Li Min Zhu[1]
State Key Laboratory of Mechanical System and Vibration,
School of Mechanical Engineering,
Shanghai Jiao Tong University,
Shanghai 200240, P.R. China
e-mail: zhulm@sjtu.edu.cn

Xiao Ming Zhang
State Key Laboratory of Digital Manufacturing Equipment and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, P.R. China

Han Ding
State Key Laboratory of Mechanical System and Vibration,
School of Mechanical Engineering,
Shanghai Jiao Tong University,
Shanghai 200240, P.R. China

You Lun Xiong
State Key Laboratory of Digital Manufacturing Equipment and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, P.R. China

Geometry of Signed Point-to-Surface Distance Function and Its Application to Surface Approximation

This paper presents a unified framework for computing a surface to approximate a target shape defined by discrete data points. A signed point-to-surface distance function is defined, and its properties are investigated, especially, its second-order Taylor approximant is derived. The intercorrelations between the signed and the squared distance functions are clarified, and it is demonstrated that the squared distance function studied in the previous works is just the Type I squared distance function deduced from the signed distance function. It is also shown that surface approximations under different criteria and constraints can all be formulated as optimization problems with specified requirements on the residual errors represented by the signed distance functions, and that classical numerical optimization algorithms can be directly applied to solve them since the derivatives of the involved objective functions and constraint functions are all available. Examples of global cutter position optimization for flank milling of ruled surface with a cylindrical tool, which requires approximating the tool envelope surface to the point cloud on the design surface following the minimum zone criterion, are given to confirm the validity of the proposed approach. [DOI: 10.1115/1.3510588]

Keywords: CAGD, distance function, surface approximation, Taylor approximant, flank milling, cutter position optimization

1 Introduction

Computing a surface to approximate data points is a problem encountered frequently in many applications in computer-aided design/computer-aided manufacturing (CAD/CAM), computer graphics, computer vision, and image processing. During the approximating process, the surface may adjust its location (position/orientation), shape, or both. If the least-squares (LS) criterion is employed, such three problems are referred to as registration, surface fitting, and combination of surface fitting and registration [1]. Normally, they are formulated as a large-scale nonlinear LS problem; the objective function is the sum of the squared distances of the data points from the model surface, and the variables are the displacement (from a nominal location) and/or shape parameters of the candidate surface and n pairs of surface coordinates, where n is the point number. The optimal solution gives the surface location and/or surface model. In Ref. [2], the Gauss–Newton method was directly used to solve the registration problem. However, a large computer memory is required, and much computational effort is wasted because the matrices involved are sparse, owing to the independence of the surface coordinates. By representing the optimal shape parameters as the functions of the surface coordinates, Sarkar and Menq [3] simplified the surface fitting problem as a nonlinear LS problem defined on $\mathbb{R}^{2n}$. However, the two drawbacks still remained. Currently, the popular way to solve such a large-scale problem is the alternating variable approach [4–12]. With this method, first, the surface displacement and/or shape parameters are fixed, and the objective function is minimized by changing the surface coordinates, which leads to n times computation of the nearest surface point. Then, with all the surface coordinates fixed, the objective function is minimized with respect to the surface displacement and/or shape parameters, providing an improved solution. Such a process continues iteratively until convergence occurs. The different considerations in updating the surface location and/or shape parameters at each iteration lead to the different surface approximation algorithms. Except for the two special cases, i.e., registration and surface fitting using B-spline surface, there is no closed-form solution to the subproblem of improving the surface location and/or shape parameters. In fact, this is a multivariable nonlinear optimization problem. Although there exist various algorithms for nonlinear programming, they are typically computationally expensive. Moreover, it is found that the algorithms solving the subproblem exactly do not result in a fewer number of iterations performed before convergence [1,4]. Therefore, most of the algorithms just find a properly approximate solution at the intermediate stage.

In the above studies, these three problems are investigated separately, and the interconnections among them are completely overlooked. For instance, performance of the five typical registration algorithms are studied and compared in Ref. [4]. Arising from this study, the Hong–Tan algorithm distinguishes itself in terms of performance measures including efficiency, robustness with respect to variations in initial conditions, and accuracy, but the counterparts for the other two problems are undiscovered. Recently, Pottmann and his co-workers introduced the squared distance function of a surface and derived its second-order Taylor approximant [13]. On this basis, they developed two Newton-like algorithms for LS surface approximation in a unified manner [1,14–16]. At almost the same time, we introduced the signed distance function of a surface and investigated its differential properties. Using signed distance function, we presented a unified framework for best-fitting of surface to 3D coordinate data [17–20]. Within this framework, we also discussed the problems of surface approximating by minimizing the l_∞- or l_1-norm of the residual errors, geometric symmetry identification, probe radius

[1]Corresponding author.

Contributed by the Computational Geometry/Visualization Committee of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received July 16, 2009; final manuscript received June 29, 2010; published online November 23, 2010. Assoc. Editor: A. Fischer.

compensation, and motion analysis under point contact constraints. It is worth mentioning that although the concept of the signed distance function was presented before our works [21], its properties and applications were not systematically addressed there. In this paper, we will investigate thoroughly the differential properties of the signed distance function and show its intercorrelations with the squared distance function. Also, we will discuss in detail a special surface approximation problem arising from five-axis CNC machining, i.e., global cutter position optimization for flank milling of ruled surface, which could not be well handled with the squared distance function.

The remainder of this paper is organized as follows: In Sec. 2, the signed point-to-surface distance function is defined, and its properties are investigated, especially, its second-order Taylor approximant is derived. Also, the intercorrelations between the signed and the squared distance functions are explored, and their applications to surface approximation are briefly discussed. In Sec. 3, optimum cutter positioning for five-axis flank milling of ruled surfaces with a cylindrical cutter is addressed from the perspective of surface approximation. By using the signed distance function, the mathematical models and algorithms for tool path optimization for rough and finish millings are developed in a unified framework. Examples are given in Sec. 4, and conclusions are in Sec. 5.

Remark. When revising this paper, we were aware of Flöry and Hofer's work [22]. In that work, the authors derived approximations of the unsigned distance function to B-spline surfaces and point clouds, and then employed these approximations to solve the geometric matching problems of registration of point clouds and surface fitting to point sets based on the l_1-norm. The distance function discussed there is just the Type I simplified distance function deduced from the signed distance function defined in this work. Also, we focus on a special surface approximation problem arising from five-axis CNC machining, which requires to approximate the envelope surface of the tool to the point cloud on the design surface based on the l_∞-norm, which is recommended by ANSI and ISO standards for tolerance evaluation.

2 Distance Function

2.1 Definition and Differential Properties. DEFINITION 1. *Given a regular surface* $\mathbf{S}(\mathbf{w}) \in \mathbb{R}^3$, *where* $\mathbf{w}=[w_1,\ldots,w_m]^T \in \mathbb{R}^m$ *denotes the collection of the shape parameters, and a point* $\mathbf{p} \in \mathbb{R}^3$, *there exists at least one closest point* $\mathbf{q} \in \mathbf{S}(\mathbf{w})$, *termed as foot point, such that* $\|\mathbf{p}-\mathbf{q}\|=\min_{\mathbf{x}\in\mathbf{S}(\mathbf{w})}\|\mathbf{p}-\mathbf{x}\|$, *where* $\|\cdot\|$ *stands for the Euclidean norm on* $\mathbb{R}^3$. *The distance function is defined as* $d(\mathbf{p},\mathbf{w})=\|\mathbf{p}-\mathbf{q}\|$.

For most engineering surfaces such as planes, cylinders, cones, and spheres, the unsigned distance function can be explicitly computed, while for complex algebraic surfaces and parametric sculptured surfaces such as Bézier, B-spline, and non-uniform rational B-spline (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 [23,24].

PROPOSITION 1 [24]. *If* $\mathbf{q}$ *lies in the interior of* $\mathbf{S}(\mathbf{w})$, *then the error vector* $\mathbf{p}-\mathbf{q}$ *is normal to* $\mathbf{S}(\mathbf{w})$, *i.e.,* $\mathbf{p}-\mathbf{q}=\pm d(\mathbf{p},\mathbf{w})\mathbf{n}^{\mathbf{q}}$, *where* $\mathbf{n}^{\mathbf{q}}$ *is the unit outward normal vector of surface* $\mathbf{S}(\mathbf{w})$ *at point* $\mathbf{q}$. *The choice of plus or minus sign depends on the direction of* $\mathbf{n}^{\mathbf{q}}$.

