ELSEVIER

PII: S0010-4485(98)00025-6

Computer-Aided Design, Vol. 30, No. 8, pp. 649–655, 1998

Printed in Great Britain
0010-4485/98/$19.00+0.00

A new approach to CNC tool path generation

Chih-Ching Lo

The feedrate is one of the most important factors to machining efficiency and quality. Current methods for tool path generation adopt a constant velocity along the cutter location path and do not satisfy the desired feedrate along the sculptured surface. This paper presents a new approach to tool path generation so as to cope with this problem. Methods based on linear, curve, and surface interpolators are presented and analyzed, respectively.

Keywords: computer numerical control, tool path generation, interpolator

INTRODUCTION

Computer-aided manufacturing (CAM) systems as well as computer numerical control (CNC) machine tools are widely used in today's manufacturing industries. To produce the desired part, the CAM system generates the tool path for the CNC machine tool so that it can cut the surfaces of the part. In the current approach[1–4], the CAM system first computes the cutter contact (CC) points and then offsets these points to get the cutter location (CL) points. The CC points are a set of points located on the sculptured surface so that the *edge* of the tool is scheduled to pass through, while the CL points are a set of points that the *center* (or the bottom tip) of the tool is scheduled to pass through. In this paper the CC path and the CL path are defined as the paths passing through the CC points and the CL points, respectively. In addition, the CC velocity and the CL velocity are defined as the velocities along the CC path and the CL path, respectively.

The CL path is in practice the tool path. Conventionally, the CL path is represented by a set of consecutive linear segments (that connect the CL points), and then fed to the CNC machine that adopts a linear interpolator. The interpolator implemented in the CNC machine can convert the cutter path to motion trajectories of all the axes in order to coordinate their motion in multi-axis machining. Because the above method requires a large CL file for these linear motion commands and results in feedrate fluctuation along the cutter path[5], parametric curves are recommended to represent the CL path. To achieve this purpose, curve interpolators are continuously being developed[6–9].

Department of Mechanical Engineering, Feng Chia University, Taichung, Taiwan

The feedrate is one of the most important cutting conditions that determine the machining efficiency and quality. Because the production rate is proportional to the feedrate, a large feedrate is recommended. However, on the basis of cutting principles as well as experts' experiences, the feedrate must be small enough so as to achieve a quality machining that is in terms of dimension accuracy and surface integrity[10–12]. The feedrate denotes the velocity that the cutter edge moves along the part and is in practice the CC velocity defined above. In the existing approach, the CL path together with a preset (usually constant) CL velocity is fed to the CNC machine. Although the preset CL velocity is usually called the feedrate in a lot of literature, it is not the real one related to the machining efficiency and quality. For this reason, the existing approach to tool path generation cannot satisfy the desired feedrate (i.e., CC velocity) requirement, and consequently, may result in low efficient and/or low quality machining.

This paper presents a new approach to tool path generation so as to cope with the above problem. Three methods based on linear, curve, and surface interpolators are developed and analyzed, respectively.

EXISTING APPROACH

The existing approach to tool path generation can be depicted by *Figure 1*. In this approach, the first step is to schedule the CC paths for the cutter edge to follow. The path interval (or side step) between two neighboring CC paths is determined based on the limit of the scallop height[3]. Current CAM systems frequently adopt the isoparametric machining method that assigns the CC path along the *u*- or the *v*-direction of the parametric surface. Usually, the isoparametric machining method does not correspond to a constant scallop height along the CC path, and a conservative path interval is determined so that the maximum scallop height all over the CC path will not exceed the limit (typical value is 0.001~0.01 mm). However, constant scallop-height methods can be adopted to improve the machining efficiency[2,4].

The second step is to offset the CC path so as to get the CL path that the cutter center is scheduled to pass through. The path offset depends on the surface and the tool geometry. Analytical solutions for the CL path are only available for simple contours such as straight lines and circular arcs. For general curves, the CL path is originally represented by a series of CL points. Note that the number and the locations of the CC points must be determined previously so as to get the corresponding CL points. Usually, the forward step size for two neighboring CC points is chosen so that the chordal

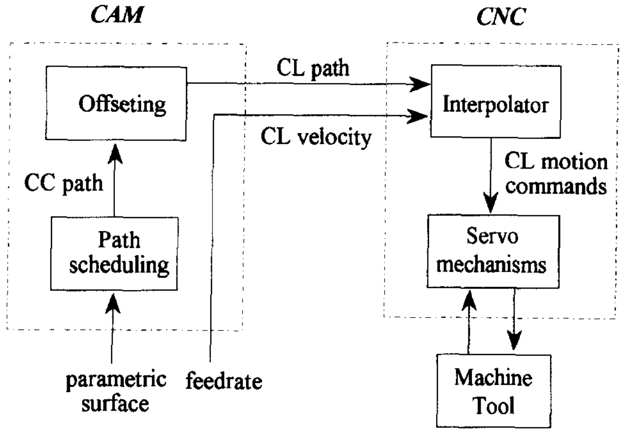


Figure 1 The existing approach to tool path generation

deviation will not exceed a limit[3] (typical value is 0.001~0.01 mm).

