Int J Adv Manuf Technol (2017) 89:941–956
DOI 10.1007/s00170-016-9144-9

ORIGINAL ARTICLE

Minimum time corner transition algorithm with confined feedrate and axial acceleration for nc machining along linear tool path

Qiang Zhang[1] · Xiao-Shan Gao[1] · Hong-Bo Li[1] · Ming-Yong Zhao[1]

Received: 7 March 2016 / Accepted: 4 July 2016 / Published online: 15 July 2016

Abstract Linear tool paths, or G01 codes, are the most widely used form of tool paths for NC machining. Because of the inborn tangent discontinuity, it is difficult to realize the machining with both high speed and high quality along the G01 codes. To solve this problem, a local corner transition algorithm combined with global motion planning strategy is proposed. Very different from recent corner transition studies in which the transition path and feedrate are planned separately, this work uses a one-step strategy to generate the transition trajectory. Besides, the more reasonable axial acceleration limits are considered in this algorithm, since the acceleration performances of individual axes in conventional machine tools may be significantly different. To enhance the machining efficiency, a modified corner transition strategy with more free transition space is proposed. The feasibility and efficiency of the proposed algorithm are verified by simulations and experiments.

Keywords Interpolation · Linear tool path · Corner transition · Axial acceleration · NC machines

1 Introduction

In NC technology, the interpolation has been studied extensively because the productivity and machining precision can be improved by developing advanced interpolation algorithms. Recently, following the development of modern CAD technology, the interpolation algorithms of parametric spline path are widely studied, such as the real-time methods [1–3] and the off line optimization methods [4–8]. Smooth and fast motion is expected to be obtained by using these algorithms. However, the output of NC codes in parametric format is not popular in most CAD/CAM software. Moreover, the problems of accurate computation of curve lengths [9] and suppression of feed fluctuations [10] are difficult yet must be solved. Therefore, it is worthwhile to further improve the performance of G01 interpolation. The main difficulty of G01 interpolation is caused by the tangency discontinuities at the command points, which result in low motion speed and poor machining quality.

There exist two major approaches for G01 interpolation: the corner transition methods and the global curve fitting methods. The corner transition or corner smoothing methods change the tool path around the corner points locally, allowing smooth transition at the corner points. The global curve fitting methods fit the whole G01 tool path with certain parametric curve and then design the feedrate along the parametric curve, refer to [11–14]. Due to the locality of planning process, the corner transition methods are easy to implement and hence are more widely used in practical CNC machining, and is also the focus of this paper.

In corner transition methods, by employing certain parametric curves to blend adjacent linear segments, higher feedrate can be allowed when the tool passes through the corner points. Zhang et al. [15] applied cubic polynomial spline to blend linear segments. Zhao et al. [16] applied B-spline curve with five control points as the transition curve and obtained curvature-continuous blended path. In [17], circular arc was used to blend linear segments and contour error model was induced to control the contour error in

✉ Xiao-Shan Gao
xgao@mmrc.iss.ac.cn

[1] KLMM, UCAS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190, China

real-time. Sencer et al. [18] used quartic B-splines to blend adjacent straight lines and optimized the curvature of the transition path to deliver short cycle time. Sun et al. [19] proposed a dynamic B-spline transition scheme for short line segments and also optimized the transition curve to reduce machining time. Fan et al. [20] used quartic Bezier curve to smooth the tool path and achieved G3 continuity. The general seven-phase jerk-limited look-ahead algorithms were applied in all the above papers to realize the feedrate update, except [20] in which a 15-phase jounce-limited scheme was used, and [18] in which a jerk-limited S-acceleration scheme was adapted.

Although smooth feedrate can be obtained with the above methods, the following drawbacks still exist in these methods. Firstly, the calculation of curve lengths is indispensable in these methods while the accurate computation of curve lengths is difficult. Secondly, in these interpolation algorithms, only tangential kinematics limits are considered for look-ahead feedrate planning. According to Dong et al. [21], the individual axes of a conventional machine tool may have significantly different dynamic performance characteristics. Hence considering tangential kinematics limits may result in either infeasible acceleration demands or overly conservative planning decisions. Thirdly, these interpolation algorithms adopt a two steps process, by first generating the geometric path and then the feedrate along the path, which complicates the solution process and is not time-optimal.

In Sencer et al. [22] and Zhang et al. [23], one-step corner trajectory planning algorithms were proposed. In [22], the FIR (finite impulse response) filtering technology was used, where the optimality of the feedrate planning was not considered. In [23], the error tolerance was not fully used and the designed feedrate was also not optimal.

The purpose of this paper is to give a corner transition strategy and its corresponding motion planning algorithm to minimize the total machining time in certain sense. Different from most recent corner transition studies where transition path and feedrate are planned separately, this work uses a one-step transition trajectory generation strategy where the transition trajectory is a parametric quadratic curve in the time parameter, and is guaranteed to be time-optimal in certain sense. Besides, the reasonable axial acceleration limits are considered in this algorithm. Also, a new corner transition model is proposed to fully use the corner tolerance. The transition path is a time parameterized quadratic curve that is determined by the maximal axial accelerations and the maximal error tolerance.

The paper is organized as follows. In Section 2, the corner transition model is presented. In Section 3, the minimum time motion algorithm is introduced, which includes the corner classification and treatment, the transition trajectory construction within different type corners, and the total interpolation flow with look-ahead scheme. Section 4 presents the simulations and experiments. The conclusions are drawn in Section 5.

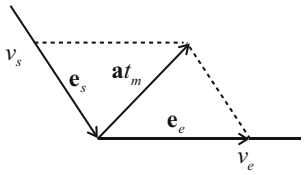


Fig. 1 Kinematic description of the corner transition

2 Design of corner transition trajectory

2.1 Constant acceleration transition

According to Fig. 1, for a given constant acceleration vector **a**, the corner transition velocity satisfies the following relation

$$v_e\mathbf{e}_e - v_s\mathbf{e}_s = \mathbf{a}T, \tag{1}$$

where T denotes the corner transition time, v_s denotes the starting velocity of the transition trajectory along the velocity direction $\mathbf{e}_s$, and v_e denotes the ending velocity along the velocity direction $\mathbf{e}_e$, and $\|\mathbf{e}_s\| = 1$, $\|\mathbf{e}_e\| = 1$. The acceleration vector **a** can be written as

$$\mathbf{a} = a_e\mathbf{e}_e - a_s\mathbf{e}_s \tag{2}$$

with $a_s > 0$ and $a_e > 0$. Then we have

$$v_s = a_sT,\ v_e = a_eT. \tag{3}$$

In Fig. 2, let **P** be the corner point, **S** the start point of the transition trajectory, **E** the end point, and **Q** an arbitrary point on the trajectory. Then the trajectory is

$$(\mathbf{Q} - \mathbf{P}) = (\mathbf{S} - \mathbf{P}) + v_st\mathbf{e}_s + \frac{1}{2}\mathbf{a}t^2. \tag{4}$$

Let $l_s = |\mathbf{SP}|$ and $l_e = |\mathbf{PE}|$. Then $\mathbf{P} - \mathbf{S} = l_s\mathbf{e}_s$ and $\mathbf{E} - \mathbf{P} = l_e\mathbf{e}_e$. We have

$$l_s\mathbf{e}_s + l_e\mathbf{e}_e = v_sT\mathbf{e}_s + \frac{1}{2}\mathbf{a}T^2. \tag{5}$$

From the above equation, we can obtain

$$l_s = \frac{1}{2}T^2a_s,\ l_e = \frac{1}{2}T^2a_e. \tag{6}$$

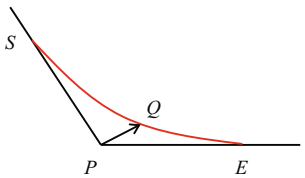


Fig. 2 The transition trajectory of the corner

Define a relative motion trajectory as $\mathbf{C}(t) = \mathbf{Q}(t) - \mathbf{P}$, then

$$\mathbf{C}(t) = -a_s \mathbf{e}_s \frac{(T-t)^2}{2} + a_e \mathbf{e}_e \frac{t^2}{2}. \tag{7}$$

2.2 Acceleration direction and transition time

We will show how to determine the optimal acceleration vector **a**. According to Eq. 2, the direction of acceleration **a** should between $\mathbf{e}_e$ and $(-\mathbf{e}_s)$, that is, inside the angle ∠**SPE** in Fig. 2. With the acceleration direction, the acceleration amplitude can be determined by

$$a_B = \min\left(\frac{A_x}{\cos(\theta_x)}, \frac{A_y}{\cos(\theta_y)}\right),$$

where A_x and A_y are the maximum accelerations along x axis and y axis, respectively. θ_x and θ_y are the corresponding angles of $\mathbf{e}_a$ w.r.t $\mathbf{e}_x$ and $\mathbf{e}_y$ respectively.

To optimize the machine performance, the amplitude and direction of corner acceleration should be determined appropriately. Here we consider two optimization indexes. One is to minimize the transition time T. Another is to maximize the minimum value of v_s and v_e.

For the problem of minimum transition time T, assume that the total transition length of the corner is a constant, i.e.,

$$l_c = l_s + l_e = \frac{1}{2}T^2(a_e + a_s).$$

Then the minimum of corner transition time T can be realized by maximizing the following function.

$$\max J = \mathbf{a} \times (\mathbf{e}_e + \mathbf{e}_s). \tag{8}$$

Since the acceleration vector **a** can be expressed as a linear combination of $(A_x\mathbf{e}_x, A_y\mathbf{e}_y)$, the optimal acceleration can be found at the vertex of the feasible acceleration polygon, see Fig. 3a–d.

However, the local optimal of corner trajectory does not indicate that the global trajectory is also optimal. Actually, in some conditions local optimality of corner trajectory may make things worse. See Fig. 3d, based on Eq. 8 the optimized acceleration should satisfy

$$\mathbf{a}//\mathbf{e}_e \text{ or } \mathbf{a}//\mathbf{e}_s.$$

The corresponding velocity becomes $v_s = 0$ or $v_e = 0$, which means full stop at the corner point.