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

DEFINITION 2. *If* $\mathbf{q}$ *is unique and lies in the interior of* $\mathbf{S}(\mathbf{w})$, *the signed distance function is defined as* $d^s(\mathbf{p},\mathbf{w})=(\mathbf{p}-\mathbf{q})\cdot\mathbf{n}^{\mathbf{q}}$, *where the bold dot denotes scalar product.*

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

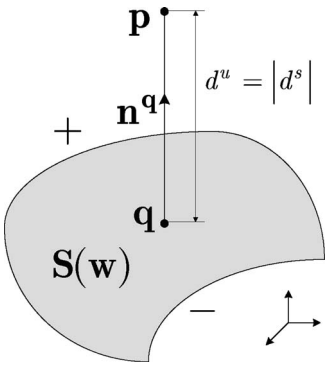

Fig. 1 Point-to-surface distance function

The signed distance function has the following differential property:

PROPOSITION 2 [17,18]. *Assume that surface* $\mathbf{S}(\mathbf{w})$ *has a locally parametric representation* $\boldsymbol{\psi}(\mathbf{w},u,v)$ *and* $\mathbf{q}=\boldsymbol{\psi}(\mathbf{w},u^*,v^*)$, *the first-order Taylor approximant of* $d^s(\mathbf{p},\mathbf{w})$ *is given by*

$$\bar{d}^s(\mathbf{p}+\Delta\mathbf{p},\mathbf{w}+\Delta\mathbf{w}) = d^s(\mathbf{p},\mathbf{w}) + \mathbf{n}^{\mathbf{q}}\cdot\Delta\mathbf{p} - [\mathbf{n}^{\mathbf{q}}\cdot\boldsymbol{\psi}_{w_1},\ldots,\mathbf{n}^{\mathbf{q}}\cdot\boldsymbol{\psi}_{w_m}]^T\cdot\Delta\mathbf{w} \quad (1)$$

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

COROLLARY 1. *If surface* $\mathbf{S}(\mathbf{w})$ *is an algebraic surface implicitly represented as* $f(x,y,z,\mathbf{w})=0$ *and* $\mathbf{q}=[x^*,y^*,z^*]^T$, *then one has*

$$\bar{d}^s(\mathbf{p}+\Delta\mathbf{p},\mathbf{w}+\Delta\mathbf{w}) = d^s(\mathbf{p},\mathbf{w}) + \|[f_x,f_y,f_z]\|^{-1}([f_x,f_y,f_z]\Delta\mathbf{p} + [f_{w_1},\ldots,f_{w_m}]\Delta\mathbf{w}) \quad (2)$$

where the partial derivatives are all evaluated at $(x^*,y^*,z^*,\mathbf{w})$.

The proof of Corollary 1 is given in Appendix A. When the shape parameters of surface $\mathbf{S}(\mathbf{w})$ are fixed, i.e., $\mathbf{w}$ is a constant vector, we get a simplified distance function $d^s(\mathbf{p}) \triangleq d^s(\mathbf{p},\mathbf{w}_0)$ and its gradient vector $\nabla d^s(\mathbf{p})=\mathbf{n}^{\mathbf{q}}$. Under certain condition, the simplified distance function $d^s(\mathbf{p})$ is also second-order differentiable, which is shown in the following proposition:

PROPOSITION 3. *Assume that surface* $\mathbf{S}$ *has a locally parametric representation* $\boldsymbol{\psi}(u,v)$ *and* $\mathbf{q}=\boldsymbol{\psi}(u^*,v^*)$. *If matrix* $\mathbf{A} \triangleq d^s(\mathbf{p})\boldsymbol{\Omega} - \mathbf{g}$ *is invertible, where* $\mathbf{g}$ *and* $\boldsymbol{\Omega}$ *are the first and second fundamental matrices of surface* $\mathbf{S}$ *at point* $\mathbf{q}$, *respectively, then* $d^s(\mathbf{p})$ *is second-order differentiable and its Hessian matrix is* $\nabla^2 d^s(\mathbf{p}) = [\boldsymbol{\psi}_u,\boldsymbol{\psi}_v]\mathbf{g}^{-1}\boldsymbol{\Omega}\mathbf{A}^{-1}[\boldsymbol{\psi}_u,\boldsymbol{\psi}_v]^T$, *where the partial derivatives* $\boldsymbol{\psi}_u$ *and* $\boldsymbol{\psi}_v$ *are evaluated at* (u^*,v^*) .

The proof of Proposition 3 can be seen in Appendix B. After some complicated but straightforward calculations, the determinant of matrix $\mathbf{A}$ is obtained as follows:

$$\det(\mathbf{A}) = \det(\mathbf{g})(\kappa_1 d^s(\mathbf{p}) - 1)(\kappa_2 d^s(\mathbf{p}) - 1) \quad (3)$$

where κ_1 and κ_2 are the principal curvatures at point $\mathbf{q}$. It shows that matrix $\mathbf{A}$ can become singular when point $\mathbf{p}$ is at the center of one of the principal curvatures at point $\mathbf{q}$ of surface $\mathbf{S}(\mathbf{w})$, as can be seen in Fig. 2. In practical computation, such singular cases seldom occur, especially when $\mathbf{p}$ is close to $\mathbf{S}(\mathbf{w})$.

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

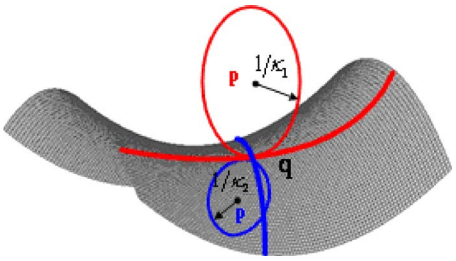

Fig. 2 Singular cases

get two families of coordinate curves on surface **S** of which the tangents at point **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 **g** and **Ω** become diagonal matrices; we obtain a simple expression of the Hessian matrix of $d^s(\mathbf{p})$, as shown in the following corollary:

COROLLARY 2. *Denote by* $\mathbf{n}_1$ *and* $\mathbf{n}_2$ *the two mutually orthogonal unit vectors that determine the two principal directions associated with the two principal curvatures* κ_1 *and* κ_2 *at* $\mathbf{q}$ *, respectively, the Hessian matrix of* $d^s(\mathbf{p})$ *has the form*

$$\nabla^2 d(\mathbf{p}) = \frac{1}{d^s(\mathbf{p}) - 1/\kappa_1}\mathbf{n}_1\mathbf{n}_1^T + \frac{1}{d^s(\mathbf{p}) - 1/\kappa_2}\mathbf{n}_2\mathbf{n}_2^T \tag{4}$$

The proof of Corollary 2 appears in Appendix C. In many applications, the squared distance function $d^2(\mathbf{p}) \triangleq [d^s(\mathbf{p})]^2$ is adopted as a measure of the approximation error [1]. Based on the above discussions, we can easily get the gradient vector and Hessian matrix of $d^2(\mathbf{p})$ because $\nabla d^2 = 2d^s\nabla d^s$ and $\nabla^2 d^2 = 2\nabla d^s(\nabla d^s)^T + 2d^s\nabla^2 d^s$. When expressed in terms of $\mathbf{n}^\mathbf{q}$ and $\mathbf{n}_1$ and $\mathbf{n}_2$ at $\mathbf{q}$, the second-order Taylor approximant of $d^2(\mathbf{p})$ is written in the form

$$\tilde{d}^2(\mathbf{p}+\Delta\mathbf{p}) = [d^s(\mathbf{p}) + \mathbf{n}^\mathbf{q}\cdot\Delta\mathbf{p}]^2 + \frac{d^s(\mathbf{p})}{d^s(\mathbf{p}) - 1/\kappa_1}[\mathbf{n}_1\cdot\Delta\mathbf{p}]^2 + \frac{d^s(\mathbf{p})}{d^s(\mathbf{p}) - 1/\kappa_2}[\mathbf{n}_2\cdot\Delta\mathbf{p}]^2 \tag{5}$$

which is, in essence, the same as that reported in Refs. [1,13] due to the fact that $\mathbf{p}-\mathbf{q} = d^s(\mathbf{p})\mathbf{n}^\mathbf{q}$. In the special case of $d^s(\mathbf{p})=0$ (i.e., $\mathbf{p}=\mathbf{q}$), the approximant $\tilde{d}^2(\mathbf{p}+\Delta\mathbf{p})$ equals the squared distance function to the tangent plane of surface **S** at point **q**. Thus, if **p** is close to **S**,

$$\breve{d}^2(\mathbf{p}+\Delta\mathbf{p}) = [d^s(\mathbf{p}) + \mathbf{n}^\mathbf{q}\cdot\Delta\mathbf{p}]^2 \tag{6}$$

is a good approximation of $\tilde{d}^2(\mathbf{p}+\Delta\mathbf{p})$.

