IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 31, 2019, accepted April 26, 2019, date of publication May 3, 2019, date of current version May 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914509

S-Model Speed Planning of NURBS Curve Based on Uniaxial Performance Limitation

JIANGANG LI[1], (Member, IEEE), YE LIU[1], YANAN LI[2], (Member, IEEE), AND GANGGANG ZHONG[1]

[1]School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518000, China
[2]School of Engineering and Informatics, University of Sussex, Brighton BN1 9RH, U.K.

Corresponding author: Jiangang Li (jiangang_lee@163.com)

This work was supported in part by the Shenzhen Basic Research Program under Grant JCYJ20150731105106111, and in part by the Shenzhen Key Lab for Advanced Motion Control and Modern Automation Equipment.

ABSTRACT As more complex curves are used in current productions, curve speed planning has become a key technique to overcome the bottleneck of high-speed and high-precision computerized numerical control (CNC) systems. This paper first establishes the model of particle velocity, acceleration, and jerks in the Cartesian coordinate system, and then improves the uniaxial performance limit algorithm. We set up a real-time speed planning process of spline curves, design an S-model look-ahead algorithm, simplify the S-model speed planning algorithm, and achieve real-time non-uniform rational B-splines (NUBRS) curve S-model speed planning based on uniaxial performance limitation. The simulation results show that all the actual interpolation velocity, acceleration, and jerk under the proposed method meet the preset single-axis limit. The experimental results show that the tracking performance under the proposed method has been significantly improved compared with that under the small line segments method. Compared with the NUBRS curve trapezoidal model speed planning, the vibration spikes during machining can be eliminated.

INDEX TERMS NURBS curve speed planning, S model speeding planning, unaxial performance limitation.

I. INTRODUCTION

In CNC machining, speed planning has a direct impact on the subsequent servo control, which is one of the key technologies that determine the machining efficiency and precision of the machine tool. The input of speed planning is the cutter location (CL) path file given by computer aided manufacturing (CAM) software, which usually uses the small line segments to approximate a curve. On the one hand, there is an inevitable bow-height error when approaching a curve, and when the trajectory of the curve is complex a large number of small line segments are needed to achieve high fitting accuracy, resulting in a heavy burden of data transmitting between the CAM software and CNC system. On the other hand, the curve will lose information after approximation by small segments, which will bring certain difficulties to the subsequent high-speed and high-precision speed planning. Non-uniform rational B-splines (NURBS) can perfectly construct various free curves due to their flexibility. Using NURBS curves can greatly reduce the amount of numerical control (NC) codes. A new generation of CAM software already supports direct output of NURBS curve trajectories. Recently, there have been a large number of research works devoted to trajectory optimization, in order to replace the traditional small line segments by continuous smooth optimal curves.

Many researchers carry out relevant works on acceleration and deceleration of NURBS curve speed planning and look-ahead algorithms. In [1], a constant velocity NURBS curve speed planning was proposed, which laid the foundation for subsequent related studies. In [2]–[4], acceleration and deceleration control was added to speed planning, but the acceleration and deceleration method was applied to only the first and last points of the curve. An adaptive acceleration and deceleration control strategy was proposed in [5], which was only based on bow-height error. This method was applied in [6], with trapezoidal forward prediction and processing. In [7], sensitive points were found out and used as dividing points for trapezoidal model acceleration and deceleration control. In [8], a look-ahead algorithm was proposed which accounted the uniaxial velocity and acceleration limits in the trapezoidal look-ahead algorithm. In order to

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

reduce the velocity jump in trapezoidal velocity planning, the exponential speed programming model was studied in [9] and [10], which reduced the effect to the machine tool by a sudden acceleration change but it still could not achieve a continuous change of the acceleration. An S-type speed model was proposed in [11], which achieved a continuous change of acceleration but did not specify how to calculate the intermediate deceleration point. In [12]–[17], abundant effort was made on improving the S-model speed planning of the curve. In [18], 17 kinds of acceleration and deceleration conditions were discussed based on the S-type speed model to find the deceleration points. In order to simplify the S-model speed planning, a speed-planning node separation method was developed to segment the independent S-type speed path in [19]. According to the machine tool's velocity, acceleration and jerk of the NURBS curve of the S-model speed planning, and using the S-model look-ahead window deceleration calculation, pre-processing was moved to the CAD/CAM system to achieve real-time speed planning and interpolation in [15]. In this method, a synthesis value was used for the velocity, acceleration and jerk limit, so it may not be able to meet the requirements of the machine tool's uniaxial performance. In [20], axis-based look-ahead NURBS interpolation (ALANI) was conducted through detection-correction of the pre-interpolation process to meet the uniaxial velocity, acceleration and jerk limit, without consideration of the S-model speed planning. In [21], a S-type speed planning method was developed to account single-axis velocity and uniaxial acceleration, but the objective was oriented to a small line segment set.

Based on above discussions, this paper develops an uniaxial performance limit algorithm which can control the synthesized velocity, trajectory acceleration and the change rate of trajectory acceleration in presence of the single-axis velocity, acceleration and jerk limits. This algorithm is combined with S-model speed planning to form a S-model NURBS curve look-ahead algorithm, which achieves the given trajectory acceleration and its change rate. Moreover, we simplify the conditions of the S-type speed planning model and eventually realize real-time S-model speed planning with the uniaxial velocity, acceleration, jerk limits accounted.

The rest of the paper is organized as follows. The uniaxial performance limit algorithm is introduced in Section 2. Section 3 describes the design of the speed planning process and details the S-model speed planning with consideration of the uniaxial performance limits. In Section 4, the proposed S-model speed planning algorithm is tested by simulations. In Section 5, its performance in tracking and vibration attenuation is demonstrated through comparative experiments.

II. UNIAXIAL PERFORMANCE LIMIT ALGORITHM

The core of trajectory planning of machine tool is to determine the tip point position when a trajectory to be machined is given. The variable that can be controlled in the trajectory planning process is the parameter in a model that describes the particle motion. In particular, the particle's velocity, acceleration, and jerk are modeled in Fig. 1, which are respectively defined as

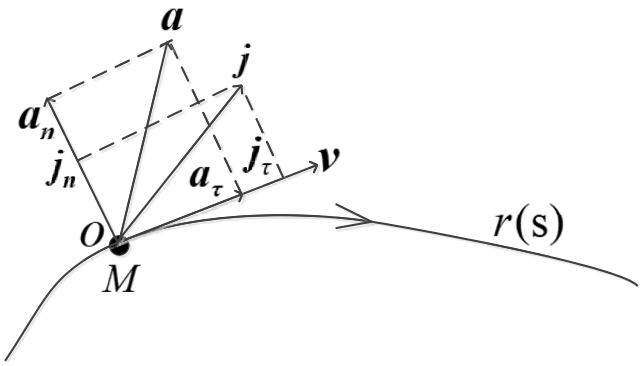


FIGURE 1. Particle motion.

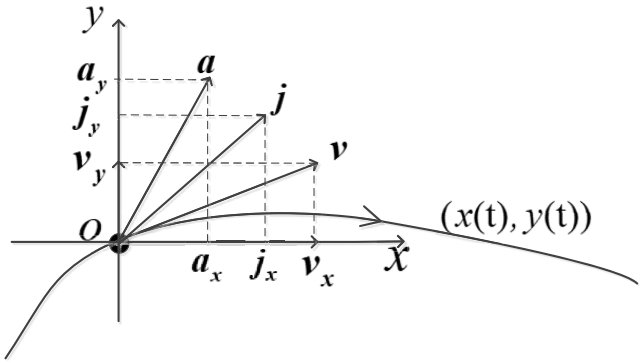


FIGURE 2. Particle motion in Cartesian coordinate system.

$$v = v_\tau \tau \tag{1}$$

$$a = \frac{dv_\tau}{dt}\tau + \frac{{v_\tau}^2}{R}n \tag{2}$$

$$j = \left(\frac{d^2 v_\tau}{dt^2} - \frac{{v_\tau}^3}{R^2}\right)\tau + \left(\frac{v_\tau}{R}\right)\left(\frac{3v_\tau}{dt} - \frac{v_\tau\left(\frac{dR}{dt}\right)}{R}\right)n \tag{3}$$

where $v_\tau = |v|$, τ is the unit tangent vector, n is the unit normal vector and R is the radius of curvature.

