Science in China Series E: Technological Sciences

www.scichina.com
tech.scichina.com
www.springerlink.com

Global optimization of tool path for five-axis flank milling with a cylindrical cutter

DING Han[1†] & ZHU LiMin[2]

[1] State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China;
[2] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

In this paper, optimum positioning of cylindrical cutter for five-axis flank milling of non-developable ruled surface is addressed from the perspective of surface approximation. Based on the developed *interchangeability principle*, global optimization of the five-axis tool path is modeled as approximation of the tool envelope surface to the data points on the design surface following the minimum zone criterion recommended by ANSI and ISO standards for tolerance evaluation. By using the signed point-to-surface distance function, tool path plannings for semi-finish and finish millings are formulated as two constrained optimization problems in a unified framework. Based on the second order Taylor approximation of the distance function, a sequential approximation algorithm along with a hierarchical algorithmic structure is developed for the optimization. Numerical examples are presented to confirm the validity of the proposed approach.

five-axis flank milling, tool path optimization, tool axis trajectory surface, surface approximation, distance function, minimax optimization

1 Introduction

Ruled surface is widely used in industry. For slender parts, like turbine blades and impellers, free-form surfaces are usually approximated by ruled surfaces. Flank milling can be effectively employed to machine ruled surface for the advantage of larger material removal rate as compared with point milling.

Recently, increasing attention has been drawn onto the problem of optimum positioning cutter for flank milling ruled surfaces. Liu[1] proposed a double point offset (DPO) strategy. The main idea was to offset two specific points on the ruling along the surface normals by the radius of the cutting tool, then obtain the cutter axis location by joining the two offset points. Bohez et al.[2] suggested an approach that positioned the cutter tangent to a point on the ruling such that the angles between the surface normal at that point and the surface normals at the two end points of the ruling were equal. To reduce the tool overcut, the cutter axis could be moved away from the surface along the normal at that point until the cutter was approximately tangent to the two guiding rails. Lee and Suh[3] developed a strategy to adjust the cutter orientation and the offset distance so as to minimize the sum of the distances from some points of the cutter axis to the ruled surface. Rubio et al.[4] presented a standard cutter positioning method, in which the tool axis was chosen to be parallel to the ruling and a point on the axis was calculated such that the interference errors on the two ends of the ruling were equal. Redonnet et al.[5] proposed an improved method to position a cylindrical cutter tangent to the ruled surface at three points: two points on two directrices and one point on a ruling. Bedi et al.[6] proposed an approach to slide

Received November 18, 2008; accepted February 4, 2009
doi: 10.1007/s11431-009-0168-3
†Corresponding author (email: mehanding@gmail.com)
Supported by the National Natural Science Foundation of China (Grant Nos. 50775147 and 50835004), the National Basic Research Program of China ("973" Project) (Grant No. 2005CB724103), and the Science & Technology Commission of Shanghai Municipality (Grant No. 07JC14028)

the cutter along two directrices, keeping the cutter tangent to both curves at every parameter value. The computational efficiency of this method was higher than that of Redonnet's method, but the accuracy became lower. In a subsequent work[7], an improved three-step optimization method was developed. Tsay and Her[8] derived an analytical description of the overcut error in the plane perpendicular to the ruling, and then determined the tool axis location by using statistical analysis to minimize this error. Gong et al.[9] proposed a three points offset (TPO) method to determine the initial location of a cylindrical cutter. They selected three curves on the offset surface of the designed surface firstly, and then slid the cutter axis along two of the curves until the minimum distance between the third curve and the tool axis was less than a threshold value. Chiou[10] described a swept envelope-based method for tool positioning. The initial cutter positions were located to contact with two directrices. Then the swept profile of the cutter was calculated based on the cutter motion. Finally, the cutter locations were adjusted to reduce the machining errors by comparing the swept profile with the designed ruled surface. Recently, Ding et al.[11] discussed cutter location planning from the viewpoint of Chebyshev fitting of spatial straight line.

The aforementioned works focused on the individual cutter location planning, however, the tool path optimization from a global perspective has been addressed little except for ref. [9]. Lartigue et al.[12] proposed a global tool path planning method. The basic idea was to deform the two curves that defined the tool trajectory so that the tool envelope surface fitted the design surface as much as possible. The geometric deviation between the two surfaces was evaluated by the sum of the squared distances of the points on the design surface to the envelope surface. To simplify the computation, an approximate distance measure was employed. Based on the individual cutter locations provided by the TPO method, Gong et al.[9] adopted the lofting method to get an initial tool axis trajectory surface, then deformed it to approximate the offset surface of the designed surface by the method of least squares (LS) surface fitting. In Gong's study, the envelop surface of the cutter was not concerned since the *error propagation principle* proved there showed that the extremum distance between the envelop surface of the cylindrical cutter and the designed surface was equal to that between the tool axis trajectory surface and the offset surface of the designed surface. Although the LS method was easy for implementation and efficient in computation, it could not incorporate readily the non-overcut constraint required by semi-finish milling, and more importantly it did not conform to the minimum zone criterion recommended by ANSI and ISO standards for tolerance evaluation[13,14], which requires the maximum norm of the error vector be minimized. Furthermore, the geometric deviation of the machined surface from the nominal one was not clearly defined and the influence of the deformation of the tool axis trajectory surface on the change of this deviation was not quantitatively analyzed.

In this paper, we introduce the maximum orthogonal distance from the point on the design surface to the tool envelope surface to characterize the geometric error of the machined surface, and then develop the complete principle, model and algorithm for global tool path planning based on the properties of the distance function. The remainder of this paper is organized as follows. In section 2, the signed point-to-surface distance function is defined, and its properties are investigated. Especially, its second order Taylor approximant is derived. In section 3, the geometric deviation between two surfaces is defined by using the distance function, and the *interchangeability principle* is proposed, which establishes the mathematical ground for modeling the problem of tool path planning from the perspective of surface approximation. In section 4, global tool path optimizations for semi-finish and finish flank millings are formulated as two constrained optimization problems in a unified framework. Based on the differential properties of the distance function, a sequential approximation algorithm along with a hierarchical algorithmic structure is developed for the optimization. Numerical examples are presented in section 5 to confirm the validity of the proposed approach, and conclusions are given in section 6.

2 Distance function and its differential properties

In this section, the signed point-to-surface distance function is introduced to characterize the geometric deviation of a surface from the nominal one.

Definition 1. Given a regular surface $\boldsymbol{S}(\boldsymbol{w}) \in \mathbf{R}^3$, where $\boldsymbol{w} = [w_1, \cdots, w_m]^{\mathrm{T}} \in \mathbf{R}^m$ denotes the collection of the shape parameters, and a point $\boldsymbol{p} \in \mathbf{R}^3$, there exists

at least one closest point $\boldsymbol{q} \in \boldsymbol{S}(\boldsymbol{w})$, termed foot point, such that $\|\boldsymbol{p}-\boldsymbol{q}\| = \min_{\boldsymbol{x}\in\boldsymbol{S}(\boldsymbol{w})} \|\boldsymbol{p}-\boldsymbol{x}\|$, where $\|\cdot\|$ stands for the Euclidean norm on $\mathbf{R}^3$. The distance function is defined as $d_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w}) = \|\boldsymbol{p}-\boldsymbol{q}\|$.