The final step is to feed the CL path and the CL velocity to the interpolator implemented in the CNC machine, which can convert the cutter path to motion trajectories of all the axes in order to coordinate their motion in multi-axis machining. Conventionally, the CL path is approximated by a set of consecutive linear segments (that connect the CL points), and then fed to the CNC machine that adopts a linear interpolator. Because this method requires a large CL file for the linear motion commands and results in feedrate fluctuation and non-smoothness on the joints of the linear segments, parametric curves are recommended for approximation of the CL path[5,7]. However, curve interpolators must be available in the CNC machines.

In this paper, we consider a parametric surface that is described by

$$S = S(u,\ v) \tag{1}$$

where u and v are the spatial parameters. The CC and the CL paths are denoted as P^{cc} and P^{cl}, respectively. The CC and CL velocities are denoted as V^{cc} and V^{cl}, respectively. Besides, we adopt the isoparametric machining method and assign the CC path along the u-direction, i.e., $P^{cc}(u) = S(u, v^*)$, where v^* is fixed. A ball-end mill that is frequently utilized for three-axis surface machining is considered here. The equation for cutter offsetting can be formulated as

$$p^{cl} = p^{cc} \pm r\mathbf{n} \tag{2}$$

where r is the radius of the cutter; the plus–minus ($\pm$) sign depends on whether the sculptured surface is convex or concave; $\mathbf{n}$ is the unit normal vector to the surface and can be calculated by

$$\mathbf{n} = \frac{\dfrac{\partial S}{\partial u} \times \dfrac{\partial S}{\partial v}}{\left| \dfrac{\partial S}{\partial u} \times \dfrac{\partial S}{\partial v} \right|} \tag{3}$$

CC VELOCITY VERSUS CL VELOCITY

In the existing approach referred to in *Figure 1*, a CL velocity that is set as the desired feedrate is fed to the CNC interpolator. Consequently, a constant CL velocity is maintained along the CL path. However, the surface cutting occurs along the CC path, rather than the CL path. Therefore, we should be concerned with the CC velocity more than the CL velocity.

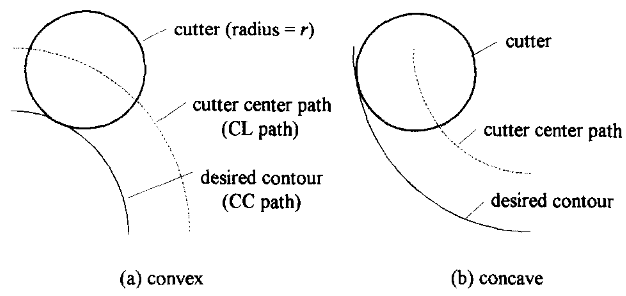


Figure 2 Machining of 2D circular contours

For the case of 2D circular contours, Koren[12] proposed a simple method to compensate for the CL velocity so as to maintain a constant CC velocity along the machined profile. As illustrated in *Figure 2*, in order to achieve the desired feedrate along the circular contour, we should either increase the CL velocity (for the convex case) or decrease the CL velocity (for the concave case). The feedrate compensation method proposed by Koren can be formulated as

$$V^{cl} = V^{cc}\frac{R \pm r}{R} \tag{4}$$

where R is the radius of the circular arc, r is the radius of the cutter, the plus–minus ($\pm$) sign depends on whether the circular contour is convex or concave.

Although Koren has pointed out the necessity for a constant CC velocity and provided a solution for 2D circular cases, the solution for 3D surface machining cases is not available. Since the existing tool path generation approach adopts a fixed CL velocity, the resultant CC velocity along the sculptured surface is variable. Note that eqn (4) is also adequate for relating the CC and the CL velocities for surface machining. But, the parameter R should be replaced by ρ, the radius of curvature on the surface in the CC path direction.

Because the existing approach results in a variable CC velocity, it will cause non-uniform machining efficiency and quality. Low efficiency occurs when the CC velocity is smaller than the preset feedrate. This is because we can increase the CL velocity without causing a CC velocity that exceeds the preset feedrate. Low quality occurs when the CC velocity is larger than the preset feedrate. Note that here we assume all the other machining parameters (such as spindle speed, tool wear, up/down cutting, etc.) related to the surface quality are fixed. In practical machining, a CC velocity that exceeds the preset feedrate may not necessarily be harmful to surface quality (because in the industry a conservatively low feedrate is usually utilized). However, in the view of machining efficiency, a less conservative feedrate can be utilized if a constant CC velocity can be well controlled.