The single-axis performance limits considered in this paper include uniaxial velocity, acceleration and jerk, which are described in Cartesian coordinate system. The modeling of particle motion in Cartesian coordinate system is shown in Fig. 2. The corresponding velocity, acceleration and jerk are respectively described as

$$v = \frac{dx}{dt}i_x + \frac{dy}{dt}i_y \tag{4}$$

$$a = \left(\frac{d^2 x}{dt^2}\right)i_x + \left(\frac{d^2 y}{dt^2}\right)i_y \tag{5}$$

$$j = \left(\frac{d^3 x}{dt^3}\right)i_x + \left(\frac{d^3 y}{dt^3}\right)i_y \tag{6}$$

where i_x and i_y are respectively unit vectors in x and y directions.

The main idea of the uniaxial performance limit algorithm is to limit the synthesized velocity, tangential and normal

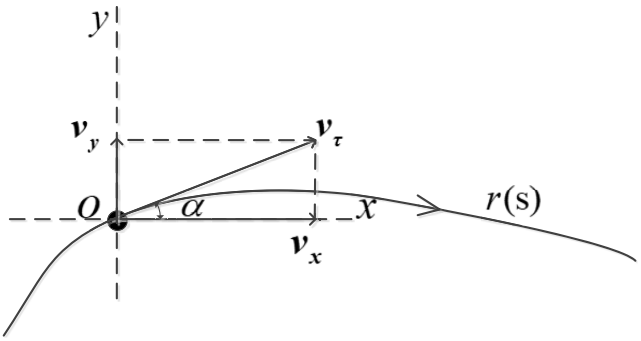


FIGURE 3. Uniaxial velocity.

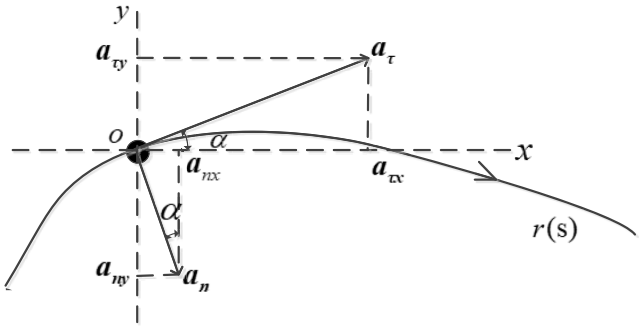


FIGURE 4. Uniaxial acceleration.

acceleration, tangential and normal jerk below the uniaxial velocity, acceleration and jerk in Cartesian coordinate system. For this purpose, we need to transform the limits of synthesized velocity, tangential and normal acceleration, tangential and normal jerk to the limits of synthesized velocity, tangential acceleration and change rate of tangential acceleration which are controllable in the following speed planning.

A. PERFORMANCE LIMITS FOR UNIAXIAL VELOCITY

The modeling of velocity is shown in Fig. 3. To establish the relationship between the uniaxial velocity and the synthesized velocity, we have

$$\begin{cases} v_x = |v_\tau| k_x \\ v_y = |v_\tau| k_y \end{cases} \tag{7}$$

where

$$\begin{cases} k_x = cos\alpha \\ k_y = sin\alpha \end{cases} \tag{8}$$

where α is the angle between the tangential direction and x axis.

Suppose the uniaxial velocity of the machine is limited to $v_{x,max}$ and $v_{y,max}$. Then, the limit of the synthesized speed v_τ is given as

$$\begin{cases} v_{mx} = \dfrac{v_{x,max}}{k_x} \\ v_{my} = \dfrac{v_{y,max}}{k_y} \end{cases} \tag{9}$$

Therefore, the synthesized velocity limited by uniaxial velocity is

$$v_{mxy} = min\left\{v_{mx}, v_{my}\right\} \tag{10}$$

B. PERFORMANCE LIMITS FOR UNIAXIAL ACCELERATION

The modeling of acceleration is shown in Fig. 4. To establish the relationship between the uniaxial acceleration and the tangential and normal acceleration, we have

$$\begin{cases} a_x = |a_\tau| k_x + |a_n| h_x \\ a_y = |a_\tau| k_y + |a_n| h_y \end{cases} \tag{11}$$

where

$$\begin{cases} k_x = cos\alpha \\ h_x = sin\alpha \\ k_y = sin\alpha \\ h_y = cos\alpha \end{cases} \tag{12}$$

Suppose the uniaxial acceleration of the machine is limited to $a_{x,max}, a_{y,max}$. In addition to the path acceleration (tangential acceleration), normal acceleration also has an effect on curved contours. If this is not taken into account during parametrization of the path parameters, the effective axial acceleration during acceleration and deceleration on the curved contour can, for a short time, reach 2 times of the maximum value. Therefore, we consider the effective acceleration as the sum of the path acceleration and normal acceleration.

Influence of path curvature on dynamic path response can be used to set the proportion of the axis-specific acceleration that is to be taken into account for normal acceleration. To simplify the calculation, the limitation is transformed to the limitations of the tangential acceleration a_t and normal acceleration a_n, with the following formulas:

$$A = min\left\{a_{x,max}, a_{y,max}\right\} \tag{13}$$

$$a_{n,max} = AM \tag{14}$$

$$a_{t,max} = A(1 - M) \tag{15}$$

where A is the minimum of the allowable maximum accelerations for each single axis. We use M to represent the radial acceleration coefficient, whose value is obtained by experiments for a machine tool. $a_{t,max}$ can be used directly as a controllable performance parameter in the following speed planning but for $a_{n,max}$, we need to transform it to the limitation of the synthesized velocity, i.e.

$$v_{max} = \sqrt{a_{n,max} R} \tag{16}$$

where R is the radius of curvature at this point.

In summary, the limitation of uniaxial acceleration is expressed as the arc acceleration limitation described in equation (15) and the synthesized speed limitation in equation (16).

C. PERFORMANCE LIMITS FOR UNIAXIAL JERK

Suppose the uniaxial jerk of the machine is limited by $j_{x,max}, j_{y,max}$. Similarly as in dealing with the uniaxial acceleration limit, we extend the use of the M parameter obtained by experiments, then the tangential and normal jerk limitation can be given as

$$J = min\left\{j_{x,max}, j_{y,max}\right\} \tag{17}$$

$$j_{n,max} = JM \tag{18}$$

$$j_{t,max} = J(1-M) \tag{19}$$

The tangential and normal jerk limitation cannot directly guide the speed planning process, so they need to be further transformed. The tangential jerk limitation is given by

$$|\frac{d^2 v_\tau}{dt^2} - \frac{{v_\tau}^3}{R^2}| \le j_{t,max} \tag{20}$$

We see from the above equation that the tangential jerk includes two parts: one in tangential direction of the change of the tangential acceleration and the other in tangential direction of the change of the normal acceleration. The part that can directly guide the speed planning is the change rate of the tangential acceleration, given by

$$j_a = \ddot{v} \tag{21}$$

Then, the limitation of tangential jerk in equation (20) can be transformed to

$$|j_a - \frac{v^3}{R^2}| \le j_{t,max} \tag{22}$$

Then, we can obtain

$$\frac{v^3}{R^2} - j_{t,max} \le j_a \le j_{t,\ max} + \frac{v^3}{R^2} \tag{23}$$

The positive and negative limits of $j_{a,max}$ should be the same, so we have

$$\frac{v^3}{R^2} - j_{t,max} \le -j_{a,\ max} \le j_a \le j_{a,\ max} \le j_{t,\ max} + \frac{v^3}{R^2} \tag{24}$$

We obtain

$$\begin{cases} j_a \le j_{t,max} \\ j_{a,max} \le j_{t,\ max} - (\frac{v^3}{R^2})^3 \le j_{t,\ max} - \frac{v^3}{R^2} \end{cases} \tag{25}$$

If $(\frac{v^3}{R^2})_{max} \ge j_{t,max}$, then

$$\begin{cases} j_{a,\ max} = 0 \\ v_{mjt} = (\sqrt[3]{j_{t,\ max} R^2})_{max} \end{cases} \tag{26}$$

If $(\frac{v^3}{R^2})_{max} \le j_{t,max}$, then

$$j_{a,max} = j_{t,\ max} - (\frac{v^3}{R^2})_{max} \tag{27}$$

Therefore, the normal jerk limitation is given as

$$|\frac{v_\tau}{R}(\frac{3dv_\tau}{dt} - \frac{v_\tau}{R}\frac{dR}{dt})| \le j_{n,max} \tag{28}$$

By expansion of $\frac{dR}{dt}$, we have

$$\frac{dR}{dt} = \frac{dR}{ds}\frac{ds}{dt} = v_\tau \frac{dR}{ds} = -(\frac{v_\tau}{k^2})\frac{dk}{ds} \tag{29}$$

