The International Journal of Advanced Manufacturing Technology (2020) 111:2095–2104
https://doi.org/10.1007/s00170-020-06258-x

ORIGINAL ARTICLE

A NURBS curve interpolator with small feedrate fluctuation based on arc length prediction and correction

Shuai Ji[1] · Tianliang Hu[2] · Zuguang Huang[3] · Chengrui Zhang[2]

Received: 19 May 2020 / Accepted: 15 October 2020 / Published online: 22 October 2020

Abstract

Non-uniform rational B-spline (NURBS) curve has been widely used in manufacturing systems. A good interpolator can help the system to improve the contour accuracy and get smooth dynamics performance, but it is hard to get a balance between the interpolation performance and computational load. As the derivative and curvature of NURBS curves used in manufacturing systems are high-order continuous, it is possible to predict a desired interpolation arc length based on the relationship of historical feed chord length and its corresponding arc length in one interpolation cycle. Therefore, this paper proposes a novel interpolation method, which consists three stages. Firstly, a NURBS curve is split into several high-order continuous segments based on its degrees and control points. Secondly, a prediction model based on Newton's divided differences interpolation equation is derived from the relationship of interpolated chord length and its corresponding arc length, so that the target arc length of the next interpolation cycle can be predicted. Finally, target parameter u of every interpolation cycle is calculated with Taylor's expansion, whose values are corrected by iteration, and then the position of the target interpolation point can be achieved. Performance of the proposed algorithm is tested and compared with other methods, and the simulation results show the proposed method can achieve smaller velocity fluctuation with low computational load.

Keywords NURBS interpolation · Feedrate fluctuation · Arc length prediction · Taylor's expansion

1 Introduction

Non-uniform rational B-splines (NURBS) has been widely used in the designing of free-form surface, as it can offer an exact uniform representation of analytical shapes as well as free-form parametric curves and surfaces [1, 2]. However, as a kind of parametric curve, there is no analytic relationship between its parameter u and the corresponding arc length, so it is difficult to design a high accurate and real-time interpolator with small feedrate fluctuation.

Therefore, NURBS interpolator is always a research hotspot in the last decades, and many interpolation algorithms have been proposed in order to get better performance [3–5], which can be divided into three categories: approximation method, fitting method, and feedback method.

Approximation is first proposed by Koren et al., where time-based Taylor's expansion is used to approximate the NURBS curve [3]. Different algorithms are developed based on this idea after that [6–8]. Runge-Kutta [9] and Adams-Bashforth [10] methods are also used in order to improve the approximation accuracy. However, truncation errors are inevitable for these methods in practice, so feedrate fluctuations are also inevitable, although high-order terms can improve the accuracy, which means more computational load and worse real-time performance.

Fitting method can also be called remapping method, and Cheng et al. employs a high-order polynomial spline to fit the relationship of arc length and the curve parameter u [11]. Liu *et al.* proposed a quartic equation-based interpolation algorithm, which can reduce the feedrate fluctuation for NURBS curves to satisfy the command feedrate [12]. This method can get a direct relationship between the parameter and arc length,

✉ Tianliang Hu
tlhu@sdu.edu.cn

[1] School of Mechanical and Electronic Engineering, Shandong Jianzhu University, Jinan 250101, China

[2] School of Mechanical Engineering, Shandong University, Jinan 250061, China

[3] China National Machine Tool Quality Supervision Testing Center, Beijing 100102, China