First of all, a demonstration example that shows the non-uniform machining efficiency and quality due to the existing tool path generation approach is given in the following.

Example 1. A revoluted surface is described by

$$\begin{cases} x = (60u^3 - 90u^2 + 90u + 20)\cos(v) \\ y = (60u^3 - 90u^2 + 90u + 20)\sin(v), \\ \quad z = 60u^3 - 90u^2 + 50 \\ \quad 0 \le u \le 1, \ -\pi \le v \le -0.2\pi \end{cases} \tag{5}$$

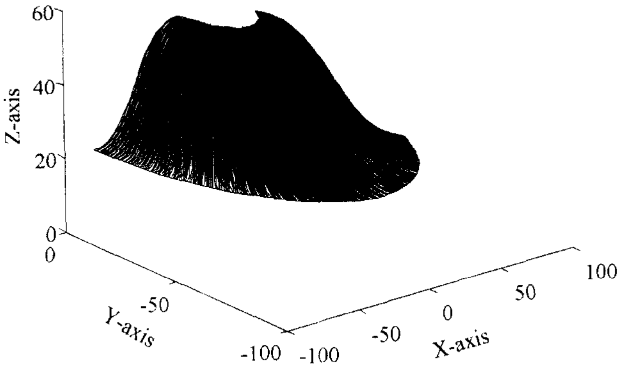


Figure 3 A revoluted surface and the desired CC paths

The above parametric surface and the desired CC paths are shown in *Figure 3*. In the machining, a ball-end cutter (whose radius is 10 mm) is utilized and 319 CC paths are assigned so that the scallop height will not exceed a limit of 0.005 mm. A specified CC path (along $v = -0.6\pi$) and the corresponding CL path are shown in *Figure 4*. The correspondent CC and CL velocities are shown in *Figure 5*. As can be seen in *Figure 5*, when utilizing a constant CL velocity (=10 mm/s), the CC velocity varies along the path. The variable CC velocity will result in low efficiency (for $0 < u < 0.5$, the feedrate is smaller than the desired 10 mm/s) and low quality (for $0.5 < u < 1$, the feedrate is larger than 10 mm/s).

To achieve high efficient and quality machining, it is necessary to compensate for the CL velocity so that an invariant feedrate (or CC velocity) is maintained all over the CC path. To achieve this purpose, three methods based on different types of CNC interpolators are presented in the following.

METHOD BASED ON LINEAR INTERPOLATOR

The first and the second methods that are respectively based on linear and curve interpolators are illustrated in *Figure 6*. As can be seen, a compensation algorithm is introduced to modify the CL velocity so as to maintain a constant feedrate along the CC path. In this section, the method based on the linear interpolator is presented, while the method based on the curve interpolator will be presented in the next section.

As stated above, the CL path is frequently approximated

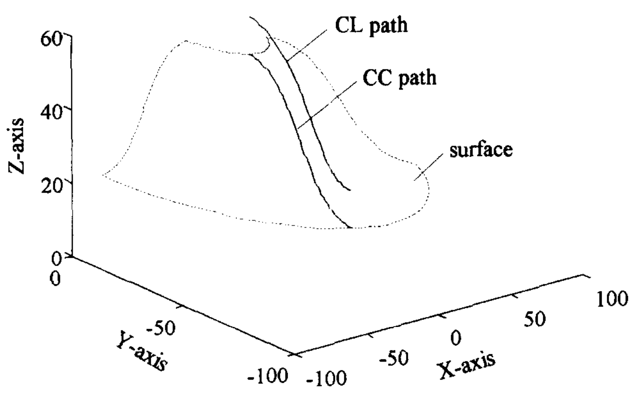


Figure 4 A specified CC path and its correspondent CL path

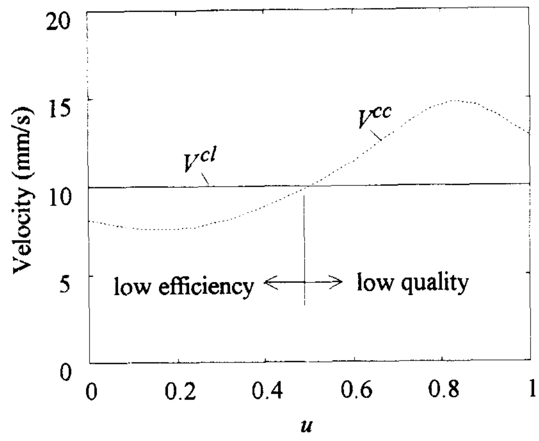


Figure 5 The CC and the CL velocities

by a set of consecutive linear segments connecting the CL points. To get the CL points, we need first to calculate the CC points and then to offset them. Note that the algorithm for the offsetting has been stated in eqn (2) eqn (3). The forward step along the CC path (i.e., the space between two consecutive CC points) is usually determined so that the chordal deviation error will not exceed a maximum allowable value. Consequently, the spatial parameter corresponding to each CC point can be determined recursively by

