www.cadanda.com Computer-Aided Design AND APPLICATIONS

Hybrid Line-Arc Toolpath Machining with Corner Transition and Grouping Lookahead Scheme

Hongyu Ma[1], Chun-Ming Yuan[2], Li-Yong Shen[1], Fengming Lin[1] and Lixian Zhang[2]

[1]University of Chinese Academy of Sciences, hongyuma@amss.ac.cn
[2]Academy of Mathematics and Systems Sciences, CAS, cmyuan@mmrc.iss.ac.cn
[1]University of Chinese Academy of Sciences, lyshen@ucas.ac.cn
[1]University of Chinese Academy of Sciences, linfengming15@mails.ucas.ac.cn
[2]Academy of Mathematics and Systems Sciences, CAS, shadowfly12@126.com

Corresponding author: Li-Yong Shen, lyshen@ucas.ac.cn

Abstract. In this paper, a corner transition method is presented to improve the feedrates at the junctions of micro-line segments. Grouping look-ahead scheme is presented to improve the global machining speed. At the meantime, a feedrate override method is also presented which is suitable for the machining feedrate online adjustment. This algorithm has been implemented on a commercial CNC system, experimental results show the feasibility of our methods. Compared with several existing algorithms, this algorithm can highly improve the manufacturing efficiency. Moreover, the manufacturing quality can be guaranteed since the velocity and acceleration for each axis are strictly controlled by the capabilities of the CNC machine.

Keywords: CNC, hybrid line-arc toolpath, corner transition, grouping lookahead scheme, feedrate override
DOI: https://doi.org/10.14733/cadaps.2022.854-867

1 INTRODUCTION

As the industry standard for many years, there exists a large amount of numbers of actual workpieces described by the G01 (line segments) and G02/G03 (arcs) codes. Also, the paths of a Computer Numerical Control (CNC) machine described by smooth curves are usually approximated by micro-line segments and arcs. These facts motive us focus on the interpolation algorithm of the G01 and G02/G03 codes. In the process of hybrid line-arc toolpath machining, the direction of machining velocity changes suddenly at the junction between two adjacent line-arc toolpaths, which confines the machining efficiency, or causes vibrations harmfully for both the machine and the product quality. A simple but inefficient solution is to set turning velocity to zero at conjunctions. The key issue is to improve the machining efficiency within the precision and acceleration constraints to achieve high-speed and high-precision CNC machining. The existing methods include two types: one is to fit the global line segments and arcs into smooth curves and interpolate the fitted curves, another one is to transition the local corners between consecutive segments and interpolate the line-arc paths and the corners.

For the first type method, people often fit the G01 and G02/G03 codes using splines and then interpolate the splines. Many work fitted the G01 micro-line segments into conics or NURBS curves, which reduced the amount of data, and improved the overall machining speed and precision [1, 4, 9, 18, 23]. Once the spline curves were generated, one can plan the time optimal interpolation under the CNC machine's acceleration and jerk constraints [13, 19, 20, 23]. In general, spline interpolation has the advantage of less data and smooth paths, while the G01 and G02/G03 codes have the advance of computation simplicity. However, for the spline curves with degree three or higher, the calculations will be complicated for time optimal interpolation.

For the second type method, there are two strategies: one is to design a transition curve to smooth the corners within the tolerance, and then plan the velocities at corners; another one is to merge the above two steps into one step, i.e., design a curve with the time parameter within the machining kinematic constraints and tolerance. In the first strategy, fixed Ferguson splines, arcs, splines or Pythagorean-hodograph corner curves were usually taken as the turning traces [3, 5]. Yang et al. and Zhao et al. [17, 25] took a curvature-continuous B-spline into the corners between consecutive segments. Han et al. [6] smoothed the corners by cubic B-splines. For the five-axis machining, one can also derive double NURBS/spline curves into the corners satisfying the pre-defined error limit for linear tool path [14, 15, 16]. These methods improve the turning velocities in some senses, but the transition paths at the corners are fixed, which limit the turning velocities. In the second strategy, Zhang et al. [22] presented the multi-period turning transition method, in which the transition curves are adjustable according to the maximum velocity, acceleration constraints and machining error bound, which can be shown to be time optimal in some situations. Tajima and Sencer [11, 12] presented the kinematic corner smoothing with the acceleration and jerk limits of drives. Sencer et al. [10] generated the cornering trajectory of the tool through Finite Impulse Response (FIR) filtering. Zhang et al. [24] used a one-step strategy to generate the transition trajectory within the acceleration limits, and proposed a modified corner transition strategy within the tolerance. Li et al. [7] presented a corner smoothing algorithm combining both the tool offset and corner smoothing algorithm to optimize the error constraint. Duan and Okwudire [2] smoothed the corner by a NURBS curve with time parameter. Li et al. [8] adopted the FIR filters method to generate the smoothing curve at the corner, examining the contour error, acceleration and jerk limits. Compared with the fixed curves at the corners, these methods have lower computation costs. However, the toolpaths in the above methods mainly dealt with the line segments. In practical machining, there are also arc toolpaths, hence, the optimization of the hybrid line-arc toolpath machining need to be considered further.

In this paper, we present a transition algorithm which is suitable for hybrid line-arc toolpath, in which the turning acceleration is chosen based on the principle of *bang-bang* control (see Section 2.1 for details). The algorithm achieves optimization under the velocity, acceleration and tolerance constraints. In addition, the lookahead scheme is crucial for improvement of the global machining efficiency, but the computation is time consuming if the number of G01 and G02/G03 codes in lookahead procedure is large. A real-time lookahead algorithm which keeps high machining efficiency is a key issue for real-time machining. In practical machining, the feedrate is usually adjusted online by workers for various cutting situations, that is the feedrate override. We present a grouping look-ahead scheme which guarantees the accessibility of the velocities on each line segment. At the meantime, our grouping look-ahead scheme satisfies the realtime machining and keeps high transition speeds.

The paper is organized as follows: Section 2 introduces the arc discretization algorithm and corner transition algorithm. We present the grouping look-ahead scheme and feedrate override algorithm in Section 3. The experimental results for line-arc toolpath are shown in Section 4. In Section 5, we draw the conclusions.

