Towards G^1-Continuous Multi-Strip Path-Planning for 5-Axis Flank CNC Machining of Free-Form Surfaces Using Conical Cutting Tools

Rajain, Kanika; Bizzarri, Michal; Lávička, Miroslav; Kosinka, Jiří; Bartoň, Michael

Published in:
CAD Computer Aided Design

DOI:
10.1016/j.cad.2023.103555

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2023

Computer-Aided Design 163 (2023) 103555

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Towards G^1-Continuous Multi-Strip Path-Planning for 5-Axis Flank CNC Machining of Free-Form Surfaces Using Conical Cutting Tools

Kanika Rajain [a], Michal Bizzarri [c,a,*], Miroslav Lávička [c], Jiří Kosinka [d], Michael Bartoň [a,b]

[a] BCAM – Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Basque Country, Spain
[b] Ikerbasque – Basque Foundation for Sciences, Maria Diaz de Haro 3, 48013 Bilbao, Basque Country, Spain
[c] Department of Mathematics, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 301 00 Plzeň, Czech Republic
[d] Bernoulli Institute, University of Groningen, Nijenborgh 9, 9747 AG, Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 16 April 2023
Accepted 13 May 2023

Keywords:
5-axis CNC machining
Conical or cylindrical tools
Finishing operations
Tangential movability
Free-form shape manufacturing
Tool path-planning

ABSTRACT

Existing flank milling path-planning methods typically lead to tiny gaps or overlaps between neighboring paths, which causes artifacts and imperfections in the workpiece. We propose a new multi-strip path-planning method for 5-axis flank milling of free-form surfaces which targets G^1 (tangent-plane) continuity of the neighboring strips along shared boundaries. While for some geometries one cannot achieve G^1 continuity and high approximation quality at the same time, our optimization framework offers a good trade-off between machining accuracy in terms of distance error and the G^1 connection of neighboring strips. We demonstrate our algorithm on synthetic free-form surfaces as well as on industrial benchmark datasets, showing that we are able to meet fine industrial tolerances and simultaneously significantly reduce the kink angle of adjacent strips, and consequently to improve the surface finish in terms of smoothness.

1. Introduction

Highly accurate manufacturing is essential in many industrial sectors, automotive or aeronautical to name a few, as the functionality and/or the life spans of manufactured objects is heavily affected by the accuracy and smoothness of their surface finish. Industrial objects, such as turbine blades, blisks, or gears (see Fig. 1), need to be stiff and are therefore typically manufactured from a single material block, with 5-axis Computer Numerically Controlled (CNC) machining being the leading subtractive technology to achieve this.

5-axis CNC machining typically consists of three main stages: (i) roughing, (ii) semi-finishing, and (iii) finishing. Our work deals with the finishing stage where the to-be-manufactured object is being finished by the side of the cutting tool, which is known as *flank* (aka side) milling; see Fig. 2(a). In this final stage, the critical issue is not only the machining accuracy in terms of distance error, but also the continuity/smoothness of neighboring milling paths.

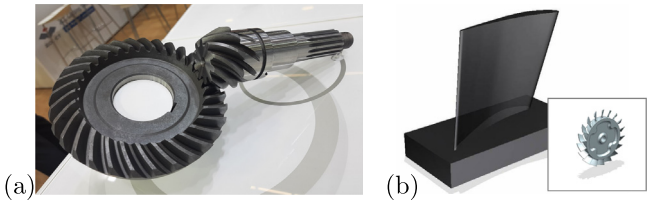

Fig. 1. Industrial benchmarks. (a) A spiral bevel gear and its pinion. (b) A single blade of a blisk (right frame).

Traditional methods, as well as the state-of-the-art industrial software, consider mostly conical or cylindrical tools for this finishing stage as the majority of the cutting tools are, in essence, straight. However, industrial benchmarks, such as blisks or gears, are not straight and therefore the path-planning stage becomes very challenging when one aims to use straight tools to approximate curved geometries. Inevitably, one needs many (tens to hundreds) of flank-milling paths to reach fine industrial tolerances (usually tens of microns, depending on the particular workpiece).

The main challenge, however, is not only the accuracy in terms of distance between the designed and the machined surface. The surface finish quality also depends on the smoothness across the neighboring machining paths. One can see clear imperfections in

* Corresponding author at: Department of Mathematics, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 301 00 Plzeň, Czech Republic.
E-mail addresses: kanika@bcamath.org (K. Rajain), bizzarri@kma.zcu.cz (M. Bizzarri), lavicka@kma.zcu.cz (M. Lávička), j.kosinka@rug.nl (J. Kosinka), mbarton@bcamath.org (M. Bartoň).