For most engineering surfaces such as planes, cylinders, cones and spheres, the distance function can be explicitly computed, while for complex algebraic surfaces and parametric sculptured surfaces such as Bézier, B-spline and NURBS surfaces, it must be computed by an iterative approach. The procedure of finding the foot point on a parametric surface given a point in space is addressed in many texts on curve and surface mathematics.

Proposition 1[15]**.** If $\boldsymbol{q}$ is not on the boundary of $\boldsymbol{S}(\boldsymbol{w})$, the error vector $\boldsymbol{p}-\boldsymbol{q}$ is normal to $\boldsymbol{S}(\boldsymbol{w})$, i.e., $\boldsymbol{p}-\boldsymbol{q} = \pm d_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\boldsymbol{n}^{\boldsymbol{q}}$, where $\boldsymbol{n}^{\boldsymbol{q}}$ is the unit outward normal vector of surface $\boldsymbol{S}(\boldsymbol{w})$ at point $\boldsymbol{q}$. The choice of plus or minus sign depends on the direction of $\boldsymbol{n}^{\boldsymbol{q}}$.

Based on this proposition, we can define the following signed distance function.

Definition 2. If $\boldsymbol{q}$ is unique and not on the boundary of $\boldsymbol{S}(\boldsymbol{w})$, the signed distance function is defined as $d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w}) = (\boldsymbol{p}-\boldsymbol{q})\cdot\boldsymbol{n}^{\boldsymbol{q}}$, where the bold dot denotes scalar product.

Obviously, the absolute value of the signed distance function yields the distance function. As shown in Figure 1, the signed distance between the point and surface is positive if the point lies in the outer side of the surface, and negative if the inner side of the surface. It is worth noting that ambiguity exists where there are two or more foot points such as when $\boldsymbol{S}(\boldsymbol{w})$ represents a spherical patch and $\boldsymbol{p}$ is located in the center of the sphere. Special cases like this are ignored because they hardly occur, especially when $\boldsymbol{p}$ is close to $\boldsymbol{S}(\boldsymbol{w})$.

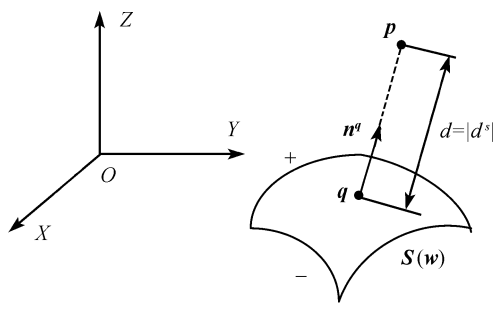

Figure 1 Point-to-surface distance function.

The signed distance function has the following differential properties.

Proposition 2[15]**.** If surface $\boldsymbol{S}(\boldsymbol{w})$ has a locally parametric representation $\boldsymbol{\psi}(\boldsymbol{w},u,v)$ and $\boldsymbol{q}=\boldsymbol{\psi}(\boldsymbol{w},u^*,v^*)$, then the gradient vector of $d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})$ is

$$\nabla d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w}) = -\left[\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{w_m}\right]^{\mathrm{T}}, \tag{1}$$

where the partial derivatives $\boldsymbol{\psi}_{w_i}, i=1,\cdots,m$ are evaluated at $(\boldsymbol{w},u^*,v^*)$.

Proposition 3. If matrix $\boldsymbol{A} \triangleq d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\boldsymbol{\Omega}-\boldsymbol{g}$ is invertible, where $\boldsymbol{g}$ and $\boldsymbol{\Omega}$ are the first and second fundamental matrices of surface $\boldsymbol{S}(\boldsymbol{w})$ at point $\boldsymbol{q}$, respectively, then $d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})$ is second-order differentiable and the i-th row and j-th column element of its Hessian matrix $\nabla^2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})$ is

$$\begin{aligned}
&\left[\nabla^2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\right]_{ij}\\
&=\left[\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_i},\boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_i}\right]\boldsymbol{g}^{-1}\boldsymbol{\Omega}\boldsymbol{A}^{-1}\left[\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_j},\boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_j}\right]^{\mathrm{T}}\\
&\quad - d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\left[\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_i},\boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_i}\right]\boldsymbol{g}^{-1}\boldsymbol{\Omega}\boldsymbol{A}^{-1}\\
&\quad\cdot\left[\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{uw_j},\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{vw_j}\right]^{\mathrm{T}} - \boldsymbol{n}^{\boldsymbol{q}}_{w_j}\cdot\boldsymbol{\psi}_{w_i}\\
&\quad-\left[\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{w_iu},\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{w_iv}\right]\boldsymbol{A}^{-1}\left[\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_j},\boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_j}\right]^{\mathrm{T}}\\
&\quad+ d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\left[\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{w_iu},\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{w_iv}\right]\boldsymbol{A}^{-1}\\
&\quad\cdot\left[\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{uw_j},\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{vw_j}\right]^{\mathrm{T}} - \boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{w_iw_j},
\end{aligned} \tag{2}$$

where all the partial derivatives are evaluated at $(\boldsymbol{w},u^*,v^*)$.

The proof is given in Appendix A. The determinant of matrix $\boldsymbol{A}$ reads

$$|\boldsymbol{A}| = |\boldsymbol{g}|\left(\kappa_1 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})-1\right)\left(\kappa_2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})-1\right), \tag{3}$$

where κ_1 and κ_2 are the principal curvatures at point $\boldsymbol{q}$. It shows that matrix $\boldsymbol{A}$ can become singular when point $\boldsymbol{p}$ is at the center of radius of one of the principal curvatures of point $\boldsymbol{q}$ on surface $\boldsymbol{S}(\boldsymbol{w})$. In practical computation, such singular cases seldom occur, especially when $\boldsymbol{p}$ is close to $\boldsymbol{S}(\boldsymbol{w})$.

The Hessian matrix $\nabla^2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})$ is complex in form and computationally expensive. We will make a simplification by neglecting all the terms involving the second-order partial derivatives of the function $\boldsymbol{\psi}(\boldsymbol{w},u,v)$.