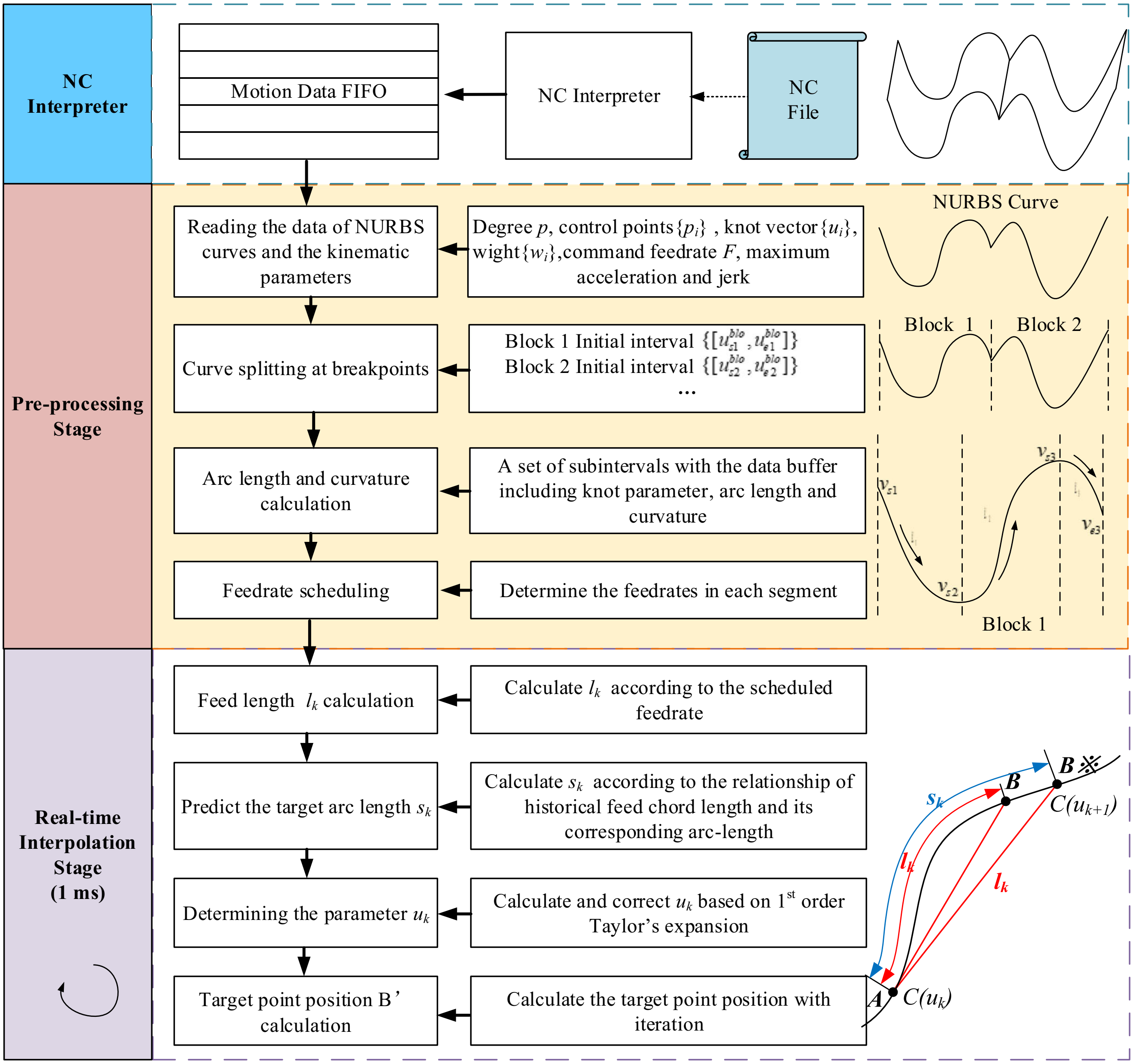

NC Interpreter
Motion Data FIFO
NC Interpreter
NC File
Pre-processing Stage
Reading the data of NURBS curves and the kinematic parameters
Degree p, control points $\{p_i\}$, knot vector $\{u_i\}$, wight $\{w_i\}$,command feedrate F, maximum acceleration and jerk
NURBS Curve
Curve splitting at breakpoints
Block 1 Initial interval $\{[u_{s1}^{blo}, u_{e1}^{blo}]\}$
Block 2 Initial interval $\{[u_{s2}^{blo}, u_{e2}^{blo}]\}$
…
Block 1
Block 2
Arc length and curvature calculation
A set of subintervals with the data buffer including knot parameter, arc length and curvature
v_{s1}
v_{s2}
v_{s3}
v_{e3}
Feedrate scheduling
Determine the feedrates in each segment
Block 1
Real-time Interpolation Stage (1 ms)
Feed length l_k calculation
Calculate l_k according to the scheduled feedrate
Predict the target arc length s_k
Calculate s_k according to the relationship of historical feed chord length and its corresponding arc-length
Determining the parameter u_k
Calculate and correct u_k based on 1st order Taylor's expansion
Target point position B' calculation
Calculate the target point position with iteration
s_k
l_k
l_k
B※
B
$C(u_{k+1})$
A
$C(u_k)$

Fig. 1 interpolation process of NURBS curve

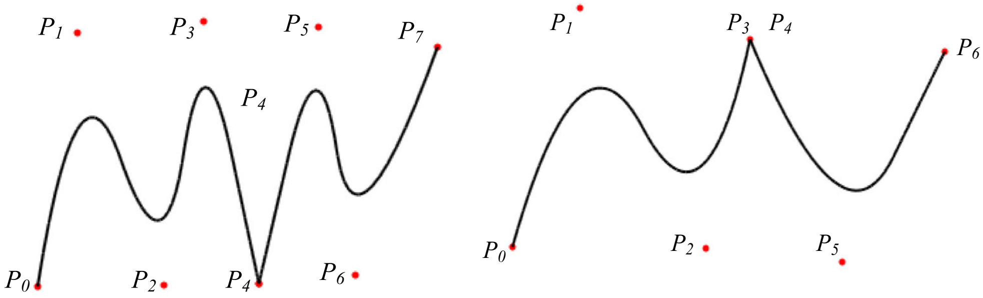
but the fitting operation is time-consuming, and large amount memory is required, so it is not suitable for real-time interpolation.

Feedback parametric interpolation is an improvement of the approximation method proposed by Lo [13], which employs an iterative way to approximate a desired point. Cheng *et al.* proposed a real-time predictor–corrector interpolator, the feedrate command error can fall within the specified tolerance with a predicted initial value and iteration [14]. Zhao *et al.* proposed a feedback interpolator with arc-length compensation and the feedback correction, good efficiency and accuracy can be achieved with their method [15]. These feedback methods can get desired feedrate fluctuation with enough times of iteration, but the iteration time is uncertain, so it is also not suitable for real-time interpolator [16].

In Summary, there is always a tradeoff between the computational load and the desired performance. An interpolator with high accuracy, high efficiency, and small feedrate fluctuation is still a big challenge for motion control systems in practice [17].

Therefore, an interpolation method based on Taylor's expansion with arc length prediction and feedback correction is proposed in this paper. Firstly, a NURBS curve is split into several continuous segments, and the velocity profile is generated based on the arc length of every segment. Secondly, a prediction model is

Fig. 2 Division of second-degree NURBS, (**a**) NURBS curve with repeated knot vector elements, (**b**) NURBS with multiple control points



(a)NURBS curve with repeated knot vector elements

(b)NURBS with multiple control points

derived from the relationship of the previous arc length and the corresponding feed length in one interpolation interval. As derivative and curvature of NURBS curves used in CAD/CAM system are continuous, the arc length can be predicted with this model and the command feed length. Finally, the value of parameter u is calculated with Taylor's expansion method and corrected by iteration, so that the target position with small feedrate fluctuation can be achieved from the parametric equations.

The rest of this paper is organized as follows. Section 2 is the introduction of Taylor's expansion-based NURBS interpolation and the analysis of feedrate fluctuation. Section 3 is the details of the proposed online interpolation method. The simulation results are given in Section 4. Conclusions are summarized in Section 5.

2 Interpolation of NURBS curve

As a non-uniform rational B-spline curve, NURBS curve $C(u)$ can be expressed as in Eq. (1)

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

where the index $i=0, 1, \ldots, n$, $\boldsymbol{P_i}$ is the set of control point, w_i is the set of corresponding weight of $\boldsymbol{P_i}$, $(n+1)$ is the number of the control points, and p is the degree of the NURBS curve. $N_{i,p}(u)$ is the B-spline basis function defined on the non-uniform knot vector $\boldsymbol{U}=\{u_0,u_1,\cdots,u_{n+p+1}\}$. The B-spline basic function $N_{i,p}(u)$ is defined as

$$N_{i,0}(u)=\begin{cases}0 \ \ if\ u_i<u\le u_{i+1}\\ 1 \ \ otherwise\end{cases}$$
$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u)$$

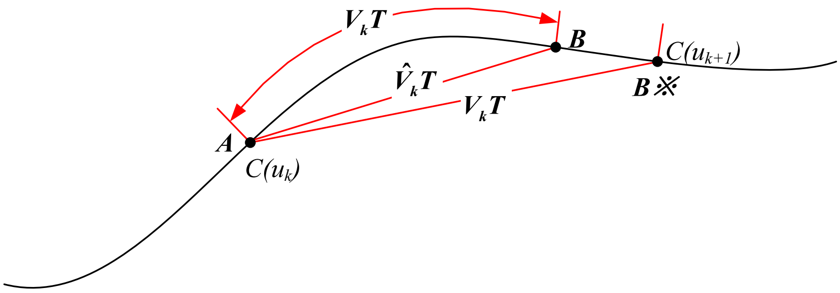
The interpolation of NURBS curve can be divided into two stages: pre-processing and online interpolation. In pre- processing, a NURBS curve is split into several continuous segments based on its degree and control points. In online interpolation stage, the parameter u is derived with Taylor's expansion or other interpolation algorithms, and then the position of the target interpolation points can be obtained. The process is in Fig. 1.

