Mathematics Mechanization Research Preprints
KLMM, Chinese Academy of Sciences
Vol. 31, 21–42, March, 2012

Time-Optimal Interpolation for Five-axis CNC Machining along Parametric Tool Path based on Linear Programming[1)]

Wei Fan, Xiao-Shan Gao, Ke Zhang
KLMM, Academy of Mathematics and Systems Science
Chinese Academy of Sciences

Abstract. In this paper, the time-optimal velocity planning problem for five axis CNC machining along a given parametric tool path under chord error, acceleration, and jerk constraints is studied. The velocity planning problem under confined chord error and acceleration is reduced to an equivalent linear programming problem by discretizing the tool path and other quantities. As a consequence, a polynomial-time algorithm with computational complexity $O(N^{3.5})$ is given to find the optimal solution, where N is the number of discretized segments of the tool path. The velocity planning problem under confined confined chord error, acceleration, and jerk is reduced to a linear programming program by using a linear function to approximate the nonlinear jerk constraint. As a consequence, a polynomial-time algorithm is given to find the approximate optimal solution. Simulation results are used to show the efficiency and effectiveness of the algorithms.

Keywords. Time-optimal interpolation, velocity planning, parametric tool path, jerk constraint, linear programming algorithm, polynomial-time algorithm.

1. Introduction

Interpolation algorithms, which determine how the machine tool moves along the tool path, play a key role in high speed and high precisions CNC machining and hence are widely studied in the literature. An interpolation algorithm in the CNC controller usually consists of two phases: velocity planning and parameter computation. Let $\eta(u), u \in [0,1]$ be the tool path. The phase to determine the feedrate $v(u)$ along $\eta(u)$ is called velocity planning. When the feedrate $v(u)$ is known, the phase to compute the next interpolation point at $u_{i+1} = u_i + \triangle u$ during one sampling period is called parameter computation. This paper focuses on velocity planning along a given parametric tool path for five-axis CNC machines.

In order to achieve high speed machining, it is required that the planned feedrate to be as large as possible. In order to achieve high quality machining, it is required that the planned feedrate satisfies constraints such as confined acceleration, confined jerk, and confined chord error. Therefore, velocity planning is usually formulated as a time-minimum optimization problem under kinematic and chord error constraints.

[1)] Partially supported by a National Key Basic Research Project of China (2011CB302400) and by a grant from NSFC (60821002).

The velocity planning algorithms for parametric tool pathes can be roughly divided into four classes: the phase space analysis methods [1, 2, 3, 4, 5, 6], the direct sampling methods [7, 8, 9, 10, 11, 12, 13], the critical point methods [14, 15, 16, 17], and the numerical optimization methods [18, 19, 20, 21].

With the phase analysis methods, closed form optimal solutions are obtained with the concepts of velocity limiting curve and integration trajectory in the case of confined acceleration by Bobrow et al [1], Shin and McKay [2], Farouki et al[3], Zhang et al [4], and Yuan et al [5]. In the case of confined jerk, Zhang et al gave a greedy algorithm based on the concept of velocity limiting surface [6]. The main drawback of this approach is that the computations needed are time consuming if the tool path $\eta(u)$ is complicated, and the method works practically only for simple tool pates such as those described by quadratic and cubic PH curves [4, 5].

In the direct sampling methods, the interpolation points at every sampling time $kT, k = 0, 1, \ldots$ are computed based certain strategies. For instance, Bedi et al [7] and Yang-Kong [8] used a constant feedrate, and Yeh and Hsu used a chord error bound to control the feedrate if needed and used a constant feedarte in other places [10]. These methods are simple and efficient. However, the machine acceleration capabilities were not considered. Emami, Arezoo [9] and Lai et al[11] proposed velocity planning methods with confined acceleration, jerk, and confined chord error by adjusting the velocity if any of the bounds is violated through backtracking. The backtracking procedure makes it difficult to estimate the control complexities. In [12], the idea of phase space analysis is adopted to give a direct sampling method under confined acceleration, whose computational complexity is $O(M)$ where M is the number of discretization grids. In [13], Tikhon et al proposes a NURBS interpolator based on the adaptive feedrate control for the constant material removal rate and nevertheless the velocity profile adopted ignores the constraints of machine performance.

In the critical point methods, critical points of the tool path with extremal curvatures are identified and maximum feedrates at the critical points are determined according to chord error or acceleration constraints. Furthermore, feedrate function for each tool path segment between two critical points are planned with various velocity profiles such as S-shape profile by Narayanaswami and Yong [14], trigonometric profile by Lee et al [16], and jounce confined profile by Fan et al [15]. Tsai et al [17] further introduced dynamics constraints. This approach is very practical, but it is not time-optimal and the chord error is not guarantied.

In the numerical optimization methods, the velocity planning problem is discretized as a nonlinear optimization problem which is solved with standard numerical methods. This is a quite standard method to solve continuous optimization problems. Nonlinear optimization based interpolation methods under confined jerk and chord error were given by Erkorkmaz-Altintas [21], Sencer-Altintas-Croft [19]. In [18], Gasparetto et al proposed to use a linear combination of the machining time and the total jerk as the objective function in order to minimize vibration. Numerical optimization methods are very general and powerful, but solving nonlinear programming problems is generally time-consuming and the obtained solutions are not guarantied to be global optimal.

In this paper, the time-minimum velocity planning problem for five axis CNC machining along a given parametric tool path $\eta(u), u \in [0, 1]$ under maximal feedrate, chord error, acceleration, and jerk constraints is studied. The numerical optimization approach is adopted, but instead of nonlinear optimization method, the linear programming method is used to solve the velocity planning problem. As a consequence, we give time-minimum velocity planning algorithms whose computational complexity is polynomial of the form $O(N^{3.5})$ where N is the number of discretized segments of the tool path. Furthermore, the computational complexity is the same for any tool path, making the method an efficient one for the quite general velocity planning problem mentioned above.

By discretizing the differential quantities as finite differences, the velocity planning problem under confined acceleration is reduced to an equivalent linear programming problem. It is proved that the linear programming problem has a unique solution which is the solution to the original velocity planning problem. As a consequence, a polynomial time algorithm with complexity $O(N^{3.5})$ is given to find the optimal solution, where N is the number of discretized segments of the tool path.

In the case of confined jerk, the nonlinear jerk constraint is approximated with a stronger linear linear function, and the velocity planning problem under confined jerk is also reduced to a linear programming program problem. Simulation results show that the solution to the linear programming problem gives nice approximation to the original problem.

Simulation results for tool pathes from real CNC models are used to show the effectiveness of the algorithms. Complexity analysis is given to show the efficiency of the algorithm.

The rest of this paper is organized as follows. In Section 2, the kinematic constraints and chord error constraints of CNC machining are presented. In Section 3, velocity planning under confined acceleration and chord error is studied. In Section 4, velocity planning under confined jerk, acceleration, and chord error is studied. In Section 5, the time complexity of computation is analyzed. In Section 6, the conclusion is presented.

2. Kinematic and chord error constraints

In this section, the kinematics and chord error constraints of CNC machining will be presented, including maximal feedrate, maximal accelerations of the five axes, and maximal chord error. The jerk constraint will be given in Section 3.. At the start and end points of the tool path, the feedrates are assumed to be zero.

2.1. The feedrate and acceleration constraints

We assume that the tool path of the five-axis CNC machine is given by a set of parametric functions with at least C^1 continuity $\eta(u) = (x(u),\ y(u),\ z(u),\ a(u),\ c(u))^T, (u \in [0,1])$, where $r(u) = (x(u), y(u), z(u))$ is the machining path, $a(u)$ and $c(u)$ are rotary angles around the X-axis and Z-axis respectively. The parametric functions could be NURBS, B-spline curves, etc. Note that the coordinates of $\eta(u)$ and $r(u)$ are in the machine coordinate system (MCS). In this paper, we use "·" to denote "$\frac{d}{dt}$" and use " $'$ " to denote " $\frac{d}{du}$".