Substituting equation (29) to equation (28), we can obtain

$$|3kv_\tau \frac{v_\tau}{dt} + \frac{dk}{ds}v^3| \le j_{n,max} \tag{30}$$

Then,we have

$$\begin{cases} v_\tau^3 \frac{dk}{ds} + 3kv_\tau \frac{dv_\tau}{dt} - j_{n,max} \le 0 \\ v_\tau^3 = \frac{dk}{ds} + 3kv_\tau \frac{dv_\tau}{dt} + j_{n,max} \ge 0 \end{cases} \tag{31}$$

When $\frac{dk}{ds} > 0$, $\frac{dv_\tau}{dt} \le 0$; when $k > 0$, the extreme situation is $a_t = \frac{dv_\tau}{dt} = 0$. To find v_{min}, equation (30) is simplified as follows

$$v_\tau^3 \frac{dk}{ds} - j_{n,max} \le 0 \tag{32}$$

When $k < 0$, the extreme situation is $a_t = \frac{dv_\tau}{dt} = -a_{t,max}$. To find v_{min}, equation (31) is simplified as follows

$$v_\tau^3 \frac{dk}{ds} - 3kv_\tau a_{t,max} - j_{n,\ max} \le 0 \tag{33}$$

When $\frac{dk}{dt} < 0$, $\frac{dv_\tau}{dt} \ge 0$; when $k > 0$, the extreme situation is $a_t = \frac{dv_\tau}{dt} = 0$. To find v_{min}, equation (31) is simplified as follows

$$v_\tau^3 \frac{dk}{ds} + j_{n,max} \ge 0 \tag{34}$$

When $k < 0$, the extreme situation is $a_t = \frac{dv_\tau}{dt} = a_{t,max}$. To find v_{min}, equation (31) is simplified as follows

$$v_\tau^3 \frac{dk}{ds} - 3kv_\tau a_{t,max} + j_{n,\ max} \ge 0 \tag{35}$$

As shown above, the limitation of uniaxial jerk is expressed by the synthesized speed limitation v_{min} and the rate of the change of tangential jerk.

III. S-MODEL LOOK-AHEAD ALGORITHM AND SPEED PLANNING

In order to simplify its speed planning, the flow chart of S-model speed planning process is shown in Fig. 5. Then, each part in the flow chart is explained respectively.

A. FEATURE POINTS SELECTION

The method for feature point selection is as follows: take point $u_0 = 0$ as the first feature point and also the starting searching point, then test if the point whose parameter increases by u_i comparing with the parameter of the previous feature point satisfies selection conditions; if yes, the point is set as the next feature point; otherwise, continue the searching process; until $u_i = 1$, take this point as the last feature point. The schematic diagram is shown in Fig. 6. Specifically, this paper focuses on the following aspects.

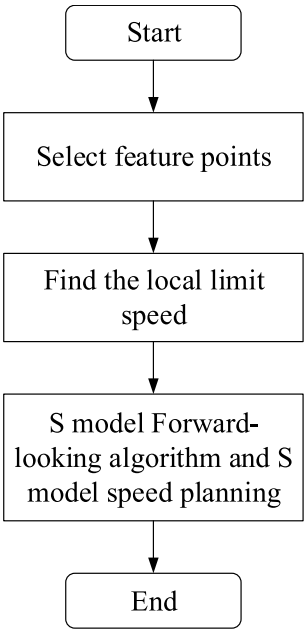


FIGURE 5. Flow chart of S-model speed planning.

FIGURE 6. Schematic diagram of feature point.

1) SAMPLING POINT INCREMENT U_I

The distribution of the increment u_i is related to the shape and length of the actual NURBS curve. Therefore, we adopt an adaptive method of finding the increment u_i according to the same chord error, which is determined by the following method: based on the chord difference and the curvature radius of the current point, the step length is obtained, and then the first-order Taylor interpolation algorithm is used to obtain the corresponding step length. In particular, we have

$$s = 2\sqrt{(r^2 - r - e^2)} \tag{36}$$

where s is the chord length between two sampling points, r is the radius of curvature at the current sample point and e is the default chord error limit. After obtaining the arc length increment, it needs to be further transformed to the increment of the NURBS curve parameter u_i, where the first-order Taylor expansion is adopted:

$$u_i = \frac{s}{|\frac{dc(u)}{du}|}|_{u=u_i} \tag{37}$$

2) CONDITIONS OF FEATURE POINTS

On the one hand, the extreme point of the curvature reflects the characteristics of the trajectory, so it should be set as the feature point and used as the condition of the highest priority. On the other hand, selecting only the extreme point of the curvature may make the fitting straight-line segments too long or too short, leading to a large allowable maximum speed difference between the fitting straight-line segments. Therefore, restrictions need to be added.

Feature points should be able to represent the limits of the allowable speed for the curve geometry within the nearby area. Whether or not a new feature point needs to be identified is based on whether the change of the speed limit of the curve geometry relative to the previous feature point speed limit reaches a threshold. In order to improve the efficiency, the speed limit by centripetal acceleration can be approximated as the speed limit of the curve geometry in this area, as below

$$v_{(i+1)m} = v_{im} + c \tag{38}$$

where c is the preset speed limit increment threshold. The centripetal acceleration limits the allowable speed to

$$v_m = \sqrt{a_{n,max} r} \tag{39}$$

Thus, we obtain

$$r_0 = c^2 + \frac{2c\sqrt{a_{n,max} r}}{a_{n,max}} \tag{40}$$

With the radius r at current point, the radius increment r_0 can be obtained. The value of c is determined by

$$c = k a_{t,\ max} T \tag{41}$$

where T is the machine interpolation cycle.

We also consider the maximum and minimum of the arc length: S_{max}, S_{min}. The limit S_{max} is set mainly to ensure the accuracy of the speed limitation, and S_{min} is set to prevent poor local curvature variation characteristics influencing the rationality of feature point selection.

The overall conditions are summarized as follows: when a point reaches the upper limit of the arc length, reaches the upper limit of the variation of the curvature radius or is the extreme point of the curvature, the point is selected as the candidate of the feature point; if the arc length of the candidate from the previous feature point satisfies the lower limit of the arc length, the point is selected as the feature point; otherwise, the previous feature point candidate is deleted, and the current point is added. The flow chart of feature point selection is shown in Fig. 7.

B. LOCAL SPEED LIMITATION FOR FEATURE POINTS

The local speed limitation for feature points is given by

$$v_{ml} = min\left\{v_{mxy}, v_{man}, v_{mjt}, v_{mjn}, F\right\} \tag{42}$$

The method to find v_{mxy} and v_{man} is known and the derivation of v_{mjt} and v_{mjn} has been carried out. It is important to note that: to find v_{mjt}, in the formula (27), $(\frac{v^3}{R^2})_{max}$ needs to be the maximum value when we take the two extreme points of the fitting straight-line segments into calculation; to find v_{min}, in equations (32), (33), (34) and (35), the value of $\frac{dk}{ds}$ needs to be approximately replaced by the value of k/s in the nearby area of feature points.

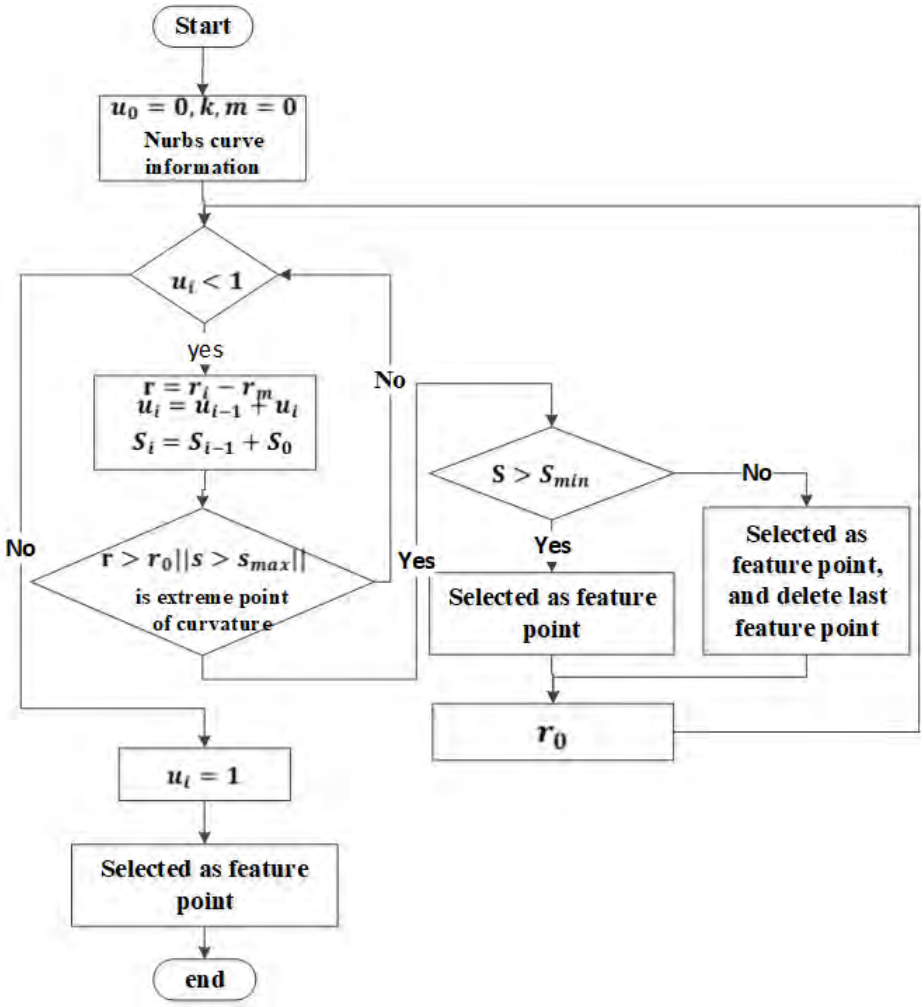


FIGURE 7. Flow chart of feature points selection.