https://doi.org/10.1016/j.cad.2023.103555
0010-4485/

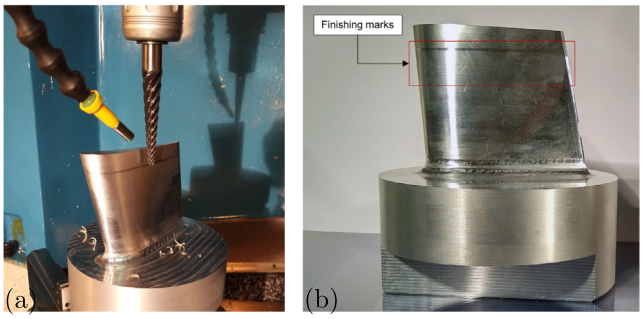

Fig. 2. (a) Flank milling of a blisk blade using a conical cutter. (b) Imperfections of flank milling. The blade of the blisk has been finished by more than 20 paths with a conical tool, using a commercial software. Note the non-smooth light reflections and the horizontal artifact (red frame). Illustrations courtesy of [1]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 2(b), where several tens of sweeps were applied using a given conical cutter. These imperfections are related to the fact that adjacent paths of the tool are not perfectly aligned in terms of their normal fields (and thus not G^1 continuous).

Two neighboring paths/motions of the tool form two envelopes, and these envelopes typically intersect under a small angle. This angle, sometimes called the *kink* angle, measures the deviation of the two surface normals along the intersection (aka boundary) curve. We address this important machining issue by designing motions of a given conical tool such that the neighboring envelopes are (numerically) G^1 continuous (i.e., the deviation of the kink angle is aimed to be as small as possible). In the case of relatively flat geometries, we show that we are able to construct paths that are almost G^1 continuous, up to a tiny kink angle deviation. In the case of highly complex surfaces, where G^1 continuous strips are not (globally) possible, we propose a trade-off between the machining accuracy and the kink angle deviation, controlled via weights in our optimization-based framework.

The remainder of the paper is organized as follows. We start by surveying related prior work in Section 2. Section 3 formulates the problem and states our assumptions, and Section 4 gives a quick overview of basic notions such as envelopes of conical tools and how to compute them. Section 5 describes how the desired machining directions are controlled, and Section 6 details our global optimization approach. Section 7 demonstrates our algorithm on several benchmark examples, and finally Section 8 concludes the paper.

2. Related work

Geometric modeling of 3- and 5-axis CNC machining is a prime example of Computer-Aided Design (CAD) in action where one simulates a process, relatively fast and with minimal cost, before going to expensive and time-consuming physical realizations. It is not surprising that modeling of CNC machining has attracted attention of many researchers in the past several decades, in particular the geometrically more challenging five axis variant [2–6], including non-standard cutting tools [7–10].

Our work belongs to the category of 5-axis flank (aka peripheral) milling where the milling tool moves along the material block tangentially. This tangential side contact makes the problem difficult as one typically has only a few degrees of freedom to position the tool, while the side contact imposes infinitely many constraints (tangential contact at infinitely many points of the characteristic). The tools frequently used for 5-axis flank CNC machining are typically conical or cylindrical, basically leading to solutions where one approximates the input surface by a ruled surface, and then approximates the ruled surface by a motion of a cone/cylinder, the rulings being the lines of contact [11]. Such solutions are possible, however, they reduce the solution space towards very special instantaneous motions (e.g. translations). This is because even for the straight tools (cylinders or cones) the characteristic is not, in general, a straight line, but an algebraic curve of degree four: the intersection of the cone (cylinder) with its derivative in time [12].

For cylindrical tools, one can think of the offset surface (offsetting distance being the radius of the cylinder) and a ruled surface, a trajectory of the axis of the cylinder. Therefore, a lot of work in the CAD literature is devoted to free-form surface approximation by ruled surfaces [13–18]. However, such a conceptual simplification is not possible for cones and one has to look for alternative approaches. One typically ends up with an optimization problem that minimizes the error between the points on the characteristic and the surface [14,19,20].

To find good initial tangential positions of a cone w.r.t. a given surface, a second order approximation of the point-surface distance can be used [21]. Such an approach gives a closed-form solution for the direction of the axis for a given point of contact on the input surface. However, in order to get a good position of the axis, one needs to integrate, in general, a multi-valued vector field and look for straight segments, which is computationally expensive. An alternative approach for a good initial position of a given cone is presented in [22]. Since a cone can be seen as a one-parameter family of tangent planes, one can use the isotropic model of Laguerre geometry and map the cone into an isotropic circle and study a higher order contact between the circle and the isotropic image of the input surface, which is conceptually simpler than the surface-surface case.

Another class of research considers curved tools for flank milling, either of a given shape [7,23], or custom-shaped [24–28]. These approaches do not look only for the motion of the tool as a variable, but optimize also the shape of the tool itself. With recent advances in additive manufacturing, one can 3D-print a custom-shaped tool specifically designed for a particular geometry and, naturally, such a tools performs better than a straight (conical or cylindrical) tool [27].

The most related research to our work is the multi-pass-planning method for 5-axis flank milling of free-form surfaces introduced recently [29]. The reference surface is partitioned based on a tangent vector field and each partitioned patch is approximated by a set of piecewise quad strips. These strips are constructed to be quasi-developable, which allows good approximation by motions of cones. However, the proposed approach guarantees only G^0 continuity along adjacent strips. In contrast, we target G^1 continuity, and in geometries where this is not possible (e.g. highly convex or concave patches), we offer a good trade-off between G^1 continuity and approximation quality.

