \delta_2 < 1$.

3) Contraction: Contract the vector $(X_{m+2} - X_{m+1})$ or $(X_g - X_{m+1})$ by computing

$$X_{m+4} = X_{m+1} + \beta(X_{m+2} - X_{m+1}) \quad (36)$$

or

$$X_{m+4} = X_{m+1} + \beta(X_g - X_{m+1}), \quad (37)$$

where $0 < \beta < 1$ is the contraction coefficient.

4) Reduction: Reduce all the vectors $(X_i - X_s)$, $i = 1,2,\cdots,m$, by one half from X_s by computing

$$X_i \leftarrow X_s + 0.5(X_i - X_s), \; i = 1,2,\cdots,m. \quad (38)$$

After the initial flexible polyhedron is formed, the algorithm should enter an iterative process. When $\sum\limits_{i=1}^{m} \| X_i^k - X_s^k \| < \varepsilon_1$ (ε_1 is a preselected small number and k denotes stage number), the polyhedron will become very small and the solution is obtained at a local optimum point. To further search the global optimum solution, X_s^k will be used as the first vertex for setting up another new initial polyhedron, and then another cycle of iterative reduction process will be started again. In the algorithm given below, whenever the iterative process reaches Step14, it completes one iterative cycle; whenever the process reaches Step13, it completes one iterative stage of that particular cycle. kk is the cycle number and k is the stage number. If the difference of the solutions from two consecutive cycles is less than a preselected small number ε_2, the vector of optimal time intervals is obtained.

Optimization Algorithm

Step 1 Select $(m-2)$ knots from the initial point to the terminal point along the specified hand path in Cartesian space, which are labeled with $1, 3, 4, \cdots, m-2$, and m in order. Knot 2 and knot $m-1$ are the two extra ones that are not selected on the path.

Step 2 Call the program for solving the inverse kinematical equations of the considered robot to transform the Cartesian coordinates of the selected $(m-2)$ knots into the corresponding joint displacements $q_{j1}, q_{j3}, q_{j4}, \cdots, q_{j,m-2}, q_{jm}, j = 1,2,\cdots,N$. Compute the displacements of the two extra knots by (8) and (9).

Step 3 Set $kk = 0$ and $k = 0$. Select δ_1, δ_2, and β in (33), (35), and (36), (37). Also select ε_1 and ε_2. Set OLD $X_s = [0, 0, \cdots, 0]$.

Step 4 Compute X' by (27). If X' is feasible, then set $X_1^k = X'$; else set X_1^k = the feasible vertex converted from X' using FSC.

Step 5 For $i = 2,3,\cdots,m$, compute X_i' by (28). If X_i' is feasible, then set $X_i^k = X_i'$; else set X_i^k = the feasible vertex converted from X_i' using FSC.

Step 6 From $X_1^k, X_2^k, \cdots, X_m^k$, determine X_g^k, X_l^k, and X_s^k, which have the greatest, second greatest, and least value of the objective function, respectively.

Step 7 Compute the centroid X_{m+1}^k by (31).

Step 8 Reflect to obtain X_{m+2}^k by (32) and (33). Make X_{m+2}^k feasible by FSC if X_{m+2}^k is originally infeasible.

Step 9 If $T(X_{m+2}^k) < T(X_s^k)$, then follow a) and b) specified below; else go to Step 10.

a) Expand to obtain X_{m+3}^k by (34) and (35). Make X_{m+3}^k feasible by FSC if it is infeasible originally.

b) Set

$$X_g^k = \begin{cases} X_{m+2}^k, \text{ if } T(X_{m+2}^k) \leqslant T(X_{m+3}^k), \\ X_{m+3}^k, \text{ if } T(X_{m+2}^k) > T(X_{m+3}^k), \end{cases}$$

then set $k = k + 1$ and go to Step 13.

Step 10 If $T(X_{m+2}^k) < T(X_l^k)$, set $X_g^k = X_{m+2}^k$, then set $k = k + 1$ and go to Step 13. If $T(X_l^k) \leqslant T(X_{m+2}^k) < T(X_g^k)$, contract to obtain X_{m+4}^k by (36).

If $T(X_{m+2}^k) \geqslant T(X_g^k)$, contract to obtain X_{m+4}^k by (37). Convert X_{m+4}^k to become feasible by FSC if it is infeasible originally.

Step 11 If $T(X_{m+4}^k) < T(X_g^k)$, set $X_g^k = X_{m+4}^k$, then set $k = k + 1$ and go to Step 13. Otherwise continue the step.

Step 12 Reduce the size of the polyhedron using (38). Make those vertices feasible by FSC if they are originally infeasible. Then, find the new best vertex X_s^k with the least value of the objective function, and set $k = k + 1$. Continue the next step.

Step 13 If $\sum_{i=1}^{m} \| X_i^k - X_s^k \| < \varepsilon_1$, then go to Step14; else go to Step 6.

