Real-time smooth trajectory generation for 3-axis blending tool-paths based on FIR filtering

Jun Fang (fangj19@mails.tsinghua.edu.cn)
Tsinghua University https://orcid.org/0000-0002-1912-7533

Bingran Li
https://orcid.org/0000-0002-7765-3902

Hui Zhang

Peiqing Ye

Research Article

Keywords: CNC, smooth trajectory, velocity planning, FIR filter

Posted Date: May 17th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1637727/v1

Title.

Real-time smooth trajectory generation for 3-axis blending tool-paths based on FIR filtering

Author names and Affiliations.

Jun Fang[a,b], Bingran Li[a,b], Hui Zhang[a,b], Peiqing Ye[a,c].

[a] Department of Mechanical Engineering, Tsinghua University, Room A946-2, Lee Shau-Kee Science & Technology Building, Beijing 100084, China

[b] Beijing Key Lab of Precision/Ultra-precision Manufacturing Equipment and Control, Tsinghua University, Beijing, China

[c] The State Key Laboratory of Tribology Tsinghua University, Beijing, China

Corresponding author.

Corresponding author name: Bingran Li,

Addresses: Room A946-2, Lee Shau-Kee Science & Technology Building, Tsinghua University, Beijing, China,

E-mail addresses: libingran1992@126.com

Abstract.

3-axis machining tool-paths are generated using computer-aided manufacturing (CAM) systems and programmed by series of G01 codes mixed with long and short segments. The computer numerical control (CNC) system needs to analyze the path information according to the G codes, through the path smoothing, speed planning, and interpolation process, generate the smooth trajectory and ensure the high-order continuity of velocity. This paper proposes a real-time smooth trajectory generation algorithm with simple structure and low calculation for 3-axis blending machining tool-paths. FIR filters are used to generate a smooth trajectory with bounded acceleration in real-time. Due to the delay of the filter, it will inevitably bring contour error. The feed rate is adjusted by considering the influence of the adjacent segments on the contour error in the case of long segments, then this idea is extended to continuous micro-segments. Finally, through the pre-discretization with certain rules, a direct one-step smooth trajectory generation algorithm for the blending tool-paths is realized. The algorithm not only can effectively control the contour error, but also is suitable for blending tool-paths. Consequently, the effectiveness of the proposed algorithms are validated in simulations and also experimentally on a 5-axis machine tool.

Keywords.

CNC, smooth trajectory, velocity planning, FIR filter

Declarations.

Funding:

The authors did not receive support from any organization for the submitted work.

Conflicts of interest:

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Availability of data and material:

The data sets supporting the results of this article are available from the corresponding author on reasonable request.

Code availability:

The code used during the current study are available from the corresponding author on reasonable request.

Ethics approval: Not applicable

Consent to participate: Not applicable

Consent for publication: Not applicable

Abstract

3-axis machining tool-paths are generated using computer-aided manufacturing (CAM) systems and programmed by series of G01 codes mixed with long and short segments. The computer numerical control (CNC) system needs to analyze the path information according to the G codes, through the path smoothing, speed planning, and interpolation process, generate the smooth trajectory and ensure the high-order continuity of velocity. This paper proposes a real-time smooth trajectory generation algorithm with simple structure and low calculation for 3-axis blending machining tool-paths. FIR filters are used to generate a smooth trajectory with bounded acceleration in real-time. Due to the delay of the filter, it will inevitably bring contour error. The feed rate is adjusted by considering the influence of the adjacent segments on the contour error in the case of long segments, then this idea is extended to continuous micro-segments. Finally, through the pre-discretization with certain rules, a direct one-step smooth trajectory generation algorithm for the blending tool-paths is realized. The algorithm not only can effectively control the contour error, but also is suitable for blending tool-paths. Consequently, the effectiveness of the proposed algorithms are validated in simulations and also experimentally on a 5-axis machine tool.

Keywords: CNC, smooth trajectory, velocity planning, FIR filter

1 Introduction

In typical 3-axis machining, in order to ensure the geometric dimension accuracy of the machined workpiece, computer-aided manufacturing (CAM) systems generally uses linear lines instead of curves to discretize the tool-paths into a large number of G01 line segments [1]. In order to realize high-speed continuous motion, computer numerical control (CNC) systems usually adopts the corner smoothing algorithm based on pre-planning [2]. However, due to the existence of multiple buffers, the real-time performance of the system is not strong, and there are computational efficiency and accuracy issues in spline interpolation. Thus, this paper presents a novel real-time smooth trajectory generation technology for accurate interpolation of 3-axis machining.

The G code generated by CAM is a linear tool-path of point-to-point (P2P) and only has position continuity, so it must be completely stopped at the segment transfer, resulting in a long processing time [3,4]. To ensure the smoothness of machining path and the high-order continuity of speed, CNC usually adopts the "local blending/smoothing" or the "global blending/smoothing" algorithms to improve the geometric continuity of the path, then carries out speed planning and interpolation. In related research, by inserting a straight line segment, arc, B-spline, and Bezier curve at the adjacent tool-path [5-7], the smooth transition at the corner is realized and the speed continuity of the path corner is improved. The local smoothing has a good local property and can effectively control the contour error at the corner [8]. Compared with the local smoothing method, the global smoothing method adopts the fitting or interpolation to fit a large number of continuous short linear line segments into one or more smooth spline curves under the contour error constraint, commonly used splines include Bezier, B-spline and NURBS curves [9-11]. However, most algorithms adopt iterative calculation or even offline mode, which may affect the real-time performance of the system [1]. In actual machining, the tool-path is often a complex path mixed with long and short segments. Therefore, it is necessary to use the local smoothing and the global smoothing algorithms in segments according to the track type, and some tracks at the junction of the two fairing algorithms need to be repeatedly faired. The algorithm structure is complex and the contour accuracy is not easy to control.

After the trajectory is smoothed, the CNC system needs to perform speed planning based on the kinematic constraints of the machine tool, the geometric constraints of the machining path, and the contour errors. The acceleration of S-type acceleration/ deceleration is continuous, the vibration of the machine tool is reduced and the surface quality of the workpiece is improved, so it is widely used in practical applications [12-14]. Furthermore, algorithms with continuous acceleration and even continuous jerk had also been proposed [15-17]. As the order of velocity continuity increases, the amount of calculation also increases geometrically.

Finite Impulse Response (FIR) technology has the characteristics of good smoothness, simple calculation and no spline fitting. Song et al. [18] proved that the velocity curve after FIR filtering is equivalent to the polynomial velocity curve. Therefore, some scholars proposed to use the FIR filters to realize the trajectory control process of the CNC system in one step. The algorithm has a simple structure and is conducive to real-time calculation. For the long segment tool-paths, B Sencer [19] and Liu [20] proposed a path smoothing algorithm based on FIR filtering for adjacent corners. The dwell time is deduced according to the given contour error, so as to adjust the speed overlapping area of the adjacent segments, and realize the precise contouring control while meeting the requirements of vibration avoidance, high precision, and short machining time. R Bearee [21] realized the maximum utilization of acceleration and shortened the processing time by changing the time constant of the filter. Furthermore, in order to realize jerk continuity, Li [22] further deduced the model of constant velocity sequence passing through the 3rd filter and obtained the 3rd order continuous velocity curve satisfying the contour error constraint. Another scheme of contour error control is adjusting the adjacent speed within the delay time, also called the velocity-controlled blending algorithm [23], which can further reduce the velocity fluctuation at the corner. The research of the above scholars only considers the two adjacent trajectories at the corner but does not give the processing method of continuous micro-segments. T Hayasaka [24] proposed a block splitting method to adjust the speed within the delay time, realized the machining of continuous micro-segments based on FIR filtering. B Sencer [25-26] proposed a highly computationally efficient global smoothing interpolation method, determined the final speed proportion coefficient by solving the influence of each corner on multi-stage speed. In the algorithm, trapezoidal acceleration/deceleration are used to solve the contour error, but S-type acceleration/ deceleration are commonly used in processing.