2 CORNER TRANSITION ALGORITHM FOR HYBRID TOOLPATH

A key issue in CNC machining is to improve the machining feedrate while keeping the machining precision and satisfying the acceleration constraints of the CNC machine. For the consecutive micro-line segments interpolation, the velocities at the junctions of two segments are the bottlenecks for the machining efficiency. Zhang et al. [22] proposed an effective multi-period turning method to improve the feedrate at the junctions using the linear acceleration and deceleration mode, which utilized the maximal acceleration capabilities of the NC machine while satisfying the machining precision. Moreover, this method has holes in making quick respond when the feedrate changes online by workers, which is crucial for real-time manufacturing. However, they didn't consider the corner transition algorithm of G02/G03 codes. In this section, we firstly discretize the arcs to generate a toolpath only composed of line-segments, and then approximate the arcs with several even line segments in order to transform the G02/G03 codes to G01 codes. Then the interpolation of G01-G02/G03 hybrid toolpath could be implemented by the method in Zhang et al. [22].

2.1 Arc Discretization

The arc toolpath can be described by $s(\theta) = (x(\theta), y(\theta))$, $\theta \in [\theta_s, \theta_e]$, where θ is the angle parameter, and θ_s, θ_e are the starting angle and ending angle respectively. Without loss of generality, we assume that the center of the circle is the origin, then

$$\begin{cases} x(\theta) = r\cos(\theta), \\ y(\theta) = r\sin(\theta) \end{cases} \tag{1}$$

where r is the radius of the arc. Since the line interpolation is developed with high efficiency, arc discretization into line segments is a simple and effective way to do arc interpolation.

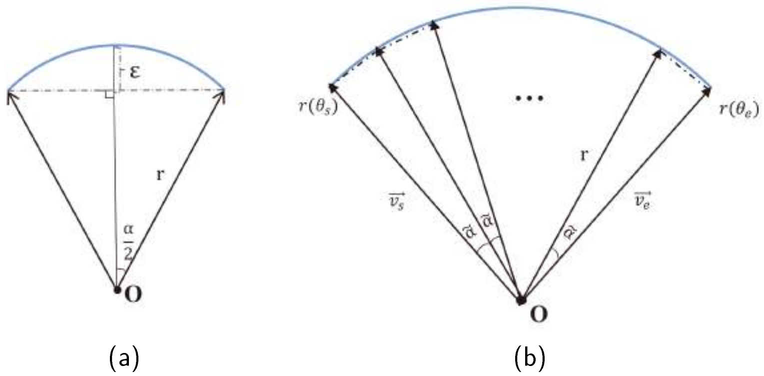


Figure 1: Arc Division:(a)The longest arc with chord error ε can be approximated as a micro-line; (b)Divide arc $s(\theta)$ per $\tilde{\alpha}$, $\theta \in [\theta_s, \theta_e]$.

A part of arc whose chord error is not large than ε can be approximated as one micro-line, where ε is the error, see Figure 1. Then the maximal angle of the arc is denoted as α.

$$r - r\cos(\alpha/2) = \varepsilon,\ \alpha = 2\arccos(1 - \varepsilon/r).$$

There may be very short arc left if the total arc is divided per α. Micro-line, if too small, is supposed to be avoided which will result in low machining velocity [22]. Thus we adjust the angle denoted as $\tilde{\alpha}$ to divide the arc evenly,

$$N = [\frac{|\theta_e - \theta_s|}{\alpha}] + 1,\ \tilde{\alpha} = \frac{|\theta_e - \theta_s|}{N}.$$

We split the arc $s(\theta), \theta \in [\theta_s, \theta_e]$ into N even line segments. After arc discretization, we obtain a tool-path with only line segments. We use method in section 2.2 to design the corner transition algorithm for line segments.

2.2 Corner Transition Algorithm

Let $\mathbf{P} = \{\overline{\mathbf{p}_{k-1}\mathbf{p}_k}, k = 1, \ldots, n\}$ be a serial of consecutive G01 toolpaths. The corner transition algorithm is shown in Figure 2. We denote $L_k = \|\mathbf{p}_k - \mathbf{p}_{k-1}\|, 1 \le k \le n$. At every corner $\angle \mathbf{p}_{k-1}\mathbf{p}_k\mathbf{p}_{k+1}$, $\mathbf{p}_{k,s}$ and

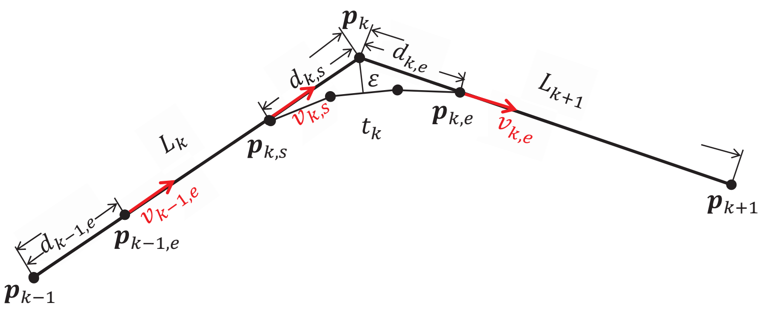


Figure 2: Multi-period turning interpolation at corner.

$\mathbf{p}_{k,e}$ are the starting and ending points of the turning interpolation respectively. Without loss of generality, we assume that $v_{0,s} = V_{0,e} = v_{n,e} = v_{n,s} = 0$. Let $\widetilde{\mathbf{p}_{k,s}\mathbf{p}_{k,e}}$ denote the corner transition curve. And $v_{k,s}$ and $v_{k,e}$ are ex-turning and post-turning velocity respectively. The line segment $\mathbf{p}_{k,s}\mathbf{p}_k$ is the ex-turning line, and $d_{k,s} = \|\mathbf{p}_k - \mathbf{p}_{k,s}\|$, $\mathbf{e}_s = \overrightarrow{\mathbf{p}_{k,s}\mathbf{p}_k}/d_{k,s}$. The line segment $\mathbf{p}_k\mathbf{p}_{k,e}$ is the post-turning line, and $d_{k,e} = \|\mathbf{p}_{k,e} - \mathbf{p}_k\|$, $\mathbf{e}_e = \overrightarrow{\mathbf{p}_k\mathbf{p}_{k,e}}/d_{k,e}$. The deviation from the corner transition curve to the corner point is the error denoted by E. And the total turning interpolation time is $t_k, 1 \le k \le n-1$.