Denote the machining velocity to be $\mathbf{v} = (v_x, v_y, v_z)$ and the tangential feedrate to be $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$. Let s be the arc length of $r(u)$ and $\sigma(u) = ds/du = \sqrt{x'(u)^2 + y'(u)^2 + z'(u)^2}$ the parametric speed. Introduce the following important quantity

$$q = \left(\frac{v}{\sigma}\right)^2, \tag{1}$$

which will be used as the optimization variable in our velocity planning problem. As will be shown later, the acceleration constraints can be written as a linear inequality about q and its derivative which is one of the reasons allowing us to reduce the velocity planning to a linear programming problem. Then

$$\dot{u} = \frac{du}{dt} = \frac{du}{ds}\frac{ds}{dt} = \frac{v}{\sigma} = \sqrt{q},$$

For the x-axis,

$$v_x = \dot{x} = \frac{dx}{du}\frac{du}{dt} = x'\sqrt{q} \tag{2}$$

$$a_x = \dot{v_x} = \frac{dv_x}{dx}\frac{dx}{dt} = v_x\frac{dv_x}{dx} = \frac{1}{2}\frac{d{v_x}^2}{dx} = \frac{1}{2}\frac{dqx'^2}{dx} \tag{3}$$

The maximal feedrate along the tool path is an important CNC parameter reflecting the machining ability. During the machining process, the feedrate must obey the constraint of maximal feedrate. If the maximal feedrate V_m is given, the feedrate v must satisfy the following condition

$$v(u) \leq V_m, u \in [0,1]. \tag{4}$$

From (2), $v = \sigma\sqrt{q}$. Then, the inequality (4) can be written as an inequality about q

$$q(u) \leq \frac{V_m^2}{\sigma^2(u)}, u \in [0,1]. \tag{5}$$

For CNC machines, the maximal acceleration limits of each axis, reflecting the ability to accelerate, are important parameters. The following formulas are the acceleration constraints of the five axes:

$$|a_\tau(u)| \leq A_\tau, \tau \in \{x,y,z,a,c\}, u \in [0,1]. \tag{6}$$

where A_τ is the maximal acceleration of τ-axis. From (3), the constraints (6) are equivalent to the following inequalities about q

$$|\frac{1}{2}\frac{d(q\tau'^2)}{d\tau}| \leq A_\tau, \tau \in \{x,y,z,a,c\}, u \in [0,1]. \tag{7}$$

2.2. Chord error constraint for five-axis CNC machines

Chord error is the error caused by machining the tool path within one sampling period. Only the start point and end point for one sampling period are given to the CNC machine, and the actual machining path is different from the tool path. Chord error is used to measure the difference between these two pathes and is defined to be the distance between the line segment connecting the start point and end point and the tool-path as shown in Figure 1.

Different from the three-axis CNC machines, for five-axis CNC machines, the chord error is measured in the workpiece coordinate system (WCS) instead of the machine coordinate system(MCS) [20]. Therefore, the coordinate transformation between WCS and MCS is needed. Such a transformation depends on the structure of the CNC machines and the widely used table-tilting (Figure 2) five-axis CNC machine is adopted in this paper. Other types of five-axis CNC machines can be treated similarly.

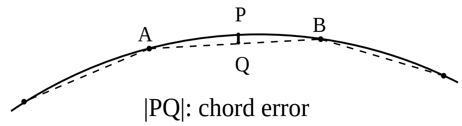


Fig. 1. The schematic diagram of chord error

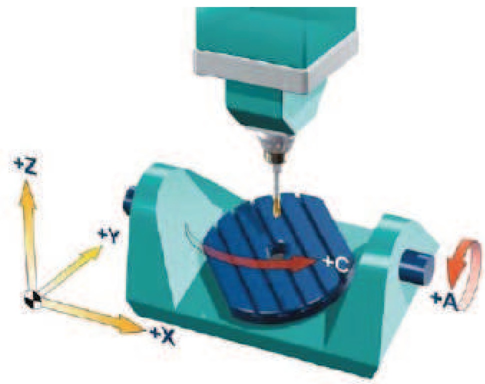


Fig. 2. Typical table-tilting 5-axis machine

In a table-titling machine shown in Figure 2, C is the workbench rotary angle around the Z-axis and A is the rotary angle around the X-axis. Denote $r_w(u) = (x_w(u), y_w(u), z_w(u))$ as the coordinates of tool path curve $r(u)$ in WCS. The initial coordinate of origin of MCS is denoted as $O_w = (x_{w0}, y_{w0}, z_{w0})$.

Let $Rot(a, x)$ and $Rot(c, z)$ be the rotary matrices of rotating angle a around X-axis and angle c around Z-axis respectively, which have following forms

$$Rot(a,x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos a & -\sin a & 0 \\ 0 & \sin a & \cos a & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Rot(c,z) = \begin{pmatrix} \cos c & -\sin c & 0 & 0 \\ \sin c & \cos c & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Denote the translation matrix to be $Tran(O_w)$ which has the form

$$Tran(O_w) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_{w0} & y_{w0} & z_{w0} & 1 \end{pmatrix}.$$

The relationship between the coordinate (x_w, y_w, z_w) in WCS and the coordinate (x, y, z) in MCS is given below.

$$(x_w \;\; y_w \;\; z_w \;\; 1) = (x \;\; y \;\; z \;\; 1) \cdot Tran(O_w) \cdot Rot(a,x) \cdot Rot(c,z) \tag{8}$$

which can be written as

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \begin{pmatrix} \cos c & \cos a \sin c & \sin a \sin c \\ -\sin c & \cos a \cos c & \sin a \cos c \\ 0 & -\sin a & \cos a \end{pmatrix} \cdot \begin{pmatrix} x + x_{w0} \\ y + y_{w0} \\ z + z_{w0} \end{pmatrix} \tag{9}$$

By (9), the coordinate transformation from MCS to WCS is obtained.

Let $\mu = (x_w, y_w, z_w)^T$ and $\eta = (x, y, z, a, c)^T$. Differentiate two sides of the equation (9) and obtain the following differential relationship

$$\frac{d\mu}{du} = M_t \cdot \frac{d\eta}{du} \tag{10}$$

where

$$M_t = \begin{pmatrix} \cos c & \cos a \sin c & \sin a \sin c & z_w \sin c & y_w \\ -\sin c & \cos a \cos c & \sin a \cos c & z_w \cos c & -x_w \\ 0 & -\sin a & \cos a & -(y + y_{w0}) \cos a - (z + z_{w0}) \sin a & 0 \end{pmatrix}.$$

Differentiate two sides of the equation (10) and obtain the second-order derivative of vector μ as follow

$$\frac{d^2\mu}{du^2} = \frac{dM_t}{du} \cdot \frac{d\eta}{du} + M_t \cdot \frac{d^2\eta}{du^2} \tag{11}$$

where

$$\frac{dM_t}{du} = \begin{pmatrix} -c' \sin c & -c' \cos c & 0 \\ -a' \sin a \sin c + c' \cos a \cos c & -a' \sin a \cos c - c' \cos a \sin c & -a' \cos a \\ a' \cos a \sin c + c' \sin a \cos c & a' \cos a \cos c - c' \sin a \sin c & -a' \sin a \\ c' z_w \cos c + z'_w \sin c & -c' z_w \sin c + z'_w \cos c & -y' \cos a - z' \sin a - z_w a' \\ y'_w & -x'_w & 0 \end{pmatrix}^T.$$

and x'_w, y'_w, z'_w can be determined by (10).

Now, we can give the chord error constraint. Since one sampling period is very small, the tool path during one sampling period can be approximated as a piece of circle arc. If the sampling period is T and and tool path is $r_w(u) = (x_w(u), y_w(u), z_w(u))$ in WCS, then the chord error $\delta(u)$ has the following relationship with velocity $v_w(u)$ in WCS [5],

$$\delta(u) = \rho_w(u) - \sqrt{\rho_w(u)^2 - \frac{v_w(u)^2 T^2}{4}}$$

where $\rho_w(u)$ is the curvature radius of tool path under WCS and $\rho_w(u) = \frac{|r'_w(u)|^3}{|r'_w(u) \times r''_w(u)|}$. Let δ_m be the maximal chord error. So

$$\rho_w(u) - \sqrt{\rho_w(u)^2 - \frac{v_w(u)^2 T^2}{4}} \leq \delta_m.$$

That is

$$v_w(u) \leq \frac{\sqrt{8\rho_w(u)\delta_m - 4\delta_m^2}}{T}.$$

Since the δ_m is a tiny quantity compared with $\rho_w(u)$, under WCS the limit of velocity $v_w(u)$ at u can be approximately written to be

$$v_w(u) \leq \frac{\sqrt{8\delta_m \rho_w(u)}}{T} \tag{12}$$

Since $\frac{v_w}{\sigma_w} = \dot{u} = \sqrt{q}$, (12) is reduced to

$$q \leq \frac{8\delta_m \rho_w(u)}{\sigma_w^2 T^2} = \frac{8\delta_m \sigma_w}{|r_w'(u) \times r_w''(u)|T^2} \tag{13}$$

From (10) and (11), constraint (13) can be written as an inequality about q

$$q \leq \frac{8\delta_m |M_t \eta'|}{|M_t \eta' \times (M_t' \eta' + M_t \eta'')|T^2}. \tag{14}$$

3. Optimization velocity planning under acceleration constraint

In this section, we will present a polynomial time algorithm to find the time optimal solution to the velocity planning problem under acceleration and chord error constraints.

3.1. Formulation of the problem

In this section, the velocity planning problem is formulated as an optimal problem under differential constraints.

The machining time is $t = \int_0^1 \frac{du}{\dot{u}} = \int_0^1 \frac{du}{\sqrt{q}}$. So the time optimal velocity planning problem under the confined feedrate, acceleration, and chord error can be described as the following form

$$\min_q \int_0^1 \frac{du}{\sqrt{q}} \tag{15}$$

subject to constraints (5), (7), and (14).

