2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

A New Feedrate Adaptation Control NURBS Interpolation Based on de Boor Algorithm in CNC Systems

Tianmiao Wang, Yunan Cao, Youdong Chen, Hongxing Wei, Bin Wang, Zili Shao

Abstract—**The interpolation method and feedrate adaptation control are the two most important factors to influence the quality and accuracy of manufacturing for NURBS on CNC systems. In this paper, we propose an integrated approach to solve the feedrate adaptation control NURBS interpolation problem by considering interpolation and feedrate control together with the "tail" handling. We propose a de-Boor-algorithm-based NURBS interpolation model with the second-order Taylor expansion and design a new algorithm to calculate actual deceleration points to remove the "tail" based on a novel concept, reverse NURBS interpolation. To supplement the tail removal in velocity control, a novel feedrate adaptation control algorithm is proposed. The experimental results show that our integrated method can significantly improve the accuracy.**

I. Introduction

There are many different representations for parametric curves, among them, NURBS has become the standard curve and surface description in the field of CAD (Computer-Aided Design). The interpolation method and feedrate adaptation control are the two most important factors to influence the quality and accuracy of manufacturing for NURBS on CNC systems. Therefore, in this paper, we focus on solving the feedrate adaptation control NURBS interpolation problem.

A lot of techniques have been proposed to solve the feedrate adaptation control NURBS interpolation problem. In [1], an approach is proposed to solve the chord error of NURBS curve that is approximate to the distance between the mid-point of curve and the mid-point of chord length. This approach may bring some errors in curve inflection points. In [2], the arc is used to displace the original NURBS curve to compute the chord error that can achieve the required accuracy. In [3], a NURBS interpolation technique is proposed which can keep constant material removal rate. In [4], the acceleration is concerned in the feedrate control algorithm. The above techniques only focus on solving one part of the problem, either on interpolation or feedrate control. Here we can achieve better accuracy by considering both together.

Another important problem is neglected from the previous work is to handle "tail". The "tail" is rising from the discreterization in the pre-interpolation acc/dec (acceleration/deceleration). Since the actual deceleration point may be different from the theoretical deceleration point, the velocity and the distance may not arrive at the zero point simultaneously, which lowers the accuracy and increases the interpolation processing time.

In this paper, we propose an integrated approach to solve the above problems by considering interpolation and feedrate control together with the tail handling. The main contributions of our work are summarized as follows: 1) The NURBS curve interpolation method based on de Boor algorithm is analyzed, and the second-order Taylor expansion for NURBS is deduced for convolution. 2) Based on this NURBS interpolation model, we propose a new pre-interpolation acc/dec method to solve the tail problem. 3) We propose a novel feedrate adaptation control algorithm which is the supplement of the tail removal in velocity control. 4) We conduct a series of experiments with our method.

This paper is organized as follows: The NURBS curve interpolation based on de Boor algorithm is presented in Section II. In Section III, we propose our NURBS curve feedrate adaptation control interpolation. The experiments and discussions are presented in Section IV. Finally, we conclude the paper in Section V.

II. NURBS curve interpolation based on de Boor algorithm

In this section, we present the NURBS interpolation algorithm.

A. NURBS

NURBS is the abbreviation of Non-Uniform Rational B-spline. For the NURBS curve interpolation algorithm, one of the core problems is the computation of the B-spline basic function. The basic method is to calculate it directly [5]. This method is easy, but the computation load is heavy. Another method is to use the de Boor interpolation algorithm [6]. As the de-Boor-based interpolation algorithm can avoid the convolution computation of B-spline basic function, its can greatly simplify the interpolation calculation so as to improve the real-time property and interpolation efficiency.

Another core problem in NURBS interpolation

The work described in this paper is partially supported by the grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (PolyU A-PH13, PolyU A-PA5X, PolyU A-PH41, and PolyU B-Q06B), the National Nature Science Foundation of China (60525314), the 973 Program of China (2002CB312204-04) and the 863 Program of China (2006AA04Z206).

Tianmiao Wang, Yunan Cao, Youdong Chen, Hongxing Wei and Bin Wang are with the Robotic Institute, Beijing University of Aeronautics&Astronautics, China (e-mail: chaosishere@126.com), and Zili Shao (the corresponding author, phone: (852)2766-7287; e-mail: cszlshao@comp.polyu.edu.hk) is with the Department of Computing, the Hong Kong Polytechnic University, Hong Kong.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

computation is the convolution procedure of computing knot vector, which can be solved in many different methods such as the adams-moulton expansion and the Hermite. The Taylor expansion has many advantages over above expansion methods since it has a more clear expression and the order of expansion can be changed according to the accuracy demand. When using the taylor expansion, the second-order algorithm is more accurate than first-order one [7]. However, as more computation is needed, the second-order term is neglected from the previous work. With the performance improvement of processors, however, to calculate the second-order term is not a problem any more in modern CNC systems. As it is needed in our tail removal and feedrate control, we will use the Taylor expansion with the second-order term in this paper.

