The International Journal of Advanced Manufacturing Technology (2019) 104:645–659
https://doi.org/10.1007/s00170-019-03914-9

ORIGINAL ARTICLE

Collision-free regions of tool posture in five-axis machining of blisk with a filleted end mill

Zhiwei Wang[1] · Yaoyao Shi[1] · Xiaojun Lin[1] · Yuan Gao[1]

Received: 29 January 2019 / Accepted: 21 May 2019 / Published online: 6 June 2019

Abstract
In five-axis machining of blisk, the filleted end mill has attracted more and more attention because of its larger cutting width. However, it is still a hard work to find the tool posture without interference. In this paper, a method is proposed to solve the collision-free regions. Based on the visibility of free-form surface, a tool-surface tangent model of the filleted end mill is established. With the model, only critical points on profiles of checking surface are searched with a self-adapting step length and the corresponding critical vectors are calculated and mapped to construct the collision-free regions. Firstly, critical points on the boundaries are searched according to the given precision. Meanwhile, the corresponding critical vectors are calculated and some special searched points are selected as the endpoints of each profile. Then, the adjacent critical points are searched along the profile by adjusting iteratively with a self-adapting step length in the parameter domain one by one. During the search, the corresponding critical vectors are calculated too. After that, the critical vectors are mapped to construct the subinterval collision-free regions in two-dimensions. And a method is adopted to combine collision-free regions. This algorithm is finally verified with a closed blisk and compared with a referenced method. The results show that it can efficiently solve collision-free regions in five-axis milling of blisk with a filleted end mill.

Keywords Five-axis · Tool posture · Collision-free region · Blisk machining · Filleted end mill

1 Introduction

Five-axis CNC machines are widely used in machining of sculptured surfaces such as turbine blades, impellers, blisk, molds, and dies [1, 2]. As one of the key parts used in new jet engines, the integrated structure of blisk reduces the engine's weight, part count, and so on. However, it also introduces more geometrical constraints to the parts, which increase the complexity of tool orientation planning in milling process. When machining these complex products, it is necessary to flexibly control the direction of the tool posture to avoid interference [3–5]. And the filleted end mill has attracted more and more attention because of its larger cutting width [6–10]. So collision detection and avoidance becomes one of the key problems in machining of blisk with filleted end mill.

Collision problems in five-axis machining are divided into local interference and global collision [2, 4, 11]. The local interference usually refers to local gouging and rear gouging. Gouging means that the cutter cuts into the machined surface deeper than the expected geometry. Many methods are proposed to avoid the local interference [2, 12]. These techniques dealt with the local interference to obtain gouging-free tool orientations or locations with the concept of curvature matching.

Moreover, global collision is considered to be more serious as accidents may happen if such interference occurs in five-axis machining. It mainly focuses on the interference between the cutter and objects involved in machining. Global interference detection and avoidance are still a current technique challenge in five-axis milling [7, 13]. In general, there are two kinds of approaches to find the collision-free tool postures. The one firstly generates tool orientations according to some strategies and then adjusts them with interference detection, while the other one directly calculates collision-free regions and then optimizes tool orientations within these spaces [14].

✉ Zhiwei Wang
wangzw87@yeah.net

[1] School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

Many methods are proposed based on the first approach [2, 15]. Wu et al. [16] presented a rotation method to eliminate the interference in milling of impellers with a non-orthogonal four-axis machine tool. In these studies, although the noninterference of existing tool orientations is mainly guaranteed, it is not convenient for further tool orientation optimization. In addition, these adjustment methods usually execute interference detection and correction iteratively for several times, which require large computation time and resource.

To avoid deficiencies of iterative adjustments, some research work is devoted to solving the problem of collision-free region directly. Since the interference between a cutter and free-form surfaces is difficult to identify, a convenient approach is converting involved surfaces into points, lines, or subdivided meshes [17, 18]. Chen et al. [17] expressed the sculptured surface as a triangular mesh body and solved the collision-free regions to optimize and smooth tool orientations. Lin et al. [18] expressed the machined surface as point cloud and detect collisions between the cutter and these points. These researches successfully calculate noninterference spaces of tool orientations with adjustment or search method. The efficiency and accuracy of interference detection are largely dependent upon the quantity of sampled or subdivided objects. As the quantity of objects increases, the computation cost will increase rapidly.

There is another perspective to detect the collision by taking advantage of the graphics processing unit in graphics hardware. Bi et al. [19] computed collision-free regions in milling of an impeller with the assistance of occlusion query functionality of the graphics hardware. Wang et al. [20] proposed a two-phase strategy to detect collisions between the cutter and triangulated obstacles. The hardware approach relies on expensive hardware and supporting software system.

An improved approach is presenting involved surfaces as a hierarchical structure, and only the interested parts involved in interference detection are further subdivided iteratively. Hu et al. [14] presented rigorous analyses of the obstacles in five-axis machining and propose efficient numerical algorithms for calculating and representing them. Tang et al. [21] approximated the machined part as an octree of bounding sphere to recursively conduct collision detection in five-axis machining. With the hierarchical approaches, the computational efficiency can be improved because the data involved in the interference calculation can be reduced. However, it is still time consuming to detect the interference repeatedly.

In addition, a novel approach was presented by Liang et al. [4] to solve the accessible regions of tool orientations of ball-end mill. Based on the visibility, only critical points related to the boundaries of accessible regions are searched and processed in two-dimensions. During the search, the checking surface was replaced by an offset surface and the ball-end cutter is converted to a ray accordingly that rotates around the known fixed cutter location point. Therefore, the line between the point on the offset surface and the fixed point is the critical vector to be verified. However, this method is not suitable for filleted end mill. This is because the cutter location point also changes when adjusting the tool posture. And it is almost impossible to construct the offset surface of the checking surface and a ray to replace the cutter.

On the basis of the method described in literature [4], in this paper, an improved method is proposed to solve the collision-free regions which can be applied to the filleted end mill. Based on the concept of free-form visibility, the tool-surface model in critical state can be established. With the model, the boundaries of subinterval feasible regions are related to the critical points on the profiles of checking surface. And the critical points can be searched one by one along the profiles. The rest of this paper is arranged as follows. In Section 2, the strategy of this approach is introduced. And the algorithms to search critical points are proposed in Section 3. Then, the combination of collision-free region is introduced in Section 4. In Section 5, the proposed algorithm is verified and evaluated with a closed blisk and comparing with a referenced method. The work is concluded in the last section.

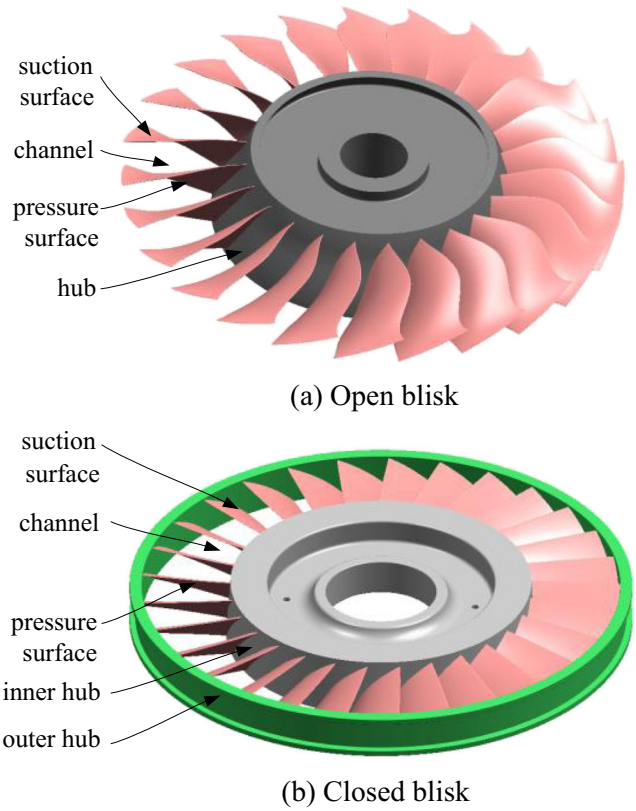

Fig. 1 Typical structure of blisk

2 Strategy to find collision-free region

As shown in Fig. 1, a channel of a blisk is formed by two adjacent surfaces and hubs. To machine a point located inside of this channel, tool orientations must be planned appropriately to avoid the potential collisions from the surfaces and hubs. For convenience, the surfaces and hubs of a blisk are represented by the checking surface in the following part.

A collision-free region is a set of tool posture along which the cutter will not collide or interfere with the checking surfaces. Imagine that there is a virtual sphere which can cover all the checking surfaces centered at the cutter location point. When putting a light source at this point, the region on the sphere that the rays can reach to is not obstructed by these surfaces. It is called collision-free region, while the obstructed region is called collision region. Geometrically, these regions can be presented by their boundaries which are constituted with central projections of these checking surfaces' profiles.

As shown in Fig. 2, it can be observed how the rays are obscured by the checking surface in a plane which passes through the cutter location point P_M. Those rays similar to l_{R5} can reach the virtual sphere, while the rays similar to l_{R1}, l_{R2}, and l_{R3} will be occluded. And the rays similar to l_{R4} will be in the critical state, which just reach the boundaries of the collision-free regions on the virtual sphere. It is these special rays that determine the boundaries of the collision-free regions. Moreover, the critical rays are different in a and b in Fig. 2. In Fig. 2a, the ray l_{R4} is just across the boundary of the checking surface. However, the ray l_{R4} in Fig. 2b is tangent to the checking surface. In addition, the ray l_{R2} is also special. In a very small neighborhood δp, the ray l_{R2} is tangent to the checking surface at the point P_a. In other words, the ray l_{R4} is the special state of the ray l_{R2}.

In this paper, the tool similar to the rays of l_{R2} or l_{R4} is called in the critical state. And the vector used to represent the tool posture is named critical vector. Accordingly, the points on the checking surface similar to the points P_a or P_b in Fig. 2b is called critical point. And the curve formed by those critical points is called the profile of the checking surface. For convenience, all the points on the boundary is also called critical point.