This results in an approximate Hessian matrix

$$\tilde{\nabla}^2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w}) = \left[\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_{w_m}\right]^{\mathrm{T}}\left[\boldsymbol{\psi}_u,\boldsymbol{\psi}_v\right]\boldsymbol{g}^{-1} \cdot \boldsymbol{\Omega}\boldsymbol{A}^{-1}\left[\boldsymbol{\psi}_u,\boldsymbol{\psi}_v\right]^{\mathrm{T}}\left[\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_{w_m}\right]. \quad (4)$$

It is well-known that in a neighborhood of a point on a regular surface, there always exist orthogonally parametric nets; especially for a nonumbilical point, an orthogonally parametric net formed by the lines of curvature can be found. So, we can always get two families of coordinate curves on surface $\boldsymbol{S}(\boldsymbol{w})$, of which the tangents at point $\boldsymbol{q}$ give the two mutually orthogonal principal directions (Note that at an umbilical point, each tangent direction is a principal direction). In this case, both $\boldsymbol{g}$ and $\boldsymbol{\Omega}$ become diagonal matrices, and we obtain a simple expression of the approximate Hessian matrix $\tilde{\nabla}^2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})$ as shown in the following proposition.

Proposition 4. If $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ are the two mutually orthogonal unit vectors that determine the two principal directions associated with the two principal curvatures κ_1 and κ_2 at $\boldsymbol{q}$, respectively, then the approximate Hessian matrix $\tilde{\nabla}^2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})$ has the form

$$\tilde{\nabla}^2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w}) = \sum_{i=1}^{2}\frac{1}{d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w}) - 1/\kappa_i}\left[\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_{w_m}\right]^{\mathrm{T}}\boldsymbol{n}_i\boldsymbol{n}_i^{\mathrm{T}}\left[\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_{w_m}\right]. \quad (5)$$

The proof appears in Appendix B.

3 Interchangeability principle

A problem often encountered in the field of CAD/CAM is to evaluate the deviation between two surfaces $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. An example is to evaluate the deviation between the machined surface and the designed surface. According to the tolerance definition recommended by ANSI and ISO standards, if surface $\boldsymbol{S}_1$ is selected as the datum surface, the deviation between the two surfaces is defined as

$$e_1 = \max_{\boldsymbol{p}\in\boldsymbol{S}_2} d_{\boldsymbol{p},\boldsymbol{S}_1}. \quad (6)$$

On the contrary, if surface $\boldsymbol{S}_2$ is selected as the datum surface, the deviation between the two surfaces is defined as

$$e_2 = \max_{\boldsymbol{p}\in\boldsymbol{S}_1} d_{\boldsymbol{p},\boldsymbol{S}_2}. \quad (7)$$

Using the optimality condition for an unconstrained optimization problem, we find that under certain condition the evaluation result is independent of the selection of the datum surface, which is stated as the *interchangeablility principle*. Figure 2 gives an illustration and the complete proof is given in Appendix C.

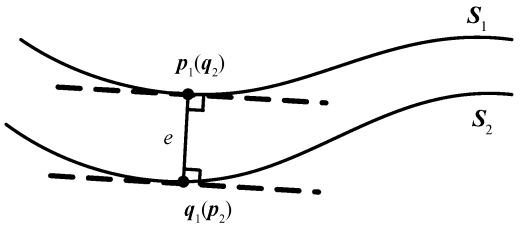

Figure 2 Normal vectors at the two points that provide the maximum distance.

Proposition 5 (Interchangeability Principle). If the two points corresponding to the maximum distance $\boldsymbol{e}_1$ are not on the boundaries of the two surfaces, they also yield the maximum distance $\boldsymbol{e}_2$ and the normals at them are collinear with the line that connects them.

NC machining is a process that subtracts the swept volumes generated by the cutter moving along the programmed tool paths from the current raw stock. Since the swept volume is enclosed by the swept envelope, which represents the set of points on the moving cutter that also lie on the machined surface, from the viewpoint of geometric simulation, the envelope surface of the cutter can be treated as the machined surface. Obviously, we hope that the machined surface $\boldsymbol{S}_{\text{envelope}}$ approximates to the designed surface $\boldsymbol{S}_{\text{design}}$ as much as possible. It is difficult to compare two continuous surfaces directly to obtain the deviation between them. Usually the maximum distance from the point cloud on the non-datum surface to the datum one offers a measure for the deviation. To compute the geometric error of the machined surface $\boldsymbol{S}_{\text{envelope}}$, the designed surface $\boldsymbol{S}_{\text{design}}$ should be selected as the datum surface. Since the surface $\boldsymbol{S}_{\text{envelope}}$ is deformed during the tool path optimization, it is not convenient to discretize it. According to the *interchangeability principle*, we can select instead the machined surface $\boldsymbol{S}_{\text{envelope}}$ as the datum surface, and sample a dense set of points from the designed surface $\boldsymbol{S}_{\text{design}}$.

4 Model and algorithm for global tool path optimization

Given a surface $\boldsymbol{S}$, its offset surface $\boldsymbol{S}_{\text{offset}}$ is the surface with constant offset to it. Obviously, for a cylindrical tool, the tool axis trajectory surface $\boldsymbol{S}_{\text{axis}}$ is the offset

surface of the tool envelope surface $\boldsymbol{S}_{\text{envelope}}$, and the offset distance is the tool radius r. As a result, we have

$$d^s_{\boldsymbol{p},\boldsymbol{S}_{\text{envelope}}} = d^s_{\boldsymbol{p},\boldsymbol{S}_{\text{axis}}} - r \tag{8}$$

for a point $\boldsymbol{p}$ on the design surface. Here, it is assumed that the normal vector of the tool axis trajectory surface $\boldsymbol{S}_{\text{axis}}$ points to the design surface $\boldsymbol{S}_{\text{design}}$. Evidently, $d^s_{\boldsymbol{p},\boldsymbol{S}_{\text{envelope}}}$ and $d^s_{\boldsymbol{p},\boldsymbol{S}_{\text{axis}}}$ have the same gradient vector and Hessian matrix.

The tool axis trajectory surface $\boldsymbol{S}_{\text{axis}}$ is a ruled surface. As shown in Figure 3, it can be described by two B-spline curves as follows:

$$\boldsymbol{P}(u) = \sum_{i=0}^{l} N_{i,k}(u)\boldsymbol{b}_i,\ \boldsymbol{Q}(u) = \sum_{i=0}^{l} N_{i,k}(u)\boldsymbol{d}_i, \tag{9}$$

$$\boldsymbol{S}_{\text{axis}}(\boldsymbol{w}):\ \boldsymbol{\psi}(\boldsymbol{w},u,v) = v\boldsymbol{P}(u) + (1-v)\boldsymbol{Q}(u), \tag{10}$$

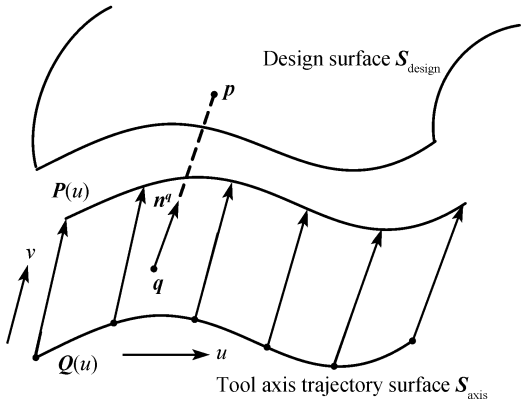

Figure 3 Tool axis trajectory surface represented as a ruled surface.