For the problem of maximizing the minimum value of v_s and v_e, the objective function is

$$\max(\min(v_s, v_e)). \tag{9}$$

Based on Eq. 3, the optimization function can be written as

$$\max(\min(\|\mathbf{a} \times \mathbf{e}_e\|, \|\mathbf{a} \times \mathbf{e}_s\|)).$$

Define $\mathbf{a} = \alpha\mathbf{e}_e - \beta\mathbf{e}_s$ where $\alpha > 0, \beta > 0$. Then the optimization function can be further written as

$$\max(\min(\beta, \alpha)),$$

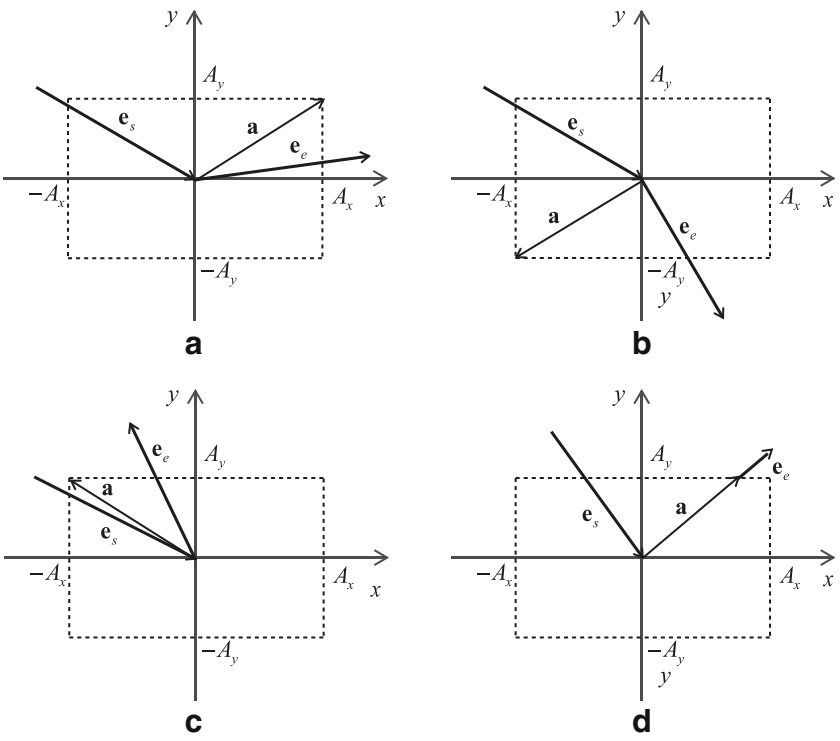


Fig. 3 Optimal acceleration vector determined by Eq. 8 [24, 25]

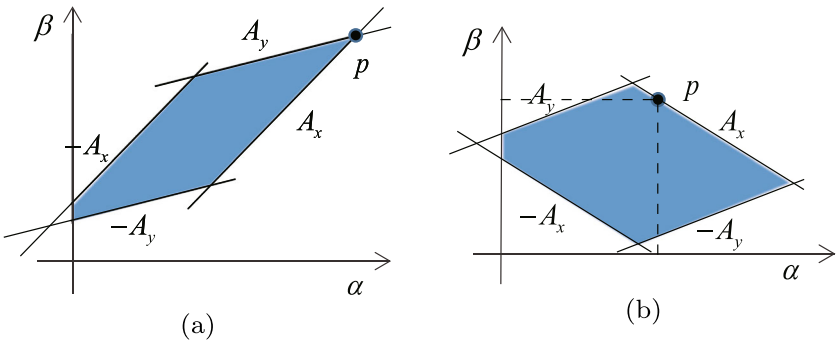


Fig. 4 Optimal acceleration variables α, β determined by Eq. 9 [24]

where the variables α, β are limited by the acceleration constraints as follows

$$|(\alpha \mathbf{e}_e - \beta \mathbf{e}_s) \cdot \mathbf{e}_x| \leq A_x, \left|(\alpha \mathbf{e}_e - \beta \mathbf{e}_s) \cdot \mathbf{e}_y\right| \leq A_y. \tag{10}$$

From constraints (10), the feasible search region of α, β is surrounded by a polygon (e.g. parallelogram for 2 axis problem), then the optimal values of α, β can be found: (i) At some vertex of the polygon, e.g. point **p** on Fig. 4a. (ii) On an edge of the polygon under the condition $\alpha = \beta$, see point **p** on Fig. 4b. In case (ii), since $\alpha = \beta$, the bisector direction of $\mathbf{e}_e$ and $(-\mathbf{e}_s)$ is the acceleration direction, we have

$$\mathbf{e}_a = \frac{\mathbf{e}_e - \mathbf{e}_s}{\|\mathbf{e}_e - \mathbf{e}_s\|}, \tag{11}$$

where $\mathbf{e}_a$ denotes the acceleration direction.

We now consider the optimization of the motion planning. For motion planning of a free path, the optimal solution is obtained while along the motion at least one of the axis accelerations reaches its maximum. To satisfy this condition, a criterion is proposed in the following to determine the acceleration direction.

As shown in Fig. 3, the feasible acceleration space constructed by the acceleration limits is a polygon (e.g., rectangle for a two-axis problem). If one of the polygon vertexes fall within the corner region, then the acceleration determined by Eq. 8 is selected, else the acceleration determined by Eq. 9 is selected.

The transition time calculation is shown in Fig. 5. For transition trajectory (4), if we assume the trajectory travels through a given fixed point **M** at time t_1 and set $t_2 = T - t_1$, then we have

$$\begin{aligned}(\mathbf{M} - \mathbf{P}) &= -l_s \mathbf{e}_s + v_s t_1 \mathbf{e}_s + \frac{1}{2}\mathbf{a}{t_1}^2, \\ &= l_e \mathbf{e}_e - v_e t_2 \mathbf{e}_e + \frac{1}{2}\mathbf{a}{t_2}^2.\end{aligned} \tag{12}$$

Here we rewrite $(\mathbf{M} - \mathbf{P})$ as $(\mathbf{M} - \mathbf{P}) = M_e \mathbf{e}_e - M_s \mathbf{e}_s$. Then

$$t_1 = \sqrt{\frac{2M_e}{a_e}}, t_2 = \sqrt{\frac{2M_s}{a_s}}, \tag{13}$$

and the total transition time at the corner region is

$$T = \sqrt{\frac{2M_e}{a_e}} + \sqrt{\frac{2M_s}{a_s}}. \tag{14}$$

In Eq. 14, the acceleration vector has been determined in this section, and the only unknown variable **M** will be determined in the next section.

2.3 Control of the contouring error

As shown in Fig. 6, the corner transition algorithm introduces a geometric error around the corner that should be limited by the geometric error tolerance ε_B. To fully use the machine performance, we expect the maximum offset ε of the transition trajectory to approach ε_B while satisfying $\varepsilon \leq \varepsilon_B$. Actually, the maximum offset ε of the transition trajectory is the distance (denoted by d) from the corner point **P** to the transition trajectory $\mathbf{Q}(t)$, which is an extremum of $\mathbf{C}^2(t)$. Since $\mathbf{C}^2(t)$ is a quartic polynomial w.r.t time t, here we need to ensure the following result.

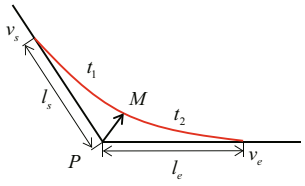


Fig. 5 Transition time calculation

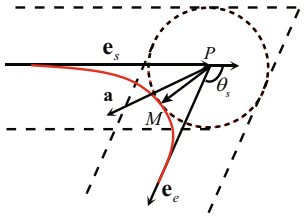


Fig. 6 The maximum offset of the corner transition trajectory

Theorem 1 *Function $\boldsymbol{C}^2(t)$ has one and only one extremum in interval $[0, T]$, which means the nearest distance point $\boldsymbol{M}$ on trajectory $\boldsymbol{Q}(t)$ is unique.*

Proof Define $f(t) = 2\mathbf{C}(t) \cdot \mathbf{C}'(t)$, we have

$$\begin{aligned} f(t) &= 2\left(-a_s\mathbf{e}_s\tfrac{(T-t)^2}{2} + a_e\mathbf{e}_e\tfrac{t^2}{2}\right)\cdot\left(a_s\mathbf{e}_s(T-t) + a_e\mathbf{e}_e t\right) \\ &= T^3{a_s}^2\left(-\left(1-\tfrac{t}{T}\right)^3 + \left(\tfrac{a_e}{a_s}\right)^2\left(\tfrac{t}{T}\right)^3\right. \\ &\quad \left. + \mathbf{e}_e\cdot\mathbf{e}_s\left(\tfrac{a_e}{a_s}\right)\left(\tfrac{t}{T}\right)\left(1-\tfrac{t}{T}\right)\left(2\tfrac{t}{T}-1\right)\right). \end{aligned}$$

Define $x = \frac{t}{T}$, $\lambda = \frac{a_e}{a_s}$ and $\mu = \mathbf{e}_s \cdot \mathbf{e}_e$, then $0 \le x \le 1$, $0 < \lambda < \infty$ and $-1 < \mu < 1$. Function $f(t)$ becomes

$$f(x) = T^3{a_s}^2\left(-(1-x)^3 + \lambda^2x^3 + \mu\lambda x(1-x)(2x-1)\right). \tag{15}$$

Let $f(x) = T^3{a_s}^2 g(x)$, then the extreme points of function $\mathbf{C}(t)^2$ can be determined by solving the equation $g(x) = 0$.