This paper proposes a universal algorithm of acceleration continuity based on Finite Impulse Response (FIR) filtering for complex path machining of CNC systems. Using the 2nd order FIR filters, by adjusting the velocity proportional coefficient, the interpolation smoothing trajectory with continuous acceleration can be directly generated. Firstly, a non-stop velocity proportional compression algorithm for continuous long linear segments is proposed, and the adjustment coefficients of the speed of the front and rear segments are calculated according to the contour errors at the corners. On this basis, a blending velocity proportional compression algorithm for continuous micro-segments is proposed. By considering the influence of multiple trajectories before and after a single corner, the final velocity proportional compression coefficient is calculated, so that the generated trajectory satisfies the contour of each corner. Finally, a pre-discrete velocity proportional compression algorithm for blending tool-paths is proposed, which is suitable for complex trajectories in practical machining. Henceforth, the paper is organized as follows. Section 2 introduces the generation of acceleration continuous jerk-bounded tool-paths based on FIR filters. Section 3 presents the continuous trajectory generation that satisfies the contour error constraint. Section 4 verifies the effectiveness of the algorithm through simulation and actual machining experiments.

2 Jerk-bounded trajectory generation based on FIR filtering

2.1 FIR filter principle

The commonly used acceleration/deceleration curve of the numerical control system is S-shaped acceleration/deceleration, that is, the acceleration is continuous and the jerk is bounded, which can be easily obtained by passing through two cascaded FIR filters through the equivalent speed sequence [20]. The transfer function (TF) of the first-order FIR filter under Laplace(s) is:

$$H_i(s) = \frac{1}{T_i}\frac{1-e^{-sT_i}}{s}, \quad i = 1\mathrm{K}\ N \tag{1}$$

where T_i is the time duration for the i th FIR filter, s denotes the Laplace operator. The transfer function consists of an integrator $\frac{1}{s}$ and a pure delay element e^{-sT_i}. The impulse response is obtained by inverse Laplace transform:

$$h_i(t) = L^{-1}(H_i(s)) = \frac{u(t)-u(t-T_i)}{T_i} \text{ where } u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{2}$$

$h_i(t)$ is essentially a rectangular pulse of unit area, as shown in Fig. 1, which reveals that the final position of the trajectory will not change after passing through the FIR filter, which is the "position invariance" of the FIR filter [20].

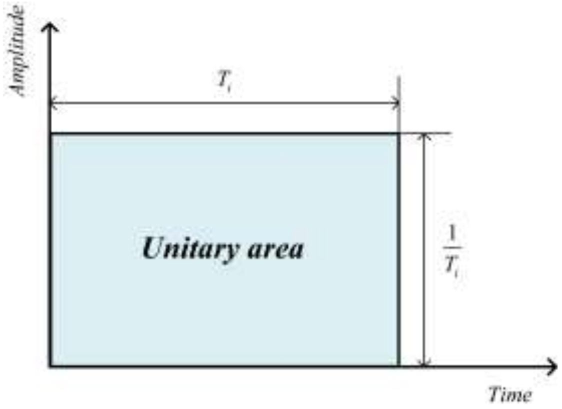


Fig.1. Impulse response of FIR filter.

2.2 Generation of S-shaped velocity curve

In 3-axis machining, for each line of G code, the start position coordinate $P_s(X_s,Y_s,Z_s)$, end position coordinate $P_E(X_E,Y_E,Z_E)$ and feed rate F can be obtained by compiling and analyzing. The speed planning of the rectangular-frame is obtained from the segment length L and the feed rate F:

$$L = \| P_E - P_S \| \quad \text{and} \quad T_v = \frac{L}{F} \tag{3}$$

Where T_v represents the duration of the rectangular frame speed pulse; When the velocity curve of the rectangular-frame is convolved with an FIR filter with a time constant of T_1, and then convoluted with an FIR filter with a time constant of T_2, the S-shaped velocity curve commonly used in CNC system can be obtained [18]. To obtain a 7 segmented S-shaped velocity curve, that is, the acceleration curve of the system is a trapezoid and its jerk is bounded, it must also be ensured that $T_v > T_1 + T_2$ and $T_2 < T_1$. At this time, the maximum acceleration and jerk of the system is:

$$a_{\max} = \frac{F}{T_1} \quad \text{and} \quad j_{\max} = \frac{F}{T_1 T_2} \tag{4}$$

After the filtering effect of these two cascaded FIR filters, the obtained speed is as follows:

$$v'(t) = v(t) * h_1(t) * h_2(t)$$
$$= \begin{cases} \frac{1}{2}\frac{F}{T_1T_2}t^2 & 0 \le t < T_2 \\ \frac{1}{2}\frac{FT_2}{T_1} + \frac{F}{T_1}(t-T_2) & T_2 \le t < T_1 \\ F - \frac{1}{2}\frac{F}{T_1T_2}((T_1+T_2)-t)^2 & T_1 \le t < T_1+T_2 \\ F & T_1+T_2 \le t < T_v \\ F - \frac{1}{2}\frac{F}{T_1T_2}(t-T_v)^2 & T_v \le t < T_v+T_2 \\ F - \frac{1}{2}\frac{FT_2}{T_1} - \frac{F}{T_1}(t-(T_v+T_2)) & T_v+T_2 \le t < T_v+T_1 \\ \frac{1}{2}\frac{F}{T_1T_2}((T_v+T_1+T_2)-t)^2 & T_v+T_1 \le t < T_v+T_1+T_2 \\ 0 & t \ge T_v+T_1+T_2 \end{cases} \quad (5)$$

The acceleration and position can be obtained by differentiating and integrating the velocity. The whole process is shown in Fig. 2.

The total processing time of the system is:

$$T = T_v + T_d = T_v + T_1 + T_2 \quad , \quad \text{where} \quad T_d = T_1 + T_2 \quad (6)$$

Where T_d is the delay time brought by the introduction of the filter, if the system introduces an nth order FIR filter, then theoretically the speed curve can achieve nth order continuity [19,27], and the delay time T_d is:

$$T_d = T_1 + \mathrm{L}\ \mathrm{L} + T_n \quad (7)$$

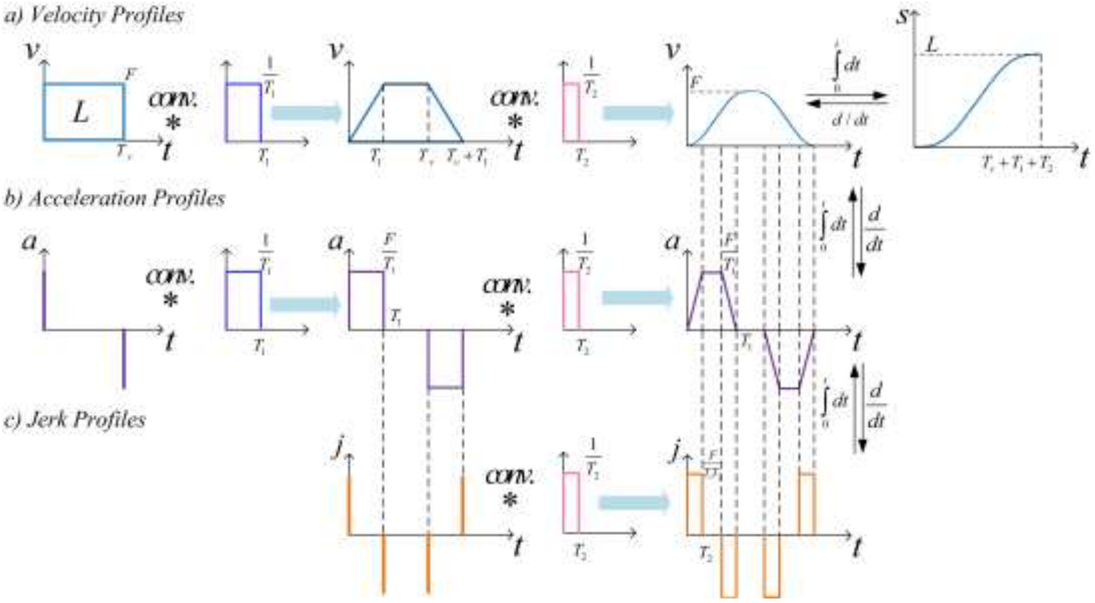


Fig.2. S-shaped speed curve generation by FIR filtering.

3 Continuous trajectory generation under contour error constraints

It has been introduced that a single-segment S-shaped velocity curve with continuous acceleration can be generated based on FIR filtering. On this basis, this chapter will further introduce the trajectory generation method for continuous tool-paths following contour control. Section 3.1 first solves the velocity proportional coefficients of the two front and rear trajectories under the constraint of contour error at a single corner for long segments. Section 3.2 extends this idea to the case of continuous micro-segments, solves the influence of multiple trajectories at a single corner under the constraint of contour error one by one, and obtains the velocity proportional coefficient of each micro-segment by synthesizing the results at multiple corners. Section 3.3 adopts pre-discretization to scatter the complex trajectory into continuous micro-segments, to satisfy the application of the algorithm in Section 3.2, and realize a universal continuous trajectory generation method based on FIR filtering.