2.1 Pre-processing of NURBS curve interpolation

NURBS curve is the linear combination of a series of B-spline basis functions, so it is (p-k) times continuously differentiable at a knot of multiplicity k for a p degree curve [14]. The continuity degree of the curve at each knot can be achieved by its degrees and knot vector. So that an NURBS curve can be divided into several separate segments of infinitely differentiable spline base on the orders of its continuity and number of control points.

Two examples with repeated knot vector values and multiple control points are shown in Fig. 2 where $\boldsymbol{P}=\{P_1,P_2,\ldots P_7\}$

Fig. 3 Desired and actual feed in an interval



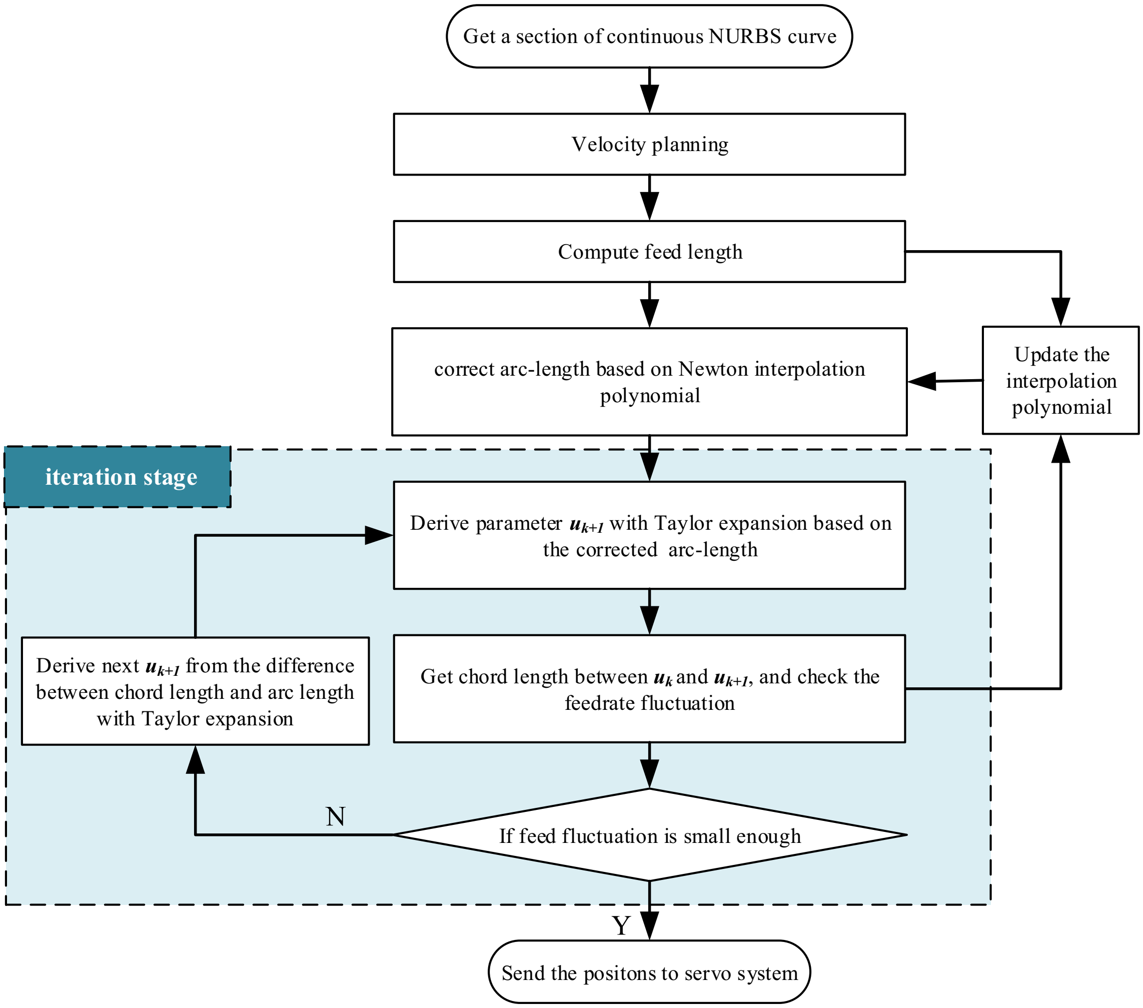

Get a section of continuous NURBS curve
Velocity planning
Compute feed length
correct arc-length based on Newton interpolation polynomial
Update the interpolation polynomial
iteration stage
Derive parameter u_{k+1} with Taylor expansion based on the corrected arc-length
Derive next u_{k+1} from the difference between chord length and arc length with Taylor expansion
Get chord length between u_k and u_{k+1}, and check the feedrate fluctuation
N
If feed fluctuation is small enough
Y
Send the positons to servo system

Fig. 4 Flowchart of the arc length predictor and correction algorithm

are control points of the NURBS curve. Both of these two curves are second degree. In Fig. 2 (a), knot vector is $\boldsymbol{U}$=[*0, 0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1, 1*], and in Fig. 2 (b), control points $P_3=P_4$. These two curves both can be split into two smooth segments with the abovementioned method.

It is necessary to get the accurate arc length of the split continuous segments first before velocity scheduling and interpolation. The arc length of the NURBS curve between [a, b] can be expressed as in Eq. (2)

$$s = \int_a^b \|C'(u)\| du \tag{2}$$

where $\| * \|$ denotes the vector norm and $\|C'(u)\|$ is the norm of the first-order derivative of $C'(u)$.