C. S-MODEL LOOK-AHEAD AND SPEED PLANNING

1) S-MODEL LOOK-AHEAD

Take the acceleration process as an example. After the modeling of uniaxial acceleration and jerk, $a_{t,max}$ is obtained, and the value of $j_{a,maxi}$ for each segment is also known. Thus, for each segment, when the three-stage S-model speed planning is adopted, the relationship between the speed and time is determined, as shown in Fig. 8. In this figure, we have

$$\begin{cases} v_0 = v_{m(b)} - v_{m(a)} = \dfrac{a_{t,max}^2}{j_{a,maxi}} \\ s_0 = \dfrac{2a_{t,max}v_{m(a)}}{j_{a,maxi}} + \dfrac{a_{t,max}^3}{j_{a,maxi}^2} \end{cases} \tag{43}$$

After the pre-look-ahead, there is a pair of v and s for the segment between two feature points. When compared to v_0 and s_0, there are four cases as shown in Table 1 which are separated by the acceleration modes, as detailed in the following.

TABLE 1. Specific mode determination.

s,v	$v > v_0$	$v \leq v_0$
$s > s_0$	$flag1$	$flag2$
$s \leq s_0$	$flag3$	$flag4$

Flag10 After the acceleration value reaches $a_{t,max}$ by $j_{a,maxi}$, it does not go directly to deceleration, but to the uniform acceleration over a period of time. The time of uniform acceleration needs to be determined by s or v. It is important to note that: s will determine a uniform acceleration time t_1, and v will determine another uniform acceleration time t_2; generally speaking, $t_1 \neq t_2$, so we set the uniform acceleration time as $t_0 = min(t_1, t_2)$. If $t_0 = t_1$. Then, when the acceleration decreases to $a_\tau = 0$, $s = s_0$, which means that the tip will arrive at point b. Since the velocity at point b may be smaller than $v_{m(b)}$, we modify the local speed limit from $v_{m(b)}$ to $v_{t(b)}$, as shown in Fig.9a). If $t_0 = t_2$, then when the acceleration decreases to $a_\tau = 0$, $v = v_0$, which means that when the segment is finished, the velocity of the tip will

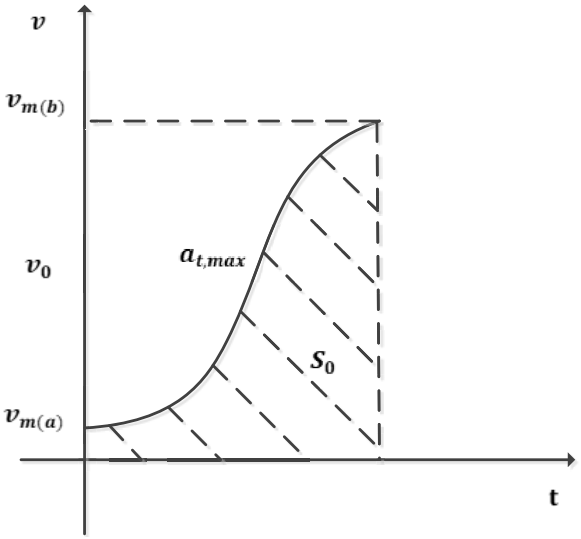


FIGURE 8. Sketch of v_0 and s_0.

be $v_{m(b)}$. As s may not equal to s_0, we add a uniform velocity section until the tip arrives at point b. The time of the uniform velocity section is supposed to be t_0, as shown in Fig.9b).

Next, we consider the specific S-model look-ahead steps for Flag1. First, we need to solve the following equations to find the uniform acceleration time:

$$(2v_{m(a)} + \frac{a_{t,max}^2}{j_{a,maxi}} + a_{t,maxi}t_1)(\frac{a_{t,max}}{j_{a,maxi}} + 0.5t_1) = s \quad (44)$$

$$a_{t,max}t_2 + \frac{a_{t,max}^2}{j_{a,maxi}} = v \quad (45)$$

Then, we need to compare the values of t_1 and t_2: if $t_1 < t_2$, then we correct the velocity at point b as below

$$v_{t(b)} = v_{m(a)} + a_{t,max}t_1 + \frac{a_{t,max}^2}{j_{a,maxi}} \quad (46)$$

Flag20 We consider that the acceleration value does not reach $a_{t,max}$ by $j_{a,maxi}$. Suppose that the maximum acceleration value is $a'_{t,max}$ which is determined by v. When this segment is finished, the tip may not arrive at point b, which means that s may be smaller than s_0. Therefore, we add a uniform velocity section at the end of the segment to meet the requirement of s_0, and the time of uniform velocity is t_0, as shown in Fig. 10. In this case, the speed of point b does not need to change.

Flag30 Since $s \leq s_0$, similarly as for flag2, we consider that the acceleration value does not reach $a_{t,max}$ by $j_{a,maxi}$. Suppose that the maximum acceleration value is $a'_{t,max}$, which is determined by s. As the velocity at point b may be smaller than $v_{m(b)}$, we modify the speed to $v_{t(b)}$ according to the local speed limit $v_{m(b)}$, as shown in Fig. 11.

To determine the value of $v_{t(b)}$, we need to determine the actual maximum acceleration $a'_{t,max}$. First, the value of t_1 is obtained by solving the following equation:

$$j_{a,maxi}t_1^3 + 2v_{m(a)}t_1 - s = 0 \quad (47)$$

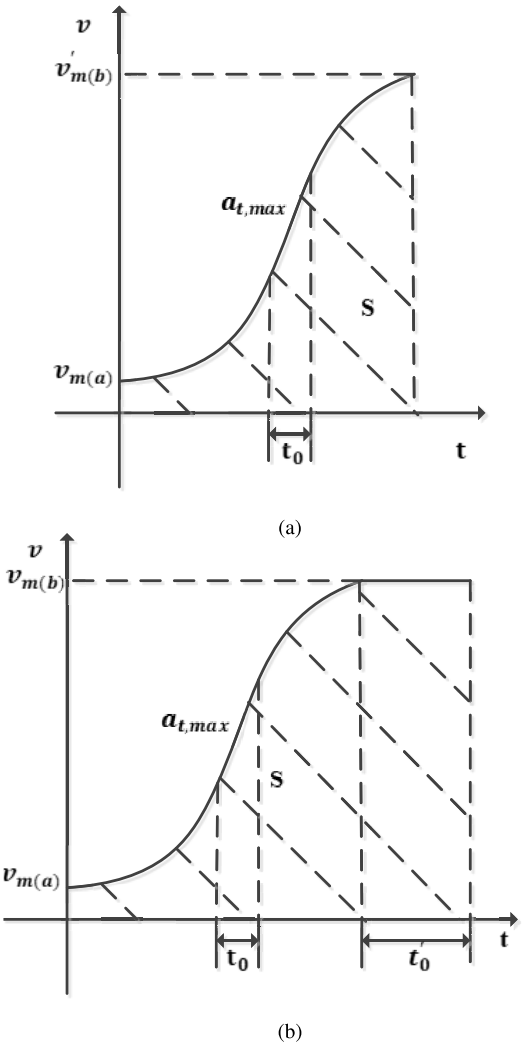


FIGURE 9. Schematic figure of flag10. (a) flag 100. (b) flag 101.