3.1 Non-stop velocity proportional compression for long segments

As described in Section 2, the premise of obtaining the S-shaped curve of the maximum feed speed F is to ensure $T_v > T_1 > T_2$ and $T_v \geq T_d = T_1 + T_2$. Then, for continuous trajectory, by carrying out rectangular-frame speed planning (RSP), a series of rectangular-frame speeds can be obtained firstly. If the dwell time between adjacent rectangular-frame is greater than or equal to the delay time, a full-stop and an exact P2P motion can be obtained, that is, the system will continue to run the next track after each track has completely stopped, as shown in Fig. 3. At this time, there is no contour error in the system, but this movement will seriously affect the machining efficiency of the system and greatly increase the machining time. Therefore, Ref. [18] proposed the methods based on dwell time control and mixing speed control. By adjusting the dwell time between adjacent rectangular-frame (NS-DCB) or the mixing speed between adjacent rectangular-frame (NS-VCB), the system can shorten the processing time under the condition of satisfying the contour error constraint. However, as pointed out in the literature, both methods are only suitable for long segments, which require $T_v \geq T_d$.

To propose a general trajectory control algorithm based on FIR filtering, firstly for long segments, this paper proposes a non-stop velocity proportional compression algorithm (NS-VPC), which makes the system dwell time T_{dwell} as 0, then the mixing time $T_{mix} = T_d - T_{dwell} = T_d$ at the corner. At this time, the system starts to perform the trajectory interpolation of the second segment when the first segment of the trajectory has not reached the end position, resulting in mixed corner and contour errors, as shown in Fig. 4.

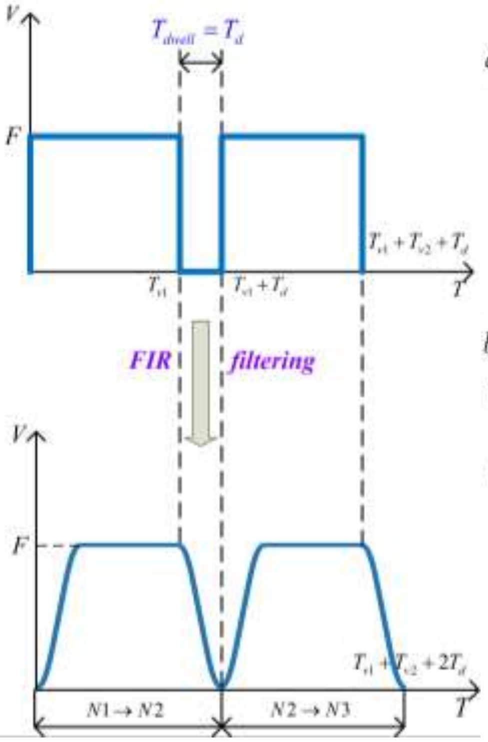


Fig.3. P2P mode by FIR filtering

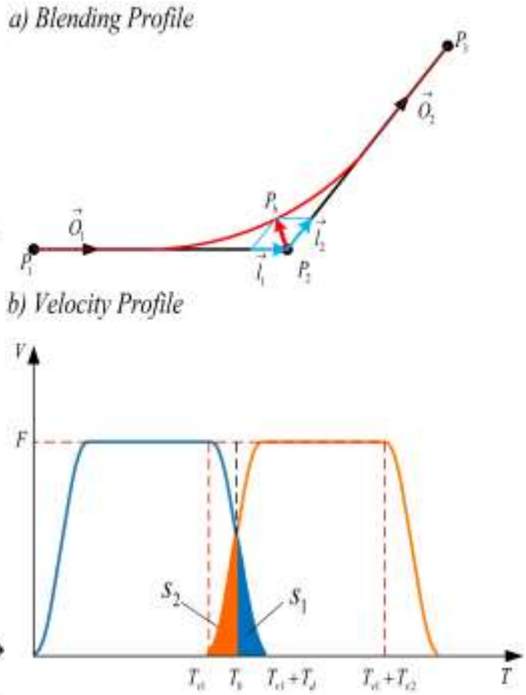


Fig.4. Blending error under long segments

Line *N1* corresponds to the starting point $P_1(X_1,Y_1,Z_1)$, line *N2* corresponds to the intermediate point $P_2(X_2,Y_2,Z_2)$, and line *N3* corresponds to the ending point $P_3(X_3,Y_3,Z_3)$. The unit vector of the N_1N_2 track segment is $\vec{O}_1$, the unit vector of the N_2N_3 track segment is $\vec{O}_2$, and it is assumed that the feed rate of the two tracks is the same, that is $F_1 = F_2 = F$. The maximum contour error of the system occurs at the angle bisector of $\angle P_1P_2P_3$ [23], the corresponding time is set to T_b, and the maximum contour error is set to ε.

Where

$$\begin{cases} T_b = T_{v1} + \frac{1}{2}T_d \\ \varepsilon = \| \overrightarrow{P_2P_b} \| = \| \vec{l}_2 - \vec{l}_1 \| \end{cases} \tag{8}$$

$\vec{l}_1$ corresponds to the vector of the post-influence area s_1 at the maximum contour error of the first track, and $\vec{l}_2$ corresponds to the vector of the pre-influence area s_2 at the maximum contour error of the second track.

From Eq. (5),

$$s_1 = s_2 = \int_0^{t=\frac{T_d}{2}} v^{'}(t)dt \tag{9}$$

$$= \begin{cases} \frac{1}{6}\frac{FT_2^{\ 2}}{T_1} + \frac{1}{2}\frac{FT_2}{T_1}(t - T_2) + \frac{1}{2}\frac{F}{T_1}(t - T_2)^2 & T_d < 2T_1 \\ \frac{1}{6}\frac{FT_2^{\ 2}}{T_1} + \frac{1}{2}\frac{FT_2}{T_1}(T_1 - T_2) + \frac{1}{2}\frac{F}{T_1}(T_1 - T_2)^2 - \frac{1}{6}\frac{F}{T_1T_2}(t - T_1)^3 + \frac{1}{2}\frac{F}{T_1}(t - T_1)^2 + \frac{1}{2}\frac{F(2T_1 - T_2)}{T_1}(t - T_1) & T_d \geq 2T_1 \end{cases}$$

Therefore,

$$\begin{cases} \vec{l_1} = s_1 \cdot \vec{O_1} \\ \vec{l_2} = s_2 \cdot \vec{O_2} \end{cases} \tag{10}$$

where

$$\begin{cases} \vec{O_1} = \frac{(X_2 - X_1, Y_2 - Y_1, Z_2 - Z_1)}{L_1} = (x_1, y_1, z_1) \\ \vec{O_2} = \frac{(X_3 - X_2, Y_3 - Y_2, Z_3 - Z_2)}{L_2} = (x_2, y_2, z_2) \end{cases} \tag{11}$$

And thus by plugging Eq. (9)-(12) into Eq. (8), ε as:

$$\varepsilon = \text{P}(s_2x_2 - s_1x_1, s_2y_2 - s_1y_1, s_2z_2 - s_1z_1)\,\text{P} \tag{12}$$

Assuming that the allowable error given by the numerical control system is ε_{allow}, the speed proportional coefficient k_v should be

$$k_v = \frac{\varepsilon_{allow}}{\varepsilon} \tag{13}$$

To satisfy the contour error of the system, the velocity of the N_1N_2 trajectory segment and the N_2N_3 trajectory segment should be adjusted as:

$$\begin{cases} v_1^{'} = k_v \cdot F_1 \\ v_2^{'} = k_v \cdot F_2 \end{cases} \tag{14}$$

In order to ensure that the length of the track segment remains unchanged, the time of these two segments should also be adjusted accordingly.

$$\begin{cases} T_{v,1}^{\ '} = \frac{L_1}{v_1^{'}} \\ T_{v,2}^{\ '} = \frac{L_2}{v_2^{'}} \end{cases} \tag{15}$$

It should be noted that the above derivation is based on $F_1 = F_2$. If $F_1 \neq F_2$, the maximum contour error at the corner is not at the angle bisector of $\angle P_1P_2P_3$, but the actual contour error is greater than the calculated value of Eq. (12), so the above algorithm can still meet the contour error constraint of the system [18].