Each profile or boundary determines a subinterval collision-free region on the virtual sphere. If all the profiles associated with the channel can be found, the collision-free regions will then be solved with the intersection of those subinterval regions. So the key to solving the problem of collision-free region is to find the profiles of a single checking surface.

The tool is composed of two parts, as shown in Fig. 3, where the non-cutting part of the tool is the cylinder with radius R, while the cutting part is the torus with radius r and $R > r > 0$. The normal vector $\mathbf{n}_M$ of the torus at any machining point P_M points to the axis of the cutter. And in the plane passing through point P_M and the axis of the cutter, the vector $\mathbf{n}_M$ passes through the fixed point E, which is the center of arc. So the distance between the two points P_M and E is equal to radius r. The geometric relationship between the tool and the checking surface and the machining surface will be discussed respectively when it is in the critical state in the next.

Shown in Fig. 4 is the tool in the critical state. The torus of the cutter is tangent to the machining surface $S_M(u,v)$ at the

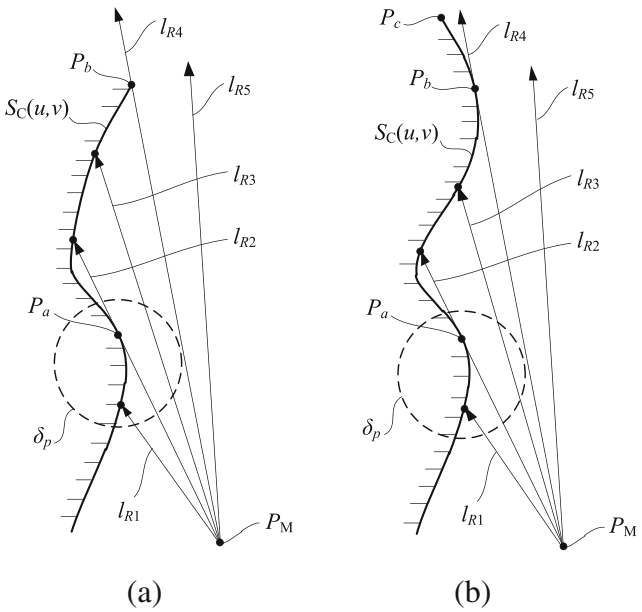

Fig. 2 Rays obscured by the checking surface

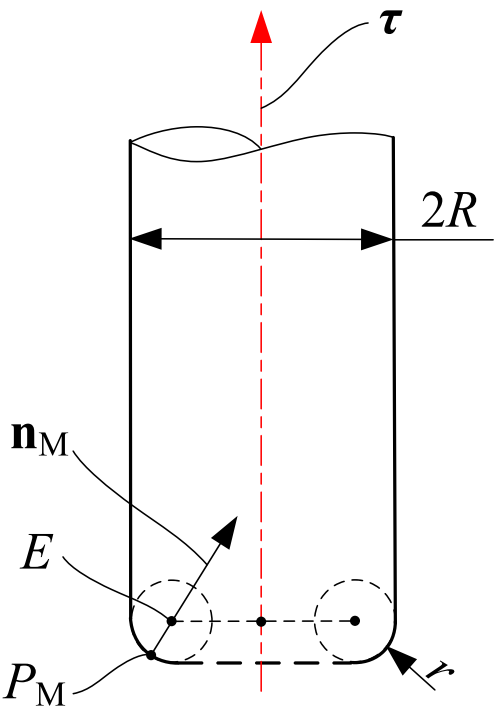

Fig. 3 Parameters of tool structure

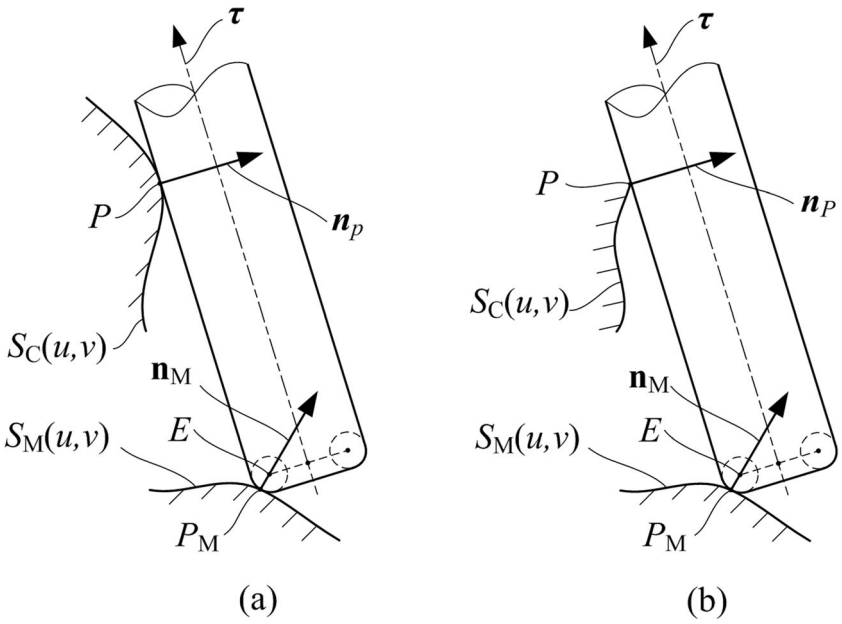

Fig. 4 The tool in the critical state

machining point P_M. The tool posture is represented by vector τ. Here, the normal vector at the tangent point is the vector $\mathbf{n}_M$ in Fig. 3. At the same time, the tool is tangent to the checking surface $\mathbf{S}_C(u,v)$ at the point P by adjusting the vector of $\boldsymbol{\tau}$. Then, the point P is a critical point. The common normal vector of the cutter and the checking surface at the critical point is denoted by $\mathbf{n}_P$. For the convenience of data processing, a local orthogonal coordinate system $E(x_m,y_m,z_m)$ is set up at the point P_M as follows: (1) the point E is set to the origin of the coordinates, (2) and the vector $\mathbf{n}_M$ is set to the z_m axis, and (3) the movement direction of the cutter at point E is set to the x_m axis. Then the y_m axis is determined by the right-hand criterion. All the descriptions and calculations are carried out in the coordinates in the following part, unless otherwise specified.

However, the tool-surface tangent model is divided into two different cases. Firstly, as shown in a in Fig. 4, the tangent point is inside the surface, which means that the normal vectors at the critical point of the two surfaces are in the same direction. Secondly, as shown in b in Fig. 4, the tangent point is on the boundary of the checking surface. And there is no special requirement for the direction of the common normal vector and the normal vector of the checking surface at the critical point. In other words, any point on the boundary is a critical point and the corresponding critical vector can be found, while not all points inside the checking surface are critical points. For some special critical points on the boundary, they may satisfy the above two conditions. Then, these points are both on the boundary and on the profile.

Any curve on the checking surface can be represented by sample points distributed over it, whether it is a boundary or a profile. And those points can be searched one by one along the curve. In this paper, all sample points on the four boundaries of the checking surface will be searched with a given discrete precision and the corresponding critical vectors are calculated. And the endpoints of the profiles of the checking surface are selected from the searched sample points. Then, the critical points on the profiles are searched one by one, starting from those endpoints.

Each profile, if it exists, has two endpoints. One is called the starting point and is denoted by $P_{i,s}$. The other one is called the end point, which is denoted by $P_{i,e}$. As shown in Fig. 5, starting from a starting point $P_{1,s}$, the first profile PL_1 and the end point $P_{1,e}$ of the profile PL_1 can be searched. Delete the point closest to point $P_{1,e}$ in the set of searched endpoints. At the same time, the point $P_{1,s}$ removed also from the set. Then, a new starting point $P_{2,s}$ will be taken out of the updated set at random and the second profile PL_2 continues to be searched. After that, the process will be repeated until there are no new

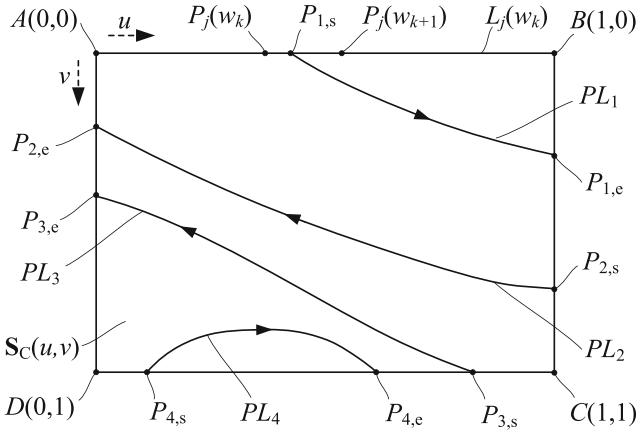

Fig. 5 The process of searching profiles

starting points. The algorithm to search the critical points on a checking surface is shown in Fig. 6.

3 Algorithm to search sample critical points

In this part, the algorithm to solve sample critical points and the corresponding critical vectors will be introduced in detail. During the search, the maximum allowable angle between the adjacent critical vectors is called the discrete precision. This value, denoted by θ_δ, is pre-set as needed which is a very small number. According to the critical point position, the process is divided into two phases. First, the critical points on the boundaries are searched and the endpoints of the profile are selected. Then, the sample critical points on the profile of the checking surface will be searched starting from the selected endpoints.