where $\boldsymbol{w}^{\mathrm{T}} = \left[\boldsymbol{b}_0^{\mathrm{T}},\cdots,\boldsymbol{b}_l^{\mathrm{T}},\boldsymbol{d}_0^{\mathrm{T}},\cdots,\boldsymbol{d}_l^{\mathrm{T}}\right] \in \mathbf{R}^{6(l+1)}$, and $\boldsymbol{b}_0,\cdots,\boldsymbol{b}_l$ and $\boldsymbol{d}_0,\cdots,\boldsymbol{d}_l$ are the control points of the two B-spline curves, respectively, which are treated as the shape parameters of surface $\boldsymbol{S}_{\text{axis}}$. As stated above, global optimization of the tool path for five-axis flank milling requires to approximate the envelope surface of the tool $\boldsymbol{S}_{\text{envelope}}$ to the point cloud on the design surface $\boldsymbol{S}_{\text{design}}$ following the minimum zone criterion, which states that the deviation measure e defined in eq. (6) should be minimized. For a dense set of data points $\{\boldsymbol{p}_i \in \mathbf{R}^3,\ 1 \leqslant i \leqslant n\}$ sampled from $\boldsymbol{S}_{\text{design}}$, this leads to the following minimax problem, or the Chebyshev approximation problem

$$\textbf{P1}\qquad \min_{\boldsymbol{w}\in\mathbf{R}^{6(l+1)}}\ \max_{1\leqslant i\leqslant n}\left|d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}) - r\right|. \tag{11}$$

By introducing one extra variable ξ, problem **P1** can be reformulated as the following differentiable constrained optimization problem:

$$\textbf{P2}\qquad \begin{aligned}&\min_{(\boldsymbol{w},\xi)\in\mathbf{R}^{6(l+1)+1}}\ \xi\\ &\text{s.t.}\quad -\xi \leqslant d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}) - r \leqslant \xi,\ 1\leqslant i\leqslant n.\end{aligned} \tag{12}$$

In problem **P2**, the constraints explicitly require to reduce the overcut and undercut errors simultaneously. For semi-finish milling, non-overcut is the basic requirement which means $d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{envelope}}}(\boldsymbol{w}) \geqslant 0,\ 1\leqslant i\leqslant n$. In a similar way, tool path optimization for semi-finish milling is modeled as the constrained optimization problem

$$\textbf{P3}\qquad \begin{aligned}&\min_{\boldsymbol{w}\in\mathbf{R}^{6(l+1)}}\ \max_{1\leqslant i\leqslant n}\ d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}) - r\\ &\text{s.t.}\quad d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}) - r \geqslant 0,\ i = 1,\cdots,n,\end{aligned} \tag{13}$$

or, equivalently,

$$\textbf{P4}\qquad \begin{aligned}&\min_{(\boldsymbol{w},\xi)\in\mathbf{R}^{6(l+1)+1}}\ \xi\\ &\text{s.t.}\quad 0 \leqslant d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}) - r \leqslant \xi,\ i = 1,\cdots,n.\end{aligned} \tag{14}$$

The method of sequential approximation programming has been used successfully on many practical nonlinear constrained optimization problems[16]. The basic idea of this method is to proceed iteratively by linearizing the objective function and the constraint functions about the current candidate solution, thereby reducing the given nonlinear problem to a sequence of linear programming problems. Here, we apply this method to problem **P2**. Problem **P4** can be solved in a similar way.

Let $(\boldsymbol{w}^k,\xi^k)$ be a candidate solution to problem **P2** and consider a perturbation of the form

$$(\boldsymbol{w}^k + \Delta\boldsymbol{w},\xi^k + \Delta\xi). \tag{15}$$

Using Proposition 2, we have the linearized constraint functions

$$\begin{cases} d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^k) - \left[\boldsymbol{n}^{\boldsymbol{q}_i}\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}_i}\cdot\boldsymbol{\psi}_{w_m}\right]^{\mathrm{T}} \\ \cdot\Delta\boldsymbol{w} - r \leqslant \xi^k + \Delta\xi, \\ d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^k) - \left[\boldsymbol{n}^{\boldsymbol{q}_i}\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}_i}\cdot\boldsymbol{\psi}_{w_m}\right]^{\mathrm{T}} \\ \cdot\Delta\boldsymbol{w} - r \geqslant -\xi^k - \Delta\xi, \end{cases} i = 1,\cdots,n. \tag{16}$$

Obviously, the linearized objective function is equivalent to $\Delta\xi$. Thus, we obtain the corresponding linear programming problem

LP

$$\min_{(\Delta \boldsymbol{w},\Delta\xi)\in\mathbf{R}^{6(l+1)+1}} \quad \Delta\xi$$

$$\text{s.t.} \quad \begin{cases} d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^k)-\left[\boldsymbol{n}^{\boldsymbol{q}_i}\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}_i}\cdot\boldsymbol{\psi}_{w_m}\right]^{\mathrm{T}} \\ \cdot\Delta\boldsymbol{w}-r\leqslant\xi^k+\Delta\xi, \\ d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^k)-\left[\boldsymbol{n}^{\boldsymbol{q}_i}\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}_i}\cdot\boldsymbol{\psi}_{w_m}\right]^{\mathrm{T}} \\ \cdot\Delta\boldsymbol{w}-r\geqslant-\xi^k-\Delta\xi, \end{cases} \quad i=1,\cdots,n. \tag{17}$$

Now, we present the following algorithm for global tool path optimization for five-axis flank milling with a cylindrical cutter.

Algorithm (global tool path optimization).

Input: Initial tool axis trajectory surface $\boldsymbol{S}_{\text{axis}}(\boldsymbol{w}^0)$; threshold ε specifying the desired accuracy of the algorithm.

Output: Optimum tool axis trajectory surface $\boldsymbol{S}_{\text{axis}}(\boldsymbol{w}^*)$; maximum overcut value τ_{over} and maximum undercut value τ_{under}.

Step 0.

(1) Set k=0;

(2) Compute $d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^0)$, $i=1,\cdots,n$;

(3) Set $\xi^0=\max\limits_{1\leqslant i\leqslant n}\left|d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^0)-r\right|$;

Step 1.

