

Smooth tool path generation for 5-axis machining of triangular mesh surface with nonzero genus



Yuwen Sun^{a,*}, Jinting Xu^{b,*}, Chunming Jin^b, Dongming Guo^a

^a School of Mechanical Engineering, Dalian University of Technology, Dalian, 116024, China

^b School of Automotive Engineering, Dalian University of Technology, Dalian, 116024, China

ARTICLE INFO

Article history:

Received 15 June 2015

Accepted 1 June 2016

Keywords:

Triangular mesh

5-axis machining

Tool path generation

Tool orientation optimization

ABSTRACT

NC machining of a nonzero genus triangular mesh surface is being more widely confronted than before in the manufacturing field. At present, due to the complexity of geometry computation related to tool path generation, only one path pattern of iso-planar type is adopted in real machining of such surface. To improve significantly 5-axis machining of the nonzero genus mesh surface, it is necessary to develop a more efficient and robust tool path generation method. In this paper, a new method of generating spiral or contour-parallel tool path is proposed, which is inspired by the cylindrical helix or circle which are a set of parallel lines on the rectangular region obtained by unwrapping the cylinder. According to this idea, the effective data structure and algorithm are first designed to transform a nonzero genus surface into a genus-0 surface such that the conformal map method can be used to build the bidirectional mapping between the genus-0 surface and the rectangular region. In this rectangular region, the issues of spiral or contour-parallel tool path generation fall into the category of simple straight path planning. Accordingly, the formula for calculating the parameter increment for the guide line is derived by the difference scheme on the mesh surface and an accuracy improvement method is proposed based on the edge curve interpolation for determining the cutter contact (CC) point. These guarantee that the generated tool path can meet nicely the machining requirement. To improve further the kinematic and dynamic performance of 5-axis machine tool, a method for optimizing tool orientation is also preliminarily investigated. Finally, the experiments are performed to demonstrate the proposed method and show that it can generate nicely the spiral tool path or contour-parallel tool path on the nonzero genus mesh surface and also can guarantee the smooth change of tool orientation.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Triangular mesh is commonly used in the fields of NC machining and rapid prototyping, etc., especially after the STL format has become a de facto standard data input in many commercial CAD/CAM systems [1]. Compared with the parametric model, triangular mesh is more flexible to approximate the sculptured surface with complex shapes and easy of gouge/collision checking in NC machining. However, triangular mesh machining is still confronted with many difficulties due to the lack of the feasible tool path methods. At present, most existing tool path generation methods are only suitable to the parametric surface, and as a result, iso-planar tool path, which is obtained by slicing the mesh

model by using a series of parallel planes, is sometimes the first choice for machining the triangular mesh even at the expense of machining quality and efficiency [2,3]. Though iso-planar path method is simple and robust, it has inherently the disadvantage of the excessive short paths resulted from the irregular boundaries which can usually increase the number of interruptions or rapid traversals during the machining process and decrease the machining efficiency [4], especially when machining the nonzero genus triangular mesh surface. Actually, in NC machining of the sculptured surfaces, the surfaces with holes, concave cavities or convex platforms that all can be viewed as nonzero genus surfaces are being more widely confronted than before. Particularly, in industries of consumer products, automotive and aerospace, etc. when machining a complex part on NC machine tool, tool path is often conducted on a nonzero genus surface. To realize high efficiency machining of the triangular mesh surface of this type, a flexible and efficient tool path, which includes continuous and smooth tool trajectory (also referred to as CC path) and

* Corresponding authors.

E-mail addresses: xians@dlut.edu.cn (Y.W. Sun), xujt@dlut.edu.cn (J.T. Xu).

gradually changing tool orientation that can improve significantly the kinematic and dynamic performance of 5-axis machine, is desired and is also the objective of this paper. In the context that follows, some background and past works will be first reviewed on tool trajectory planning and tool orientation optimization for 5-axis machining of the triangular mesh surface.

1.1. Tool trajectory planning

To enrich the path patterns suitable for the triangular mesh machining and improve the machining efficiency, some different strategies of planning tool trajectory have been proposed. Kim et al. [5] gave a cutter location (CL) surface deformation-based method. This method deforms the CL surface to a new surface and then inversely maps the curves obtained by slicing the surface into iso-scallop tool path. After extending the concepts of the guide plane and drive surface, Lee et al. [6] also obtained iso-scallop tool path using a series of planes driven by the side-steps to slice the CL surface. An iso-parameter tool path planning method on the triangular mesh is proposed by Sun et al. [7]. On the basis of this work, boundary-conformed and contour-parallel path methods have been also developed sequentially by Sun et al. [8,9] by means of the conformal map. Based on the built mapping relationship between the surface to be machined and 2D mapping domain, the complicated task of generating tool path is reduced from the surface to a plane so that the geometry computation related to tool path planning is greatly simplified and tool trajectories of different patterns can be designed conveniently. In the following sections, new spiral and contour-parallel tool trajectories methods which also benefit from this idea will be presented in detail.

Recently, some notable works have been also done for triangular mesh machining due to the increasing requirements in the real industry applications. Bolaños et al. [10] proposed a method for generating three-axis tool path on a sculptured surface represented by the triangular mesh, which focuses mainly on construction of the mesh offsets. Their method avoids the requirement for the topology of triangular mesh when offsetting the triangular mesh, thus speeding up the tool path generation. In the tool-adaptive offset tool path generation method proposed by Kout and Müller [11], the direction- and contour-parallel tool paths are generated as a family of iso-curves of an anisotropic distance function of a seed curve on the triangular mesh. Their method is based on an implicit path representation with emphasis on metric tensor for the adaptation of the scallop height or path interval, and the central work is to improve the path adaptation to various production processes, such as milling and spray coating. Zhang and Tang [12] developed an efficient greedy strategy for 5-axis tool path generation on a dense triangular mesh. In their method, the kinematic characteristics of the machine tool are considered, and the cutting strip width and the maximum feedrate at each mesh vertex are analyzed. The sequence of CC points is then determined using a greedy searching strategy from the mesh vertices. The generated tool paths can increase the efficiency of the final machining, but the resulting path patterns are uncontrollable.

The ultimate goal of tool path planning is to fulfill the machining requirements with the most suitable tool path pattern in an acceptable machining time. As Lee pointed out in [13], the best possible way to machine a sculptured surface would be to do in a continuous fashion, reducing the number of interruptions or rapid traversals during the machining operation. Zhang and Tang also mentioned in their conclusion [12], in 5-axis machining, especially in the stage of finish machining, smooth tool path patterns without sharp turning, such as the spiral path, are usually desirable. To generate the spiral tool paths for cutting the triangular mesh, Makhanov and Ivanenko [14] adopted grid generation technique in the tool path optimization to construct the spiral tool paths

embedded into the zigzag tool paths on the complex surface. By the conformal map, Sun et al. [15] proposed a spiral cutting strategy which can realize cutting of mesh surface without tool retractions. In [16], Hernández et al. also presented a spiral tool path generation method by the non-deterministic technique. In their method, the first 2D profile is created by projecting the part boundary on the z-plane and offset continuously toward inside, subsequently a 2D spiral path is constructed as a diagonal curve between two adjacent parallel profiles which is then projected onto the triangular mesh to form the spiral tool path. However, these methods above are only applicable to the genus-0 mesh surface and their extensions to the spiral tool path on the nonzero genus triangular mesh are not obvious. By far, smooth tool path, such as spiral tool path, method on the mesh surface with nonzero genus has not been still sufficiently addressed.

1.2. Tool orientation optimization

Tool orientation optimization is usually performed by optimizing the two angles, i.e. tilt angle α and yaw angle β , that define the tool orientation in 5-axis tool path. By far, most of existing methods of determining tool orientation are focused mainly on eliminating the local gouging and global collision in 5-axis machining, and the state of the art review can be referred to [17]. In addition, some works attempt to optimize the tool orientation at each CC point by matching the curvatures of the cutter swept envelope surface and the part surface [18], maximizing the machining strip width [19] or the material removal volume [20], minimizing the deflection cutting force [21] and considering process mechanics [22], etc. Since these methods optimize the tool orientation only at the vicinity of CC point, sometimes in order to reach the above optimal objectives or to avoid the local gouging and the obstacles, tool has to make a dramatic change in its tool orientation along CC path. Such an extreme change is not allowed in real machining due to the physical limitations such as the kinematical capacities of machine tools [23]. Thus, when optimizing the tool orientation, smooth continuous changes of tool orientation must be taken into account.

Initially, tool orientation smoothing is conducted in part coordinate system (PCS). For example, in the method proposed by Morishige et al. [24], 3D C-space which describes precisely the tool orientations without collision is first constructed in PCS and then a smoother curve in the 3D C-space is defined to the machining conditions to avoid the drastic angle changes. Jun et al. [25] also proposed a method based on C-space. Their method uses both forward and backward searching to find two paths from the boundaries of the feasible C-space region and the one with smaller change of angle is taken as the smooth tool orientations. Lauwers et al. [26] controlled the drastic changes of tool orientation by limiting the change of angle per unit distance. Wang and Tang [27] took into account the angular velocity limitation of tool and first generated the gouge-free and angular-velocity-compliant five-axis tool orientation, though also in PCS. To guarantee a constant cutting speed of tool edge, Farouki and Li [28] and Han [29] proposed, respectively, the methods optimizing tool orientation which can minimize the actuation of the rotary axes that orient the part relative to the tool. All the above works are done with respect to PCS, owing to the nonlinearity of the inverse kinematics (IK) from PCS to machine coordinate system (MCS), smooth tool orientation in PCS does not necessarily corresponds to smooth rotary axis movement in MCS, thus sometimes heavy dynamic loading on the rotary axes of 5-axis machine tools can occur yet, such as severe angular velocity and angular acceleration which are imposed on the A, B or C axis of a specific 5-axis machine tool [30].

It has been confirmed by Castagnetti et al. [31] that, the kinematic behavior of 5-axis machine tool can be further improved by

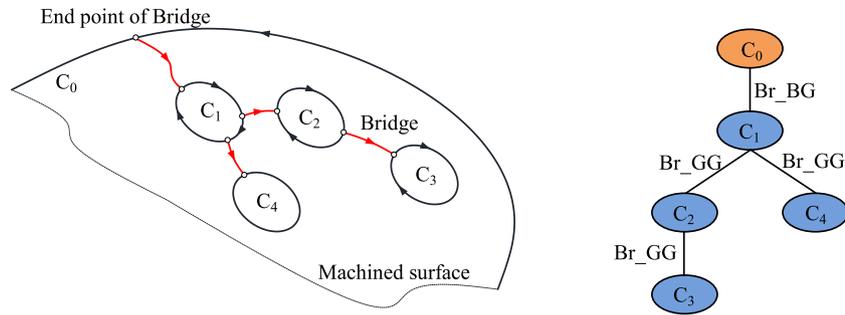


Fig. 1. Tree structure of the mesh surface boundaries. Br_BG denotes a bridge between the outer boundary and the interior boundary, and Br_GG denotes a bridge that connects two interior boundaries.

smoothing the rotary axis coordinates that define the tool orientation in MCS, in contrast with the methods optimizing the tool orientation in PCS. In their method [31], the changes of tool orientation in MCS are constrained in a DAO (Domain of Admissible Orientation) without local gouging and global collision and then a constrained optimization, which minimizes the angular difference between two CC points, is developed and solved by the gradient-based method in Matlab. The method proposed by Lavernhe et al. in [32] is to optimize the tool orientation under the kinematic constraints of machine tool in MCS, which also involves a complicated constraint optimization. Wang and Tang [23] presented a so called iso-conic concept, in which a special tool direction is selected such that the angular velocity and angular acceleration of the A-axis of a spindle-tilting machine is reduced to zero, but it only applies to the proposed path pattern by them. On the basis of the works [23,27], Hu and Tang [30] improved further the kinematics and dynamics of 5-axis machining by optimizing workpiece setup and tool orientation. In their method, as done in [23,31,32], the feasible tool orientation in PCS is first transformed into MCS by IK of 5-axis machine tool, after that, three objectives of minimizing angular acceleration constrained by feasible tool orientation, are presented and then are solved by the constraint optimization function in Matlab. It is noted that, the tool orientation optimization constrained by the machining interference or other limitations is not a trivial issue. Heavy computational time and less robustness are both the problems that the algorithms have to face. Also, the machine components, surface shapes and obstacles, etc. appear probably in various forms so that the feasible tool orientation region will have complex shape and irregular boundaries [33] which lead to that the constraints are difficult to be expressed analytically. In the situation, it is often not easy to solve the constrained optimization and even produce a proper initial solution to ensure the convergence to the optimization.

