Abstract—NURBS is the standard model of CNC interpolation calculation at present. It is the basis of CNC system position and speed calculation. Especially, the algorithm to inverse calculate parameter of arc length determines the advantages and disadvantages of CNC speed planning. Due to the lack of analytical calculation relationship between the arc length of NURBS curve and its parameters, it is impossible to accurately calculate parameters through the arc length, which makes it difficult to accurately predict the interpolation points in speed planning. To solve the above problems, a high-precision inverse calculation method of NURBS curve parameters from arc length is proposed. Based on the rapid calculation of arc length by adaptive Simpson integral method, the corresponding relationship between arc length and parameters is established by weighting, and then the high-precision inverse calculation of arc length to parameters is realized by proportional optimization results. The example of curve speed planning shows that the algorithm not only has higher accuracy than other methods, but also can quickly calculate the parameters corresponding to any arc length, and can meet the requirements of high-speed and high-precision interpolation for high-performance CNC.

Keywords—NURBS, CNC interpolation, Parameter back calculation, Speed planning

I. INTRODUCTION

Interpolation is one of the core calculations of CNC system and it directly affects the accuracy and speed of the CNC system [1]. Early interpolation methods based on small line segment and arc are difficult to meet the needs of modern CNC system of high-performance. NURBS has been widely used as a basic model for real-time interpolating calculation in the world [2]. The calculation of NURBS involves control points, weights, knot vector and parameter. The purpose of calculating the parameter is to calculate the Cartesian spatial points on NURBS, so as to carry out later calculations of position and velocity. In the definitions of NURBS, the parameter is determined by a sequence of real numbers called knot vector mathematically. As is known, arc length and chord length parameter are widely used in description of parameter curves. Since the parameter of NURBS has no geometric significance, it is difficult to directly find the corresponding relationship between the parameter and the arc length or chord length of two interpolation points[3]. Therefore, algorithms to calculate parameter from the arc length have been concerned and adopted by many scholars in interpolation calculation.

The early algorithms for NURBS parameter back calculation mainly include the first-order Taylor’s expansion method and the second-order Taylor’s expansion method. The algorithm for calculating NURBS curve parameter based on chord length has large amount of calculation and low precision, which is difficult to meet the requirements of high-speed and high-precision interpolation of CNC. Recent studies on the inverse calculation of NURBS curve parameter mainly include: Shuai Ji[4] proposed a method of inverse calculation of parameters by equal chord length, which can obtain small velocity fluctuation under low calculation; Kang min[5] proposed a B-spline arc length parameterization method with C2 continuity, but it was not applied to NURBS curves; Xue Chengxi[6] proposed an extended unit incremental interpolation method for generalized NURBS curves, but the incremental interpolation method will cause cumulative error and lead to insufficient accuracy. Mo Chen[7] proposed an augmented Taylor interpolation method of B-spline based on the traditional Taylor interpolation method, but it will have a large amount of calculation in the application of NURBS. Zezhong C[8] generates NURBS tool path with arc length as parameter, which can obtain smooth surface in three-axis machining, and its accuracy is 10^-4, which is difficult to meet the requirements of high-performance CNC interpolation; Jia Chunyang[9] used curve discretization to obtain NURBS curve interpolation with equal arc length, and the error range is only 10^-2; W Zhao[10] proposed a generalized arc length increment method, which obtains the interpolation through the arc length increment in each interpolation cycle, but it will still produce cumulative error; Jichun Wu[11] fitted the curve by double arc method and obtained the corresponding relationship between arc length and parameters, so as to realize the inverse calculation of arc length to parameters. The calculation speed is fast, but the accuracy is not enough, resulting in large speed fluctuation; Zezhong C[12] realized the high-speed and high-precision machining of B-spline through the method of piecewise path, but it was not applied to NURBS; Michele Heng[13] proposed a numerically effective interpolation algorithm by applying the concept of feed correction polynomial to NURBS tool path, but polynomial fitting takes more time; Fang Kui[14] realized the parameterization of curve arc length by numerical method, which has high feasibility, but the experiment shows that the...
process of this method is unstable and the speed fluctuates greatly.

