Journal of Southeast University (English Edition) Vol. 22, No. 1, pp. 73 – 77 Mar. 2006 ISSN 1003—7985

Direct interpolation algorithm for pen-cutting of sculptured surfaces

Meng Shuyun Zhao Dongbiao

(College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: A direct interpolation algorithm for spatial curves on a surface is proposed for pen-cutting of sculptured surfaces. The algorithm can carry out the direct interpolation for projective curves lying on the sculptured surface. It is based on the geometric and kinetic relationships between drive curves and cutter-contact (C-C) curves. It evaluates the parameter of drive curves corresponding to interpolation points by the Taylor formula. Then it gains coordinates of interpolation points indirectly by inverse calculation. Finally, it generates the motion commands for machine tool controller. This method extends the locus-controlled function in the computer numerical control (CNC) system effectively and improves the efficiency for the numerical control (NC) machining of sculptured surfaces. The simulation shows that the proposed algorithm is feasible and practical. This algorithm can also be applied to the machining of the whole surface.

Key words: sculptured surface; pen-cutting; direct interpolation

The traditional approach, for instance zig/zag cutting or spiral cutting, is inefficient for satisfying the needs of sculptured surface machining, because it can make the cutter move on this area of the surface that need not be machined. Therefore, a new mode of surface machining—pen-cutting is proposed. It can drive the cutter along specified free-form curves on the surface. This can improve surface machining.

Today, many researchers have studied curve or surface interpolators to solve these problems, such as NURBS interpolator[1-3], B-spline interpolator[4], cubic parametric spline interpolator[5], PH curve interpolators[6], hybrid interpolator[7] and surface interpolator[8-10]. But these interpolators can be used when tool paths are NURBS or B spline, etc. When tool paths are not these curves, tool paths need to be approximated to these curves. The procedure is complex and approximate errors would be introduced[11,12]. Hence the aim of this study is to develop an interpolation algorithm based on the Taylor's expansion for realizing one kind of free-form curve-on-surface interpolator on the open-architecture CNC system.

1 Curve-on-Surface of Parametric Modeling

The non-uniform B-spline method is one of the methods used to sculptured surface modeling in the field of CAD/CAM because it has many strong points. A general form to describe a parametric surface in 3-D space can be expressed as

$$\boldsymbol{S}(u, v) = \sum_{i=1}^{m} \sum_{j=1}^{n} \boldsymbol{d}_{i,j} N_{i,3}(u) N_{j,3}(v) \tag{1}$$

where $\boldsymbol{d}_{i,j}$ are 3-D control points; $N_{i,3}(u)$, $N_{j,3}(v)$ are called the blending functions. The recurrence formulas for computing $N_{i,3}(u)$, $N_{j,3}(v)$ are shown in the following equations, respectively.

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

$$N_{i,3}(u) = \frac{u - u_i}{u_{i+3-1} - u_i} N_{i,2}(u) + \frac{u_{i+3} - u}{u_{i+3} - u_{i+1}} N_{i+1,2}(u)$$

$$N_{j,1}(v) = \begin{cases} 1 & v_j \leqslant v < v_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{j,3}(v) = \frac{v - v_j}{v_{j+3-1} - v_j} N_{j,2}(v) + \frac{v_{j+3} - v}{v_{j+3} - v_{j+1}} N_{j+1,2}(v)$$

where $\boldsymbol{U} = \{u_i, \dots, u_{i+3}\}$ represents the knot vector for the u direction, respectively. The same to parameter v. Likewise, a free-form curve can be expressed as

$$\boldsymbol{P}(t) = \sum_{i=1}^{w} \boldsymbol{p}_i V_{i,3}(t) \qquad t_3 \leqslant t \leqslant t_w \tag{2}$$

where $V_{i,3}$ is the cubic blending function, $\boldsymbol{p}_i$ are the 3-D control points, and the knot vector is $\boldsymbol{W} = \{t_0, t_1, \dots, t_{w+3}\}$.