A global collision detection algorithm that incorporates normal curvatures of the input surface was presented in [30]. Given a convex shaped tool, collision free 5-axis tool-paths are generated using the normal curvature bounds of a surface, without calculating surface offsets. The advantage of the method is that there is no sampling involved, which allows the tool-paths to be globally optimized while guaranteeing to be globally collision free.

The real machining process also involves other factors than the approximation error between the design and manufactured surfaces. One such example is the cutting force, which describes how much force has to be applied to the milling tool; it is preferably low as high forces are typically strongly correlated with the milling tool vibrations. Mathematical modeling of the cutting force under flank milling is also an active research area, see e.g. [31–35] and the references cited therein. Looking for tool-paths that minimize (the mean value of) the tool deflection force

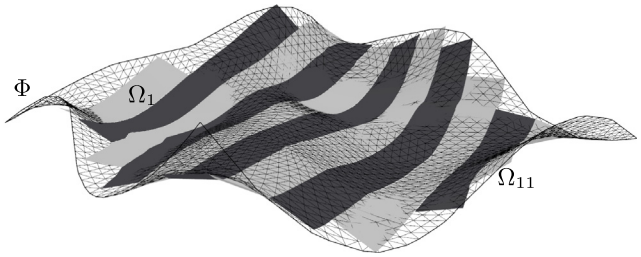

Fig. 3. A free-from surface Φ, represented as a triangular mesh, is approximated by $M = 11$ envelopes Ω_i of a fixed conical tool such that neighboring strips meet with G^1 continuity.

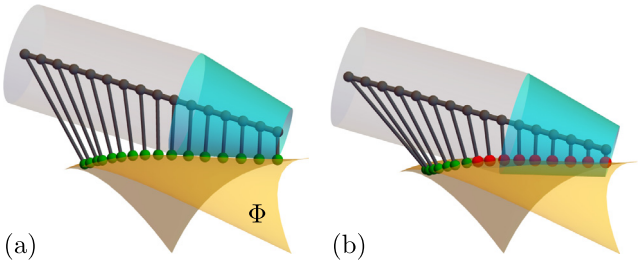

Fig. 4. Global collision test. Footpoints on Φ passing the distance check (green) and those that fail (red) are shown. When all points pass the check (a), the position is marked as non-colliding. If at least one point fails the check, the position is marked as colliding (b) and is eliminated from the list. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

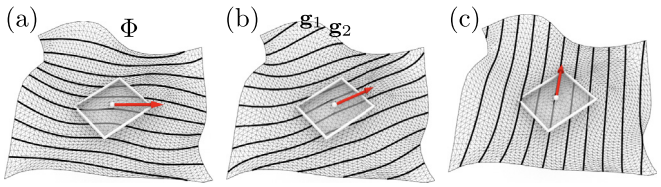

Fig. 5. Machining direction control. Given a free-form surface Φ, a tangent vector (red) at its central point, and an approximate distance between neighboring paths, a set of guiding curves $\mathbf{g}_i$ is computed. The direction of the tangent vector indicates the preferable machining direction and the guiding curves are initializations of the boundaries between neighboring envelopes of a given conical tool. The sets of guiding curves for three different guiding directions are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

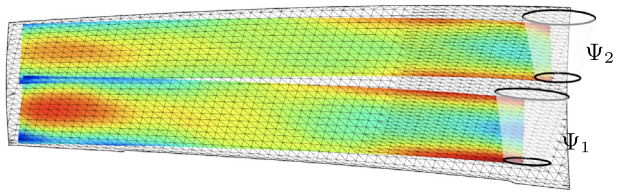

Fig. 6. Challenge towards a single tool. Two milling paths of two different custom-shaped tools Ψ_1 and Ψ_2 on one flank of a spiral bevel gear, cf. Fig. 1(a), are shown [42]. The milling strips are color-coded by the signed distance to the reference surface (wire-frame). Our objective is to optimize the tool and the path towards a single conical tool and G^1-continuous envelopes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

were studied in [36]. Another physical factor strongly correlated with the cutting force is not only the vibration of the milling tool but also of the milled object itself. The vibration phenomenon of thin impeller blades during flank milling is discussed in [33].

Other important factors in high-performance machining are high velocities and accelerations, which both need to be controlled throughout the milling process [37–39]. An algorithm that seeks the optimal feedrate interpolation, considering both tangential and in-axis jerk of the milling tool, was presented in [37]. A trade-off between the geometric accuracy and smoothness of the motion was discussed in [40]. The reader is also referred to the survey paper [41] for other relevant literature on flank 5-axis CNC milling. In our contribution, we do not consider these physical issues and focus solely on the geometric approximation between the design surface and the milled conical envelope. In contrast to [20,25], we consider conical tools, but our computational framework supports fixed conical tools as well as those resulting from our optimization process.