Remark. When the foot point **q** is a parabolic point or a flat point, i.e., one or both of the two principal curvatures vanish, the above second-order Taylor approximant still holds. We just need to take the limit of the function $\tilde{d}^2$ as κ_1 or/and κ_2 approaches zero.

Similarly, when point **p** is fixed, we get another simplified distance function $d^s(\mathbf{w}) \triangleq d^s(\mathbf{p}_0, \mathbf{w})$ and its gradient vector $\nabla d^s(\mathbf{w}) = -[\mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{w_1}, \ldots, \mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{w_m}]^T$. The Hessian matrix of $d^s(\mathbf{w})$ can then be derived in a way similar to the proof of Proposition 3 and is given in the following proposition:

PROPOSITION 4. *If matrix* **A** *is invertible,* $d^s(\mathbf{w})$ *is second-order differentiable and the* ith *row,* jth *column element of its Hessian matrix* $\nabla^2 d^s(\mathbf{w})$ *is*

$$\begin{aligned}[\nabla^2 d^s(\mathbf{w})]_{ij} = {} & [\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_i}, \boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_i}]\mathbf{g}^{-1}\boldsymbol{\Omega}\mathbf{A}^{-1}[\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_j}, \boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_j}]^T \\ & - d^s(\mathbf{w})[\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_i}, \boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_i}]\mathbf{g}^{-1}\boldsymbol{\Omega}\mathbf{A}^{-1} \\ & \times[\mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{uw_j}, \mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{vw_j}]^T \\ & - \mathbf{n}^\mathbf{q}_{w_j}\cdot\boldsymbol{\psi}_{w_i} \\ & - [\mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{w_iu}, \mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{w_iv}]\mathbf{A}^{-1}[\boldsymbol{\psi}_u\cdot\boldsymbol{\psi}_{w_j}, \boldsymbol{\psi}_v\cdot\boldsymbol{\psi}_{w_j}]^T \\ & + d^s(\mathbf{w})[\mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{w_iu}, \mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{w_iv}]\mathbf{A}^{-1} \\ & \times[\mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{uw_j}, \mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{vw_j}]^T - \mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{w_iw_j}\end{aligned} \tag{7}$$

Differentiating both sides of the system of identities

$$\boldsymbol{\psi}_u\cdot\mathbf{n}^\mathbf{q} = 0$$

$$\boldsymbol{\psi}_v\cdot\mathbf{n}^\mathbf{q} = 0$$

with respect to shape parameter w_j, we obtain

$$\boldsymbol{\psi}_{uw_j}\cdot\mathbf{n}^\mathbf{q} + \boldsymbol{\psi}_u\cdot\mathbf{n}^\mathbf{q}_{w_j} = 0$$

$$\boldsymbol{\psi}_{vw_j}\cdot\mathbf{n}^\mathbf{q} + \boldsymbol{\psi}_u\cdot\mathbf{n}^\mathbf{q}_{w_j} = 0 \tag{8}$$

With $\|\mathbf{n}^\mathbf{q}\| \equiv 1$ we see $\mathbf{n}^\mathbf{q}_{w_j} \in T_\mathbf{q}S$, so we assume $\mathbf{n}^\mathbf{q}_{w_j} = a\boldsymbol{\psi}_u + b\boldsymbol{\psi}_v$, which is substituted into Eq. (8), then we obtain

$$\mathbf{n}^\mathbf{q}_{w_j} = -[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]\mathbf{g}^{-1}[\mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{uw_j}, \mathbf{n}^\mathbf{q}\cdot\boldsymbol{\psi}_{vw_j}]^T \tag{9}$$

By using Eq. (9), it is easy to verify that $[\nabla^2 d^s(\mathbf{w})]_{ij} - [\nabla^2 d^s(\mathbf{w})]_{ji} = 0$. This means that matrix $\nabla^2 d^s(\mathbf{w})$ is indeed symmetric. The Hessian matrix $\nabla^2 d^s(\mathbf{w})$ 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 $\boldsymbol{\psi}(\mathbf{w}, u, v)$. This results in an approximate Hessian matrix

$$\tilde{\nabla}^2 d^s(\mathbf{w}) = [\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}]^T[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]\mathbf{g}^{-1}\boldsymbol{\Omega}\mathbf{A}^{-1} \times[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]^T[\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}] \tag{10}$$

In a way similar to the proof of Corollary 2, we get a simple expression for $\tilde{\nabla}^2 d^s(\mathbf{w})$ as follows.

PROPOSITION 5. *Denote by* $\mathbf{n}_1$ *and* $\mathbf{n}_2$ *the two mutually orthogonal unit vectors that determine the two principal directions associated with the two principal curvatures* κ_1 *and* κ_2 *at* $\mathbf{q}$ *, respectively, the approximate Hessian matrix of* $d^s(\mathbf{w})$ *has the form*

$$\tilde{\nabla}^2 d^s(\mathbf{w}) = \sum_{i=1}^{2}\frac{1}{d^s(\mathbf{w}) - 1/\kappa_i}[\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}]^T\mathbf{n}_i\mathbf{n}_i^T[\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}] \tag{11}$$

Then, we can define another squared distance function $d^2(\mathbf{w}) \triangleq [d^s(\mathbf{w})]^2$ and obtain its two quadratic approximants,

$$\begin{aligned}\tilde{d}^2(\mathbf{w}+\Delta\mathbf{w}) = {} & [d^s(\mathbf{w}) - \mathbf{n}^T[\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}]\Delta\mathbf{w}]^2 \\ & + \frac{d^s(\mathbf{w})}{d^s(\mathbf{w}) - 1/\kappa_1}[\mathbf{n}_1^T[\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}]\Delta\mathbf{w}]^2 \\ & + \frac{d^s(\mathbf{w})}{d^s(\mathbf{w}) - 1/\kappa_2}[\mathbf{n}_2^T[\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}]\Delta\mathbf{w}]^2\end{aligned} \tag{12}$$

$$\breve{d}^2(\mathbf{w}+\Delta\mathbf{w}) = [d^s(\mathbf{w}) - \mathbf{n}^T[\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}]\Delta\mathbf{w}]^2 \tag{13}$$

2.2 Intercorrelations Among Different Distance Functions. So far, we have defined three types of signed distance functions and two types of squared distance functions and derived their first-order and second-order approximants. The relationships among them are summarized in Fig. 3. The Type I squared distance function is actually the one extensively studied by Pottmann and his co-workers [1,13–16] and the Type I simplified distance function the one recently reported by Flöry and Hofer [22].