Fig. 5 relationship of chord length, curvature and arc length

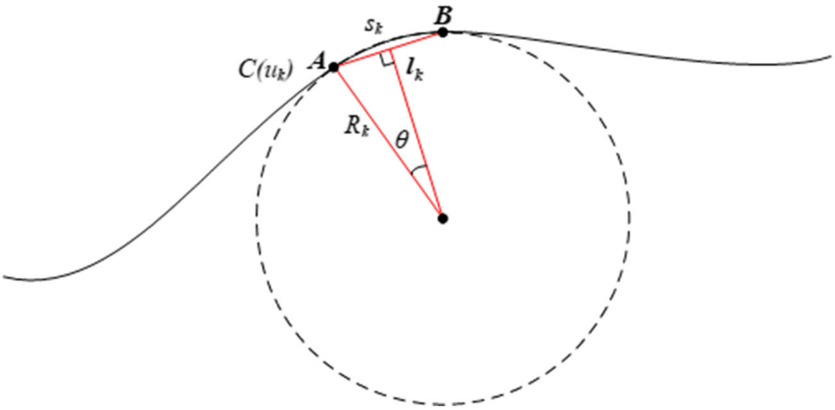


Table 1 Newton backward difference formula

Step	Value	1st-order divided quotient	2nd-order divided quotient	3rd-order divided quotient
k-3	λ_{k-3}			
k-2	λ_{k-2}	$\nabla\lambda_{k-2}=\lambda_{k-2}-\lambda_{k-3}$		
k-1	λ_{k-1}	$\nabla\lambda_{k-1}=\lambda_{k-1}-\lambda_{k-2}$	$\nabla^2\lambda_{k-1}=\nabla\lambda_{k-1}-\nabla\lambda_{k-2}$ $=(\lambda_{k-1}-\lambda_{k-2})-(\lambda_{k-2}-\lambda_{k-3})$ $=\lambda_{k-1}-2\lambda_{k-2}+\lambda_{k-3}$	
k	λ_k	$\nabla\lambda_k=\lambda_k-\lambda_{k-1}$	$\nabla^2\lambda_k=\nabla\lambda_k-\nabla\lambda_{k-1}$ $=(\lambda_k-\lambda_{k-1})-(\lambda_{k-1}-\lambda_{k-2})$ $=\lambda_k-2\lambda_{k-1}+\lambda_{k-2}$	$\nabla^3\lambda_k=\nabla^2\lambda_k-\nabla^2\lambda_{k-1}$ $=(\lambda_k-2\lambda_{k-1}+\lambda_{k-2})-(\lambda_{k-1}-2\lambda_{k-2}+\lambda_{k-3})$ $=\lambda_k-3\lambda_{k-1}+3\lambda_{k-2}-\lambda_{k-3}$

However, there is no analytical relation between the arc length and its parameter for NURBS curve, as in the case of lines and circles, so a numerical integration method Simpsons rule is adopted considering the computational accuracy and load. The approximate arc length *s(a, b)* of a segment of NURBS curve on the interval [a, b] can be expressed as

$$s(a,b)=\left[C(a)+4C\left(\frac{a+b}{2}\right)+C(b)\right]\frac{(b-a)}{6}$$

For a specified tolerance ε, if the condition in Eq. (3) is satisfied, then the approximation is within the given tolerance of the true arc length.

$$\frac{|s(a_1,b_1)+s(a_2,b_2)-s(a,b)|}{10}<\varepsilon \qquad (3)$$

where [a_1 ,b_1] and [a_2 ,b_2] are the intervals split from [*a*, *b*]. If the condition in Eq. (2) cannot be satisfied, the subintervals should be further refined.

2.2 Interpolation based on Taylor's expansion

Arc length of NURBS is in a nonlinear relationship with its parameter *u*, so its interpolation algorithm is much more complex than linear and circle interpolation. The interpolation of NURBS is a process of computing a proper parameter value *u* to satisfy that the feed length in an interpolation interval is equal to the command velocity.

The feedrate *V(t)* of can be defined as

$$V(t)=\left\|\frac{dC(u)}{dt}\right\|$$

where *u(t)* is a monotone increasing function of time in the NURBS interpolation, then the equation as in Eq. (4) can be derived

$$V(t)=\left\|\frac{dC(u)}{dt}\right\|=\left\|\frac{dC(u)}{du},\cdot,\frac{du}{dt}\right\|=\left\|\frac{dC(u)}{du}\right\|\cdot\frac{du}{dt} \qquad (4)$$

therefore, the first-order derivatives of $u(t)$ can be achieved as in Eq. (5).

$$\frac{du}{dt}=\frac{V(t)}{\left\|\frac{dC(u)}{du}\right\|} \qquad (5)$$

the second-order derivatives of $u(t)$ can be achieved as in Eq. (6).

$$\frac{d^2u}{dt^2}=\frac{V_k^2\left\langle\frac{dC(u)}{du},\frac{d^2C(u)}{du^2}\right\rangle}{\left\|\frac{dC(u)}{du}\right\|^4} \qquad (6)$$

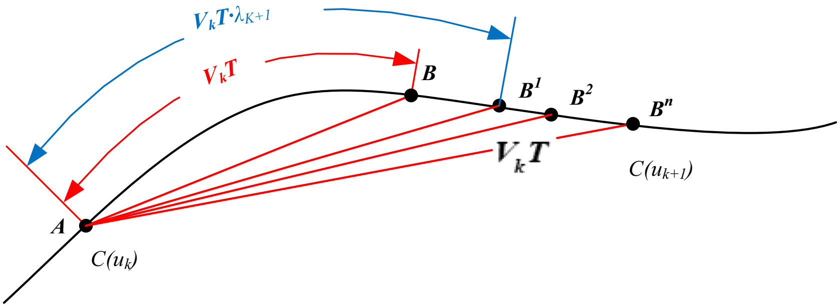
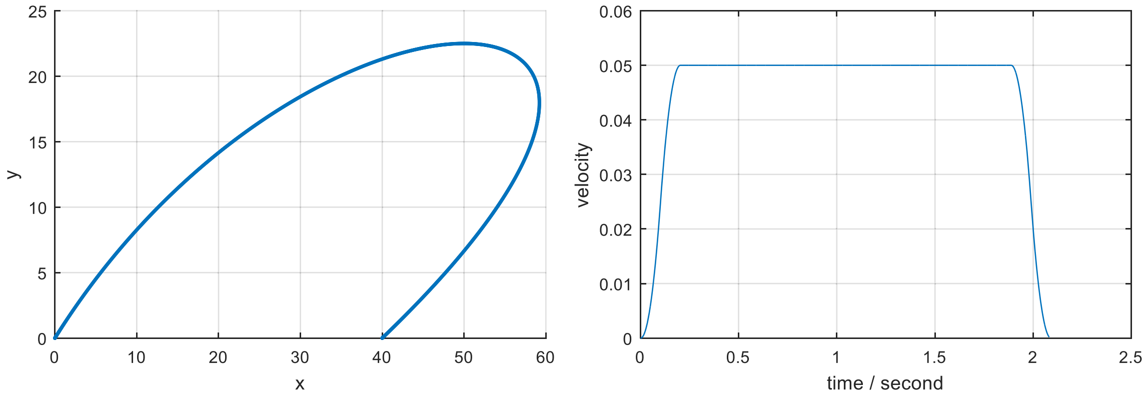