Decoupling the optimization problem solving from the constraints, as done by Ho et al. [34], may be an effective method to solve the above problems. The initial representative tool orientations at crucial areas are assigned and then the tool orientations at general areas are obtained by using the smoothing method to join those representative tool orientations. Of course, the machining interferences of the new generated tool orientations need to be checked. Compared with the complicated constraint optimization process, the idea of first smoothing and then checking is more simple and clear. Although the proposed method is similar in spirit with Ho's method, it is an overall least-squared optimization that performs directly on the rotary axis coordinates in MCS, not a locally linear interpolation between two assigned tool orientations in PCS as [34], so the proposed optimization model and solving method is quite different.

1.3. Contributions and organization of the paper

In this paper, a new method which can nicely generate smooth spiral or contour-parallel tool trajectory on the triangular mesh

with nonzero genus is first proposed. This method reduces the complicated task of generating tool path on the surface to the simple straight line planning on the plane and derives the formula of calculating the parameter increment for guide line such that the path interval can be limited strictly as well as the scallop height. The model of tool orientation optimization is also proposed, which is performed by smoothing directly the rotary axis coordinates in MCS, not the tool orientation traditionally in PCS, to minimize the angular velocity and angular acceleration in 5-axis machining and improve the kinematics and dynamics performance of 5-axis machine tool.

This paper is organized as follows. The genus-0 mesh surface construction, the basic theory of conformal map and its realization are first discussed respectively in Sections 2 and 3. Section 4 presents the methods of designing the path guide line and calculating its parameter increment, then CC path is generated in Section 5. Tool orientation is optimized subsequently in Section 6. Section 7 discusses the experiments. The concluding remarks and the future works are given in Section 8.

2. Construction of genus-0 mesh surface

In the proposed method, the nonzero genus mesh surface needs to be first transformed into a simply-connected surface by introducing a concept of branch cut in the physical region such that the following conformal mapping which applies to the genus-0 mesh surface can be used to generate the spiral or contour-parallel tool paths. In pocket machining, we have already noticed that PowerMill[®] and [35] adopt the line to bridge the islands and the outer boundary, however our bridge is constructed on the physical surface and the bridging algorithm is also different. Here, we assume that the outer and interior boundaries of mesh surface have been stored in a boundary set in which the 0th element is the outer boundary in the counterclockwise direction and the other elements are the interior boundaries in the clockwise direction, as shown in Fig. 1.

2.1. Tree structure of the boundaries

The branch cut in the physical region is realized by constructing the branch bridges between the outer boundary and one interior boundary and between the interior boundaries. The basic requirement for constructing the bridges is that only one interior boundary can be connected to the outer boundary and the other bridges are constructed between the interior boundaries. According to this rule, the outer and interior boundaries of the mesh surface can be well organized using a tree structure as shown in Fig. 1. This layout begins with the outer boundary C_0 , called the root node of the tree structure. Its child node can be specified by the interactive operation from the interior boundaries or select one interior boundary which can maximize or minimize the distance from C_0 . Specially, if

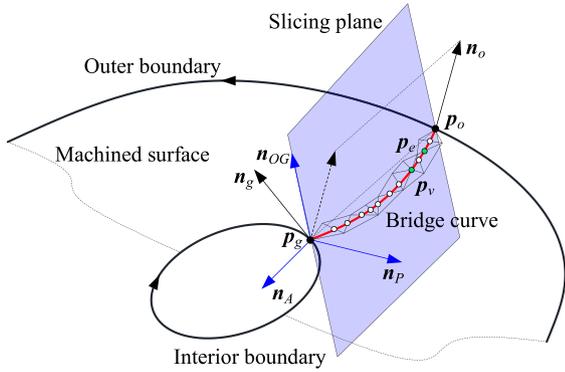


Fig. 2. Bridge curve obtained by slicing the machined surface.

the number of the interior boundaries (or holes) is more than two, all parent nodes, except the root node, have the equal opportunity to be linked to one potential interior boundary. Right of Fig. 1 gives the tree structure of left boundaries.

2.2. Constructing bridge

According to the above tree structure, it is easy to set a bridge between two nodes. For the convenience of explanation, the definition of the bridge and its data structure is first given as follows.

Definition 1. A bridge is a PS (point-sequence) curve on the machined surfaces that connects the parent node and its child node and is stored in the data structure OBridge {Bcurve1, Bcurve2, Pset}.

In OBridge, Bcurve1 and Bcurve2 are the indices of two nodes connected by the bridge, at the same time they also determine the direction of the bridge which is from Bcurve1 to Bcurve2, and Pset stores sequentially the bridge points according to this direction. According to two end points of the bridge, the bridge can be constructed by slicing the mesh surface using a plane through these two end points by the following method.

2.2.1. Calculating intersection point

As shown in Fig. 2, \mathbf{n}_o and \mathbf{n}_g are the normal of mesh surface at the end points of bridge, \mathbf{p}_o and \mathbf{p}_g . According to the geometry relationship shown in Fig. 2, the equation of the slicing plane can be written as

$$\begin{cases} (\mathbf{p} - \mathbf{p}_o) \cdot \mathbf{n}_p = 0 \\ \mathbf{n}_p = \mathbf{n}_{OG} \times \mathbf{n}_A \end{cases} \quad (1)$$

where \mathbf{p} is a point on the slicing plane, \mathbf{n}_p is the normal of the slicing plane, $\mathbf{n}_A = (\mathbf{p}_g - \mathbf{p}_o) / \|\mathbf{p}_g - \mathbf{p}_o\|$, and $\mathbf{n}_{OG} = (\mathbf{n}_o + \mathbf{n}_g) / \|\mathbf{n}_o + \mathbf{n}_g\|$. To calculate conveniently the intersection of the slicing plane and an edge of the triangular mesh, the line equation of the edge is given in the form of vector $\mathbf{p} = \mathbf{p}_s + t(\mathbf{p}_e - \mathbf{p}_s)$ where \mathbf{p}_s and \mathbf{p}_e are the start point and the end point of the edge, respectively, then substituting it for \mathbf{p} in Eq. (1) can obtain the intersection of the slicing plane and the edge:

$$\mathbf{p} = \frac{(\mathbf{p}_o - \mathbf{p}_s) \cdot \mathbf{n}_p}{(\mathbf{p}_e - \mathbf{p}_s) \cdot \mathbf{n}_p} (\mathbf{p}_e - \mathbf{p}_s) + \mathbf{p}_s. \quad (2)$$

2.2.2. Tracing bridge point

The process of calculating the bridge can start from arbitrary one of its two end points, the subsequent points of the bridge can be efficiently calculated using Eq. (2). To speed up this process, the topology of the mesh surface is combined in the calculation of the bridge, so that not all edges are required to intersect the slicing plane. In this process, two cases for the intersection need to be addressed.

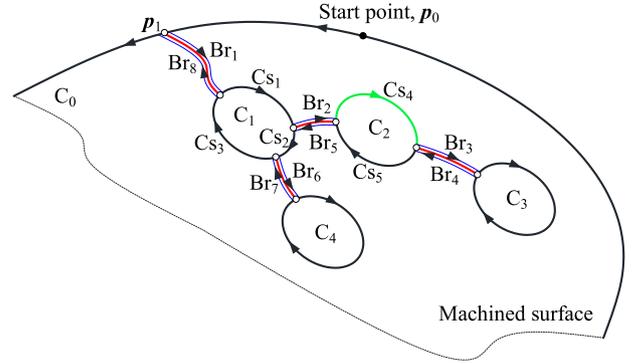


Fig. 3. Bridges, boundary curve segments divided by bridges and boundary curves connection.

Case 1. If the calculated point of bridge is just a vertex of mesh, for example, \mathbf{p}_o and \mathbf{p}_v in Fig. 2, then the next point of bridge needs to be calculated by Eq. (2) from the edge set of the 1-ring neighbor triangles of this vertex.

Case 2. If the calculated point of bridge is on an edge of mesh (exclude the end points of this edge), for example, \mathbf{p}_e in Fig. 2, the next point of bridge must be on the other edges of the triangle which shares this edge and does not contain the previous bridge point.

The obtained intersection points are stored sequentially in OBridge.Pset. When one boundary curve is bridged to another one, two bridges with opposite directions are created, one bridge in and the other out, which are then used to connect the boundary curves. For a bridge, its opposite bridge can be easily created by exchanging the values of BcIndex1 and BcIndex2 and reversing the storage order of the bridge points in Pset.

2.3. Connecting boundary curves

From Fig. 3, it is seen that the boundaries are cut by the bridges into segments and each segment is enclosed by a pair of bridge with opposite directions, the inward bridge and the outward bridge. These segments and bridges will be used to connect into a PS curve, namely the boundary of the new genus-0 mesh surface. Before connecting, the data structure of the segment is first defined.

Definition 2. OSegment {Bcurve, Epoint1, Epoint2}.

In OSegment, Bcurve is the index of the boundary curve to which the segment belongs, Epoint1 and Epoint2 is the start point and the end point of this segment along the direction of the boundary curve. Using the three parameters, the inward bridge and the outward bridge of the segment can be efficiently identified from the bridge set.

The boundary curve connecting process can start from any point in the outer boundary and then proceed along the direction of the current boundary. If an end point of a bridge is reached, the PS curve turns into its child node connected by this bridge and continues the traverse along the boundary of this node. If the node being traversed has also a child node, its child node will be first traversed. According to this rule, i.e., the child node is first traversed, all boundary curves can be connected rapidly as a whole to form a PS curve without any traverse breaks. This PS curve becomes the only outer boundary of the new mesh surface with genus-0. For example, the outer and interior boundaries in Fig. 3 can be connected into a PS curve by using this method, $\mathbf{p}_0 \rightarrow \mathbf{p}_1 \rightarrow \text{Br}_1 \rightarrow \text{Cs}_1 \rightarrow \text{Br}_2 \rightarrow \text{Cs}_4 \rightarrow \text{Br}_3 \rightarrow \text{C}_3 \rightarrow \text{Br}_4 \rightarrow \text{Cs}_5 \rightarrow \text{Br}_5 \rightarrow \text{Cs}_2 \rightarrow \text{Br}_6 \rightarrow \text{C}_4 \rightarrow \text{Br}_7 \rightarrow \text{Cs}_3 \rightarrow \text{Br}_8 \rightarrow \mathbf{p}_1 \rightarrow \mathbf{p}_0$.

3. Boundary-conforming mapping

The purpose of introducing boundary-conforming mapping is to transform the genus-0 mesh surface obtained in Section 2 into a rectangle region so that the operation of generating tool path can be performed conveniently on the rectangle regardless of the surface boundary shape.

3.1. Mathematical model

Here, for integrality of the method, the essential theory of boundary-conforming mapping is first reviewed briefly. Mathematically, given a physical surface Ω and a planar region Ω' , the unique mapping $\psi: \Omega \rightarrow \Omega'$ that satisfies the following Laplace equation [36]

$$\Delta\psi = 0 \quad (3)$$

subject to the Dirichlet boundary condition $b: \partial\Omega \rightarrow \partial\Omega'$, is called as harmonic mapping, also a bijective boundary mapping, where Δ is Laplacian and ψ is a mapping over the physical surface Ω to the planar region Ω' .