It is not difficult to see that the current research on NURBS parameter inverse calculation is mainly divided into two directions: chord length and arc length. The inverse calculation of parameters from chord length is convenient for the research on speed planning of CNC interpolation, but the chord length is variable, and the uniqueness of the result is greatly reduced. Only when the interpolation is subdivided into small enough, can we get higher precision results. Because the arc length is unique and the interpolation obtained by the inverse arc length parameters can obtain a smooth surface, it has natural advantages compared with the chord length method. Based on this, a method for inverse calculation of arc length parameters of NURBS curves is proposed in this paper. Experimental results show that this method can quickly calculate the parameters corresponding to the arc length of any NURBS curve, and has high accuracy.

II. RELATED WORK

A. NURBS curve definition

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file.

A NURBS curve of degree p can be expressed as:

$$C(u) = \frac{\sum_{i=0}^{n} \omega_i d_i N_{i,p}(u)}{\sum_{i=0}^{n} \omega_i N_{i,p}(u)}, u \in [0,1]$$

(1)

which:

$$N_{i,0} = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u)$$

0/0=0 is defined here.

In the above formula, u is the parameter of the curve, p is the number of NURBS curves, which also represents order p+1, and $$d_i(i = 0,1,\cdots,n)$$ represents the control point of the curve, $$\omega_i(i = 0,1,\cdots,n)$$ expressed as the ith weight factor. The knot vector of p-degree NURBS curve controlled by N+1 control points are:

$$U = [u_0, u_1\cdots ,u_n+1]$$

The first-order derivation of NURBS curves can be expressed as:

$$P'(u) = \sum_{i=0}^{n} \omega_i d_i N_{i,p}(u)$$

$$= \sum_{i=0}^{n} \omega_i d_i N_{i,p}(u) - \sum_{i=0}^{n} \omega_i N_{i,p}(u) \sum_{i=0}^{n} \omega_i d_i N_{i,p}(u)$$

(2)

B. Calculation of NURBS Arc length

Let the curve: $$P = P(u), u \in [a,b]$$, then the arc length of the curve can be expressed as:

$$s = \int_a^b \|P'(u)\| du$$

The first-order derivation of NURBS curve is obtained by equation (2). In this paper, Simpson integral method is selected to realize the integral operation through multiple iterations. The more iterations, the higher the accuracy. Simpson formula is as follows:

$$\int_a^b f(x)dx = \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

Because the Simpson integral operation is difficult to control the error, based on it, the adaptive Simpson[15] algorithm is adopted in this paper.

Adaptive Simpson divides the interval $$[a,b]$$ into two sections each time, and calculates the left and right intervals respectively by the three-point formula. The three-point Simpson formula is as follows:

$$S(a,b) = \int_a^b f(x)dx \approx \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

According to the input accuracy EPS, determine whether it meets the accuracy requirements by equation above:

$$|S(a,c) + S(c,b) - S(a,b)| < 15\text{EPS}$$

If equation above holds, it is judged that the set accuracy is met in the corresponding interval; otherwise, the interval that does not meet the accuracy is divided into two sections again, and the accuracy is reduced by half, and then it is judged whether it meets the conditions. This continues until all intervals meet the judgment conditions.

Using the adaptive Simpson algorithm to calculate the arc length of NURBS curve can not only ensure the accuracy, but also reduce repeated calculation, improve the calculation speed, and quickly calculate the arc length of parameter u between any sampling points.

III. NURBS ARC LENGTH PARAMETERIZATION METHOD

Set a parameter curve:

$$P = P(u), a \leq u \leq b$$
The interval \([a, b]\) where the parameter \(u\) is located is equally divided into \(N\) parts, and the sampling points of \(N+1\) Parameter \(u\) are obtained. The step between adjacent sampling points is taken as \(\Delta u\). Available:

\[
a = u_0 < u_1 < \ldots < u_n = b
\]

\[
\Delta u = \frac{b-a}{n}
\]

Calculate the arc length between each two sampling points \(ds_i\):

\[
ds_i = \int_{u_{i-1}}^{u_i} |P'(u)| du, i = 0, 1, \ldots, n - 1
\]

From \(ds_i\) calculate Arc length \(s_i\) between each sampling point \(u_i (i = 0, 1, \ldots, n)\) and \(u_0\).

\[
s_i = \sum_{j=0}^{i-1} ds_j, (i = 1, \ldots, n)
\]

For any parameter \(s \in [s_{i-1}, s_i]\), there are:

\[
u_n = u_{i-1} + \frac{s - s_{i-1}}{ds_{i-1}} \Delta u
\]

The above algorithm can preliminarily calculate the parameter UN corresponding to the arc length S. In this paper, the above algorithm is referred to as the decentralization method (the decentralization method is similarly introduced in literature [14]). Obviously, the parameter accuracy obtained by the decentralization method is difficult to meet the requirements of CNC interpolation and needs to be optimized. Figure 1 below is an example to analyze its shortcomings and give the optimization idea.