Pen-cutting of sculptured surfaces differs from traditional methods in tool path planning. Now take milling a groove on the surface as an example. Fig. 1 illustrates the paths by applying the traditional method. Fig. 2 illustrates the path by the new machining method. From Fig. 2 it can be seen that the procedure planning tool path for pen-cutting is firstly to plan the tool path for the whole surface without regard to the groove, and secondly to plan the tool path for the

Received 2005-06-06.

Foundation item: The Natural Science Foundation of Jiangsu Province (No. BK2003005).

Biographies: Meng Shuyun(1978—), female, graduate; Zhao Dongbiao (corresponding author), male, doctor, professor, zdbme@nuaa.edu.cn.

groove after the tool path planning for the whole surface is finished. Especially when the groove is very narrow, we use a big diameter cutter to machine the whole surface and then use a small diameter cutter to machine the groove. Thus, it not only improves the efficiency of surface machining but also satisfies the precision of groove and avoids damaging on the machining tools because of uneven cutting. However, in Fig. 1 the smaller diameter cutter must be used to mill in order to satisfy the precision of groove in the whole procedure of surface machining. It is obvious that the advantage of pen-cutting is to shorten the machining time, thereby improving the efficiency of surface machining.

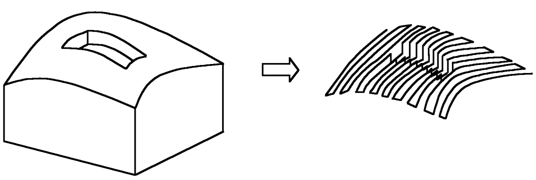

Fig. 1 Tool path by applying traditional zig/zag method

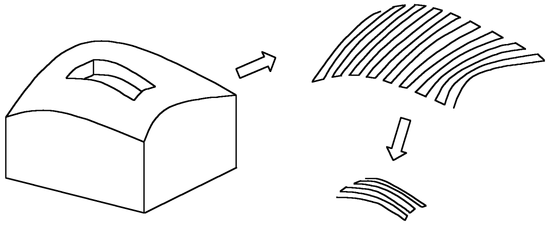

Fig. 2 Tool path by applying pen-cutting method

This paper does some research on the direct interpolation of curve on the sculptured surface. This is a projective curve that is generated by projecting one free-form curve onto the sculptured surface along a specified vector.

Let $\boldsymbol{P}_1(t)$ be the specified curve, which is a spatial free-form curve derived from translation and rotation of initial curve $\boldsymbol{P}(t)$, also defined as drive curve. It is expressed as

$$\boldsymbol{P}_1(t) = \boldsymbol{P}(t)\boldsymbol{A} + \boldsymbol{B} \tag{3}$$

where $\boldsymbol{A}$ is the rotation matrix and $\boldsymbol{B}$ is the translation matrix.

Suppose one vector $\boldsymbol{O}$ as a projective direction, then the free-form curve-on-surface of parametric model can be expressed as

$$\left.\begin{aligned} \boldsymbol{S}(u,v) &= \sum_{i=1}^{m}\sum_{j=1}^{n} \boldsymbol{d}_{i,j} N_{i,k}(u) N_{j,l}(v) \\ \boldsymbol{F}(t,n) &= \boldsymbol{P}_1(t) + n\boldsymbol{O} \end{aligned}\right\} \tag{4}$$

where $\boldsymbol{F}(t,n)$ is the parametric drive surface; t, n are two parameters of this surface. The C-C path is the intersection of a parametric surface S and a parametric surface F.

2 Algorithm Design

It is difficult to derive the mathematical expression of the intersection from formula (4). This is because two surfaces have common characteristics of bi-parameters and high degree and parameter coupling. The following gives the direct interpolation algorithm for the C-C path curve that has not a direct mathematical expression by an indirect approach.