B. NURBS and B-Spline Function

NURBS is a one more order expression form of B-spline curve with weight factor as shown below:

$$p(u)=\sum_{i=0}^{n}\omega_i d_i N_{i,k}(u)\Big/\sum_{i=0}^{n}\omega_i N_{i,k}(u) \tag{1}$$

in which, ω_i *(i=0,1,...,n)* is the weight factors corresponding to the control points $d_i(i=0,1,...,n)$. $N_{i,k}(u)$ is k degree B-spline basic function, and it is determined by the de Boor-Cox recursive formula which is defined on the knot vector $U=[u_0,u_1,...,u_{n+k+1}]$,

$$N_{i,0}(u)=\begin{cases}1, u_i\le u\le u_{i+1}\\ 0, others\end{cases} \tag{2}$$

$$N_{i,k}(u)=\frac{u-u_i}{u_{i+k}-u_i}N_{i,k-1}(u)+\frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}}N_{i,k-1}(u) \tag{3}$$

Here, 0/0=0.

From Equations 1-3, the coordination value corresponding to u along NURBS curve can be computed.

C. The NURBS based on de Boor Algorithm

From Equation 1, the denominator and numerator of NURBS curve are both the B-spline curve, so they can be computed separately with de Boor algorithm by the ratio of the two values. The de Boor algorithm of the NURBS is,

$$\begin{aligned}q(u)&=\sum_{j=0}^{i}d_jN_{j,k}(u)=\sum_{j=i-k}^{i}d_jN_{j,k}(u)\\&=\sum_{j=i-k}^{i-l}d_j^{(l)}N_{j,k-l}(u)=\cdots=d_{i-k}^{(k)}\end{aligned} \tag{4}$$

in which, $u\in[u_i,u_{i+1}]\subset[u_k,u_{n+1}]$,

$$d_j^{(l)}=\begin{cases}d_j, l=0\\ (1-a_j^{(l)})d_j^{(l-1)}+a_j^{(l)}d_{j+1}^{(l-1)},\\ j=i-k,i-k+1,\cdots,i-l; l=1,2,\cdots,k\end{cases} \tag{5}$$

$$a_j^{(l)}=\frac{u-u_{j+l}}{u_{j+k+1}-u_{j+l}} \tag{6}$$

Here, 0/0=0.

Because the r order derivative vector of k order B-spline curve is *(k-r)* order B-spline curve, the new derivative vector of B-spline curve can still be used in Equations 4-6. For the r order derivative vector $q^{(r)}(u)$ [7],

$$q^{(r)}(u)=\frac{\partial^r}{\partial u^r}\sum_{j=0}^{n}d_jN_{j,k}(u)=\sum_{j=i-k}^{i-r}d_j^{(r)}N_{j,k-r}(u) \tag{7}$$

The new control points are,

$$d_j^{(l)}=\begin{cases}d_j, l=0\\ (k-l+1)\dfrac{d_{j+1}^{(l-1)}-d_j^{(l-1)}}{u_{j+k+1}-u_{j+1}}\end{cases} \tag{8}$$

Here, $j=i-k,i-k+1,\cdots,i-r; l=1,2,\cdots,r$.

The knot vector is,

$$U^r=\left[u_0^r,u_1^r,\cdots,u_{n+k-2r+1}^r\right]=[0,\cdots,0,u_{k+1},\cdots,u_n,1,\cdots,1] \tag{9}$$

in which, the number of 0 and 1 is *(k+1-r)*.

Let $C(u)=\sum_{i=0}^{n}\omega_i d_i N_{i,k}(u)$, $W(u)=\sum_{i=0}^{n}\omega_i N_{i,k}(u)$,

From Equations 4-6, we can get $p(u)$ from $C(u)$ and $W(u)$,

$$p(u)=C(u)/W(u) \tag{10}$$

Differentiating eq.(10), we can achieve the one order derivative vector of $P(u)$,

$$p'(u)=\left(W(u)C'(u)-W'(u)C(u)\right)\big/[W(u)]^2 \tag{11}$$

Differentiating eq.(11), we can achieve the two order derivative vector of $P(u)$,

$$p''(u)=\frac{C''(u)[W(u)]^2-C(u)W''(u)W(u)-2C'(u)W(u)W'(u)+2C(u)[W'(u)]^2}{[W(u)]^3} \tag{12}$$

in which, $C'(u)$, $W'(u)$, $C''(u)$, $W''(u)$ can be computed by de Boor algorithm of Equations 4-9. Equations 11 and 12 will be used in the NURBS real-time interpolation and feedrate adaptation control algorithm.