3.1 Search sample points on boundary

Suppose four boundaries of the checking surface are $\mathbf{S}_C(u,0)$, $\mathbf{S}_C(u,1)$, $\mathbf{S}_C(0,v)$, and $\mathbf{S}_C(1,v)$. For convenience, use $L_j(w)$ to represent the jth boundary. Here, w is the curve parameter ($w_{jS} \le w \le w_{jE}$), while w_{jS} and w_{jE} are the parameters at the start and end points of the jth boundary, respectively. The kth sample critical point on the boundary $L_j(w)$ is represented by $P_j(w_k)$. The tangent vector of the curve $L_j(w)$ at point $P_j(w_k)$ is denoted by $\mathbf{m}_j(w_k)$. And the common normal vector at the point $P_j(w_k)$ is denoted by $\mathbf{n}_j(w_k)$, as shown in Fig. 7. The intersection of the common normal line at the point $P_j(w_k)$ and the axis of the cutter is denoted by $F_j(w_k)$, while the intersection of the common normal line at the point P_M and the axis of the cutter is denoted by $G_j(w_k)$. The intersection between the perpendicular line passing through the point E and the axis of the cutter is denoted as $M_j(w_k)$. The tool posture, which is represented by the vector $\boldsymbol{\tau}_j(w_k)$, can be determined by points $M_j(w_k)$ and $G_j(w_k)$.

The coordinate values of point $M_j(w_k)$ and point $G_j(w_k)$ can be solved by the geometric constraint relation that the cutter is tangent to both boundary $L_j(w)$ and the machining surface $\mathbf{S}_M(u,v)$. The unit vector $\mathbf{n}_M$ can be calculated with Eq. 1. Here, the outward direction is used in the algorithm.

$$\mathbf{n}_M = \frac{\frac{\partial \mathbf{S}_M(u,v)}{\partial u} \times \frac{\partial \mathbf{S}_M(u,v)}{\partial v}}{\left| \frac{\partial \mathbf{S}_M(u,v)}{\partial u} \times \frac{\partial \mathbf{S}_M(u,v)}{\partial v} \right|} \tag{1}$$

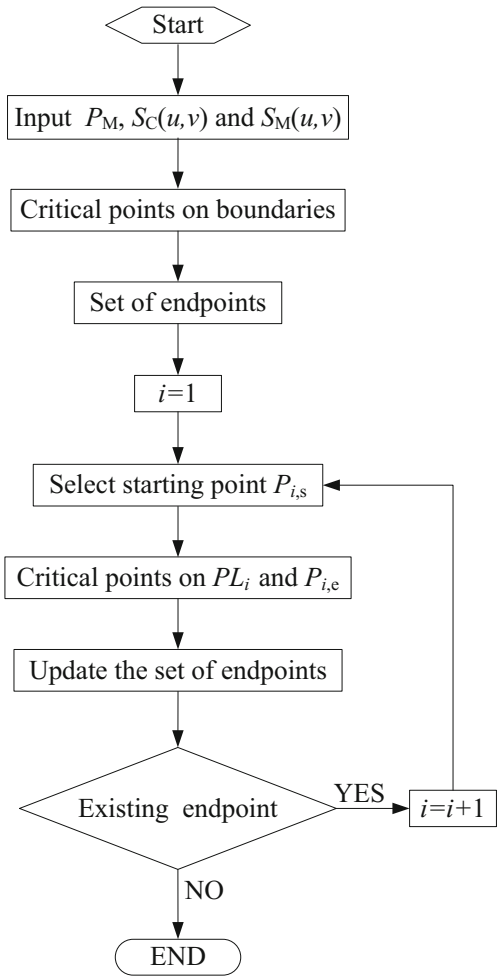

Fig. 6 Flow chart of searching critical points

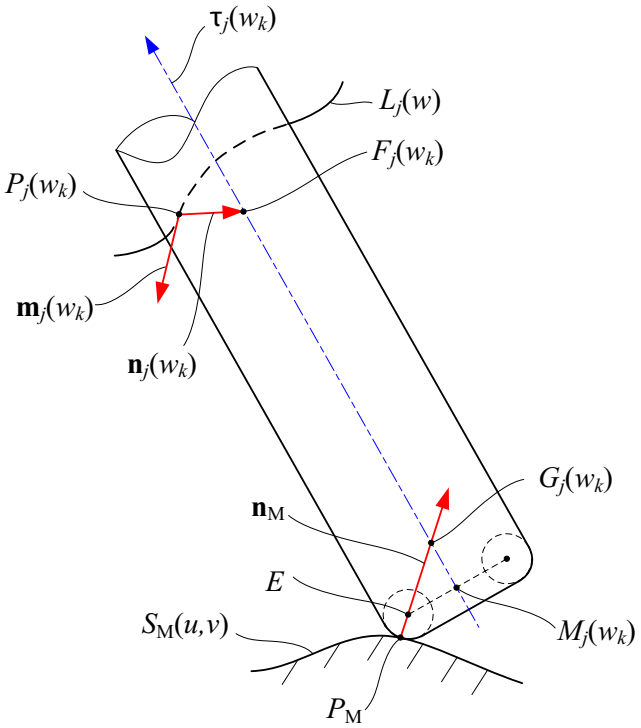

Fig. 7 The cutter is tangent to the boundary

And the vector $\overrightarrow{EG_j(w_k)}$ can be represented by the vector $\mathbf{n}_M$ with a positive real number a_k as Eq. 2.

$$\overrightarrow{EG_j(w_k)} = a_k \cdot \mathbf{n}_M \tag{2}$$

In addition, the points $M_j(w_k)$, $G_j(w_k)$, and $F_j(w_k)$ are all on the axis of cutter, so there should be a non-zero b_k which satisfies Eq. 3.

$$\overrightarrow{M_j(w_k)F_j(w_k)} = b_k \cdot \overrightarrow{M_j(w_k)G_j(w_k)} \tag{3}$$

At the same time, the distance between point $P_j(w_k)$ and point $F_j(w_k)$ should be equal to $R + \Delta s$ and the distance between the point E and the axis of cutter should be equal to R-r. Then, the point $M_j(w_k)$ and point $F_j(w_k)$ should satisfy Eq. 4. Here, Δs is safety allowance, which is the distance between the cutter and the checking surface when the cutter is in critical state.

$$\begin{cases} |\overrightarrow{P_j(w_k)F_j(w_k)}| = R + \Delta s \\ |\overrightarrow{EF_j(w_k)}| = R - r \end{cases} \tag{4}$$

Furthermore, the vector $\boldsymbol{\tau}_j(w_k)$ is perpendicular to vector $\overrightarrow{EM_j(w_k)}$ and vector $\overrightarrow{P_j(w_k)F_j(w_k)}$, respectively. And the vector $\mathbf{m}_j(w_k)$ also is perpendicular to vector $\overrightarrow{P_j(w_k)F_j(w_k)}$. Then, the point $M_j(w_k)$, point $G_j(w_k)$, and point $F_j(w_k)$ should also satisfy Eq. 5.

$$\begin{cases} \mathbf{m}_j(w_k) \cdot \overrightarrow{P_j(w_k)F_j(w_k)} = 0 \\ \overrightarrow{EM_j(w_k)} \cdot \overrightarrow{M_j(w_k)G_j(w_k)} = 0 \\ \overrightarrow{P_j(w_k)F_j(w_k)} \cdot \overrightarrow{M_j(w_k)G_j(w_k)} = 0 \end{cases} \tag{5}$$

In Eq. 4 and Eq. 5, there are five unknowns and five independent equations. The values of a_k, b_k and the coordinates of point $M_j(w_k)$ can be solved. And the outward direction of vector $\overrightarrow{P_j(w_k)F_j(w_k)}$ is used in the algorithm. Then, the coordinates of points $G_j(w_k)$ and $F_j(w_k)$ can be solved with Eq. 2 and Eq. 3. Furthermore, the unit vector $\boldsymbol{\tau}_j(w_k)$ can be calculated by Eq. 6.

$$\boldsymbol{\tau}_j(w_k) = \frac{\overrightarrow{M_j(w_k)G_j(w_k)}}{|\overrightarrow{M_j(w_k)G_j(w_k)}|} \tag{6}$$

Next, the sample points on the boundary $L_j(w)$ are searched. And the included angle between the adjacent critical vectors $\boldsymbol{\tau}_j(w_k)$ and $\boldsymbol{\tau}_j(w_{k+1})$ is denoted by $\Delta\theta_k$, which should not be greater than the criterion θ_δ. The search process is divided into five steps, as shown below.

Step 1 Starting at the start point of the boundary $L_j(w)$, the corresponding unit critical vector $\boldsymbol{\tau}_j(w_k)$ is calculated on the sample points $P_j(w_k)$. Here, $k = 1$ and $w_k = 0$.

Step 2 Search the next sample point $P_j(w_{k+1})$ on the boundary with a step of Δw and calculate the corresponding critical vector $\boldsymbol{\tau}_j(w_{k+1})$. Here, $w_{k+1} = w_k + \Delta w$.

Step 3 Calculate the included angle $\Delta\theta_k$ between the adjacent critical vectors $\boldsymbol{\tau}_j(w_k)$ and $\boldsymbol{\tau}_j(w_{k+1})$. If $\Delta\theta_k > \theta_\delta$, then turn to the next step; otherwise, turn to step 5.

Step 4 Change the step Δw by $0.5\Delta w$ temporarily. Search the point $P_j(w_{k+1})$, and calculate the vector $\boldsymbol{\tau}_j(w_{k+1})$ again. Judge whether the parameter $\Delta\theta_k$ meets the condition. If $\Delta\theta_k > \theta_\delta$, update the parameter Δw again in the same way; otherwise, turn to step 5.

Step 5 If $w_{k+1} + \Delta w < w_{jE}$, let $k = k + 1$ and turn to step 2; otherwise, the search is over.