(1) Solve the linear programming problem **LP** to determine the differential increment of the surface shape parameters $\Delta\boldsymbol{w}$;

(2) Update $\boldsymbol{w}^{k+1}=\boldsymbol{w}^k+\Delta\boldsymbol{w}$;

(3) Compute $d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^{k+1})$, $i=1,\cdots,n$;

(4) Update $\xi^{k+1}=\max\limits_{1\leqslant i\leqslant n}\left|d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^{k+1})-r\right|$;

(5) If $|1-\xi^k/\xi^{k+1}|>\varepsilon$, then set k=k+1 and go to Step 1(1); else exit and report

$$\boldsymbol{w}^*=\boldsymbol{w}^{k+1},\ \tau_{\text{over}}=\min_{1\leqslant i\leqslant n} d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^{k+1})-r,$$

and $$\tau_{\text{under}}=\max_{1\leqslant i\leqslant n} d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}^{k+1})-r.$$

As with any nonlinear optimization problem, good initial solution is needed. Note that the real machined part has a profile error to the order of micron, i.e., the discrete data points can be well approximated by the tool envelope surface, so the solution to problem **P1** is very close to that to the LS approximation problem defined as

$$\textbf{P5} \qquad \min_{\boldsymbol{w}\in\mathbf{R}^{6(l+1)}} \ \sum_{i=1}^{n}\left[d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w})-r\right]^2. \tag{18}$$

Since problem **P5** can be solved more easily, its solution can serve as an initial estimate to problem **P1**. The Gauss-Newton method or quasi-Newton method can be directly applied to problem **P5** because the gradient vector and Hessian matrix of the function $d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w})$ are both available, which result in a quadratic approximant of the function $\left[d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w})-r\right]^2$ as expressed in eq. (19). The detailed description of the algorithm is omitted. As for problem **P5** itself, the required initial tool axis trajectory surface can be generated by interpolating the discrete cutter axes provided by other methods, such as the TPO method[9].

$$\begin{aligned}&\left[d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w}+\Delta\boldsymbol{w})-r\right]^2 \\ &\approx\left[d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w})-r-\boldsymbol{n}^{\mathrm{T}}\left[\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_{w_m}\right]\Delta\boldsymbol{w}\right]^2 \\ &\quad+\frac{d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w})-r}{d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w})-1/\kappa_1}\left[\boldsymbol{n}_1^{\mathrm{T}}\left[\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_{w_m}\right]\Delta\boldsymbol{w}\right]^2 \\ &\quad+\frac{d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w})-r}{d^s_{\boldsymbol{p}_i,\boldsymbol{S}_{\text{axis}}}(\boldsymbol{w})-1/\kappa_2}\left[\boldsymbol{n}_2^{\mathrm{T}}\left[\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_{w_m}\right]\Delta\boldsymbol{w}\right]^2.\end{aligned} \tag{19}$$

5 Numerical examples

Example 1. In order to demonstrate the validity of the proposed method, we give a simulation of tool path planning for flank milling of the ruled surface expressed in eq. (20), which appears in the previous works[1,9].

$$\begin{aligned}\boldsymbol{S}(u,v)=&(1-v)\left[u\quad 20.429\quad 0\right]^{\mathrm{T}} \\ &+v\left[u\quad 0.0382u^2\quad 33.995\right]^{\mathrm{T}},\end{aligned} \tag{20}$$

where $0\leqslant u\leqslant 23.014$ and $0\leqslant v\leqslant 1$.

The surface was machined using a cylindrical tool with radius of 10 mm. In order to make a comparison with the work presented in ref. [9], 30 tool locations were calculated with the TPO method. By using the lofting method, an initial tool axis trajectory surface which interpolated all these 30 tool axes was generated. 100×100 points were sampled from the design surface. The tool path was optimized using the approach presented in section 4. Figures 4 and 5 show the distributions of the geometric errors before and after optimization, respectively. The maximum overcut reduced from

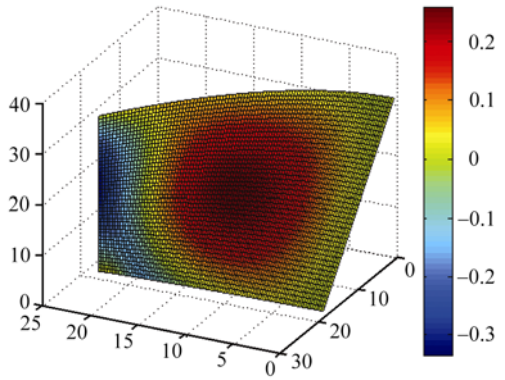

Figure 4 Distribution of the geometric errors before optimization.

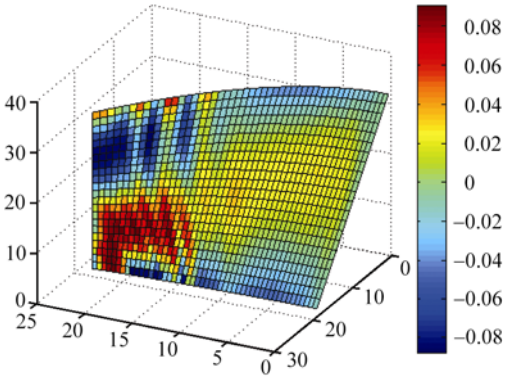

Figure 5 Distribution of the geometric errors after optimization.

0.172 to 0.068 mm and the maximum undercut reduced from 0.228 to 0.067 mm.

In Table 1, we show the maximum undercuts and overcuts of the surfaces machined with the tool paths provided by different approaches. It is seen that our approach is much better than the others. Especially, the maximum undercut and overcut reduce by 26% and 43% in comparison with Gong's results.

Table 1 Geometric errors of the surfaces machined with different tool paths

	Liu[1]	RRD[5]	MBM[7]	Gong[9]	Global optimization
Maximum undercut (mm)	0.582	0.220	0.264	0.093	0.068
Maximum overcut (mm)	0.585	0.220	0.211	0.119	0.067

Example 2. The ruled surface considered is a part surface marked in Figure 6. It is defined by two directrices, which are both B-spline curves of order 3. They have the same knot vector [0, 0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0], and their control points are listed in Table 2. The value of the tool radius chosen for simulation is r=5 mm. Eight cutter locations were determined by using Chiou's method[10], and a smooth axis trajectory surface was generated by interpolating eight pairs of points on these cutter axes with two B-spline curves of order 3. The knot vectors of the obtained B-spline curves are the same as the one mentioned above, and their control points are listed in Table 3. 50×100 points were sampled from the design surface, and two optimum tool paths were obtained with the Minmax and LS optimization algorithms described in section 4. The control points of the two pairs of B-spline curves that determine the two optimum axis trajectory surfaces are listed in Tables 4 and 5. The interferences between the tool envelope surfaces resulting from the initial and the optimum tool paths and the design surface are illustrated in Figures 7 to 9. When LS optimization was applied, the maximum undercut reduced from 0.0247 to 0.0050 mm and the maximum overcut reduced from 0.0167 to 0.0017 mm. When Minmax optimization was applied, the maximum undercut and overcut both became 0.0020 mm. It was seen that the global tool path optimization approach improved the machining accuracy greatly.

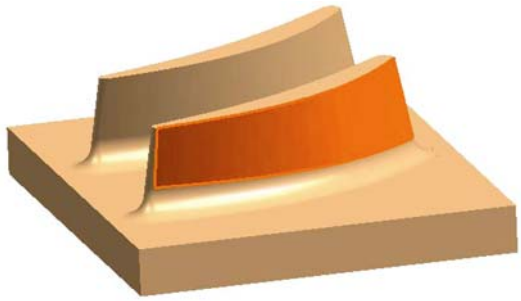

Figure 6 Surface model of a part.