D. The NURBS Curve Real-time de Boor Interpolation Algorithm Without Velocity Control

For NURBS interpolation, this procedure is to compute the value of u_{i+1} base on u_i . So it is a procedure of convolution computing for u_i.

By differentiating u to t for the second order Taylor series expansion, we obtain:

$$u_{i+1}=u_i+(t_{i+1}-t_i)\left.\frac{du}{dt}\right|_{t=t_i}+\frac{1}{2}(t_{i+1}-t_i)^2\left.\frac{d^2u}{dt^2}\right|_{t=t_i}+H.O.T \tag{13}$$

in which, $t_{i+1}-t_i=T$.

For three-axis interpolation,

$$\left.\frac{du}{dt}\right|_{t=t_i}=\left.\frac{ds/dt}{ds/du}\right|_{t=t_i}=\frac{V(t_i)}{\sqrt{(x'(u_i))^2+(y'(u_i))^2+(z'(u_i))^2}} \tag{14}$$

$$\begin{aligned}\left.\frac{d^2u}{dt^2}\right|_{t=t_i}&=\left.\frac{\frac{du}{dt}}{dt}\right|_{t=t_i}=\frac{\left.\frac{dV(t)}{dt}\right|_{t=t_i}}{\sqrt{(x'(u_i))^2+(y'(u_i))^2+(z'(u_i))^2}}\\&-\frac{V(t_i)^2(x'(u_i)x''(u_i)+y'(u_i)y''(u_i)+z'(u_i)z''(u_i))}{\left((x'(u_i))^2+(y'(u_i))^2+(z'(u_i))^2\right)^2}\end{aligned} \tag{15}$$

Substitute Equation 14 and 15 into Equation 13, and in which, $x(u_i)', x(u_i)'', y(u_i)', y(u_i)'', z(u_i)', z(u_i)''$ can be computed by de Boor algorithm from Equations 7-12. For constant velocity NURBS interpolation, $V(t_i)=V_0$, $\left(dV(t)/dt\right)\big|_{t=t_i}=0$ and neglect the high-order terms, so

$$u_{i+1}=u_i+\frac{TV_0}{\sqrt{(x'(u_i))^2+(y'(u_i))^2+(z'(u_i))^2}}-\frac{T^2V_0^2(x'(u_i)x''(u_i)+y'(u_i)y''(u_i)+z'(u_i)z''(u_i))}{2((x'(u_i))^2+(y'(u_i))^2+(z'(u_i))^2)^2} \quad (16)$$

From Equation 16, and the value of u_{i+1}, the interpolation coordinate, in the next cycle can be calculated based on u_i.

The flowchart of the NURBS curve real-time de Boor interpolation algorithm with no velocity control is shown in Figure 1, in which Fg is the flag of interpolation finishing and (x_e, y_e, z_e) is the end point coordinate value of NURBS curve.

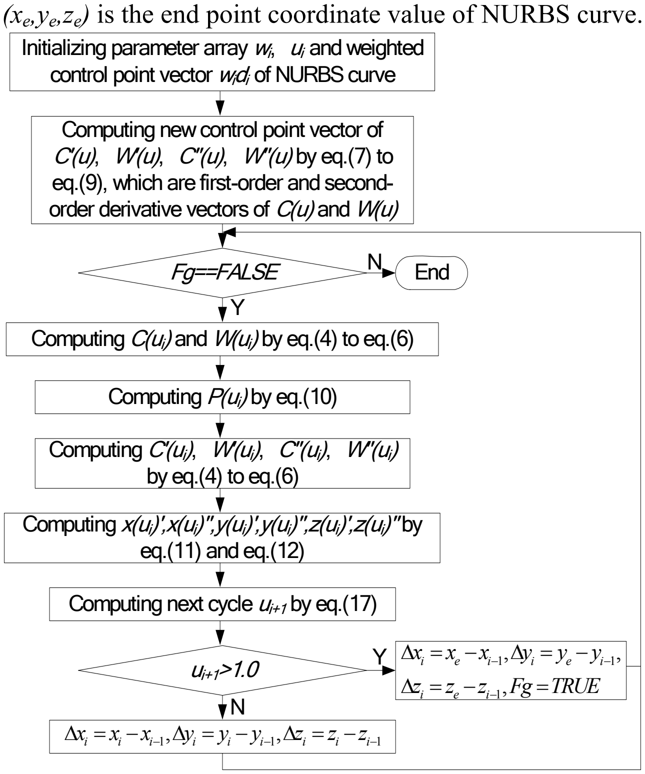

Fig.1 The flowchart of NURBS curve real-time de Boor interpolation algorithm

III. Model of NURBS Curve Feedrate Adaptation Control

In this section, we propose our feedrate adaptation control NURBS interpolation technique.