3. Problem formulation

Given a reference free-form surface Φ and a conical tool Ψ, find a set of M motions of Ψ such that the M envelopes $\Omega_1, \ldots, \Omega_M$ share, pairwise, the normal fields along the common boundaries (i.e., they form a G^1 collection of strips), and approximate Φ as closely as possible; see Fig. 3.

Since there might be several possible machining directions, we assume, as additional input, that there is a preferred, dominant, direction in which the tool should move. Such an assumption is in accordance with the machining practice as there are typically admissible directions for the axis of the tool, arising form the collision detection analysis. We do not incorporate collision detection in our setup, however, we validate, as a post-process, that the extended axis of the tool towards the shank does not collide with the reference geometry; see Fig. 4.

We support two options for this additional input: either (i) Φ is equipped with a smooth tangent vector field that indicates the desired motions of the tool, or (ii) we are given a single tangent vector at a central point of Φ. While (i) is more precise in terms of the desired motions, (ii) gives a simple tool for a user to easily control the desired motion; see Fig. 5.

We consider only one family of strips, i.e., we do not support T-junctions or more complex configurations. However, the individual strips can differ in length, depending on the input geometry; see Fig. 3. We also consider full engagement of the tool, i.e., the tool is assumed to be involved as a whole, not only a tip or a part of the cone.

Even though the cutting tool is not exactly a cone (see again Fig. 2(a)), the tool spins around its axis at a very high speed, and therefore we conceptualize the tool as a truncated cone in our considerations.

Although it would be conceptually possible to handle multiple tools, we assume that only a single truncated cone Ψ is used. This, again, is in accordance with the real-life machining practice, where a single tool is preferable.

To further motivate our approach, we show results from [42] in Fig. 6 where one flank of a spiral bevel gear is approximated by motions of two custom-shaped tools Ψ_1 and Ψ_2. Even though only two strips are needed to meet machining tolerance close to 13 μm, there are two different tools, which first overlap at the beginning of the milling paths and then form a gap between the strips. Our objective is to offer a path-planning algorithm that will lead to only one tool and, more importantly, will stitch together the neighboring strips in a G^1 fashion.

4. Preliminaries

Consider as a machining tool a truncated cone Ψ, which can be thought of as the envelope of a one-parameter family of spheres

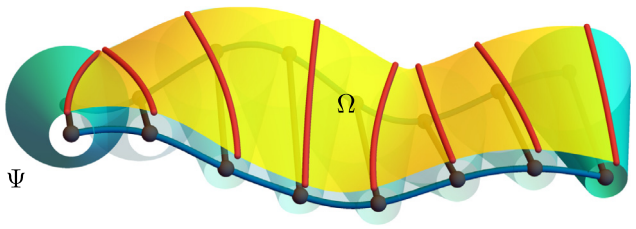

Fig. 7. An envelope Ω of a moving tool Ψ. General characteristics (red curves) depend on the instantaneous motion and therefore change dynamically in time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

centered along the axis

$$\mathbf{l}(s) = (1-s)\mathbf{a} + s\mathbf{b}, \quad s \in [0,1]. \tag{1}$$

The radii of the spheres are determined by the linear *radial function*

$$r(s) = (1-s)r_1 + sr_2, \quad s \in [0,1]. \tag{2}$$

We assume that the point $\mathbf{a}$ corresponds to the tool tip, and $\mathbf{b}$ to the base of the tool (i.e., closer to the shank), hence $r_1 \leq r_2$. Under this convention, it is sufficient to think of a motion of Ψ as a ruled surface $\mathbf{y}$ which is defined as

$$\mathbf{y}(s,t) = (1-s)\mathbf{a}(t) + s\mathbf{b}(t), \qquad s,t \in [0,1], \tag{3}$$

where $\mathbf{a}(t)$ and $\mathbf{b}(t)$ are known as the *rail curves*. As the ruled surface represents a motion of a *rigid* line, it holds

$$L = \|\mathbf{a}(t) - \mathbf{b}(t)\|, \qquad \forall t \in [0,1], \tag{4}$$

which will be further referred to as the *rigidity constraint*. Parameter s corresponds to the ruling direction, while t can be interpreted as time (or pseudotime).

The ruled surface $\mathbf{y}(s,t)$ describes a motion of the tool axis $\mathbf{l}(s)$ in time t. The tool Ψ creates (under this motion) the envelope Ω, see Fig. 7, which can be parameterized as

$$\Omega : \mathbf{x}^{\pm} = \mathbf{y} - r\,\mathbf{n}^{\pm}, \tag{5}$$

where

$$\mathbf{n}^{\pm} = \frac{\Delta_r(G\mathbf{y}_s - F\mathbf{y}_t) \pm (\mathbf{y}_s \times \mathbf{y}_t)\sqrt{(E-\Delta_r^2)G - F^2}}{EG - F^2} \tag{6}$$

is the normal vector of the envelope. Here, $\Delta_r = r_2 - r_1$, and $\mathbf{y}_s$ and $\mathbf{y}_t$ denote the partial derivative of $\mathbf{y}$ with respect to s and t, respectively, and E, F, G are the components of the first fundamental form of $\mathbf{y}(s,t)$. The envelope formula (6) is a special instance of in [43, formula (4)] because $\mathbf{y}(s,t)$ is linear in s and the radii of the spheres are constant in time (parameter t) and linear in the direction of the rulings (parameter s).