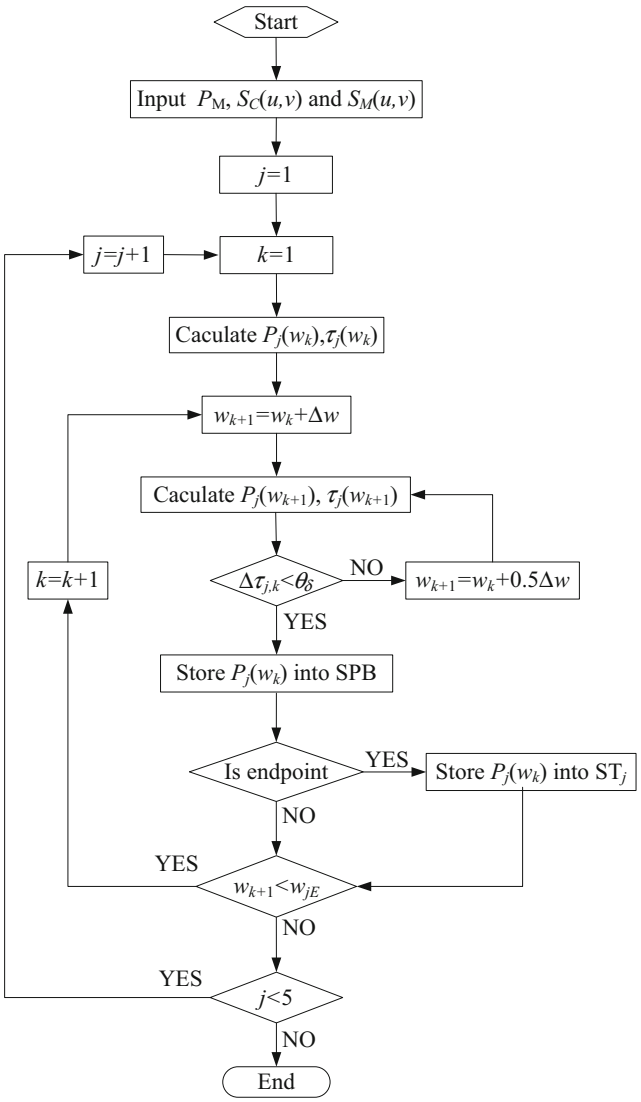

Fig. 8 Flow chart of solving critical points on the boundaries

In this way, all sample points on the boundaries can be searched one by one and the flow chart of algorithm is shown in Fig. 8. After searching the sample points on the boundaries of checking surface, the endpoints of the profiles should be solved if there are. So some special points will be selected from the searched critical points on the boundaries.

Assume that the sample point $P_j(w_k)$ is one of the endpoints, at where the cutter and the checking surface are tangent. Then, the normal vectors at that point of the cutter and the checking surfaces should be in the same direction. Here, the normal vector $\overrightarrow{P_j(w_k)F_j(w_k)}$ of the cutter has been calculated in Eq. 5 and the normal vector of the checking surfaces can be calculated, which will be described in Section 3.2. Considering the adjacent sample points on the boundary are very close and the number of endpoints is relatively small, the point $P_j(w_k)$ is directly taken as the endpoint of the profile if the angle between normal vectors of the cutter and the checking surfaces is no greater than the precision θ_δ.

During the search, the searched points are stored into set SPB in order and the selected endpoints are stored into set ST_j accordingly.

3.2 Search adjacent sample critical point on profile

The process of searching adjacent sample critical point is divided into two parts. It firstly moves with a large step in a certain direction to approach the desired point and then adjusts with adaptive steps to reach the adjacent critical point. This part mainly focuses on the direction and step length of the search process. And the second part will be discussed in Section 3.3 and Section 3.4.

For convenience, the ith profile on the checking surface $\mathbf{S}_C(u,v)$ is represented by PL_i. The current known critical point and the next adjacent critical point on the ith profile are denoted by $P_{i,C}$ and $P_{i,N}$, respectively, as shown in Fig. 9. In addition, the corresponding critical vector at the point $P_{i,C}$ is denoted as $\boldsymbol{\tau}_{i,C}$, and the normal vector of the checking surface at the point $P_{i,C}$ is denoted by $\mathbf{n}_{i,C}$, Here, the vector $\mathbf{n}_{i,C}$ can be got with Eq. 7 and the outward direction is used in the algorithm.

$$\mathbf{n}_{i,C} = \frac{\frac{\partial S_C(u,v)}{\partial u} \times \frac{\partial S_C(u,v)}{\partial v}}{\left|\frac{\partial S_C(u,v)}{\partial u} \times \frac{\partial S_C(u,v)}{\partial v}\right|} \tag{7}$$

Keeping the cutter tangent to the checking surface, when the critical point moves from the point $P_{i,C}$ to the next sample critical point $P_{i,N}$, the vector of the tool orientation also changed accordingly. The angle between the adjacent critical vectors is denoted by $\Delta\theta$. And the angle $\Delta\theta$ should not be greater than the accuracy θ_δ, which means that the points $P_{i,C}$ and $P_{i,N}$ are pretty close together. Then, the direction vector $\mathbf{n}_{\tan}$, from point $P_{i,C}$ to point $P_{i,N}$, is almost perpendicular to both the vectors $\boldsymbol{\tau}_{i,C}$ and $\boldsymbol{\tau}_{i,N}$, respectively. In addition, the vector $\mathbf{n}_{\tan}$ is almost parallel to the tangent plane of the checking surface at the point $P_{i,C}$. Then, the normal vector $\mathbf{n}_{i,C}$ should be perpendicular to the searching direction. Therefore, the search direction vector $\mathbf{n}_{\tan}$ can be calculated by Eq. 8. Here, the direction to the next point is used in this algorithm.

$$\mathbf{n}_{\tan} = \pm\left(\mathbf{n}_{i,C} \times \boldsymbol{\tau}_{i,C}\right) \tag{8}$$

Under the condition of a given precision θ_δ, the rough step length Δd can be calculated with Eq. 9 and will be adjusted as needed in the search. Here, l is the distance d between the points $P_{i,C}$ and E.

$$\Delta d \approx 2l \cdot \sin\frac{\Delta\theta_\delta}{2} \tag{9}$$

Then, a point P_i near the target point $P_{i,N}$ can be searched from the given point $P_{i,C}$ along the direction vector $\mathbf{n}_{\tan}$ with a step length Δd, as shown in Fig. 9.

In the parameter domain of the checking surface, the point $P_{i,C}$ can be denoted as $(u_{i,C}, v_{i,C})$, while the search step Δd will change into $(\Delta u, \Delta v)$, where Δu and Δv are the variations of parameters from point $P_{i,C}$ to point $P_{i,N}$ and that should satisfy Eq. 10.

$$(\Delta u \cdot \mathbf{n}_u + \Delta v \cdot \mathbf{n}_v) \cdot \mathbf{n}_{\tan} = \pm|(\Delta u \cdot \mathbf{n}_u + \Delta v \cdot \mathbf{n}_v)| \cdot |\mathbf{n}_{\tan}| \tag{10}$$

where $\mathbf{n}_u$ and $\mathbf{n}_v$ are the directional derivatives along the direction of u and v respectively at the point $P_{i,C}$. The parameters Δu and Δv are also the decomposition of the search vector $\mathbf{n}_{\tan}$ in the direction of the vectors $\mathbf{n}_u$ and $\mathbf{n}_v$. And at least one parameter is not zero.

The process of searching adjacent critical point $P_{i,N}$ is divided into six steps, as shown below, where the method of searching critical point in step 5 will be described in detail in Sections 3.3 and 3.4.

Step 1 Set Δu or Δv to a small value, such as 0.001, and calculate the other one with Eq. 10.

Step 2 Search the point P_i on the checking surface $\mathbf{S}_C(u,v)$ with the step $(\Delta u, \Delta v)$.

Step 3 Calculate the ratio λ_d. Here, $\lambda_d = \Delta d/\Delta d_P$ and Δd_P is the distance between points $P_{i,C}$ and P_i.

Step 4 If $0.95 < \lambda_d < 1$, the point P_i is considered to be near the adjacent critical point $P_{i,N}$. Otherwise, let $\Delta u = \lambda_d \Delta u$ and $\Delta v = \lambda_d \Delta v$ and turn to step 2 until λ_d is satisfied. Here, the value 0.95 can be adjusted as needed.

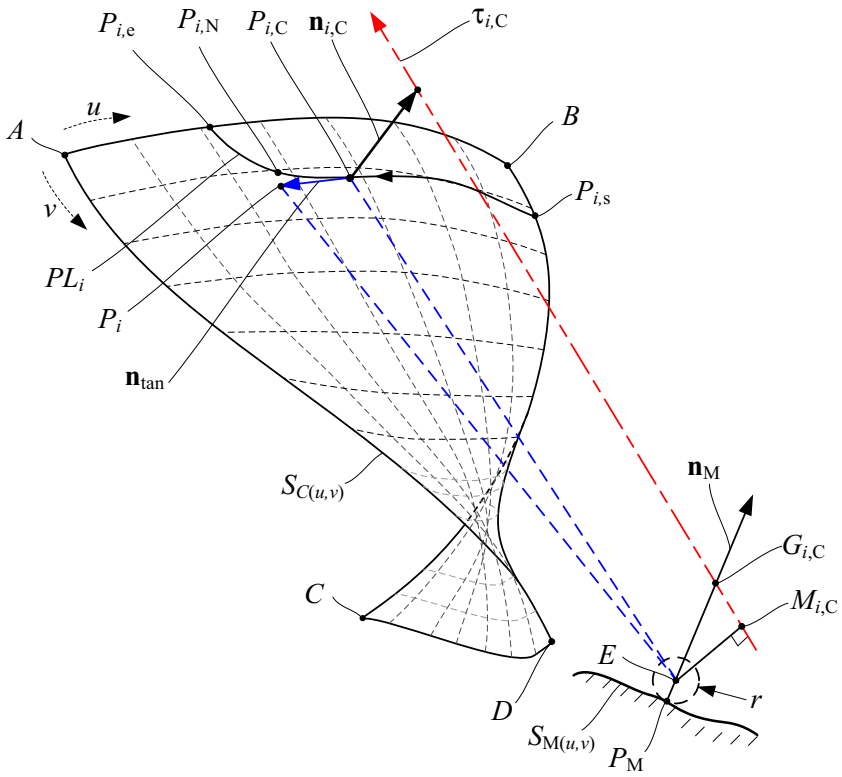

Fig. 9 Search adjacent critical point along the profile

Step 5 Search the critical point $P_{i,N}$ with the point P_i as the starting point, and calculate the corresponding critical vector $\tau_{i,N}$.