Table 2 Control points of the two directrices of the designed ruled surface

Top directrix			Bottom directrix		
x_0	y_0	z_0	x_1	y_1	z_1
22.5248	88.2770	25.4000	27.4429	89.2469	8.0535
23.9034	79.3817	25.4000	27.2857	82.2509	7.7996
24.2725	69.8340	25.4000	27.9495	67.7686	7.2405
31.2813	49.6940	25.4000	31.0249	52.8044	6.6266
37.4088	38.6031	25.4000	35.8382	39.1707	6.0241
47.2773	25.8031	25.4000	44.3164	24.7250	5.3623
56.3744	18.6647	25.4000	50.5996	18.1743	5.0201
61.0944	14.9329	25.4000	56.1879	12.0318	4.7164

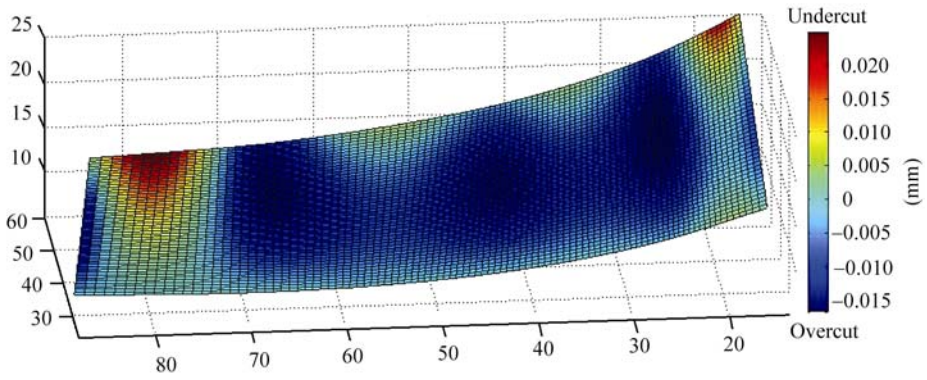

Figure 7 Interference between the tool envelope surface and the design surface before optimization.

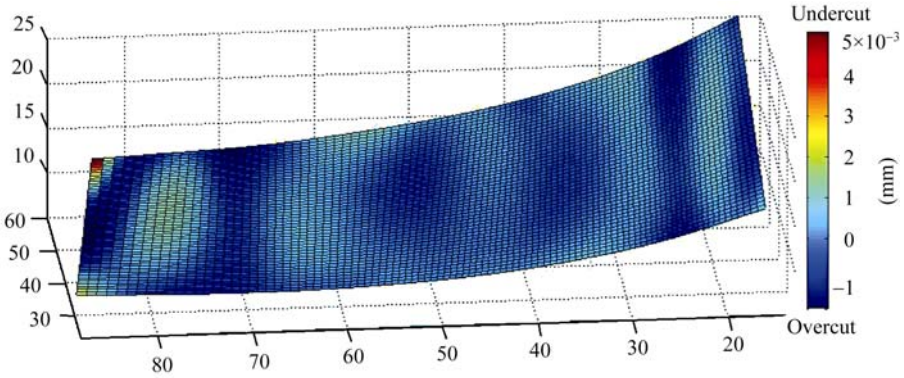

Figure 8 Interference between the tool envelope surface and the design surface after LS optimization.

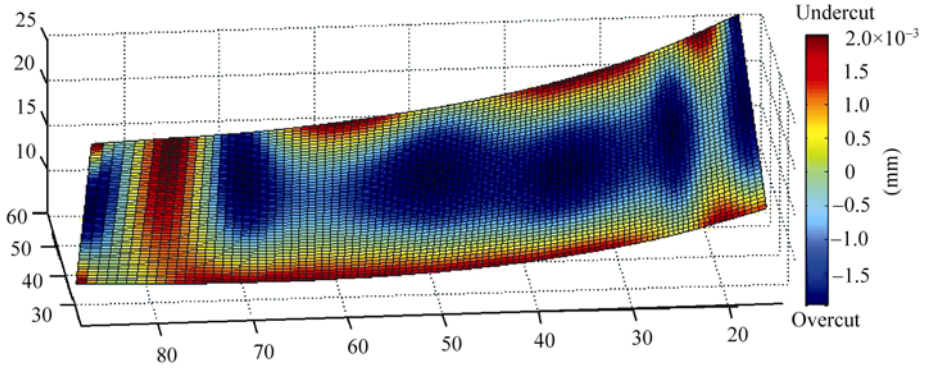

Figure 9 Interference between the tool envelope surface and the design surface after Minimax optimization.

Table 3 Control points of the two directrices of the initial axis trajectory surface

Top directrix			Bottom directrix		
x_0	y_0	z_0	x_1	y_1	z_1
17.7777	87.5413	24.0130	22.6326	89.4042	6.6985
18.9398	79.3519	24.2308	22.3977	82.2619	6.7173
19.4997	68.4959	24.6623	23.0162	67.2096	6.4082
26.5941	47.8533	25.1768	26.1604	51.5201	6.5011
33.2810	35.7067	25.6107	31.2957	37.0155	6.2873
43.6187	22.4632	26.1689	40.2930	21.7872	6.1515
53.5426	14.5699	26.3404	47.1823	14.6575	6.1048
58.0909	11.1345	26.6453	52.6488	8.7479	6.0166

Table 4 Control points of the two directrices of the optimum axis trajectory surface obtained with the LS algorithm

Top directrix			Bottom directrix		
x_0	y_0	z_0	x_1	y_1	z_1
17.7559	87.6522	23.9861	22.6313	89.2745	6.6741
19.0006	79.3551	24.3005	22.3994	82.1775	6.7175
19.3901	68.8180	24.5589	23.0274	67.0947	6.4005
26.5799	47.9077	25.2785	26.2074	51.3427	6.4736
33.1227	35.9506	25.5457	31.3686	36.8589	6.2756
43.6146	22.4305	26.2636	40.4168	21.6045	6.1365
53.3274	14.7956	26.3159	47.3330	14.5054	6.0500
57.9639	11.2542	26.7064	52.7479	8.6332	6.0113

Table 5 Control points of the two directrices of the optimum axis trajectory surface obtained with the Minimax algorithm

Top directrix			Bottom directrix		
x_0	y_0	z_0	x_1	y_1	z_1
17.7917	87.6913	23.8630	22.6658	89.2542	6.5485
18.9843	79.2090	24.3808	22.4056	82.1119	6.6673
19.4451	68.6459	24.5123	23.0396	67.0596	6.3622
26.5936	47.8724	25.3268	26.2029	51.3701	6.4153
33.1778	35.8567	25.4865	31.3543	36.8655	6.1410
43.6434	22.4132	26.3189	40.3844	21.6372	6.1178
53.4084	14.7171	26.2847	47.3092	14.5221	6.0481
58.0016	11.2561	26.7952	52.7987	8.5979	6.0584

6 Conclusions

In this work, a novel approach was proposed to optimize tool path for five-axis flank milling of ruled surface with a cylindrical cutter. The signed point-to-surface distance function was introduced to define the geometric deviation between two surfaces. The presented *interchangeability principle* can lead the problem of globally optimizing the tool path to that of approximating the tool envelope surface to the point cloud on the design surface following the minimum zone criterion. The second order Taylor approximant of the distance function was derived. It characterizes quantitatively the change of the geometric error of the machined surface under the deformation of the tool axis trajectory surface. On this basis, the mathematical models and algorithm for tool path optimizations for semi-finish and finish millings were developed in a unified framework. Comparing with the existing approaches, the present one improves the machining accuracy greatly. Moreover, it is available for semi-finish milling as well as finish milling with trivial modifications of the computer codes. The ideas can also be applied to reconstruction of ruled surface from unorganized point cloud, which is the work in preparation.