Fig. 1. Error analysis

In Figure 1, \(un\) is obtained by the above decentralization method. It is assumed that \(unew\) is the accurate value of the parameter corresponding to the arc length \(s\), and the arc length corresponding to arc AB is the arc length between the parameters \(u=un\) and \(u=unew\), so that the length of this arc length is \(ds\).

Firstly, the arc length \(sun\) corresponding to \(u=un\) is calculated by the adaptive Simpson integral method,

\[
sun = s_{i-1} + \int_{s_{i-1}}^{sun} |P'(u)| du
\]

\(ds\) is obtained from the following equation: (\(ds\) can be negative):

\[
ds = s - sun
\]

Considering that it is difficult to calculate the distance between the parameters \(un\) and \(unew\) on the arc AB in general, and when it degenerates into a straight line, the distance between \(un\) and \(unew\) can be accurately calculated through the proportional relationship between similar triangles. A parameter \(uc\) between \(un\) and \(unew\) can be found out according to the "differential triangle" relationship in equation above, and the parameter \(u\):

\[
u_c = un + \frac{ds}{2ds_i} \Delta u
\]

Let the parameter \(uc\) correspond to point D on the curve, as shown in Fig. 1. Firstly, the coordinates of the points corresponding to \(un\) and \(uc\):

\[
\begin{align*}
&[P(un_{i}), P(un_{i+1})] , \\
&[P(uc_{i}), P(uc_{i+1})]
\end{align*}
\]

and then the length of ad can be calculated as follows:

\[
|AD| = \sqrt{[P(uc_{i}) - P(un_{i+1})]^2 + [P(uc_{i+1}) - P(un_{i})]^2 + [P(uc_{i+1}) - P(un_{i})]^2}
\]

Make a straight line AC through point a and point d so that the length of AC is ds, that is:

\[
|AC| = ds
\]

Then the parameter \(u\) corresponding to point C can accurately approximate the exact value \(unew\), and there is the following proportional relationship:

\[
\frac{|AD|}{|AC|} = \frac{uc - un}{ur - un}
\]

By simplification:

\[
um = \frac{uc - un}{AD} \cdot AC + un = un + \frac{uc - un}{suc} \cdot ds \quad (3)
\]

The flow chart of realizing the above algorithm is shown in Fig. 2.
Get a NURBS curve \( P(u) \)

Bisection parameter interval
That is:
\[
0 - u_0 < u_1 < \ldots < u_n = 1
\]

Calculate the arc length of adjacent sections
\[
ds_i = \int_{u_{i-1}}^{u_i} |P'(u)| \, du, \quad i = 0, 1, \ldots, n - 1
\]

Input \( s, k = 0 \)

IV. EXPERIMENT AND COMPARISON

A. subject: Butterfly curve

This section takes the commonly used cubic NURBS curve as an example to realize the arc length parameterization algorithm by programming in ARX, verify the feasibility of the algorithm described in Section 2, and analyze the error of the results.

The experimental spatial NURBS curve, its control polygon and control vertex are shown in Fig. 3:

Because the butterfly NURBS curve drawn in Fig. 3(a) consists of four NURBS non closed curves, as shown in Fig. 3(b). Here, take the long first paragraph as an example for verification and speed planning.

There are 18 control points of the first NURBS curve. Using Hartley Judd [16] algorithm to calculate the knot vector, the knot vector of the cubic NURBS curve of these 18 control points is obtained as follows:
\[
U = [0, 0, 0, 0, 0.05635873, 0.12824176, 0.20256601, 0.29967945, 0.38319131, 0.45319495, 0.48849730, 0.51150269, 0.54680504, 0.61680869, 0.70032054, 0.79743398, 0.87175823, 0.94364126, 1, 1, 1, 1]
\]

There are 22 knot vectors in total, and the weight factor is 1. The first NURBS curve as shown in Fig. 3(b) can be drawn, and the total arc length of the curve is 37.8707857920mm.