Assuming that the turning trajectory is determined by a constant acceleration $\mathbf{a}$ for each corner respectively, we have the relationships for corner transitions

$$v_{k,e}\mathbf{e}_e - v_{k,s}\mathbf{e}_s = t_k\mathbf{a},\ 1 \le k \le n-1.$$

Then we can derive $v_{k,s}$ and $v_{k,e}$ as

$$v_{k,s} = t_k\|\frac{\mathbf{a} \times \mathbf{e}_e}{\mathbf{e}_e \times \mathbf{e}_s}\|,\ v_{k,e} = t_k\|\frac{\mathbf{a} \times \mathbf{e}_s}{\mathbf{e}_e \times \mathbf{e}_s}\|. \tag{2}$$

The trajectory of turning interpolation is a parabola which can be written as follows

$$d_{k,s}\mathbf{e}_s + d_{k,e}\mathbf{e}_e = v_{k,s}\mathbf{e}_s t_k + 0.5\mathbf{a}t_k^2. \tag{3}$$

Combining (2) and (3), we have

$$d_{k,e} = t_k^2\|\frac{\mathbf{a} \times \mathbf{e}_s}{2\mathbf{e}_e \times \mathbf{e}_s}\| = t_k\frac{v_e}{2},\ v_{k,e} = v_{k,s}t_k - t_k^2\|\frac{\mathbf{a} \times \mathbf{e}_e}{2\mathbf{e}_e \times \mathbf{e}_s}\| = t_k\frac{v_s}{2}. \tag{4}$$

According to the multi-period method proposed in [22], we could obtain the turning interpolation time $t_{k,m}$according to the error E,

$$t_{k,m} = \sqrt{\frac{8E}{|\mathbf{a}|}}. \tag{5}$$

Finally, we can get the corresponding turning velocities ($v_{k,sm}$, $v_{k,em}$) and turning distances ($d_{k,sm}$, $d_{k,em}$) which depends on $t_{k,m}$ by (2) and (4).

Given the machining feedrate constraint F, or equivalently, the maximal velocity $v_{\max}$, through the whole manufacturing process, the turning velocities need to be not greater than $v_{\max}$. The lengths of $d_{k,sm}$, and $d_{k,em}$ need to be not greater than the half of the length of L_k and L_{k+1} respectively for the convenient of look-ahead scheme [22]. Hence, we can get the updated turning time t_k by the following equation.

$$t_k = t_{km} \cdot \min(1, \frac{v_{\max}}{\max(v_{k,sm}, v_{k,em})}, \sqrt{\frac{L_k}{2d_{k,sm}}}, \sqrt{\frac{L_{k+1}}{2d_{k,se}}}) \tag{6}$$

The corresponding turning velocities ($v_{k,s}$, $v_{k,e}$) and turning distances ($d_{k,s}$, $d_{k,e}$) which depends on t_k can be obtained by (2) and (4). The ratio of $v_{k,s}$ to $v_{k,e}$ is constant because the acceleration of turning interpolation is constant. We have that the ratios of $\lambda_k = v_{k,e}/v_{k,s}$ and $\eta_k = t_k/v_{k,s}$ are constant at each corner for $1 \leq k \leq n-1$.

3 GROUPING LOOKAHEAD SCHEME AND FEEDRATE OVERRIDE ALGORITHM

3.1 General Lookahead Scheme

In addition to the corner transition interpolation, the interpolation of G01 codes includes the interpolation of line segments. The line segment interpolation algorithm is to calculate the interpolation point sequence on the line segment. We adopt the linear acceleration and deceleration (acc/dec) mode, which is shown in Figure 3 and Figure 4, where $t_{k,a}$, $t_{k,c}$ and $t_{k,d}$ are the constant acceleration time, constant speed time and constant deceleration time respectively.

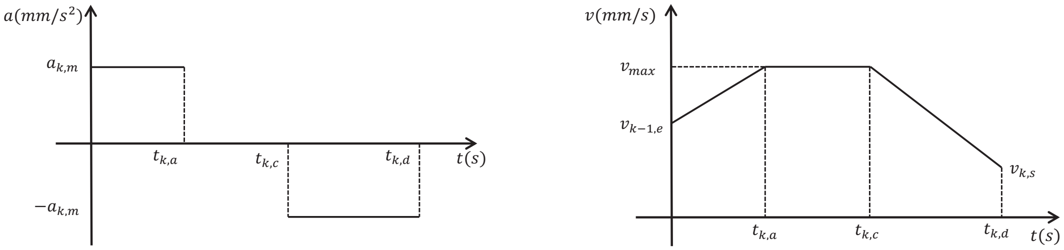


Figure 3: Linear acc/dec mode when the maximal velocity can be reached.

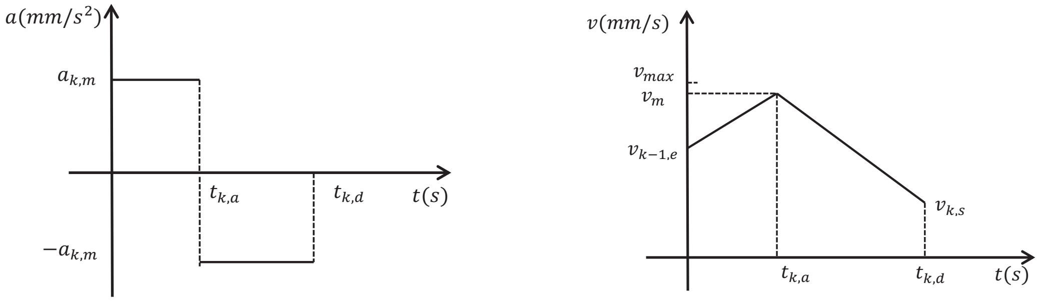


Figure 4: Linear acc/dec mod when the maximal velocity cannot be reached.

Look-ahead scheme gives an effective way to plan velocity along the whole toolpath. The principle of look-ahead scheme is to ensure that all $(v_{k,s}, v_{k,e}), k = 1, \ldots, n$ are accessible. The accessibility test is imperative.