It should be noted that from the obtained tool path to the actual machining, a great deal of topics should be taken into account, such as the post-processing, the feedrate scheduling, the kinematical performance of the 5-axis machine tool, and the machining dynamics.

Appendix A Proof of Proposition 3

Proof. The differential increment of the ith element of the gradient vector $\nabla d^s_{p,S}(\boldsymbol{w})$ has the form

$$\begin{aligned}\Delta(-\boldsymbol{n}^q\cdot\boldsymbol{\psi}_{w_i}) &= -\Delta\boldsymbol{n}^q\cdot\boldsymbol{\psi}_{w_I}-\boldsymbol{n}^q\cdot\Delta\boldsymbol{\psi}_{w_i}\\ &= -\Delta\boldsymbol{n}^q\cdot\boldsymbol{\psi}_{w_i}-\boldsymbol{n}^q\cdot\left[\boldsymbol{\psi}_{w_iu},\boldsymbol{\psi}_{w_iv}\right]\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}\\ &\quad -\boldsymbol{n}^q\cdot\left[\boldsymbol{\psi}_{w_iw_1},\cdots,\boldsymbol{\psi}_{w_iw_m}\right]\Delta\boldsymbol{w}. \end{aligned}\tag{A1}$$

According to Weingarten's equations, we have

$$\begin{aligned}\Delta\boldsymbol{n}^q &= \left[\boldsymbol{n}_u,\boldsymbol{n}_v\right]\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}+\left[\boldsymbol{n}_{w_1},\cdots,\boldsymbol{n}_{w_m}\right]\Delta\boldsymbol{w}\\ &= -\left[\boldsymbol{\psi}_u,\boldsymbol{\psi}_v\right]\boldsymbol{g}^{-1}\boldsymbol{\Omega}\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}+\left[\boldsymbol{n}_{w_1},\cdots,\boldsymbol{n}_{w_m}\right]\Delta\boldsymbol{w}. \end{aligned}\tag{A2}$$

Because $\boldsymbol{p}-\boldsymbol{\psi}(\boldsymbol{w},u^*,v^*)=d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\boldsymbol{n}^{\boldsymbol{q}}$, it is obvious that

$$\begin{cases}\left[\boldsymbol{p}-\boldsymbol{\psi}(\boldsymbol{w},u^*,v^*)\right]\cdot\boldsymbol{\psi}_u=0,\\ \left[\boldsymbol{p}-\boldsymbol{\psi}(\boldsymbol{w},u^*,v^*)\right]\cdot\boldsymbol{\psi}_v=0.\end{cases}\tag{A3}$$

Computing the differential increment of eq. (A3), we get

$$\begin{cases}\left(-\boldsymbol{\psi}_u\Delta u-\boldsymbol{\psi}_v\Delta v-\boldsymbol{\psi}_{w_1}\Delta w_1-\cdots-\boldsymbol{\psi}_{w_m}\Delta w_m\right)\cdot\boldsymbol{\psi}_u\\ \quad+d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\boldsymbol{n}^{\boldsymbol{q}}\cdot\left(\boldsymbol{\psi}_{uu}\Delta u+\boldsymbol{\psi}_{uv}\Delta v\right.\\ \quad\left.+\boldsymbol{\psi}_{uw_1}\Delta w_1+\cdots+\boldsymbol{\psi}_{uw_m}\Delta w_m\right)=0,\\ \left(-\boldsymbol{\psi}_u\Delta u-\boldsymbol{\psi}_v\Delta v-\boldsymbol{\psi}_{w_1}\Delta w_1-\cdots-\boldsymbol{\psi}_{w_m}\Delta w_m\right)\cdot\boldsymbol{\psi}_v\\ \quad+d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\boldsymbol{n}^{\boldsymbol{q}}\cdot\left(\boldsymbol{\psi}_{vu}\Delta u+\boldsymbol{\psi}_{vv}\Delta v\right.\\ \quad\left.+\boldsymbol{\psi}_{vw_1}\Delta w_1+\cdots+\boldsymbol{\psi}_{vw_m}\Delta w_m\right)=0,\end{cases}\tag{A4}$$

or, equivalently,

$$\begin{aligned}&-\boldsymbol{g}\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}-\begin{bmatrix}\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_m}\\ \boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_m}\end{bmatrix}\Delta\boldsymbol{w}\\ &+d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\boldsymbol{\Omega}\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}\\ &+d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\begin{bmatrix}\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{uw_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{uw_m}\\ \boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{vw_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{vw_m}\end{bmatrix}\Delta\boldsymbol{w}=0.\end{aligned}\tag{A5}$$

Then we can solve for Δu and Δv from eq. (A5), i.e.,

$$\begin{aligned}\begin{bmatrix}\Delta u\\ \Delta v\end{bmatrix}=\boldsymbol{A}^{-1}&\left(\begin{bmatrix}\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_m}\\ \boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_1},\cdots,\boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_m}\end{bmatrix}\right.\\ &\left.-d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\begin{bmatrix}\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{uw_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{uw_m}\\ \boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{vw_1},\cdots,\boldsymbol{n}^{\boldsymbol{q}}\cdot\boldsymbol{\psi}_{vw_m}\end{bmatrix}\right)\Delta\boldsymbol{w}.\end{aligned}\tag{A6}$$

Substituting eqs. (A2) and (A6) into eq. (A1) provides the finial result.

Appendix B Proof of Proposition 4

Proof. Letting $\boldsymbol{g}^{1/2}=\begin{bmatrix}\|\boldsymbol{\psi}_u\| & 0\\ 0 & \|\boldsymbol{\psi}_v\|\end{bmatrix}$, we have $[\boldsymbol{\psi}_u,\ \boldsymbol{\psi}_v]=[\boldsymbol{n}_1,\boldsymbol{n}_2]\boldsymbol{g}^{1/2}$, $\boldsymbol{g}=\boldsymbol{g}^{1/2}\boldsymbol{g}^{1/2}$, and $\boldsymbol{g}^{-1}\boldsymbol{\Omega}=\begin{bmatrix}\kappa_1 & 0\\ 0 & \kappa_2\end{bmatrix}$.