Suppose that Q is a point on the intersection. Vectors $\boldsymbol{N}_S$ and $\boldsymbol{N}_F$ are surface normal vectors of surface S and surface F at Q such that $\boldsymbol{N}_S = \boldsymbol{S}_u \times \boldsymbol{S}_v$ and $\boldsymbol{N}_F = \boldsymbol{F}_t \times \boldsymbol{F}_n$. A tangent vector $\boldsymbol{T}_{\text{intersect}}$ at Q on the intersection can be expressed as

$$\boldsymbol{T}_{\text{intersect}} = \frac{\boldsymbol{N}_S \times \boldsymbol{N}_F}{|\boldsymbol{N}_S \times \boldsymbol{N}_F|} \tag{5}$$

The differential form of one curve on the drive surface F can be expressed as

$$\mathrm{d}\boldsymbol{l} = \boldsymbol{F}_t \mathrm{d}t + \boldsymbol{F}_n \mathrm{d}n \tag{6}$$

where $\mathrm{d}t$ and $\mathrm{d}n$ are the corresponding infinitesimal changes in t and n directions.

The drive curve and the intersection of surface S and F all lie on the surface F. When a C-C point B is moving with a constant velocity along the intersection, there is the corresponding point A moving on $\boldsymbol{P}_1$. The two points are entitled "point couple". They are shown in Fig. 3.

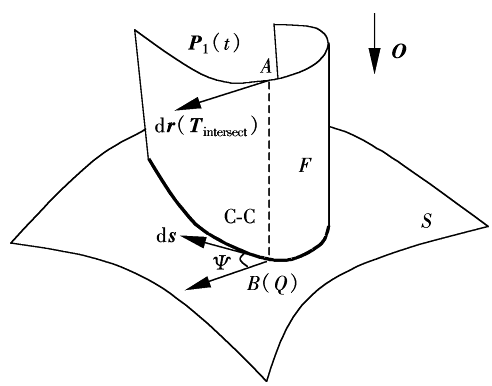

Fig. 3 Principle of the algorithm

Let $\mathrm{d}\boldsymbol{s}$ be the differential form of tangent vector on the intersection at one C-C point B, and let $\mathrm{d}\boldsymbol{r}$ be the differential form of tangent vector on drive curve at A. There is the following geometrical relation between $\mathrm{d}\boldsymbol{s}$ and $\mathrm{d}\boldsymbol{r}$:

$$\cos\Psi = \left.\frac{\mathrm{d}\boldsymbol{s} \cdot \mathrm{d}\boldsymbol{r}}{|\mathrm{d}\boldsymbol{s}||\mathrm{d}\boldsymbol{r}|}\right|_{(A,B)} = \left.\frac{\boldsymbol{T}_{\text{intersect}} \cdot \boldsymbol{P}_1'(t)}{|\boldsymbol{P}_1'(t)|}\right|_{(A,B)} \tag{7}$$

where Ψ is the angle between $\mathrm{d}\boldsymbol{s}$ and $\mathrm{d}\boldsymbol{r}$, and (A, B) is "point couple".

The length of tiny segment for the intersection can be expressed as

$$|\mathrm{d}\boldsymbol{s}| = \boldsymbol{T}_{\text{intersect}} \cdot \mathrm{d}\boldsymbol{l} \tag{8}$$

where $|\mathrm{d}\boldsymbol{s}|$ is the length of curve on surface F along the direction of the tangent vector of intersection.

Suppose $\mathrm{d}T$ is a tiny segment of time. Feed rate $\boldsymbol{V}$ can be expressed as

$$\boldsymbol{V} = \frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}T} \tag{9}$$

This is the velocity of C-C point when the cutter moves along the tangent vector of the intersection. Thus the relation between the velocity of moving along the drive curve and the intersection can be approximated as

$$\left|\frac{\mathrm{d}\boldsymbol{P}_1(t)}{\mathrm{d}T}\right| = \frac{\left|\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}T}\right|}{\cos\Psi} = \frac{|V|}{\cos\Psi} \tag{10}$$

The velocity of point A can be calculated by Eq. (10) under the condition that $\mathrm{d}\boldsymbol{s}$ is small. The velocity at A can also be represented as

$$\frac{\mathrm{d}\boldsymbol{P}_1(t)}{\mathrm{d}T} = \frac{\mathrm{d}\boldsymbol{P}_1(t)}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}T} \tag{11}$$