$$u_i = u_{i-1} + \frac{L^{cc}}{\left\|\frac{d}{du}(P^{cc})\right\|} = u_{i-1} + \frac{\sqrt{8\rho\delta}}{\left\|\frac{d}{du}(P^{cc})\right\|} \tag{6}$$

where L^{cc} is the distance between two consecutive CC points; ρ is the radius of curvature on the surface in the CC path direction; δ is the maximum allowable chordal deviation error. Consequently, we can get the CC point by substituting the spatial parameter into the CC path function, i.e.,

$$P_i^{cc} = P^{cc}(u_i)$$

To satisfy the desired machining feedrate, the CL velocity for the *i*th linear segment can be calculated by

$$V_i^{cl} = f\frac{L_i^{cl}}{L_i^{cc}} \tag{7}$$

where $f(=V^{cc})$ is the desired feedrate along the CC path; $L_i^{cl}(=|P_i^{cl} - P_{i-1}^{cl}|)$ and $L_i^{cc}(=|P_i^{cc} - P_{i-1}^{cc}|)$ are the lengths of the *i*th linear CL and CC segments, respectively.

The above method is the simplest one. The only modification on the existing method is to change the CL velocity one segment by one segment. Consequently, only a traditionally linear interpolator is required for the CNC machines. However, this method has the following drawbacks: (1) it requires a large CL file for these linear motion commands, and (2) it results in feedrate fluctuation.

According to the numerous CL segments, the size of the machining program generated by the CAM system is usually very large and may exceed the limit of the CNC's memory. Under this condition, the current approach is to break the machining program into several portions or to utilize an on-line communication between the CAM and the CNC systems. The former method requires interruption of the machining process, and consequently causes an increase in the machining time. In the latter method, the

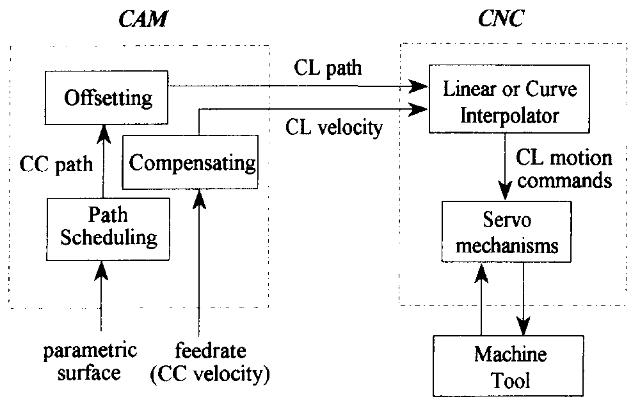


Figure 6 The proposed tool path generation approach (the first and second methods)

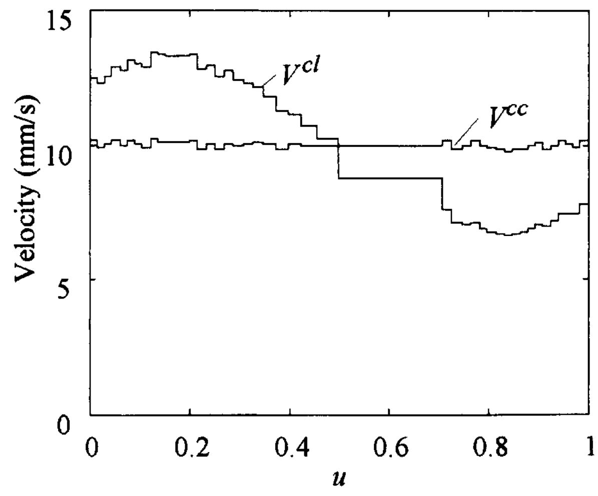


Figure 8 The fluctuation in the CL and the CC velocities

CAM system continuously generates commands to the CNC system while the machining is being executed. This requires very reliable CAM and CNC I/O interfaces so that the communication between the two systems will not cause any error during the machining.

The feedrate fluctuation is due to (1) the truncation error for each CL segment which must be correspondent to an integer number of sampling periods[5], and (2) the suddenly stepped change of the CL velocity from one CL segment to another. Both the two sources cause discontinuities in the CL velocity commands, and thereby, disturb the feed drives. It is obvious that if the feedrate fluctuation becomes too serious, the surface quality will be degraded.

In the following, we first present an example to illustrate the efficiency improvement according to the proposed method. Then, an example to show the problems of large CL file and feedrate fluctuation is given.