Firstly, we test accessibility at each line segment. The forward accessibility: After turning interpolation at $\angle \mathbf{p}_{k-2}\mathbf{p}_{k-1}\mathbf{p}_k$, the end speed is the post-turning velocity $v_{k-1,e}$, see Figure 2, the next ex-turning velocity $v_{k,s}$ should be accessible from $v_{k-1,e}$ along line segment $\mathbf{p}_{k-1,e}\mathbf{p}_{k,s}$.

$$l_k = |\mathbf{p}_{k-1,e}\mathbf{p}_{k,s}| = L_k - d_{k-1,e} - d_{k,s} \tag{7}$$

For each line segment $\mathbf{p}_{k-1,e}\mathbf{p}_{k,s}$, the maximal acceleration $a_{k,m}$ is determined by the direction of the line and the acceleration ability, i.e. the maximal accelerations of X, Y, Z-axis A_x, A_y, A_z. As for 3-axis machine, $a_{k,m}$ can be computed

$$a_{k,m} = \min\{\frac{A_x}{\cos\theta_x}, \frac{A_y}{\cos\theta_y}, \frac{A_z}{\cos\theta_z}\} \tag{8}$$

where θ_x, θ_y, θ_z are the angles between the interpolation line and the corresponding axis respectively.

During the CNC machining, the interpolation should be able to move in speed $v_{k-1,e}$ to $v_{k,s}$ along the line segment $\mathbf{p}_{k-1,e}\mathbf{p}_{k,s}$; otherwise, $v_{k,s}$ should be modified. Note that the starting speed at point $\mathbf{p}_0$ is zero, that is, $v_{0,s} = v_{0,e} = 0$, the forward accessibility test is as follows:

Forward accessibility test from $\mathbf{p}_0$ to $\mathbf{p}_{n+1}$:

For $k = 1$ *to* n **do**

If $l_k < |v_{k,s}^2 - v_{k-1,e}^2|/(2a_{k,m})$

If $v_{k-1,e} \leq v_{k,s}$

$$v_{k,s} = \sqrt{\frac{2a_{k,m}L_k - a_{k,m}v_{k-1,e}t_{k-1} + v_{k-1,e}^2}{1 + a_{k,m}t_k/v_{k,s}}},$$

$$v_{k,e} = \lambda_k v_{k,s},\ t_k = \eta_k v_{k,s},\ k = k+1.$$

Else

$k = k+1.$

Return.

Backward accessibility test is similar as the forward accessibility test. The ex-turning velocity $v_{k,s}$ at the last line with post-turning point $\mathbf{p}_{k,s}$ should be able to reach the starting velocity $v_{k-1,e}$, within the maximal acceleration on each line segment and the length of each line segment. We test the backward accessibility from the last line until the test is done when one of the following two conditions is satisfied: firstly, the ending velocity is no less than the starting velocity; secondly, the acc/dec mode is reachable.

Note that the ending speed at $\mathbf{p}_n$ is zero, that is, $v_{n,s} = v_{n,e} = 0$, then the backward accessibility test is as follows:

Backward accessibility test from $\mathbf{p}_n$ to $\mathbf{p}_0$:

For $k = n$ *to* 0 **do**

If $l_k < |v_{k,s}^2 - v_{k-1,e}^2|/(2a_{k,m})$

If $v_{k-1,e} \geq v_{k,s}$

$$v_{k-1,e} = \sqrt{\frac{2a_{k,m}L_k - a_{k,m}v_{k,s}t_k + v_{k,s}^2}{1 + a_{k,m}t_k/v_{k-1,e}}},$$

$$v_{k-1,s} = v_{k-1,e}/\lambda_{k-1},\ t_{k-1} = \eta_{k-1}v_{k-1,s},\ k = k-1.$$

Else

Return.

After the accessibility tests, the velocity planning is obtained. However, this lookahead scheme could not be embedded in real machining system directly with the limitation of memory and hardware. There are always up to millions of line segments to be processed in the real world problems (see examples in section 4). The practical CNC system could store within three thousands line segments, for example, the Blue Sky CNC System.

Assume that the CNC system could store no more than N_d line segments. A basic method is to divide the toolpath into groups of N_d segments each. For each group including N_d line segments, we could build up its look-ahead scheme. However, the machining need to stop and restart at every N_d line segments. This way would cut down the machining efficiency and quality. Without losing machining efficiency, we propose the grouping look-ahead scheme to figure out the memory limitation problem in the next Section.

3.2 Grouping Lookahead Scheme

In order to avoid losing efficiency, the "zero velocity" needs to be eliminated. Given the line length and the acceleration, we could obtain the maximal number $N_{d0,i}$ of line segments within which the maximal velocity may be reached.

For each group $\{\mathbf{p}_{i_0}, \ldots, \mathbf{p}_{i_{N_d}}\}$, let $d_{i,\min} = \min\{L_{i_1}, \ldots, L_{i_{N_d}}\}$. Then we need to seek the minimum $N_{d0,i}$ which satisfying the following inequality

$$2a_{i,\min} N_{d0,i} d_{i,\min} \geq v_{\max}^2$$

where $a_{i,\min} = \min(A_{x,m}, A_{y,m}, A_{z,m})$.

All the computing works above need to be finished within an interpolation period. Computing each $N_{d0,i}$ is time consuming, without losing of generality, we set $d_{i,\min} = \varepsilon$ which can be read from the G codes. Then a fixing number for $N_{d0,i}$ is obtained as follows.

$$N_{d0} = \lceil v_{\max}^2 / (2a_{i,\min}\varepsilon) \rceil$$

In the segments $iN_d - iN_{d0}, \ldots, iN_d - (i-1)N_{d0}$, the velocity is reachable at each point (See Figure 5). Therefore, we maintain the larger one among the two velocity values of two groups. Let $v_{k,s}^*$ record the ex-turning velocity obtained by the last look-ahead computing, similarly as $v_{k,e}^*$, t_k^*; $v_{k,s}$, $v_{k,e}$ and t_k are the latest value.

$$v_{k,s} = \max\{v_{k,s}^*, v_{k,s}\}, \; v_{k,e} = \lambda_k v_{k,s}, \; t_k = \eta_k v_{k,s}$$
$$k = iN_d - iN_{d0}, \ldots, iN_d - (i-1)N_{d0}.$$

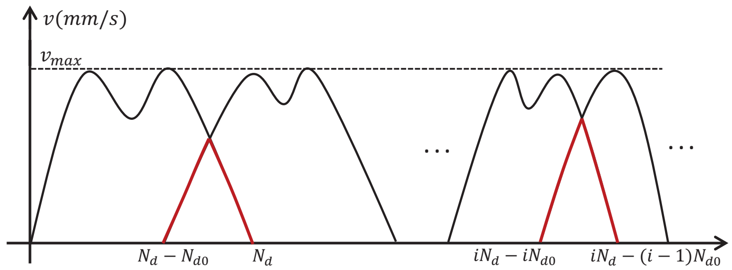


Figure 5: Grouping method.