The three types of LS surface approximation problems are independently investigated in Refs. [1,14,15,17–19] by using the squared and the signed distance functions, i.e., $d^2(\mathbf{p})$ and $d^s(\mathbf{p}, \mathbf{w})$,

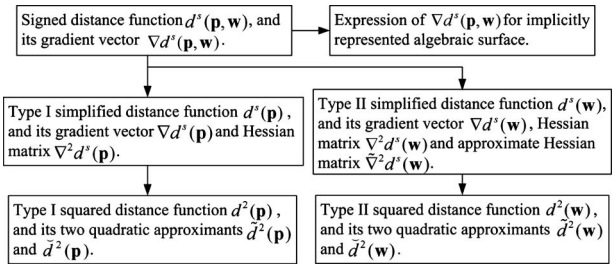

Fig. 3 Relationships among the distance functions discussed

respectively. In Pottmann and his co-workers' works, surface approximating is treated as a general optimization problem, and based on the second-order approximants of the Type I squared distance function, i.e., $\tilde{d}^2$ and $\breve{d}^2$, two Newton-like algorithms, called SDM and TDM, are developed. In our works, surface approximation is stated as a nonlinear LS problem, and an improved Levenberg–Marquardt algorithm, which only requires the first-order approximant of the signed distance function, i.e., $\bar{d}^s$, is adopted. Interestingly, it is found that our algorithm is identical to TDM except for the strategy for step-size control employed to ensure the stability of the iterations in the case of a bad initial solution, and that for the registration problem, they lead to the Hong–Tan algorithm [4] although they are derived from different viewpoints. Proposition 2 offers a solid mathematical basis of this algorithm. Also, Corollary 1 shows that this algorithm is applicable to simultaneous registration and fitting of an implicit surface, which is a topic for further research stated in Ref. [1]. In fact, for surface fitting problem, the introduction of SDM and TDM based on the Type I squared distance function $d^2(\mathbf{p})$ is not justifiable because the shape parameters of the surface are not considered in its definition. The investigation of the Type II squared distance function $d^2(\mathbf{w})$ in Sec. 2 would overcome this difficulty. Besides the three problems mentioned above, there are many other problems involving surface approximating under different criteria and constraints [17,19]. They can all be formulated as optimization problems with specified requirements on the residual errors represented by the signed distance functions. These optimization problems can then be efficiently solved by the sequential approximation algorithms because the derivatives of the involved objective functions and constraint functions are all available. Examples of surface approximation by minimizing the l_∞-norm or l_1-norm of the residual errors can be found in Refs. [17,19,20,22]. In Sec. 3, we will discuss in detail a special surface approximation problem arising from five-axis CNC machining, i.e., global cutter position optimization for flank milling of ruled surfaces, which involves one-sided constraint and thus could not be well handled with the squared distance function.

3 Surface Approximation for Flank Milling Tool Path Optimization

3.1 Mathematical Models. Ruled surfaces are widely used in industry. For slender parts, such as turbine blades and impellers, freeform surfaces are usually approximated by piecewise ruled surfaces. Elber and Fish [25] presented a scheme for automatic piecewise ruled surface approximation of freeform surfaces. Subag and Elber [26] developed a semi-automatic algorithm for piecewise developable surface approximation of general NURBS surfaces. Flank milling can be effectively employed to machine ruled surface for the advantage of larger material removal rate as compared with point milling. Also, no scallops are left behind in single pass flank milling. Recently, increasing attention was drawn onto the problem of optimum positioning of the cutter for flank milling [27–32]. However, most of the works focused on the individual cutter location planning; the tool path optimization from a global perspective was little addressed. Lartigue et al. [31] proposed a global tool path planning method. The basic idea was to deform the two curves that defined the tool axis 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 was employed. For a cylindrical cutter, Gong et al. [32] presented the *error propagation* principle and formulated the problem as that of LS fitting of tool axis trajectory surface to point cloud on the offset surface of the design surface. Although the LS method is easy for implementation and efficient in computation, it cannot incorporate readily the nonovercut constraint required by rough milling, and more importantly, it does not conform to the minimum zone criterion recommended by ANSI and ISO standards for tolerance evaluation [33,34], which requires the maximum norm of the error vector to 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. Here, we revisit this problem from the perspective of surface approximation based on the l_∞-norm.

Numerical control (NC) machining is a process that subtracts the swept volume 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 S_{envelope} approximates to the designed surface S_{design} as much as possible. It is difficult to compare two continuous surfaces directly to obtain the deviation between them. Alternatively, we can use the distances from the sample points on the design surface S_{design} to the envelop surface S_{envelope} to evaluate the geometrical deviation. According to the tolerance definition in ANSI and ISO standards, the maximum distance can serve as the measure of the geometrical deviation between the two surfaces.

As stated in Ref. [35], the state-of-the-art methods for modeling swept envelope surface generated by the motion of NC tools along complex 3D paths are either based on approximation techniques or on analytical solutions with high computational complexity, which are not suitable for practical applications. In our studies followed, we provide a method to compute the distance from a point to surface S_{envelope} without constructing the envelope surface. Given a surface S, its offset surface S_{offset} is the surface with a constant offset to it. Obviously, for a cylindrical tool, the tool axis trajectory surface S_{axis} is the offset surface of the tool envelope surface S_{envelope}, and the offset distance is the tool radius r. As a result, we have

$$d^s_{\mathbf{p},S_{\text{envelope}}} = d^s_{\mathbf{p},S_{\text{axis}}} - r \tag{14}$$

for a point $\mathbf{p}$ on the design surface. Here, it is assumed that the normal vector of the tool axis trajectory surface S_{axis} points to the design surface S_{design}. Evidently, if $d^s_{\mathbf{p},S_{\text{envelope}}}$ and $d^s_{\mathbf{p},S_{\text{axis}}}$ are well-defined, they share the common gradient vector and Hessian matrix.

As illustrated in Fig. 4, the tool motion is usually represented by two guiding curves $\boldsymbol{P}(u)$ and $\boldsymbol{Q}(u)$. They determine the following tool axis trajectory surface

$$S_{\text{axis}}(\mathbf{w}): \boldsymbol{\psi}(\mathbf{w},u,v) = (1-v)\boldsymbol{P}(u) + v\boldsymbol{Q}(u) \tag{15}$$

where $\mathbf{w}=[w_1,\ldots,w_m]^T \in \mathbb{R}^m$ denotes the collection of the shape parameters of the two curves. If the guiding curve is a B-spline curve, the coordinates of its control points can be viewed as the shape parameters. As stated above, global optimization of the tool path for five-axis flank milling requires to approximate the tool

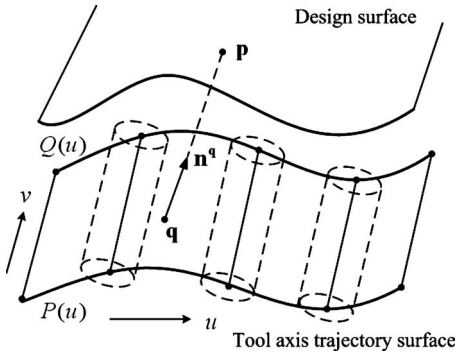

Fig. 4 Tool axis trajectory surface represented as a ruled surface

envelope surface S_{envelope} to the point cloud on the design surface S_{design}, following the minimum zone criterion recommended by ANSI and ISO standards for tolerance evaluation, which states that the deviation measure discussed above should be minimized. For a dense set of data points $\{\mathbf{p}_i \in \mathbb{R}^3, 1 \le i \le n\}$ sampled from S_{design}, this leads to the following min-max problem or Chebyshev approximation problem:

$$\min_{\mathbf{w} \in \mathbb{R}^m} \max_{1 \le i \le n} |d^s_{\mathbf{p}_i, S_{\text{envelope}}}(\mathbf{w})| \tag{16}$$

By introducing one auxiliary variable ξ, problem (16) can be reformulated as the following differentiable constrained optimization problem:

$$\begin{aligned} &\min_{(\mathbf{w},\xi) \in \mathbb{R}^{m+1}} \quad \xi \\ &\text{such that} \quad -\xi \le d^s_{\mathbf{p}_i, S_{\text{envelope}}}(\mathbf{w}) \le \xi, \quad 1 \le i \le n \end{aligned} \tag{17}$$

In problem (17), the constraints explicitly require to reduce the overcut and undercut errors simultaneously. Here, $d^s_{\mathbf{p}_i, S_{\text{envelope}}}(\mathbf{w}) \le 0$ indicates an overcut and $d^s_{\mathbf{p}_i, S_{\text{envelope}}}(\mathbf{w}) \ge 0$ indicates an undercut. For rough milling, nonovercut is the basic requirement, which means that $d^s_{\mathbf{p}_i, S_{\text{envelope}}}(\mathbf{w}) \ge 0$ holds for all $1 \le i \le n$. In a similar way, tool path optimization for rough milling is modeled as the constrained optimization problem,

$$\begin{aligned} &\min_{\mathbf{w} \in \mathbb{R}^m} \max_{1 \le i \le n} d^s_{\mathbf{p}_i, S_{\text{envelope}}}(\mathbf{w}) \\ &\text{such that} \quad d^s_{\mathbf{p}_i, S_{\text{envelope}}}(\mathbf{w}) \ge 0, \quad i = 1, \ldots, n \end{aligned} \tag{18}$$

or, equivalently,

$$\begin{aligned} &\min_{(\mathbf{w},\xi) \in \mathbb{R}^{m+1}} \quad \xi \\ &\text{such that} \quad 0 \le d^s_{\mathbf{p}_i, S_{\text{envelope}}}(\mathbf{w}) \le \xi, \quad i = 1, \ldots, n \end{aligned} \tag{19}$$

As with any nonlinear optimization problem, a 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 (16) is very close to that of the LS approximation problem defined as

$$\min_{\mathbf{w} \in \mathbb{R}^m} \sum_{i=1}^{n} [d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}) - r]^2 \tag{20}$$

Since problem (20) can be solved more easily, its solution can serve as an initial estimate to problem (16). As for problem (20) itself, the required initial tool axis trajectory surface can be generated by interpolating the discrete cutter axes provided by other methods, such as the double points offset (DPO) method [27], three points offset (TPO) method [32], and Chiou's method [30].