A. NURBS Curve Pre-interpolation acc/dec

1) Pre-interpolation acc/dcc and The Tail Phenomenon

Because the pre-interpolation is performed through a discreterization computation procedure, the velocity and the distance may not arrive at the zero point simultaneously in the end. So it takes extra time to adjust the distance at a very low velocity. This is the "tail" phenomenon. As the tail will lower the accuracy and increase the interpolation processing time, we need to remove it in the interpolation.

The key to solve the tail problem is to calculate the actual deceleration point. A deceleration point is the transfer point from a constant velocity zone into a deceleration zone. Theoretical deceleration point is defined as the point obtained based on the idealized continuous velocity profile; actual deceleration point is defined as the point from which the interpolation actually begins to decelerate. For linear or arc interpolation procedure, it is relatively easy to calculate the deceleration point [8]. However, the problem becomes more complicated for NURBS interpolation.

In NURBS curve acc/dec, because the shape of curve is related with velocity, the prediction of deceleration zone is very difficult. In order to compute the deceleration point, the new flag should be introduced. In this paper, we use the velocity instead of length to predict the deceleration point.

In order to compute the deceleration point of NURBS curve. We propose a new concept, NURBS reverse interpolation. NURBS reverse interpolation is the procedure in which we let the start point of NURBS curve become end point, the end point become start point, and then generate the NURBS curve by the reverse interpolation. NURBS curve reverse interpolation is used to compute the start velocity $V_{achieve}$ of deceleration zone. If $V_{achieve}$ is less than or equal to the next cycle velocity V_{next}, it should begin to decelerate in current cycle. And then, let the current velocity $V_{current}$ equal to $V_{achieve}$, and let the current u value equal to $1\text{-}u_r$, where u_r is corresponding to velocity $V_{achieve}$. Let $V_{achieve}$ be the start velocity, $1\text{-}u_r$ be start u value, and a_l be the acceleration in deceleration zone, the velocity V and knot vector u can reach the zero point simultaneously. Next we give our detailed algorithm for calculating the deceleration point.

2) The Method for Determining NURBS curve deceleration point

In NURBS curve forward interpolation procedure, it should predict the next cycle velocity V_{next} repeatedly, and the reverse interpolation is running simultaneously to compute the velocity $V_{achieve}$. And follow the whole procedure, u changes from zero to $1\text{-}u_c$, where u_c is corresponding to the current interpolation point. And the interpolation uses a_l as the acceleration value. The relationship between parameters of the reverse NURBS interpolation and forward interpolation is,

$$\begin{cases}\omega_{r,i}=\omega_{n-i}, d_{r,i}=d_{n-i}\\ u_{r,i}=1.0-u_{n+k+1-i}\\ \text{here, } i=0,1,\cdots,n\end{cases} \quad (18)$$

The recursive formula of reverse interpolation $u_{r,i}$ is,

$$u_{r,i+1}=u_{r,i}+\frac{TV_{r,i}+\frac{T^2}{2}\frac{dV_{r,i}}{dt}\Big|_{t=t_i}}{\sqrt{(x'(u_{r,i}))^2+(y'(u_{r,i}))^2+(z'(u_{r,i}))^2}}-\frac{T^2{V_{r,i}}^2(x'(u_{r,i})x''(u_{r,i})+y'(u_{r,i})y''(u_{r,i})+z'(u_{r,i})z''(u_{r,i}))}{2((x'(u_{r,i}))^2+(y'(u_{r,i}))^2+(z'(u_{r,i}))^2)} \quad (19)$$

The recursive procedure is finished when $u_{r,i+1} \geq 1-u_c$, and the times of iteration n' is computed. Then the computation formula of $V_{achieve}$ is,

$$V_{achieve} = \sum_{i=1}^{n'} a_l \cdot T \tag{20}$$

The flowchart of computing $V_{achieve}$ is given in Figure 2. Here, Fgc represents the finishing flag of NURBS reverse interpolation.