Fig. 6 Process of getting target feedrate fluctuation



(a) shape of the curve

(2) velocity profile

Fig. 7 The tested cubic polynomial curve, **(a)** Shape of the curve, **(b)** Velocity profile

The algorithm based on the first- or second-order Taylor's expansion method is one of the most used methods to obtain the value of parameter u. Therefore u_{k+1} can be calculated with Taylor's expansion.

$$u_{k+1} = u_k + T\left(\frac{du}{dt}\right)\Big|_{u=u_k} + \frac{T^2}{2}\left(\frac{d^2u}{dt^2}\right)\Big|_{u=u_k} + o\left(T^3\right) \tag{7}$$

where T is the interpolation interval. Substituting Eq. (5) into (7), then u_{k+1} can be derived. If the interpolation interval is small enough, the higher order infinitesimal in Taylor's expansion can be neglected, and the first-order Taylor's expansion can be derived as in Eq. (6)

$$u_{k+1} \approx u_k + \frac{V_kT}{\left\|\frac{dC(u_k)}{du}\right\|} \tag{8}$$

More accurate result can be achieved with second- or even higher-order Taylor's expansion, and the-second order Taylor's expansion is shown in Eq. (9).

$$u_{k+1} \approx u_k + \frac{V_kT}{\left\|\frac{dC(u_k)}{du}\right\|} + \frac{(V_kT)^2\left\langle \frac{dC(u)}{du}, \frac{d^2C(u)}{du^2}\right\rangle\Big|_{u=u_k}}{\left\|\frac{dC(u)}{du}\right\|^4_{u=u_k}} \tag{9}$$

Therefore, the next interpolation point can be achieved by $C(u_{k+1})$, where the feed arc length of current interpolation interval is expressed as V_kT.

2.3 Feedrate fluctuation analysis

The interpolation point $C(u_{k+1})$ on the curve can be achieved with u_{k+1}, so that the resultant displacement of the motion can be achieved

$$l_k = \widehat{V}_kT = \|C(u_{k+1}) - C(u_k)\| \tag{10}$$

There are two problems need to be noticed. Firstly, V_kT is the arc length of current curve as shown in Eq. (11), which is different from the chord length $\widehat{V}_kT$ in Eq. (10).

$$s_k = V_kT = \int_{u_k}^{u_{k+1}}\left\|\frac{dC(u)}{du}\right\|du \tag{11}$$

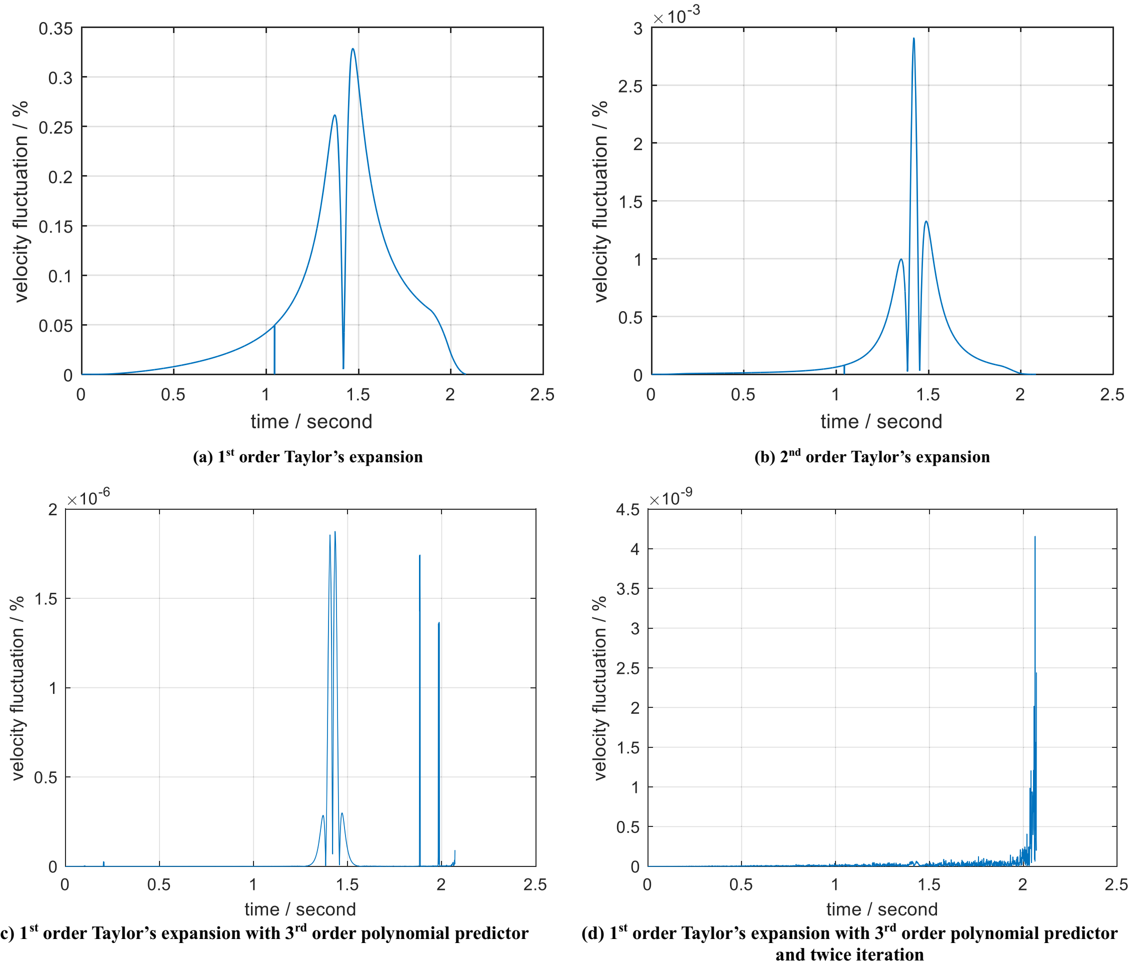
Secondly, the high-order term of Taylor's expansion is neglected. Although the higher-order Taylor's expansion can achieve better precision, truncation errors are still inevitable. These two factors lead to the difference of desired and actual feed length in an interpolation interval as in Fig. 3. In another words, the feedrate fluctuation ε from the difference of the actual velocity $\widehat{V}_k$ and the desired velocity V_k is unavoidable as in Eq. (12).

$$\varepsilon = \left|\frac{V_k - \widehat{V}_k}{V_k}\right| \times 100\% \tag{12}$$

In order to achieve smaller feedrate fluctuation, another interpolation target point B' as in Fig. 3 needs to be found, so that the actual feed length $|AB'|$ could be equal to the arc length AB, and feedrate fluctuation can be eliminated. So how to get the target point B' to insure that the chord length $|AB'|$ is equal to its arc length AB is the target of this paper.