The above optimal problem has a very important property given in the following theorem, which was proved in many cases and using different methods in [1, 2, 3, 5, 6, 20].

Theorem 3.1 *The optimal feedrate $v_o(u), u \in [0,1]$ of problem (15) is maximum for any parameter u. That is, let $v_f(u)$ be another feedrate satisfying constraints (5), (7), and (14). Then, $v_f(u) \leq v_o(u)$ for all $u \in [0,1]$. Furthermore, the optimal feedrate is bang-bang-singular in the sense that for any parameter u, the equality sign holds at least in one of the three constraints constraints (5), (7), and (14).*

As a consequence of Theorem 3.1, it is easy to see that the optimal solution to problem (15) is unique.

3.2. The discrete form of optimization problem

The three-axis version for the above optimization problem can be solved analytically as shown in [1, 2, 3, 5]. Although the analytical solution approach can be theoretically extended to the five-axis case, it is difficult to obtain a practically effective method due to the complicated expression introduced in constraint (14). Therefore, to develop efficient numerical method is inevitable for the five-axis CNC velocity planning with chord error constraint. In order to use numerical method to solve problem (15), in this section, we give a discrete version of this problem.

We divide the parametric interval $[0,1]$ into N equal parts with endpoints are $u_i = \frac{i}{N}$, $i = 0, \ldots, N$. The length of each sub-interval is $\Delta = 1/N$. Since the velocities at the two endpoints of tool path are zero, we have $q_0 = q_N = 0$. Since Δ is very small, the constraints (5), (7), and (14) can be approximately transformed into the following linear inequalities.

$$0 \le q_i \le \frac{V_m^2}{\sigma_i^2} \tag{16}$$

$$|q_{i+1}\tau_{i+1}'^2 - q_i\tau_i'^2| \le 2A_\tau|\tau_{i+1} - \tau_i|, \tau \in \{x, y, z, a, c\} \tag{17}$$

$$0 \le q_i \le \frac{8\delta_m|M_{ti}\eta_i'|}{|M_{ti}\eta_i' \times (M_{ti}'\eta_i' + M_{ti}\eta_i'')|T^2}. \tag{18}$$

where $q_i = q(u_i)$, $\sigma_i = \sigma(u_i)$, $\tau_i = \tau(u_i)$, $\tau_i' = \tau'(u_i)$, and the vectors η_i', M_{t_i} and M_{t_i}' are the values of $\eta'(u)$, $M_t'(u)$ and $M_t(u)$ at u_i respectively. Correspondingly, the optimization problem (15) is transformed into the following *nonlinear programming problem* with objective function

$$\min_{\{q_i\}} \frac{1}{N} \sum_{i=1}^{N-1} \frac{1}{\sqrt{q_i}} \tag{19}$$

and constraints (16), (17), and (18).

Instead of solving problem 19, we will solve the following *linear programming problem* with objective function

$$\max_{\{q_i\}} \sum_{i=1}^{N-1} q_i \tag{20}$$

and constraints (16), (17), and (18).

In the following, we will show that the solution to the linear programming problem (20) provides an approximate solution to the original problem (15). The key idea is to show that the optimal solution $q_i^*, (i = 1, \ldots, N-1)$ to the linear programming problem (20) is unique and each q_i^* achieves the maximal value among all possible feasible solutions. Combing this property and Theorem 3.1, the claim can be proved.

Theorem 3.2 *The unique solution to problem* (15) *can be sufficiently approximated with the unique solution of the linear programming problem* (20) *when* N *is large enough.*

Proof: The feasible region for $q_i = 0, (i = 1, \ldots, N-1)$ determined by constraints (16), (17), and (18) is clearly a finite compact set. Then, the linear programming program always has a solution $Q^* = \{q_i^*, (i = 1, \ldots, N-1)\}$.

We claim that for each i, q_i^* is maximal among all feasible solutions. Assume the contrary. Then there exists another feasible solution $\{q_k, (k = 1, \ldots, N-1)\}$ such that $q_j > q_j^*$ holds for at least one j. Since for a linear programming problem, the optimal value for the objective function is unique, there must exist a k such that $q_k < q_k^*$. Notice that $q_0 = q_0^* = q_N = q_N^* =$

0. Then, there exists $i < j$ such that $q_i^* \geq q_i$ and $q_k^* < q_k, (k = i+1, \ldots, j-1)$, and $q_j^* \geq q_j$. Since q_i, q_{i+1} and q_i^*, q_{i+1}^* satisfy the constraint (17), we have

$$|q_{i+1}\tau_{i+1}'^2 - q_i\tau_i'^2| \leq 2A_\tau|\tau_{i+1} - \tau_i|,$$
$$|q_{i+1}^*\tau_{i+1}'^2 - q_i^*\tau_i'^2| \leq 2A_\tau|\tau_{i+1} - \tau_i|.$$

Let $a_1 = q_{i+1}\tau_{i+1}'^2$, $a_2 = q_i\tau_i'^2$, $b_1 = q_{i+1}^*\tau_{i+1}'^2$, $b_2 = q_i^*\tau_i'^2$, $B = 2A_\tau|\tau_{i+1} - \tau_i|$. So $|a_1 - a_2| \leq B$, $|b_1 - b_2| \leq B$. By the assumption $q_i^* \geq q_i$ and $q_{i+1}^* < q_{i+1}$, we have $a_1 > b_1$ and $a_2 \leq b_2$. Then

$$a_1 - b_2 = a_1 - b_1 + b_1 - b_2 \geq b_1 - b_2 \geq -B,$$
$$a_1 - b_2 = a_1 - a_2 + a_2 - b_2 < a_1 - a_2 \leq B.$$

Thus $|a_1 - b_2| \leq B$. That is

$$|q_{i+1}\tau_{i+1}'^2 - q_i^*\tau_i'^2| \leq 2A_\tau|\tau_{i+1} - \tau_i|.$$

That is, q_i^* and q_{i+1} satisfy the constraint (17). Similarly, it can be shown that q_{j-1} and q_j^* also satisfy the constraint (17). Since constrains (16) and (18) are automatically satisfied, by replacing $q_k^*, (k = i+1, \ldots, j-1)$ with $q_k, (k = i+1, \ldots, j-1)$, we obtain a new feasible solution of the linear programming problem, which has a larger value for the objective function, leading a contradiction. Thus the claim is proved.

From the claim, the linear programming problem (20) has a unique solution $q_i^*, (i = 0, \ldots, N)$. Furthermore, since the solution q_i^* is maximal for each i among all feasible solutions, a solution to problem (20) is also a solution to problem (19). And the objective function (19) is the discrete form of (15). Therefore, the solution to linear problem (20) can approximate the solution to problem (15) as good as possible. ▮

3.3. Optimization velocity planning based on linear programming

In the preceding section, we show that the problem of velocity planning is transformed into a linear optimization problem. In this section, the algorithm is given.

In order to use the standard solver from MatLab, we rewrite the linear programming problem (20) has the following standard form.

$$\max_Q cQ \tag{21}$$

with constraints $MQ \leq R$ and $Q \geq 0$, where $Q = (q_1, ..., q_{N-1})^T$, $q_0 = q_N = 0$, the coefficient vector c is $(1, ..., 1)$ with size $1 \times (N-1)$, R and M will be given below.

Denote $h(u) = \min\{\frac{V_m^2}{\sigma(u)^2}, \frac{8\delta_m|M_t\eta'(u)|}{|M_t\eta'(u)\times(M_t{}'\eta'(u)+M_t\eta''(u))|T^2}\}$ and $\lambda_i = (x_i'^2, y_i'^2, z_i'^2, a_i'^2, c_i'^2)^T$, where $x_i' = x'(u_i)$. Let $\pi_i = (2A_x|x_{i+1} - x_i|, 2A_y|y_{i+1} - y_i|, 2A_z|z_{i+1} - z_i|, 2A_a|a_{i+1} - a_i|, 2A_c|c_{i+1} - c_i|)$. Then $R = (\pi_0, \pi_0, h(u_0), ..., \pi_{N-2}, \pi_{N-2}, h(u_{N-2}), \pi_{N-1}, \pi_{N-1})^T$ whose size is $(11N - 1) \times 1$. The coefficient matrix M has the following form whose size is

$(11N-1)\times(N-1)$.

$$M=\begin{pmatrix} \lambda_1 & 0 & 0 & ... & 0 & 0 \\ -\lambda_1 & 0 & 0 & ... & 0 & 0 \\ 1 & 0 & 0 & ... & 0 & 0 \\ -\lambda_1 & \lambda_2 & 0 & ... & 0 & 0 \\ \lambda_1 & -\lambda_2 & 0 & ... & 0 & 0 \\ 0 & 1 & 0 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & -\lambda_{N-2} & \lambda_{N-1} \\ 0 & 0 & 0 & ... & \lambda_{N-2} & -\lambda_{N-1} \\ 0 & 0 & 0 & ... & 0 & 1 \\ 0 & 0 & 0 & ... & 0 & -\lambda_{N-1} \\ 0 & 0 & 0 & ... & 0 & \lambda_{N-1} \end{pmatrix} \quad (22)$$

After solving the linear programming problem (21), we obtain a set of optimal solution $Q^*=(q_1^*,\ldots,q_{N-1}^*)^T$ with $q_0^*=q_N^*=0$. From equation (1), the optimal feedrate can be obtained $v_i^*=v(u_i)=\sigma(u_i)\sqrt{q_i^*},(i=0,\ldots,N)$. We can uses standard techniques to fit the discrete values v_i to obtain the optimal feedrate function $v^*(u),u\in[0,1]$.