3.2 Multi-stage mixing velocity proportional compression for micro-segments

In NC machining, CAM usually breaks up the spline into a large number of continuous tiny linear line segments, then because of $T_v < T_2 < T_1$, the NS-VPC algorithm is no longer applicable. Therefore, based on the idea of the NS-VPC algorithm in Section 3.1, this section proposes a multi-stage mixing velocity proportional compression (MM-VPC) algorithm for continuous micro-segments, which can effectively ensure that the system meets the contours. Under the premise of error constraints, a smooth transition of the angular velocity is achieved.

Due to $T_v < T_2 < T_1$, the maximum speed and acceleration that the system can reach at this time is:

$$\begin{cases} v_{\max} = \dfrac{FT_v}{T_1} \\ a_{\max} = \dfrac{FT_v}{T_1 T_2} \end{cases} \tag{16}$$

The jerk is the same as that in the case of the long segment, which is still $j_{\max} = \dfrac{F}{T_1 T_2}$.

Then, after the rectangular-frame speed passes through the second-order FIR filter, the S-shaped speed curve is shown in Fig. 5.

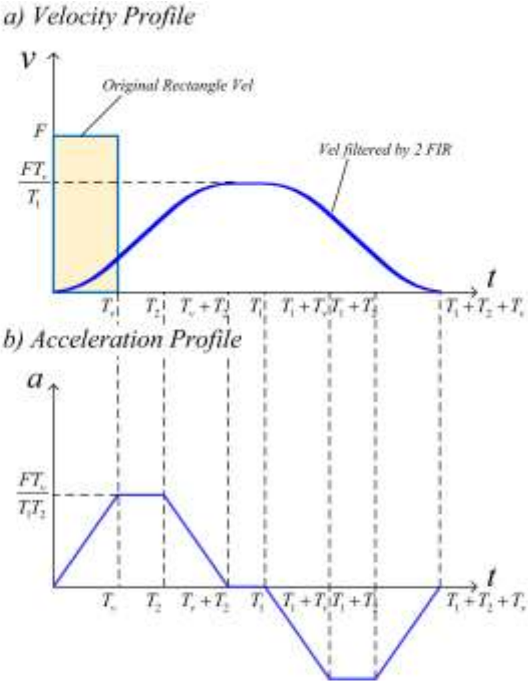


Fig.5. S-shaped velocity curve

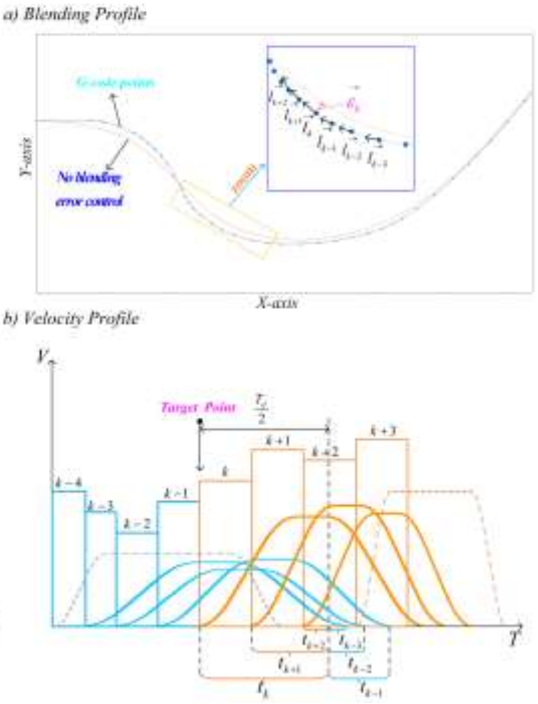


Fig.6. Blending error under micro-segments

Its speed formula is:

$$v^{''}(t)=v(t)*h_1^{'}(t)*h_2^{'}(t)$$
$$=\begin{cases}\frac{1}{2}\frac{F}{T_1T_2}t^2 & 0\leq t<T_v\\ \frac{1}{2}\frac{FT_v^{\,2}}{T_1T_2}+\frac{FT_v}{T_1T_2}(t-T_v) & T_v\leq t<T_2\\ \frac{FT_v}{T_1}-\frac{1}{2}\frac{F}{T_1T_2}((T_v+T_2)-t)^2 & T_2\leq t<T_v+T_2\\ \frac{FT_v}{T_1} & T_v+T_2\leq t<T_1\\ \frac{FT_v}{T_1}-\frac{1}{2}\frac{F}{T_1T_2}(t-T_1)^2 & T_1\leq t<T_v+T_1\\ \frac{FT_v}{T_1}-\frac{1}{2}\frac{FT_v^{\,2}}{T_1T_2}-\frac{FT_v}{T_1T_2}(t-(T_v+T_1)) & T_v+T_1\leq t<T_1+T_2\\ \frac{1}{2}\frac{F}{T_1T_2}((T_v+T_1+T_2)-t)^2 & T_1+T_2\leq t<T_v+T_1+T_2\\ 0 & t\geq T_v+T_1+T_2\end{cases} \quad (17)$$

For a continuous micro-segment trajectory, the maximum value of the contour error at the kth contour point is generated in $T_{b,k}$, and the maximum contour error ε_k is affected by the displacement vector of the rear area of the front m rectangular-frames and the front area displacement of the rear n rectangular-frames, as shown in Fig. 6.

where,

$$\begin{cases}T_{b,k}=T_k+\frac{T_d}{2}\\ \varepsilon_k=\mathrm{P}\sum_{i=0}^{n-1}\vec{l}_{k+i}-\sum_{j=0}^{m-1}\vec{l}_{k-j-1}\mathrm{P}\end{cases} \quad (18)$$

Where T_k represents the starting time of the kth rectangular-frame. Then the key of the algorithm is to solve the displacement vectors $\vec{l}_{k+i}$ and $\vec{l}_{k-j-1}$.

$$\begin{cases}\vec{l}_{k+i}=(\int_0^{t_{k+i}}v^{''}(t))\cdot\vec{O}_{k+i}\ ,\ i=0,\dots,n-1\\ \vec{l}_{k-j-1}=(L_{k-j-1}-\int_0^{t_{k-j-1}}v^{''}(t))\cdot\vec{O}_{k-j-1}\ ,\ j=0,\dots,m-1\end{cases} \quad (19)$$

Where L_{k-j-1} represents the length corresponding to the ($k-j-1$)th rectangular-frame, t_{k+i} represents the influence time of each rectangular-frame in the previous m

rectangular-frames on the contour error point, and $\overrightarrow{O_{k+i}}$ is its direction vector. t_{k-j-1} represents the influence time of each of the following n rectangular-frames on the contour error point, and $\overrightarrow{O_{k-j-1}}$ is its direction vector.

$$\begin{cases} t_{k+i} = T_{b,k} - T_{k+i},\ i = 0,\dots,n-1 \\ t_{k-j-1} = T_{k-j-1} + T_{v,k-j-1} + T_d - T_{b,k},\ j = 0,\dots,m-1 \end{cases} \tag{20}$$

By substituting Eq. (19) and Eq. (20) into Eq. (18), the contour error ε_k of the system at the k th contour point can be obtained, and the speed proportion coefficient is:

$$k_{v,k} = \frac{\varepsilon_{allow,k}}{\varepsilon_k} \tag{21}$$

Therefore, the velocity scale coefficient of the $m+n$ rectangular-frames at the k th contour point is $k_{v,k}$. Because each rectangular-frame may affect more than one corner, it may have multiple speed scale factors. In order to ensure that the contour error of each contour point meets the constraint, the minimum value is taken as the final speed proportion coefficient among the multiple speed proportion coefficients of each rectangular-frame,