3 Interpolation with online arc length prediction and correction

The proposed online interpolation method can be divided into two stages. Firstly, a prediction model is derived based on the relationship of the arc length and chord length of previous interpolation results, and then target arc length AB' is



(a) 1st order Taylor's expansion

(b) 2nd order Taylor's expansion

(c) 1st order Taylor's expansion with 3rd order polynomial predictor

(d) 1st order Taylor's expansion with 3rd order polynomial predictor and twice iteration

Fig. 8 Feedrate fluctuation comparison of different methods, **(a)** 1st order Taylor's expansion, **(b)** 2nd order Taylor's expansion (**c**)1st order Taylor's expansion with 3rd order polynomial predictor, (**d**) 1st order Taylor's expansion with 3rd order polynomial predictor and twice iteration

predicted based on the desired feed length and the derived model. Secondly, target parameter u of the curve is derived with Taylor's expansion and corrected by iteration, so the position of the target interpolation points can be obtained. The whole process is as shown in Fig. 4.

3.1 Arc length predictor

Mark the arc length between u_k and u_{k+1} as s_k, and its corresponding chord length as l_k. When the arc length is small enough, it can be approximated with a circle of same curvature as shown in Fig. 5

Therefore, the arc length s_k can be expressed as a function of the chord length l_k and the radius of curvature $1/R_k$, as shown in Eq. (13).

$$s_k = f(R_k, l_k) \approx 2R_k \cdot sin\frac{l_k}{2R_k} \tag{13}$$

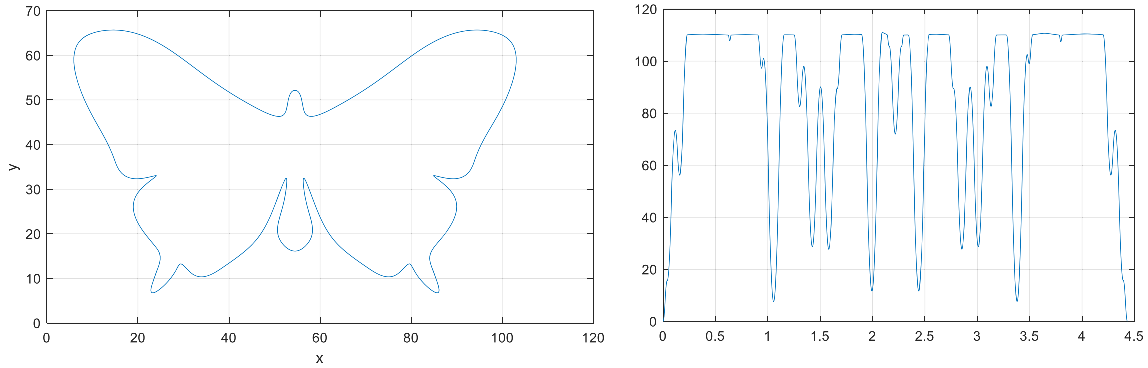
The chord length l_k is the actual feed length in an interpolation interval, which is computed in the feedrate scheduling stages. It can be seen that the arc length s_k is a function of the its curvature $1/R_k$ and the chord length l_k. In another word, if the curvature of the spline is continuous, the arc length s_k corresponding to the given chord length l_k will change continuously with the curvature of the spline.

$$\lim_{\theta\to 0} s_k = \lim_{\theta\to 0} 2R_k \cdot sin\frac{l_k}{2R_k} = 2R_k \cdot \frac{l_k}{2R_k} = l_k \tag{14}$$

When the chord length tends to zero, it will be equal to the arc length. Take the quotient of arc length s_k and chord length l_k as λ, then λ can be expressed as in Eq. (15).

$$\lambda = \frac{s}{l} = \frac{\int_{u_k}^{u} \left\| \frac{dC(u)}{du} \right\| du}{\|C, (u), -, C, (u_k)\|} \tag{15}$$

A polynomial is employed to approximate this relationship in an interpolation interval. It is known that cubic spline is



(a) shape of the curve

(b) velocity profile

Fig. 9 The tested cubic polynomial curve, (**a**) shape of the curve, (**b**) velocity profile

most used in CAD/CAM, whose derivative and curvature are continuous, so λ is continuous. Arc length λ_{k+1} can be interpolated based on the previous value λ_k, λ_{k-1} and λ_{k-2}. Here Newton's divided difference interpolation polynomial is employed to get the correction factor.