3.2 Optimization Algorithms. A number of classical numerical optimization algorithms can be applied to problems (17) and (19) because the derivatives of the involved objective functions and constraint functions are all available. The method of sequential approximation programming has been used successfully on many practical nonlinear constrained optimization problems due to its ease of understanding and implementation [36]. 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 solve problems (17) and (19).

Let $(\mathbf{w}^k, \xi^k)$ be a candidate solution to (19) and consider a perturbation of the form

$$(\mathbf{w}^k + \Delta\mathbf{w}, \xi^k + \Delta\xi) \tag{21}$$

Using Proposition 2, we have the linearized constraint functions

$$\begin{aligned} &d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^k) - [\mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_1}, \ldots, \mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_m}]^T \cdot \Delta\mathbf{w} - r \le \xi^k + \Delta\xi \\ &d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^k) - [\mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_1}, \ldots, \mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_m}]^T \cdot \Delta\mathbf{w} - r \ge -\xi^k - \Delta\xi, \\ &\qquad i = 1, \ldots, n \end{aligned} \tag{22}$$

Obviously, the linearized objective function is equivalent to $\Delta\xi$. Thus, we obtain the corresponding linear program for problem (17) as follows:

$$\begin{aligned} &\min_{(\Delta\mathbf{w},\Delta\xi) \in \mathbb{R}^{m+1}} \quad \Delta\xi \\ &\text{such that} \begin{cases} d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^k) - [\mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_1}, \ldots, \mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_m}]^T \cdot \Delta\mathbf{w} - r \le \xi^k + \Delta\xi \\ d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^k) - [\mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_1}, \ldots, \mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_m}]^T \cdot \Delta\mathbf{w} - r \ge -\xi^k - \Delta\xi \end{cases}, \quad i = 1, \ldots, n \end{aligned} \tag{23}$$

In a similar way, we get the following linear program for problem (19)

$$\begin{aligned} &\min_{(\Delta\mathbf{w},\Delta\xi) \in \mathbb{R}^{m+1}} \quad \Delta\xi \\ &\text{such that} \begin{cases} d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^k) - [\mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_1}, \ldots, \mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_m}]^T \cdot \Delta\mathbf{w} - r \le \xi^k + \Delta\xi \\ d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^k) - [\mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_1}, \ldots, \mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_m}]^T \cdot \Delta\mathbf{w} - r \ge 0 \end{cases}, \quad i = 1, \ldots, n \end{aligned} \tag{24}$$

Normally, it is useful to impose upper bounds on the magnitudes of the variables $\Delta w_i, 1 \le i \le m$ to ensure validity of the linear approximation, which results in the following additional constraints in problems (23) and (24):

$$-w_0 \le \Delta w_i \le w_0, \quad i = 1, \ldots, m \tag{25}$$

where $w_0 > 0$ is an appropriate upper bound value.

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

Algorithm 1 (Global tool path optimization)

Input: Point cloud $\{\mathbf{p}_i \in \mathbb{R}^3, 1 \le i \le n\}$ on design surface S_{design}; initial tool axis trajectory surface $S_{\text{axis}}(\mathbf{w}^0)$; threshold ε specifying the desired accuracy of the algorithm.

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

Step 0:

(1) Set $k=0$;

(2) Compute $d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^0)$, $i=1, \ldots, n$;

(3) Set $\xi^0 = \max_{1 \le i \le n} |d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^0) - r|$.

Step 1:

(1) Solve linear program (23) or (24) to determine the differential increment of the surface shape parameters $\Delta \mathbf{w}$;

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

(3) Compute $d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^{k+1})$, $i=1, \ldots, n$;

(4) Update $\xi^{k+1} = \max_{1 \le i \le n} |d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^{k+1}) - r|$;

(5) If $|1 - \xi^k / \xi^{k+1}| > \varepsilon$, then set $k = k+1$ and go to 1(1); else, exit and report $\mathbf{w}^* = \mathbf{w}^{k+1}$, $\tau_{\text{over}} = \min_{1 \le i \le n} d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^{k+1}) - r$, and $\tau_{\text{under}} = \max_{1 \le i \le n} d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^{k+1}) - r$.

Remark. Based on Proposition 5, we can get the quadratic approximants of the constraint functions around the current intermediate solution, which results in a second-order cone program. The use of cone programming instead of linear programming in above algorithm is worth considering in the future studies.

Problem (20) is a nonlinear LS problem. The Gauss–Newton method can be applied to solve it. Its idea is to solve a sequence of linear LS problems obtained by linearizing the terms $d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w})$, $i=1,2, \ldots, n$, about the intermediate candidate solutions. Let $\mathbf{w}^k$ be a candidate solution to problem (20) and consider a perturbation of the form $\mathbf{w}^k + \Delta \mathbf{w}$. Using Proposition 2, we obtain readily the corresponding linear LS problem,

$$\min_{\Delta \mathbf{w} \in \mathbb{R}^m} \sum_{i=1}^{n} \{d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^k) - [\mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_1}, \ldots, \mathbf{n}^{\mathbf{q}_i} \cdot \boldsymbol{\psi}_{w_m}]^T \cdot \Delta \mathbf{w} - r\}^2 \tag{26}$$

Again, it is useful to control the step-size to ensure validity of the linear approximation. The Levenberg–Marquardt method is such an improved approach. It introduces an adjustable damp term to avoid the singularity of the problem (26) and to make sure that the moving step is small enough so that the value of the objective function of problem (20) monotonously decreases during the iterations. The interested reader can refer to Ref. [36] for a more detailed description.

Another way to solve problem (20) is to use Newton method. With this method, one iteratively approximates the objective function by a quadratic function around the intermediate solution, and then takes a step toward the minimum of that quadratic function. According to Proposition 5, we obtain readily the corresponding quadratic minimization problem,

$$\begin{aligned} \min_{\Delta \mathbf{w} \in \mathbb{R}^m} \sum_{i=1}^{n} & [d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}) - r - (\mathbf{n}^{\mathbf{q}_i})^T [\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}] \Delta \mathbf{w}]^2 \\ & + \frac{d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}) - r}{d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}) - 1/\kappa_1} [(\mathbf{n}_1^{\mathbf{q}_i})^T [\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}] \Delta \mathbf{w}]^2 \\ & + \frac{d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}) - r}{d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}) - 1/\kappa_2} [(\mathbf{n}_2^{\mathbf{q}_i})^T [\boldsymbol{\psi}_{w_1}, \ldots, \boldsymbol{\psi}_{w_m}] \Delta \mathbf{w}]^2 \end{aligned} \tag{27}$$

Based on the above discussions on problem (20), we present the following algorithm for near-optimal solution estimation:

Algorithm 2 (Near-optimal solution estimation)

Input: Point cloud $\{\mathbf{p}_i \in \mathbb{R}^3, 1 \le i \le n\}$ on design surface S_{design}; initial tool axis trajectory surface $S_{\text{axis}}(\mathbf{w}^0)$; threshold ε specifying the desired accuracy of the algorithm.

Output: Near-optimum tool axis trajectory surface $S_{\text{axis}}(\mathbf{w}^*)$.