Step 6 Calculate the angle $\Delta\theta$ between the vectors $\tau_{i,C}$ and $\tau_{i,N}$. If $\Delta\theta \le \theta_\delta$, let $j = j + 1$ and search for the next adjacent critical point. Otherwise, let $\lambda_\theta = (\Delta\theta\text{-}\theta_\delta)/\Delta\theta$, $\Delta u = \lambda_d \lambda_\theta \Delta u$, and $\Delta v = \lambda_d \lambda_\theta \Delta v$. Then, update the point $P_{i,0}$ and turn to step 5.

Step 7 All the adjacent critical points on one profile of the checking surface can be searched one by one through the search process described above. During the search, the searched points and the corresponding critical vectors on the ith profile are stored into set SP_i in order.

3.3 Critical point identification

During the search of adjacent critical point in Section 3.2, the point P_i near the target point has been found and the target critical point needs to be searched. It is mainly divided into two steps. First, determine whether the point is a critical point. Then, if not, search a critical point near it. Here, the first part will be solved in this section and the second will be described in Section 3.4.

Suppose that the point P_i is a tangent point at where the cutter and the checking surface are tangent. Then, it should be a critical point only if the cutter and the machining surface are at the point P_M. And this can be determined by the distance from the fixed point E to the axis of the cutter.

As shown in Fig. 10, the normal vector of the checking surface at the point P_i is denoted as $\mathbf{n}_i$. And the point F_i is

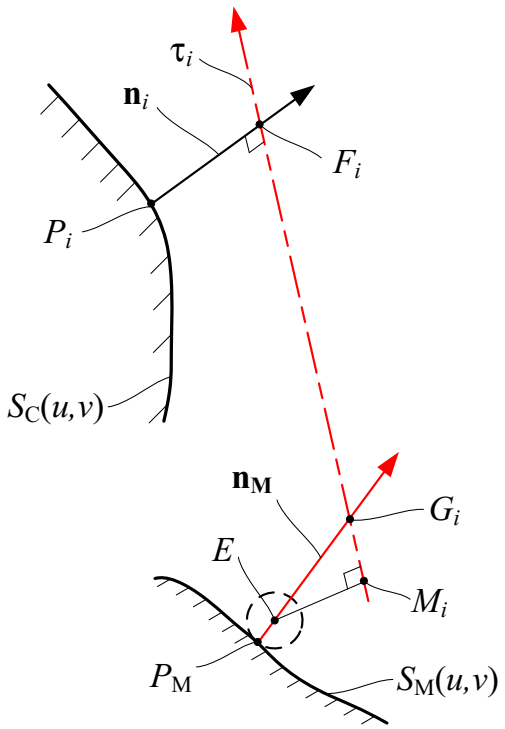

Fig. 10 The geometric relationship at the critical point

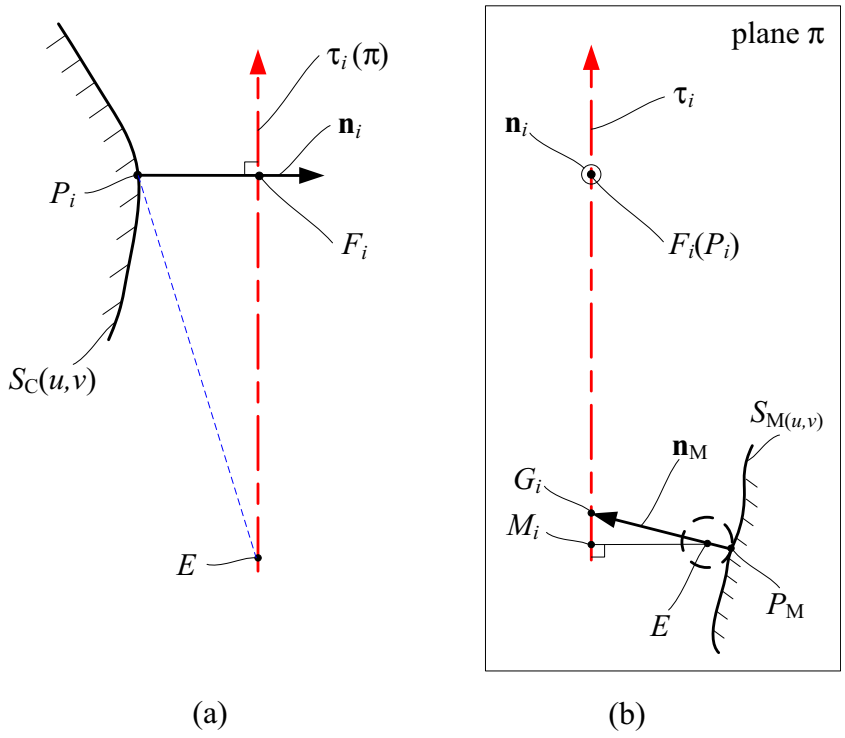

Fig. 11 The plane π passes through the point *E*

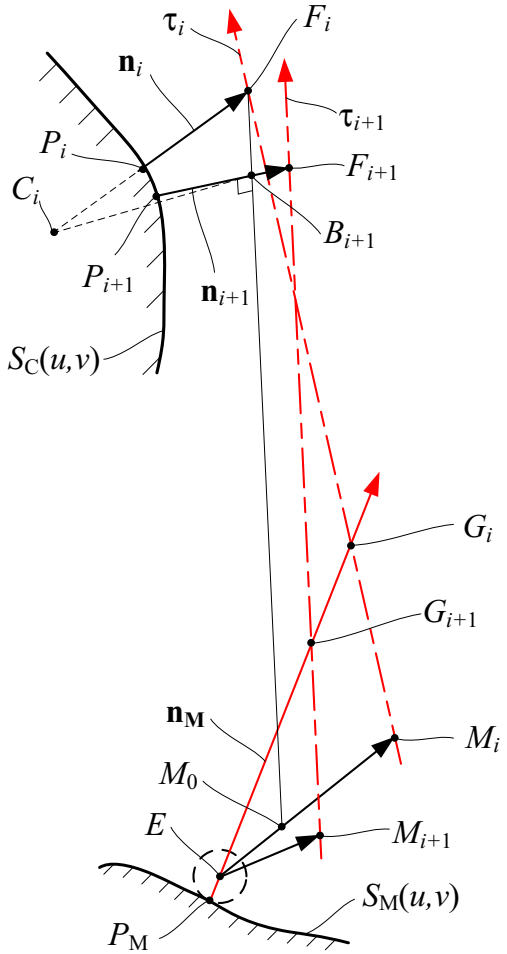

Fig. 12 Search critical point around the starting point

the intersection of the normal line passing through point P_i and the axis of cutter. It can be calculated by Eq. 11. Here, Δs is safety allowance.

$$F_i = P_i + (R + \Delta s)\cdot\mathbf{n}_i \tag{11}$$

In addition, the point G_i is the intersection of the normal line passing through point P_M and the axis of cutter. Then, there must be a plane, denoted by π, that goes through the point F_i and has a normal vector as $\mathbf{n}_i$. And the axis of cutter should be in the plane π. It is obvious that the distance between the point E and the axis of cutter is determined by the position of $P_{i,j}$ and the normal vector $\mathbf{n}_i$ on the checking surface. It should be equal to *R-r* if the cutter and the machining surface are tangent at the point P_M. Then, the corresponding critical vector $\boldsymbol{\tau}_i$ can be got with Eq. 12. Here, the point M_i is the perpendicular foot from point E to the axis of cutter.

$$\boldsymbol{\tau}_i = \frac{\overrightarrow{M_iF_i}}{|\overrightarrow{M_iF_i}|} \tag{12}$$

Based on the above, the method for determining critical point and calculating critical vector will be discussed below in terms of whether the vectors $\mathbf{n}_{i,j}$ and $\mathbf{n}_M$ are perpendicular or not.

(a) The vectors $\mathbf{n}_i$ and $\mathbf{n}_M$ are not perpendicular. In this case, the point where the normal line P_ME intersects the plane π is point G_i, because it is on the axis of the cutter, as

shown in Fig. 9. Therefore, the coordinates of the point G_i can be solved with Eq. 13. Here, a is non-zero.

$$\begin{cases} \mathbf{n}_i \cdot \overrightarrow{G_i F_i} = 0 \\ \overrightarrow{G_i F_i} = \overrightarrow{P_M F_i} - \overrightarrow{P_M G_i} \\ \overrightarrow{P_M G_i} = a \cdot \mathbf{n}_M \end{cases} \tag{13}$$

Since the line EM_i is perpendicular to the line F_iG_i, the coordinates of the points M_i can be solved with Eq. 14. Here, the parameter b is non-zero.

$$\begin{cases} \overrightarrow{F_i G_i} \cdot \overrightarrow{EM_i} = 0 \\ \overrightarrow{EM_i} = \overrightarrow{F_i E} - \overrightarrow{F_i M_i} \\ \overrightarrow{F_i M_i} = b \cdot \overrightarrow{F_i G_i} \end{cases} \tag{14}$$

The point P_i on the checking surface $\mathbf{S}_C(u,v)$ is the critical point only if the solution to Eq. 11 is $a > 0$ and the distance from the point E to the axis of cutter is equal to R-r, which makes the cutter tangent to the machining surface at point P_M.