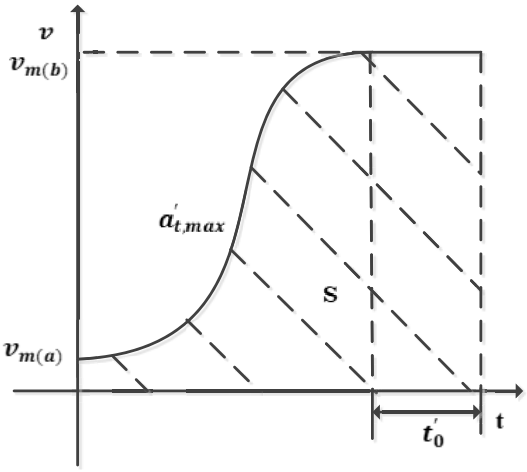


FIGURE 10. Schematic figure of flag20.

Then, the value of $v_{t(b)}$ can be determined as

$$v_{t(b)} = v_{m(a)} + j_{a,maxi}t_1^2 \quad (48)$$

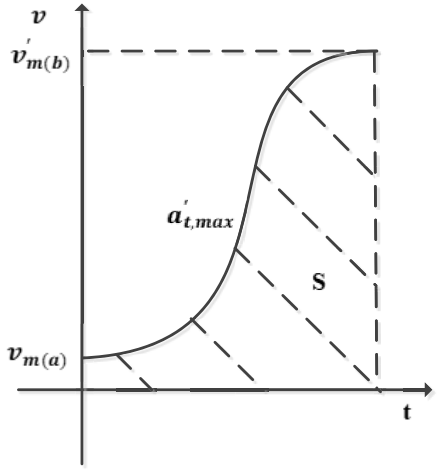


FIGURE 11. Schematic figure of flag30.

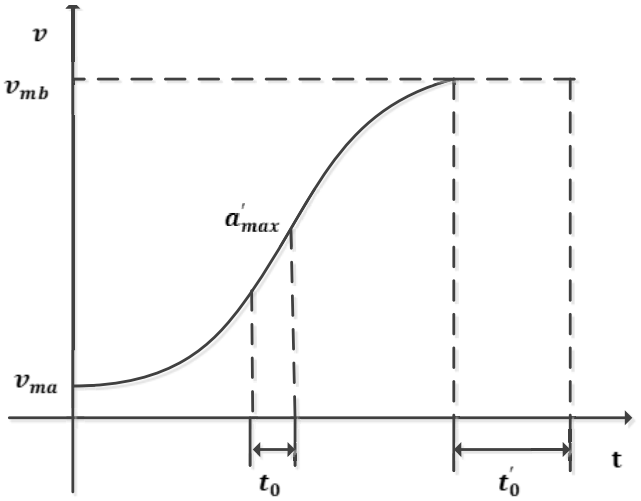


FIGURE 12. Acceleration process.

Flag40 Consider that the acceleration value does not reach $a_{t,max}$ by $j_{a,maxi}$, but the actual maximum acceleration is determined by s or v. Suppose that the maximum acceleration value determined by s is $a_{t,max1}$ and the value determined by v is $s_{t,max2}$, and the corresponding acceleration time are t_1 and t_2. Then, we let $t_0 = min(t_1, t_2)$. When $t_0 = t_1$, the actual maximum acceleration value $a_{t,max}$ is determined by s. Therefore, when the tip arrives at point b, the velocity may not reach $v_{m(b)}$, and we need to correct it to $v_{t(b)}$, as shown in Fig.11. When $t_0 = t_2$, the actual maximum acceleration value $a_{t,max}$ is determined by v, then to meet the requirement of s_0, we need to add a uniform velocity section at the end of segment. Suppose the time of the uniform velocity section is t_0, the schematic figure is shown as Fig. 10.

Now we consider the steps for the look-ahead method of Flag4. First, we solve the following equation to find out t_1 and t_2:

$$j_{a,maxi}t_1^3 + 2v_{m(a)}t_1 - s = 0 \tag{49}$$

$$a_{t,max}t_2 + \frac{a_{t,max}^2}{j_{a,maxi}} = v \tag{50}$$

Then, we need to compare t_1 with t_2. If $t_1 < t_2$, then the velocity of the point b is modified to

$$v_{t(b)} = v_{m(a)} + j_{a,maxi}t_1^2 \tag{51}$$

In summary, if the fitting straight line segment is an acceleration segment, the look-ahead algorithm needs to determine its specific mode and obtain the maximum global speed limit of point b according to the following rules:

- Flag100: $t_1 \le t_2$, $v_{t(b)} = v_{m(a)} + a_{t,max}t_1 + \frac{a_{t,max}^2}{j_{a,maxi}}$;
- Flag101: $t_1 \ge t_2$, $v_{t(b)} = v_{m(b)}$;
- Flag20: $v_{t(b)} = v_{m(b)}$;
- Flag30: $v_{t(b)} = v_{m(a)} + j_{a,maxi}t_1^2$;
- Flag400: $t_1 \le t_2$, $v_{t(b)} = v_{m(a)} + j_{a,maxi}t_1^2$;
- Flag401: $t_1 \ge t_2$, $v_{t(b)} = v_{m(b)}$.

As for the deceleration process, the look-ahead method is similar. The difference lies in the following two points. First, if the fitting straight line is a deceleration segment, the point which may need speed correction is point a, and when the uniform velocity segment needs to be added, the uniform velocity segment is added to point a. Specific rules are as follows:

- Flag110: $t_1 \le t_2$, $v_{t(a)} = v_{m(b)} + a_{t,max}t_1 + \frac{a_{t,max}^2}{j_{a,maxi}}$;
- Flag111: $t_1 \ge t_2$, $v_{t(a)} = v_{m(a)}$;
- Flag21: $v_{t(a)} = v_{m(a)}$;
- Flag31: $v_{t(a)} = v_{m(b)} + j_{a,maxi}t_1^2$;
- Flag410: $t_1 \le t_2$, $v_{t(a)} = v_{m(b)} + j_{a,maxi}t_1^2$;
- Flag411: $t_1 \ge t_2$, $v_{t(a)} = v_{m(a)}$.

Second, if the specific speed mode of current segment is flag110, flag31 or flag410, then we need to not only change the maximum allowable speed of point a, but also recalculate the speed planning mode of the previous segment after the maximum allowable speed of point a is updated until the corresponding speed planning mode is no longer flag110, flag31 or flag410, or until the preset number of segments of the preview is reached.

2) S-MODEL SPEED PLANNING

After the look-ahead algorithm is applied, for all kinds of fitting straight lines we can get the updated v, but do not need to consider s_0. We compare the actual v with v_0 for each segment, and then calculate the guidance information for the later interpolation process.

For the acceleration process, the actual speed-time curve is shown in Fig. 12. If $v > v_0$, the guidance information for later interpolation is as follows:

$$\begin{cases} a'_{max} = a_{max} \\ t_0 = \dfrac{v - \frac{a_{max}^2}{j_{a,maxi}}}{a_{max}} \\ t'_0 = \dfrac{s - 0.5(v_{ma} + v_{mb})(\frac{2a_{max}}{j_{a,maxi}} + t_1)}{v_{mb}} \end{cases} \tag{52}$$

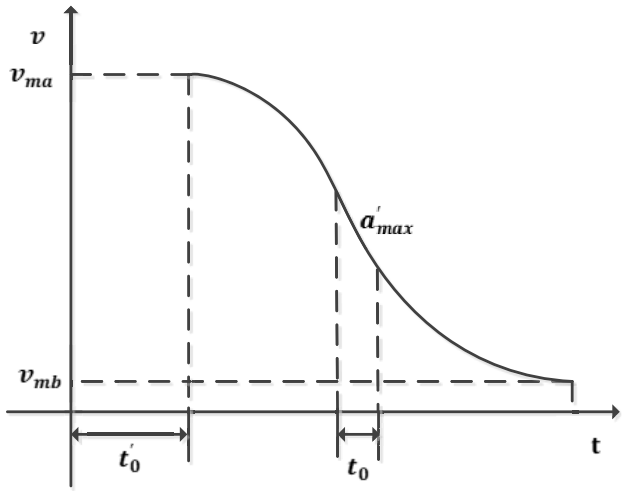


FIGURE 13. Deceleration process.