Step 14 If $\| X_s^k - \text{OLD } X_s \| < \varepsilon_2$, then output X_s^k as the vector of optimal time intervals and go to Step 15; else set $X_1^0 = X_s^k$, and OLD $X_s = X_s^k$. Then set $kk = kk + 1, k = 0$, and go to Step 5.

(* Time-optimal trajectory planning *)

Step 15 Call SUA program using the optimum solution X_s^k as the input to obtain joint accelerations $A_{j2}, A_{j3}, \cdots, A_{j,m-1}, j = 1,2,\cdots,N$, and the displacements of the two extra knots, i. e., q_{j2} and $q_{j,m-1}, j = 1,2,\cdots,N$.

Step 16 Set sampling period and compute the joint displacements, velocities, accelerations, and jerks at each sampling point by use of (3) ~ (6). These data can be used for the on-line path tracking of the robot or as the result of computer simulation.

5 An illustrative example

In this section, a Unimate PUMA 560 type robot with six revolute joints will be used to examine the effectiveness of the above time-optimal trajectory planning and control method.

Eight knots are selected from a specified Cartesian hand path. By means of the inverse kinematical equations, the joint displacements are solved for these knots as shown in Table 1.

Assume that the robot is at rest initially and comes to a full stop at the end of the path. Thus, $v_{j1} = 0, a_{j1} = 0, v_{jm} = 0, a_{jm} = 0, j = 1,2,\cdots,6, m = 10$. The velocity, acceleration, and jerk constraints of the robot are given in Table 2.

Table 1 Joint displacements of selected knots

knot	Joint					
	1	2	3	4	5	6
1	43.35	7.37	130.57	0	39.06	-46.66
2			(extra knot)			
3	43.33	-18.00	152.09	0	45.90	48.02
4	50.04	-41.85	170.66	0	51.19	-32.35
5	62.67	-53.76	179.41	0	53.59	-4.99
6	78.04	-57.32	182.70	0	54.62	33.04
7	94.40	-52.73	178.45	0	53.38	75.94
8	104.13	-42.46	173.54	0	50.33	94.76
9			(extra knot)			
10	111.91	6.79	132.80	0	40.41	112.16

Table 2 Velocity, acceleration, and jerk constraints

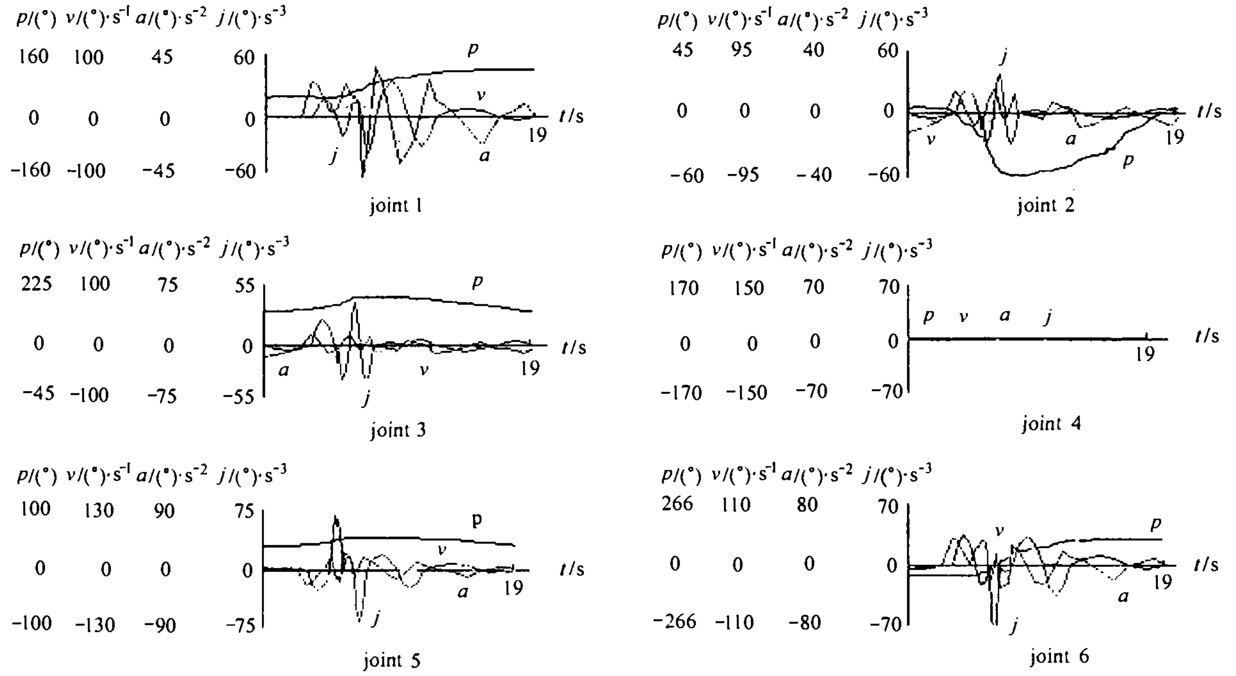
Constraints	Joint					
	1	2	3	4	5	6
Velocity $v/(°)\cdot s^{-1}$	100	95	100	150	130	110
Acceleration $a/(°)\cdot s^{-2}$	45	40	75	70	90	80
Jerk $j/(°)\cdot s^{-3}$	60	60	55	70	75	70