Since we have

$$g(x) = -1 + (3-\mu\lambda)x + 3(\mu\lambda-1)x^2 + \left(1-2\mu\lambda+\lambda^2\right)x^3, \tag{16}$$

there exist

$$g(0) = -1 < 0 \text{ and } g(1) = \lambda^2 > 0, \tag{17}$$

which means function $g(x)$ has at least one zero for $x \in [0, 1]$.

We further have

$$g'(x) = (3-\mu\lambda) + 6(\mu\lambda-1)x + 3(1-2\mu\lambda+\lambda^2)x^2, \tag{18}$$

$$g''(x) = 6(\mu\lambda-1) + 6(1-2\mu\lambda+\lambda^2)x. \tag{19}$$

Then there exist

$$g'(0) = (3-\mu\lambda),\ g'(1) = \lambda(3\lambda-\mu) \tag{20}$$

and

$$g''(0) = 6(\mu\lambda-1),\ g''(1) = 6\lambda(\lambda-\mu). \tag{21}$$

The problem will be discussed in three cases.

Case 1: $\mu\lambda \ge 1$.

In this case, first we have $\lambda > 1$ and $0 < \mu < 1$. Then we have

$$g'(1) > 0,\ g''(0) > 0,\ g''(1) > 0.$$

Since $g''(x)$ is a linear function w.r.t x, we always have

$$g''(x) > 0 \text{ for } x \in [0, 1].$$

Now if $g'(0) \ge 0$, we know function $g(x)$ is monotonely increasing and then based on Eq. 17, function $g(x)$ has a unique zero in interval $[0, 1]$ (see the red line in Fig. 7a). Else we have $g'(0) < 0$, the corresponding $g(x)$ decreases first and then increases in $[0, 1]$ (see blue dash line in Fig. 7a). And the zero of function $g(x)$ is also unique in interval $[0, 1]$.

Case 2: $\mu\lambda < 1$ and $\lambda \le \mu$.

In this case, we have

$$g'(0) > 0,\ g''(0) < 0,\ g''(1) \le 0.$$

Since the symmetry axis of quadratic function (18) is

$$x_o = \frac{1-\mu\lambda}{1-2\mu\lambda+\lambda^2}.$$

Based on $\mu\lambda < 1$ and $\lambda \le \mu$, we have $x_o \ge 1$. Then the function $g'(x)$ is monotonely decreasing in interval $[0, 1]$.

Now if $g'(1) \ge 0$, function $g(x)$ is monotonely increasing and has unique zero in interval $[0, 1]$ (see the red line in Fig. 7b). Else $g'(1) < 0$, the corresponding $g(x)$ increases first and then decreases in $[0, 1]$ (see blue dash line in Fig. 7b). And the zero of function $g(x)$ is also unique in interval $[0, 1]$.

Case 3: $\mu\lambda < 1$ and $\lambda > \mu$.

In this case, we have

$$g'(0) > 0,\ g'(1) > 0,\ g''(0) < 0,\ g''(1) > 0.$$

And we know the symmetric axis x_o of quadratic function (19) satisfies

$$0 < x_s < 1.$$

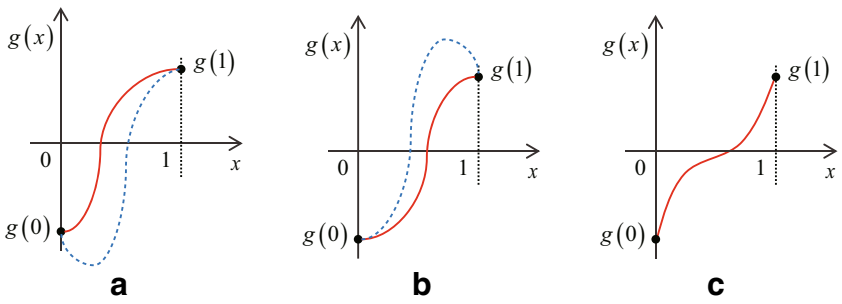


Fig. 7 The profiles of $g(x)$ in Case 1 **(a)**, Case2 **(b)** and Case 3 **(c)**

The minimum of quadratic function (18) has the following formula.

$$3-\mu\lambda-\frac{3(\mu\lambda-1)^2}{1-2\mu\lambda+\lambda^2}.$$

And it is easy to check

$$3-\mu\lambda-\frac{3(\mu\lambda-1)^2}{1-2\mu\lambda+\lambda^2}>0.$$

It means we have $g'(x)>0$ for $x\in[0,1]$. So the zero of function $g(x)$ is unique in interval $[0,1]$ (see Fig. 7c).

Hence, function $\mathbf{C}(t)^2$ has one and only one extremum in interval $[0,T]$. □

Theorem 2 *We have two more properties about the nearest distance:*

(i). *The nearest distance d is a quadratic and monotonely increasing function w.r.t the transition time T.*

(ii). *The nearest direction (denoted by $\boldsymbol{e}_\varepsilon$) from the corner point $\boldsymbol{P}$ to the trajectory $\boldsymbol{Q}(t)$ is a constant, which means it is independent of time T.*

Proof Based on Theorem 1, we name the unique zero point of $g(x)$ in $[0,1]$ as x_ε. Then according to Eq. 7, the nearest distance point on trajectory $\mathbf{C}$ becomes

$$\mathbf{C}(x_\varepsilon)=T^2\left(-a_s\mathbf{e}_s\frac{(1-x_\varepsilon)^2}{2}+a_e\mathbf{e}_e\frac{{x_\varepsilon}^2}{2}\right). \tag{22}$$

Then the nearest distance is

$$d=\sqrt{\mathbf{C}(x_\varepsilon)\cdot\mathbf{C}(x_\varepsilon)}=T^2\left\|-a_s\mathbf{e}_s\frac{(1-x_\varepsilon)^2}{2}+a_e\mathbf{e}_e\frac{{x_\varepsilon}^2}{2}\right\|_2.$$

Hence, the property (i) is proven.

In Eq. 22, it is easy to verify that the direction of $\mathbf{C}(x_\varepsilon)$ is a constant. So the property (ii) is proven. □

Based on Theorem 1, the common used numerical algorithms (such as bisection method) can be applied to calculate the nearest distance point $\mathbf{C}(x_\varepsilon)$ and the nearest distance direction $\mathbf{e}_\varepsilon$.

2.4 Re-adjustment of corner transition trajectory

In this section, we present an adjustment method for corner transition trajectory to satisfy the maximum feedrate limit and the limited line segment length.

Define feedrate as $v_{feed}(t)=\|\mathbf{v}(t)\|$. For the feedrate limit of transition trajectory

$$v_{feed}(t)\le v_B, t\in[0,T], \tag{23}$$

we have the following theorem.

Theorem 3 *The feedrate constraint $v_{feed}(t)\le v_B$ is equivalent to constraints $v_s\le v_B$ and $v_e\le v_B$.*

Proof The velocity of transition trajectory at time t satisfies $\mathbf{v}_t=v_s\mathbf{e}_s+\mathbf{a}t$. Then we have

$$\|\mathbf{v}_t\|^2=\mathbf{v}_t\cdot\mathbf{v}_t=t^2(\mathbf{a}\cdot\mathbf{a})+2v_st(\mathbf{a}\cdot\mathbf{e}_s)+{v_s}^2 \tag{24}$$

and $\frac{\mathrm{d}\|\mathbf{v}_t\|^2}{\mathrm{d}t}=2\mathbf{a}\cdot\mathbf{v}_t$. Since $\|\mathbf{v}_t\|^2$ is a quadratic function and $\mathbf{a}\cdot\mathbf{a}>0$, and the minimum velocity is obtained at $\mathbf{a}\cdot\mathbf{v}_t=0$, the corresponding time is $t=-\frac{v_s(\mathbf{a}\cdot\mathbf{e}_s)}{(\mathbf{a}\cdot\mathbf{a})}$. The maximum velocity can be found at $t=0$ or $t=T$. Hence, the feedrate constraint $v_{feed}(t)\le v_B$ along the transition trajectory can be ensured by $v_s\le v_B$ and $v_e\le v_B$. □

For a certain corner, the acceleration direction $\mathbf{e}_a$ is determined in Section 2.2, and based on Theorem 2, the nearest distance direction $\mathbf{e}_\varepsilon$ is a constant, written as

$$\mathbf{e}_\varepsilon=\varepsilon_e\mathbf{e}_e-\varepsilon_s\mathbf{e}_s$$

with $\varepsilon_e>0,\varepsilon_s>0$.

Then we define

$$C_a=\sqrt{\frac{2\varepsilon_e}{a_e}}+\sqrt{\frac{2\varepsilon_s}{a_s}}$$

which is a constant. Hence we get the transition time

$$T=\sqrt{\varepsilon}C_a, \tag{25}$$

the terminal transition velocities

$$v_s=a_sC_a\sqrt{\varepsilon},\ v_e=a_eC_a\sqrt{\varepsilon}, \tag{26}$$

and the transition lengths

$$l_s=\frac{1}{2}a_s\varepsilon{C_a}^2,\ l_e=\frac{1}{2}a_e\varepsilon{C_a}^2. \tag{27}$$

The adjustment conditions can be summarized as the velocity condition and length condition, where the velocity condition is induced by the look-ahead scheme and the maximum feedrate limit, the length condition is induced by the limited segment length.