By the linear acc/dec mode, one can show that the interpolation value $(v_{k,s}, v_{k,e}, t_k)$ is the same as the value obtained in section 3.1 by the grouping method. Therefore, the grouping method gives a feasible way to be implemented into the practical applications without losing any efficiency.

3.3 Quick Respond Method for Feedrate Override

In CNC machining, feedrate override is used frequently. For example, the machining process needs to stop when workers find something abnormal. The velocity should decrease from current speed to zero as soon as possible within the machining ability. Or if workers wish to slow down the machining when the vibration happens, that means the velocity should be reduced to the desired velocity quickly. In other cases, the velocity is better to increase quickly when the maximal velocity increase. Therefore, if the machine parameters change, the above look-ahead scheme is not suitable for the new system. However, we expect that the system could response the change as early as possible. The quick respond method for feedrate override is important for CNC machining.

Zhang et al. [22] proposed a theoretical method to solve the feedrate override problem. This method achieved the fast respond to the increase of feedrate by adjusting both at line and corner. However, this method is not easy to be implemented in the practical CNC machine.

We give a quick response method for feedrate override process. Our method could deal with kinds of cases including system pause, feedrate increasing and decreasing. In order to avoid acceleration override, it is not responded at every corner. The feedrate override is handled on the line segments. The interpolation is under acc/dec mode.

Let $\mathbf{p}_{k,a}$ denote the interpolating point on the line segment $\mathbf{p}_{k-1}\mathbf{p}_k$ when the feedrate v_{max} is adjusted. At point $\mathbf{p}_{k,a}$, the current velocity $v_{k,a}$ and the interpolation distance $d_{k,a}$ from $\mathbf{p}_{k-1}$ are recorded during the interpolation (See Figure 6).

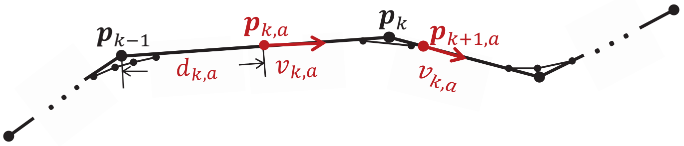


Figure 6: Feedrate override.

During the interpolation, there are two kinds of feedrate override: increasing to $v_{\max+}$, or decreasing to $v_{\max-}$. For each case, the $v_{k,a}$ and $d_{k,a}$ of point $\mathbf{p}_{k,a}$ cannot be changed because the former segment has been machined. We propose a method to deal with the two classes based on this fact. The two classes can be handled by the same way in our method. Let $v^*_{\max} = v_{\max+} \ or \ v_{\max-}$. The feedrate override at $\mathbf{p}_{k,a}$ is as follows.

Step 1. Build up a virtual point $\tilde{\mathbf{p}}_{k-1} = (\tilde{x}_{k-1}, \tilde{y}_{k-1}, \tilde{z}_{k-1})$.
Let $(\tilde{x}_{k-1}, \tilde{y}_{k-1}, \tilde{z}_{k-1}) = (x_{k-1}, y_{k-1}, z_{k-1})$, $\tilde{v}_{k-1,s} = \tilde{v}_{k-1,e} = 0mm/s, \tilde{t}_{k-1} = 0s$.

Step 2. Compute $(v^*_{j,s}, v^*_{j,e}, t^*_j)$ for each segment $\mathbf{p}_{j-1}\mathbf{p}_j$ with the new feedrate $v^*_{\max}$, $j = k, \ldots, m$, where m is index of the last point in the look-ahead buffer queue.

Step 3. Accessibility test.
If $\mathbf{p}_{k,a} \in \mathbf{p}_{k,s}\tilde{\mathbf{p}}_{k,e}$, go to **Step 5**.
If $L_k - d_{k,a} - 0.5v^*_{k,s}t^*_k < |(v^{*2}_{k,s} - v^2_{k,a})/(2a_{k,m})|$, go to **Step 4**;
Else, the new velocity is reachable: update $(v_{j,s}, v_{j,e}, t_j) = (v^*_{j,s}, v^*_{j,e}, t^*_j)$, $j = k, \ldots, m$,
return.

Step 4. **If** $v_{k,a} \le v^*_{k,s}$, **Then** $v_{k,s} = u\sqrt{\frac{2a_{k,m}(L_k - d_{k,a}) + v^2_{k,a}}{u^2 + 4ua_{k,m}}}$;
Else, $v_{k,s} = u\sqrt{\frac{2a_{k,m}(L_k - d_{k,a}) - v^2_{k,a}}{u^2 + 4ua_{k,m}}}$, where $u = \frac{\mathbf{a} \times \mathbf{e}_e}{\mathbf{e}_e \times \mathbf{e}_s}$;