Based on the above analysis, the following velocity planning algorithm is given.

Algorithm 3.3 VPA_LP
Input: *Five-axis tool path* $\eta(u)$,$u\in[0,1]$ *in MCS, the sampling period* T*, the maximal feedrate* V_m *, five-axis maximal acceleration bounds* (A_x,A_y,A_z,A_a,A_c)*, the chord error bound* δ_m*, the number of discretion number* N*, and the initial position of the origin of MCS* (x_{w0},y_{w0},z_{w0}).
Output: *The approximate optimal velocity function* $v(u),u\in[0,1]$ *of problem 15.*

1. *Compute* $\eta_i=\eta(u_i)$, λ_i, *and* π_i *defined in this section, where* $u_i=\frac{i}{N}$, $i=0,\ldots,N$ *to obtain the coefficient matrix* M *in* (22).
2. *Compute the error limit function*

$$h(u_i)=\min\{\frac{V_m^2}{\sigma_i^2},\frac{8\delta_m|M_t(u_i)\eta'(u_i)|}{|M_t(u_i)\eta'(u_i)\times(M_t'(u_i)\eta'(u_i)+M_t(u_i)\eta''(u_i))|T^2}\},i=1...N-1.$$

Obtain the right-side vector

$$R=(\pi_0,\pi_0,h(u_1),...,\pi_{N-2},\pi_{N-2},h(u_{N-1}),\pi_{N-1},\pi_{N-1})^T$$

3. *Solve the linear programming problem* (21) *to obtain the optimal solution* $Q^*=(q_1^*,...,q_{N-1}^*)^T$. *Let* $q_0^*=q_N^*=0$.
4. *Compute the velocity function. For instance, in the simplest case, we can give a piecewise linear representation as follows*

$$v(u)=v_i^*\frac{u-u_{i+1}}{u_i-u_{i+1}}+v_{i+1}^*\frac{u_i-u}{u_i-u_{i+1}}$$

where $v_i^*=\sigma(u_i)\sqrt{q_i^*}$ *and* $u\in[u_i,u_{i+1}],i=0,\ldots,N-1$.

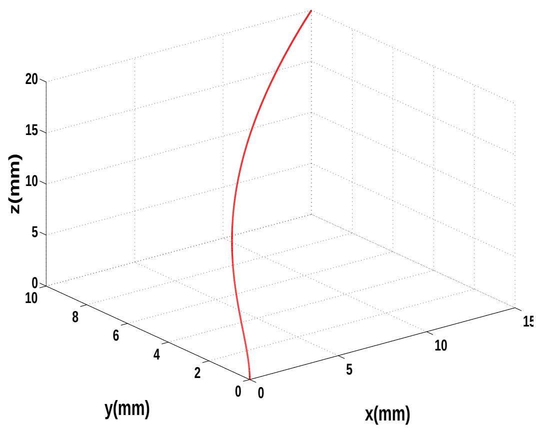


Fig. 3. The tool path curve

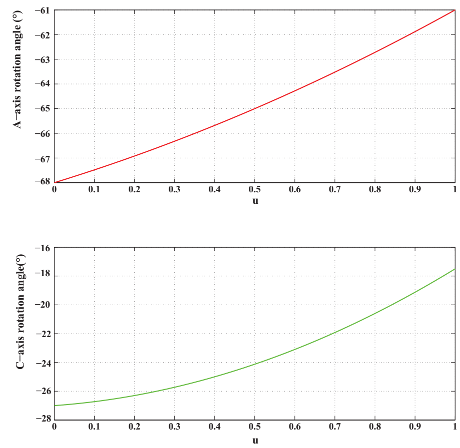


Fig. 4. The A-axis and C-axis curve

A nice feature to Algorithm 3.3 is that we can give its worst case computational complexity, which is rare for most velocity planning algorithms. Since the dominate step of Algorithm 3.3 is step 3, we need only to estimate the complexity of this step. Based on Karmarkar's famous algorithm [22] to solve linear programming problems, if using floating point number computations, the algorithm requires $O(N)$ steps and each step requires $O(N^{2.5})$ arithmetic operations. Thus, we have

Theorem 3.4 *For a given N, the worst case computational complexity for Algorithm 3.3 is $O(N^{3.5})$ in terms of floating point arithmetic operations.*

In Algorithm 3.3, we need only to compute the values of the tool path $\eta(u)$ at the parameter values $u_k, k = 0, \ldots, N$. The computation complexity for this procedure is $O(N)$. As a consequence, the computational complexity of the algorithm is the same for any tool path $\eta(u)$. This is a nice feature comparing to the phase analysis methods [1, 2, 3, 4, 5, 6] which involves integrations and nonlinear algebraic equation solving and thus only practically work for simple tool pathes [4, 5].

3.4. An illustrative example

An illustrative example is given, where the tool path is a set of cubic polynomial curve:

$$\eta(u) = (15u^3, 10u^2, 20u, 2u^2 + 5u - 68, 7.5u^2 + 2u - 27), u \in [0, 1].$$

The path curve and the curve of two rotary axes are illustrated respectively in Figure 3 and Figure 4.

The CNC parameters are given as follow: $A_x = 1000mm/s^2$, $A_y = 1000mm/s^2$, $A_z = 1000mm/s^2$, $A_a = 500°/s^2$, $A_c = 500°/s^2$, $V_m = 110mm/s$, $\delta_m = 0.05\mu m$, $T = 1ms$, $O_w = (1, 1, 1)$, $N = 200$.

The velocity curve obtained with Algorithm **VPA_LP** and the theoretical chord error of the velocity curve are shown in Figures 5 and 6 respectively, where theoretical chord error is calculated by

$$\delta(u) = \frac{q(u)|M_t\eta' \times (M_t'\eta' + M_t\eta'')|T^2}{8|M_t\eta'|}. \tag{23}$$

The acceleration curves of X, Y and Z axes for this velocity curve are shown in Figure 7(a). The acceleration curves of two rotary axes are illustrated in Figure 7(b).

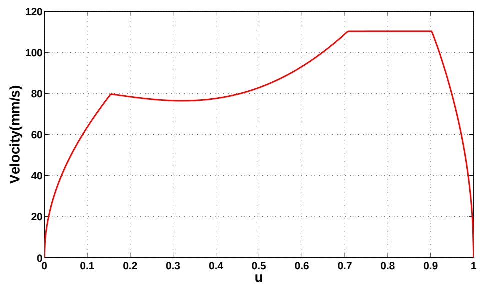


Fig. 5. The planned velocity

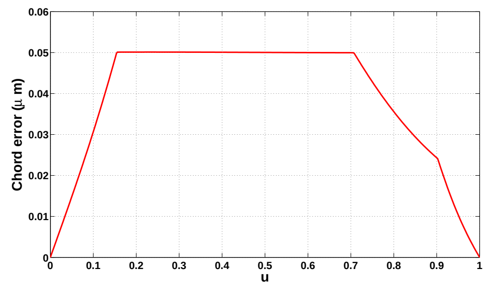
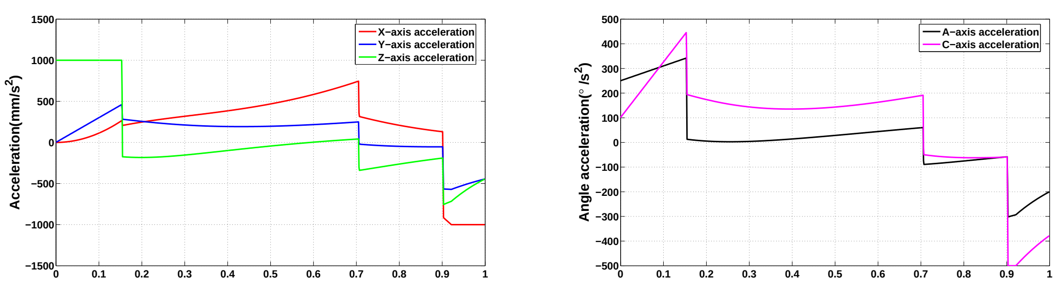


Fig. 6. The chord error

From Figures 5, 6, and 7, it is easy to see that the planned velocity function is Bang-bang-singular control, meaning that at least one of five-axis accelerations, the feedrate, and the chord error reaches its boundary value at any time. The whole machining process is divided into five phases. When $u \in [0, 0.155]$, the acceleration of Z-axis reaches maximum value $1000mm/s^2$. When $u \in [0.155, 0.71]$, the chord error reaches maximum value. When $u \in [0.71, 0.905]$, the feedrate reaches maximal velocity V_m. When $u \in [0.905, 0.925]$, the acceleration of C-axis reaches minimum value $-500°/s^2$. Finally, when $u \in [0.925, 1]$, the acceleration of X-axis reaches minimum value $-1000mm/s^2$.