B. experimental content: S-curve comparison

The feed speed of CNC has the process of acceleration, uniform speed and deceleration. In this experiment, the uniform speed is the maximum feed speed, which is about \( v = 0.05 \text{mm/ms} \). The interpolation period is taken as \( t = 1 \text{ms} \), so the arc length of each cycle is \( ds = v \cdot T \). In the acceleration stage, the curve is divided into several increasing arc length segments, several equal arc length segments in the uniform speed stage, and several decreasing arc length segments in the deceleration stage, so as to realize the process of acceleration, uniform speed and deceleration under the condition of equal cycle.

Select the sine function \( \sin \) to realize the S-shaped increase or decrease of speed, that is,
\[
v = 0.05(\sin(\theta - \frac{\pi}{2}) + 1) / 2(0 < \theta < \pi)
\]

the ideal speed curve is shown in Fig. 4:
According to the algorithm described in Section 2, the start parameters and end parameters corresponding to each arc are calculated, and the error is tested by calculating the arc length between two adjacent parameters.

### C. experimental results

#### Experiment 1.

The adaptive Simpson algorithm is used to calculate the arc length, the accuracy is $10^{-10}$mm, and the curve is equally divided into 100 segments, that is, the parameter step size is taken $\Delta u = 0.01$.

After 0.689s operation, 852 points are calculated. Limited to space, only a few comparison results are listed in Table 1.

<table>
<thead>
<tr>
<th>TABLE I. THE DATE OF $\Delta u = 0.01$</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter</td>
</tr>
<tr>
<td>u0=0</td>
</tr>
<tr>
<td>u1=0.000001157</td>
</tr>
<tr>
<td>u2=0.000005784</td>
</tr>
<tr>
<td>u3=0.000016189</td>
</tr>
<tr>
<td>u4=0.0352946975</td>
</tr>
<tr>
<td>u5=0.0364409374</td>
</tr>
<tr>
<td>u6=0.09642286815</td>
</tr>
<tr>
<td>u7=0.9653750559</td>
</tr>
<tr>
<td>u8=0.965200503</td>
</tr>
<tr>
<td>u9=0.000056822</td>
</tr>
</tbody>
</table>

In Table 1, the $u_0$-$u_{95}$ arc length increases gradually, which is the acceleration stage; The arc length of $u_{95}$-$u_{730}$ is equal, which is a constant velocity stage; $u_{757}$-$u_{852}$ arc length decreases gradually, which is the deceleration stage. After comparing the calculated theoretical value with the actual value point by point, the error curve is drawn by subtracting the actual value from the theoretical value, as shown in Figure 5:

### Fig. 4. ideal speed curve

Fig. 4. ideal speed curve

It can be seen that the error curve in Figure 5 is in a central symmetrical shape as a whole, the arc length error in the acceleration and deceleration stages at both ends is large, and the constant speed stage is relatively stable with a little fluctuation. But the overall error can be maintained at $3 \times 10^{-7}$mm.

#### Experiment 2.

$\Delta u = 0.001$ calculate the interpolation point.

The running time of the program is 0.294s, and a total of 852 points are calculated (some results are listed in Table 2).

### TABLE II. THE DATE OF $\Delta u = 0.001$

<table>
<thead>
<tr>
<th>parameter</th>
<th>Theoretical value (mm)</th>
<th>Actual value (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>u0=0</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>u1=0.000001157</td>
<td>0.000136686</td>
<td>0.000136686</td>
</tr>
<tr>
<td>u2=0.000005784</td>
<td>0.000546593</td>
<td>0.000546593</td>
</tr>
<tr>
<td>u3=0.000016189</td>
<td>0.001229274</td>
<td>0.001229274</td>
</tr>
<tr>
<td>u4=0.0352946975</td>
<td>0.0499863314</td>
<td>0.0499863314</td>
</tr>
<tr>
<td>u5=0.0364409374</td>
<td>0.0500000000</td>
<td>0.0500000000</td>
</tr>
<tr>
<td>u6=0.09642286815</td>
<td>0.0500000000</td>
<td>0.0500000000</td>
</tr>
<tr>
<td>u7=0.9653750559</td>
<td>0.0499920137</td>
<td>0.0499920137</td>
</tr>
<tr>
<td>u8=0.965200503</td>
<td>0.0499623812</td>
<td>0.0499623812</td>
</tr>
<tr>
<td>u9=0.000056822</td>
<td>0.0000056813</td>
<td></td>
</tr>
</tbody>
</table>

Error curve, as shown in Fig. 6:

### Fig. 5. $\Delta u = 0.01$ error diagram

Fig. 5. $\Delta u = 0.01$ error diagram

It can be seen that the error curve in Figure 5 is in a central symmetrical shape as a whole, the arc length error in the acceleration and deceleration stages at both ends is large, and the constant speed stage is relatively stable with a little fluctuation. But the overall error can be maintained at $3 \times 10^{-7}$mm.