For the discrete triangular mesh, boundary-conforming mapping can be realized by constructing a piece-wise linear approximation. Assume that the mesh surface composes of many elastic, triangular rubber sheets sewn together along their edges. In the process of mapping it into a rectangle, the produced deforming energy, also called the harmonic energy [36], can be calculated by

$$\begin{aligned} \Phi &= \sum_{\{\mathbf{p}_i, \mathbf{p}_j\} \in E(\Omega)} w_{i,j} \|\psi(\mathbf{p}_i) - \psi(\mathbf{p}_j)\|^2 \\ &= \sum_{\{\mathbf{q}_i, \mathbf{q}_j\} \in E(\Omega')} w_{i,j} \|\mathbf{q}_i - \mathbf{q}_j\|^2 \end{aligned} \quad (4)$$

where $E(\Omega)$, $E(\Omega')$ denotes the set of edges of Ω and Ω' , respectively, \mathbf{q}_i is the corresponding point of \mathbf{p}_i in Ω' , and $w_{i,j}$ serves as elastic coefficient of the edge $\{\mathbf{p}_i, \mathbf{p}_j\}$. Now, boundary-conforming mapping becomes to minimize Eq. (4) by arranging the positions of all vertices of Ω in Ω' .

3.2. Solving method

In this paper, boundary-conforming mapping is realized by two steps, namely first specifying the planar region boundary and then arranging the positions of the interior vertices of Ω in Ω' . In the first step, in order to map the new boundary obtained in Section 2 into a rectangle, it is first divided into four segments which are consistent with the four sides of the rectangle, as shown in Fig. 4. Assume that the lengths of the four boundary segments are L_1, L_2, L_3 and L_4 , then the lower left point and the upper right point, i.e. point 1' and point 3' in Fig. 4, are set

$$\text{point } 1' \Rightarrow (0, 0) \quad \text{and} \quad \text{point } 3' \Rightarrow \left(\frac{L_1 + L_3}{2}, \frac{L_2 + L_4}{2} \right). \quad (5)$$

Then, the boundary points of each segment can be mapped onto the rectangle boundary according to the chord length parameterization. Taking segment 1–2 in Fig. 4 as an example, the boundary points on it can be mapped onto the bottom side of the rectangle by

$$b: \mathbf{p}_i \rightarrow \left(\tau_i \cdot \frac{L_1 + L_3}{2}, 0 \right), \quad \mathbf{p}_i \in \text{the first segment of } \partial\Omega \quad (6)$$

where τ_i is the normalized chord parameter with $\tau_0 = 0$ and for the remaining points on the first segment of $\partial\Omega$,

$$\tau_i = \frac{\sum_{j=1}^i \|\mathbf{p}_j - \mathbf{p}_{j-1}\|}{L_1} \quad (7)$$

and the remaining three segments of $\partial\Omega$ can be processed with the same way.

Now, the rest of the task, i.e. the second step, is to distribute the positions of the interior vertices of Ω in Ω' to minimize Eq. (4) by solving the following linear system.

$$\frac{\partial\Phi}{\partial\mathbf{q}} = 0 \Rightarrow \begin{bmatrix} A_{r \times r}^1 & A_{r \times (n-r)}^2 \\ A_{(n-r) \times n}^3 \end{bmatrix} \begin{bmatrix} X_{r \times 2}^B \\ X_{(n-r) \times 2}^I \end{bmatrix} = 0 \quad (8)$$

where A^1, A^2, A^3 are the coefficient matrices of $\partial E/\partial\mathbf{q} = 0$, and $X^B = [\mathbf{q}_1, \dots, \mathbf{q}_r]^T$, $X^I = [\mathbf{q}_{r+1}, \dots, \mathbf{q}_n]^T$. Since the boundary mapping, $b: \partial\Omega \rightarrow \partial\Omega'$, is known, the rows associated with the boundary points can be moved to the right side of Eq. (8), and then it can be rewritten as

$$\begin{aligned} \mathbf{q}_i \sum_{\mathbf{q}_j \in N(\mathbf{q}_i)} w_{i,j} - \sum_{\mathbf{q}_j \in S(\mathbf{q}_i)} (w_{i,j} \mathbf{q}_j) &= \sum_{\mathbf{q}_j \in B(\mathbf{q}_i)} (w_{i,j} \mathbf{q}_j), \\ i &= r+1, \dots, n \end{aligned} \quad (9)$$

where $\mathbf{q}_i = [u_i, v_i]^T$, $N(\mathbf{q}_i)$ is the 1-ring neighbor vertices of \mathbf{q}_i , $S(\mathbf{q}_i)$ is the set of vertex in $N(\mathbf{q}_i)$ except for the vertices in $\partial\Omega'$, and $B(\mathbf{q}_i)$ the set of vertex simultaneously in both $N(\mathbf{q}_i)$ and $\partial\Omega'$. Eq. (9) is solved for u and v coordinates of the unknown interior vertices in Ω' , respectively.

4. Guide line planning

4.1. Basic idea of designing guide line

Generating tool paths on the nonzero genus mesh surface is inspired by the cylindrical helix and cylindrical circle. As shown in Fig. 5, if cutting a cylinder along one of its rectilinear generatrix, the helix becomes a set of parallel straight lines which keep a constant angle with the u -axis on the rectangle and the cylindrical circles become a set of u -parameter lines. Based on this observation, comparing Fig. 4 with Fig. 5, it is seen that if the machined surface's outer boundary and interior boundary, shown in yellow and green respectively in Fig. 4, are viewed as the bottom and top circles of the cylinder respectively in Fig. 5, then the cylindrical circle and cylindrical helix can be extended to the machined surface with nonzero genus to become the contour-parallel tool path and the spiral tool path. Their guide lines on this rectangle are respectively u -parameter lines and a set of parallel lines which keep a constant angle with the u -axis.

4.2. Guide line and its parameter increment

According to Section 4.1, the guide lines for contour-parallel tool path and spiral tool path can be presented by the following equations. Iso-parameter guide lines:

$$v_i = v_i^s = i\Delta\delta, \quad v_i^s \leq \frac{L_2 + L_4}{2} \quad (10)$$

and spiral guide lines:

$$\begin{cases} v_i^s = i\Delta\delta, & v_i^s \leq \frac{L_2 + L_4}{2} \\ v_i = v_i^s + \frac{2\Delta\delta}{L_1 + L_3} u, & 0 \leq u \leq \frac{L_1 + L_3}{2} \end{cases} \quad (11)$$

where $\Delta\delta$ is the parameter increment for the guide line and v_i^s is the start point of the i th guide line at $u = 0$. For the convenience of tool path calculation, $\Delta\delta$ is usually taken as a constant value. But, due to the nonlinearity of the mapping between the surface and the planar region, a fixed parameter increment does not necessarily correspond to a constant path interval as well as the scallop height. To generate a desired tool path, the parameter increment

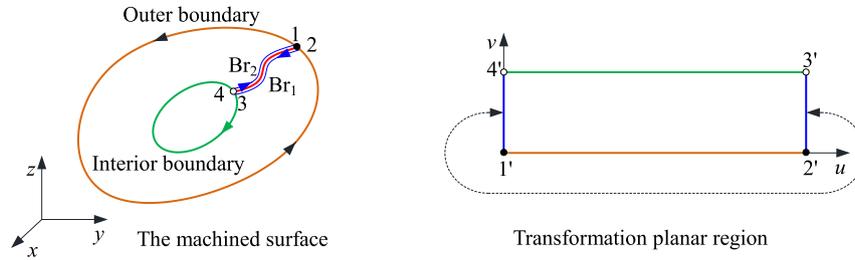


Fig. 4. Boundary-conforming mapping between the machined surface and the planar region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

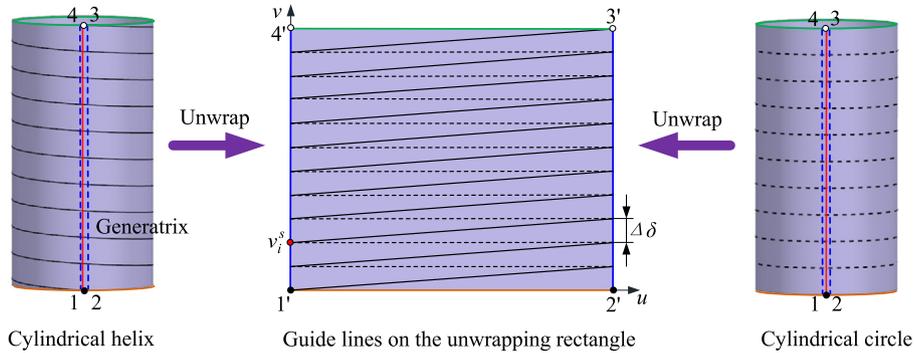


Fig. 5. The basic idea of generating tool paths on the nonzero genus surface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for the guide line has to be strictly limited so that the scallop height between two adjacent paths does not exceed the machining requirement h_s . According to Appendix, the parameter increment at CC point is calculated by

$$\begin{cases} \Delta u = \frac{L_w \Delta v_f}{\|\mathbf{r}_u\|^2 \Delta u_b \Delta v_f - \|\mathbf{r}_u\|^2 \Delta v_b \Delta u_f} \\ \Delta v = \frac{-L_w \Delta u_f}{\|\mathbf{r}_v\|^2 \Delta u_b \Delta v_f - \|\mathbf{r}_v\|^2 \Delta v_b \Delta u_f} \end{cases} \quad (12)$$

The final parameter increment, $\Delta\delta$, for the guide line is selected as the minimum parameter increment:

$$\Delta\delta = \min \{ \Delta v_1, \dots, \Delta v_n \} \quad (13)$$

which is then used to plan the guide lines in the rectangle according to Eq. (10) or (11).

5. CC path generation

In most CAD/CAM systems, the approximation error of the triangular mesh to the nominal surface can be specified by the user such that a desirable triangulation to be adequate for NC machining is obtained conveniently. If the approximation error is equal to or less than an allowable chordal error for the step length, the interpolation tolerance of the CC path can be controlled reasonably. In this scenario, only the intersections $\mathbf{q}_i(u_i, v_i)$ of the guide lines and the planar mesh edges are selected to calculate the CC points \mathbf{p}_i on the nominal surface by

$$\mathbf{p}_i = \psi^{-1} : (\Omega' \rightarrow \Omega) \mathbf{q}_i \quad (14)$$

where $\psi^{-1} : \Omega' \rightarrow \Omega$ is the inverse mapping of ψ . Using this way, the accuracy and smoothness of the machined surface are subjected to be compromised to some extent since CC points are not necessarily on the nominal surface although they are exactly on the edges of the mesh surface. To avoid this problem, a method calculating CC points is proposed, which requires reconstructing

the curve related to the edge instead of using the edge itself. This cannot only approximate better the local nominal surface, which can reduce the machining error, but also increase the flexibility to add easily the smoothing constraint.

5.1. Calculating CC point

For an edge $\{\mathbf{p}_i, \mathbf{p}_{i+1}\}$ of the triangular mesh, its approximated curve can be described as a cubic Bézier curve

$$\mathbf{r}(t) = \sum_{j=0}^3 \mathbf{v}_j B_{j,3}(t), \quad 0 \leq t \leq 1 \quad (15)$$

where $B_{j,3}(t)$ is a cubic Bernstein polynomial. If given the normal vectors associated with two end points of this edge, \mathbf{n}_i and \mathbf{n}_{i+1} , the point normal interpolation method [37] can be used to fit this edge curve whose control vertices $\{\mathbf{v}_j\}$ are as follows

$$\begin{cases} \mathbf{v}_0 = \mathbf{p}_i \\ \mathbf{v}_1 = \frac{2\mathbf{p}_i + \mathbf{p}_{i+1} - 2\eta\mathbf{n}_i - \mu\mathbf{n}'_{i+1}}{3} \\ \mathbf{v}_2 = \frac{2\mathbf{p}_{i+1} + \mathbf{p}_i - 2\mu\mathbf{n}'_{i+1} - \eta\mathbf{n}_i}{3} \\ \mathbf{v}_3 = \mathbf{p}_{i+1} \end{cases} \quad \text{and} \quad (16)$$

$$\begin{cases} \eta = -\frac{2\mathbf{b} \cdot \mathbf{n}_i + \mathbf{a} \mathbf{b} \cdot \mathbf{n}'_{i+1}}{4 - a^2} \\ \mu = -\frac{2\mathbf{b} \cdot \mathbf{n}'_{i+1} + \mathbf{a} \mathbf{b} \cdot \mathbf{n}_i}{4 - a^2} \\ \mathbf{a} = \mathbf{n}_i \cdot \mathbf{n}'_{i+1} \\ \mathbf{b} = \mathbf{p}_{i+1} - \mathbf{p}_i \end{cases}$$

where \mathbf{n}'_{i+1} is the unit vector of the projection of \mathbf{n}_{i+1} on the normal section, as shown in Fig. 6. Assume the intersection of the guide line and edge $\{\mathbf{q}_i, \mathbf{q}_{i+1}\}$ of the planar mesh is (u_0, v_0) , then the

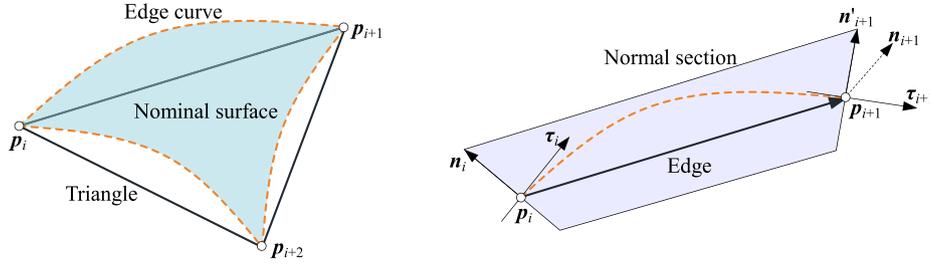


Fig. 6. Edge curve construction using point normal interpolation.