From Theorem 3.1, the Bang-bang-singular control is a necessary condition of this optimal problem. The fact that our solutions are Bang-bang-singular implies that these solutions provide nice approximation to the optimal solution of the original problem.

However from Figure 5, it is not difficult to find that the feedrate function is not smooth at some points such as $u = 0.155$, $u = 0.71$ and $u = 0.905$. At these points the large vibration of the machine tool could lead to poor machining quality. This problem is addressed in the next section by introducing jerk bounds.



(a) Accelerations of X,Y and Z-axis (b) Accelerations of A-axis and C-axis

Fig. 7. Accelerations for five axes

4. Velocity planning under jerk constraint with linear programming

In this section, we will consider time optimal velocity planning under jerk constraints and give an approximate algorithm based on linear programming.

4.1. Formulation of the problem

Let the tool path $\eta(u) = (x(u),\ y(u),\ z(u),\ a(u),\ c(u))^T, (u \in [0,1])$ have C^2 continuity. The maximal jerks of the five axes $\mathbf{J}_m = (J_x, J_y, J_z, J_a, J_c)$ are used to describe the maximal change rate of accelerations that five axes can sustain. So the jerk of every axis in the machining process should satisfy the following constraints

$$|j_\tau(u)| \le J_\tau, u \in [0,1], \tau \in \{x, y, z, a, c\}. \tag{24}$$

Use the symbols in Section 2.. Since the acceleration $\mathbf{a} = (a_x, a_y, a_z, a_a, a_c)$ can be expressed in the following form

$$a_\tau = \ddot{\tau} = \frac{d(\sqrt{q}\tau')}{dt} = \frac{d(\sqrt{q}\tau')}{du}\sqrt{q} = \tau'' q + \frac{\tau'}{2} q',$$

the jerk $\mathbf{j} = (j_x, j_y, j_z, j_a, j_c)$ can be written as

$$j_\tau = \dot{a}_\tau = a'_\tau \sqrt{q} = (\tau''' q + (\tau' + \frac{\tau''}{2}) q' + \frac{\tau'}{2} q'')\sqrt{q}, \tag{25}$$

where $\tau \in \{x, y, z, a, c\}$. Note that (25) in nonlinear about q, q', and q''.

The time optimal velocity planning problem becomes

$$\min_q \int_0^1 \frac{du}{\sqrt{q}} \tag{26}$$

subject to constraints (5), (7), (14), and (25).

Note that after adding jerk constraints, Theorem 3.1 is not fully proved yet. We do not know whether an optimal solution to problem (26) is maximum at any parameter value. But, we still know that an optimal solution must be bang-bang-singular [6, 20].

4.2. Discrete form of the velocity planning problem

Similar to Section 3., the interval $[0,1]$ is divided into N equal intervals at $u_i = i/N, (i = 0, \dots, N)$. The first order and second order derivatives of q can be approximated as

$$q'_i \approx \frac{q_{i+1} - q_{i-1}}{2\Delta u},$$
$$q''_i \approx \frac{q_{i+1} + q_{i-1} - 2q_i}{\Delta u^2}, i = 1, \dots, N-1.$$

As a consequence, constraint (25) is discretized to the following form

$$|(\alpha_{\tau_i} q_{i-1} + \beta_{\tau_i} q_i + \gamma_{\tau_i} q_{i+1})\sqrt{q_i}| \le J_\tau, i = 1, \dots, N-1 \tag{27}$$

where

$$\alpha_{\tau_i} = \frac{\tau'_i}{2\Delta u^2} - \frac{\tau'_i}{2\Delta u} - \frac{\tau''_i}{4\Delta u}, \beta_{\tau_i} = \tau'''_i - \frac{\tau'_i}{\Delta u^2}, \gamma_{\tau_i} = \frac{\tau'_i}{2\Delta u^2} + \frac{\tau'_i}{2\Delta u} + \frac{\tau''_i}{4\Delta u}$$

and $\tau_i' = \tau'(u_i)$, $\tau_i'' = \tau''(u_i)$ and $\tau_i''' = \tau'''(u_i)$.

We assume that the velocities and accelerations at the beginning and end are zero:

$$q(0) = q(1) = 0, \mathbf{a}(0) = \mathbf{a}(1) = \mathbf{0}.$$

Since

$$j_\tau = \frac{da_\tau}{dt} = a_\tau \frac{da_\tau}{dv_\tau} = \frac{1}{2}\frac{da_\tau^2}{dv},$$

for the first interval $[0, u_1]$, $|j_{\tau_1}| = |(a_{\tau_1}^2 - a_{\tau_0}^2)/(2v_{\tau_1} - 2v_{\tau_0})| = |a_{\tau_1}^2/(2\tau_1'\sqrt{q_1})| = |N^2\tau_2'^2\sqrt{q_1}q_2/8\tau_1'| \leq J_\tau$. Similarly for the last segment $[u_{N-1}, 1]$, $|j_{\tau_N}| = |N^2\tau_{N-2}'^2\sqrt{q_{N-1}}q_{N-2}/8\tau_{N-1}'| \leq J_\tau$. Therefore for the jerk constraints in the first and last intervals can be written as

$$|N^2\tau_2'^2\sqrt{q_1}q_2/8\tau_1'| \leq J_\tau \tag{28}$$
$$|N^2\tau_{N-2}'^2\sqrt{q_{N-1}}q_{N-2}/8\tau_{N-1}'| \leq J_\tau \tag{29}$$

Correspondingly, the optimization problem (26) is transformed into the following *non-linear programming problem* with objective function

$$\min_{\{q_i\}} \frac{1}{N}\sum_{i=1}^{N-1}\frac{1}{\sqrt{q_i}} \tag{30}$$

under constraints (16), (17), (18), (27), (28), and (29).

We could use existing algorithms to solve the above nonlinear programming problem. But, the solving procedure is generally quite time consuming and the solutions thus obtained is not guarantied to be optimal. In the next section, we try to reduce the problem into a linear programming problem.

4.3. Relax the problem into a linear programming problem

In this section, we will reduce problem (30) into a linear programming problem by introducing a relaxation technique.

The linear programming problem has the following objective function

$$\max_{Q} cQ \tag{31}$$

and constraints $\hat{M}Q \leq \hat{R}$, $Q \geq 0$, where $Q = (q_1, \ldots, q_{N-1})$, $c = (1, \ldots, 1)$, and $\hat{M}$ will be given below. The objective function is the same as (21), which means to maximized make the total feedrate.

Note that constraints (16), (17), and (18) are already linear in Q. In the following, we will show how to reduce constraints (27), (28), and (29) into linear form.

Let the optimal solution of problem (21) be $\{q_i^\star\}_{i=0,\ldots,N}$. Multiplying the the corresponding $\sqrt{\frac{q_i^\star}{q_i}}$ on both sides of constraints (27)-(29), we have

$$|(\sqrt{q_i^\star}\alpha_{\tau_i}q_{i-1} + \sqrt{q_i^\star}\beta_{\tau_i}q_i + \sqrt{q_i^\star}\gamma_{\tau_i}q_{i+1})| \le J_\tau\sqrt{\frac{q_i^\star}{q_i}}, i = 1, \ldots, N-1 \tag{32}$$

$$|\frac{N^2\tau_2'^2}{8\tau_1'}\sqrt{q_1^\star}q_2| \le J_\tau\sqrt{\frac{q_1^\star}{q_1}} \tag{33}$$

$$|\frac{N^2\tau_{N-2}'^2}{8\tau_{N-1}'}\sqrt{q_{N-1}^\star}q_{N-2}| \le J_\tau\sqrt{\frac{q_{N-1}^\star}{q_{N-1}}} \tag{34}$$

where $\tau \in \{x, y, z, a, c\}$. The left hand sides of (32)-(34) are linear in q_i. We will show how to relax the right hand sides of these inequalities into linear forms.