### D. experimental features: high precision and stable speed

The algorithm studied in this paper mainly focuses on the calculation of arc length. The longer the arc length, the higher the accuracy and the longer the calculation time. From the experimental results, When the parameter step $\Delta u$ is taken as 0.01, the arc length of each section is greater than $\Delta u$ is taken as 0.01 long, and the calculation time of each arc length increases accordingly. With the same precision setting, $\Delta u = 0.001$ ratio $\Delta u = 0.01$ although the calculation times are increased, the time is shorter, which shows that the accuracy is increased and the time is shorter.

---

ZHENG Shuai et al. / Fast Precise and Smooth Approach to Calculate Parameter of NURBS by Arc-Length

IEEE Office - SEET Complex, Federal University of Technology, Owerri, Nigeria, [https://aeuso.org](https://aeuso.org)

EI ISSN: 2305-0543, PISSN: 2411-6173

5222
In order to investigate the stationarity of speed with increased accuracy, the chord length between each two interpolation points is divided by the period to obtain the speed of each period, and the actual speed curve is drawn, as shown in Fig. 7.

As a whole, the speed in Figure 7 shows three stages: acceleration, uniform speed and deceleration. In the acceleration and deceleration stages, there is no abnormality in the state of increasing and decreasing speed, but there are small speed fluctuations in the constant speed stage, as shown in the part framed by the ellipse in the Figure. Therefore, the velocity fluctuation rate in the uniform stage is calculated according to equation (4).

\[
\delta_i = \frac{v_{i+1} - v_i}{v_i} = (1 - \frac{|P(u_{i+1}) - P(u_i)|}{v_i T}) \times 100\% \quad (4)
\]

And draw the velocity fluctuation curve, as shown in Figure 8:

There are three error curves shown in Fig. 9, in which the error of each arc length of the first-order Taylor expansion method is less than \(3 \times 10^{-3} \text{mm}\); The error of second-order Taylor expansion development is also less than \(1 \times 10^{-3} \text{mm}\); The arc length parameterization method in reference [14] can control the arc length error within \(2 \times 10^{-4} \text{mm}\), but there are violent fluctuations in the whole process. According to the above method, the arc length error can be controlled within 3 when the parameter step is 0.01 and 0.001, For the case within respectively \(3 \times 10^{-5} \text{mm}\), \(2 \times 10^{-8} \text{mm}\), the method described in this paper has high accuracy.

In terms of speed fluctuation, the speed fluctuation rates of the three methods are shown in Fig. 10:
The three velocity volatility curves shown in Fig. 10, of which the maximum velocity volatility of the first-order Taylor method is about 1%; The velocity fluctuation rate of the second-order Taylor method is kept within 1%; The volatility of Fang Kui[14] method is slightly larger, and the maximum volatility is about 2%. As mentioned earlier, the velocity fluctuation rate of the method in this paper can be maintained within 0.04% and has high stability.

V. CONCLUSION

This paper introduces a method to reverse calculate the parameters from the arc length of NURBS curve. Firstly, the results are obtained by dividing and weighting the parameters, and then optimized by using the proportional relationship of similar triangles. In this way, the corresponding relationship between the arc length of NURBS curve and the parameters is established. The experimental results show that When step size $\Delta t$ is 0.01, 852 interpolation points can be calculated in 0.689s, and the accuracy remains at Within $3 \times 10^{-5} \text{mm}$; When $\Delta t$ is 0.001, 852 interpolation points can be calculated in 0.294s, and the accuracy can be maintained at Within $2 \times 10^{-8} \text{mm}$ and with relatively stable speed, it can meet the requirements of high-speed and high-precision interpolation for high-performance CNC.

ACKNOWLEDGMENT

The research work is supported by the Ministry of Industry and Information Technology under project on information protection for numerical control systems with No. TC2000H03A and Department of Guangdong Science and Technology under project 2020B1111410002. The authors sincerely present thanks to them all.

REFERENCES