Step 0:

(1) Set $k=0$;

(2) Compute $d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^0)$, $i=1, \ldots, n$;

(3) Set $\xi^0 = \Sigma_{i=1}^n [d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^0) - r]^2$.

Step 1:

(1) Solve problem (26) or (27) to determine the differential increment of the surface shape parameters $\Delta \mathbf{w}$;

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

(3) Compute $\xi^{k+1} = \Sigma_{i=1}^n [d^s_{\mathbf{p}_i, S_{\text{axis}}}(\mathbf{w}^{k+1}) - r]^2$;

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

4 Examples

Example 1. In order to demonstrate the validity of the proposed method, an example of NC tool path optimization for flank milling of a ruled surface with a cylindrical tool is presented. The ruled surface is expressed in Eq. (28), which has been already tested by several approaches [27–29,32].

$$\mathbf{S}(u,v) = (1-v)[u \quad 20.429 \quad 0]^T + v[u \quad 0.0382u^2 \quad 33.995]^T \tag{28}$$

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

In order to make a comparison with the work presented in Ref. [32], which reported the minimum geometric error, the same cutter, a cylindrical tool with radius of 10 mm, is chosen, and 30 tool locations are calculated with the TOP method [32]. By using the lofting method, an initial tool axis trajectory surface, which interpolates all these 30 tool axes, is generated. Totally, 100×100 points are evenly sampled from the design surface. The optimum tool path is obtained with the optimization algorithm described in Sec. 3. Figures 5 and 6 show the distributions of the geometric errors before and after the optimization, respectively. The maximum undercut (maximum positive point-to-surface distance) reduces from 0.228 mm to 0.068 mm, and the maximum overcut (minimum negative point-to-surface distance) reduces from 0.172 mm 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 the results of Gong et al. [32]. Moreover, our method is applicable to rough milling. The involved nonovercut constraint cannot be modeled with the squared

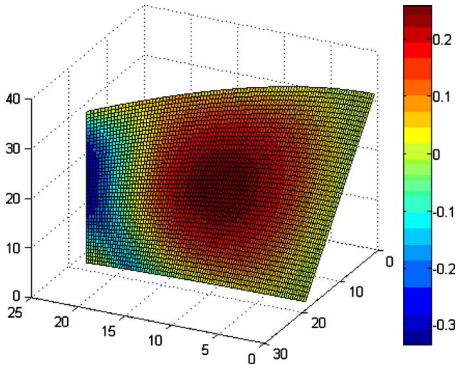

Fig. 5 Distribution of the geometric errors before optimization

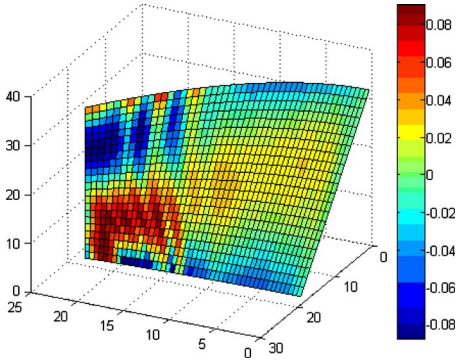

Fig. 6 Distribution of the geometric errors after optimization

distance function. To the best of our knowledge, the rough milling cannot easily be dealt with by the other methods except for Chiou's method [30].

Example 2. Consider a more practical ruled surface, which is a blade of an impeller shown in Fig. 7. The surface is defined by two directrices, which are both B-spline curves of order 3. A cylindrical cutter with the cutter radius of $r=5$ mm is used for the flank milling. With Chiou's method [30], eight cutter locations are calculated. An initial axis trajectory surface is generated by interpolating eight pairs of points on these cutter axes with two B-spline curves of order 3. Totally, 50×100 evenly distributed points are sampled from the design surface, and the tool path is optimized using the approach presented in Sec. 3. The interferences between the tool envelope surface and the design surface before and after the optimization are illustrated in Figs. 8 and 9, respectively. The maximum undercut reduces from 0.0192 mm to 0.0013 mm, and the maximum overcut reduces from 0.0446 mm to 0.0013 mm. It is seen that the global tool path optimization approach improves the machining accuracy greatly.

Example 3. Here, an example of ruled surface reconstruction from unorganized data point cloud is presented. It finds applications in feature-based CAD model reconstruction and piecewise ruled surface approximation of slender surfaces. By evenly sampling the freeform surface expressed in Eq. (29), 36×41 points are generated. The synthetic point cloud is depicted in Fig. 10. Nine discrete rulings are identified using the method presented in Ref. [37]. An initial ruled surface is generated by interpolating nine pairs of points on these rulings with two B-spline curves of order 3. Then, it is deformed to fit the 3D point data by fine-tuning the control points of the two guiding curves using Algorithm 2 described in Sec. 3. The convergence speed of the algorithm is depicted in Fig. 11. It is observed that an optimum solution is obtained within four iterations. The reconstructed ruled surface is shown in Fig. 12,

$$\mathbf{S}(u,v)=\begin{bmatrix} 4u \\ 8v-4 \\ v(4u^2+2)\sin u+(1-v)(-3u^2+8)\cos u+2\sin[u(v-0.5)]\cos(u^2) \end{bmatrix} \tag{29}$$

where $-0.875\le u\le0.875$ and $0\le v\le1$.

5 Conclusions

In this work, the differential properties of the signed distance function are fully explored, and the intercorrelations between the signed and the squared distance functions are clarified. It is demonstrated that the squared distance function studied in the previous works is just the Type I squared distance function deduced from the signed distance function. Also, it is shown that surface approximations under different criteria and constraints can all be formulated as optimization problems with specified requirements on the residual errors represented by the signed distance functions, and the classical numerical optimization algorithms can be directly applied to solve these problems as the derivatives of the involved objective functions and constraint functions are all available. From the point of view of surface approximation, the problem of tool path planning for five-axis flank milling is formulated as that of approximating the machined surface to the designed surface following the minimum zone criterion. By using the signed distance function, the mathematical models and algorithms for tool path optimization for rough and finish millings with cylindrical tools are developed in a unified framework. In comparison with the existing approaches, the one we presented improves the machining accuracy greatly.

It is worth mentioning that flank milling tool path optimization is a special surface fitting problem. It differs from the conventional ones in three ways. First, the surface of interest (tool envelope surface) is the offset surface of another simple surface (tool

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

	Liu [27]	Redonnet et al. [28]	Menzel et al. [29]	Gong et al. [32]	Global optimization (rough milling)	Global optimization (finish milling)
Maximum undercut (mm)	0.582	0.220	0.264	0.093	1.104	0.068
Maximum overcut (mm)	0.585	0.220	0.211	0.119	0	0.067

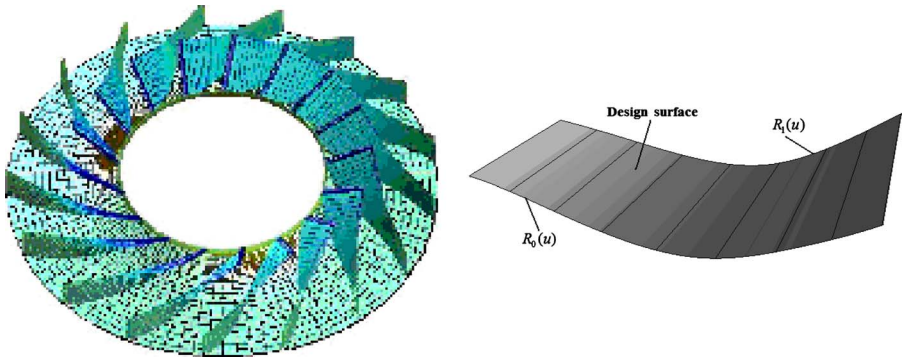

Fig. 7 Surface model of a blade of an impeller

axis trajectory surface). We need to model and solve the problem by just using the simple surface. Second, the optimization criterion for surface approximation is to minimize the l_∞-norm of the residual errors, which is recommended by ANSI and ISO standards for tolerance evaluation. Third, the one-sided constraint is imposed in tool path planning for rough milling. This typical example shows well how a practical surface approximation problem can be handled with the signed distance function. It is seen that the signed distance function is more flexible for surface approximation modeling as compared with the squared distance function. Although not all the theories of signed distance function developed in Sec. 2 are used in this special problem, they can find wide applications in the general surface approximation problems.