Example 2. A ruled surface is described by

$$\begin{cases} x = -30u^2 + 60u + 20 + 80v \\ y = -30u^2 + 60u + 20 - 20v, \quad 0 \leq u \leq 1, \; 0 \leq v \leq 1 \\ z = -30u^2 + 50 \end{cases} \tag{8}$$

For machining of the above surface, a ball-end cutter with a radius of 10 mm is utilized. The scallop height limit is set as 0.005 mm. Through the computer run, 132 CC (or CL) paths are required. Note that the algorithm for calculating the path interval (Δv) between two neighboring CC paths is given later in this paper (see to eqn (9) and eqn (10)). The chordal deviation limit is set as 0.005 mm. Based on eqn (6), each CC path should contains 45 CC points. Then, 44 CL linear segments are desired for each CL path. Consequently, 5808 (=44 segments/path × 132 paths) CL segments are required for machining of the ruled surface. The ruled surface and the desired CC paths are shown in *Figure 7*.

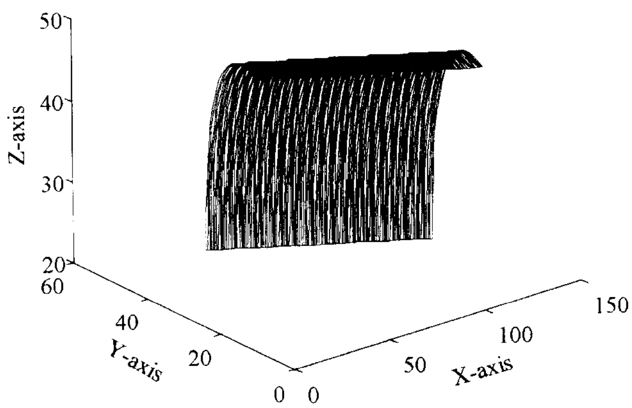


Figure 7 A ruled surface and the desired CC paths

Based on the existing tool path generation approach that adopts a constant CL velocity all over the machining, a total machining time of 14.2 min is required. When utilizing the proposed method that adopts a constant CC velocity all over the machining, a total machining time of 10.7 min is required. Consequently, the machining efficiency is improved by 25%.

Example 3. For machining of the revoluted surface described by eqn (5), the same machining parameters (cutter radius, limit of scallop height, etc.) as those in Example 2 are utilized. Through the computer run, 319 CL paths are required. Each CL path contains 46 linear CL linear segments. Consequently, 14 674 (=46 segments/path × 319 paths) CL segments are required for machining of the revoluted surface.

During the machining, the CNC linear interpolator breaks each CL segment into finer segments at the sampling period of 0.005 s. With a traditional linear interpolation algorithm, the CC and CL velocities for the tool path along $v = -0.6\pi$ are demonstrated by *Figure 8*, which shows the feedrate fluctuation in the CL and the CC velocities.

METHOD BASED ON CURVE INTERPOLATOR

The second method is similar to the first one (see *Figure 6*). However, the CNC linear interpolator is replaced by a curve interpolator. The design of curve interpolators is a feasible solution to reduce the memory for the CL file and the feedrate fluctuation duc to linear interpolator method[5–9]. However, in the existing approach, the curve geometric parameters for the CL path and a preset constant CL velocity are fed to the CNC interpolator that cannot satisfy the desired feedrate along the CC path. To cope with this problem, additional parameters are needed to be fed to the CNC interpolator that can generate a *variable* CL velocity (corresponding to the desired feedrate along the CC path) based on these parameters. In other words, in addition to offsetting the CL points and fitting them by parametric