Any q_i satisfying the constraints of problem (30) must also satisfy the constraints of Theorem 3.2. By Theorem 3.2, any such q_i must satisfy $q_i \le q_i^*, (i = 0, \ldots, N)$. Therefore, we can replace $\sqrt{\frac{q_i^\star}{q_i}} \ge 1$ in (32)-(34) by 1 and obtain

$$|(\sqrt{q_i^\star}\alpha_{\tau_i}q_{i-1} + \sqrt{q_i^\star}\beta_{\tau_i}q_i + \sqrt{q_i^\star}\gamma_{\tau_i}q_{i+1})| \le J_\tau, i = 1, \ldots, N-1 \tag{35}$$

$$|\frac{N^2\tau_2'^2}{8\tau_1'}\sqrt{q_1^\star}q_2| \le J_\tau \tag{36}$$

$$|\frac{N^2\tau_{N-2}'^2}{8\tau_{N-1}'}\sqrt{q_{N-1}^\star}q_{N-2}| \le J_\tau. \tag{37}$$

Consequently, we obtain a linear programming problem with objective function (31) and constraints (16), (17), (18), (35), (36), (36), and (37). The solution of this linear programming problem satisfy all the constraints of problem (26).

The relaxation condition $\sqrt{\frac{q_i^\star}{q_i}} \ge 1$ is very restrictive. The following lemma gives a better estimation for $\sqrt{\frac{q_i^\star}{q_i}}$.

Lemma 4.1 *If* $\{q_i^\star\}_{i=1,\ldots,N-1}$ *is the optimal solution of* (21) *and* $\{q_i\}_{i=1,\ldots,N-1}$ *is the optimized solution of problem* (30)*, then*

$$\sqrt{\frac{q_i^\star}{q_i}} \ge \frac{3}{2} - \frac{q_i}{2q_i^\star} \ge 1.$$

Proof: From Theorem 3.2, we have $q_i \le q_i^\star$, and hence $\frac{3}{2} - \frac{q_i}{2q_i^\star} \ge 1$. Let $t = \sqrt{q_i/q_i^\star}$. Then

$$\sqrt{\frac{q_i^\star}{q_i}} - (\frac{3}{2} - \frac{q_i}{2q_i^\star}) = \frac{t^2}{2} + \frac{1}{t} - \frac{3}{2} = \frac{t^2}{2} + \frac{1}{2t} + \frac{1}{2t} - \frac{3}{2} \ge 3\sqrt[3]{\frac{t^2}{2}\cdot\frac{1}{2t}\cdot\frac{1}{2t}} - \frac{3}{2} = 0. \blacksquare$$

By Lemma 4.1, if q_i satisfy the following constraints, then they also satisfy constraints (32)-(34).

$$|(\sqrt{q_i^\star}\alpha_{\tau_i}q_{i-1}+\sqrt{q_i^\star}\beta_{\tau_i}q_i+\sqrt{q_i^\star}\gamma_{\tau_i}q_{i+1})|\leq J_\tau(\frac{3}{2}-\frac{q_i}{2q_i^\star}) \tag{38}$$

$$|\frac{N^2\tau_2'^2}{8\tau_1'}\sqrt{q_1^\star}q_2|\leq J_\tau(\frac{3}{2}-\frac{q_1}{2q_1^\star}) \tag{39}$$

$$|\frac{N^2\tau_{N-2}'^2}{8\tau_{N-1}'}\sqrt{q_{N-1}^\star}q_{N-2}|\leq J_\tau(\frac{3}{2}-\frac{q_{N-1}}{2q_{N-1}^\star}) \tag{40}$$

Now, we will give the $\hat{M}$ in problem (31). Since M in (22) represents constraints (16), (17), and (18), the first $11N-1$ rows of $\hat{M}$ are the same as that of M.

Let $\hat{\alpha}_{\tau_0}=J_\tau/(2q_1^\star)$, $\hat{\beta}_{\tau_0}=\sqrt{q_1^\star}N^2\tau_2'^2/(8\tau_1')$, $\hat{\alpha}_{\tau_N}=\sqrt{q_{N-1}^\star}N^2\tau_{N-2}'^2/(8\tau_{N-1}')$, $\hat{\beta}_{\tau_N}=J_\tau/(2q_{N-1}^\star)$, $\hat{\alpha}_{\tau_i}=\alpha_{\tau_i}\sqrt{q_i^\star}$, $\hat{\beta}_{\tau_i}=\beta_{\tau_i}\sqrt{q_i^\star}+J_\tau/(2q_i^\star)$, $\hat{\gamma}_{\tau_i}=\gamma_{\tau_i}\sqrt{q_i^\star}$, $\theta_{\tau_i}=J_\tau/q_i^\star-\hat{\beta}_{\tau_i}$, $i=1,\ldots,N-1$, $\hat{\alpha}_i=(\hat{\alpha}_{x_i},\hat{\alpha}_{y_i},\hat{\alpha}_{z_i},\hat{\alpha}_{a_i},\hat{\alpha}_{c_i})^T$, $\hat{\beta}_i=(\hat{\beta}_{x_i},\hat{\beta}_{y_i},\hat{\beta}_{z_i},\hat{\beta}_{a_i},\hat{\beta}_{c_i})^T$, $\hat{\gamma}_i=(\hat{\gamma}_{x_i},\hat{\gamma}_{y_i},\hat{\gamma}_{z_i},\hat{\gamma}_{a_i},\hat{\gamma}_{c_i})^T$, and $\theta_i=(\theta_{x_i},\theta_{y_i},\theta_{z_i},\theta_{a_i},\theta_{c_i})^T$. The inequality (38)-(40) is added to M to obtain $\hat{M}$

$$\hat{M}=\begin{pmatrix}
\lambda_1 & 0 & 0 & ... & 0 & 0\\
-\lambda_1 & 0 & 0 & ... & 0 & 0\\
\hat{\beta}_1 & \hat{\gamma}_1 & 0 & ... & 0 & 0\\
\theta_1 & -\hat{\gamma}_1 & 0 & ... & 0 & 0\\
1 & 0 & 0 & ... & 0 & 0\\
-\lambda_1 & \lambda_2 & 0 & ... & 0 & 0\\
\lambda_1 & -\lambda_2 & 0 & ... & 0 & 0\\
\hat{\alpha}_2 & \hat{\beta}_2 & \hat{\gamma}_2 & ... & 0 & 0\\
-\hat{\alpha}_2 & \theta_2 & -\hat{\gamma}_2 & ... & 0 & 0\\
0 & 1 & 0 & ... & 0 & 0\\
... & ... & ... & ... & ... & ...\\
0 & 0 & 0 & ... & -\lambda_{N-2} & \lambda_{N-1}\\
0 & 0 & 0 & ... & \lambda_{N-2} & -\lambda_{N-1}\\
0 & 0 & 0 & ... & \hat{\alpha}_{N-1} & \hat{\beta}_{N-1}\\
0 & 0 & 0 & ... & -\hat{\alpha}_{N-1} & \theta_{N-1}\\
0 & 0 & 0 & ... & 0 & 1\\
0 & 0 & 0 & ... & 0 & -\lambda_{N-1}\\
0 & 0 & 0 & ... & 0 & \lambda_{N-1}\\
\hat{\alpha}_0 & \hat{\beta}_0 & 0 & ... & 0 & 0\\
\hat{\alpha}_0 & -\hat{\beta}_0 & 0 & ... & 0 & 0\\
0 & 0 & 0 & ... & \hat{\alpha}_N & \hat{\beta}_N\\
0 & 0 & 0 & ... & -\hat{\alpha}_N & \hat{\beta}_N
\end{pmatrix} \tag{41}$$

The right end vector in (31) is

$\hat{R}=(\pi_0,\pi_0,1.5\mathbf{J}_m,1.5\mathbf{J}_m,h(u_1),...,\pi_{N-2},\pi_{N-2},1.5\mathbf{J}_m,1.5\mathbf{J}_m,h(u_{N-1}),\pi_{N-1},\pi_{N-1},1.5\mathbf{J}_m,1.5\mathbf{J}_m)^T$.

Now, all the parameters in problem (31) are given and we can give the velocity planning algorithm.

Algorithm 4.2 VPJ_LP
Input: *Five-axis tool path* $\eta(u)$,$u \in [0,1]$ *in MCS, the sampling period* T*, the feedrate bound* V_m*, five-axis acceleration bounds* $(A_x, A_y, A_z, A_a, A_c)$*, the chord error bound* δ_m*, five-axis jerk bounds* $(J_x, J_y, J_z, J_a, J_c)$ *the number of discretion number* N*, and the initial position of the origin of MCS* $(x_{w_0}, y_{w_0}, z_{w_0})$*.*
Output: *The approximate optimal velocity function* $v(u), u \in [0,1]$ *of problem 15.*

The algorithm is almost the same as Algorithm 3.3. The only difference is that instead of solving linear programming (21), we now solve linear programming (31). Since the matrix $\hat{M}$ is of the form $(13N-1)\times 1$, the computational complexity of Algorithm 3.3 is also $O(N^{3.5})$ floating point arithmetical operations.

5. Simulation results

In this section, simulation results are used to show that Algorithm 4.2 can be used to find the approximate optimal solution very efficiently.