$$k_v^{'} = \min\{k_{v,0},\cdots,k_{v,k},\cdots,k_{v,n}\} \tag{22}$$

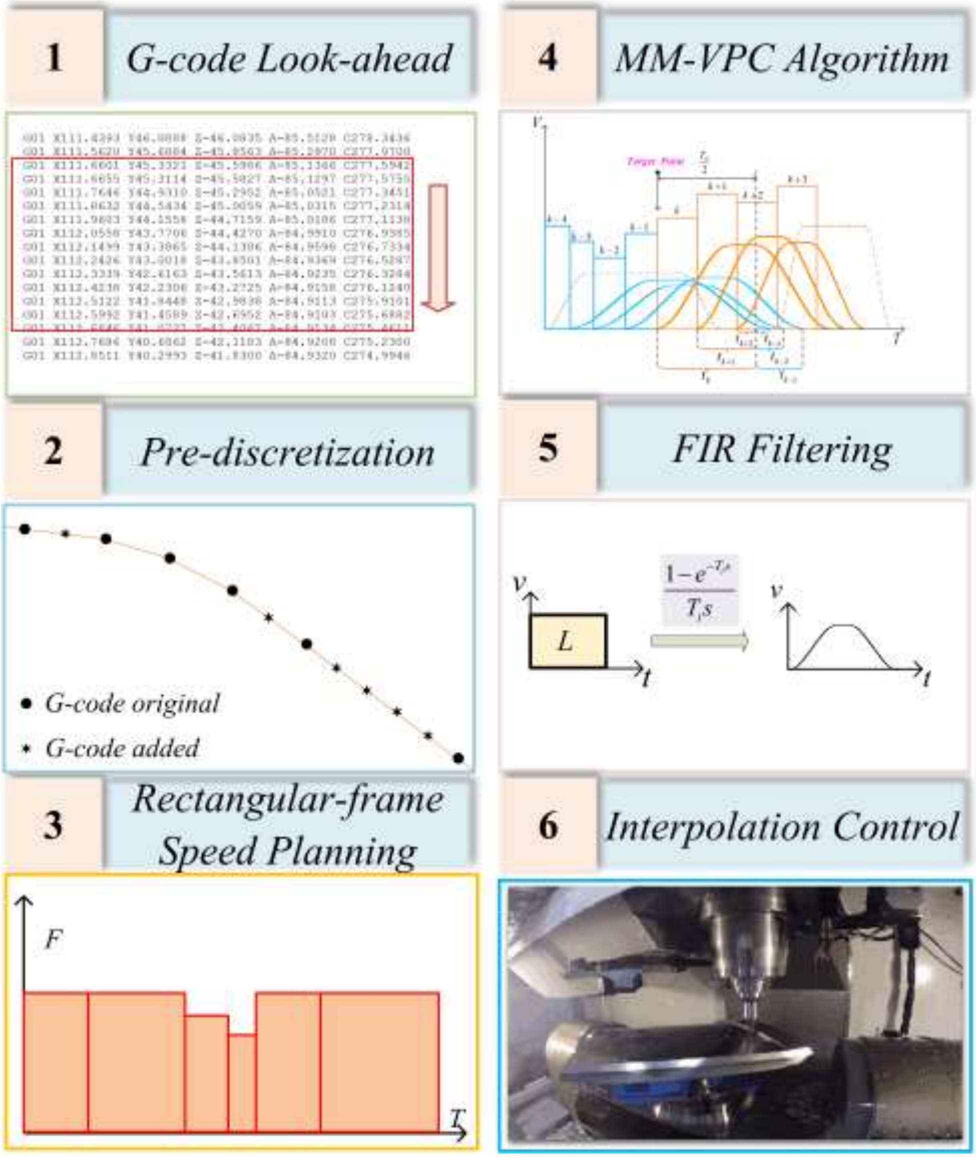
Its speed and duration are also adjusted accordingly as:

$$\begin{cases} F^{'} = k_v^{'} \cdot F \\ T_v^{'} = \dfrac{L}{F^{'}} \end{cases} \tag{23}$$

3.3 Pre-discrete velocity proportional compression algorithm for blending tool- paths

In actual processing, most G codes are mixed paths with both long segments and micro-segments. In this case, neither the NS-VPC nor MM-VPC algorithms are applicable anymore. Therefore, this section proposes a pre-discrete velocity proportional compression algorithm (PD-VPC) for mixed paths, which can meet the needs of various machining paths and improve machining efficiency on the premise of effectively ensuring contour error constraints.

The key of PD-VPC algorithm is to pre-discrete in advance, break the long G codes into micro-segments based on certain rules, and then put the broken G code into MM-VPC algorithm to realize speed adjustment and interpolation. The algorithm flow is shown in Fig. 7.



```
G01 X111.4393 Y46.0888 Z-46.0835 A-85.5128 C278.3436
G01 X111.5620 Y45.6884 Z-45.8563 A-85.2870 C277.9700
G01 X111.6601 Y45.3321 Z-45.5986 A-85.1366 C277.5942
G01 X111.6655 Y45.3114 Z-45.5827 A-85.1297 C277.5756
G01 X111.7646 Y44.9310 Z-45.2952 A-85.0521 C277.3451
G01 X111.8632 Y44.5434 Z-45.0059 A-85.0316 C277.2318
G01 X111.9603 Y44.1558 Z-44.7159 A-85.0186 C277.1130
G01 X112.0558 Y43.7708 Z-44.4270 A-84.9910 C276.9385
G01 X112.1499 Y43.3865 Z-44.1386 A-84.9590 C276.7334
G01 X112.2426 Y43.0018 Z-43.8501 A-84.9369 C276.5287
G01 X112.3339 Y42.6163 Z-43.5613 A-84.9235 C276.3284
G01 X112.4238 Y42.2306 Z-43.2725 A-84.9158 C276.1240
G01 X112.5122 Y41.8448 Z-42.9838 A-84.9113 C275.9161
G01 X112.5992 Y41.4589 Z-42.6952 A-84.9103 C275.6882
G01 X112.6846 Y41.0727 Z-42.4067 A-84.9134 C275.4611
G01 X112.7686 Y40.6862 Z-42.1183 A-84.9208 C275.2300
G01 X112.8511 Y40.2993 Z-41.8300 A-84.9320 C274.9948
```

Fig.7. PD-SPC algorithm process

The premise of the MM-VPC algorithm is to satisfy $T_v < T_2 < T_1$, the critical condition is $T_v = T_2$, and the system acceleration can just reach the maximum value at this time, so the minimum time for segmentation is $T_{div,\min} = T_2$. Firstly, read the entire G code in advance, carry out the speed planning of the rectangular-frame, then mark the line segment of $T_v \geq T_{div,\min}$ as a long segment, and then perform linear interpolation on the long segment and discretize it into micro-segments. In order to break up the long segment into a certain number of micro-segments, a discrete coefficient k_s is introduced, and the duration of each micro-segment is denoted as $T_{v,s}$,

$$T_{v,s} = k_s \cdot T_2 \text{ , } k_s \in (0,1) \tag{24}$$

It should be noted that the value of k_s will not affect the actual running time and acceleration/deceleration trend of G code trajectory, but only to break the long segment into micro segments for application to MM-VPC algorithm. However, the value of k_s will directly affect the number of broken segments, thus affecting the operation efficiency of the algorithm.

Assuming that the running time of a long segment of G code is denoted as T_i, the coordinates of the starting point are $P_{i-1}(X_{i-1},Y_{i-1},Z_{i-1})$, and the coordinates of the end point are $P_i(X_i,Y_i,Z_i)$, then the number of short segments that will be broken up into

$$N = ceil(\frac{T_i}{T_{v,s}}) \text{ , } N \in N^* \text{ and } N \geq 2 \tag{25}$$

The $ceil$ function means rounding up; Through linear interpolation, the newly added intermediate position G code point $P_n = (X_n,Y_n,Z_n)$ can be obtained, and the coordinates of each axis are solved as follows, where $n = 1,2,...,N-1$.

$$\begin{aligned} X_n &= X_{i-1} + (X_i - X_{i-1}) * \frac{n}{N} \\ Y_n &= Y_{i-1} + (Y_i - Y_{i-1}) * \frac{n}{N} \\ Z_n &= Z_{i-1} + (Z_i - Z_{i-1}) * \frac{n}{N} \end{aligned} \tag{26}$$

4 Experimental results

In this section, an actual 5-axis machine tool will be used for practical verification in combination with the interpolation point data generated by the simulation experiment. The experimental machine tool is shown in Fig. 8. The machine tool adopts five axis full closed-loop control and supports locking the rotating axes. The maximum feed speed of the linear axis is 48000 [mm / min], and the control resolution is 0.001 [mm]. In this section, the rotating axes are locked in the whole experiment, and the closed-loop sampling period of the servo control system is set to 1 [kHz].

Fig.8. Experimental machine tool

Mercedes-Benz mold testing sample is the standard test sample of 3-axis milling in the industry [1]. It is recognized by many countries in Europe and the United States and is widely used to evaluate the machining efficiency and accuracy of NC machine tools. The surface of the specimen contains a variety of complex tool-path features such as lines, curves and surfaces. In this section, a section of line cutting tool-path in the finishing stage is taken as the machining path, as shown in Fig. 9, to verify the feasibility and effectiveness of the algorithm. The red point represents the command position of G code, and the blue line is the tool-path formed by the command position. It can be seen that the tool-path is composed of long segments and continuous micro-segments, and the trajectory contour includes lines, curves, sharp corners and rounded corners. This multi-feature complex tool-path is representative in practical NC machining.