(b) The vectors $\mathbf{n}_i$ and $\mathbf{n}_M$ are perpendicular. In this case, the vector $\mathbf{n}_M$ is parallel to the plane π. Then, the normal line P_ME can intersect the tool axis of the cutter which is in the plane π only if the plane π passes through the point E, as shown in a in Fig. 11. It can be identified with Eq. 15. Here, the point F_i can be got with Eq. 11.

$$\left|\overrightarrow{EP_i}\right|^2 = \left|\overrightarrow{P_{i,j}F_i}\right|^2 + \left|\overrightarrow{EF_i}\right|^2 \tag{15}$$

If the point P_i makes Eq. 15 true, there must be a point M_i on the axis of cutter in the plane π that makes the cutter tangent to the machining surface at the point P_M, as shown in b in Fig. 11. And the point M_i can be calculated with Eq. 16.

$$\begin{cases} \overrightarrow{EM_i} \cdot \overrightarrow{F_i M_i} = 0 \\ |\overrightarrow{EM_i}| = R - r \\ \overrightarrow{EM_i} \cdot \mathbf{n}_i = 0 \\ \overrightarrow{EM_i} \cdot \mathbf{n}_M \geq 0 \end{cases} \tag{16}$$

3.4 Search critical point

In this part, a critical point will be searched around the point P_i if it is not a critical point. In fact, even if it is not on the profile, the deviation is small, which is obvious from the search in Section 3.2. And a new point can be searched by a local adjustment which can make it true. As shown in Fig. 12, it is obvious that, when the position of P_i is adjusted, the normal vector $\mathbf{n}_i$ also changes accordingly. Then, the distance from the point E to the axis of cutter will change accordingly too. The method of searching the critical point is described as shown below.

Step 1 Calculate the average curvature at the point P_i on the checking surface $\boldsymbol{S}_C(u,v)$ and the radius of curvature R_{sp}. Then, the center C_i of the curvature circle can be found.

Step 2 Select a point M_0 on the line EM_i which makes the distance between the points M_0 and E is equal to R-r, where, if the solution from Eq. 12 is $a > 0$, the point M_0 will be taken in the positive direction of the vector $\mathbf{n}_M$. Otherwise, it will be taken in the opposite direction.

Step 3 Connect the points F_i and M_0 by the straight line F_iM_0. The vertical line to straight line F_iM_0 is made by passing the point C_i, and the foot is denoted by B_{i+1}. Then, the vertical line to the checking surface

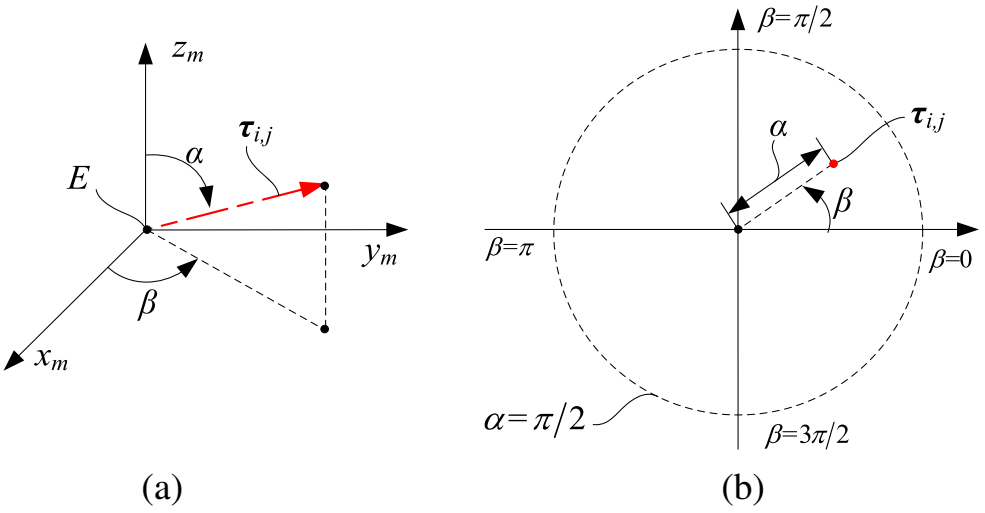

Fig. 13 Critical vector mapped into two-dimensions

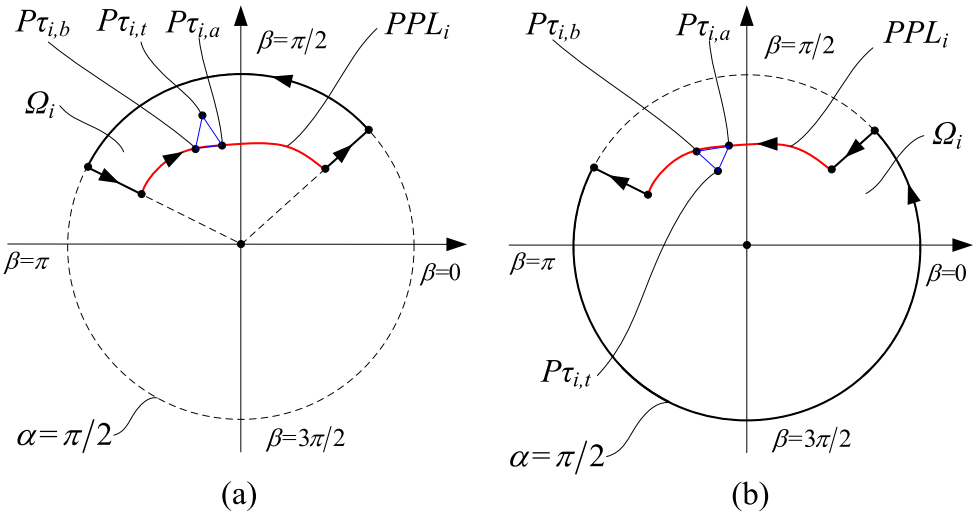

Fig. 14 Subinterval collision-free region in two-dimensions

$\boldsymbol{S}_C(u,v)$ is made by passing the point B_{i+1}, and the foot is denoted by P_{i+1}.

Step 4 Verify whether the point P_{i+1} is a critical point. If not, let $i = i + 1$ and turn to step 1 until P_{i+1} is a critical point.

However, it is difficult in the search to make the distance between the point E and the axis of the cutter is definitely equal to R-r. Of course, this is not necessary. In order to increase the robustness of the algorithm and improve the efficiency by reducing the number of iterations, it can be slightly modified and replaced by Eq. 17. Here, the introduced parameter μ is called the deviation which is a small positive real number and d is the distance between the point E and the axis of the cutter.

$$-\mu \le d-(R-r) \le \mu \tag{17}$$

Then, there will be a deviation ε between the cutter and the checking surface around the point $P_{i,j}$, which can be calculated with Eq. 18, where the angle between vectors $\mathbf{n}_{i,j}$ and the line $\overrightarrow{EM_i}$ is represented by $\left\langle \mathbf{n}_{i,j}, \overrightarrow{EM_i} \right\rangle$. And obviously, ε is not greater than μ. In order to avoid the cutter from interfering with the checking surface, the maximum deviation ε can be taken into account when designing the safety allowance Δs.

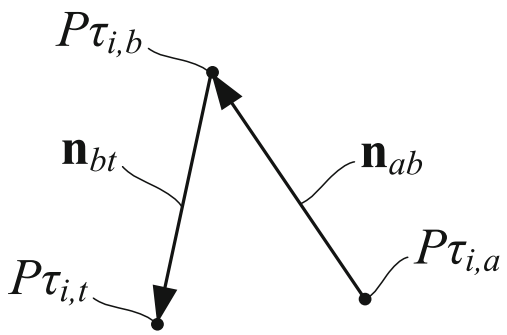

Fig. 15 The position relation between three points

$$\varepsilon = \mu \cdot \cos\left\langle \mathbf{n}_i, \overrightarrow{EM_i} \right\rangle \tag{18}$$

In addition, for the convenience of data processing in Section 4, it is necessary to find a tool posture which must be in the subinterval collision-free region. This special tool posture is represented by the identification vector $\tau_{i,t}$. It can be obtained in this way. First, a searched critical point roughly in the middle of the profile PL_i is selected as the starting point. Then, a special critical point which can be searched with the safety allowance Δs in Eq. 11 is replaced by a bigger one. And the corresponding critical vector is the right one.

4 Algorithm to solve collision-free regions

In this part, the searched sample critical points and corresponding vectors will be processed to construct the subinterval collision-free regions in two-dimensions. And the collision-free regions of the channel will be solved as the intersection of those subinterval regions.

4.1 Data processing

In order to describe the data processing process more clearly, the critical vector is uniformly expressed in terms of $\boldsymbol{\tau}_{i,j}$, whether the searched corresponding critical point is on the boundary or on the profile of checking surface. Those critical vectors are mapped into the polar coordinate system $E(\alpha,\beta)$. As shown in Fig. 13, α is the angle between the vector $\boldsymbol{\tau}_{i,j}$ and the z_m axis, and $0 \le \alpha \le \pi/2$. In addition, β is the angle between the projection of the vector $\boldsymbol{\tau}_{i,j}$ in the plane $x_m y_m$ and the x_m axis, and $0 \le \beta < 2\pi$. Correspondingly, the critical vector $\boldsymbol{\tau}_{i,j}$ in polar coordinate system $E(\alpha,\beta)$ becomes a point $P\boldsymbol{\tau}_{i,j}$.