If $v < v_0$, the guidance information for later interpolation is as follows:

$$\begin{cases} a'_{max} = j_{a,maxi}\sqrt{\dfrac{v}{j_{a,maxi}}} \\ t_0 = 0 \\ t'_0 = \dfrac{s - (v_{ma} + v_{mb})\sqrt{\dfrac{v}{j_{a,maxi}}}}{v_{mb}} \end{cases} \tag{53}$$

For the deceleration process, the actual speed-time curve is shown in Fig. 13. If $|v| > v_0$, the guidance information for later interpolation is as follows:

$$\begin{cases} a'_{max} = -a_{max} \\ t_0 = \dfrac{-v - \frac{a_{max}^2}{j_{a,maxi}}}{a_{max}} \\ t'_0 = \dfrac{s - 0.5(v_{ma} + v_{mb})(\frac{2a_{max}}{j_{a,maxi}} + t_1)}{v_{ma}} \end{cases} \tag{54}$$

If $|v| < v_0$, the guidance information for later interpolation is as follows:

$$\begin{cases} a'_{max} = -j_{a,maxi}\sqrt{\dfrac{v}{j_{a,maxi}}} \\ t_0 = 0 \\ t'_0 = \dfrac{s - (v_{ma} + v_{mb})\sqrt{\dfrac{v}{j_{a,maxi}}}}{v_{ma}} \end{cases} \tag{55}$$

IV. SIMULATION RESULTS

In the simulation, we consider the butterfly trajectory as shown in Fig. 14. The default parameters are: $F = v_{x,max} = v_{y,max} = 30\text{mm/s}$, $a_{x,max} = a_{y,max} = 1000\text{mm/s}^2$, $j_{x,max} = j_{y,max} = 10000\text{mm/s}^3$, $A = 0.2$ and interpolation period is 0.001s. In order to facilitate the comparison of the advantages and disadvantages of the algorithm, we set the parameters to be the same in our algorithm and the algorithm by GOOGOL.

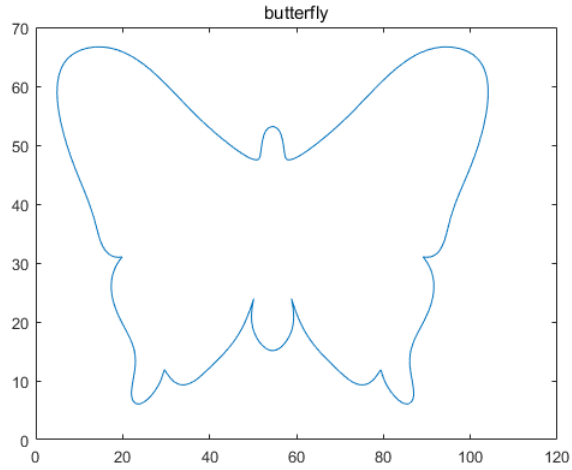


FIGURE 14. Tool path:butterfly.

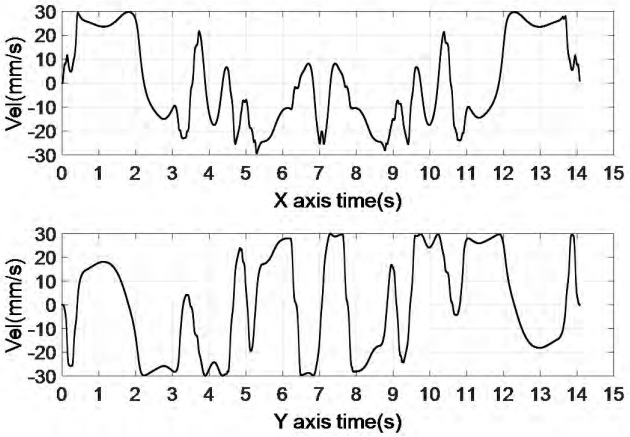


FIGURE 15. Uniaxial velocity.

After S-model speed planning, we use second-order Taylor interpolation to obtain the information of the final interpolation point. Multiple differential processing is performed on the obtained interpolation points to obtain the actual single-axis velocity, acceleration and jerk for each axis in each interpolation period after planning. The S-model speed planning and subsequent interpolation based on the uniaxial performance limits studied in this paper are implemented in VS software. The program input is the NUBRS curve and the output is the position-time (PT) information of the final interpolation point. The PT information is analyzed in MATLAB, resulting the velocity curve in Fig. 15, the acceleration curve in Fig. 16, and the jerk curve in Fig. 17. These figures show that the uniaxial velocity, acceleration and jerk of each interpolation cycle meet the preset requirements.

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL PLATFORM

The butterfly trajectory is considered with the same default parameters as in the simulation. The experimental platform used in this paper is shown in Fig. 18, which is a XY

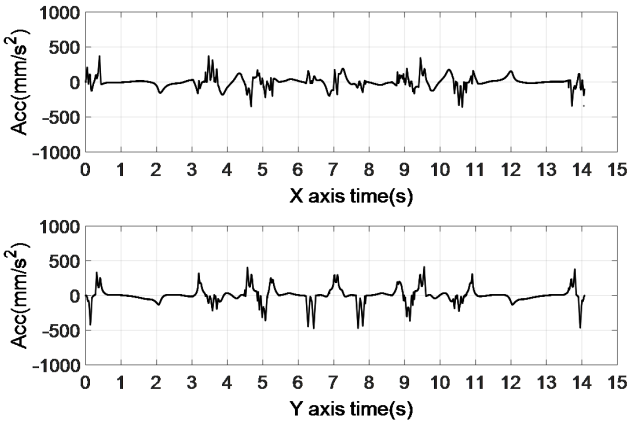


FIGURE 16. Uniaxial acceleration.

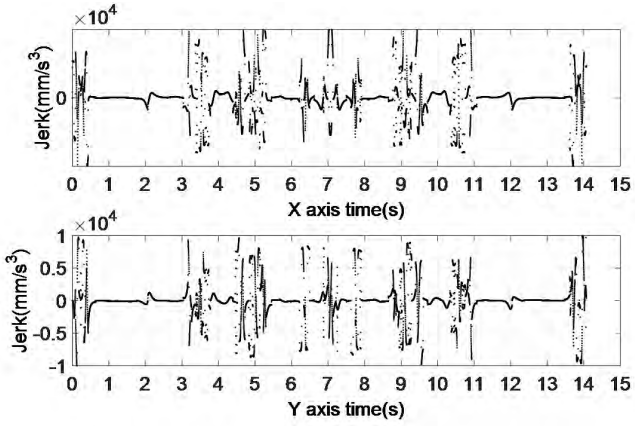


FIGURE 17. Uniaxial jerk.

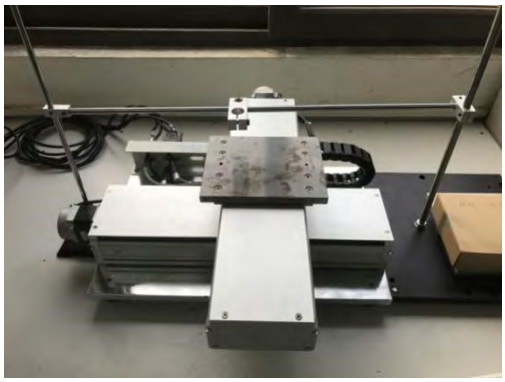


FIGURE 18. The experimental platform.

FIGURE 19. Vibration signal detector.

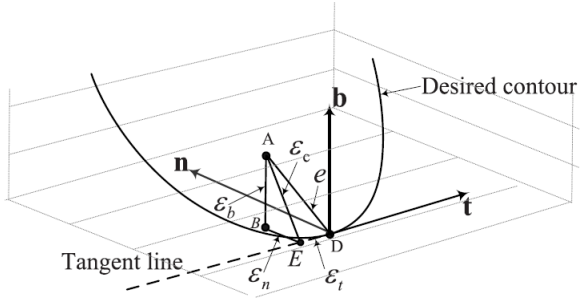


FIGURE 20. Contouring error in TCF.

double-axis screw table equipped with Tamagawa rotary motors and 17-bit absolute encoders. The motor driver is a GOOGOL GTHD servo driver controlled by the GTS-8000 control card. The algorithms are implemented in the host computer and the data acquisition is carried out by the control card. The Watch function in the GOOGOL control card can collect the pulse signal of the encoder, and can change the position information of the pitch of the ball screw by the pulse signal. The PT function of GOOGOL CNC system was used to obtain the PT information and to design the machine motion. During the experiment, the tracking performance was analyzed by collecting the information of the desired interpolation point of the platform and the actual interpolation point by the encoder when the machine was moving. A vibration detector was used for vibration signal acquisition, as shown in Fig. 19.

Two sets of comparative experiments are considered. The first one is the comparison between the S-model speed planning of NURBS curve and the new look-ahead algorithm of GOOGOL system to verify the superiority of the curve speed planning method compared with the small line segment method. The second is comparison between NURBS curve S-model speed planning and NUBRS curve trapezoid speed planning to verify that the vibration performance can be significantly improved by considering the uniaxial jerk limitation.