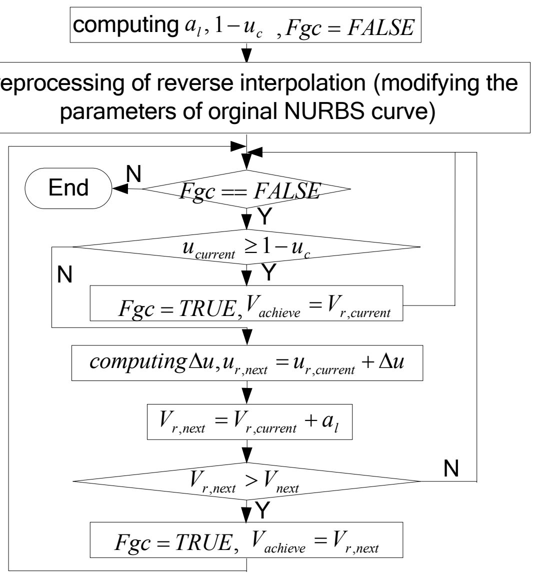

Fig.2 Flowchart of computing Vachieve

In Figure 2, in order to improve the efficiency, during the computation of reverse interpolation, when the velocity $V_{r,current}$ corresponding to u is bigger than velocity V_{next} at the next cycle, we jump out of the reverse interpolation procedure and enter into the calculation for next cycle because the current point does not enter into the deceleration zone. With this method, the recursive procedure is dramatically simplified. So the real-time property of CNC can be satisfied.

B. Feedrate Adaptation Control of NURBS Curve

1) The Analysis and Processing of Chord Error

During NURBS curve interpolation, for each interpolation cycle, it is the procedure of line substituting arc. Because all the interpolation points are on the NURBS curve, there is no accumulative error. But the chord error is introduced as shown in Figure 3.

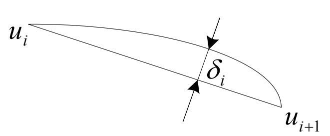

Fig.3. principle chart of chord error δ

The chord error δ is related to the feed step length ΔL, and the curvature radius ρ. If using arc segment instead of curve, in the normal condition, $\rho >> \delta$, the chord error relation formula of NURBS curve interpolation can be represented as:

$$\Delta L_i = \sqrt{4\delta_i\left(2\rho_i - \delta_i\right)} \approx \sqrt{8\rho_i\delta_i} \tag{21}$$

In practical, in order to limit the chord error to a permitted range during whole NURBS interpolation, the largest permitted chord error δ_{max} is given. When the current chord error is larger than δ_{max}, the chord error is restricted on δ_{max} by modifying the feed step length ΔL_i to $\Delta L_{i,c}$ based on Equation 21,

$$\Delta L_{i,c} = \sqrt{4\delta_{\max}\left(2\rho_i - \delta_{\max}\right)} \approx \sqrt{8\rho_i\delta_{\max}} \tag{22}$$

The calculation formula of the curvature radius ρ_i in Equations 21 and 22 is,

$$\rho_i = \frac{1}{k} = \frac{\left[\left(x'(u_i)\right)^2 + \left(y'(u_i)\right)^2 + \left(z'(u_i)\right)^2\right]^{\frac{3}{2}}}{\left[\begin{vmatrix} y'(u_i) & z'(u_i) \\ y''(u_i) & z''(u_i) \end{vmatrix}^2 + \begin{vmatrix} z'(u_i) & x'(u_i) \\ z''(u_i) & x''(u_i) \end{vmatrix}^2 + \begin{vmatrix} x'(u_i) & y'(u_i) \\ x''(u_i) & y''(u_i) \end{vmatrix}^2\right]^{\frac{1}{2}}} \tag{23}$$

in which, $x(u_i)'$, $y(u_i)'$, $z(u_i)'$, $x(u_i)''$, $y(u_i)''$, $z(u_i)''$ are calculated by de Boor algorithm in Equations 7 and 12.

From Equations 22 and 23, we can obtain $\Delta L_{i,c}$; if $\Delta L_{i,c} < \Delta L_i$, then let feed step length be $\Delta L_{i,c}$. So the chord error can be limited in a range to guarantee the accuracy of the interpolation track.

2) The Analysis and Processing of Acceleration

When dealing with acc/dec of NURBS curve, we should not only concern the tangent acceleration along the NURBS curve, but also concern the normal acceleration of the curve. When the velocity is very high and the curvature radius is very small, the oversize normal acceleration will cause the path accuracy lower and bring unnecessary impact to machine tool. So we should restrict the resultant acceleration under the permitted value of machine tool. We should also adjust the feed step length to satisfy the constraint of permitted acceleration.