5.1. Simulation results

For the tool path in Figure 3, we use Algorithm 4.2 to find the velocity function obeying jerk constraints. Set $J_x = J_y = J_z = 100000mm/s^3$, $J_a = J_c = 15000°/s^3$. Other parameters are the same as those given in Section 3.4.. The velcoty functions obtained with Algorithms 3.3 and 4.2 are given in Figure 8. The "sharp corners" of the velocity curve under confined acceleration are smoothed by adding the jerk constraint. Furthermore, the velocity curve under confined jerk does change must from that under the acceleration, which shows that the obtained velocity curve is approximate optimal.

Figure 9 shows the theoretical chord error of the velcoty curve under confined jerk, which is calculated by (23). The chord error in Figure 9 is smaller than that given in Figure 6 at some positions.

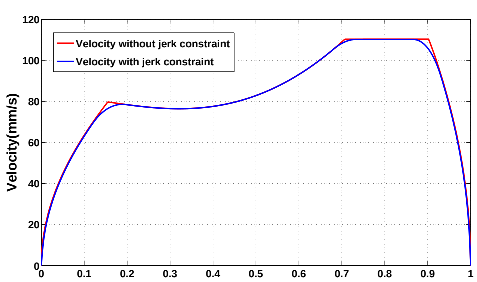


Fig. 8. Feedrates

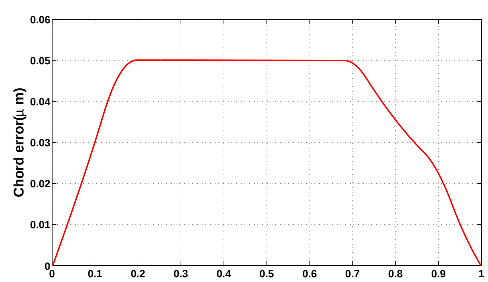


Fig. 9. Chord error with jerk bound

The acceleration curves of five axes with jerk constraints are shown in Figures 10 and 11. These acceleration curves of five axes are continuous. The jerk curves of five axes with jerk constraints are shown in Figure 12 and 13, from which we can see that the jerk bounds are satisfied.

From Figure 8 to 13, we can see that the control profile of velocity is approximate bang-bang-singular. In fact, when $u \in [0, 0.001]$, A-axis jerk reaches J_a. When $u \in [0.001, 0.125]$,

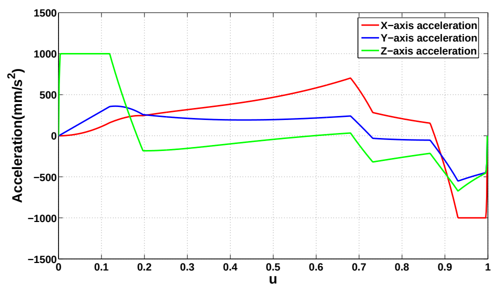


Fig. 10. Accelerations of X, Y and Z axes

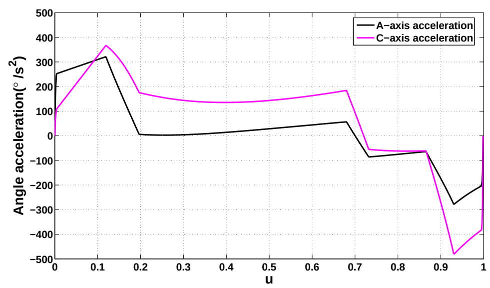


Fig. 11. Accelerations on A and C axes

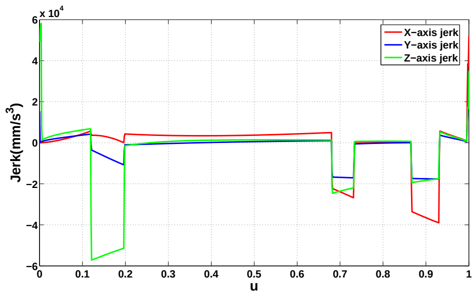


Fig. 12. Jerks of X, Y and Z axes

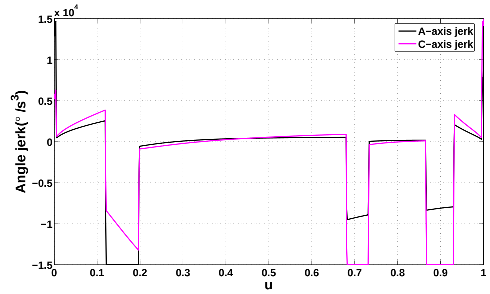
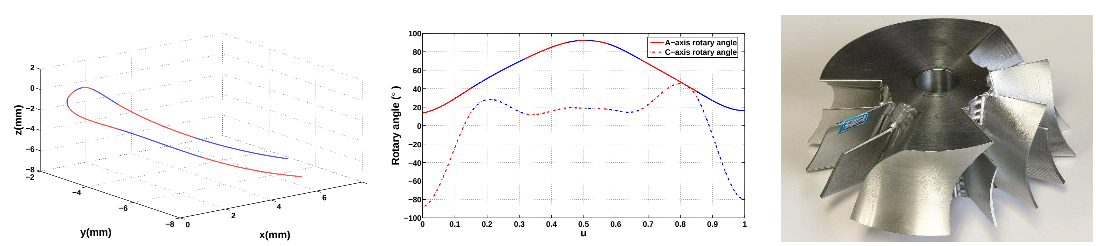


Fig. 13. Jerks of A and C axes with

Z-axis acceleration reaches A_z. A-axis jerk reaches $-J_a$ when $u \in [0.125, 0.192]$. Chord error reaches maximum value when $u \in [0.192, 0.68]$. C-axis jerk reaches $-J_c$ when $u \in [0.68, 0.733]$. Velocity reaches V_m when $u \in [0.733, 0.86]$. Once again C-axis jerk reaches $-J_c$ when $u \in [0.86, 0.93]$. The X-axis acceleration reaches $-A_x$ when $u \in [0.93, 0.997]$. The C-axis jerk reaches J_c when $u \in [0.997, 1]$.

Now compute more complicated tool path shown in Figure 14. This five-axis curve is one piece of tool path for an "impeller" shown in Figure 14(c). Both tool path curve and rotary angle curve are cubic B-splines containing ten segments and having C^2 continuity. The parameters are set to be $V_m = 10mm/s$, $A_x = A_y = A_z = 200mm/s^2$, $A_a = A_c = 500°/s^2$, $J_x = J_y = J_z = 5000mm/s^3$, $J_a = J_c = 5000°/s^3$, $\delta_m = 0.1\mu m$, $T = 3ms$, $N = 500$.



(a) Tool path (b) Rotary angles (c) The impeller

Fig. 14. The tool path and rotary angles for a segment of an impeller

The velocity curves obtained by Algorithm **VPA_LP** and Algorithm **VPJ_LP** are given in Figure 15. The VLC (velocity limit curve) by chord error, which is the maximal feedrate value not violating the chord erro, defined in [5] is also given. The chord errors with and without jerk constraints are shown in 16. Accelerations on five axes are illustrated in Figures 17 and 18. Figures 19 and 20 show the jerks of the X, Y, Z axes and rotary axes A, C respectively. It is easy to find that for the planned velocity, all of constraints are satisfied and motion profile is nearly bang-bang. As a consequence, the velocity is an appropriate solution to the original velocity planning problem (26).