Acknowledgment

The authors would like to thank Dr. Hong Gong for the valuable discussions on the problem of global cutter position optimization. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 50775147 and 50835004 and the National Key Basic Research Program under Grant Nos. 2005CB724103 and 2007CB714005.

Appendix A: Proof of Corollary 1

Assume that $f_z \neq 0$, surface $\mathbf{S}(\mathbf{w})$ can be locally represented as $z=z(\mathbf{w},x,y)$, or equivalently, it has a locally parametric description of the form $\psi(\mathbf{w},x,y)=[x,y,z(\mathbf{w},x,y)]^T$.

From $f(x,y,z(\mathbf{w},x,y),\mathbf{w})=0$, we have $f_z z_{w_i}+f_{w_i}=0$ and $z_{w_i}=-f_{w_i}/f_z$. The remainder of the proof is straightforward, and is left to the readers.

Appendix B: Proof of Proposition 3

According to Weingarten's equations, we have

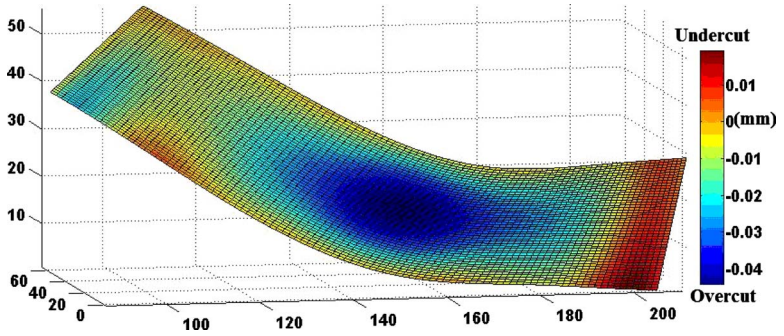

Fig. 8 Interference between the tool envelope surface and the design surface before optimization

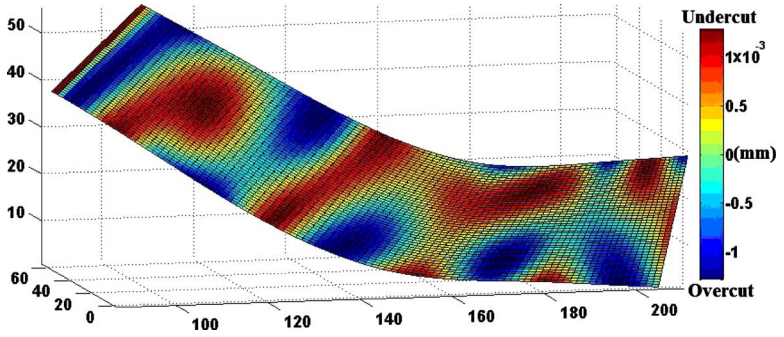

Fig. 9 Interference between the tool envelope surface and the design surface after optimization

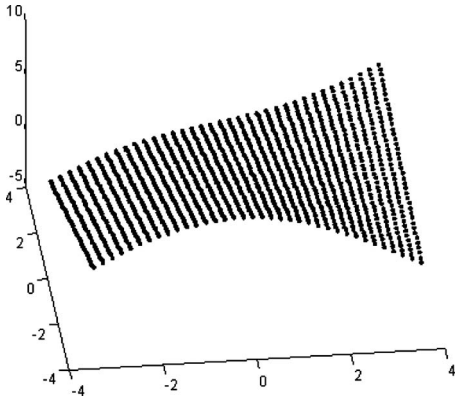

Fig. 10 Synthetic point cloud

$$\Delta\mathbf{n}^{\mathbf{q}} = [\mathbf{n}_u, \mathbf{n}_v]\begin{bmatrix}\Delta u \\ \Delta v\end{bmatrix} = -[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]\mathbf{g}^{-1}\boldsymbol{\Omega}\begin{bmatrix}\Delta u \\ \Delta v\end{bmatrix} \tag{B1}$$

Because $\mathbf{p} - \boldsymbol{\psi}(u^*, v^*) = d^s(\mathbf{p})\mathbf{n}(u^*, v^*)$, it is obvious that

$$[\mathbf{p} - \boldsymbol{\psi}(u^*, v^*)] \cdot \boldsymbol{\psi}_u = 0$$

$$[\mathbf{p} - \boldsymbol{\psi}(u^*, v^*)] \cdot \boldsymbol{\psi}_v = 0 \tag{B2}$$

Differentiating both sides of Eq. (B2), we get

$$[\Delta\mathbf{p} - \boldsymbol{\psi}_u\Delta u - \boldsymbol{\psi}_v\Delta v] \cdot \boldsymbol{\psi}_u + d^s(\mathbf{p})\mathbf{n}(u^*, v^*) \cdot (\boldsymbol{\psi}_{uu}\Delta u + \boldsymbol{\psi}_{uv}\Delta v) = 0$$

$$[\Delta\mathbf{p} - \boldsymbol{\psi}_u\Delta u - \boldsymbol{\psi}_v\Delta v] \cdot \boldsymbol{\psi}_v + d^s(\mathbf{p})\mathbf{n}(u^*, v^*) \cdot (\boldsymbol{\psi}_{vu}\Delta u + \boldsymbol{\psi}_{vv}\Delta v) = 0 \tag{B3}$$

Or, equivalently,

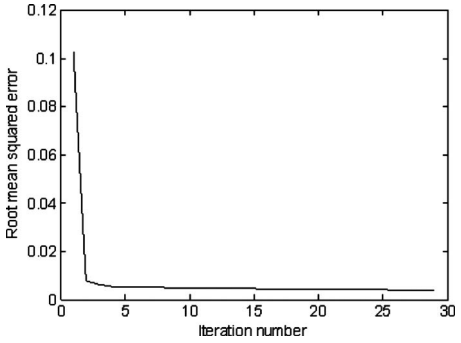

Fig. 11 Root mean squared error versus the number of iterations

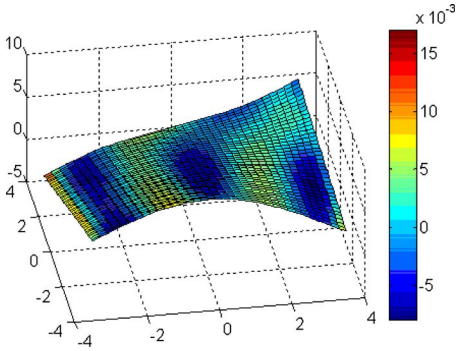

Fig. 12 Reconstructed ruled surface

$$[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]^T\Delta\mathbf{p} - \mathbf{g}\begin{bmatrix}\Delta u \\ \Delta v\end{bmatrix} + d^s(\mathbf{p})\boldsymbol{\Omega}\begin{bmatrix}\Delta u \\ \Delta v\end{bmatrix} = \mathbf{0} \tag{B4}$$

Then we can solve for Δu and Δv from Eq. (B4), i.e.,

$$\begin{bmatrix}\Delta u \\ \Delta v\end{bmatrix} = -(d^s(\mathbf{p})\boldsymbol{\Omega} - \mathbf{g})^{-1}[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]^T\Delta\mathbf{p} \tag{B5}$$

Substituting Eq. (B5) into Eq. (B1) provides

$$\Delta\mathbf{n}^{\mathbf{q}} = [\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]\mathbf{g}^{-1}\boldsymbol{\Omega}\mathbf{A}^{-1}[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]^T\Delta\mathbf{p} \tag{B6}$$

i.e.,

$$\nabla^2 d^s(\mathbf{p}) = [\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]\mathbf{g}^{-1}\boldsymbol{\Omega}\mathbf{A}^{-1}[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]^T \tag{B7}$$

Appendix C: Proof of Corollary 2

Letting

$$\mathbf{g}^{1/2} = \begin{bmatrix}\|\boldsymbol{\psi}_u\| & 0 \\ 0 & \|\boldsymbol{\psi}_v\|\end{bmatrix}$$

we have

$$[\boldsymbol{\psi}_u, \boldsymbol{\psi}_v] = [\mathbf{n}_1, \mathbf{n}_2]\mathbf{g}^{1/2}, \quad \mathbf{g} = \mathbf{g}^{1/2}\mathbf{g}^{1/2}, \quad \mathbf{g}^{-1}\boldsymbol{\Omega} = \begin{bmatrix}\kappa_1 & 0 \\ 0 & \kappa_2\end{bmatrix} \tag{C1}$$

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

$$\mathbf{A} = [d^s(\mathbf{p})\mathbf{I} - (\mathbf{g}^{-1}\boldsymbol{\Omega})^{-1}]\boldsymbol{\Omega} = \begin{bmatrix}d^s(\mathbf{p}) - 1/\kappa_1 & 0 \\ 0 & d^s(\mathbf{p}) - 1/\kappa_2\end{bmatrix}\boldsymbol{\Omega} \tag{C2}$$

Then, it is straightforward to get the final result by substituting all these expressions into the expression of the Hessian matrix $\nabla^2 d^s(\mathbf{p})$ given in Proposition 3.