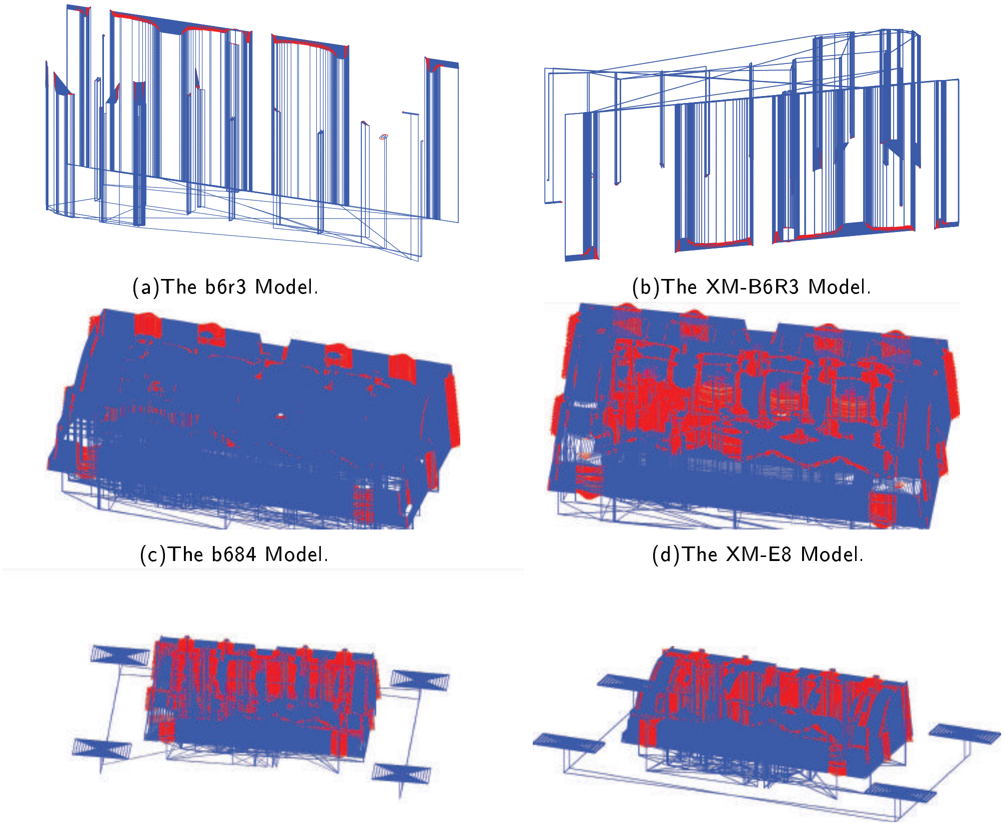
Step 5. Let $k = k + 1$, $p_{k,a}$ be the first interpolation point on the line $\mathbf{p}_{k-1}\mathbf{p}_k$.
Go to **Step 3**.

We implemented this method into the Blue Sky CNC system. The experiments shows the effectiveness of the method. This method may slow down the response sensibility by setting a virtual “zero point" in Step 1. However, this method works well in practical machining process and is easy to achieved.

4 EXPERIMENTS AND RESULTS

Our method is embedded into Blue Sky NC System of Shenyang Institute of Computing Technology Co. Ltd, CAS. We make simulation experiments for several data provided by Guangxi Yuchai Machinery Group Co., Ltd and compare the interpolation time between different algorithms: Yuchai's method provided by Guangxi Yuchai Machinery Group Co., Ltd, Zhang XH's method in [21] and our method (hereinafter referred to as *YC method* and *ZXH method*).

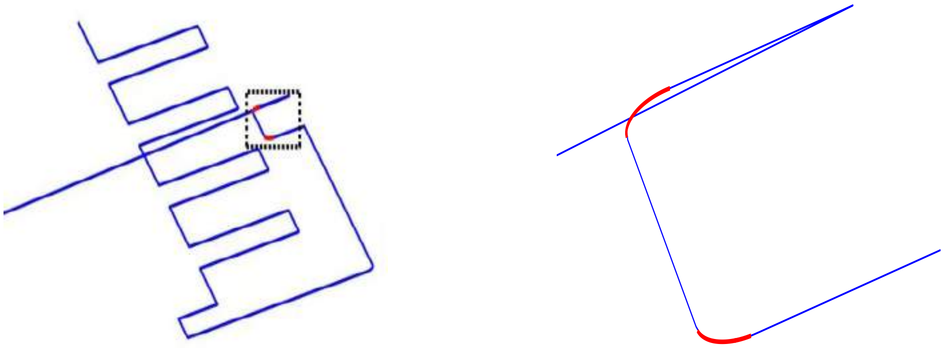
The machining parameters are: the maximal acceleration of each axis is $1600mm/s^2$, the maximal velocity (feedrate) is $250mm/s$, the interpolation time is $2ms$, and the machining error is $0.1mm$. The N_{d0} here can be set as 30.



(a)The b6r3 Model. (b)The XM-B6R3 Model.
(c)The b684 Model. (d)The XM-E8 Model.
(e)The e16 Model. (f)The XM-E16 Model.

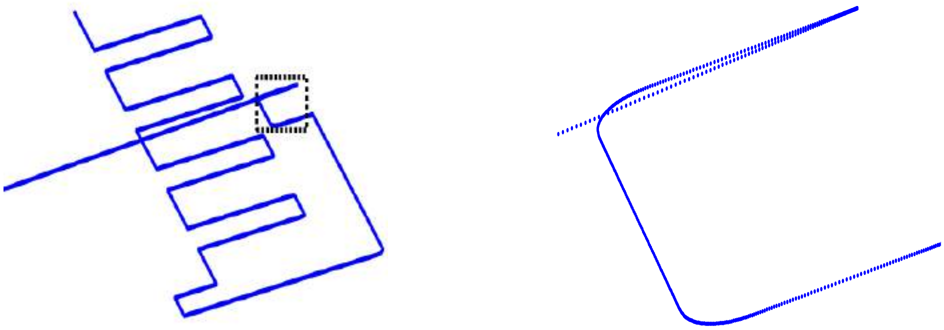
Figure 7: Machining Models: Blue lines represent the G01 codes, and red curves represent G02/G03 arcs.

We implemented our algorithm on the XM-E8 model composed by arc and linear paths, a part of the path is shown in Figure 8 and the interpolation points are shown in Figure 9. The velocity and acceleration of each axis are shown in Figure 10. One can find that velocity and acceleration are strictly controlled by the capabilities of the CNC machine.



(a)A segment path of the XM-E8 model. (b) The local enlargement of (a).