In general, there are two branches of the envelope Ω, the *upper* envelope $\mathbf{x}^+$ and the *lower* envelope $\mathbf{x}^-$, see (5) and (6). As our motivation is the CNC application, we are interested only in that branch that is closer to the reference surface Φ. Therefore, if there is no danger of confusion, we omit the superscript and write only $\mathbf{x}$, which is assumed to be the correct branch (and analogously with the normal vector field $\mathbf{n}$). At each time instant t, the tool Ψ touches envelope Ω along the *characteristic curve* $\mathbf{ch}(s) = \mathbf{x}(s, t = const.)$. This curve is in general not static; its shape depends on the instantaneous motion of the tool and therefore changes in time as the instantaneous motion changes; see Fig. 7.

Our objective is to construct a set of ruled surfaces $\mathbf{y}_i$, $i = 1,\ldots,M$ (for a given conical tool) such that two adjacent envelope strips Ω_i are connected with G^1 continuity (see Fig. 8) and also closely approximate the input surface Φ.

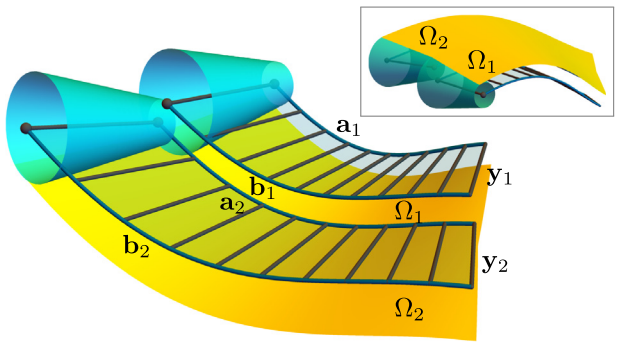

Fig. 8. A G^1 continuous join of two envelope strips Ω_1 and Ω_2. The ruled surfaces $\mathbf{y}_1$ and $\mathbf{y}_2$ are constructed such that the corresponding envelope strips Ω_1 and Ω_2 are connected with G^1 continuity.

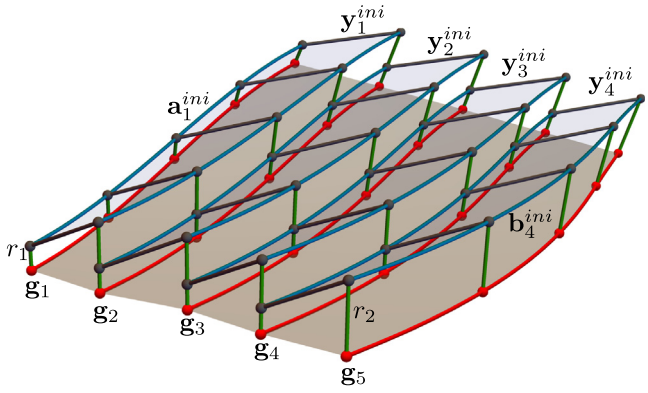

Fig. 9. Guiding curves. Five guiding curves $\mathbf{g}_i$, $i = 1,\ldots,5$ serve as projections of the initial rail curves $\mathbf{a}_i^{ini}, \mathbf{b}_i^{ini}$, $i = 1,\ldots,4$ onto the target surface Φ. Therefore, $\mathbf{a}_i^{ini}$ and $\mathbf{b}_i^{ini}$ are computed as offset curves Φ along $\mathbf{g}_i$ with radii r_1 and r_2, respectively. Rail curves $\mathbf{a}_i^{ini}$ and $\mathbf{b}_i^{ini}$ then define initial ruled surfaces $\mathbf{y}_i^{ini}$, $i = 1,\ldots,4$ describing initial positions of the axes of the given conical tool.

5. Initial positions of the tool

In this section, we look for initial positions of the tool, i.e., we construct M ruled surfaces $\mathbf{y}_i^{ini}(s,t)$, $i = 1,\ldots,M$, cf. Eq. (3), such that the corresponding envelopes Ω_i^{ini} are close enough to the target surface Φ and two neighboring envelope strips are close to one another. These positions will serve as initial guesses in our optimization-based framework (Section 6), where we aim to improve the approximation error and also to fix the tiny gaps/overlaps of the neighboring strips to be joined with G^1 continuity.