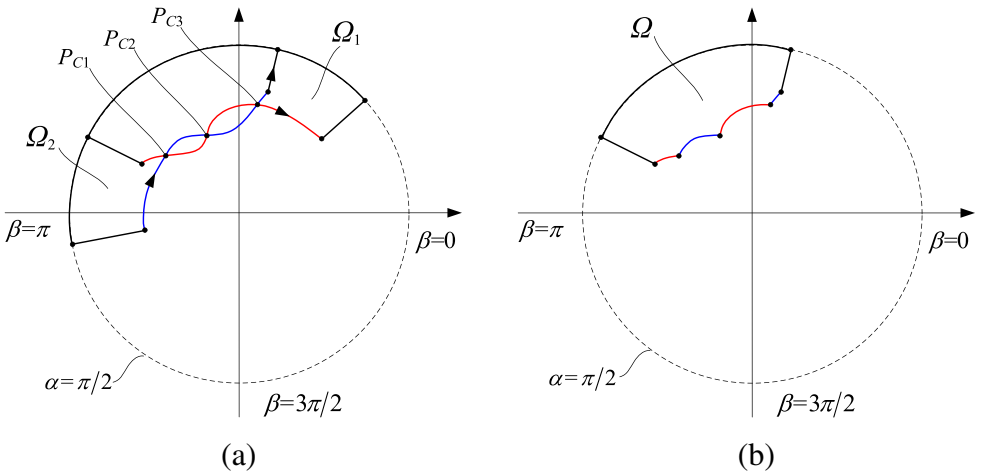

Fig. 16 The intersection of two subinterval collision-free regions

In the two-dimensions, the mapped points associated with the ith profile are connected into a curve PPL_i in the order. And the subinterval collision-free region Ω_i is the inner or outer region surrounded by the curve PPL_i if it is closed. Otherwise, there will be a closed area which is surrounded by the curve PPL_i together with the diameters at the two ends of the curve and the arc of α equal $\pi/2$, as shown in Fig. 14. The direction of counterclockwise walking along the boundaries of the region is defined as positive direction. Thus, the interior of the region is located to the left of the boundaries when walking along the positive direction.

However, any curve PPL_i just splits the area inside the circle of α equals to $\pi/2$ into two disjoint parts, as shown in Fig. 14. It needs to be distinguished by the position relation between a point $P\tau_{i,t}$ and the curve PPL_i. Here, the point $P\tau_{i,t}$ is called the identification point and it is mapped by the identification vector $\tau_{i,t}$, which has been covered in Section 3.4.

Among all points of the curve PPL_i, there are two points closest to the point $P\tau_{i,t}$ which are denoted as $P\tau_{i,a}$ and $P\tau_{i,b}$, respectively. The region Ω_i can be identified by the position of the point $P\tau_{i,t}$ relative to the points $P\tau_{i,a}$ and $P\tau_{i,b}$, as shown in Fig. 14. The region is determined as long as the sequence of passing through the points $P\tau_{i,a}$ and $P\tau_{i,b}$ is determined.

Two vectors $\mathbf{n}_{ab}$ and $\mathbf{n}_{bt}$ are constructed separately, as shown in Fig. 15, where the vector $\mathbf{n}_{ab}$ is going from point $P\tau_{i,a}$ to point $P\tau_{i,b}$ and vector $\mathbf{n}_{bt}$ is going from point $P\tau_{i,b}$ to point $P\tau_{i,t}$, assuming the sequence of the points $P\tau_{i,a}$ and $P\tau_{i,b}$ on the curve PPL_i can be determined by distinguishing the positions of vectors $\mathbf{n}_{ab}$ and $\mathbf{n}_{bt}$. If the solution of Eq. 19 is $D_t > 0$, the three points are arranged counterclockwise, as shown in Fig. 15, and the sequence of passing through the points $P\tau_{i,a}$ and $P\tau_{i,b}$ is determined. Here, the vectors $\mathbf{n}_{ab}$ and $\mathbf{n}_{bt}$ are represented as a matrix with one row and two columns, respectively.

$$D_t = [\mathbf{n}_{ab}, 0]^T \times [\mathbf{n}_{bt}, 0]^T \cdot [0, 0, 1]^T \tag{19}$$

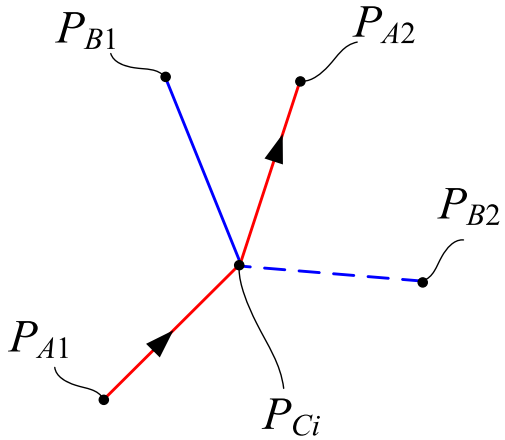

Fig. 17 The position of adjacent points at the intersection

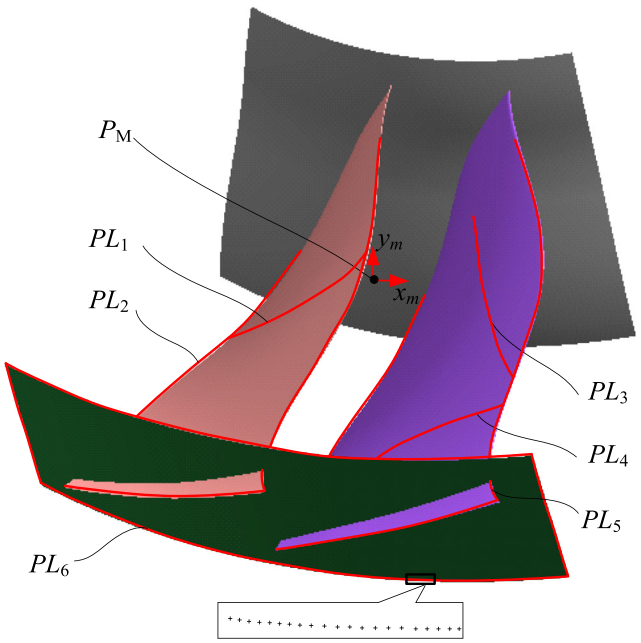

Fig. 18 Searched critical points in applications

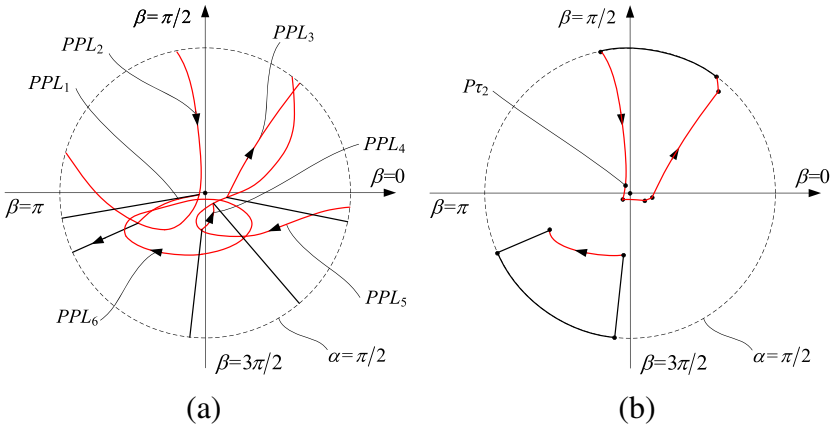

Fig. 19 Solved collision-free regions of closed blisk

4.2 Combine collision-free regions

The collision-free region of the channel can be solved with the intersection of the subinterval regions which have been constructed in section. The solution to the intersection is to add subinterval region Ω_i on the basis of Ω_1 one by one and update it.

First of all, calculate all the intersection points between the boundaries of regions Ω_1 and Ω_i. The ordered intersection is denoted by P_{Ci}, as shown in a in Fig. 16. Then, starting from any intersection point, the boundary of region Ω_i can be pruned by intersection along the positive direction of the boundary of region Ω_1. Judge whether the boundary of region Ω_i between two intersections is inside the region Ω_1. If not, it will be removed. Following the same procedure, the boundaries of region Ω_1 will also be trimmed. Finally, the region of the remaining curve is the common region of Ω_1 and Ω_i, as shown in b in Fig. 16. Where, Ω represents the intersection of two subinterval regions.

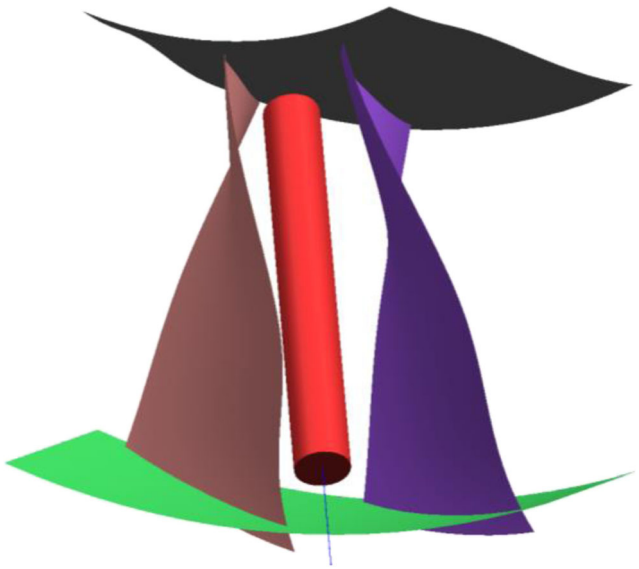

Fig. 20 Cutting simulation in UG/NX

There are two keys to finding the intersection of two regions. First, calculate the intersection of the boundaries of two regions. This can be done with mature commercial software, which is not described here. Second, determine whether the boundary between the two intersections needs to be deleted. This part gives a brief account of that.

On the boundary of region Ω_1, there are two points adjacent to the intersection P_{Ci} denoted as P_{A1} and P_{A2} respectively, as shown in Fig. 17. And on the boundary of region Ω_i, the two points adjacent to the intersection P_{Ci} are denoted as P_{B1} and P_{B2} respectively. If the point P_{B1} is to the left of the points P_{A1} and P_{Ci} or to the left of the points P_{A2} and P_{Ci}, the boundary of the region Ω_1 that contains the point P_{B1} will be preserved. Otherwise, it should be removed. The judgment method has been introduced in detail in Section 4.1.

5 Algorithm verification

5.1 Applications

The proposed algorithm is applied to calculate the collision-free regions for one channel of closed blisk. The blisk is 95.5 mm high with 24 channels. The diameter of the outer hub is 585.5 mm, while the diameter of the inner hub varies from 290.3 to 370.6 mm. The cutter in applications is a filleted end mill with $R = 8$ mm and $r = 2$ mm. The suction surfaces, pressure surfaces, and part of hub surfaces are related to the channel of the blisk. The milling point P_M is on the inner hub surface in the channel and is close to the side of pressure surface, where the constraint is more serious.