Let the tangent acceleration of ith interpolation point be $a_{i,\tau}$, normal acceleration be $a_{i,n}$, the unit tangent vector be τ, unit normal vector be n, and the permitted maximum acceleration be a_{max}, then the relation formula is ,

$$\left|a_{i,n}n + a_{i,\tau}\tau\right| < a_{\max} \tag{24}$$

That is,

$$a_{i,c} = \sqrt{a_{i,n}^2 + a_{i,\tau}^2} = \sqrt{\frac{V_i^4}{\rho_i^2} + a_{i,\tau}^2} = \sqrt{\frac{\Delta L_i^4}{T^4\rho_i^2} + a_{i,\tau}^2} < a_{\max} \tag{25}$$

Comparing Equation 21 with Equation 25, we obtain:

$$a_{i,c} = \sqrt{\frac{64\delta_i^2}{T^4} + a_{i,\tau}^2} < a_{\max} \tag{26}$$

Deforming Equation 26, we obtain

$$\delta_i < \frac{T^2}{8}\sqrt{a_{\max}^2 - a_{i,\tau}^2} \tag{27}$$

in which, $a_{i,\tau}$ is the changing value of velocity ΔV_i in consecutive interpolation cycles. From Equation 27, the restriction of NURBS curve acceleration converts to the restriction of chord error.

Let $\delta_{a\max} = \frac{T^2}{8}\sqrt{a_{\max}^2 - a_{i,\tau}^2}$. When $\delta_{amax} < \delta_{max}$, substitute

δ_{amax} into Equation 21. Then we can obtain the restriction feed step length $\Delta L_{i,a}$ which is generated by acceleration constraint.

Based on the above analysis and $\Delta L_i = V_i \cdot T$, we obtain

$$\Delta L_{i,s} = \min\left\{\Delta L_i, \Delta L_{i,c}, \Delta L_{i,a}\right\} \tag{28}$$

$$V_{i,s} = \frac{\Delta L_{i,s}}{T} \tag{29}$$

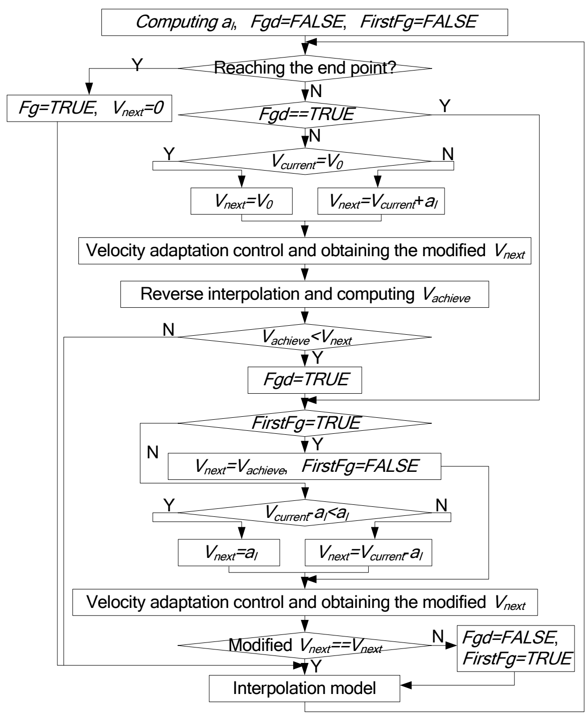

Fig.4 flowchart of pre-interpolation linear acc/dec of NURBS curve

From Equation 29, the whole NURBS curve interpolation procedure achieves the feedrate adaptation ability, which is, by repeatedly changing the velocity, the error and acceleration of NURBS curve are restricted under the permitted values.

Combining the feed adaptation control algorithm and the velocity model of pre-interpolation linear acc/dec, the real-time character of NURBS interpolation can finally be satisfied.

Suppose *Fg* is the end flag of NC program, *Fgd* is flag of entering the deceleration zone, *FirstFg* is flag of entering the deceleration zone at the first time, the flowchart of pre-interpolation linear acc/dec of NURBS is shown in Figure 4.

IV. Simulation and Experiment Result

We conduct a series of experiments to verify the effectiveness and correctness of our approach. The experimental parameters are given as follows: the order of NURBS curve is k=2, the feedrate is V_0=12.0m/min, the linear acc/dec acceleration time is t_l=1200ms, the interpolation cycle is T=4ms, the permitted maximum chord error is δ_{max}=10μm, the permitted maximum acceleration is a_{max}=2.5m/s^2. The parameter points we test in the experiments are shown in Table 1.