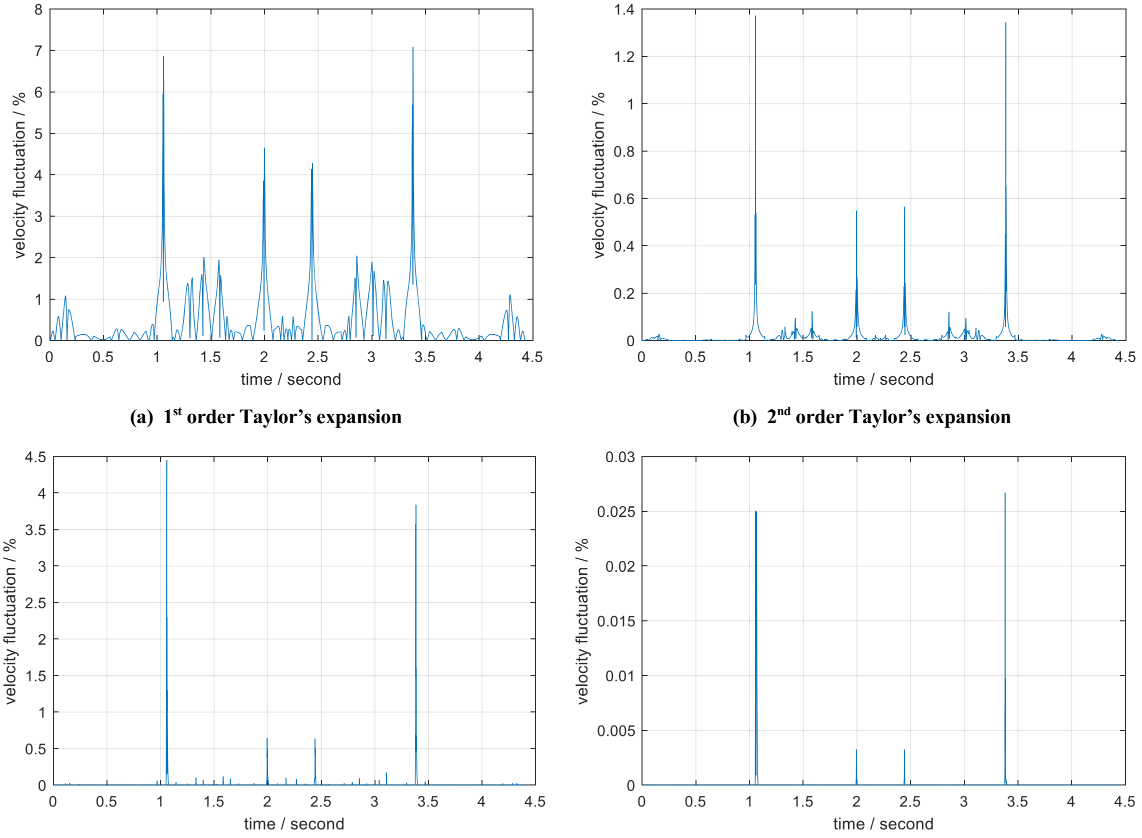
$$\lambda_k = \lambda_0 + \sum_{k=1}^{n} \frac{\Delta^k \lambda_k}{k!} s(s-1)\cdots(s-k+1), \qquad 0 \le s \le n$$

The divided differences table is shown in Table 1.

Then the target arc length s_{k+1} corresponding to the desired feed length $V_k T$ can be derived as in Eq. (16).

$$s_{k+1} = l_{k+1} \cdot \lambda_{k+1} = V_k T \cdot \lambda_{k+1} \tag{16}$$

Different orders of the divided quotient can be employed to get desired precision. When abovementioned algorithm is implemented, arc length and chord length are unknown at first,



(a) 1st order Taylor's expansion

(b) 2nd order Taylor's expansion

(c) 1st order Taylor's expansion with 3rd polynomial predictor

(d) 1st order Taylor's expansion with 3rd polynomial predictor and twice iteration

Fig. 10 Feedrate fluctuation comparison of different methods, (**a**) 1st order Taylor's expansion, (**b**) 2nd order Taylor's expansion, (**c**) 1st order Taylor's expansion with 3rd polynomial predictor, (**d**) 1st order Taylor's expansion with 3rd polynomial predictor and twice iteration

Table 2 Sum of feedrate fluctuation in the whole interpolation process

	1st order Taylor's expansion	2nd order Taylor's expansion	1st order Taylor's expansion 3rd order polynomial predictor	1st order Taylor's expansion 3rd order polynomial predictor twice iteration
$\sum \varepsilon_i$	2249.66	76.78498	49.52917	0.376616

so a buffer is needed. Only when the buffer is full with historic data, the Newton's divided difference interpolation polynomial can start working.

3.2 Feed-length correction

$$u^1_{k+1} \approx u_k + \frac{V_k T \lambda_{k+1}}{\sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2 + \left(\frac{dz}{du}\right)^2}\Bigg|_{u=u_k}} \tag{17}$$

Taylor's expansion is employed to get the value of parameter u as shown in Eq. (8) and Eq. (9). Given the computational load of second-order Taylor's expansion, firs order is employed here.

As the feed length in an interpolation interval is the predicted with Eq. (16), then another target position B^1 as in Fig. 7 is obtained, then u_{k+1} can be derived as shown in Eq. (17).

The actual feed length in this interpolation interval can also be achieved as in Eq. (18).

$$l^1_{k+1} = \sqrt{(x_{k+1}-x_k)^2 + (y_{k+1}-y_k)^2 + (z_{k+1}-z_k)^2}\Bigg|_{u=u^1_{k+1}} \tag{18}$$

Take Δl_{k+1} as the difference between the feed displacement and chord length, when the value is 0 means the feedrate fluctuation is 0. So that an iterative formula can be derived as in Eq. (19).

$$\begin{cases} u^{i+1}_{k+1} = u^i_{k+1} + \dfrac{\Delta l^i_{k+1}}{\sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2 + \left(\frac{dz}{du}\right)^2}}\Bigg|_{u=u^i_{k+1}} \\ \Delta l^i_{k+1} = V_k T - l^i_{k+1} \end{cases} \tag{19}$$

An ideal value of feedrate fluctuation can be obtained with enough times of iteration; the process is shown in Fig. 6.

4 Simulation and verification

A segment of 2D cubic curve as in Eq. (20) is interpolated with the velocity V=1.5m/min, interpolation interval T=0.001s.

$$\begin{cases} x = -140u^3 + 90u^2 + 90u \\ y = -90u^2 + 90u \end{cases} \tag{20}$$

The contour shape and velocity profile of this curve are shown in Fig. 7. The feedrate fluctuation performance of traditional Taylor's expansion method and the proposed prediction model based algorithm is shown in Fig. 8.

The maximum feedrate fluctuation of first-order Taylor's expansion interpolation method is about 0.35%, as shown in Fig. 8(a). When second-order Taylor's expansion is employed, better performance can be achieved, and the feedrate fluctuation can be reduced to 3×10^{-3} %, as shown in Fig. 8(b). The feedrate fluctuation of arc length predictor model is only one thousandth of second-order Taylor's expansion as in Fig. 8(c), and better performance can be achieved when iteration is applied as in Fig. 8(d).

For the curve in Eq. (20), even second-order Taylor's expansion based interpolator can get the desired feedrate fluctuation. The interpolation of a more complicated butterfly shape NURBS curve in Fig. 9(a) is used to verify the performance of proposed method, where Fig. 9(b) is the velocity profile. The feedrate fluctuation of these methods is shown in Fig. 10, and the sum of feedrate fluctuation in the whole interpolation process is shown in Table 2.

The interpolation results show that, although the biggest feedrate fluctuation of second-order Taylor's expansion method in Fig. 10(b) is smaller than arc length predictor model in Fig. 10(c), its sum feedrate fluctuation is bigger as in Table 2. Besides, the computational load of second-order Taylor's expansion method is also larger than third-order polynomial. When the predictor and iteration method are combined together, the feedrate fluctuation can reduce hundreds times, and high performance can be achieved as in Fig. 10(d) and Table 2.