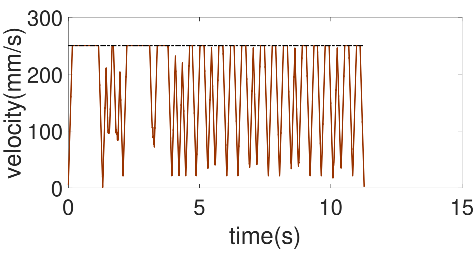
Figure 8: A part of the original manufactured path.



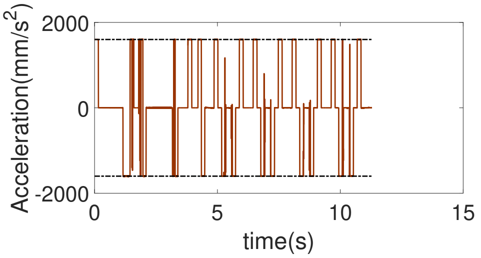
(a) The interpolation points. (b) The local enlargement of (a).

Figure 9: The interpolation points of Figure 8.

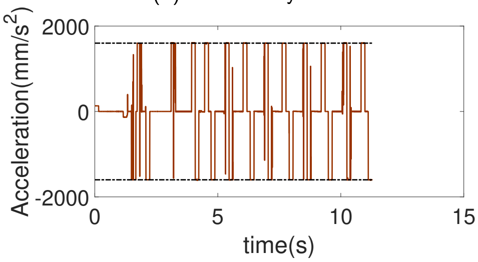
The machining tests are implemented on six models of machining products of Yuchai Machinery Group (see Figure 7). The "YC" method provided by Guangxi Yuchai Machinery Group Co., Ltd means the algorithm which was preset in the machine in Guangxi Yuchai Machinery Group Co., Ltd. To compare our results with other methods, we updated the original numerical control system with Blue Sky NC System which originally equipped with "ZXH" algorithm. And our algorithm was embedded in Blue Sky NC System. A real machining situation is shown in Figure 11. The comparison of whole simulated machining time is shown in Table 1. The data volume is evaluated by the numbers of line segments. With the same machining parameters same parameters, compared with YC's method, our method could cut down the overall interpolation time by $150\% \sim 200\%$, while the ZXH's method by $30\% \sim 50\%$.



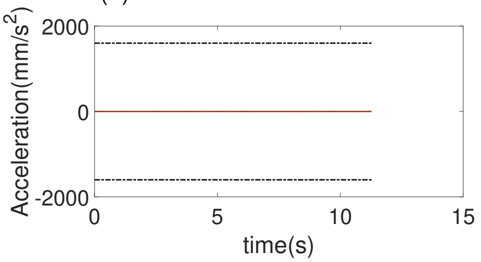
(a)The velocity.



(b)The X-axis acceleration.



(c)The Y-axis acceleration.



(d)The Z-axis acceleration.

Figure 10: The velocity and acceleration for each axis.

(a) The machine.

(b) The workpiece.

Figure 11: The experimental CNC machine and the workpiece.

Table 1: Interpolation time comparison between different interpolation algorithms.

Models	toolpath length(m)	Data Volume	Overall time (min)			Velocity improvement (%)	
			YC	ZXH	our	our vs. YC	ZXH vs. YC
b6r3	161.361	6856	40	27.40	13.84	189.02	45.99
XM-B6R3	160.548	6115	45	30.35	15.17	196.64	48.27
b684	2189.99	953892	1627	1154.41	600.91	170.76	40.94
XM-E8	1526.88	274155	639	447.96	233.73	173.39	42.65
e16	2037.45	150412	590	445.15	224.42	162.90	32.54
XM-E16	2038.54	150127	584	442.59	227.15	156.99	31.95

5 CONCLUSIONS

High speed and high precision are the main research task in the CNC machining. For the hybrid line-arc toolpath interpolation, the turning velocity improvement at the corners make sense for this goal. We proposed the corner transition interpolation method for the hybrid line-arc toolpath machining with grouping look-ahead scheme, which satisfies the machining error allowance, realtime performance, acceleration and feedrate constraints. In certain senses, the turning velocity is optimal. The grouping look-ahead scheme predetermines the global accessibility efficiently. As a result, the vibration caused by the sudden velocity change is weakened. Our algorithm improves the machining speed significantly compared with the existing algorithms. In the meantime, the calculations are simple which meets the requirements of the real-time interpolation and online feedrate override. This algorithm also has the advantage of broad applications suitable for line and arc toolpath. The implemented CNC machining results of this algorithm show the advantages of the method.

ACKNOWLEDGEMENTS

This work is partially supported by Beijing Natural Science Foundation (No. Z190004), National Natural Science Foundation of China under Grants 11688101 and National Center for Mathematics and Interdisciplinary Sciences. We thank Shenyang Institute of Computing Technology Co. Ltd, CAS for the simulation in their Blue Sky NC System and Guangxi Yuchai Machinery Group Co., Ltd for their experimental model data.

Hongyu Ma, http://orcid.org/0000-0001-6889-6286
Li-Yong Shen, http://orcid.org/0000-0001-5769-4814
Chun-Ming Yuan, http://orcid.org/0000-0003-3349-533X
Fengming Lin, http://orcid.org/0000-0002-6671-0748
Li-Xian Zhang, http://orcid.org/0000-0003-1065-916X

REFERENCES

[1] Du, x.; Huang, J.; Zhu, L.m.; Ding, H.: An error-bounded b-spline curve approximation scheme using dominant points for cnc interpolation of micro-line toolpath. Robotics and Computer-Integrated Manufacturing, 64, 101930, 2020. http://doi.org/10.1016/j.rcim.2019.101930.

[2] Duan, M.; Okwudire, C.: Minimum-time cornering for cnc machines using an optimal control method with nurbs parameterization. The International Journal of Advanced Manufacturing Technology, 85, 1405–1418, 2016. http://doi.org/10.1007/s00170-015-7969-2.

[3] Essid, M.; Gassara, B.; Baili, M.; Hbaieb, M.; Sa, W.B.: Analytical modeling of the cnc machine axis motion in high-speed milling with local smoothing. The International Journal of Advanced Manufacturing Technology, 105, 457–470, 2019. http://doi.org/10.1007/s00170-019-04157-4.