For the construction of $\mathbf{y}_i^{ini}(s,t)$ we employ the so-called *guiding curves* $\mathbf{g}_i(t) \subset \Phi$, $i = 1,\ldots,M+1$, serving as orthogonal projections of the initial sought-after rail curves $\mathbf{a}_i^{ini}(t)$ and $\mathbf{b}_i^{ini}(t)$, $i = 1,\ldots,M$ onto Φ. The guiding curves $\mathbf{g}_i(t)$ are computed using an approach based on an idea from [10]. In particular, we represent $\mathbf{g}_i(t)$ as level sets of a scalar function $G : \Phi \to \mathbb{R}$, i.e., G is constant along every $\mathbf{g}_i(t)$. The function G is computed within an optimization-based framework, where we require G to be fair and with a constant gradient perpendicular to a prescribed guiding direction. In the case when a tangent vector field is given (instead of a single vector), ∇G is required to be perpendicular to this vector field. As a result, we obtain a set of fair guiding curves $\mathbf{g}_i(t)$ in a given direction and distributed along Φ in a quasi-parallel fashion; see Fig. 9 and recall Fig. 5.

To satisfy the requirement that the guiding curves are projections of the rail curves, the individual curves (level sets) are chosen so that their mutual distance is equal to $d = \sqrt{L^2 - \Delta_r^2}$.

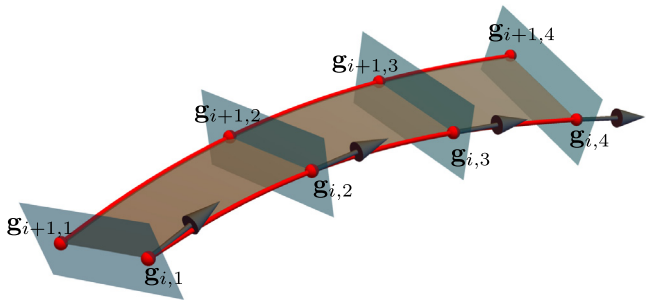

Fig. 10. Construction of the base points on two neighboring guiding curves $\mathbf{g}_i$ and $\mathbf{g}_{i+1}$. The points $\mathbf{g}_{i,j}$ and $\mathbf{g}_{i+1,j}$ both lie in the normal planes (transparent) of $\mathbf{g}_i$ at $\mathbf{g}_{i,j}$.

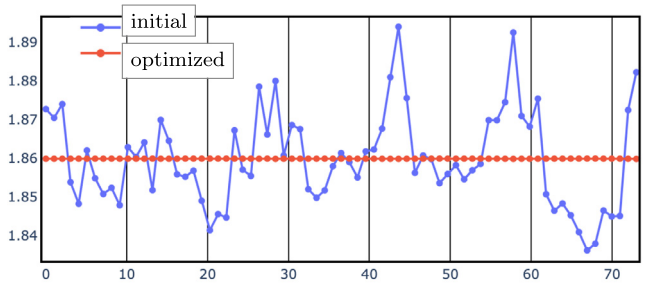

Fig. 11. Initial and optimized axis length. The construction of two ruled surfaces via (7) returns axes with varying length. The histogram shows the initial tool lengths (blue) and optimized (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Using the above approach, we obtain the guiding curves as fine polylines. However, we need to work with fewer variables during optimization (Section 6). Therefore, we choose the so-called *base points* $\mathbf{g}_{1,j}$ on $\mathbf{g}_1$ such that $\|\mathbf{g}_{1,j} - \mathbf{g}_{1,j+1}\| = h$, where h is a chosen step size; we set h as $1/100$ of the bounding box of the reference surface in our implementation. To ensure that the parameterizations $\mathbf{g}_{i+1}$ and $\mathbf{g}_i$, $i = 1, \ldots, M$, are aligned, the base points of $\mathbf{g}_{i+1,j}$ are obtained as the intersections of the normal planes of $\mathbf{g}_i$ at $\mathbf{g}_{i,j}$ with $\mathbf{g}_{i+1}$; see Fig. 10.

Remark 1. In our algorithm, we allow adjacent envelope strips to have different lengths; recall Fig. 3. This should thus be included in the considerations when computing the base points on the guiding curves. In more detail, it may be that there is no intersection of the guiding curve with any particular normal plane. Or, conversely, some base points may need to be added because they cannot be obtained as intersections with normal planes.

As the next step, we construct the rail curves $\mathbf{a}_i$ and $\mathbf{b}_i$, $i = 1, \ldots, M$, being the r_1- and r_2-offset curves of Φ along $\mathbf{g}_i$ and $\mathbf{g}_{i+1}$. In particular, we compute the base-points:

$$\begin{aligned} \mathbf{a}_{i,j} &= \mathbf{g}_{i,j} + r_1\, \mathbf{n}_{i,j}, \\ \mathbf{b}_{i,j} &= \mathbf{g}_{i+1,j} + r_2\, \mathbf{n}_{i+1,j}, \end{aligned} \tag{7}$$

where $i = 1, \ldots, M$, $j = 1, \ldots, N$ and $\mathbf{n}_{i,j}$ are the unit normal vectors of Φ at $\mathbf{g}_{i,j}$; see Fig. 9.

The construction via (7) is in accordance with the requirement that two neighboring strips should meet along their shared $\mathbf{g}_i$ curve with at least C^0 continuity. A (discrete) conical tool can be seen as a pair of spheres centered along the $\mathbf{a}_i$ and $\mathbf{b}_i$ curves, with radii r_1 and r_2, respectively. Since the tip of the tool in one sweep and the bottom of the tool in the following sweep need to nicely approximate Φ along the guiding curve $\mathbf{g}_i$, the two spheres have to move tangentially in a synchronized way along $\mathbf{g}_i$.