For the velocity re-adjustment (take v_e as example), we expect the adjusted velocity to be equal to a new value, denoted by v_{enew}. According to Eq. 26, both the recalculated a_B and ε can realize the requirement. However, from Eq. 25, shrinking a_B leads to increasing of T. Hence, the new velocity is generated by applying the new maximum offset

$$\varepsilon_v=\varepsilon\left(\frac{v_{enew}}{v_e}\right)^2. \tag{28}$$

Similarly, the new transition length l_{enew} can be generated by using the recalculated

$$\varepsilon_l=\frac{l_{enew}}{l_e}\varepsilon. \tag{29}$$

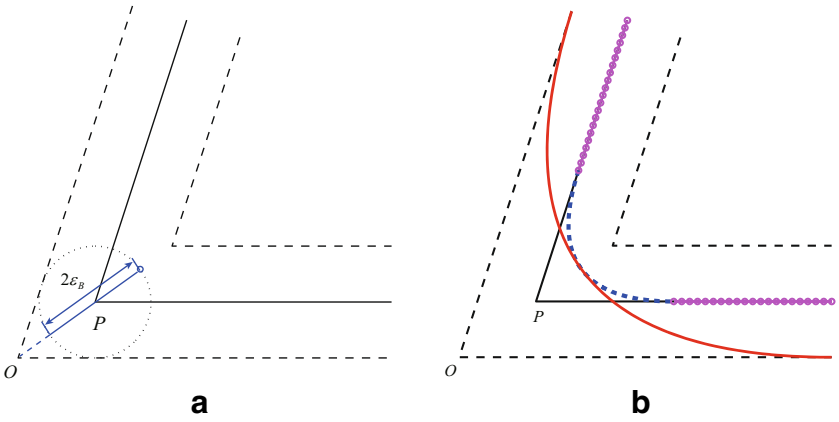


Fig. 8 **a** A typical corner. **b** Transition trajectories based on the standard model (*blue dot line*) and the modified model (*red line*)

Above all, the final applied maximum offset is determined by

$$\varepsilon_{final} = \min\{\varepsilon_v, \varepsilon_l, \varepsilon_B\}. \tag{30}$$

Combined with the known $\mathbf{P}, \mathbf{e}_s, \mathbf{e}_e, \mathbf{a}, \mathbf{e}_\varepsilon$ and ε_{final}, the transition trajectory of the corner can be determined.

3 Minimum time motion strategy

Based on the developed corner transition algorithm in Section 2, a minimum time motion planning algorithm is proposed in this section.

3.1 A modified corner transition model

According to Eqs. 25 and 26, we know that the transition time and velocity can be improved by stretching the maximum offset ε. In the standard transition model, the maximum offset ε w.r.t point **P** should satisfy $\varepsilon \leq \varepsilon_B$. We give a modified model where the corner point of the transition trajectory is moved from point **P** to point **O** (Fig. 8a). This means the maximum offset of the modified transition model is larger than $2\varepsilon_B$ in some situations.

A numerical experiment is presented to test the effectiveness of the modified transition model. For a given corner, the standard transition model and the newly proposed model are applied. To be fair, both acceleration directions are set as the direction of corner bisector and the corner length is fixed which is exactly equal to the transition length of the modified model.

Then for the modified transition model, the maximum offset point is shown in Fig. 8a and the maximum offset equals to $\varepsilon_O = \frac{\varepsilon_B}{\|\mathbf{e}_e \times \mathbf{e}_b\|} + \varepsilon_B$, where, $\varepsilon_O > 2\varepsilon_B$, $\mathbf{e}_b$ denotes the direction of corner bisector.

The transition trajectories for a typical corner based on the two models are plotted in Fig. 8b respectively. In this figure, the red line denotes the modified transition trajectory which has larger offset than the standard transition trajectory drawn by blue dot line. The pink circle-lines are the additional linear trajectories to ensure the two models travel the same corner length. And the acceleration along the linear path is calculated by

$$a_{line} = \min\left(\frac{A_x}{|\mathbf{e}_{line} \cdot \mathbf{e}_x|}, \frac{A_y}{|\mathbf{e}_{line} \cdot \mathbf{e}_y|}\right),$$

where $\mathbf{e}_{line}$ is the direction of certain linear path.

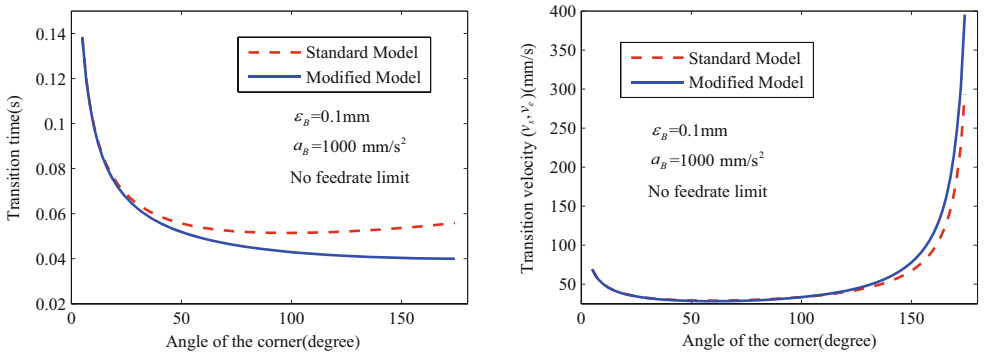
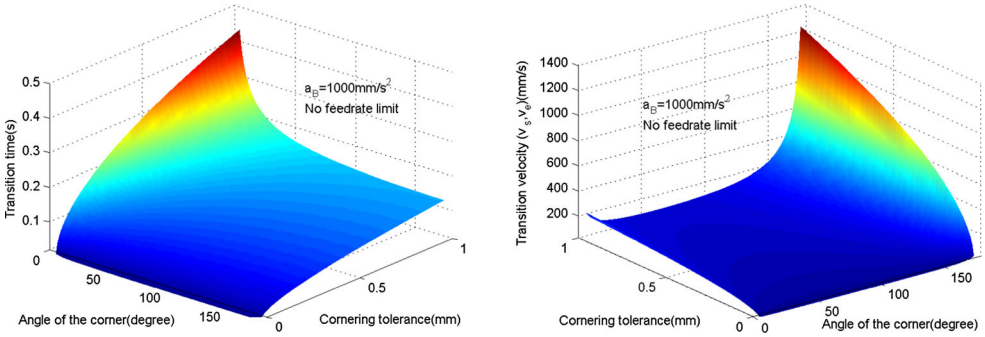


Fig. 9 Performance comparison between the standard model and the modified model



a Transition time **b** Transition velocity

Fig. 10 Performance test of the modified model

An experiment with variable corner angle is implemented to show the variations of the transition time and velocity for the two models under the same corner length. The variation range of the angle is set from 5 degree to 175 degree. The default experiment conditions are: cornering tolerance = 0.1 mm, axis acceleration = 1000 mm/s^2, no feedrate limit. The results are shown in Fig. 9. Based on the experiment data of Fig. 9, up to 28.4 % machining time can be saved by using the modified transition model, and the average time saved is 14.2 %. The maximum transition velocity improvement is 34.8 % and the mean improvement is 4.4 %.

Then the performance of the modified model w.r.t. the variables of corner angle and cornering tolerance is tested as shown in Fig. 10. From Fig. 10, we find the sensitivity of transition time w.r.t cornering tolerance increases significantly as the corner angle approaches to zero. Conversely, the sensitivity of transition velocity w.r.t cornering tolerance increases significantly as the corner angle approaches to 180 degree.

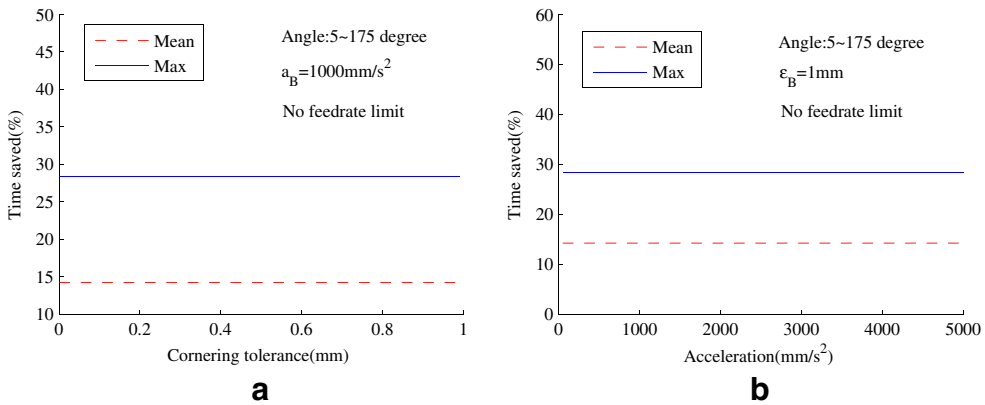
Two more tests are implemented. In the first test, the default conditions are: axis acceleration $a_B = 1000$ mm/s^2, no feedrate limit. We let the tolerance varies from 0.0001 mm to 1 mm and under each tolerance we collect the data of time saved when the corner angle varies from 5 degree to 175 degree. The collected data is drawn in Fig. 11a. Similarly, in the second test $\varepsilon_B = 0.1$ mm is set as default and the acceleration varies from 0 mm/s^2 to 5000 mm/s^2. The collected data are drawn in Fig. 11b. From Fig. 11, we find that the performance improvement of the modified model compared with the standard model is not influenced by the cornering tolerance and the acceleration.

3.2 Corner classification and treatment

In this paper, corners are classified as inner corner, outer corner, start corner and end corner (or possible flat angle corner). See Fig. 12, the shadow region denotes the workpiece, where the dotted lines are the offset with error bound as the height. An inner corner is shown in Fig. 12a. An out corner is shown in Fig. 12b. In order to be time-optimal, it is easy to see that for the inner corner shown in Fig. 12a, using **O** as the support point of the transition trajectory is the best choice. Similarly, in Fig. 12b, **Q** is the best support point for the outer corner.

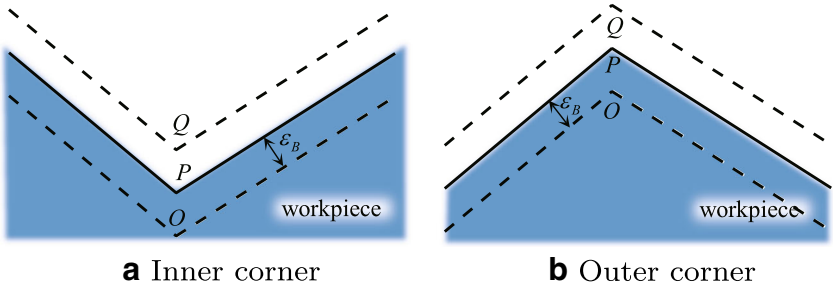
We use Fig. 13a to illustrate the planned trajectory. The tool path has one inner corner and one outer corner. By using the modified transition model, the trajectory for each separate corner can be generated.