References

[1] Liu, Y., Pottman, H., and Wang, W., 2006, "Constrained 3D Shape Reconstruction Using a Combination of Surface Fitting and Registration," Comput.-Aided Des., **38**, pp. 572–583.

[2] Sourlier, D., and Bucher, A., 1995, "Surface-Independent, Theoretically Exact Bestfit for Arbitrary Sculptured, Complex, or Standard Geometries," Precis. Eng., **17**(2), pp. 101–113.

[3] Sarkar, B., and Menq, C.-H., 1991, "Smooth-Surface Approximation and Reverse Engineering," Comput.-Aided Des., **23**(9), pp. 623–628.

[4] Li, Z., Gou, J., and Chu, Y., 1998, "Geometric Algorithms for Workpiece Localization," IEEE Trans. Rob. Autom., **14**(5), pp. 864–878.

[5] Pratt, M. J., 1985, "Smooth Parametric Surface Approximation to Discrete Data," Comput. Aided Geom. Des., **2**, pp. 165–171.

[6] Sarkar, B., and Menq, C.-H., 1991, "Parameter Optimization in Approximating Curves and Surfaces to Measurement Data," Comput. Aided Geom. Des., **8**(4), pp. 267–290.

[7] Yau, H. T., and Menq, C. H., 1996, "A Unified Least-Squares Approach to the Evaluation of Geometrical Errors Using Discrete Measurement Data," Int. J. Mach. Tools Manuf., **36**(11), pp. 1269–1290.

[8] Yau, H. T., and Chen, J. S., 1997, "Reverse Engineering of Complex Geometry Using Rational B-Splines," Int. J. Adv. Manuf. Technol., **13**(8), pp. 548–555.

[9] Ma, W., and Kruth, J. P., 1995, "Parameterization of Randomly Measured Points for Least Squares Fitting of B-Spline Curves and Surfaces," Comput.-Aided Des., **27**(9), pp. 663–675.

[10] Ma, W., and Kruth, J. P., 1998, "NURBS Curve and Surface Fitting for Reverse Engineering," Int. J. Adv. Manuf. Technol., **14**, pp. 918–927.

[11] Piegl, L., and Tiller, W., 2001, "Parameterization for Surface Fitting in Reverse Engineering," Comput.-Aided Des., **33**, pp. 593–603.

[12] Weiss, V., Andor, L., Renner, G., and Várady, T., 2002, "Advanced Surface Fitting Techniques," Comput. Aided Geom. Des., **19**, pp. 19–42.

[13] Pottmann, H., and Hoffer, M., 2003, "Geometry of the Squared Distance Function to Curves and Surfaces," *Visualization and Mathematics III*, H. Hege and K. Polthier, eds., pp. 223–244.

[14] Pottmann, H., and Leopoldseder, S., 2003, "A Concept for Parametric Surface Fitting Which Avoids the Parametrization," Comput. Aided Geom. Des., **20**, pp. 343–362.

[15] Pottmann, H., Leopoldseder, S., and Hofer, M., 2004, "Registration Without ICP," Comput. Vis. Image Underst., **95**, pp. 54–71.

[16] Wang, W. P., Pottman, H., and Liu, Y., 2006, "Fitting B-Spline Curves to Point Clouds by Curvature-Based Squared Distance Minimization," ACM Trans. Graphics, **25**(2), pp. 214–238.

[17] Zhu, L. M., 2002, "Distance Function Based Models and Algorithms for Fit-

ting of Geometric Elements to Measured Coordinate Points," Ph.D. thesis, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan.
[18] Zhu, L. M., and Ding, H., 2004, "A Unified Approach for Least-Squares Surface Fitting," Sci. China, Ser. G, **47**(SI), pp. 72–78.
[19] Zhu, L. M., Xiong, Z. H., Ding, H., and Xiong, Y. L., 2004, "A Distance Function Based Approach for Localization and Profile Error Evaluation of Complex Surface," ASME J. Manuf. Sci. Eng., **126**(3), pp. 542–554.
[20] Zhu, L. M., Ding, H., and Xiong, Y. L., 2003, "Distance Function Based Algorithm for Spatial Straightness Evaluation," Proc. Inst. Mech. Eng., Part B, **217**, pp. 931–939.
[21] Hoppe, H., ReRose, T., Duchamp, T., McDonald, J., Stuetzle, W., 1992, "Surface Reconstruction From Unorganized Points," SIGGRAPH, pp. 71–78.
[22] Flöry, S., and Hofer, M., 2010, "Surface Fitting and Registration of Point Clouds Using Approximations of the Unsigned Distance Function," Comput. Aided Geom. Des., **27**, pp. 60–77.
[23] Zhou, J., Sherbrooke, E. C., and Patrikalakis, N. M., 1993, "Computation of Stationary Points of Distance Functions," Eng. Comput., **9**(4), pp. 231–246.
[24] Pigel, L., and Tiller, W., 1995, *The NURBS Book*, Springer, Berlin.
[25] Elber, G., and Fish, R., 1997, "5-Axis Freeform Surface Milling Using Piecewise Ruled Surface Approximation," ASME J. Manuf. Sci. Eng., **119**(3), pp. 383–387.
[26] Subag, J., and Elber, G., 2006, "Piecewise Developable Surface Approximation of General NURBS Surfaces With Global Error Bounds," Lect. Notes Comput. Sci., **4077**, pp. 143–156.
[27] Liu, X. W., 1995, "Five-Axis NC Cylindrical Milling of Sculptured Surfaces," Comput.-Aided Des., **27**(12), pp. 887–894.
[28] Redonnet, J. M., Rubio, W., and Dessein, G., 1998, "Side Milling of Ruled Surfaces: Optimum Positioning of the Milling Cutter and Calculation of Interference," Int. J. Adv. Manuf. Technol., **14**(7), pp. 459–465.
[29] Menzel, C., Bedi, S., and Mann, S., 2004, "Triple Tangent Flank Milling of Ruled Surfaces," Comput.-Aided Des., **36**(3), pp. 289–296.
[30] Chiou, J. C., 2004, "Accurate Tool Position for Five-Axis Ruled Surface Machining by Swept Envelope Approach," Comput.-Aided Des., **36**(10), pp. 967–974.
[31] Lartigue, C., Duc, E., and Affouard, A., 2003, "Tool Path Deformation in 5-Axis Flank Milling Using Envelope Surface," Comput.-Aided Des., **35**(4), pp. 375–382.
[32] Gong, H., Cao, L. X., and Liu, J., 2005, "Improved Positioning of Cylindrical Cutter for Flank Milling Ruled Surfaces," Comput.-Aided Des., **37**(12), pp. 1205–1213.
[33] 1983, Technical Drawings-Geometrical Tolerancing, ISO/R 1101.
[34] 1982, Standard A. N. S. I. Y14.5, Dimensioning and Tolerancing, The American Society of Engineers.
[35] Podshivalov, L., and Fischer, A., 2008, "Modeling an Envelope Generated by 3D Volumetric NC Tool Motion," Int. J. Adv. Manuf. Technol., **38**(9–10), pp. 949–957.
[36] Nocedal, J., and Wright, S. J., 1999, *Numerical Optimization*, Springer, New York.
[37] Chen, H. Y., and Pottmann, H., 1999, "Approximation by Ruled Surfaces," J. Comput. Appl. Math., **102**(1), pp. 143–156.