The second derivative of $P_1(t)$ to T is obtained as

$$\frac{\mathrm{d}^2\boldsymbol{P}_1(t)}{\mathrm{d}T^2} = \frac{\mathrm{d}^2\boldsymbol{P}_1}{\mathrm{d}t^2}\frac{\mathrm{d}t}{\mathrm{d}T} + \frac{\mathrm{d}\boldsymbol{P}_1}{\mathrm{d}t}\frac{\mathrm{d}^2 t}{\mathrm{d}T^2} \tag{12}$$

where

$$\frac{\mathrm{d}^2\boldsymbol{P}_1(t)}{\mathrm{d}T^2} = \frac{\mathrm{d}}{\mathrm{d}T}(|\boldsymbol{V}|\cos\psi) = |\boldsymbol{V}|\frac{\mathrm{d}}{\mathrm{d}T}\left(\frac{\boldsymbol{T}_{\text{intersect}} \cdot \boldsymbol{P}_1'(t)}{|\boldsymbol{P}_1'(t)|}\right) = \frac{|\boldsymbol{V}|}{|\boldsymbol{P}_1'(t)||\boldsymbol{N}_S \times \boldsymbol{N}_F|}\frac{\mathrm{d}((\boldsymbol{N}_S \times \boldsymbol{N}_F) \cdot \boldsymbol{P}_1'(t))}{\mathrm{d}T} \tag{13}$$

Therefore, the second-order interpolation expression is

$$\boldsymbol{t}_{\text{next}} = \boldsymbol{t}_{\text{current}} + \left.\frac{\mathrm{d}t}{\mathrm{d}T}\right|_{t_{\text{current}}} \Delta T + \frac{1}{2!}\left.\frac{\mathrm{d}^2 t}{\mathrm{d}T^2}\right|_{t_{\text{current}}} (\Delta t)^2 + R_n(\Delta t) \tag{14}$$

where $R_n(\Delta t)$ is higher order terms of the Taylor's expression. The value of $\boldsymbol{t}_{\text{next}}$ depends on the curvature of curve $P_1(t)$. The higher order terms can be neglected when the curvature is big. Now the point on the drive curve can be obtained. The other point of the point couple on the intersection can also be indirectly obtained by formula (4).

3 Simulation Study

In this simulation study, the interpolation algorithm was written by Visual C++ 6.0. Suppose that the feed rate command is 6 000 mm/min and the sampling interval is 1 ms. The presented interpolator was applied to the cubic non-uniform B-spline curve as shown in Fig. 4 and the compound non-uniform B-spline curve as shown in Fig. 5.

For the case of Fig. 4, the machining G code is given in Tab. 1. The vector $\{a, b, c\}$ is a projective direction vector. The drive curve is projected on the mouse upper surface along the vector $\{0.000, -0.503, -0.864\}$. Now one G code can carry out the interpolation of this Z-shape curve. The result is shown in Fig. 4.

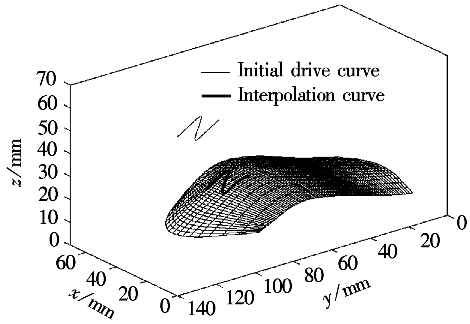

Fig. 4 Interpolation of Z-shape curve

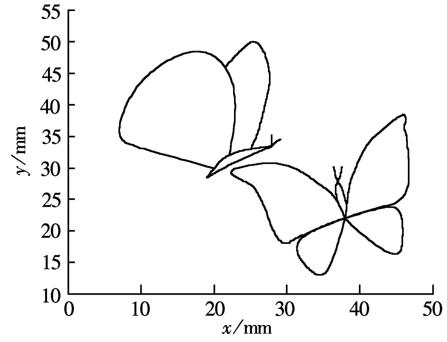

Fig. 5 Initial compound curves—butterfly

Tab. 1 Z-shape curve interpolation G code

No.	G code	x/mm	y/mm	z/mm	Feed
N010	G06.4	x40.000	y119.370	z42.347	F6000
		x40.000	y113.319	z45.867	
		x40.000	y106.405	z49.890	
		x35.000	y113.319	z45.867	
		x30.000	y119.370	z42.347	
		x30.000	y113.319	z45.867	
		x30.000	y106.405	z49.890	
		a0.000			
		b −0.503			
		c −0.864			