A transition trajectory is needed to link the two none collinear trajectories. To do this, a pair of auxiliary corners is introduced as shown in Fig. 13a. By trisecting the line



a **b**

Fig. 11 Performance of the modified model compared with the standard model



a Inner corner **b** Outer corner

Fig. 12 Description of the typical corners

segment MN, we obtain the desired auxiliary corners $\mathbf{P_s}$ and $\mathbf{P_e}$. By applying the corner transition algorithm in Section 2, the link trajectory can be generated as shown in Fig. 13b. Generally, the length of path segment is much longer than the given offset ε. So the angles of the constructed additional corners are relatively obtuse and no significant feedrate loss is caused by this modification (more details can be found in Fig. 14).

The modified transition model is time-optimal under the assumption that the length of each line segment is long enough and the transition feedrate is less than the feedrate limit. In the contrary case, we will shrink the offset ε to make the connection possible, and thus obtain an approximate optimal algorithm. We use the following simple criterion to modify the parameter ε. For a given corner, the modified transition model is applied only when the following conditions are satisfied, otherwise the standard transition model will be used for this corner.

1. The corner length conditions The left corner length should longer than

$$L_l \geq a_s {C_a}^2 \varepsilon_B \left(1 + \frac{a_s}{a_{line}}\right). \tag{31}$$

The right corner length should longer than

$$L_r \geq a_e {C_a}^2 \varepsilon_B \left(1 + \frac{a_e}{a_{line}}\right). \tag{32}$$

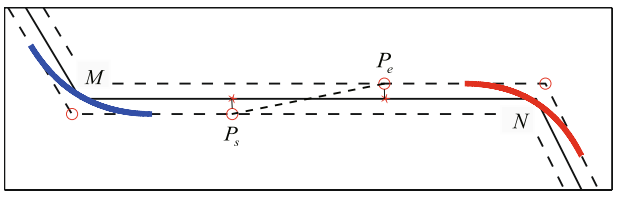
In Eqs. (31) and (32), a_{line} is the acceleration value along the corresponding linear segment.

2. The velocity conditions

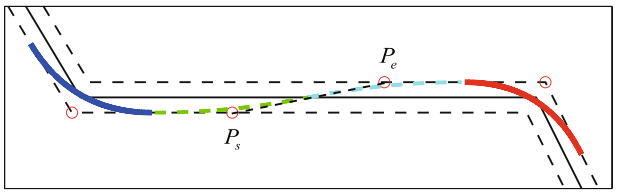
$$a_s C_a \sqrt{\varepsilon_B} \leq v_B, a_e C_a \sqrt{\varepsilon_B} \leq v_B. \tag{33}$$

3. The corner angle condition. Referring to Fig. 9a, the corner angle should be larger than $\pi/3$. This is because the performance difference between the standard model and the modified model is small when the corner angle is less than $\pi/3$.

The tool path shown in Fig. 14 is applied to test the proposed strategy. The cornering tolerance is set to be 0.8 mm and the feedrate limit is set to be $F = 200$ mm/s. Acceleration limits are $A_x = 1000$ mm/s^2, $A_y = 1500$ mm/s^2. The trajectories generated by the pure standard model and the pure modified model are compared. Figure 14b lists

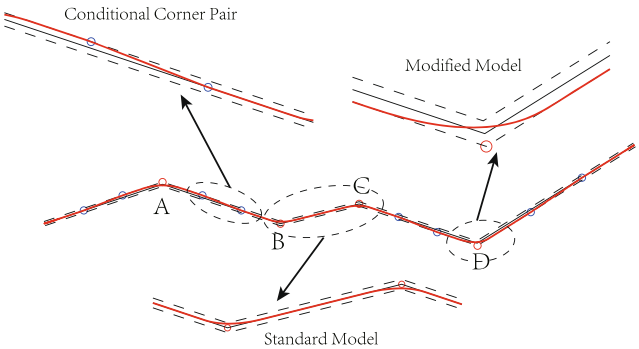


a An example path and the constructed conditional corner pair

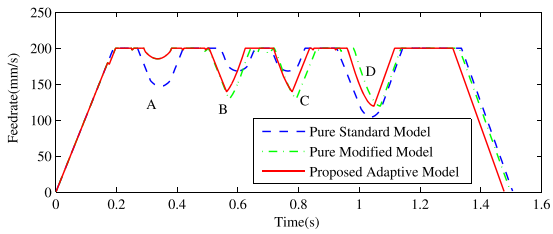


b Trajectory transition generation

Fig. 13 Trajectory transition from inner corner to outer corner



a Example path and the corresponding transition trajectory.



b Feedrate curves for the example path.

Fig. 14 Application of the conditional corner pair

the feedrate curves for the two models. As shown in the figures, corners A and D satisfy the above criteria, so the feedrates of corner A and D are larger than the standard one. The lengths of AB and BC are too short, and the shrinked transition offset makes the performance of the modified model almost the same with that of the standard model, or even worse. Therefore, from these figures, we can see that the proposed adaptive transition model selection (ATMS) method is effective. The performance improvement of this method is 1.86 % compared with the pure standard transition model (PSTM) and 1.4 % compared with the pure modified transition model (PMTM) for this example. In Section 4, more experiments will be executed to show the effectiveness of the criteria.

3.3 Minimum time motion construction

Since the transition velocity of each corner is calculated separately, to ensure the continuous of the feedrate along the whole tool path, the look-ahead scheme with feedrate update is needed, which is a standard procedure [23, 24].

For instance, in Fig. 15, if

$$v_e(i) > v_s(i+1) \text{ and} L(i) < \frac{\left(v_e{}^2(i) - v_s{}^2(i+1)\right)}{2a}, \quad (34)$$

which means that the line segment is too short to decrease the feedrate, then the velocity of corner i is changed to

$$v_{enew}(i) = \sqrt{\frac{2d(i)av_e(i) - aT(i+1)v_s(i+1)v_e(i) + v_e(i)v_s{}^2(i+1)}{v_e(i) + aT(i)}}. \quad (35)$$

In summary, we outline the proposed algorithm as follows. Suppose that the error tolerance ε_B, the feedrate limit v_B, the maximum axis acceleration $\{A_x, A_y\}$, the interpolation period T_s, and the tool path with corner points $\mathbf{P}_1, \ldots, \mathbf{P}_n$ are given.

- Step 1: For each corner point $\mathbf{P}_i$, compute the optimal acceleration $\mathbf{a}_i$ with the method given in Section 2.2 and the maximal offset point $\mathbf{M}_i$ with the method given in Section 2.3.
- Step 2: Use the procedure given in Sections 3.1 and 3.2 to determine the transition (modified or standard) model and the support point $\mathbf{S}_i$. Then v_{is}, v_{ie} can be computed with (3), l_{is}, l_{ie} can be computed with (6). Adjust the above values using the method in Section 2.4 if it is needed.
- Step 3: Perform the look-ahead loop to adjust the parameters $v_{is}, v_{ie}, l_{is}, l_{ie}$ to make the feedrates reachable at any line segment $\mathbf{P}_i\mathbf{P}_{i+1}$.
- Step 4: Compute the interpolation points. Based on the data from Step 1 and 2, the analytic trajectories along the line segments and corner transition segments can be calculated. Then we move along the analytic trajectories in sequence, and in the meantime output the interpolation points $\mathbf{p}(T_s), \mathbf{p}(2T_s), \ldots, \mathbf{p}(kT_s)$ every interpolation period T_s.

We finally remark that the feedrate curves generated with the above algorithm can be smoothed with the method given in [27] to avoid jumps of the acceleration.

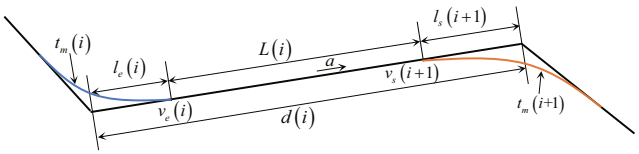


Fig. 15 Velocity reachability between two adjacent corners

4 Simulation and experimental validations

The proposed corner transition algorithm is tested through simulations and experiments. The simulation test is performed by using a simple planar contour to illustrate the effect of corner transition trajectory. The experiments are executed on a typical 3-axis CNC machine tool whose axis acceleration performances are significantly different. Finally, the time-optimality of the proposed algorithm is tested by comparing the machining times with the known time optimal algorithm [7].

4.1 Simulation results

A simple polygon contour shown in Fig. 16 is used. The start and end positions are set at point (20, −10). There are 13 corners, of which four are inner corners and seven are outer

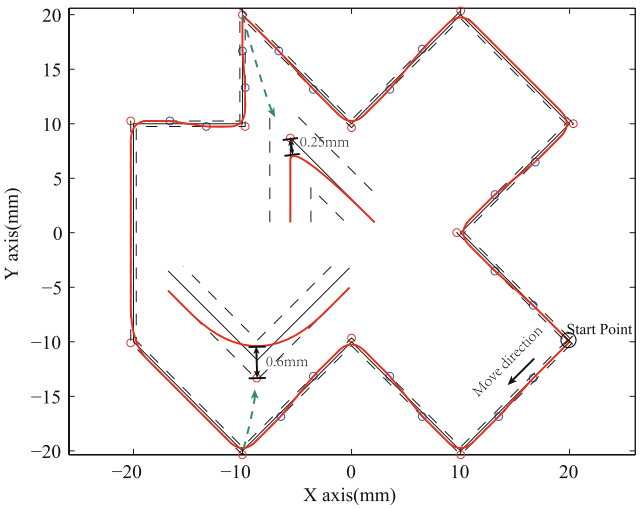


Fig. 16 A simple polygon contour and its blended motion trajectory

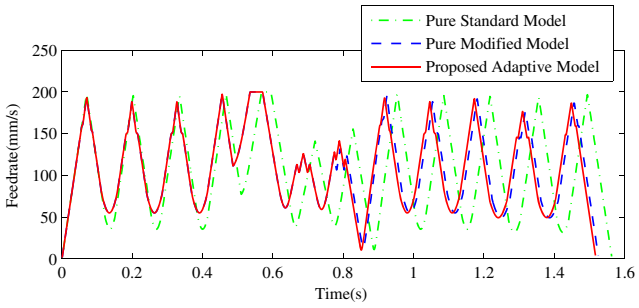


Fig. 17 The scheduled feedrate curves

corners. The feedrate limit is 200 mm/s, the error tolerance of the corner is 0.25 mm, the acceleration limits of x and y axes are 2000 mm/s^2 and 2500 mm/s^2 respectively.