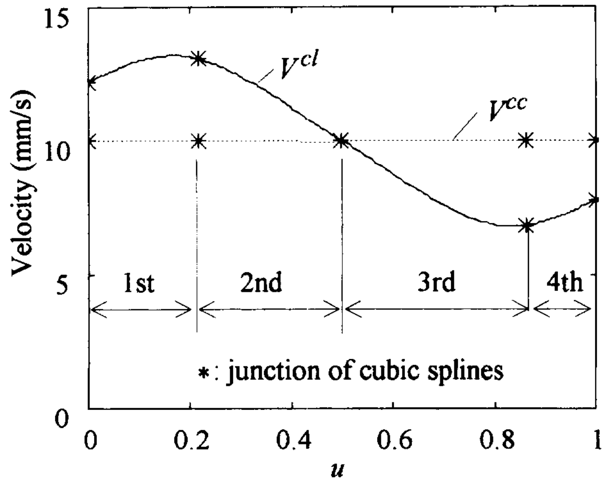


Figure 9 Results for fitting the CL path and velocity with four CS segments

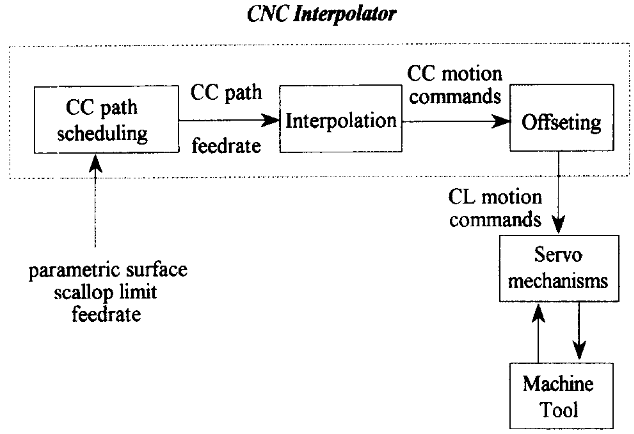


Figure 10 A surface interpolator that can accept surface machining command

functions, we need to calculate the desired CL velocities at the CL points and fit them by other parametric functions. In this paper, the composite cubic spline that is widely adopted in current CAD/CAM systems is utilized for fitting the profiles of the CL path and the CL velocity.

Example 4. Composite cubic spline functions are utilized to represent the desired CL path and CL velocity (along $v = -0.6\pi$) for the revoluted surface referred to in eqn (5). Note that the corresponding CC path is a single cubic spline and the CC velocity is 10 mm/s. In the curve fitting, the maximum allowable errors are: 0.005 mm for the CL path, and 1% (or 0.1 mm/s) for the CL velocity. Through the computer run, four cubic spline (CS) segments are required to represent the CL path. In addition, four CS functions are introduced to represent the CL velocity profile. Note that the segmentation for the CL velocity must be consistent with that for the CL path because they are fed together to the CNC machine.

Based on the CS fitting, the CL file is shortened (from 46 linear segments to four CS segments). However, the CL file for machining of the revoluted surface is still large ($319 \times 4 = 1276$ CS segments). The corresponding CL and CC velocities along the tool path are shown in *Figure 9*. As can be seen, an accurate and fluctuation-free feedrate generation has been achieved.

When adopting the above method, the CNC curve interpolator requires a minor modification that tunes the velocity along the cubic spline curve. However, the calculation of the velocity is just to substitute the current spatial parameter into the cubic polynomial function for the velocity.

METHOD BASED ON SURFACE INTERPOLATOR

As demonstrated by Example 4, the CL file for machining of the sculptured surface may still be very large, even though a curve interpolator method has been introduced. This is because the data condensation is conducted only in the tool path direction, not in the path interval direction. To achieve an effective data condensation in both directions, a feasible solution is to feed the surface parameters and the cutting conditions (such as feedrate, scallop height limit, etc.) directly to the CNC machine that adopts a *surface interpolator*. Consequently, the algorithms for the CC path scheduling and the tool offsetting that are implemented in the current CAM module are moved to the CNC interpolator. As is depicted in *Figure 10*, the proposed CNC surface interpolator consists of a CC path scheduling algorithm, an interpolation algorithm, and an offsetting algorithm. All the three algorithms are executed on-line the machining process. Since the CC path is generated in real time according to the preset feedrate, we can accurately maintain the desired feedrate along the CC path. The procedure for the tool path generation method is given below.

(1) Choose one of the surface parameters (say u) as the tool path direction. Thus the boundary curve along the other surface parameter (at $v = -\pi$ for the case in Example 1) is the first CC path.
(2) Interpolate (or generate) the CC point for every sampling period.
(3) Offset the CC point in order to get the CL point.
(4) For each CC point, calculate the radius (R_v) of curvature in the path interval (v) direction so as to find a candidate for the path interval (Δv_i).
(5) At the sampling rate of the CNC system, repeat Steps (2) through (4) until the end of the CC path. At the end of a tool path, choose the minimum one of the path interval values obtained along the path, i.e., $\Delta v = \min(\Delta v_i{'s})$, and then set the next CC path at $v_{k+1} = v_k + \Delta v$.
(6) Repeat Steps (2) through (5) until the side spatial parameter (v) reaches the other boundary (at $v = -0.2\pi$ for the case in Example 1).

In the above procedure, Steps (2), (3) and (4) are the core steps that are repeated at the sampling rate of the CNC system. Furthermore, the real-time calculation must be completed in a sampling period so as to maintain a sufficient CNC performance[13]. The desired real-time algorithms and the computational load for the three steps are stated in the following.

Step (2) is in practice the CC path interpolation. For iso-parametric machining, the CC path is chosen to be in the u- or the v-direction, while the path interval is in the other direction. Let the tool path be in the u-direction, then we have: $P^{cc}(u) = S(u, v_k)$ for the kth CC path. The interpolation algorithm generates the points (or position commands) along the CC path at the sampling rate. Many interpolation algorithms for curve generation have been

Table 1 Computational load in a sampling period for the surface interpolator.