5 Conclusion

Feedrate fluctuation in NURBS interpolation is analyzed, and a novel interpolator is proposed to get small feedrate fluctuation. A polynomial based prediction model is derived based on the continuous relationship of the interpolated chord length and its corresponding arc length, so that an accurate interpolation arc-length with small feedrate fluctuation can be achieved with this model. Then the target parameter u in every interpolation cycle is calculated with Taylor's expansion and corrected by iteration, so the desired feedrate fluctuation can be achieved. Performance of the proposed algorithm is tested with two segments of curve, and the simulation results show that the proposed algorithm can get smaller feedrate fluctuation with fewer times of iteration than traditional Taylor's expansion method.

Funding This research is supported by National Science and Technology Major Project of China (Grant No. 2019ZX04004001), Natural Science Foundation of Shandong Province (Grant No. ZR2019QEE042), The Project of Innovative Application Experiencing Center of Industrial Internet Platform.

Appendix A Parameters of the Butterfly shape NURBS curve

Control point (mm):

(54.493, 52.139), (55.507, 52.139), (56.082,49.615), (56.780, 44.971), (69.575, 51.358), (77.786, 58.573),(90.526, 67.081), (105.973, 63.801), (100.400, 47.326), (94.567,39.913), (92.369, 30.485), (83.440, 33.757), (91.892, 28.509),(89.444, 20.393), (83.218, 15.446), (87.621, 4.830), (80.945,9.267), (79.834, 14.535), (76.074, 8.522), (70.183, 12.550), (64.171,16.865), (59.993, 22.122), (55.680, 36.359), (56.925, 24.995),(59.765, 19.828), (54.493, 14.940), (49.220, 19.828), (52.060,24.994), (53.305, 36.359), (48.992, 22.122), (44.814, 16.865),(38.802, 12.551), (32.911, 8.521), (29.152, 14.535), (28.040, 9.267),(21.364, 4.830), (25.768, 15.447), (19.539, 20.391), (17.097, 28.512),(25.537, 33.750), (16.602, 30.496), (14.199, 39.803), (8.668, 47.408), (3.000, 63.794), (18.465, 67.084), (31.197, 58.572), (39.411, 51.358), (52.204, 44.971), (52.904, 49.614), (53.478, 52.139), (54.492,52.139).

Knot vector:

[0, 0, 0, 0, 0.0083, 0.0150, 0.0361, 0.0855, 0.1293, 0.1509, 0.1931, 0.2273, 0.2435, 0.2561, 0.2692, 0.2889, 0.3170, 0.3316, 0.3482, 0.3553, 0.3649, 0.3837, 0.4005, 0.4269, 0.4510, 0.4660, 0.4891, 0.5000, 0.5109, 0.5340, 0.5489, 0.5731, 0.5994, 0.6163, 0.6351, 0.6447, 0.6518, 0.6683, 0.6830, 0.7111, 0.7307, 0.7439, 0.7565, 0.7729, 0.8069, 0.8491, 0.8707, 0.9145, 0.9639, 0.9850, 0.9917, 1, 1, 1, 1]

Weight vector:

[1.0000, 1.0000, 1.0000, 1.2000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 2.0000, 1.0000, 1.0000, 5.0000, 3.0000, 1.0000, 1.1000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.1000, 1.0000, 3.0000, 5.0000, 1.0000, 1.0000, 2.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.2000, 1.0000, 1.0000, 1.0000]

References

1. Shamsuddin SM, Ahmed M, Smian Y (2006) NURBS skinning surface for ship hull design based on new parameterization method. Int J Adv Manuf Technol 28(9-10):936–941
2. Lei W, Sung M, Lin L, Huang J (2007) Fast real-time NURBS path interpolation for CNC machine tools. Int J Mach Tools Manuf 47(10):1530–1541
3. Koren Y, Lo C, Shpitalni M (1993) CNC interpolators: algorithms and analysis, vol 64. Asme Prod Eng Div Publ Ped, Asme, New York, pp 83–92
4. Chen M, Sun Y (2019) Contour error–bounded parametric interpolator with minimum feedrate fluctuation for five-axis CNC machine tools. Int J Adv Manuf Technol 103(1-4):567–584
5. Zhao K, Li S, Kang Z (2019) Smooth minimum time trajectory planning with minimal feed fluctuation. Int J Adv Manuf Technol 105(9-12):1099–1111
6. Cheng CW, Tsai MC (2004) Real-time variable feed rate NURBS curve interpolator for CNC machining. Int J Adv Manuf Technol 23(11-12):865–873
7. Lin MT, Tsai MS, Yau HT (2007) Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm. Int J Mach Tool Manu 47(15):2246–2262
8. Baek D-K, Ko T-J, Yang S-H (2012) Fast and precision NURBS interpolator for CNC systems. Int J Precis Eng Manuf 13(6):955–961
9. Jia ZY, Song DN, Ma JW, Hu GQ, Su WW (2017) A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints. Int J Mach Tool Manu 116:1–17
10. Zhang X-t (2012) An iterative feedrate optimization method for real-time NURBS interpolator. Int J Adv Manuf Technol 62(9-12):1273–1280
11. Heng M, Erkorkmaz K (2010) Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability. Int J Mach Tools Manuf 50(3):281–293
12. Liu H, Liu Q, Zhou S, Li C, Yuan S (2015) A NURBS interpolation method with minimal feedrate fluctuation for CNC machine tools. Int J Adv Manuf Technol 78(5-8):1241–1250
13. Lo CC (1997) Feedback Interpolators for CNC Machine Tools. J Manuf Sci Eng 119(4A):587–592
14. Tsai M-C, Cheng C-W (2003) A Real-Time Predictor-Corrector Interpolator for CNC Machining. J Manuf Sci Eng 125(3):449–460
15. Zhao H, Zhu LM, Han D (2013) A parametric interpolator with minimal feed fluctuation for CNC machine tools using arc-length compensation and feedback correction. Int J Mach Tool Manu 75: 12):1–12):8
16. Wang T-Y, Zhang Y-B, Dong J-C, Ke R-J, Ding Y-Y (2020) NURBS Interpolator with Adaptive Smooth Feedrate Scheduling and Minimal Feedrate Fluctuation. Int J Precis Eng Manuf 21(2): 273–290
17. Liu H, Liu Q, Sun P, Liu Q, Yuan S (2017) A polynomial equation-based interpolation method of NURBS tool path with minimal feed fluctuation for high-quality machining. Int J Adv Manuf Technol 90(9):2751–2759