By verifying the model selection conditions (31)–(33), 10 of the corners can be transited by using the modified model and the rest are transited by using the standard model. The blended motion trajectory of this contour is shown in Fig. 16 with the corresponding feedrate curve (red line) and axis acceleration profiles shown in Figs. 17 and 18c respectively. Based on the proposed algorithm, the total motion time is 1.518 s compared with 1.564 s of the pure standard transition model (PSTM) algorithm and 1.526 s of the pure modified transition model (PMTM) algorithm.

Then based on the proposed ATMS algorithm, we test the trajectory performance under different corner tolerances.

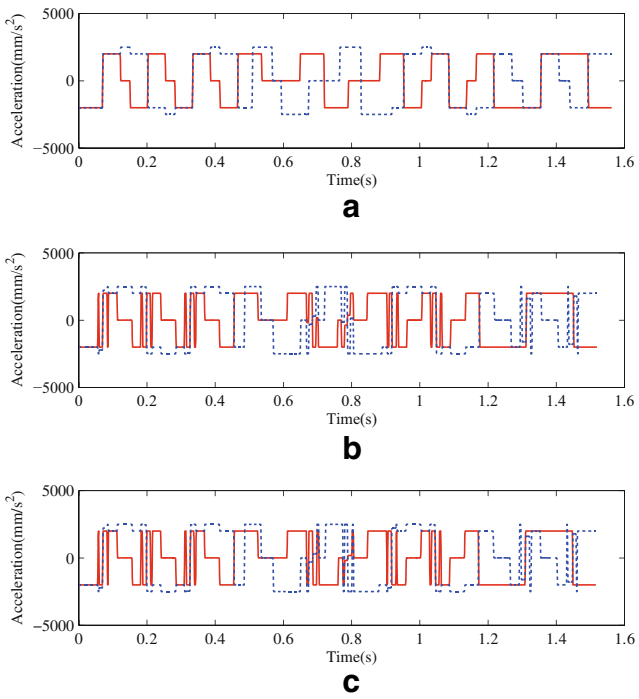


Fig. 18 Acceleration profiles of the scheduled feedrate by the pure standard model **(a)**, the pure modified model **(b)** and the proposed adaptive model **(c)**, respectively

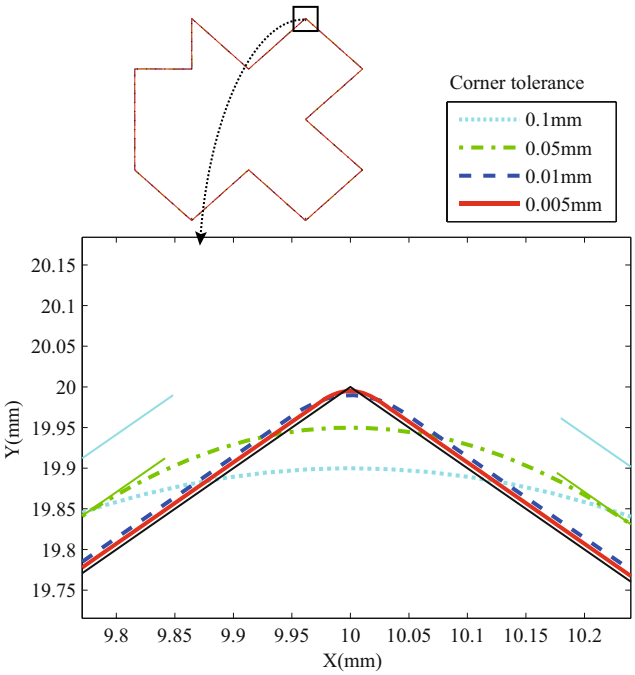


Fig. 19 Corner trajectories under different error tolerances

Here, we adjust the feedrate limit to 100 mm/s. Figure 19 shows the transition trajectories under different corner tolerances for a given corner. Figure 20 shows the corresponding feedrate curves around the corner. In Fig. 20, the PSTM algorithm with corner tolerance $\varepsilon = 0.005$ mm is applied as reference. The chart in Fig. 20 shows as the corner tolerance shrinks, the corner velocity of the trajectory decreases accordingly. Benefit from the modified corner model, the ATMS algorithm can obtain larger corner velocity than the PSTM algorithm. For the tolerance settings 0.1 mm, 0.05 mm, 0.01 mm, 0.005 mm and 0.005 mm(PSTM), the corresponding interpolation times are 1.98, 2.021, 2.077, 2.09, and 2.103 s, respectively.

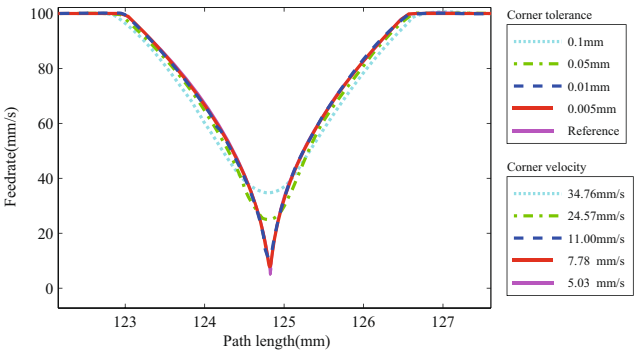


Fig. 20 Corner feedrates under different error tolerances

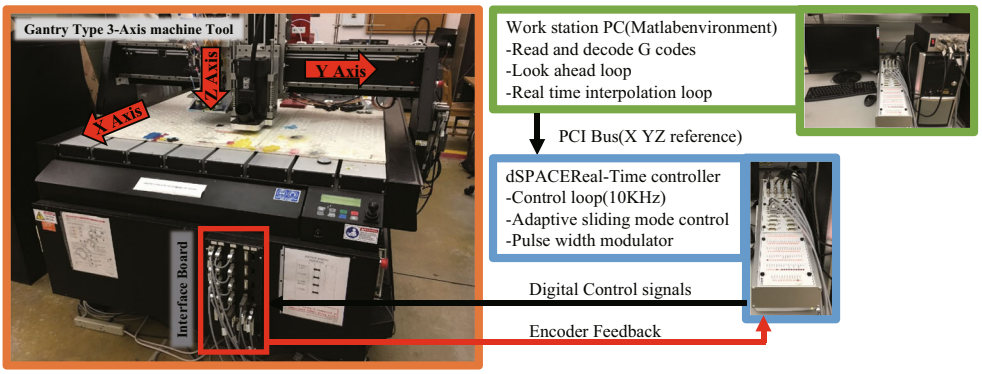


Fig. 21 Experimental setup

4.2 Experimental results

A gantry-type three-axis machine tool is retrofitted to implement the algorithm verification. The system architecture for experiment is illustrated in Fig. 21. The codes of the corner transition algorithm, the feedrate update with look-ahead and the interpolation point generation are performed on a personal PC with Intel i5 2.6 GHz CPU, 8GB SDRAM and Matlab environment. A dSpace real-time controller board (DS1103) equipped with an adaptive sliding model control core is utilized to execute the closed loop tracking mission. Digital control signals generated by pulse width modulator technology are used to drive the servo motors. Position feedback of each axis is sampled through an encoder with resolution of 0.0026 mm for x and y axes. The interpolation period for the experiment is 1 ms.

The feedrate setting of the machine is 100 mm/s. The dynamic performances are significantly different for individual axes of this machine. As shown in Fig. 21, the motion inertia along the x direction is larger than that along the y and z directions. The acceleration settings are 500mm/s^2 for x axis and 1000 mm/s^2 for y axis respectively. Real-time look-ahead routine is not considered in this experiment, the tested trajectories are all generated offline in a Matlab environment.

In this subsection, a plane cat shape linear profile (as shown in Fig. 22) is applied as test object. The error tolerance of the corner is 0.1 mm. The number of linear segments is 131 with the segment lengths varying from 2.82 mm to 30.43 mm, and the total length of the cat profile is 1654.13 mm.