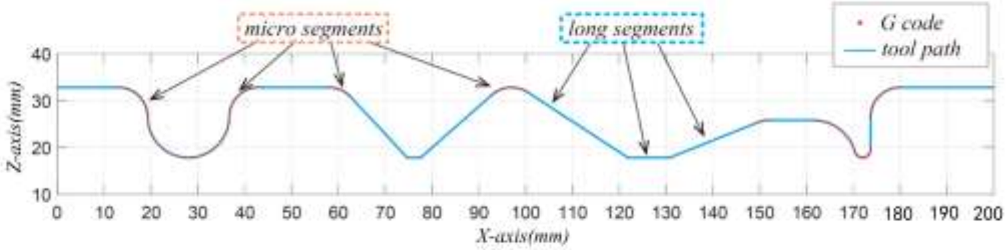


Fig.9. The line cutting of Benz pattern

The line cutting trajectory includes 233 linear segments, of which the shortest path length is 0.021 [mm] and the longest path length is 29.078 [mm]. The given feed rate during machining is 3000 [mm/min], and the set contour error is 10 [μm]. In order to make a fair and comprehensive comparison, this paper will use the following FIR filter interpolation algorithms for the line cutting trajectory: (1) P2P mode, that is, the dwell time is equal to the delay time; (2) The NS-DCB algorithm proposed in Ref. [10] controls the dwell time according to the contour error. (3) The PD-VPC algorithm proposed in this paper; (4) Velocity directly blending mode, whose dwell time is zero, hereinafter referred to as VDB mode. It is worth mentioning that the NS-VCB algorithm proposed in Ref. [23] is only applicable to long segments, so it does not participate in the comparative experiment.

For the above three algorithms, the same filter constant must be used. In this paper, $T_1 = 20[ms]$ and $T_2 = 10[ms]$ are set. The PD-VPC algorithm needs to discretize the G code points in advance. Through the simulation of the line cutting of Benz sample, the running time T_{run} of the algorithm and the total interpolation time T_{interp} of G code under different k_s are obtained, as shown in Table. 1.

Table.1 Running time and total interpolation time with different parameters

	k_s	T_{run} [s]	T_{interp} [ms]
1	1	0.086	6426
2	0.75	0.093	5770
3	0.5	0.136	5669
4	0.25	0.249	5666
5	0.1	1.004	5662

It can be seen from the table that when the value of k_s approaches 1, the running time of the algorithm is short, but because the number of discrete segments of the system is small, the speed in some areas is depressed and the duration is long, so the total interpolation time of G code is long; When the value of k_s approaches 0, the total interpolation time of G code becomes slightly shorter, but the running time of the algorithm increases significantly due to the excessive number of discrete segments of the system. Therefore, $k_s \in [0.5, 0.75]$ is usually taken, so the running time of the algorithm and the total interpolation time of the G code are more appropriate. In this paper, the discretization coefficient k_s is selected as 0.5, and the G code track before and after discretization is shown in Fig. 10 (a).

The blue point is the original G code point, and the orange point is the discrete G code point. After discretization, the number of track segments is 799, and the length of the shortest segment remains unchanged because this segment is not discretized; The longest segment length is 0.503 [mm], and the duration of its rectangular-frame speed can be calculated to be 10.12 [ms], which is less than the delay time of the second-order filter and meets the input requirements of MM-VPC algorithm. Fig. 10 (b) shows the distribution of trajectory velocity on the interpolation contour obtained by PD-VPC algorithm. RGB three primary colors are selected to represent the speed, red represents high speed and light color represents low speed. It can be seen from the figure that the result of velocity planning conforms to the geometric characteristics of the trajectory.

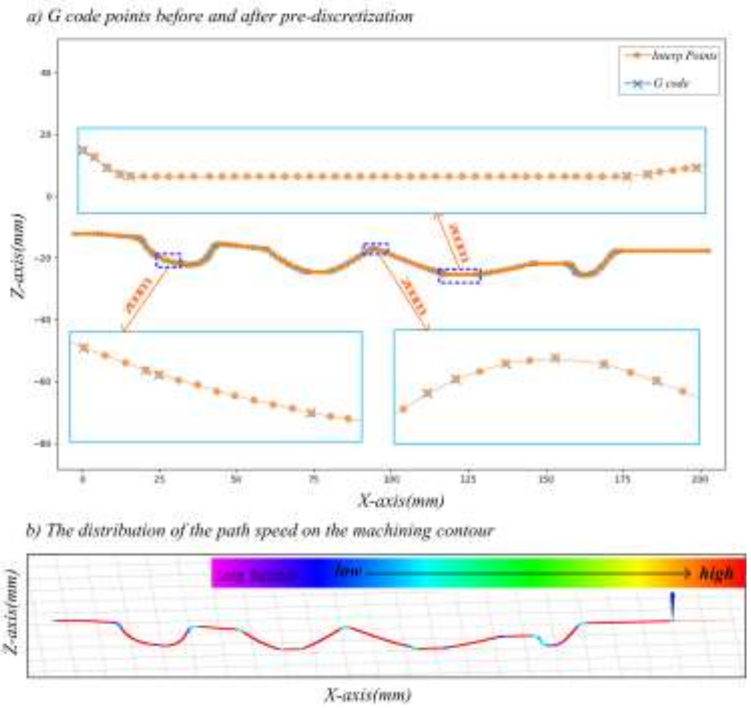


Fig.10. G-code pre-discretization and velocity distribution on the contour

Fig. 11 shows the machining contour calculated by interpolation with four algorithms, in which the blue point represents the G code point of Benz sample. As can be seen from the figure, P2P mode performs point-to-point motion, so it completely coincides with the G code point trajectory. The VDB mode directly mixes the speed without considering the contour error constraint, so there is an obvious out of tolerance at the corner. The PD-VPC algorithm proposed in this paper can well meet the contour error constraints given by the system, and the contour error is smaller than NS-DCB algorithm.

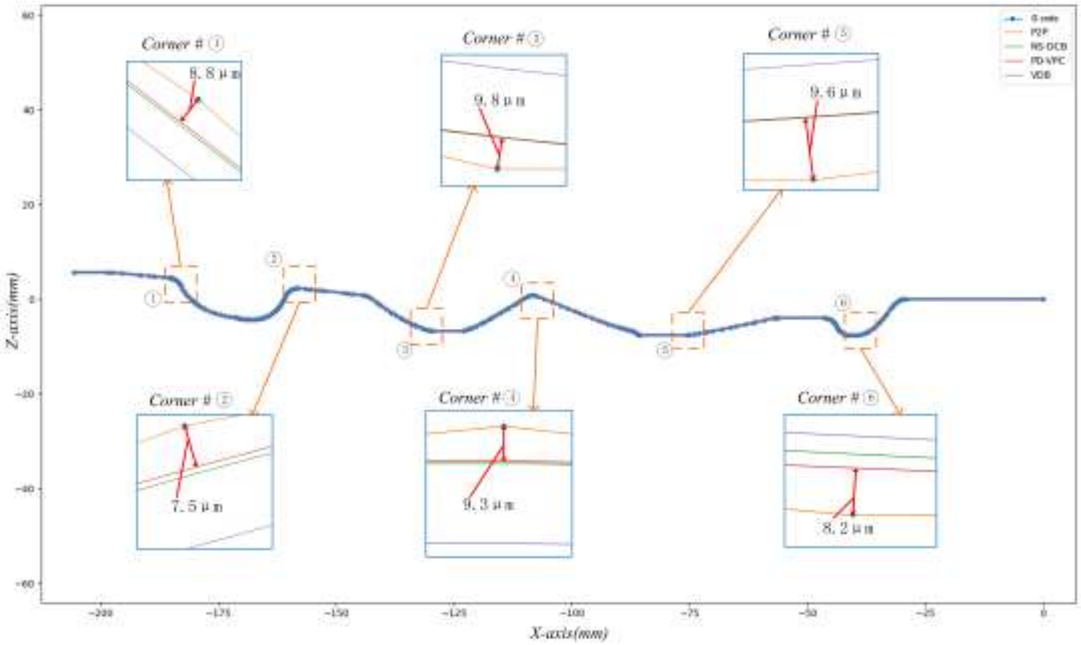


Fig.11. Benz pattern shaped tool-path

The composite velocity, acceleration and jerk of the four algorithms are shown in Fig. 12. As can be seen from the figure, there is no dwell time in VDB mode, so the processing time is the shortest, only 4.47 [s]. In P2P mode, due to frequent acceleration and deceleration, the processing time is the longest, which is 11.43 [s]. Although the processing time of the PD-VPC algorithm is slightly longer than that of the NS-DCB algorithm, it can be seen from Fig. 12(b) and Fig. 12(c) that the PD-VPC algorithm effectively reduces the acceleration and jerk amplitude of the system. It can be inferred that the system vibration will be reduced and the surface quality will be improved. Since the starting and ending of the profiles correspond to the feed section and withdrawal section of the machining, it is not considered when comparing the acceleration and jerk amplitude.