The safety allowance Δs is set to be 0.5 mm. At the same time, the small positive quantity μ in Eq. 17 is set to be 0.01 mm in these applications. And the pre-set discrete accuracy θ_δ is set to be 0.02°. The searched critical points are imported into the CAD/CAM software UG/NX to show them in Fig. 18.

In this application verification, only part of the hub surfaces related to the channel is selected. The tangent plane at point

Table 1 Results of the proposed method

		Points (num.)		Running time (s)		
		Inside surfaces	On boundaries	Searching points	Combining regions	Total
Suction surface	PL_1	1956	/	0.57	/	/
	PL_2	/	3612	0.89	/	/
Pressure surface	PL_3	1723	/	0.53	/	/
	PL_4	1842	/	0.61	/	/
	PL_5	/	2546	0.67	/	/
Outer hub	PL_6	/	3779	0.96	/	/
Total		5521	9937	4.23	0.22	31.68

P_M is represented by π_M, where the inner hub surface is below the plane π_M at the milling point P_M and there is no critical point. And part of pressure surface and suction surface are below the normal plane π_M. The searched profiles are marked with notations PL_1–PL_6 respectively.

The searched critical points are then mapped into $P_{CL}(\alpha,\beta)$ polar coordinate system and drawn in a in Fig. 19. Here, profiles PL_1–PL_6 in Fig. 18 are replaced by PPL_1–PPL_6 in a in Fig. 19. The final collision-free regions solved with the proposed method are shown in b in Fig. 19. There are two collision-free regions for the closed blisk. This is due to the constraint of the outer hub surface on the vector.

A point, which is denoted by $P\tau_2$ is selected at the boundaries of the collision-free regions, as shown in b in Fig. 19, and its corresponding vector is taken as the direction of the cutter. Then, the cutting simulation is carried out in the CAD/CAM software UG/NX and the result is shown as Fig. 20. Here, the red part is the tool used for application verification. It can be seen that the cutter is very close to the boundary of pressure surface and there is no interference. And the position of the nearest point is the same as expected. Here, the distance between the cutter and the checking surface is safety allowance Δs, which is set to be 0.5 mm.

5.2 Analysis

The proposed method is implemented in MATLAB R2014a in a 64-bit operating system with an Intel(R) Core(TM) i5-3320M CPU with 2.60-GHz processor. The total running time is 31.68 s, and the detailed results are shown in Table 1. The algorithm to search critical points is the key part in the proposed method. In the example, 15,458 points are searched in 4.23 s, which is about 13.35% of the total running time. From the results, the most time-consuming part is the algorithm to find and calculate intersection points for all segments. In general, the time taken to compute the intersections increases rapidly as the number of critical points increases. What need to be noticed is that it can be shortened because, by enlarging the distance accuracy μ. the frequencies in the process of searching are reduced. And there will be no interference.

The calculation errors are evaluated both in distance error and angle error. One thousand pairs of random points are selected to calculate the distance error and angle error. Firstly, the distance error means that the deviation between the distance from a critical point to the axis of the cutter and the theoretical distance. The mean error, maximum error, and standard deviation are 3.8×10^{-3}, 9.5×10^{-3}, and 1.2×10^{-3} mm, respectively. Secondly, the angle error means the angle between the adjacent critical vectors. The mean error, maximum error, and standard deviation are 0.19, 0.02, and 0.009°, respectively.

5.3 Comparisons

In this part, the method proposed in this paper will be compared with the algorithm in literature [4]. However, the latter is mainly applicable to ball-end mill and hard to deal with the filleted end mill. So, the ball-end cutter with the same diameter $R = 8$ is selected for comparison in the method proposed in the reference [4]. Under the condition of the roughly equal number of critical points, the results of the comparison are shown in Table 2. Here, the consistency is the ratio of the minimum and maximum angle between two adjacent critical vectors.

Table 2 Comparisons of two methods

	Points (num.)	Searching time(s)	Search mode	Tolerance	Allowance Δs	Mean error	Consistency
Reference	16,105	3.82	$L_S = 0.1$ mm	0.001°	/	0.000656°	15.5%
This paper	15,458	4.23	$\theta_\delta = 0.02°$	0.01 mm	0.5 mm	0.0035 mm	94.7%

From the comparison results, the efficiency of the two methods is basically equal. For the same checking surfaces, the number of critical points searched in unit time in this paper is about 87% of the algorithm in the literature [4]. However, the consistency of the proposed method is significantly better than that of the method in the reference.

The angle between the searched adjacent critical vectors is always checked and adjusted in the presented algorithm. And it is the reason for the better consistency. However, the constant step length is used to search the adjacent critical points on the boundaries and the profiles of checking surface in literature [4]. This saves some computing resources. But the price is relatively poor consistency, which will be more significant when the point P_M is close to the critical points.

6 Conclusion and outlook

A detailed process of generating the collision-free regions of tool postures in five-axis milling of blisk is presented in this paper. To avoid the interference collision between the filleted end mill and checking surface, the geometric relationships between the cutter and the checking surface and the machining surface in the critical state are discussed. The main contributions of this work can be summarized as follows.

(a) A tool-surface tangent model is established. With the model, the problem that the axis of the filleted end cutter does not rotate around a fixed point when search the critical point by adjusting the tool posture can be solved.
(b) An algorithm to search critical points associated with the boundaries of collision-free regions with a self-adapting step length is proposed.
(c) The angle between the adjacent critical vectors is taken to control the distribution of critical points. So that the consistency of mapping points in two-dimensional space can be improved.

The proposed method has been verified to be efficient and accurate. The future work will be dedicated to generalize the method to be applicable to the cutter of any shape.

Funding information The research was supported by the National Science Foundation of China (No. 51675439).

References

1. Lasemi A, Xue DY, Gu PH (2010) Recent development in CNC machining of freeform surfaces: a state-of-the-art review. Comput Aided Des 42(7):641–654
2. Tang TD (2014) Algorithms for collision detection and avoidance for five-axis NC machining: a state of the art review. Comput Aided Des 51(1):1–17
3. Chen KH (2011) Investigation of tool orientation for milling blade of impeller in five-axis machining. Int J Adv Manuf Technol 52(1):235–244
4. Liang YS, Zhang DH, Ren JX, Chen ZC, Xu YY (2016) Collision-free regions of tool orientations in multi-axis milling of blisks with a ball-end mill. Int J Adv Manuf Technol 85(5–8):1887–1900
5. Cai YL, Zhang FL, Xi XL (2019) Cutter orientation planning in NC machining for surface similar to revolution body with considering kinematic characteristics. Int J Adv Manuf Technol 100(1–4):503–513
6. Huang J, Du X, Zhu LM (2018) Real-time local smoothing for five-axis linear toolpath considering smoothing error constraints. Int J Mach Tools Manuf 124:67–79
7. Jun CS, Cha K, Lee YS (2003) Optimizing tool orientations for 5-axis machining by configuration-space search method. Comput Aided Des 35(6):549–566
8. Chen ZT, Li SS, Gan ZW, Zhu Y (2017) A highly efficient and convergent optimization method for multipoint tool orientation in five-axis machining. Int J Adv Manuf Technol 93(5–8):2711–2722
9. Wu BH, Liang MC, Zhang Y, Luo M, Tang K (2018) Optimization of machining strip width using effective cutting shape of flat-end cutter for five-axis free-form surface machining. Int J Adv Manuf Technol 94(5–8):2623–2633
10. Luo M, Yan DQ, Wu BH, Zhang DH (2016) Barrel cutter design and toolpath planning for high-efficiency machining of freeform surface. Int J Adv Manuf Technol 85(9–12):2495–2503
11. Tang TD, Bohez ELJ (2015) A new collision avoidance strategy and its integration with collision detection for five-axis NC machining. Int J Adv Manuf Technol 81(5–8):1247–1258
12. Li XY, Lee CH, Hu PC, Zhang Y, Yang FZ (2018) Cutter partition-based tool orientation optimization for gouge avoidance in five-axis machining. Int J Adv Manuf Technol 95(5–8):2041–2057
13. Liu W, Zhang JW, Zhu SM, Zhang CC, Yuan TJ (2017) Efficient tool posture global collision-free area generation for 5-axis point clouds machining. Int J Adv Manuf Technol 88(1–4):1013–1023
14. Hu PC, Tang K, Lee CH (2013) Global obstacle avoidance and minimum workpiece setups in five-axis machining. Comput Aided Des 45(10):1222–1237
15. Liang YS, Zhang DH, Chen ZC, Ren JX, Li XY (2014) Tool orientation optimization and location determination for four-axis plunge milling of open blisks. Int J Adv Manuf Technol 70(9):2249–2261
16. Wu BH, Zhang DH, Luo M, Zhang Y (2013) Collision and interference correction for impeller machining with non-orthogonal four-axis machine tool. Int J Adv Manuf Technol 68(1):693–700
17. Chen L, Xu K, Tang K (2015) Collision-free tool orientation optimization in five-axis machining of bladed disk. J Comput Des Eng 2(4):197–205
18. Lin ZW, Shen HY, Gan WF, Fu JZ (2012) Approximate tool posture collision-free area generation for five-axis CNC finishing process using admissible area interpolation. Int J Adv Manuf Technol 62(9–12):1191–1203
19. Bi QZ, Wang YH, Ding H (2010) A GPU-based algorithm for generating collision-free and orientation-smooth five-axis finishing tool paths of a ball-end cutter. Int J Prod Res 48(4):1105–1124
20. Wang QH, Li JR, Zhou RR (2006) Graphics-assisted approachto rapid collision detection for multi-axis machining. Int J Adv Manuf Technol 30(9):853–863
21. Tang TD, Bohez ELJ, Koomsap P (2007) The sweep plane algorithm for global collision detection with workpiece geometry update for five-axis NC machining. Comput Aided Des 39(11):1012–1024