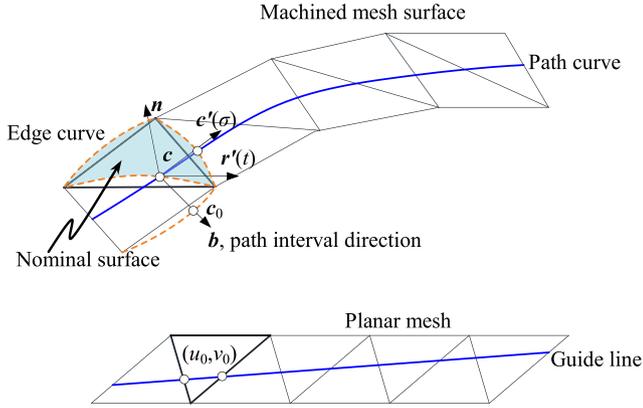


Fig. 7. CC point on the machined surface.

parameter value t_0 of CC point in this edge Bézier curve can be calculated by

$$t_0 = \frac{\sqrt{(u_i - u_0)^2 + (v_i - v_0)^2}}{\sqrt{(u_{i+1} - u_i)^2 + (v_{i+1} - v_i)^2}}. \quad (17)$$

Submitting t_0 into Eq. (15) can obtain the CC point on the nominal surface.

5.2. Calculating normal of CC point

After obtaining all CC points, the discrete CC path can be replaced with a twice continuously differentiable spline curve that is fit to the discrete path. It not only reduces the NC file size, but also smoothens the discrete CC path. By assigning a parameter value for each CC point using a quasi-arc length parameterization, the CC points can be fitted into a cubic B-spline curve.

$$\mathbf{c}(\sigma) = \sum_{s=0}^n \mathbf{d}_s N_{s,3}(\sigma), \quad 0 \leq \sigma \leq 1 \quad (18)$$

where $\{\mathbf{d}_s\}$ are the control points of B-spline curve and $N_{s,3}(\sigma)$ is the normalized B-spline basis. From Fig. 7, it is seen that the edge curve and the path curve both pass through the obtained CC point \mathbf{c} . According to this condition, the normal at the CC point can be determined by

$$\mathbf{n} = \frac{\mathbf{r}'(t) \times \mathbf{c}'(\sigma)}{\|\mathbf{r}'(t) \times \mathbf{c}'(\sigma)\|}. \quad (19)$$

5.3. Calculating path interval

Path interval, which corresponds to the parameter increment in the rectangle, is defined as the distance between two adjacent CC points and is expressed as

$$L_w = \sqrt{\frac{8h_s r \rho}{\rho + r}}, \quad h_s \ll \rho \quad (20)$$

where h_s is the scallop height, r is the cutter radius for a ball nose cutter or the effective cutting radius for a flatend cutters, and ρ is the radius of curvature of the machined surface in the plane perpendicular to feed direction, with a positive value for a convex surface and a negative value for a concave surface. From Fig. 7, it is seen that on the right-hand side of the CC path, the normal plane, which comprised of the path interval direction \mathbf{b} and the normal \mathbf{n} at the CC point, must intersect one of the other two edges of the triangle on which one edge the CC point is located. Assume that the intersection point of the normal plane and the edge curve is \mathbf{c}_0 , the normal curvature along the $\mathbf{c}_0 - \mathbf{c}$ direction can be approximated by [38]

$$k = \frac{1}{\rho} = \frac{2(\mathbf{c}(\sigma) - \mathbf{c}_0) \cdot \mathbf{n}}{\|\mathbf{c}(\sigma) - \mathbf{c}_0\|^2}. \quad (21)$$

Submitting ρ into Eq. (20) can obtain the path interval L_w .

6. Tool orientation optimization

Tool orientation optimization not only requires guaranteeing interference-free in 5-axis machining, but also must be able to minimize the dynamic loading on the rotary axes of machine tool and avoid the drastic change in the tool orientation between two neighboring CC points to improve the kinematic and dynamic performance of 5-axis NC machine tool. In the following context, a simple and efficient method is investigated to optimize the tool orientations along the tool path.

6.1. Procedures for tool orientation optimization

As shown in the dashed block in Fig. 8, the proposed method of optimizing tool orientation consists mainly of three steps. The first step is to assign the feasible tool orientations to the critical CC points. Generally, the critical points should include the first and the end points of the tool path, the points entering and leaving the interference area, and a point in the interference area, as shown in Fig. 9. Before processing this step, the feasible space of the tool orientation has to be constructed in advance for 5-axis machining without interference, which can be completed by some famous interference avoidance algorithms described in [25,27]. The second step which is also the emphasis of discussion in this section is to calculate the smooth tool orientations at other CC points from the assigned tool orientations by the proposed algorithm of optimizing tool orientation. Next, the third step is to check if the optimized tool orientation at the noncritical CC point is in its feasible space or not. If interferences appear, the tool orientation at the CC point needs to be reassigned from its feasible space, and back to the second step to continue smoothing the tool orientation. The above procedures are performed repeatedly until all tool orientations pass the interference check. From the above procedures, it is seen that our method involves mainly interference check and tool orientation optimization. Since the former has been studied extensively and many effective algorithms which are easy

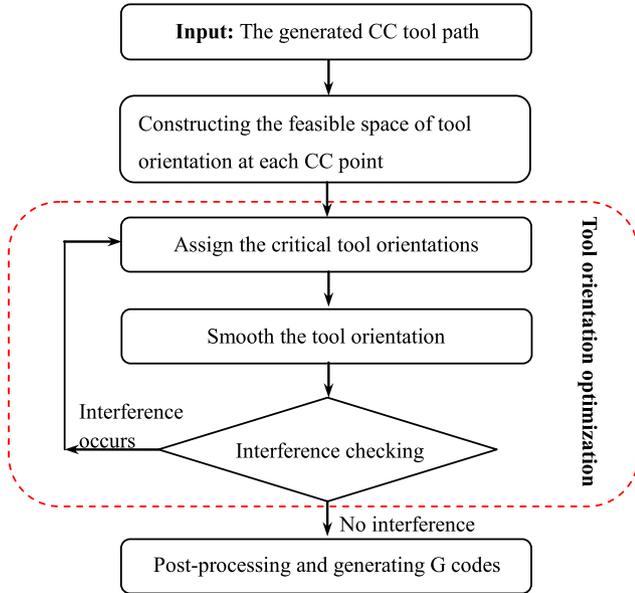


Fig. 8. The procedures for tool orientation optimization.

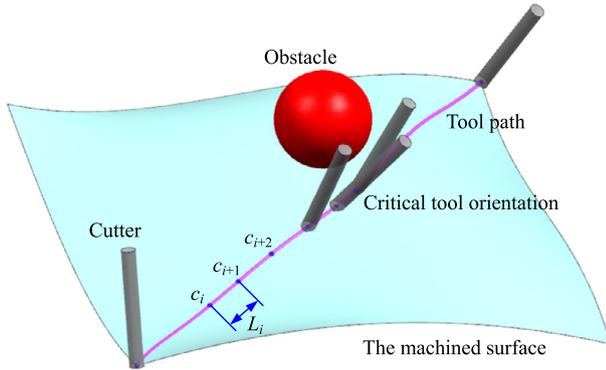


Fig. 9. Critical tool orientations.

to check the interference have been proposed, it is not included in this paper, the discussion is focused on how to optimize the tool orientations at these general CC points according to those assigned tool orientations.

6.2. Inverse kinematics of 5-axis machine tool

Different from the traditional methods which optimize the tool orientation in PCS, the proposed method works directly on the rotary axis coordinate in MCS. To do this, the assigned tool orientation at the critical CC point needs to be transformed into the rotary axis order in MCS by the IK of 5-axis machine tool. Here we focus exclusively on a 5-axis machine tool of dual rotary heads type which IK is shown in Fig. 10.

In MCS, tool orientation can be represented by

$$\begin{aligned} \mathbf{A}^{(m)} &= [a_x^{(m)}, a_y^{(m)}, a_z^{(m)}]^T \\ &= [\sin \Phi^C \sin \Phi^A, -\cos \Phi^C \sin \Phi^A, \cos \Phi^A]^T \end{aligned} \quad (22)$$

where Φ^A and Φ^C are respectively the angles of A and C axis. In PCS, tool orientation is expressed as

$$\begin{aligned} \mathbf{A}^{(w)} &= [a_x^{(w)}, a_y^{(w)}, a_z^{(w)}]^T \\ &= \mathbf{T}(s) \sin \alpha(s) \cos \beta(s) \\ &\quad + \mathbf{B}(s) \sin \alpha(s) \sin \beta(s) + \mathbf{N}(s) \cos \alpha(s). \end{aligned} \quad (23)$$

In Eq. (23), $\alpha(s)$ and $\beta(s)$ are the tilt angle and the yaw angle of the tool in LCS, respectively, $\mathbf{T}(s)$ is the feed direction, $\mathbf{N}(s)$ is the normal of the surface at CC point $\mathbf{O}^{(L)}$ and $\mathbf{B}(s)$ is the cross product of $\mathbf{T}(s)$ and $\mathbf{N}(s)$. For the dual rotary heads type machine tool, MCS and PCS have the same initial direction. According to this condition, we have

$$\begin{aligned} &[\sin \Phi^C \sin \Phi^A, -\cos \Phi^C \sin \Phi^A, \cos \Phi^A]^T \\ &= [a_x^{(w)}, a_y^{(w)}, a_z^{(w)}]^T. \end{aligned} \quad (24)$$

Regardless of the singularity of the solutions, of course, which is also an important topic for NC machining; the inverse solutions of Eq. (24) can be calculated by

$$\begin{cases} \Phi^A = \arccos(a_z^{(w)}) \\ \Phi^C = \text{atan2}(a_x^{(w)}, a_y^{(w)}). \end{cases} \quad (25)$$

6.3. Tool orientation optimization in MCS

Generally, it is difficult to optimize simultaneously the two rotary axes. But, if A-axis and C-axis can be considered individually in the optimization, this problem can be simplified. Based on this idea, the models of minimizing the angular velocities and angular accelerations of the rotary axes will be discussed respectively in detail. Here, it is assumed that the cutter moves along the CC path with a constant feedrate f such that the arc length L between two neighboring CC points is proportional to the machining time t . For the convenience of the calculation, the numerical difference is adopted to approximate the angular velocity and angular acceleration instead of using the complicated first and second derivatives of Eq. (25).