For the case of Fig. 4, Fig. 6 shows the error produced by using the interpolation algorithm proposed above. The error distribution is plotted out in Fig. 6.

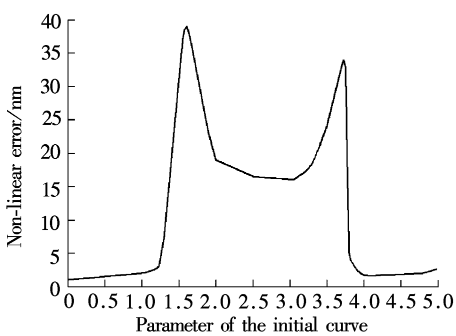

Fig. 6 Error analysis

From Fig. 6, it is obvious that the actual maximum error is less than 40 nm and the error will reach a maximum at the maximum curvature of the curve. Feed for each axis is presented in Tab. 2.

Tab. 2 Feed for each axis

No.	$\Delta x/\mu m$	$\Delta y/\mu m$	$\Delta z/\mu m$	$\Delta\alpha/('')$	$\Delta\beta/('')$
1	−85.8	−30.9	42.2	−17.28	2.52
2	−100.6	−7.1	32.4	−20.52	3.24
3	−107.9	18.9	19.0	−22.32	3.60
4	−105.6	39.3	6.1	−21.96	3.60
5	−99.1	52.0	−3.6	−20.52	3.24
6	−92.4	59.5	−10.2	−19.08	3.24
7	−86.7	64.2	−14.8	−18.00	2.88
8	−82.1	67.3	−18.2	−16.92	2.88

For the case of Fig. 4, there are 383 line segments of G code for constant C-C point feed rate machining if line interpolation function is applied. It can be seen that the transmission between CAD/CAM and CNC system is heavy.

Now we modify the shape of surface and give compound B-spline curves by replacing the above control points with new control points. The initial curve is shown in Fig. 5. Then we project these compound curves along vector $\{0.003, -0.927, 0.374\}$. The result is shown in Fig. 7.

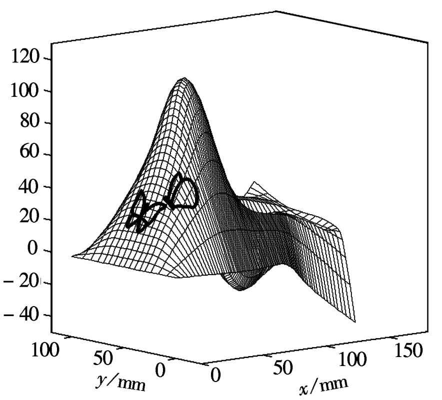

Fig. 7 Interpolation of compound curves—butterfly

From the above illustration different free-form curves are interpolated for pen-cutting of sculptured surfaces. These figures give us the path of C-C points clearly. The simulation indicates that the algorithm is applicable to the pen-cutting system of sculptured surfaces. Compact NC codes can carry out the interpolation of curve that is not expressed by the direct mathematical function. This is the characteristic of the algorithm. Once a C-C point is obtained by the given algorithm, the corresponding C-L point can be obtained by offsetting the radius of tool.

4 Conclusion

A new mode for surface machining—pen-cutting—is proposed in this paper. This method can improve the efficiency of surface machining.

A new algorithm is proposed for the interpolation of tool path derived from pen-cutting of surface machining. This tool path is the intersection of two parametric surfaces. The algorithm is based on the second order interpolation of the intersection curve and keeps the velocity of C-C point constant. The interpolation points stay on the parametric surface.