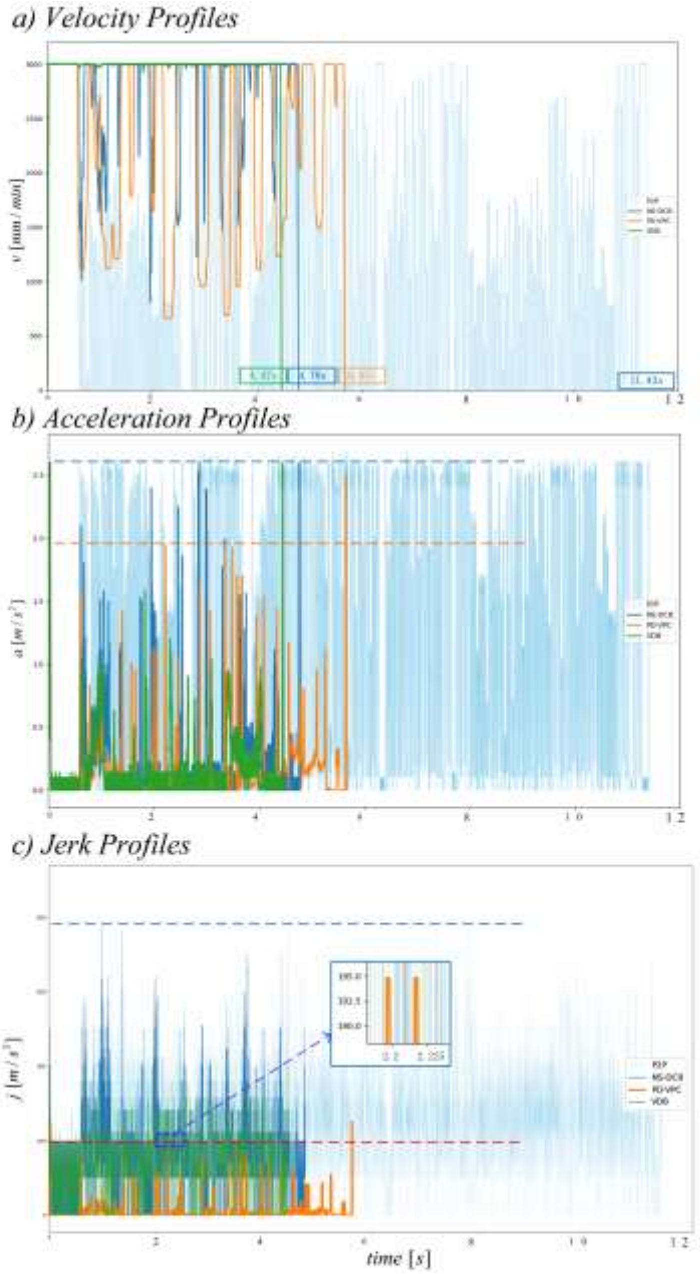


Fig.12. Interpolated kinematic profiles along Benz tool-path

Fig. 13 shows the single-axis kinematics parameters for the X-axis and Z-axis of the system. It can be seen from Fig. 13(b) and Fig. 13(c) that the motion trajectory acceleration generated by the PD-VPC algorithm is continuous and the jerk is bounded, and it can effectively reduce the uniaxial vibration of the system. Due to the trajectory characteristics of the Benz tool-path, the performance on each axis is also different. The acceleration amplitude of X-axis is reduced by 18% and the jerk amplitude is reduced by 73%. The acceleration amplitude of Z axis is reduced by 51% and the jerk amplitude is reduced by 82%. Conclusively, the acceleration amplitude of each axis is reduced to a certain extent, and the jerk amplitude is greatly reduced.

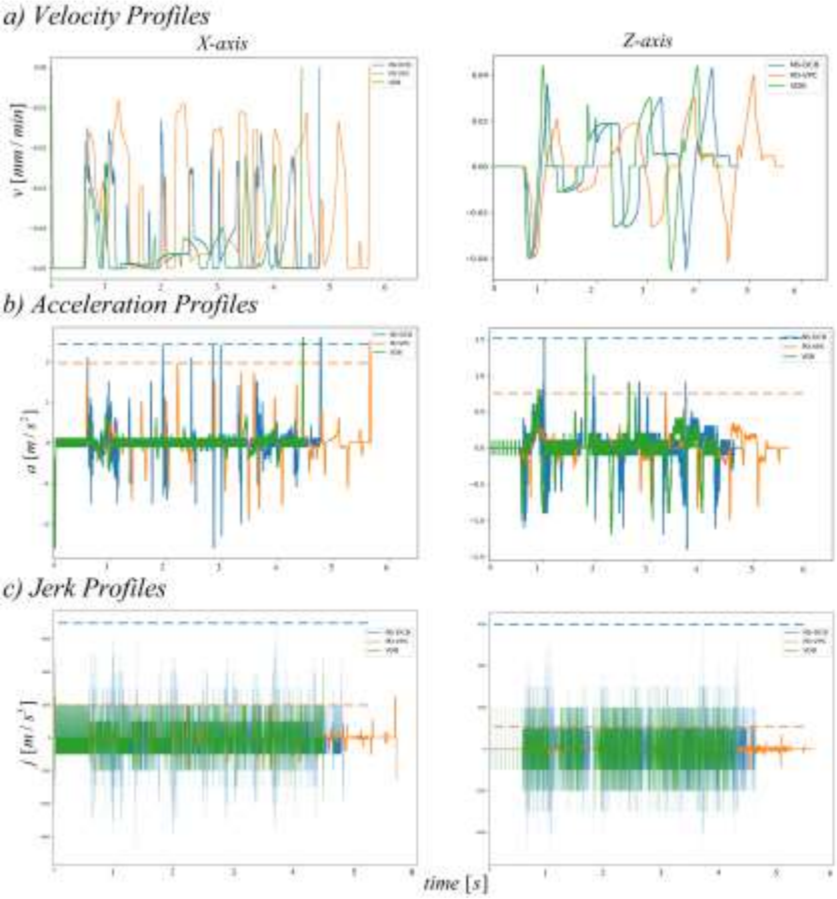


Fig.13. X-axis and Z-axis Interpolated kinematic profiles

Finally, the 3-axis contouring performance for the interpolation schemes are measured in Fig. 14. P2P motion performs precise motion, so there are no contour errors. The PD-VPC algorithm proposed in this paper complies with the set contour error requirement of 10 [μm] in all processing areas, while NS-DCB has out of tolerance phenomenon in local areas. Therefore, the NS-DCB algorithm requires a larger acceleration, which also echoes the results in Fig. 12 and Fig. 13.