6.3.1. Model of minimizing the angular velocity

In the section, only the angular velocities of two rotary axes are considered and A-axis and C-axis are processed individually. For A-axis, its angular velocity can be approximated by the numerical difference as

$$\omega_A = \frac{\Phi_{i+1}^A - \Phi_i^A}{L_i} f. \quad (26)$$

To improve the kinematics and dynamics performance of machine tool, it is desirable to decrease as much as possible the change of A-axis rotary angle between two neighboring points. Thus, the optimization objective function minimizing the changes of Φ^A can be written as

$$\Theta = \sum_i \left(\frac{\Phi_{i+1}^A - \Phi_i^A}{L_i} f \right)^2. \quad (27)$$

The condition that the least-square (LS) problem has the extremes is $\partial \Theta / \partial \Phi_i^A = 0$. Omitting the specific derivation, $\partial \Theta / \partial \Phi_i^A = 0$ can be restated as

$$\frac{\partial \Theta}{\partial \Phi_i^A} = L_{i-1}^2 \Phi_{i+1}^A - (L_i^2 + L_{i-1}^2) \Phi_i^A + L_i^2 \Phi_{i-1}^A = 0 \quad (28)$$

Eq. (28) has n unknowns, $\Phi_i^A, i = 1, 2, \dots, n$, but there are only $n - 2$ equations. For the optimization problem having a non-trivial solution, some of Φ_i^A must be set to a priori values. Once the tool orientations at m critical CC points are pinned, Eq. (28) can be rewritten in form of matrix as

$$\mathbf{M}^{A,\omega} \Phi_f^{A,\omega} = \mathbf{B}^{A,\omega} \quad (29)$$

where $\mathbf{M}^{A,\omega}$ is a $(n - m) \times (n - m)$ coefficient matrix, $\mathbf{B}^{A,\omega}$ and $\Phi_f^{A,\omega}$ are vectors consisting respectively of m pinned Φ_i^A and $n - m$

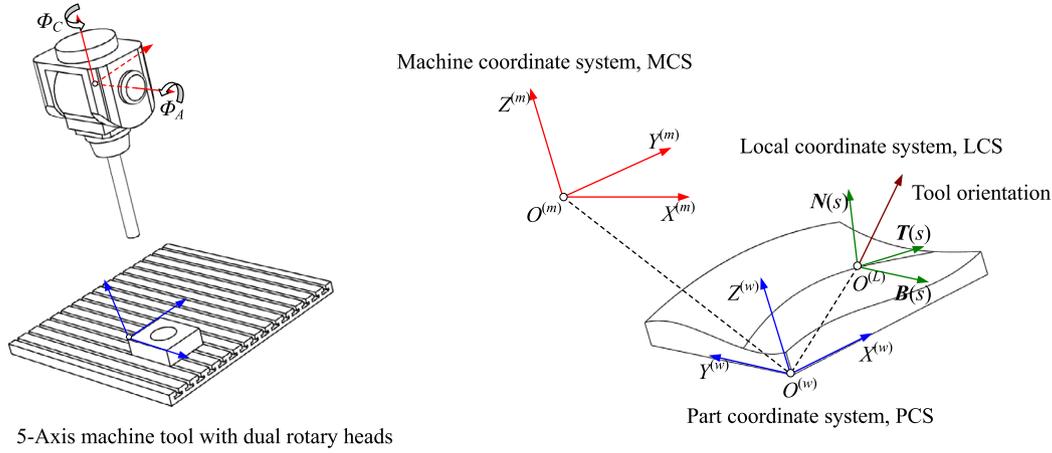


Fig. 10. 5-axis machine tool with dual rotary heads and kinematics transformation.

unknown, Φ_i^A . In the same way, for C-axis, we can also derive a similar matrix equation as follows

$$\mathbf{M}^{C,\omega} \Phi_f^{C,\omega} = \mathbf{B}^{C,\omega}. \quad (30)$$

The solution of the minimization problem (Eqs. (29) and (30)) can be given by the following unified formula

$$\Phi_f = \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1} (\mathbf{H}^T \mathbf{H}) \mathbf{H}^T \mathbf{B} \quad (31)$$

where \mathbf{M} represents $\mathbf{M}^{A,\omega}$ or $\mathbf{M}^{C,\omega}$ and $\mathbf{M} = \mathbf{G}\mathbf{H}$ where \mathbf{G} is a $(n-m) \times s$ matrix and \mathbf{H} is a $s \times (n-m)$ matrix and they both have rank s .

6.3.2. Model of minimizing the angular acceleration

The angular acceleration of A-axis can be approximated by the numerical difference as

$$a_A = 2f^2 \left[\left(\frac{\Phi_{i+1}^A - \Phi_i^A}{Q_{i-1}L_i} \right) - \left(\frac{\Phi_i^A - \Phi_{i-1}^A}{Q_{i-1}L_{i-1}} \right) \right] \quad (32)$$

where $Q_{i-1} = L_i + L_{i-1}$. Here, it has been assumed that the dynamics of A-axis is described only by the angular acceleration, then for A-axis the least-square objective function, which minimizes the change of the angular acceleration of A-axis, can be written as

$$\Omega = \sum_i \left(2f^2 \left(\left(\frac{\Phi_{i+1}^A - \Phi_i^A}{Q_{i-1}L_i} \right) - \left(\frac{\Phi_i^A - \Phi_{i-1}^A}{Q_{i-1}L_{i-1}} \right) \right) \right)^2. \quad (33)$$

The condition, $\partial\Omega/\partial\Phi_i^A = 0$, that Eq. (33) can achieve the extremes, can be derived as

$$\frac{\partial\Omega}{\partial\Phi_i^A} = K_{i-2}\Phi_{i-2}^A + K_{i-1}\Phi_{i-1}^A + K_i\Phi_i^A + K_{i+1}\Phi_{i+1}^A + K_{i+2}\Phi_{i+2}^A \quad (34)$$

where

$$\begin{cases} K_{i-2} = A \\ K_{i-1} = -A - B - C \\ K_i = B + C + D + E \\ K_{i+1} = -D - E - F \\ K_{i+2} = F \end{cases} \quad \text{and} \quad \begin{cases} A = Q_{i-2}^2 Q_{i-1} L_{i-2} L_{i-1} L_i \\ B = Q_{i-2}^2 Q_{i-1} L_{i-2} L_{i-1}^2 L_{i+1} \\ C = Q_{i-2}^2 Q_i^2 L_{i-2} L_{i-1} L_{i+1} \\ D = Q_{i-2}^2 Q_i^2 L_{i-2} L_i L_{i+1} \\ E = Q_{i-1} Q_i^2 L_{i-2} L_i^2 L_{i+1} \\ F = Q_{i-1} Q_i^2 L_{i-1} L_i^2 L_{i+1}. \end{cases}$$

Similar to the processing of angular velocity, Eq. (34) can be also restated in form of matrix as

$$\mathbf{M}^{A,\alpha} \Phi_f^{A,\alpha} = \mathbf{B}^{A,\alpha} \quad (35)$$

where $\mathbf{M}^{A,\alpha}$ is a $(n-m) \times (n-m)$ coefficient matrix, $\mathbf{B}^{A,\alpha}$ and $\Phi_f^{A,\alpha}$ are vectors consisting respectively of m pinned Φ_i^A and $n-m$ unknown Φ_i^A . And for C-axis, we also have

$$\mathbf{M}^{C,\alpha} \Phi_f^{C,\alpha} = \mathbf{B}^{C,\alpha}. \quad (36)$$

Eqs. (35) and (36) can be solved using the same way as Eq. (31).

7. Experimental results

The methods involved in the paper have been coded in C++ language and implemented on a PC with an Intel 3.4 GHz and 8.0G physical memory. In the following, several examples will be presented and discussed in detail. All examples are modeled in Unigraphics NX 7.5 and the surfaces for test are then saved by STL format. When using our codes to read the STL files, the topological relations between the elements, such as vertex, edge and facet, are also constructed and the redundant vertices are removed for the convenience of the following processing.

7.1. Feasibility of the proposed tool path method

Three tested parts are shown respectively in Figs. 11(a), 12(a) and 13(a), and the surfaces for test are shown in yellow and have one genus, two genera and three genera, respectively. According to the discussion of Section 2, regardless of the genus number, our method views them as whole, i.e. a new genus, by bridging the interior boundaries so that the surface to be machined is homeomorphic topologically to a cylindrical surface, the outer boundary is corresponding to the bottom circle of the cylinder and the inner boundary formed by bridging the interior boundaries to the top circle. According to Sections 4 and 5, the spiral tool path and the contour-parallel tool path can be easily generated. The generated spiral paths and contour-parallel paths are shown in Figs. 11–13. From these figures, it is seen that the generated paths have no sharp turning and can be conformed to the surface boundary, thus when the nonzero genus surface needs to be machined using a smooth and continuous tool path, the proposed method is feasible and applicable.

7.2. Machining strategy for model with discontinuities

In this example, the tool path planning is conducted on a compound surface with multi-patches. The test surfaces, shown in Fig. 14(a), are retrieved from a football model, and for the need of demonstration a hole is made on the right patch, thus becoming a genus-one surface. Different from the machined surfaces used

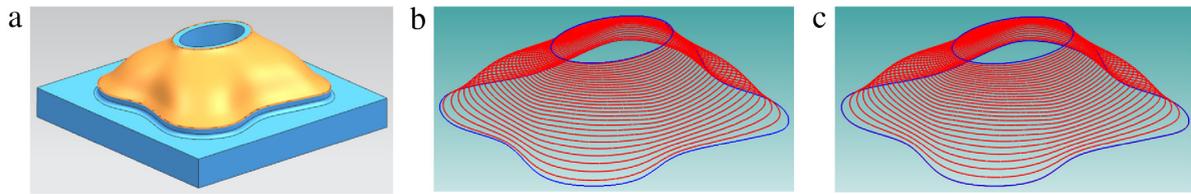


Fig. 11. Tool paths generated by the proposed method on the genus-1 surface. (a) The genus-1 surface; (b) Spiral tool paths; (c) Contour-parallel tool paths. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

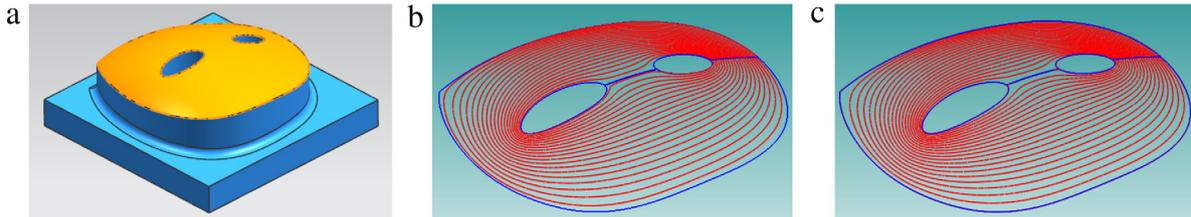


Fig. 12. Tool paths generated by the proposed method on the genus-2 surface. (a) The genus-2 surface; (b) Spiral tool paths; (c) Contour-parallel tool paths. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

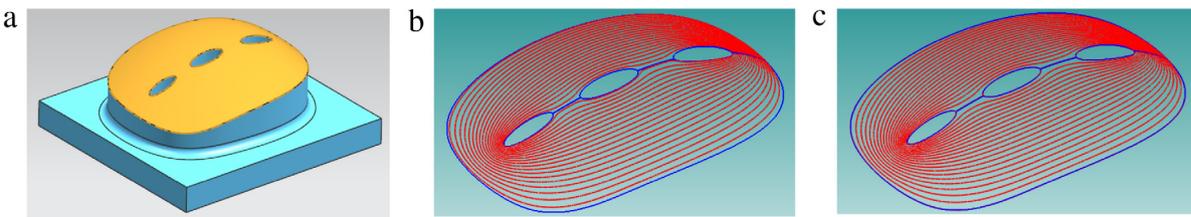


Fig. 13. Tool paths generated by the proposed method on the genus-3 surface. (a) The genus-3 surface; (b) Spiral tool paths; (c) Contour-parallel tool paths.