Besides the proposed algorithm, the common used point–point motion and constant feedrate motion are also employed for comparison. Figure 23a shows the scheduled feedrate of the point–point motion strategy with the corresponding acceleration profiles shown in Fig. 23b. The scheduled motion satisfies the feedrate and acceleration limits. However, since the motion must fully stop at every corner, the 33.05 sec of the motion time is very long. In contrast, the constant feedrate motion (Fig. 24a blue dot line) has the shortest motion time which is 16.715 sec, however large violations of acceleration limit exist in this strategy as shown in Fig. 24b and c. Here by using the corner transition strategy, continuous quadratic curve is calculated for each corner within the cornering tolerance, which increases the passing feedrate at each corner point significantly. Then

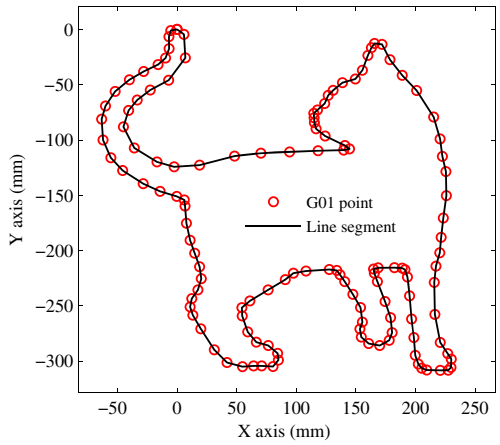


Fig. 22 Experimental tool path

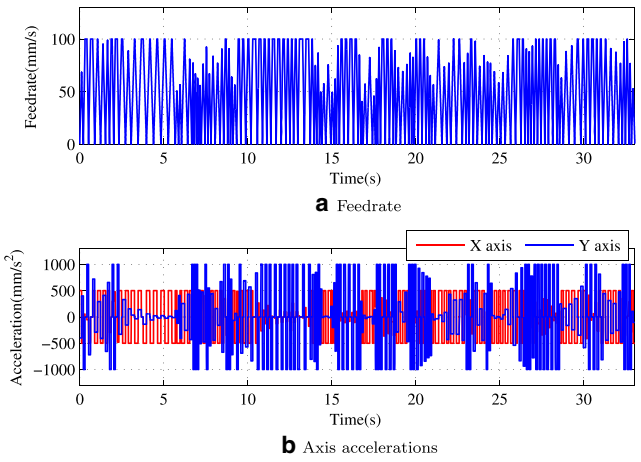


Fig. 23 Profiles of point to point motion

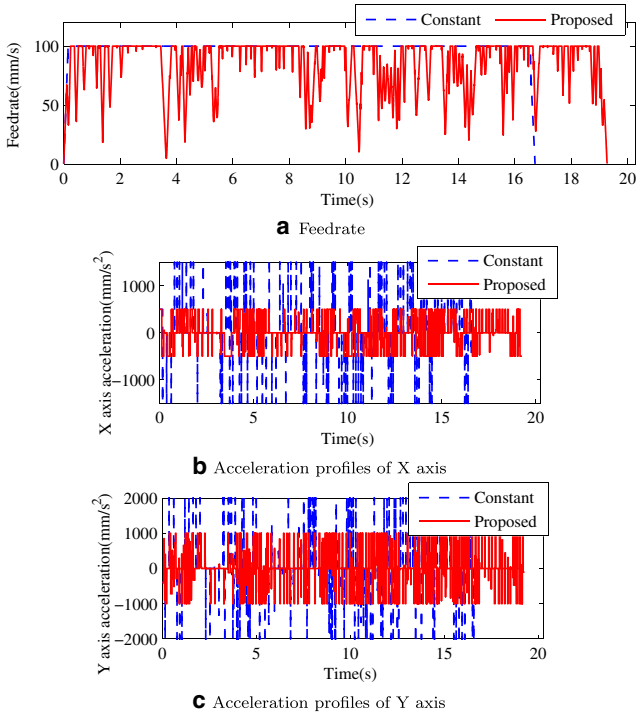


Fig. 24 Comparison between the scheduled and constant feed profiles

combined with the look-ahead process, the scheduled feedrate curve of the proposed algorithm is generated and shown in Fig. 24a by red line. The corresponding acceleration profile is shown in Fig. 24b and c and none acceleration limit is violated. The final motion time in the proposed algorithm is 19.271 sec which is 15.3 % longer than that of the constant feedrate motion and 41.7 % shorter than that of the point to point motion.

The motion performance of the scheduled feedrate profiles are tested. The contour error profiles of real motions are shown in Fig. 25. In addition, a detailed description of the motion error is shown in Fig. 26. For the constant feedrate motion, large tracking errors occur frequently since every large acceleration output is denied by the drives. The motion error is acceptable for the point to point motion, however the motion efficiency is poor. For the proposed algorithm, the error tolerance and machine performance are considered in the feedrate planning. So a proper balance is found. On one side, the drive performance is fully used and high feedrate is

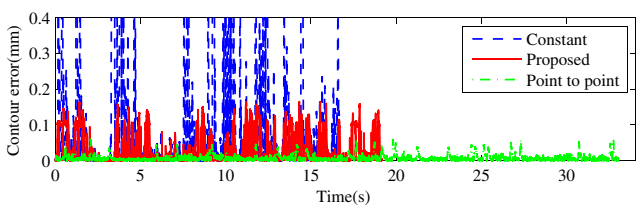


Fig. 25 Contour error profiles of the point to point, constant and proposed algorithms

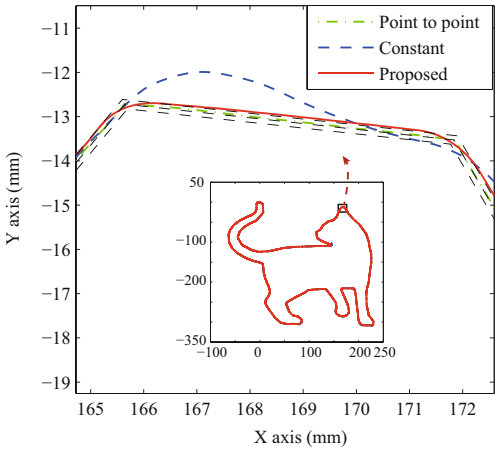


Fig. 26 Detailed description of the motion error

achieved. On the other side, the motion error is controllable by adjusting the feedrate properly. The statistical data of this experiment are listed in Table 1. Since the use of corner transition, the literal record of contour error of the proposed algorithm is relative poor than the point to point algorithm.

Among all the three algorithms, the proposed algorithm can achieve highest productivity while satisfying the requirement of error tolerance. Finally, the detailed trajectories at two corner points are given in Fig. 26.

To further demonstrate the performance of the proposed algorithm, we compare our method with that in [16, 26]. In the compared algorithm, each path corner is blended by a quintic Bezier curve under the corner tolerance setting. Then based on the constructed parametric path, a jerk-limited S-shape feedrate scheduling is applied. Here we name the compared algorithm as Bezier-Jerk Alg for short. Then Bezier-Acc Alg is used to represent the compared algorithm without jerk limit.

Considering the machine setup in Fig. 21, the experiment settings of the Bezier-Jerk Alg are: the feedrate 100 mm/s, the acceleration 500 mm/s^2, the jerk 5000 mm/s^3. The feedrate comparisons are plotted in Fig. 27. The partial enlarged drawing of Fig. 27 shows the propose ATMS algorithm and PSTM algorithm can apply to larger acceleration than the Bezier-Acc algorithm because of the applied axial limits not the tangential limits. Hence, the local maximum feedrate of our algorithms reach 67.22 mm/s, which is larger than 61.21 mm/s of the Bezier-Acc algorithm. Then at the local corner, because of the new transition model, the transition velocity 34.37 mm/s of the ATMS algorithm is larger than the 23.6 mm/s of the PSTM algorithm and the 19.79 mm/s of the Bezier-Acc algorithm. In total, the interpolation time in the ATMS algorithm is 19.27 s, which is 88 % of the time in Bezier-Acc algorithm and 70 % of the time in Bezier-Jerk algorithm. Figure 28 presents the contour error profiles of each algorithm. Because of the

Table 1 Statistical data of the experiment

Algorithm	Motion time (s)	Tracking error (mm)				Contour error (mm)	
		X axis		Y axis			
		Max	Mean	Max	Mean	Max	Mean
P2P	33.05	0.0817	0.0109	0.049	0.0124	0.0604	0.008
Constant	16.715	6.4897	0.2147	0.9874	0.0375	3.4489	0.14
Proposed	19.271	0.0890	0.0127	0.0671	0.0117	0.1654	0.0396

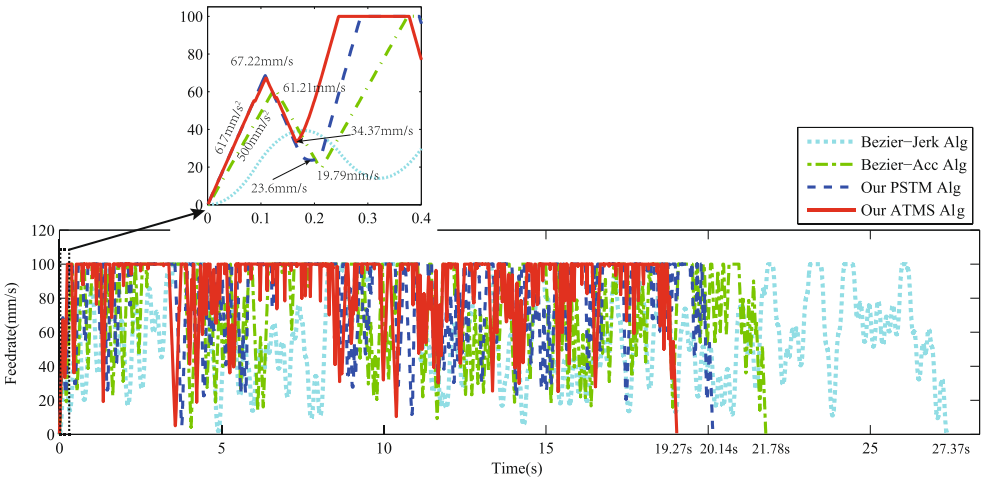


Fig. 27 Feedrate comparisons of the Bezier-Jerk, Bezier-Acc, PSTM and ATMS algorithms