Simulations of the interpolation for tool path of pen-cutting have been carried out to verify the effectiveness of the proposed algorithm. These show that the proposed algorithm is useful not only for the interpolation of tool path in the parametric surface but also path planning for robots.

References

[1] Bian Yuchao, Zhang Liyan, Dai Yingying, et al. A study on the algorithm for real-time interpolation of NURBS curve in CNC system [J]. *Machine Building & Automation*, 2003, **32**(6): 36 − 39. (in Chinese)

[2] Tsai M C, Cheng C W, Cheng M Y. A real-time NURBS surface interpolator for precision three-axis CNC machining [J]. *International Journal of Machine Tools and Manufacture*, 2003, **43**(12): 1217 − 1227.

[3] Tikhon Marchenko, Ko Tae Jo, Lee Seung Hyun, et al. NURBS interpolator for constant material removal rate in open NC machine tools [J]. *International Journal of Machine Tools and Manufacture*, 2004, **44**(2, 3): 237 − 245.

[4] Lartiguea C, Thiebauta F, Maekawa T. CNC tool path in terms of B-spline curves [J]. *Computer-Aided Design*, 2001, **33**(4): 307 − 319.

[5] Bahr B, Xiao X, Krishnan K. A real time scheme of cubic parametric curve interpolations for CNC systems [J]. *Computers in Industry*, 2001, **45**(3): 309 − 317.

[6] Farouki Rida T, Tsai Yi-Feng, Yuan Guo-Feng. Contour machining of free-form surfaces with real-time PH curve CNC interpolators [J]. *Computer Aided Geometric Design*, 1999, **16**(1): 61 − 76.

[7] Tam Hon-Yuen, Xu Haiyin, Tse Peter W. An algorithm for the interpolation of hybrid curves [J]. *Computer-Aided Design*, 2003, **35**(3): 267 − 277.

[8] Lin Rong-Shine. Real-time surface interpolator for 3-D parametric surface machining on 3-axis machine tools [J]. *International Journal of Machine Tools and Manufacture*, 2000, **40**(10): 1513 − 1526.

[9] Zhou Ji, Zhou Yanhong, Zhou Yunfei. Technology of CNC direct machining for free-form surfaces [J]. *High Technology Letters*, 1998, **8**(11): 30 − 34. (in Chinese)

[10] Ni Qimin, Wang Shuilai, Li Congxin, et al. Surface interpolation technique in CNC machining [J]. *Mechatronics*, 2000, **6**(6): 34 − 36. (in Chinese)

[11] Yang Changqi, Qin Datong, Shi Wankai. Tool paths in terms of NURBS curves for five-axis sculptured surface using flat-end tools [J]. *Journal of Chongqing University*: *Natural Science Edition*, 2003, **26**(2): 82 − 89. (in Chinese)

[12] Dong Chenshi, Chen Zhiyang, Wang Guozhao, et al. Optimal B-spline approximation of intersection curves of two surfaces [J]. *Journal of Computer-Aided Design & Computer Graphics*, 2004, **16**(4): 449 − 453. (in Chinese)

复杂曲面笔式加工的直接插补算法

孟书云　　赵东标

(南京航空航天大学机电学院, 南京 210016)

摘要:针对复杂曲面笔式加工时位于曲面上的空间曲线型刀具轨迹,给出一种直接插补算法. 即对以投影方式形成的位于曲面上的空间曲线形式的刀轨,根据导动曲线和刀触点轨迹线之间的几何运动关系,通过泰勒展开近似得到刀触点轨迹线上插补点所对应的导动线的参数,再反求间接得到刀触点坐标,从而生成控制机床运动的指令. 该方法的实现扩充了 CNC 系统的轨迹控制功能,提高了复杂曲面的加工效率. 仿真结果表明算法可行而且有效. 该算法也可以应用到整体曲面加工中.

关键词:复杂曲面;笔式加工;直接插补

中图分类号:TG659;TP391.7