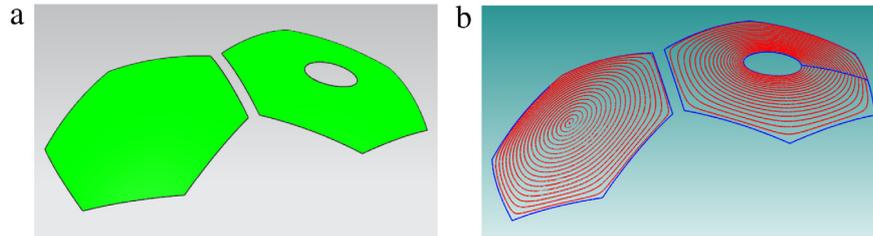


Fig. 14. The spiral tool path generation strategy on the model with discontinuities. (a) The model surfaces; (b) Spiral tool paths: the left is the spiral tool paths generated on the surface without genus by the method in Ref. [15] and the right is the spiral tool paths generated on the genus-1 surface with by the proposed method.

in the above examples, the discontinuities appear on the model surface, which is also often confronted for the complex model. In order to realize the high efficiency machining of such model, the surface patches of different types can be processed individually and adopt different machining strategy. Since the left patch that has none holes is a nonzero genus surface, it is suitable to be machined by the spiral tool path of Ref. [15], but the proposed method in this paper is more applicable to machine the right surface patch with genus-one. Using this machining strategy, the surfaces with discontinuities, shown in Fig. 14(a), can be machined efficiently using the spiral tool paths in Fig. 14(b).

7.3. Machining and comparisons of tool paths

In order to evaluate the benefits bought by our method, the comparisons and real machining experiments of three kinds of tool paths are carried out on the genus-2 surface in Fig. 15(a) and the real mouse surface in Fig. 16(a). In addition to our tool path, denoted by Proposed in Table 2, the other two kinds of tool paths are also representative, which are the contour-parallel tool path used widely in practice and the spiral tool path, similar to ours, generated by the famous UG/CAM software and are denoted by CP

path and SP path in Table 2, respectively. The parameters used in tool path generation and real machining are listed in Table 1.

The generated three kinds of tool paths on the genus-2 surface are shown respectively in Fig. 15(b), (c) and (d), and Fig. 16(b), (c) and (d) give the tool paths on the mouse surface. Tool paths are first compared by analyzing the characteristics closely associated with the kinematics performance of tool path, such as rapid traversal or interruption, lifting cutter, path linking and sharp corner. From Table 2, it is seen that the proposed tool path holds the best performance and nicely conforms to the criterion of the high performance tool path mentioned in [12,13], i.e., the least number of interruptions or rapid traversals and smooth continuous tool paths without sharp corners. This point is also observed in Figs. 15(d) and 16(d).

The actual kinematic performance of tool path, in this experiment, is chartered by the real feed rate and the loss of feed rate which are defined, respectively, as follows

$$f_{\text{actual}} = \frac{\text{Length of tool path}}{\text{Actual machining time}} \quad (37)$$

$$\Delta f_{\text{loss}} = \frac{f_{\text{actual}} - f_{\text{programmed}}}{f_{\text{programmed}}} \times 100 \quad (38)$$

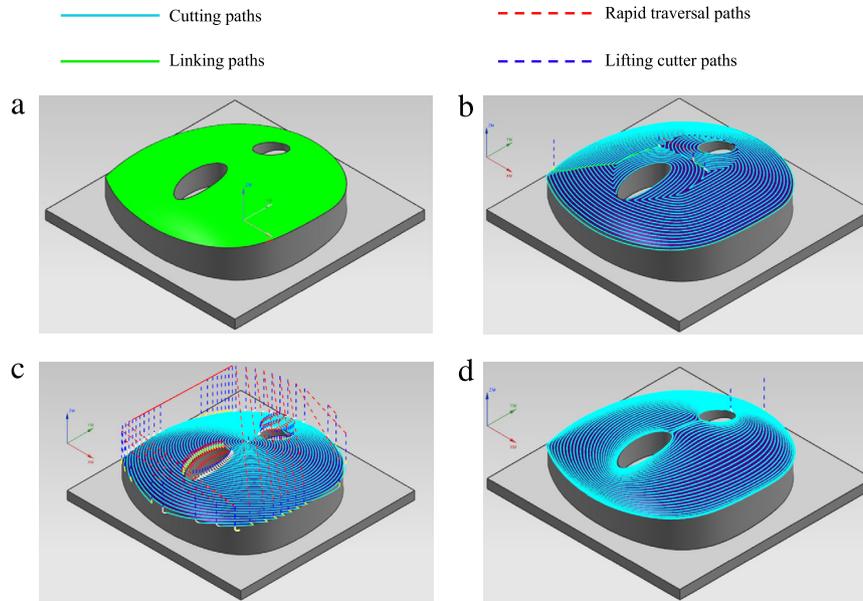


Fig. 15. Three kinds of tool paths on a genus-2 surface; (a) The genus-2 model surface for test. (b) Contour-parallel tool paths generated by UG/CAM; (c) Spiral tool paths generated by UG/CAM; and (d) Spiral tool paths generated by the proposed method in this paper.

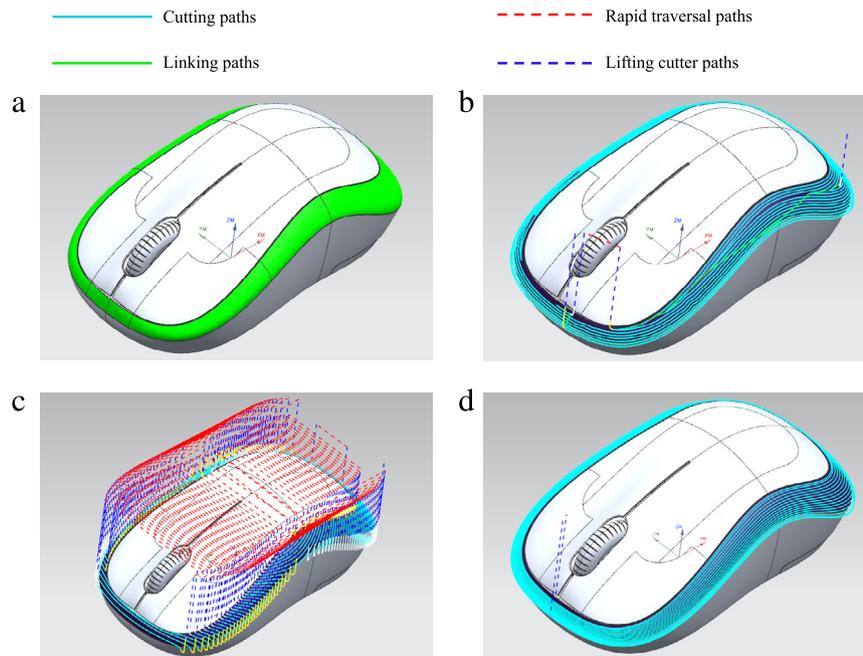


Fig. 16. Three kinds of tool paths on the mouse surfaces; (a) The surfaces of a mouse model for test; (b) Contour-parallel tool paths generated by UG/CAM; (c) Spiral tool paths generated by UG/CAM; and (d) Spiral tool paths generated by the proposed method in this paper.

Table 1

The parameters of tool path generation and real machining experiments.

Model surface	Cutter radius (mm)	Scallop height (mm)	Tolerance (mm)	Spindle speed (rpm)	Feed rate (mm/min)
Fig. 15(a)	R3	0.200	0.010	3000	300
Fig. 16(a)	R1	0.100	0.010	5000	250

where f_{actual} is the real feed rate, $f_{programmed}$ is the programmed one and Δf_{loss} is the loss of feed rate. It is well known that excessive path interruptions and sharp corners will cause frequent decelerations and accelerations when the cutter moves into and out of such area such that the actual feed rate cannot almost match the programmed one. This point is also confirmed by our experiments. From Table 2, it is seen that there are 94 and 24 sharp corners in CP paths on the genus-2 surface in Fig. 15(b) and

the mouse surface in Fig. 16(b), respectively. As expected, these sharp corners make the cutter difficult to achieve the specified speed and lead to -7.47% and -3.40% loss of programmed feed rate, respectively. In SP paths in Figs. 15(c) and 16(c), the path interruptions are dominant, but the paths entering the cutting area may be optimized by UG/CAM, they only lead to -3.27% and -7.04% loss of feed rate although the times of lifting cutter are up to 94 and 520. In contrast with CP paths and SP paths, the

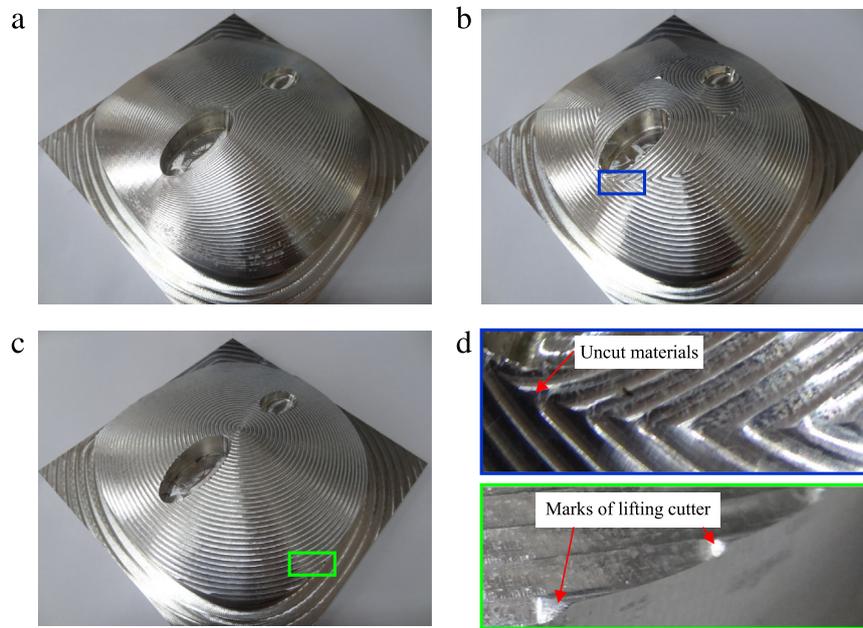


Fig. 17. Machining experiments of the genus-2 surface using three kinds of tool paths; (a) Machining of the proposed spiral tool paths; (b) Machining of contour-parallel tool paths generated by UG/CAM; (c) Machining of spiral tool paths generated by UG/CAM; and (d) The magnified views of the machining areas. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2

Comparisons of characteristics and machining results of three kinds of tool path.

Model surface	Tool path generation method	Number of rapid traversal	Number of lifting	Number of linking	Number of sharp corner	Have uncut material or not	Actual feed rate (mm/min)	Loss of feed rate
Fig. 15(a)	CP path	1	4	26	92	Y	277.6	−7.47%
	SP path	46	94	0	0	N	290.2	−3.27%
	Proposed	0	2	0	0	N	298.9	−0.37%
Fig. 16(a)	CP path	1	4	22	24	Y	241.5	−3.40%
	SP path	259	520	0	0	Y	232.4	−7.04%
	Proposed	0	2	0	0	N	249.1	−0.36%

proposed spiral tool paths not only have no sharp corners but also the changes of its curvature along the generated tool path are very smooth so that the cutter presents a more continuous, fast and smooth movement. This has been also confirmed by our experimental results, in Table 2, that the proposed paths maintain almost the programmed feed rate in the overall machining.

Sharp corners in CP paths not only decrease the actual feed rate, but also lead to the occurrence of uncut phenomenon, reducing the machined surface quality, when the path angle at the sharp corner reaches the uncut condition [9]. From Fig. 17(b) and (d) and Fig. 18(b) and (d), it is seen that the uncut materials occur at the sharp corner region enclosed by the blue line. Uncut is the inherent drawback of CP path and is difficult to be removed by itself. Under the case, the transition arcs or other smooth curves have to be added into CP paths to remove the uncut material and round the sharp corners. For SP paths in Fig. 15(c), though the paths cutting into and out of the machined area have already optimized by UG/CAM, the lifting cutter marks at these areas are still apparent, as shown in the blow local magnified view enclosed by green line in Fig. 17(d). Also, it is noted that, when using SP paths in Fig. 16(c) to machine the mouse surface, the marks of lifting cutter and the uncut materials remain on the machined surface at the same time, as shown in the blow local magnified view enclosed by green line in Fig. 18(d). These uncut materials on the machined mouse surface are resulted from a distinct disadvantage of UG/CAM's spiral tool path generation method. Since the method is to project the Archimedean spiral onto the designed surface to generate the spiral paths, when the slope of surface increase,

the distance between two adjacent paths will increase. In our experiments, the normal of the mouse surface is almost vertical to the projection direction at some machining area, thus the path interval between two adjacent paths at these regions are larger than the specified one. This results in the appearance of uncut materials. In contrast with CP path and SP path, when using the proposed spiral tool paths to cut the mouse surface, no uncut materials remain on the machined surface, which can be observed from Figs. 17(a) and 18(a). This also indicates the proposed method can gain a better machining quality.