B. TRACKING PERFORMANCE: NURBS CURVE S-MODEL SPEED PLANNING VS. LOOK-AHEAD ALGORITHM OF GOOGOL SYSTEM

The algorithm we used to evaluate the contour accuracy can be found in [22] and [23]. As illustrated in Fig. 20, a task coordinate frame (TCF) is attached to the current desired position D. In linear approximation [23], the desired contour is locally approximated by the tangent line passing the desired position D. The distance from the actual position A to the tangent line is the estimated contouring error, denoted by ϵ_c

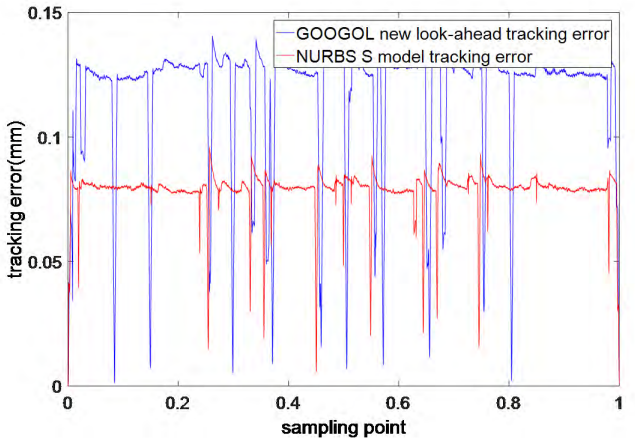


FIGURE 21. Tracking error under NURBS curve S-model speed planning has been greatly reduced compared to that under the new look-ahead algorithm of the GOOGOL system.

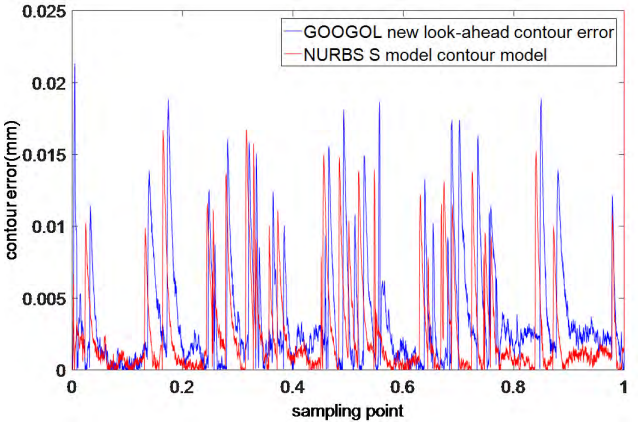


FIGURE 22. Contour error under NURBS curve S-model speed planning and the new look-ahead algorithm of GOOGOL system.

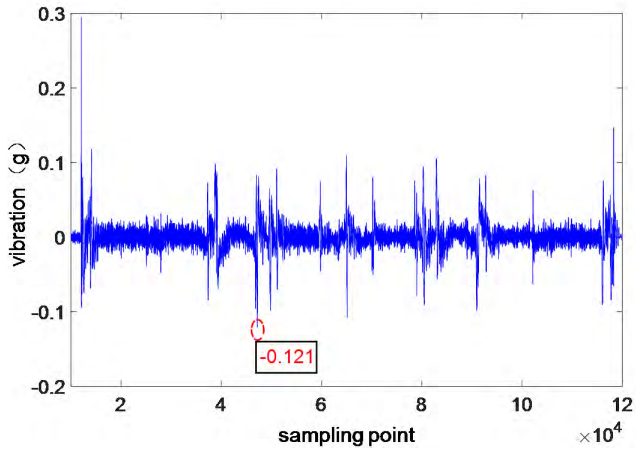


FIGURE 23. Vibration signals under NURBS curve trapezoidal model speed planning for butterfly trajectory.

in Fig. 20, and point E is the nearest position on the tangent line from A. Point B is the projection of A on the osculating plane, which is spanned by vectors t and n. The tracking error vectore $= \overrightarrow{AD}$ can thus be decomposed into components in the tangential, normal, and binormal directions, which are denoted by ϵ_t, ϵ_n, and ϵ_b, respectively. The tangent error ϵ_t is supposed to characterize an advancing performance and the estimated contouring error ϵ_c that characterizes a contouring performance is a composition of ϵ_n and ϵ_b, i.e.

$$\epsilon_c = \sqrt{{\epsilon_n}^2 + {\epsilon_b}^2} \tag{56}$$

New lookahead algorithm of GOOGOL system is developed by Shenzhen GOOGOL high-tech company, which uses small line blocks with the look-ahead algorithm [24]. In particular, small line blocks are used to represent the curves and lookahead algorithm to guarantee that the acceleration does not exceed the predefined value. In that way, the approximate optimal feedrate model is obtained.

In order to compare the tracking performance of each trajectory planning method at every position point on the trajectory, the sampling points are normalized. The resulting tracking error curves of NURBS curve S-model speed planning and the new look-ahead algorithm of GOOGOL system are shown in Fig. 21. It can be seen that the tracking error of NURBS curve S-model speed planning is significantly reduced compared with that of the new look-ahead algorithm of GOOGOL system. The maximum tracking error of the new look-ahead algorithm of GOOGOL system is 0.14mm and the average is 0.12mm, while the counterparts of S-model speed planning are 0.09mm and 0.08mm, respectively. The maximum and average tracking errors are reduced by 33%.

Further analysis of the contour error of the desired trajectory and the actual trajectory is conducted. The obtained contour error curve is shown in Fig. 22. The maximum contour error of the new look-ahead algorithm of GOOGOL system is 0.021mm and the average value is 0.004mm, while the counterparts of NURBS curve S-model speed planning are 0.017mm and 0.003mm, respectively. The maximum contour error is reduced by 21% and the average reduced by 29%.

C. VIBRATION PERFORMANCE: NURBS CURVE S-MODEL SPEED PLANNING VS. NURBS CURVE TRAPEZOID MODEL SPEED PLANNING

The vibration signals during the machining process of NURBS trapezoidal speed planning and S-model speed planning were collected and mapped in MATLAB and the resulting vibration curves are shown in Figs. 23 and 24, respectively.

Compared with the vibration signals under NURBS curve trapezoidal speed planning, the spikes in the middle section do not exist under the S-model speed planning. With the vibrations at the beginning and the end accounted, the maximum intermediate vibration is 0.04g, which is below 0.05g. These results demonstrate that the vibration performance has significant improvement.

As for the vibration spikes at the beginning and the end, we further applied S-model speed planning for a straight line.

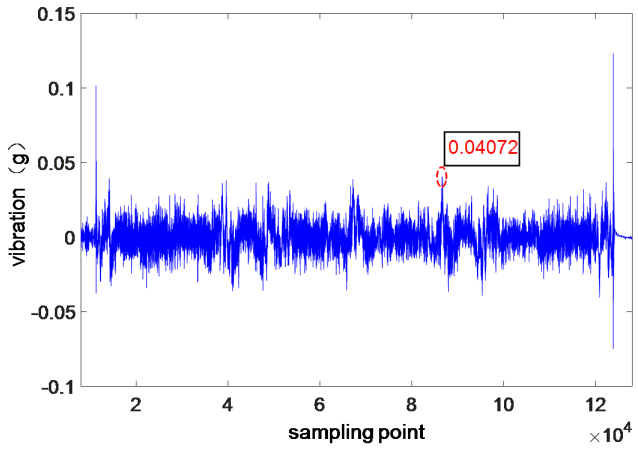


FIGURE 24. Vibration signals under NURBS curve S-model speed planning for butterfly trajectory.

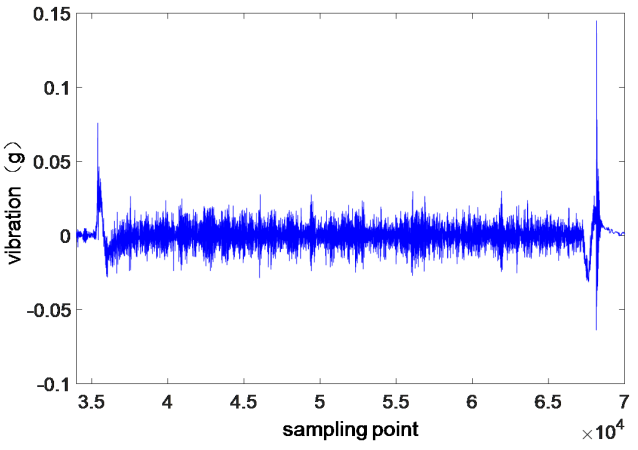


FIGURE 25. Vibration signals under NURBS curve S-model speed planning for straight line.

As shown in Fig. 25, similar vibration spikes still exist, which can be interpreted to be due to mechanical characteristics of the experimental platform and are irrelevant to the S-model speed planning algorithm in this paper.