The relevant parameters are selected as $\delta_1 = 0.85$, $\delta_2 = 0.85, \beta = 0.5, \varepsilon_1 = 0.1$, and $\varepsilon_2 = 0.1$. For this example, the optimization algorithm performs five cycles of search sequences to obtain the optimum solution. The results from the five cycles are given in Table 3. From the table, the total traveling time is reduced in each cycle, and the final value is 18.907 seconds.

Table 3 Results from the five cycles

h_i	Cycle				
	1	2	3	4	5
h_1	4.825	3.483	3.307	3.272	3.270
h_2	0.938	1.778	1.722	1.674	1.673
h_3	0.903	0.948	0.934	0.927	0.926
h_4	0.857	0.783	0.753	0.747	0.747
h_5	1.135	0.928	0.927	0.924	0.924
h_6	1.582	2.290	2.359	2.343	2.343
h_7	2.029	1.990	1.964	1.951	1.950
h_8	7.443	4.847	4.515	4.460	4.457
h_9	4.340	2.693	2.652	2.617	2.617
Total times (t/s)	24.052	19.740	19.133	18.915	18.907

Figure 1 shows the joint displacements, velocities, accelerations, and jerks for this example. From this figure, one can find that some accelerations and jerks are very close to their constrained values, which means that the velocities can be raised in a shorter time so that the total traveling time of the robot is shortened.



p— position, *v*—velocity, *a*—acceleration, *j*—jerk

Fig. 1 Optimum joint trajectories for this example

Fig.2 shows the graphic simulation result for this example, which illustrates the specified hand path.

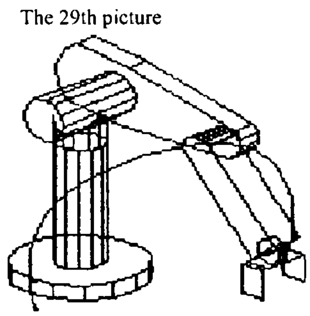


Fig. 2 Graphic simulation result for this example

6 Conclusions

In this paper, a new kind of joint trajectory for industrial robots is proposed. The optimization algorithm is established, which can be used for controlling a robot to track a specified path on line with minimum traveling time. From the results of computer simulation and robot experiment, it is concluded that this new method is very effective to raise a robot's working efficiency.

References:

[1] BOBROW J E, DUBOWSKY S, GIBSON J S. Time-optimal control of robotic manipulators along specified paths [J]. *Int J of Robotics Research*, 1985,4(3):3 - 17.

[2] TONDU B, BAZAZ S A. The three-cubic method: an optimal online robot joint trajectory generator under velocity, acceleration, and wandering constraints [J]. *Int J of Robotics Research*, 1999, 18(9): 893 - 901.

[3] WEINREB A, BRYSON A E. Optimal control of systems with hard control bounds [J]. *IEEE Trans on Automatic Control*, 1985, 30 (11): 1135 - 1138.

[4] GALICKI M. The planning of robotic optimal motions in the presence of obstacles [J]. *Int J of Robotics Research*, 1998,17(3):248 - 259.

[5] BOBROW J E. Optimal robot path planning using the minimum time criterion [J]. *IEEE Trans on Robotics and Automation*, 1988,4(4): 443 - 450.

[6] LIN C S, CHANG P R, LUH J Y S. Formulation and optimization of cubic polynomial joint trajectories for industrial robots [J]. *IEEE Trans on Automatic Control*, 1983,28(12):1066 - 1073.

[7] SIMON D. The application of neural networks to optimal robot trajectory planning [J]. *Robotics and Autonomous Systems*, 1993, 11 (1):23 - 34.

[8] HIMMELBLAU D M. *Applied Nonlinear Programming* [M]. New York: McGraw-Hill, 1972.

作者简介:

谭冠政 (1962 —),男,工学博士,现为中南大学机器人研究所所长,信息科学与工程学院教授,模式识别与智能系统学科博士生导师,主要研究方向:人工智能及其应用,仿生机器人与智能仿生系统,先进控制理论与先进算法,E-mail:tgz@mail.csu.edu.cn;

王越超 (1961 —),男,工学博士,现为中国科学院沈阳自动化研究所副所长,机器人学开放研究实验室主任,教授,博士生导师,国家"863 计划"先进制造与自动化技术主题专家组专家,主要研究领域:智能机器人系统及控制,先进控制理论与先进算法.