7.4. Effectiveness of tool orientation optimization

This example is used to demonstrate the proposed method of optimizing tool orientation in Section 6. Fig. 19(a) shows the traditional tool orientations that only avoid the possible interferences occurring in 5-axis machining. It is seen, to avoid the collision between the obstacle and the cutter, the cutter appears an abrupt swing when being close to the obstacle, which will lead to sudden increase of angular velocities and angular accelerations for A-axis and C-axis. In our experiments, the cutter feed rate is set 500 mm/min. When the cutter closes to the interference area, the angular velocity and angular acceleration of A-axis suddenly increase to 1.48 rad/s and 7.95 rad/s², respectively, subsequently the drastic fluctuations appear until the cutter avoids the collision and enters completely the interference area. When the cutter leaves the interference area, the similar phenomenon appears again. For C-axis, there is also the same problem. These are fully

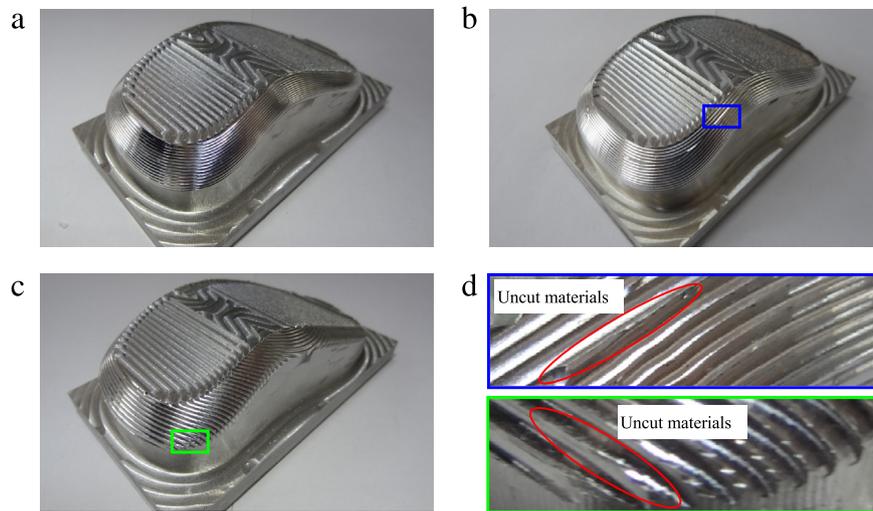


Fig. 18. Machining experiments of the mouse surface using three kinds of tool paths; (a) Machining of the proposed spiral tool paths; (b) Machining of contour-parallel tool paths generated by UG/CAM; (c) Machining of spiral tool paths generated by UG/CAM; and (d) The magnified views of the machining areas. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

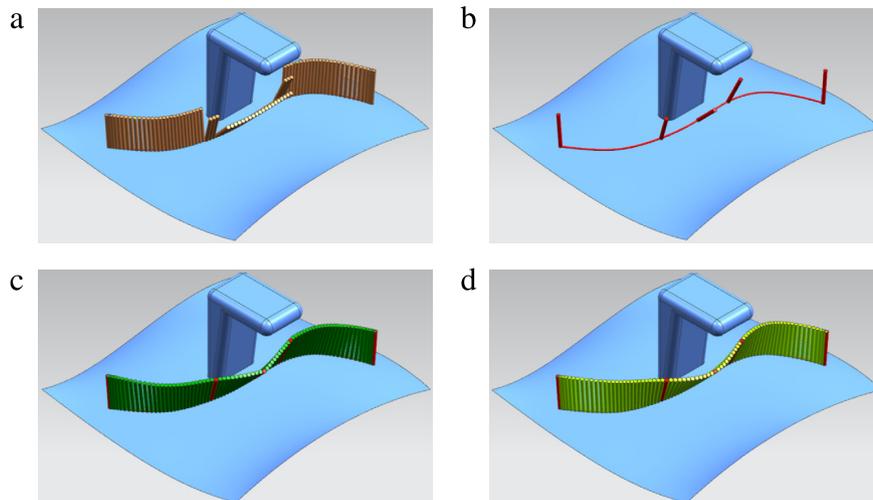


Fig. 19. Tool orientation optimization. (a) Tool orientation before optimization; (b) Critical tool orientations; (c) Result of tool orientation optimization minimizing the changes of angular velocity; (d) Result of tool orientation optimization minimizing the changes of angular acceleration.

reflected in Fig. 20. The sudden changes of the angular velocity and the angular acceleration will lead to a bad dynamic performance of machine tool and even damage possibly the components of machine tool.

To improve the dynamics of 5-axis machining, the proposed method in Section 6 is used to minimize the changes of the rotation angles of A-axis and C-axis. In our experiments, the tool orientations at 5 ($m = 5$) critical CC points, as shown in Fig. 19(b), are pinned in advance, at which the cutter has to be guaranteed to be interference-free. The tool orientations at other CC points are then determined by using the optimization models in Section 6.3. Fig. 19(c) shows the optimized tool orientations by using the method described in Section 6.3.1. It can be seen that the changes of tool orientation after optimization is more continuous than that before optimization. In spite of this, from Fig. 20, it is noted that only optimizing the angular velocity cannot still avoid some small fluctuations of the angular velocity or the angular acceleration. Tool orientation needs to be smoothed further by the method given in Section 6.3.2. Fig. 19(d) gives the optimization result of tool orientation of minimizing the changes of angular acceleration. Compared with Fig. 19(c), tool orientation in Fig. 19(d) becomes smoother and is no longer a

simple approximately linear interpolation. Fig. 20 also shows, after optimizing the angular acceleration, the angular velocity and angular acceleration of either A-axis or C-axis does not appear a sudden change, thus improving further the kinematic and dynamic performance of machine tool. In addition, it is worth noting, since the proposed method only involves solving a sparse linear system, the calculation of optimal tool orientation in each step is very efficient, and for this example, the running times of these two methods described in Section 6, are both within 0.05 s, which also demonstrates our algorithm is very efficient.

8. Conclusion and future works

This paper proposes a novel method of generating tool path for 5-axis machining of the mesh surface with nonzero genus, including the continuous CC tool path and the smooth tool orientation. The proposed tool path method reduces the task of generating the spiral or contour-parallel tool path to the planning of much simple guide line in a rectangle, thus avoiding the complicated geometry computation, such as curve offsetting and self-intersection elimination involved in the traditional methods, etc. The analytical formula of computing CC point and parameter

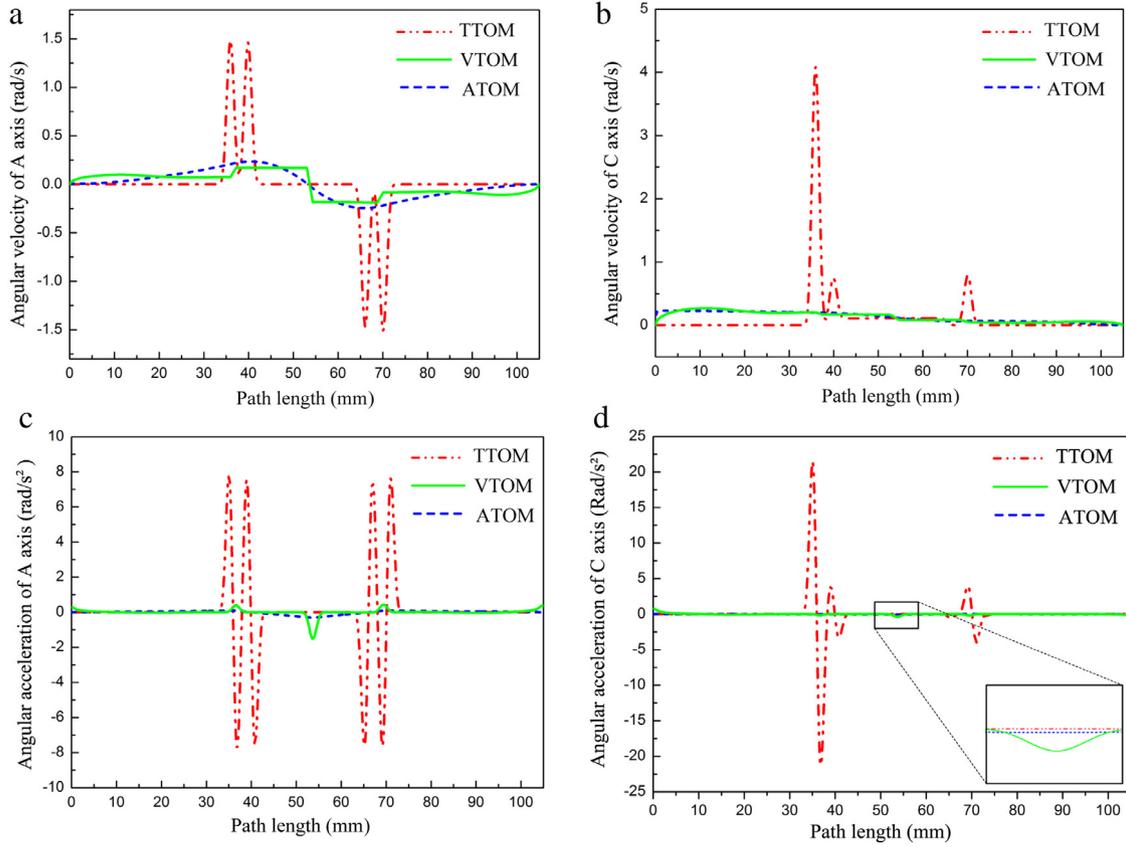


Fig. 20. The optimization results of tool orientation, TTOM stands for traditional tool orientation method, VTOM for angular velocity based tool orientation optimization method, ATOM for angular acceleration based tool orientation optimization method. (a) Angular velocity of A-axis before and after optimization; (b) Angular velocity of C-axis before and after optimization; (c) Angular acceleration of A-axis before and after optimization; (d) Angular acceleration of C-axis before and after optimization.

increments for the guide line are first derived on the discrete mesh. This makes the users can control more easily the machining accuracy than before. The real machining experiments are also performed to demonstrate the superiority to other methods. Another point worth mentioning is a simple and efficient method of optimizing tool orientation is also proposed which works directly on the rotary axis coordinate in MCS and can guarantee the motions of rotary axes of 5-axis machine tool as smooth as possible and the improvement on the kinematic and dynamic performance of 5-axis machine tool has been confirmed by the experimental results. Moreover, since the proposed algorithm only involves solving a linear equation system, it is also very efficient, which has been also already demonstrated by our experiments. In our experiments, it is also found that the benefits of our method may decrease with the increase of genus and holes distribution becoming more complicated. Although it can be nicely applicable to the industrial part like mouse, how to further enhance the adaptability of our method still needs more researches to be done in the future works. In addition, the experiment of tool orientation optimization validates the feasibility of the proposed method, but the influences of adjusting tool orientation on the motion of three translational axes, X-, Y- and Z-axis, of machine tool and the federate and how to reposition the tool according to the optimized tool orientation are also worth being further explored.

Acknowledgments

Grateful thanks to the reviewers for their constructive suggestions which help us improve this work. We also thank Prof. Y.-S. Lee of North Carolina State University for his valuable suggestions. This work is supported by the NSFC (Grant No. 51575086, 51525501, 11290143).