Based on the above experimental results, we conclude that the tracking performance under NURBS curve S-model speed planning is significantly improved compared with that under small line segment planning. In terms of vibration, the performance under the NURBS curve S-model speed planning is better than that under the NURBS curve trapezoidal speed planning.

VI. CONCLUSIONS

The current continuous curve speed planning generally considers the limits of combined velocity, acceleration and jerk, so the planned speed curve may not meet the performance requirements of each single axis, leading to low machining accuracy and even damage to the machine tool. The traditional NURBS curve S-shape trajectory model speed planning has seventeen specific modes and requires complex deceleration point searching, so it is subject to a heavy computational burden. These problems were addressed in this paper, with the following results.

- Based on the kinematics of particle motions, the models of uniaxial velocities and combined velocities, uniaxial accelerations and tangential accelerations and centripetal accelerations, uniaxial jerk and tangential jerk and normal jerk were established, and the uniaxial performance limit algorithm was developed.
- The S-model look-ahead algorithm of NURBS curve was proposed, which greatly simplifies S-model speed planning. The NURBS curve S-model speed planning was combined with the uniaxial performance limit algorithm to realize real-time S-model speed planning that meets the uniaxial performance limitations.
- The simulation results showed that the actual velocity, acceleration and jerk of each axis under NURBS curve S-model speed planning satisfied the limits of each axis and met the preset design requirements. The experimental results showed that NURBS curve S-model speed planning had a significant performance improvement of about 30% compared with that under small line segment speed planning. Moreover, NURBS curve S-model speed planning could completely eliminate the vibration peaks that were found under NURBS curve trapezoidal speed planning.

REFERENCES

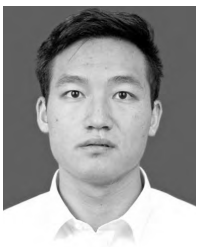
[1] Q. G. Zhang and R. B. Greenway, "Development and implementation of a NURBS curve motion interpolator," *Robot. Comput.-Integr. Manuf.*, vol. 14, no. 1, pp. 27–36, Feb. 1998.
[2] M.-Y. Cheng, M.-C. Tsai, and J.-C. Kuo, "Real-time NURBS command generators for CNC servo controllers," *Int. J. Mach. Tools Manuf.*, vol. 42, no. 7, pp. 801–813, May 2002.
[3] C.-W. Cheng and M.-C. Tsai, "Real-time variable feed rate NURBS curve interpolator for CNC machining," *Int. J. Adv. Manuf. Technol.*, vol. 23, nos. 11–12, pp. 865–873, Jun. 2004.
[4] Y. Sun, J. Zhou, and D. Guo, "Variable feedrate interpolation of NURBS Toolpath with geometric and kinematical constraints for five-axis CNC machining," *J. Syst. Sci. Complex.*, vol. 26, no. 5, pp. 757–776, Oct. 2013.
[5] S.-S. Yeh and P.-L. Hsu, "Adaptive-feedrate interpolation for parametric curves with a confined chord error," *Comput.-Aided Des.*, vol. 34, no. 3, pp. 229–237, Mar. 2002.
[6] D. Du, Y. Liu, C. Yan, and C. Li, "An accurate adaptive parametric curve interpolator for NURBS curve interpolation," *Int. J. Adv. Manuf. Technol.*, vol. 32, nos. 9–10, pp. 999–1008, Apr. 2007.
[7] T. Yong and R. Narayanaswami, "A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining," *Comput.-Aided Des.*, vol. 35, no. 13, pp. 1249–1259, Nov. 2003.
[8] L. Yinpeng and S. Wang, "Improved exponential acceleration and deceleration algorithm," *Mach. Tool Hydraul.*, vol. 1, no. 1, pp. 39–40, Jan. 2006.
[9] F. Peng, Y. He, and B. Li, "Look-ahead control in high feed rate NURBS curve interpolation," *J. Comput.-Aided Des. Comput. Graph.*, vol. 18, no. 5, pp. 625–629, May 2006.
[10] J. Xiong and W. Liu, "Judgment and calculation the deceleration point during the exponential curve acceleration and deceleration control," *Techn. Autom. Appl.*, vol. 32, no. 1, pp. 7–10, Jun. 2013.
[11] S.-H. Nam and M.-Y. Yang, "A study on a generalized parametric interpolator with real-time jerk-limited acceleration," *Comput.-Aided Des.*, vol. 36, no. 1, pp. 27–36, Jan. 2004.
[12] M.-T. Lin, M.-S. Tsai, and H.-T. Yau, "Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm," *Int. J. Mach. Tools Manuf.*, vol. 47, no. 15, pp. 2246–2262, Dec. 2007.
[13] K. Erkorkmaz and M. Heng, "A heuristic feedrate optimization strategy for NURBS toolpaths," *CIRP Ann.*, vol. 57, no. 1, pp. 407–410, 2008.

[14] M.-S. Tsai, H.-W. Nien, and H.-T. Yau, "Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery," *Comput.-Aided Des.*, vol. 40, no. 5, pp. 554–566, May 2008.
[15] M. Heng and K. Erkorkmaz, "Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability," *Int. J. Mach. Tools Manuf.*, vol. 50, no. 3, pp. 281–293, Mar. 2010.
[16] Y.-F. Chang, T.-G. Nguyen, and C.-P. Wang, "Design and implementation of look-ahead linear jerk filter for a computerized numerical controlled machine," *Control Eng. Pract.*, vol. 18, no. 12, pp. 1399–1405, Dec. 2010.
[17] W. Zhang, S. Gao, S. Zhang, and F. Zhang, "The research on NURBS adaptive interpolation technology," *AIP Conf. Proc.*, vol. 1834, no. 1, Apr. 2017, Art. no. 040021.
[18] H. Dong, J. Xie, and Z. Zhou, "An accurate NURBS curve interpolation algorithm with short spline interpolation capacity," *Int. J. Adv. Manuf. Technol.*, vol. 63, nos. 9–12, pp. 1257–1270, Dec. 2012.
[19] Z. Peng, P. Lou, M. Liu, and Z. Man, "Cnc machining velocity look-ahead strategy restricted by machine axis drive ability," *J. Xian Jiaotong Univ.*, vol. 46, no. 2, pp. 64–69, Feb. 2012.
[20] H. B. Liang, Y. Z. Wang, and L. I. Xia, "Research and implementation of nurbs interpolation algorithm for adaptive feed speed," *Comput. Integr. Manuf. Syst.*, vol. 12, no. 3, p. 428, 2006.
[21] L. Wang, "A look-ahead and adaptive speed control algorithm for high-speed CNC equipment," *Int. J. Adv. Manuf. Technol.*, vol. 63, nos. 5–8, pp. 705–717, Nov. 2012.
[22] R. Shi, Y. Lou, X. Zhang, and J. Li, "A novel task coordinate frame reduced- dimension 3-D contouring control," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1852–1863, Oct. 2018.
[23] G. T.-C. Chiu and M. Tomizuka, "Contouring control of machine tool feed drive systems: A task coordinate frame approach," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 1, pp. 130–139, Jan. 2001.
[24] J. Hu, L. Xiao, Y. Wang, and Z. Wu, "An optimal feedrate model and solution algorithm for a high-speed machine of small line blocks with look-ahead," *Int. J. Adv. Manuf. Technol.*, vol. 28, nos. 9–10, pp. 930–935, Jul. 2006.

JIANGANG LI received the B.Eng., M.Eng., and Ph.D. degrees from the Xi'an Jiaotong University, China, in 1999, 2002, and 2005, respectively. Since 2007, he has been an Associate Professor in control science and engineering with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, China. From 2015 to 2016, he was a Visiting Associate in computing and mathematical sciences with the California Institute of Technology. His general research interests include high speed and high-performance control system design, motion control, and motion planning.

YE LIU received the B.S. and M.S. degrees in control science from the Harbin Institute of Technology, in 2015 and 2017, respectively. Her research interests include motion control and motion planning.

YANAN LI received the B.Eng. and M.Eng. degrees from the Harbin Institute of Technology, China, in 2006 and 2008, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2013. From 2015 to 2017, he was a Research Associate with the Department of Bioengineering, Imperial College London, London, U.K. From 2013 to 2015, he was a Research Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. He is currently a Lecturer in control engineering with the Department of Engineering and Design, University of Sussex, Sussex, U.K. His current research interests include human–robot interaction, robot control, and control theory and applications.



GANGGANG ZHONG received the B.Eng. degree from Guangzhou University, in 2017. He is currently pursuing the degree with the Harbin Institute of Technology, Shenzhen, China. His research interests include trajectory planning and machine learning.