Remark 2. Note that the initial ruling surfaces $\mathbf{y}_i^{ini}$ do not satisfy the rigidity constraint (4). However, this is corrected in Section 6, where the rigidity constraint has the highest priority in the optimization; see Fig. 11. The histogram shows all lengths of the rulings (tool's axis) for two strips on the gear geometry shown in Fig. 6. Observe a minor variation of the axis' lengths that got optimized perfectly.

6. Optimized positions of the tool

In this section, we describe a global optimization method that we apply to the initial tool positions. This leads to tool motions producing envelope strips that are globally G^1 continuous, cover most of the target surface, and approximate it sufficiently well. For geometries where the G^1 continuity is in contradiction with the approximation quality (such as on highly curved input surfaces), our optimization-based framework offers a compromise, with the option to prioritize the desired objective (e.g. the G^1 continuity); this is facilitated in an intuitive fashion using optimization weights.

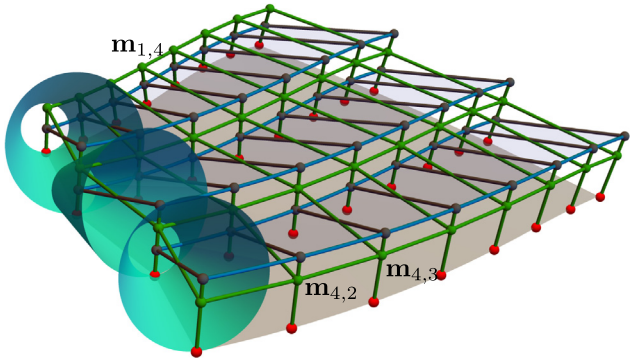

Fig. 12. Quad structure. Instead of describing the tool motions using the base points $\mathbf{a}_{i,j}$, $\mathbf{b}_{i,j}$ on the rail curves, we use a grid of points $\mathbf{m}_{i,j}$ to describe the whole structure. The grid points (green) create a quadrilateral mesh uniquely describing the particular orientations of the tool covering the whole surface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In order to reduce the number of variables in the optimization process, we define the so-called *grid points* $\mathbf{m}_{i,j}$ (or $\mathbf{m}$-*grid* for short) as the average of $\mathbf{a}_{i+1,j}$ and $\mathbf{b}_{i,j}$, that is

$$\mathbf{m}_{i+1,j} = \frac{\mathbf{a}_{i+1,j} + \mathbf{b}_{i,j}}{2}, \quad i = 1, \ldots, M-1. \tag{8}$$

Note that $\mathbf{m}_{i+1,j}$, $\mathbf{a}_{i+1,j}$ and $\mathbf{b}_{i,j}$ lie on the common normal of Φ with the normal vector $\mathbf{n}_{i+1,j}$; see Fig. 12. These grid points form a quad mesh that lies on the offset surface of Φ at distance $(r_1 + r_2)/2$. On the boundary where the $\mathbf{a}$ and $\mathbf{b}$ points are not defined, i.e., the very first path of the tool tip and the very last path of the bottom of the tool (see Fig. 12), the grid points $\mathbf{m}_{1,j}$ and $\mathbf{m}_{M+1,j}$ are defined analogously via the surface normal, to make the description of the whole quad structure uniform. That is, we define the boundary quad points as

$$\begin{aligned} \mathbf{m}_{1,j} &= \mathbf{a}_{1,j} + \frac{\Delta_r}{2}\mathbf{n}_{1,j}, \\ \mathbf{m}_{M+1,j} &= \mathbf{b}_{M,j} - \frac{\Delta_r}{2}\mathbf{n}_{M+1,j}, \end{aligned} \qquad j = 1, \ldots, N. \tag{9}$$

The grid points $\mathbf{m}$ are the unknowns in our optimization, and the base points $\mathbf{a}_{i,j}$ and $\mathbf{b}_{i,j}$ on the rail curves are computed from them

via

$$\begin{aligned} \mathbf{a}_{i,j} &= \mathbf{m}_{i,j} - \frac{\Delta_r}{2}\mathbf{n}_{i,j}, \\ \mathbf{b}_{i,j} &= \mathbf{m}_{i+1,j} + \frac{\Delta_r}{2}\mathbf{n}_{i+1,j}, \end{aligned} \qquad \begin{aligned} &i = 1,\dots,M, \\ &j = 1,\dots,N. \end{aligned} \tag{10}$$

To summarize, we have the whole quad structure described by the grid of points $\mathbf{m}_{i,j}$, $i = 1,\dots,M+1$, $j = 1,\dots,N$ and we can arbitrarily change the representation of the ruled surfaces $\mathbf{y}_i$ (describing the motion of the tool) from $\mathbf{a}_{i,j}$, $\mathbf{b}_{i,j}$ to $\mathbf{m}_{i,j}$ and back.