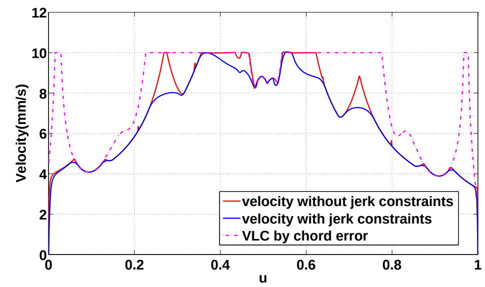


Fig. 15. Velocity curves

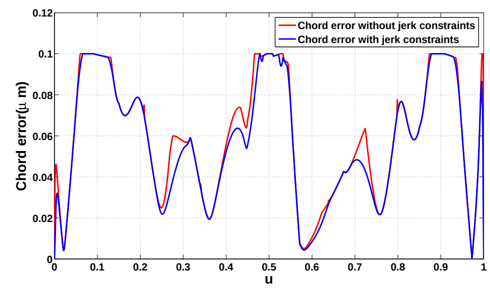


Fig. 16. Chord error curves

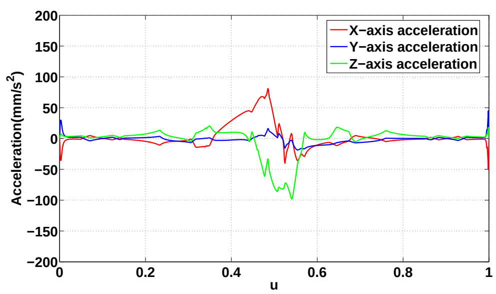


Fig. 17. Accelerations with jerk constraints

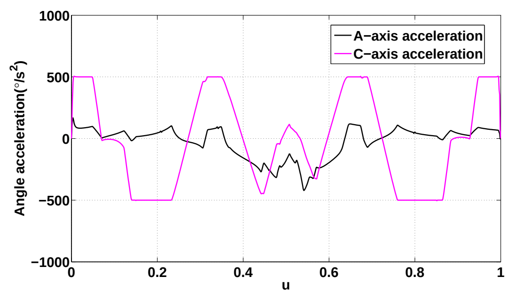


Fig. 18. Accelerations with jerk constraints

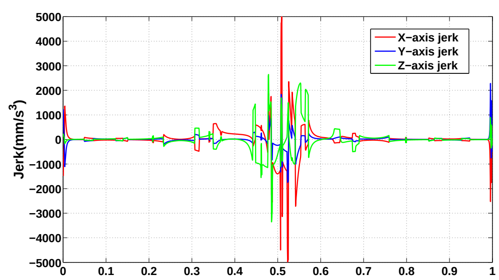


Fig. 19. Jerks with jerk constraints

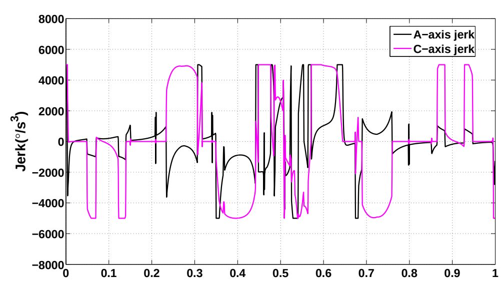


Fig. 20. Jerks with jerk constraints

5.2. Computational costs analysis

In Algorithm **VPA_LP**, linear programming is used to obtain the optimal solution. The larger the number of refined intervals N is, the more closely the solution of Algorithm **VPA_LP** approximates to the solution of the original problem (15). On the other hands,

N	100	200	300	500	1000
Fig. 3	0.328s	0.625s	0.922s	2.187s	3.594s
Fig. 14	0.390s	0.844s	1.218s	2.485s	5.047s

Table 1. Computation times of Algorithm **VPA_LP** for different N

N	Fig. 3		Fig. 14	
	VPJ_LP	**NP**	**VPJ_LP**	**NP**
100	1.609s	42min	2.740s	67min
200	3.766s	149min	5.000s	175min
300	5.797s	failure	6.641s	failure
500	10.875s	failure	10.813s	failure
1000	32.640s	failure	27.047s	failure

Table 2. Computation times of Algorithms **NP** and **VPJ_LP** for different N

computational costs will rise as N becomes lager. In this section, we will use experimental data to show that for our problems the computational complexity is much better than the worst case complexity $O(N^{3.5})$.

In Table 1, the computation times using Algorithm **VPA_LP** to plan the velocities for the tool pathes in Figure 3 and Figure 14 under different N are given.

From Table 1, we have the following observations. Firstly, the actual computational complexity of Algorithm **VPA_LP** is about $O(N)$, that is, the computational time is proportional to N. This is in accordance with Smale's result that the number of operations required to solve a linear programming problem grows in proportion to the number of variables on the average [23] under certain conditions. Secondly, for different tool paths, the computational times are of little difference, which is very significant for practical usage since tool path curves in the practical problem may be rather complicated.

Algorithm **VPJ_LP** computes approximate solutions of the time-optimal velocity planning problem under confined jerk using linear programming methods. The original velocity planning problem under confined jerk is a nonlinear programming problem (30), which is denoted by (**NP**) and can be solved with the nonlinear programming software in Matlab. In Table 2, we compare the computation time for Algorithms **VPJ_LP** and **NP** for different N.

From Table 2, we can see that Algorithm **VPJ_LP** is much faster than nonlinear programming **NP**. When N exceeds 300, **NP** cannot solve the nonlinear problem while Algorithm **VPJ_LP** stills works well.

6. Conclusion

In this paper, the time optimal velocity planning problem for five axis CNC machining along a given parametric tool path under chord error, acceleration, and jerk constraints is studied.

The parameter interval $[0,1]$ is divided into N equal parts and the differential quantities are discretized into finite differences. Then, the velocity planning problem under confined acceleration is reduced to a equivalent linear programming problem, and as a consequence, a polynomial time algorithm with complexity $O(N^{3.5})$ is given to find the optimal solution.

In the case of confined jerk, the nonlinear jerk constraint is approximated with a linear linear function, and the velocity planning problem under confined jerk is approximated with a linear programming program. As a consequence, a polynomial time algorithm is given to find the approximate optimal solution under confined jerk.

As a consequence, efficient polynomial-time algorithms are given for time-optimal velocity planning of five-axis CNC machining along any parametric tool path under confined chord error, acceleration, and jerk. Simulation results and complexity analysis are used to show the efficiency and effectiveness of the algorithms.

References

[1] J.E. Bobrow, S. Dubowsky, J.S. Gibson, Time-optimal control of robotic manipulators along specified paths. *Int. J. Robot. Res.*, 4(3)(1985)3-17

[2] K. Shin, N. McKay, Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control* 30 (1985)531-541.

[3] R.T. Farouki, Y.F. Tsai, Exact Taylor series coefficients for variable-feedrate CNC curve interpolators. *Computer-Aided Design* 33(2)(2001)155-165.

[4] M. Zhang, W. Yan, C.M. Yuan, D.K. Wang, X.S. Gao, Curve fitting and optimal interpolation on CNC machines based on quadratic B-splines. *Science China, Series E* 54(7)(2011)1407-1418.

[5] C.M. Yuan, K. Zhang, W. Fan, X.S. Gao, Time-optimal interpolation for CNC machining along curved tool paths with confined chord error. *MM Research Preprints* 30(2011)57-89.

[6] K. Zhang, X.S. Gao, H.B. Li, C.M. Yuan, A greedy algorithm for feed-rate planning of CNC machines along curved tool paths with confined jerk for each axis. *Robotics and Computer Integrated Manufacturing* 28(2012) 472C483.

[7] D. Bedi, I. Ali, N. Quan, Advanced techniques for CNC machines, *Journal of Enginerring for Industry* 115(1993)329-336.

[8] D.C.H. Yang, T. Kong, Parametric interpolator versus linear interpolator for precision CNC machining. *Computer-Aided Design* 26(3)(1994)225-234.

[9] M.M. Emami, B. Arezoo, A look-ahead command generator with control over trajectory and chord error for NURBS curve with unknown arc length. *Computer-Aided Design* 4(7)(2010)625-632.

[10] S.S. Yeh, P.L. Hsu, Adaptive-feedrate interpolation for parametric curves with a confined chord error. *Computer-Aided Design* 34(2002)229-237.

[11] J.Y. Lai, K.Y. Lin, S.J. Tseng, W.D. Ueng, On the development of a parametric interpolator with confined chord error, feedrate, acceleration and jerk. *Int. J. Adv. Manuf. Technol.* 37(2008)104-121

[12] K. Zhang , C.M. Yuan, X.S. Gao, Efficient Algorithm for Feedrate Planning and Smoothing with Confined Chord Error and Acceleration for Each Axis. *MM-Research Preprints* 30(2011)39-56.

[13] M. Tikhon, T.J. Ko, S.H. Lee, H.S. Kim, NURBS interpolator for constant material removal rate in open NC machine tools. *Int. J. of Mach. Tools and Manu.* 44(2004)237-245

[14] T. Yong, R. Narayanaswami, A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining. *Computer-Aided Design* 35(2003)1249-1259.

[15] W. Fan, X.S. Gao, W. Yan, C.M. Yuan, Interpolation of Parametric CNC Machine Tool Path Under Confined Jounce. *Int. J. Adv. Manuf. Technol.*(2011)DOI:10.1007/s00170-011-3842-0

[16] A.C. Lee, M.T. Lin, Y.R. Pana, W.Y. Lin, The feedrate scheduling of NURBS interpolator for CNC machine tools. *Computer-Aided Design* 43(2011)612-628.

[17] M.S. Tsai, H.W. Nien, H.T. Yau, Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery. *Computer-Aided Design* 40 (2008) 554-566.

[18] A. Gasparetto, A. Lanzutti, R. Vidoni, V. Zanotto, Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning *Robotics and Computer-Integrated Manufacturing* 28(2012)164-181.

[19] B. Sencer, Y. Altintas, E. Croft, Feed optimization for five-axis CNC machine tools with drive constraints. *Int. J. of Mach. Tools and Manu.*48 (2008) 733-745.

[20] S.R. Li, Q. Zhang, X.S. Gao, H. Li, Minimum Time Trajectory Planning for Five-Axis Machining with General Kinematic Constraints. *MM-preprints*, 31(2012)1-20.

[21] K. Erkorkmaz, Y. Altintas, High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation. *Int. J. of Mach. Tools and Manu.* 41(2001)1323-1345

[22] N. Karmarkar, A new polynomial time algorithm for linear programming. *Combinatorica*, 4(4): 373C395, 1984.

[23] S. Smale On the average number of steps of the simplex method of linear programming. *Mathematical Programming*, 27(3):241-262, 1983.