[4] Fan, W.; Lee, C.; Chen, J.: A realtime curvature-smooth interpolation scheme and motion planning for cnc machining of short line segments. International Journal of Machine Tools & Manufacture, 96, 27–46, 2015. http://doi.org/10.1016/j.ijmachtools.2015.04.009.

[5] Farouki, R.T.; Pelosi, F.; Lucia Sampoli, M.: Optimization of corner blending curves. Computer-Aided Design, 117, 102739, 2019. http://doi.org/10.1016/j.cad.2019.102739.

[6] Han, J.; Jiang, Y.; Tian, X.; Chen, F.; Lu, C.; Xia, L.: A local smoothing interpolation method for short line segments to realize continuous motion of tool axis acceleration. The International Journal of Advanced Manufacturing Technology, 95, 1729–1742, 2018. http://doi.org/10.1007/s00170-017-1264-3.

[7] Li, B.; Zhang, H.; Ye, P.: Error constraint optimization for corner smoothing algorithms in high-speed cnc machine tools. The International Journal of Advanced Manufacturing Technology, 99, 635–646, 2018. http://doi.org/10.1007/s00170-018-2489-5.

[8] Li, D.; Zhang, L.; Yang, L.; Mao, J.: Accurate interpolation and contouring control algorithm based on fir filters for the corner transition. The International Journal of Advanced Manufacturing Technology, 109, 1775–1788, 2020. http://doi.org/10.1007/s00170-020-05491-8.

[9] Lin, F.; Shen, L.Y.; Yuan, C.M.; Mi, Z.: Certified space curve fitting and trajectory planning for cnc machining with cubic b-splines. Computer-Aided Design, 106, 13–29, 2019. http://doi.org/https://doi.org/10.1016/j.cad.2018.08.001.

[10] Sencer, B.; Ishizaki, K.I.; Shamoto, E.: High speed cornering strategy with confined contour error and vibration suppression for cnc machine tools. CIRP Annals, 64(1), 369–372, 2015. http://doi.org/10.1016/j.cirp.2015.04.102.

[11] Tajima, S.; Sencer, B.: Kinematic corner smoothing for high speed machine tools. International Journal of Machine Tools & Manufacture, 108, 27–43, 2016. http://doi.org/10.1016/j.ijmachtools.2016.05.009.

[12] Tajima, S.; Sencer, B.: Global tool-path smoothing for cnc machine tools with uninterrupted acceleration. International Journal of Machine Tools & Manufacture, 121, 81–95, 2017. http://doi.org/10.1016/j.ijmachtools.2017.03.002.

[13] Timar, S.D.; Farouki, R.T.: Time-optimal traversal of curved paths by cartesian cnc machines under both constant and speed-dependent axis acceleration bounds. Robotics and Computer-Integrated Manufacturing, 23, 563–579, 2007. http://doi.org/10.1016/j.rcim.2006.07.002.

[14] Tulsyan, S.; Altintas, Y.: Local toolpath smoothing for five-axis machine tools. International Journal of Machine Tools & Manufacture: Design, research and application, 96, 15–26, 2015. http://doi.org/10.1016/j.ijmachtools.2015.04.014.

[15] Xu, F.; Sun, Y.: A circumscribed corner rounding method based on double cubic b-splines for a five-axis linear tool path. The International Journal of Advanced Manufacturing Technology, 94, 451–462, 2018. http://doi.org/10.1007/s00170-017-0869-x.

[16] Yan, Y.; Zhang, L.; Zhang, K.: Corner smoothing transition algorithm for five-axis linear tool path. Procedia CIRP, 56, 604–609, 2016. http://doi.org/10.1016/j.procir.2016.10.119.

[17] Yang, J.; Hu, Q.; Ding, H.: A two-stage cnc interpolation algorithm for corner smoothing trajectories with geometric error and dynamics constraints. Procedia CIRP, 56, 306–310, 2016. http://doi.org/10.1016/j.procir.2016.10.022.

[18] Yang, Z.; Shen, L.Y.; Yuan, C.M.; Gao, X.S.: Curve fitting and optimal interpolation for cnc machining

under confined error using quadratic b-splines. Computer-Aided Design, 66, 62–72, 2015. http://doi.org/10.1016/j.cad.2015.04.010.

[19] Yuan, C.M.; Zhang, K.; Fan, W.: Time-optimal interpolation for cnc machining along curved tool pathes with confined chord error. Journal of Systems ence & Complexity, 26, 836–870, 2013. http://doi.org/10.1007/s11424-013-3180-4.

[20] Yuen, A.; Zhang, K.; Altintas, Y.: Smooth trajectory generation for five-axis machine tools. International Journal of Machine Tools & Manufacture, 71, 11–19, 2013. http://doi.org/10.1016/j.ijmachtools.2013.04.002.

[21] Zang, X.; Yu, D.; Yang, D.; Hu, Y.; Han, W.: Corner curve transition interpolation algorithm for high speed machining of micro-line segment. Chinese Journal of Mechanical Engineering, 46(19), 183–191, 2010. http://doi.org/10.3901/JME.2010.19.183.

[22] Zhang, L.; Sun, R.; Gao, X.S.; Li, H.: High speed interpolation for micro-line trajectory and adaptive real-time look-ahead scheme in cnc machining. Science China Technological Sciences, 54(6), 1481–1495, 2011. http://doi.org/10.1007/s11431-011-4329-9.

[23] Zhang, M.; Yan, W.; Yuan, C.M.; Wang, D.; Gao, X.S.: Curve fitting and optimal interpolation on cnc machines based on quadratic b-splines. Science China Information Sciences, 54, 1407–1418, 2011. http://doi.org/0.1007/s11432-011-4237-4.

[24] Zhang, Q.; Gao, X.S.; Li, H.B.L.; Zhao, M.Y.: Minimum time corner transition algorithm with confined feedrate and axial acceleration for nc machining along linear tool path. The International Journal of Advanced Manufacturing Technology, 89, 941–956, 2017. http://doi.org/10.1007/s00170-016-9144-9.

[25] Zhao, H.; Zhu, L.; Ding, H.: A real-time look-ahead interpolation methodology with curvature-continuous b-spline transition scheme for cnc machining of short line segments. International Journal of Machine Tools & Manufacture, 65, 88–98, 2013. http://doi.org/10.1016/j.ijmachtools.2012.10.005.