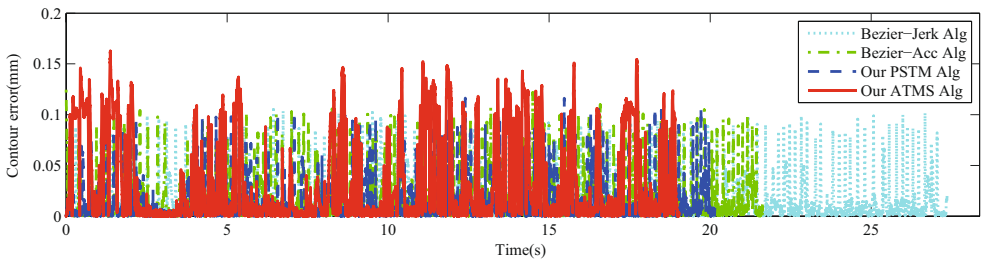


Fig. 28 Contour error profiles of the Bezier-Jerk, Bezier-Acc, PSTM and ATMS algorithms

Table 2 Statistical data of a further experiment

Algorithm	Motion time(s)	Tracking error(mm)				Contour error(mm)	
		X axis		Y axis			
		Max	Mean	Max	Mean	Max	Mean
Bezier-Jerk Alg	27.367	0.0930	0.0131	0.0598	0.0122	0.1049	0.0144
Bezier-Acc Alg	21.776	0.1307	0.0187	0.1062	0.020	0.1246	0.0165
Our ATMS Alg	19.271	0.0890	0.0127	0.0671	0.0117	0.1654	0.0396

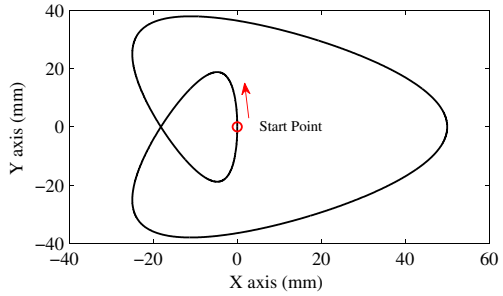


Fig. 29 An parametric curve

corner tolerance setting, the local maximum contour errors of all the four algorithms are comparable and our PSTM algorithm is the best. Table 2 lists the details of these tests.

In summary, among all the four algorithms the proposed ATMS algorithm can achieve highest productivity while the contour error is controllable.

4.3 Test the time-optimality of the proposed algorithm

In this section, we use an example to illustrate the near time-optimality of the proposed approach. The tool path is an analytical curve shown in Fig. 29 with the parametric equation:

$$\begin{cases} x = 25\left(cos\left(2\pi as\right) - cos(2\pi bs)^3\right), \\ y = 25\left(sin\left(2\pi cs\right) - sin(2\pi ds)^3\right), \\ z = -1, \\ a = 2, b = 1, c = 2, d = 1. \end{cases}$$

The error tolerance is set to 0.1 mm. The feedrate setting of the machine is 100 mm/s. The acceleration settings are 500 mm/s^2 for x axis and 1000 mm/s^2 for y axis respectively.

The method proposed in [7] is used to compute the time-optimal feedrate for the tool path. The tool path is discretized into 2000 segments by sampling the curve with $s = 0 : 1/2000 : 1$. Then, the algorithm proposed in [7] is used with these sample points to compute the optimal machining time which is 3.659 s.

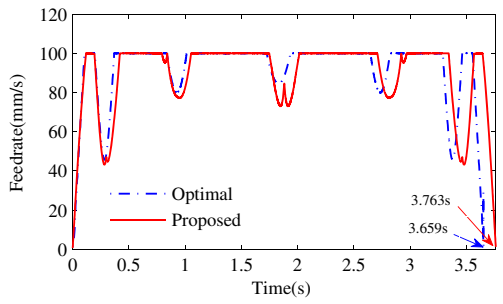


Fig. 30 Feedrate curves

Then the 2000 segments are treated as G01 codes and the method proposed in this paper is used to compute the interpolation points and the machining time is 3.763 s, which is close to that given by the method in [7]. The computation time with our algorithm is 0.94 sec in Matlab, and the feedrate curve computed with our method compared with the time optimal feedrate curve is shown in Fig. 30.

5 Conclusions and further work

In this paper, a near minimum time motion planning algorithm is presented for fast and accurate interpolation of linear tool paths. The discontinuous path corners are smoothed using the proposed local corner transition model. Then the presented global motion planning algorithm is applied to realize the continuous minimum time motion along the whole tool path.

Compared to the previous work, the proposed algorithm has the following advantages: (1). It is a one-step transition trajectory generation algorithm compared with the most studied two steps algorithms. Since the trajectories are time parameterized, the interpolation points can be computed easily. (2). The more reasonable axial acceleration limits are used instead of the tangential acceleration limit. (3). A new corner transition strategy is applied to minimize the machining time. Finally, a NC machine with distinct axis acceleration performances is used to test the newly proposed algorithm to verify the feasibility and efficiency of the proposed algorithm.

Future work will focus on two key issues. First, since the acceleration jumps cannot be realized by the actual physical actuator, limited acceleration change rate will be investigated in the newly proposed approach. Second, more detailed servo dynamics, such as the velocity-dependent acceleration phenomena in the newly proposed approach will be considered.

Acknowledgments The authors are grateful to Precision Controls Laboratory (University of Waterloo, Canada), who provided the equipment for the experiments.

References

1. Wang YS, Yang DS, Gai RL, Wang SH, Sun SJ (2015) Design of trigonometric velocity scheduling algorithm based on preinterpolation and look-ahead interpolation. Int J Mach Tools Manuf 96:94–105
2. Chen M, Zhao WS, Xi XC (2015) Augmented Taylor's expansion method for B-spline curve interpolation for CNC machine tools. Int J Mach Tools Manuf 94:109–119

3. Jahanpour J, Alizadeh MR (2015) A novel acc-jerk-limited NURBS interpolation enhanced with an optimized S-shaped quintic feedrate scheduling scheme. Int J Adv Manuf Technol 77(9):1889–1905
4. Bharathi A, Dong JY (2016) Feedrate optimization for smooth minimum-time trajectory generation with higher order constraints. Int J Adv Manuf Technol 82(5):1029–1040
5. Altintas Y, Erkorkmaz K (2003) Feedrate optimization for spline interpolation in high speed machine tools. CIRP Ann-Manuf Techn 52(1):297–302
6. Guo JX, Zhang Q, Gao XS, Li H (2015) Time optimal feedrate generation with confined tracking error based on linear programming. J Syst Sci Complex 28:80–95
7. Zhang Q, Li SR, Guo JX, Gao XS (2015) Tractable algorithm for robust time optimal trajectory planning of robotic manipulators under confined Torque. Int J Comput Commun Control 10: 123–135
8. Zhang Q, Li SR, Guo JX, Gao XS (2015) Time-optimal path tracking for robots under dynamics constraints based on convex optimization. doi:10.1017/S0263574715000247
9. Snchez-Reyes J, Chacn JM (2015) A polynomial Hermite interpolant for c^2 quasi arc-length approximation. Comput Aided Des 62:218–226
10. Erkorkmaz K (2015) Efficient fitting of the feed correction polynomial for real-time spline interpolation. ASME J Manuf Sci Eng 137(4):MANU-14-1119
11. Tsai MS, Nien HW, Yau HT (2010) Development of a real-time look-ahead interpolation methodology with spline-fitting technique for high-speed machining. Int J Adv Manuf Technol 47(5):621–638
12. Wang JB, Yau HT (2009) Real-time NURBS interpolator: application to short linear segments. Int J Adv Manuf Technol 41(11):1169–1185
13. Wang YS, Yang DS, Liu YZ (2014) A real-time look-ahead interpolation algorithm based on Akima curve fitting. Int J Mach Tools Manuf 85:122–130
14. Yang ZY, Shen LY, Yuan CM, Gao XS (2015) Curve fitting and optimal interpolation for CNC machining under confined error using quadratic B-splines. Comput Aided Des 66: 62–72
15. Zhang LB, You YP, He J, Yang XF (2011) The transition algorithm based on parametric spline curve for high-speed machining of continuous short line segments. Int J Adv Manuf Technol 52(1):245–254
16. Zhao H, Zhu LM, Ding H (2013) A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments. Int J Mach Tools Manuf 65:88–98
17. Dong JC, Wang TY, Li B, Ding YY (2014) Smooth feedrate planning for continuous short line tool path with contour error constraint. Int J Mach Tools Manuf 76:1–12
18. Sencer B, Ishizaki K, Shamoto E (2015) 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(9):1977–1992
19. Sun SJ, Lin H, Zheng LM, Yu JG, Hu Y (2016) A real-time and look-ahead interpolation methodology with dynamic B-spline transition scheme for CNC machining of short line segments. Int J Adv Manuf Technol 84(5):1359–1370
20. Fan W, Lee CH, Chen JH (2015) A realtime curvature-smooth interpolation scheme and motion planning for CNC machining of short line segments. Int J Mach Tools Manuf 96:27–46
21. Dong JJ, Stori JA (2004) Optimal feed-rate scheduling for high-speed contouring. ASME J Manuf Sci Eng 129(1):63–76
22. Sencer B, Ishizaki K, Shamoto E (2015) High speed cornering strategy with confined contour error and vibration suppression for CNC machine tools. CIRP Ann-Manuf Techn 64(1):369–372
23. Zhang LX, Sun RY, Gao XS, Li HB (2011) High speed interpolation for micro-line trajectory and adaptive real-time look-ahead scheme in CNC machining. Sci China Technol Sci 54:1–15
24. Li HB, Gao XS, Zhang LX, Sun RY (2012) Discrete interpolation of g01 codes in 2D machining under bounded accelerations. Math Comput Sci 6(3):327–344
25. Li HB (2013) Estimation and control of the geometric error in a linear interpolator with parabola blending. ASME 2013 International Mechanical Engineering Congress and Exposition, San Diego
26. Du X, Huang J, Zhu LM (2015) A complete S-shape feed rate scheduling approach for NURBS interpolator. J Comput Des Eng 2(4):206–217
27. Zhang K, Yuan CM, Gao XS (2013) Efficient algorithm for time-optimal feedrate planning and smoothing with confined chord error and acceleration. Int J Adv Manuf Technol 66(9): 1685–1697