Computational requirements	$\pm$	$\times$	$\div$	$\sqrt{}$	sin cos	Total CPU time (μs) (based on a PC-486-66 MHz)
Example 5$^{(u)}$	39	52	8	4	0	88
Example 5$^{(v)}$	21	40	8	4	2	80
Example 6	75	92	9	4	0	143

(u): tool path is in the u-direction; (v): tool path is in the v-direction.

developed[5-9]. Here we utilize the following algorithm[8]:

$$u_i = u_{i-1} + \frac{fT}{\left|\frac{dP^{cc}}{du}\right|} - \frac{(fT)^2\left(\frac{dP^{cc}}{du}\cdot\frac{d^2P^{cc}}{du^2}\right)}{2\left|\frac{dP^{cc}}{du}\right|^4} \quad (9)$$

where T is the sampling period; f is the desired feedrate along the CC path; u_i and u_{i-1} are the values of the spatial parameter at two consecutive sampling instants, iT and $(i-1)T$, respectively. Note that in the above equation, $dP^{cc}/du = \partial S/\partial u$ and $d^2P^{cc}/du^2 = \partial^2 S/\partial u^2$.

Based on eqn (9), we can calculate the spatial parameter, recursively, at each sampling instant, and then substitute it into the CC path function, i.e., $P_i^{cc} = P^{cc}(u_i) = S(u_i, v_k)$. Accordingly, we get the CC position command at the sampling instant.

The desired offsetting algorithm for Step (3) has been presented above. Please see eqn (2) and eqn (3).

For Step (4), we need first to calculate R_v, the radius of curvature in the path interval (v) direction. R_v can be calculated by[4,14]

$$R_v = \frac{E\alpha^2 + 2F\alpha + G}{L\alpha^2 + 2M\alpha + N} \quad (10)$$

where

$$\alpha = \frac{\frac{\partial S}{\partial v}\cdot\mathbf{t}}{\frac{\partial S}{\partial u}\cdot\mathbf{t}},\ E = \frac{\partial S}{\partial u}\cdot\frac{\partial S}{\partial u}\ F = \frac{\partial S}{\partial u}\cdot\frac{\partial S}{\partial v},\ G = \frac{\partial S}{\partial v}\cdot\frac{\partial S}{\partial v},$$

$$L = \frac{\partial^2 S}{\partial u^2}\cdot\mathbf{n},\ M = \frac{\partial^2 S}{\partial u\partial v}\cdot\mathbf{n},\ M = \frac{\partial^2 S}{\partial v^2}\cdot\mathbf{n}$$

where $\mathbf{n}$ is the unit normal vector to the surface and $\mathbf{t}$ is the unit tangent vector in the CC path direction. Since the CC path is assigned to be in the u-direction, we have:

$$\mathbf{t} = \frac{\partial S}{\partial u}/\left|\frac{\partial S}{\partial u}\right|$$

At each sampling instant, the suggested path interval can be calculated by [2]

$$\Delta v_i = \frac{l}{(n\times t)\cdot\frac{\partial S}{\partial v}} = \frac{\sqrt{\frac{8R_v rh}{R_v \pm r}}}{(n\times t)\cdot\frac{\partial S}{\partial v}} \quad (11)$$

where l is the incremental length for the path interval; h is the limit for the scallop height; r is the radius of the ball miller; the plus–minus ($\pm$) sign depends on whether the sculptured surface in the side direction is convex or concave.

Two examples are now given to show the computational load for the above real-time algorithms.

Example 5. For the revoluted surface described by eqn (5), the computational requirements for the proposed surface interpolator are summarized in *Table 1*. Both the cases that the tool path is in the u-direction or the v-direction are investigated. Note that for this example, the u-direction is orthogonal to the v-direction (i.e., $(\partial S/\partial u)\cdot(\partial S/\partial v) = 0$), and consequently, many formulae (e.g., eqn (10)) can be further simplified. Besides, the computational requirement also depends on the programming efficiency and the number of temporary variables introduced in the program. For this reason, the computational requirement listed in *Table 1* is only an approximated estimate. However, as is shown in *Table 1*, the computational requirement ($< 90\,\mu s = 0.09$ ms) shares only a small percentage of the sampling period (typical value[5,8] for current CNC machines is equal to or larger than 1 ms).