Parameter number	Coordinate value of control points(mm)	weights	Knot vector
1	(0,0,0)	1	0,0,0,0.06 ,0.09,0.15 ,0.20,0.24 ,0.32,0.40 ,0.56,0.60 ,0.66,0.72 ,0.80,0.88 ,0.94,1,1, 1
2	(-50,80,80)	25	
3	(0,100,100)	25	
4	(50,80,80)	25	
5	(0,0,0)	25	
6	(80,30,30)	25	
7	(100,0,0)	25	
8	(80,-30,-30)	25	
9	(0,0,0)	25	
10	(50,-80,-80)	25	
11	(0,-100,-100)	25	
12	(-50,-80,-80)	25	
13	(0,0,0)	25	
14	(-80,-30,-30)	25	
15	(-100,0,0)	25	
16	(-80,30,30)	25	
17	(0,0,0)	1	

TALBE.1 DATA TABLE OF NURBS PARAMETER POINTS

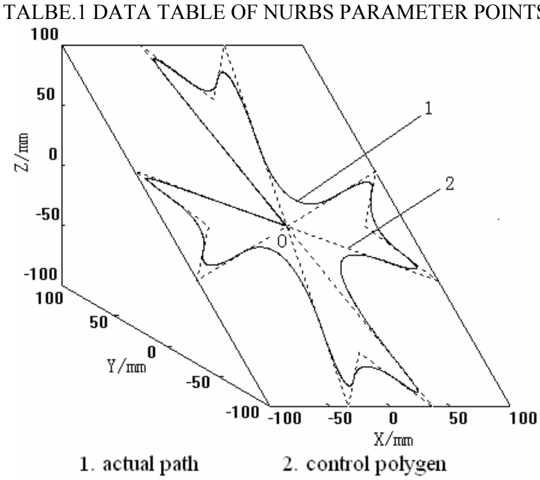

Fig.5 simulation of NURBS curve figure

The NURBS simulation curve which is generated by the new second-order de Boor algorithm is shown in Figure 5. The NURBS curve is drawn in a three-axis x-y-z coordinate system. In this figure, 1 is the NURBS curve using the parameters in Table 1, while 2 is from the original control polygon whose vertexes are control points.

Figure.6 shows the changes of curvature k of NURBS along the NURBS curve interpolation, in which x-axis represents the NURBS parameter u and y-axis represents k, the vertical coordinate.

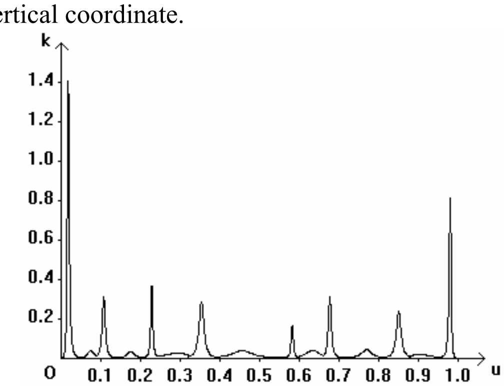

Fig.6 curvature changing figure along NURBS curve

Figure.7, Figure.8 and Figure.9 shows the changes of velocity, chord error and consultant acceleration of whole NURBS interpolation respectively when time elapses.

From Figures 6-9, we can see that:

1) Comparing with Figure 6, Figure 8 and Figure 9, corresponding the maximum value of curvature radius k, the chord error δ and resultant acceleration $a_{i,c}$ are also reached the maximum value point. 2) From Figure 7, in order to improve the system accuracy and acceleration property, it decelerates when the chord error or the consultant acceleration becomes bigger, which increases the processing time. 3) From Figure 8, when concerning the interpolation algorithm with limited chord error, it can successfully limit the chord error under the permitted maximum chord error $\delta_{max}=10\mu m$. Because this restriction is also applied for the acceleration with the maximum permitted resultant acceleration a_{max}, we can further decrease the chord error and improve the accuracy of path by more than 10 times. 4) From Figure9, when concerning the interpolation algorithm of restricting permitted maximum resultant acceleration a_{max}, the acceleration can be successfully restricted under the maximum acceleration $a_{max}=2.5m/s^2$. And this computation procedure has significant effect in acceleration extreme points.

V. Conclusion

In this paper, we proposed an integrated approach to solve the feedrate adaptation control NURBS interpolation problem in CNC systems. In our approach, we considered interpolation and feedrate control together with the tail handling. We proposed a de-Boor-algorithm-based NURBS interpolation model with the second-order Taylor expansion and designed a new algorithm to calculate actual deceleration points to remove the "tail" based on a novel concept, reverse NURBS interpolation. We proposed a novel feedrate adaptation control algorithm to supplement the tail removal in velocity control. We conducted a series of experiments and the experimental results show that our integrated method can significantly improve the accuracy.