As a result, matrix $\boldsymbol{A}$ becomes

$$\begin{aligned}\boldsymbol{A}&=\left[d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})\boldsymbol{I}-(\boldsymbol{g}^{-1}\boldsymbol{\Omega})^{-1}\right]\boldsymbol{\Omega}\\ &=\begin{bmatrix}d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})-1/\kappa_1 & 0\\ 0 & d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})-1/\kappa_2\end{bmatrix}\boldsymbol{\Omega}.\end{aligned}$$

Then, it is straightforward to get the final result by substituting all these expressions into the expression of the approximate Hessian matrix $\tilde{\nabla}^2 d^s_{\boldsymbol{p},\boldsymbol{S}}(\boldsymbol{w})$ given in Proposition 4.

Appendix C Proof of Proposition 5

Proof. The deviation evaluation problem expressed in eq. (6) or eq. (7) can be treated as the saddle point programming problem. Assume that surfaces $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ have the parametric representations $\boldsymbol{P}(u, v)$ and $\boldsymbol{Q}(s, t)$, respectively. Eq. (6) can be stated equivalently as

$$\max_{(s,t)}\ \left|\boldsymbol{Q}(s,t)-\boldsymbol{P}(u(s,t),v(s,t))\right|,\tag{C1}$$

where $(u(s,t),v(s,t))$ is the optimal solution to the optimization problem:

$$\min_{(u,v)}\ \left|\boldsymbol{Q}(s,t)-\boldsymbol{P}(u,v)\right|,\tag{C2}$$

which satisfies the first order optimality condition:

$$\begin{cases}\boldsymbol{P}_u\cdot\left[\boldsymbol{Q}(s,t)-\boldsymbol{P}(u,v)\right]=0,\\ \boldsymbol{P}_v\cdot\left[\boldsymbol{Q}(s,t)-\boldsymbol{P}(u,v)\right]=0.\end{cases}\tag{C3}$$

The first order optimality condition for the optimization problem (C1) is

$$\begin{cases}\left[\boldsymbol{Q}_s-\boldsymbol{P}_u\dfrac{\mathrm{d}u}{\mathrm{d}s}-\boldsymbol{P}_v\dfrac{\mathrm{d}v}{\mathrm{d}s}\right]\cdot\left[\boldsymbol{Q}(s,t)-\boldsymbol{P}(u,v)\right]=0,\\ \left[\boldsymbol{Q}_t-\boldsymbol{P}_u\dfrac{\mathrm{d}u}{\mathrm{d}t}-\boldsymbol{P}_v\dfrac{\mathrm{d}v}{\mathrm{d}t}\right]\cdot\left[\boldsymbol{Q}(s,t)-\boldsymbol{P}(u,v)\right]=0.\end{cases}\tag{C4}$$

Substituting eq. (C-3) into eq. (C-4) yields

$$\begin{cases}\boldsymbol{Q}_s\cdot\left[\boldsymbol{Q}(s,t)-\boldsymbol{P}(u,v)\right]=0,\\ \boldsymbol{Q}_t\cdot\left[\boldsymbol{Q}(s,t)-\boldsymbol{P}(u,v)\right]=0.\end{cases}\tag{C5}$$

Let (u_*,v_*,s_*,t_*) be the optimal solution to the optimization problem expressed in eq. (6). Eqs. (C2) and (C4) indicate that the surface normals at points $\boldsymbol{P}(u_*,v_*)$ and $\boldsymbol{Q}(s_*,t_*)$ are collinear with the line that connects the two points, and that (s_*,t_*) is the optimal solution to the following optimization problem:

$$\min_{(s,t)}\left|\boldsymbol{P}(u_*,v_*)-\boldsymbol{Q}(s,t)\right|.$$

Then we obtain

$$e_1 = \left|\boldsymbol{Q}(s_*,t_*) - \boldsymbol{P}(u_*,v_*)\right| = \min_{(s,t)}\left|\boldsymbol{P}(u_*,v_*) - \boldsymbol{Q}(s,t)\right|$$
$$\leqslant \max_{(u,v)}\min_{(s,t)}\left|\boldsymbol{P}(u,v) - \boldsymbol{Q}(s,t)\right| = e_2.$$

In a similar way, we can obtain $e_2 \leqslant e_1$ through solving the saddle point programming problem expressed in eq. (7), and derive the result

$$e_1 = e_2.$$

The authors would like to thank Dr. GONG Hu for the valuable discussion with him.

1 Liu X W. Five-axis NC cylindrical milling of sculptured surfaces. Comput-Aided Des, 1995, 27(12): 887－894

2 Bohez E L J, Senadhera S D R, Pole K, et al. A geometric modeling and five-axis machining algorithm for centrifugal impellers. J Manuf Syst, 1997, 16(6): 422－436

3 Lee J J, Suh S H. Interference-free tool-path planning for flank milling of twisted ruled surfaces. Int J Adv Manuf Tech, 1998, 14(11): 797－805

4 Rubio W, Lagarrigue P, Dessein G, et al. Calculation of tool paths for a torus mill on free-form surfaces on five-axis machines with detection and elimination of interference. Int J Adv Manuf Tech, 1998, 14(1): 13－20

5 Redonnet J M, Rubio W, Dessein G. Side milling of ruled surfaces: Optimum positioning of the milling cutter and calculation of interference. Int J Adv Manuf Tech, 1998, 14(7): 459－465

6 Bedi S, Mann S, Menzel C. Flank milling with flat end milling cutters. Comput-Aided Des, 2003, 35(3): 293－300

7 Menzel C, Bedi S, Mann S. Triple tangent flank milling of ruled surfaces. Comput-Aided Des, 2004, 36(3): 289－296

8 Tsay D M, Her M J. Accurate 5-axis machining of twisted ruled surfaces. ASME J Manuf Sci Eng, 2001, 123(4): 731－738

9 Gong H, Cao L X, Liu J. Improved positioning of cylindrical cutter for flank milling ruled surfaces. Comput-Aided Des, 2005, 37(12): 1205－1213

10 Chiou C J. Accurate tool position for five-axis ruled surface machining by swept envelope approach. Comput-Aided Des, 2004, 36(10): 967－974

11 Ding Y, Zhu L M, Ding H. Semidefinite programming for Chebyshev fitting of spatial straight line with applications to cutter location planning and tolerance evaluation. Precis Eng, 2007, 31(4): 364－368

12 Lartigue C, Duc E, Affouard A. Tool path deformation in 5-axis flank milling using envelope surface. Comput-Aided Des, 2003, 35(4): 375－382

13 ISO/R 1101. Technical Drawings-Geometrical Tolearncing. Geneva: International Organization for Standardization, 1983

14 ANSI Standard Y14.5. Dimensioning and Tolerancing. New York: The American Society of Engineers, 1982

15 Zhu L M, Ding H. A unified approach for least-squares surface fitting. Sci China Ser G-Phys Mech Astron, 2004, 47(Suppl I): 72－78

16 Nocedal J, Wright S J. Numerical Optimization. New York: Springer-verlag, 1999