Example 6. For a general bicubic spline surface that is described by

$$S(u, v) = [u^3,\ u^2,\ u,\ 1][\mathbf{D}]\begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix},\ 0 \le u \le 1,\ 0 \le v \le 1 \quad (12)$$

where $[\mathbf{D}]$ is a 4×4 coefficient matrix. Since a general case is considered here (i.e., $[\mathbf{D}]$ is arbitrary), no simplification for the formulae can be made. Besides, the computational requirement will be the same no matter which direction (u or v) the tool path is assigned. The computational requirements for this case are also listed in *Table 1*. As can be seen, approximately double computational load is needed as comparing with the case in Example 5. However, the computational load will not cause a problem for current CNC systems.

CONCLUDING REMARKS

The feedrate is one of the most important factors to machining efficiency and quality. Current tool path generation methods, which adopt a constant velocity along the cutter location path, do not satisfy the desired feedrate (or cutter contact velocity) along the sculptured surface, and consequently, may result in low efficient and/or low quality machining. To cope with this problem, three methods that are based on linear, curve and surface interpolators, respectively, have been presented in this paper.

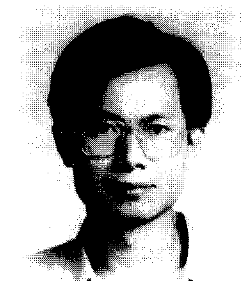
The first method requires only a traditionally linear interpolator in the CNC machine. In contrast to the existing method that adopts a constant CL velocity, the proposed method adapts the CL velocity one segment by one segment so that a constant CC velocity is maintained. Although this method is the simplest one, it requires a large CL file for these CL segments, and usually results in feedrate fluctuation.

The second method requires a modified curve interpolator in the CNC machine. In addition to the curve geometric parameters, the parameters related to the CL velocity profile (that corresponds to a constant CC velocity) are fed to the modified interpolator for the path generation. Consequently, off-line curve fitting for the CL path and the CL velocity is needed. This method can solve the problem of feedrate fluctuation and condense the CL file.

The third method requires a surface interpolator that is new to current CNC machines. The surface interpolator consists of three real-time algorithms for the interpolation, the offsetting and the path interval calculation (for path scheduling of the next CC path), respectively. This method can maintain the desired feedrate along the sculptured surface and condense the CL file significantly. Although a lot of real-time computation is needed for the surface interpolator, it does not cause a problem for current CNC systems.

REFERENCES

1. Hwang, J. S., Interference-free tool-path generation in the NC machining of parametric compound surfaces. *Computer-Aided Design*, 1992, **24**(12), 667–676.
2. Lin, R. S. and Koren, Y., Efficient tool-path planning for machining free-form surfaces. *ASME Journal of Engineering for Industry*, 1996, **118**, 20–28.
3. Loney, G. C. and Ozsoy, T. M., NC machining of free form surfaces. *Computer-Aided Design*, 1987, **19**(2), 85–90.
4. Suresh, K. and Yang, D. C. H., Constant scallop-height machining of free-form surfaces. *ASME Journal of Engineering for Industry*, 1994, **116**, 253–259.
5. Yang, D. C. H. and Kong, T., Parametric interpolator versus linear interpolator for precision CNC machining. *Computer-Aided Design*, 1994, **26**(3), 225–233.
6. Kiritsis, D., High precision interpolation algorithm for 3D parametric curve generation. *Computer-Aided Design*, 1994, **26**(11), 850–856.
7. Lo, C. C., Feedback interpolators for CNC machine tools. *ASME Journal of Manufacturing Science and Engineering*, 1997, **119**, 587–592.
8. Shpitalni, M., Koren, Y. and Lo, C.C., Realtime curve interpolators. *Computer-Aided Design*, 1994, **26**(11), 832–838.
9. Yeung, M. K. and Walton, D. J., Curve fitting with arc splines for NC tool path generation. *Computer-Aided Design*, 1994, **26**(11), 845–849.
10. Chou, J. J. and Yang, D. C. H., Command generation for three-axis CNC machining. *ASME Journal of Engineering for Industry*, 1991, **113**, 305–310.
11. Shaw, M. C., *Metal Cutting Principles*. Oxford University Press, 1984.
12. Koren, Y., *Computer Control of Manufacturing Systems*. McGraw-Hill, New York, 1983.
13. Astrom, K. J. and Wittenmark, B., *Computer Controlled Systems: Theory and Design*. Prentice-Hall, Englewood Cliffs, NJ, 1984.
14. Faux, I. D. and Pratt, M. J., *Computational Geometry for Design and Manufacturing*. John Wiley and Sons, New York, 1981.



Chih-Ching Lo received a BS in power mechanical engineering from the National Tsing-Hua University, Taiwan, in 1984. He received an MS and a PhD in mechanical engineering from the University of Michigan, USA, in 1989 and 1992, respectively. He is now an Associate Professor of the Mechanical Engineering Department, Feng-Chia University, Taiwan. His current research interests include computer-numerical control, robotics and computer-aided manufacturing.