Appendix. Parameter increment for guide line

Assume \mathbf{c}_a is the adjacent CC point of the current CC point \mathbf{c} , then they satisfy the following equation.

$$\begin{cases} (\mathbf{c}_a(u, v) - \mathbf{c}(\sigma)) \cdot \mathbf{c}'(\sigma) = 0 \\ \|\mathbf{c}_a(u, v) - \mathbf{c}(\sigma)\| = L_w \end{cases} \quad (\text{A.1})$$

where $\sigma = (u_c, v_c)$ is the parameter of CC path $\mathbf{c}(\sigma)$ and L_w is the path interval. The first equation guarantees that the path interval direction, \mathbf{b} , is perpendicular to the feed direction and the second one guarantees that the distance between \mathbf{c}_a and \mathbf{c} is equal to the path interval L_w . $\mathbf{c}_a(u, v)$ is extended as one-order Taylor series:

$$\mathbf{c}_a(u, v) = \mathbf{c}(u_c, v_c) + \mathbf{r}_u(u_c, v_c)\Delta u + \mathbf{r}_v(u_c, v_c)\Delta v. \quad (\text{A.2})$$

With the condition of Eq. (A.2), Eq. (A.1) is rewritten as

$$\begin{cases} (\mathbf{c}_a(u, v) - \mathbf{c}(u_c, v_c)) \cdot \mathbf{c}'(u_c, v_c) = 0 \\ (\mathbf{c}_a(u, v) - \mathbf{c}(u_c, v_c)) \cdot \mathbf{b} = L_w \end{cases} \quad (\text{A.3})$$

$\mathbf{c}'(\sigma)$ and \mathbf{b} are expressed as

$$\begin{cases} \mathbf{c}'(u_c, v_c) = \mathbf{r}_u(u_c, v_c)\Delta u_f + \mathbf{r}_v(u_c, v_c)\Delta v_f \\ \mathbf{b} = \mathbf{r}_u(u_c, v_c)\Delta u_b + \mathbf{r}_v(u_c, v_c)\Delta v_b. \end{cases} \quad (\text{A.4})$$

It is known that, when the mapping from the mesh surface to the rectangle is conformal, \mathbf{r}_u and \mathbf{r}_v are orthogonal on the surface. Under this condition, submitting Eqs. (A.2) and (A.4) into Eq. (A.3) can rewrite Eq. (A.3) as

$$\begin{cases} \|\mathbf{r}_u\|^2 \Delta u_f \Delta u + \|\mathbf{r}_v\|^2 \Delta v_f \Delta v = 0 \\ \|\mathbf{r}_u\|^2 \Delta u_b \Delta u + \|\mathbf{r}_v\|^2 \Delta v_b \Delta v = L_w. \end{cases} \quad (\text{A.5})$$

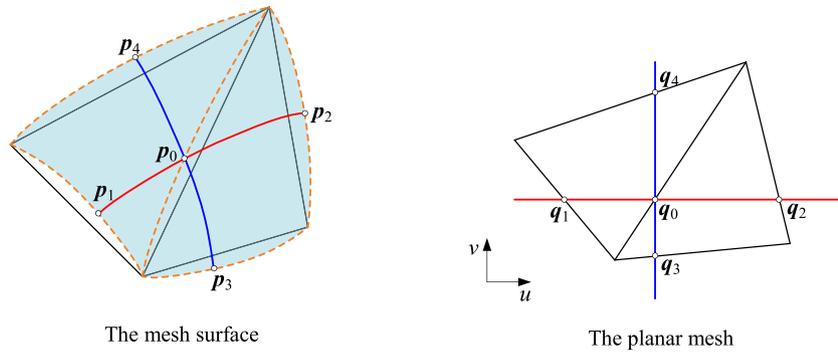


Fig. A.1. Calculation of the first order partial derivatives of the mesh surface.

The solution of Eq. (A.5) is

$$\begin{cases} \Delta u = \frac{L_w \Delta v_f}{\|\mathbf{r}_u\|^2 \Delta u_b \Delta v_f - \|\mathbf{r}_u\|^2 \Delta v_b \Delta u_f} \\ \Delta v = \frac{-L_w \Delta u_f}{\|\mathbf{r}_v\|^2 \Delta u_b \Delta v_f - \|\mathbf{r}_v\|^2 \Delta v_b \Delta u_f} \end{cases} \quad (\text{A.6})$$

In Eq. (A.6), only the first order partial derivatives of the nominal surface with respect to parameter u and v are unknown, which can be approximated by the difference scheme. As shown in Fig. A.1, for a given planar point q_0 , q_1 and q_2 are inserted in v parameter line through point q_0 . Their corresponding points, p_0 , p_1 and p_2 , on the nominal surface can be obtained using the method in Section 5.1. Extend p_1 and p_2 as Taylor series at p_0 .

$$\begin{cases} p_1 = \mathbf{r}(u_0 - \Delta u_1, v_0) = \mathbf{r}(u_0, v_0) - \mathbf{r}_u(u_0, v_0) \Delta u_1 + O(\Delta u_1^2) \\ p_2 = \mathbf{r}(u_0 + \Delta u_2, v_0) = \mathbf{r}(u_0, v_0) + \mathbf{r}_u(u_0, v_0) \Delta u_2 + O(\Delta u_2^2) \end{cases} \quad (\text{A.7})$$

Omitting the second order small quantity, we have

$$\mathbf{r}_u(u_0, v_0) = \frac{p_2 - p_1}{\Delta u_2 + \Delta u_1}, \quad \text{and similarly}$$

$$\mathbf{r}_v(u_0, v_0) = \frac{p_4 - p_3}{\Delta u_4 + \Delta u_3}.$$

Submitting them into Eq. (A.7) can obtain the parameter increments (Δu , Δv) at CC point c .

References

- [1] Wu T, Cheung EHM. Enhanced stl. *Int J Adv Manuf Technol* 2006;29(11–12):1143–50.
- [2] Kiswanto G, Lauwers B, Kruth JP. Gouging elimination through tool lifting in tool path generation for five-axis milling based on faceted models. *Int J Adv Manuf Technol* 2007;32(3–4):293–309.
- [3] Park SC, Chang M. Tool path generation for surface model with defects. *Comput Ind* 2010;61(1):75–82.
- [4] Xu JT, Sun YW, Wang SK. Tool path generation by offsetting curves on polyhedral surfaces based on mesh flattening. *Int J Adv Manuf Technol* 2013;64(9–12):1201–12.
- [5] Kim SJ, Yang MYA. CL surface deformation approach for constant scallop height tool path generation from triangular mesh. *Int J Adv Manuf Technol* 2006;28(3–4):314–20.
- [6] Lee SC, Kim HC, Yang MY. Mesh-based tool path generation for constant scallop-height machining. *Int J Adv Manuf Technol* 2008;37(1–2):15–22.
- [7] Sun YW, Guo DM, Jia ZY, et al. Iso-parametric tool path generation from triangular meshes for free-form surface machining. *Int J Adv Manuf Technol* 2006;28(7–8):721–6.
- [8] Sun YW, Ren F, Zhu XH, et al. Contour-parallel offset machining for trimmed surfaces based on conformal mapping with free boundary. *Int J Adv Manuf Technol* 2012;60(1–4):261–71.
- [9] Xu JT, Sun YW, Zhang L. A mapping-based approach to eliminating self-intersection of offset paths on mesh surfaces for CNC machining. *Comput Aided Des* 2015;62:131–42.
- [10] Bolaños GS, Bedi S, Mann S. A topological-free method for three-axis tool path planning for generalized radiused end milled cutting of a triangular mesh surface. *Int J Adv Manuf Technol* 2014;70(9–12):1813–25.
- [11] Kout A, Müller H. Tool-adaptive offset paths on triangular mesh workpiece surfaces. *Comput Aided Des* 2014;50:61–73.
- [12] Zhang K, Tang K. An efficient greedy strategy for five-axis tool path generation on dense triangular mesh. *Int J Adv Manuf Technol* 2014;74(9–12):1539–50.
- [13] Lee EK. Contour offset approach to spiral toolpath generation with constant scallop height. *Comput Aided Des* 2003;35(6):511–8.
- [14] Makhanov SS, Ivanenko SA. Grid generation as applied to optimize cutting operations of the five-axis milling machine. *Appl Numer Math* 2003;46(3–4):331–51.
- [15] Sun YW, Guo DM, Jia ZY. Spiral cutting operation strategy for machining of sculptured surfaces by conformal map approach. *J Mater Process Technol* 2006;180(1–3):74–82.
- [16] Huertas-Talón JL, García-Hernández C, Berges-Muro L, et al. Obtaining a spiral path for machining STL surfaces using non-deterministic techniques and spherical tool. *Comput Aided Des* 2014;50:41–50.
- [17] Tang TD. Algorithms for collision detection and avoidance for five-axis NC machining: a state of the art review. *Comput Aided Des* 2014;5:1–17.
- [18] Chiou JC, Lee YS. Optimal tool orientation for five-axis tool-end machining by swept envelope approach. *ASME J Manuf Sci Eng* 2005;127(4):810–8.
- [19] Fard MJB, Feng HY. Effective determination of feed direction and tool orientation in five-axis flat-end milling. *ASME J Manuf Sci Eng* 2005;132(6):061011.
- [20] Fard MJB, Feng HY. New criteria for tool orientation determination in five-axis sculptured surface machining. *Int J Prod Res* 2011;49(20):5999–6015.
- [21] De Lacalle LL, Lamikiz A, Sanchez JA, Salgado MA. Toolpath selection based on the minimum deflection cutting forces in the programming of complex surfaces milling. *Int J Mach Tools Manuf* 2007;47(2):388–400.
- [22] Yigit IE, Lazoglu I. Analysis of tool orientation for 5-axis ball-end milling of flexible parts. *CIRP Ann Manuf Technol* 2015;64(1):97–100.
- [23] Wang N, Tang K. Five-axis tool path generation for a flat-end tool based on iso-conic partitioning. *Comput Aided Des* 2008;40(10):1067–79.
- [24] Morishige K, Takeuchi Y, Kase K. Tool path generation using C-space for 5-axis control machining. *ASME J Manuf Sci Eng* 1999;121(1):144–9.
- [25] Jun CS, Cha K, Lee YS. Optimizing tool orientations for 5-axis machining by configuration-space search method. *Comput Aided Des* 2003;35(6):549–66.
- [26] Lauwers B, Dejonghe P, Kruth JP. Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation. *Comput Aided Des* 2003;35(5):421–32.
- [27] Wang N, Tang K. Automatic generation of gouge-free and angular-velocity-compliant five-axis toolpath. *Comput Aided Des* 2007;39(8):841–52.
- [28] Farouki RT, Li SQ. Optimal tool orientation control for 5-axis CNC milling with ball-end cutters. *Comput Aided Geom Design* 2013;30(2):226–39.
- [29] Han CY. Tractrix-based tool orientation control for 5-axis CNC machining. *Appl Math Comput* 2016;272:92–9.
- [30] Hu PC, Tang K. Improving the dynamics of five-axis machining through optimization of workpiece setup and tool orientations. *Comput Aided Des* 2011;43(12):1693–706.
- [31] Castagnetti C, Duc E, Ray P. The domain of admissible orientation concept: a new method for five-axis tool path optimization. *Comput Aided Des* 2008;40(9):938–50.
- [32] Lavernhe S, Tournier C, Lartigue C. Optimization of 5-axis high-speed machining using a surface based approach. *Comput Aided Des* 2008;40(10):1015–23.
- [33] Chen L, Xu K, Tang K. Collision-free tool orientation optimization in five-axis machining of bladed disk. *J Comput Des Eng* 2015;2(4):197–205.
- [34] Ho MC, Hwang YR, Hu CH. Five-axis tool orientation smoothing using quaternion interpolation algorithm. *Int J Mach Tools Manuf* 2003;43(12):1259–67.
- [35] Lin ZW, Fu JZ, He Y, Gan WF. A robust 2D point- sequence curve offset algorithm with multiple islands for contour-parallel tool path. *Comput Aided Des* 2013;45(3):657–70.
- [36] Gu X. Parameterization for surfaces with arbitrary topologies (Ph.D. thesis), Cambridge: Harvard University; 2002.
- [37] Walton DJ, Meek DS. A triangular G^1 patch from boundary curves. *Comput Aided Des* 1996;28(2):113–23.
- [38] Meyer M, Desbrun M, Schröder P, et al. Discrete differential-geometry operators for triangulated 2-manifolds. *Vis Math* 2002;3(2):52–8.