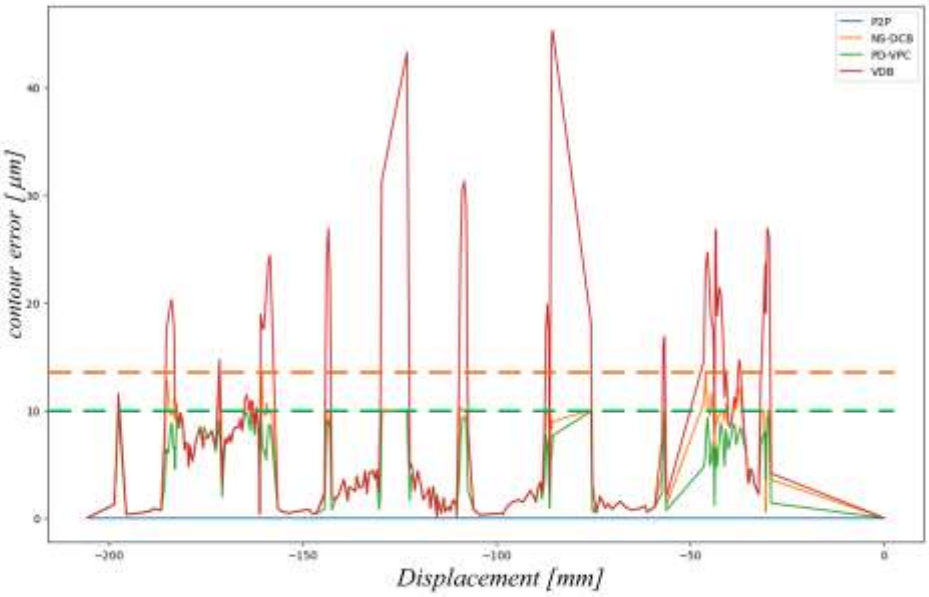


Fig.14. Experimental contouring errors

5 Conclusions

This paper presents a universal algorithm for blending tool-paths machining of CNC system based on FIR filtering. The results show that the algorithm can control the contour error effectively, reduce system vibration and improve machining quality. The existing algorithms mainly focus on the smooth trajectory generation for long segments, and few studies for continuous micro-segments only adopt the trapezoidal acceleration/deceleration scheme when calculating the corner contour error. In this paper, a general algorithm for generating continuous acceleration trajectory based on FIR filtering is proposed, which can not only be applied to the blending tool-paths with long segments and micro-segments, but also effectively control the contour error. Experimental results show that the overall processing time can be reduced up to 50% as compared to the P2P mode. On the other hand, the acceleration amplitude of the single axis can be reduced up to 10%-50% and the jerk amplitude can be reduced up to 70%-80%. So the algorithm can effectively suppress the vibration in the processing process and improve machining surface quality.

References

[1] Y. Altintas, A. Verl, C. Brecher, L. Uriarte, G. Pritschow, Machine tool feed drives, CIRP Ann. - Manuf. Technol. 60 (2011) 779–796, https://doi.org/10.1016/j.cirp.2011.05.010.

[2] B. R. Li, H. Zhang, P. Q. Ye, J. S. Wang, Trajectory smoothing method using reinforcement learning for computer numerical control machine tools, Robot. Comput. Integr. Manuf. 61 (2020), https://doi.org/10.1016/j.rcim.2019.101847.

[3] B. Sencer, Y. Altintas, E. Croft, Feed optimization for five-axis CNC machine tools with drive constraints, Int. J. Mach. Tool Manufact. 48 (2008) 733–745, https://doi.org/10.1016/j. ijmachtools.2008.01.002.

[4] X. Beudaert, S. Lavernhe, C. Tournier, 5-Axis local corner rounding of linear tool path discontinuities, Int. J. Mach. Tool Manufact. 73 (2013) 9–16, https://doi.org/10.1016/j.ijmachtools. 2013.05.008.

[5] Y. Zhang, P. Q. Ye, J. Q. Wu, et al, An optimal curvature-smooth transition algorithm with axis jerk limitations along linear segments, Int. J. Adv. Manuf. Technol. 95 (2018) 875−888 https://doi.org/10.1007/s00170-017-1274-1.

[6] H. Zhao, L. M. Zhu, H. Ding, A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments, Int. J. Mach. Tool Manufact. 65 (2013) 88-98, https://doi.org/10.1016/j.ijmachtools.2012.10.005.

[7] B. Sencer, K. Ishizaki, E. Shamoto, A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths, Int. J. Adv. Manuf. Technol. 76 (2015) 1977-1992, https://doi.org/10.1007/s00170-014-6386-2.

[8] C. A. Ernesto, R.T. Farouki, High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error, Int. J. Adv. Manuf. Technol. 58 (2012) 327–338, https://doi.org/10.1007/s00170-011-3394-3.

[9] H. Yau, J. Wang, Fast Bezier interpolator with real-time lookahead function for high-accuracy machining, Int. J. Mach. Tool Manufact. 47 (2007) 1518−1529, https://doi.org/10.1016/j. ijmachtools.2006.11.010.

[10] H. Park, B-spline surface fitting based on adaptive knot placement using dominant columns. Comput.-Aided Des. 43 (2011) 258−264, https://doi.org/10.1016/j.cad.2010.12.001.

[11] M. Tsai, H. Nien, H. Yau, Development of a real-time look-ahead interpolation methodology with spline-fitting technique for high-speed machining, Int. J. Adv. Manuf. Technol. 47 (2010) 621−638, https://doi.org/10.1007/s00170-009-2220-7.

[12] K. Erkorkmaz, Y. Altintas, High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation, Int. J. Mach. Tool Manufact. 41 (2001) 1323-1345, https://doi.org/10.1016/S0890-6955(01)00002-5.

[13] C. W. Ha, K. H. Rew, K. S. Kim, A complete solution to asymmetric S-curve motion profile: Theory & experiments, Int. Conf. Control Autom. Syst. (2008) 2845-2849, https://doi.org/ 10.1109/ICCAS.2008.4694244.

[14] S. Nam, M. Yang, A study on a generalized parametric interpolator with real-time jerk-limited acceleration, Comput.-Aided Des. 36 (2004) 27-36, https://doi.org/10.1016/S0010- 4485(03) 00066-6.

[15] A. Y. Lee, Y. J. Choi, Smooth trajectory planning methods using physical limits, Proc. Inst. Mech. Eng., Part C. 229 (2015) 2127-2143, https://doi.org/10.1177/0954406214553982.

[16] J. Huang, L. M. Zhu, Feedrate scheduling for interpolation of parametric tool path using the sine series representation of jerk profile, Proc. Inst. Mech. Eng., Part B. 231 (2016) 2359-2371, https://doi.org/10.1177/0954405416629588.

[17] Y. S. Wang, D. S. Yang, R. L. Gai, et al, Design of trigonometric velocity scheduling

algorithm based on pre-interpolation and look-ahead interpolation, Int. J. Mach. Tool Manufact. 96 (2015) 94-105, https://doi.org/10.1016/j.ijmachtools.2015.06.009.

[18] S. Fang, S. H. Hao, M. H. Hao, et al, Research on Acceleration and Deceleration Control Algorithm of NC Instruction Interpretations with High-Order Smooth, Intell. Robot. Appl. 5315 (2008), https://doi.org/10.1007/978-3-540-88518-4_59.

[19] B. Sencer, K. Ishizaki, E. Shamoto, High speed cornering strategy with confined contour error and vibration suppression for CNC machine tools, CIRP Ann. - Manuf. Technol. 64 (2015) 369-372, https://doi.org/10.1016/j.cirp.2015.04.102.

[20] Y. Liu, M. Wan, X. B. Qin, et al, FIR filter-based continuous interpolation of G01 commands with bounded axial and tangential kinematics in industrial five-axis machine tools, Int. J. Mech. Sci. 169 (2020), https://doi.org/10.1016/j.ijmecsci.2019.105325.

[21] P. Besset, R. Béarée, FIR filter-based online jerk-constrained trajectory generation, Control Engineering Practice. 66 (2017) 169-180, https://doi.org/10.1016/j.conengprac.2017.06.015.

[22] D. D. Li, L. Q. Zhang, L. Yang, et al, Accurate interpolation and contouring control algorithm based on FIR filters for the corner transition, Int. J. Adv. Manuf. Technol. 109 (2020), 1775–1788, https://doi.org/10.1007/s00170-020-05491-8

[23] S. Tajima, B. Sencer, Accurate real-time interpolation of 5-axis tool-paths with local corner smoothing, Int. J. Mach. Tool Manufact. 142 (2019) 1-15, https://doi.org/10.1016/j.ijmachtools.2019.04.005.

[24] T. Hayasaka, K. Minoura, K. Ishizaki, et al, A lightweight interpolation algorithm for short-segmented machining tool paths to realize vibration avoidance, high accuracy, and short machining time, Precision Engineering. 59 (2019) 1-17, https://doi.org/10.1016/j.precisioneng.2019.05.006.

[25] S. Tajima, B. Sencer, Real-time trajectory generation for 5-axis machine tools with singularity avoidance, CIRP Ann. - Manuf. Technol. 69 (2020) 349-352, https://doi.org/10.1016/j.cirp.2020.04.050.

[26] S. Tajima, B. Sencer, Online interpolation of 5-axis machining toolpaths with global blenging, Int. J. Mach. Tool Manufact. 175 (2020), https://doi.org/10.1016/j.ijmachtools.2022.103862.

[27] S. Tajima, B. Sencer, E. Shamoto, Accurate interpolation of machining tool-paths based on FIR filtering, Precision Engineering. 52(2018) 332-344, https://doi.org/10.1016/j.precisioneng.2018.01.016.