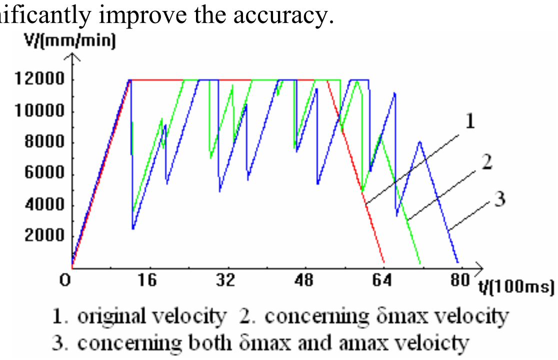

Fig.7 comparison figure of velocity

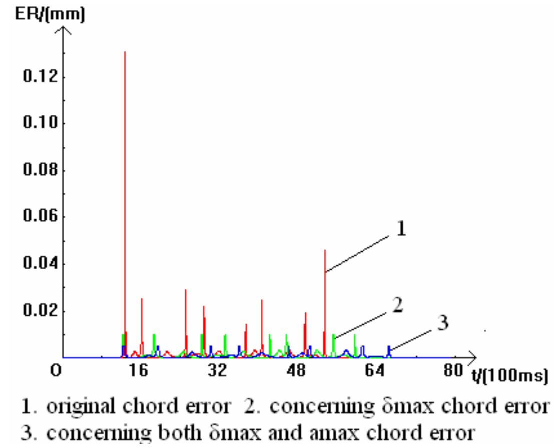

Fig.8 comparison figure of chord error

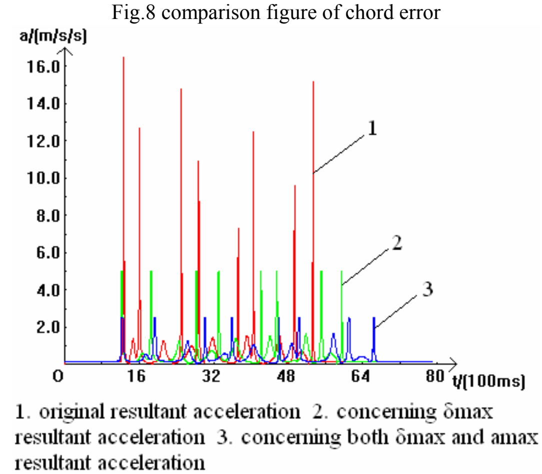

Fig.9 comparison figure of consultant acceleration

References

[1] You Youpeng, Wang Min, Zhu Jianying, "An Interpolator for NURBS Curve Machining with High-Speed and High Accuracy," *Journal of Computer Aided Design & Computer Graphics*, Vol.13, No.10, pp.943-947, Oct.2001.

[2] Marchenko Tikhon, Tae Jo Ko, Seung Hyun Lee and Hee Sool Kim, "NURBS interpolator for constant material removal rate in open NC machine tools," *International Journal of Machine Tools and Manufacture*, Vol.44, No.2-3, pp.237-245, Feb.2004.

[3] Tae Jo Ko, Hee Sool Kim and Sung Ho Park, "Machineability in NURBS interpolator considering constant material removal rate," *International Journal of Machine Tools and Manufacture*, Vol.45, No.6, pp.665-671, May.2005.

[4] Peng Fangyu, He Ying, Luo Zhongcheng, Li Bin, "NURBS curve interpolation algorithm with the adaptation to machine tool kinetic character," *Journal of Huazhong University of Science and Technology*, Vol.33, No.7, pp.80-83, July.2005.

[5] H. B. Jung, K. Kim, "A New Parameterisation Method for NURBS Surface Interpolation," *The International Journal of Advanced Manufacturing Technology*, Vol.16, No.11, pp.784-790, Sept.2000.

[6] Valle, L., Rivas, F., Catedra, M.F., "Combining the moment method with geometrical modelling by NURBS surfaces and Bezier patches," *Antennas and Propagation, IEEE Transactions on*, Vol.42, No.3, pp.373-381, March 1994.

[7] M. -C. Tsai, C. -W. Cheng and M. -Y. Cheng, "A real-time NURBS surface interpolator for precision three-axis CNC machining," *International Journal of Machine Tools and Manufacture*, Vol.43, No.12, pp.1217-1227, Sept.2003.

[8] Xianming Chen, Riesenfeld, R.F., Cohen, E., "Degree Reduction for NURBS Symbolic Computation on Curves," Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on, Vol.14-16, pp.28-28, June 2006.