So far, we have only incorporated our objective that the tip and the bottom of the tool should move tangentially to Φ. This would cause under- and over-cuts of the internal parts of the conical tool. In order to further improve the approximation error for the whole tool, i.e., to bring the tool closer to Φ globally, we sample individual rulings and the corresponding values of the radial function, i.e.,

$$\begin{aligned} \mathbf{y}_{i,j}^k &= \mathbf{a}_{i,j}(1-s_k) + \mathbf{b}_{i,j}s_k, \\ r^k &= r_1(1-s_k) + r_2 s_k, \end{aligned} \qquad s_k = \frac{k-1}{K-1}, \quad k = 1,\dots,K, \tag{11}$$

which introduces additional K constraints for every position of the axis.

In summary, we have constructed an initial $\mathbf{m}$-grid of $N\times(M+1)$ points that we aim to optimize. On top of this, each ruling has K samples, but these are not optimization variables as they are fully determined by the position of the rulings. Consequently, we have $3N(M+1)$ variables for the optimization process. The pair of radii (r_1, r_2) and the length of the tool L are, unless stated otherwise, also taken as given (and thus fixed).

Remark 3. Recall that the strips can have varying lengths, i.e., each strip can have its own number of samples $N_1,\dots,N_M$, which is also how our algorithm is implemented. However, to simplify the arguments and notation, the formulae that follow rely on a unified N for all the strips.

We are now ready to present the optimization of the initial $\mathbf{m}$-grid. We have several objectives that need to be satisfied: the axis of the tool should be rigid, the motion of the tool should closely approximate the references geometry, the motion should be fair, and the strips should be joined with G^1 continuity. We now address these objectives, one at a time.

Rigidity of the axis. In order to guarantee that the ruling surfaces $\mathbf{y}_i$ actually describe the rigid motion of the tool's axis, the distance between $\mathbf{a}_i$ and $\mathbf{b}_i$ must be constant in time and equal to L, which we capture by

$$F_{rigid}(\mathbf{m}) = \frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\|\mathbf{a}_{i,j}-\mathbf{b}_{i,j}\|^2 - L^2\right)^2. \tag{12}$$

Approximation quality. Another requirement is the quality of the approximation, i.e., we want the corresponding set of envelope strips to be as close as possible to the reference geometry. We use the first order approximation of the surface (tangent planes at the footpoints) and require the points on the axes to move parallel to these planes at the distance given by the radius r^k, leading to

$$F_{approx}(\mathbf{m}) = \frac{1}{3NMK}\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{K}\left(\left\langle \mathbf{y}_{i,j}^k - \mathbf{y}_{i,j}^{k,\perp}, \mathbf{n}_{i,j}^k\right\rangle - r^k\right)^2, \tag{13}$$

where $\mathbf{y}_{i,j}^{k,\perp}$ is the closest point (aka footpoint) of $\mathbf{y}_{i,j}^k$ on Φ.

Fair motion. Another natural requirement is the fairness of the motion (to avoid tool chattering). To force the rail curves behave nicely, which is equivalent that the $\mathbf{m}$-grid behaves nicely in the time direction, we formulate

$$F_{fair}(\mathbf{m}) = \frac{1}{3(N-2)(M+1)}\sum_{i=1}^{M+1}\sum_{j=2}^{N-1}\|\mathbf{m}_{i,j+1} - 2\mathbf{m}_{i,j} + \mathbf{m}_{i,j-1}\|^2. \tag{14}$$

Observe that F_{fair} is applied only in the time direction. In the ruling direction, the $\mathbf{m}$-grid is subjected to F_{approx}.

G^1 continuity. To ensure the G^1 continuity between two adjacent envelope strips, we require the projections of the rulings (tool axis in time) of the neighboring ruled surfaces to the corresponding normal vectors to be equal. Denoting the two involved angles between the tool axes and the surface normal (see Fig. 14) via

$$\begin{aligned} \cos(\theta_{i,j}^1) &= \frac{\langle \mathbf{n}_{i+1,j}, \mathbf{b}_{i,j} - \mathbf{a}_{i,j}\rangle}{\|\mathbf{b}_{i,j} - \mathbf{a}_{i,j}\|}, \\ \cos(\theta_{i,j}^2) &= \frac{\langle \mathbf{n}_{i+1,j}, \mathbf{b}_{i+1,j} - \mathbf{a}_{i+1,j}\rangle}{\|\mathbf{b}_{i+1,j} - \mathbf{a}_{i+1,j}\|}, \end{aligned} \tag{15}$$

we define

$$F_{G^1}(\mathbf{m}) = \frac{1}{N(M-1)}\sum_{i=1}^{M-1}\sum_{j=1}^{N}\left(\cos(\theta_{i,j}^1) - \cos(\theta_{i,j}^2)\right)^2. \tag{16}$$

Proximity of the initial guess. As a regularizer, we include a term that prevents the optimized positions to move too far from the initial guesses, that is

$$F_{ini}(\mathbf{m}) = \frac{1}{3N(M+1)}\sum_{i=1}^{M}\sum_{j=1}^{N}\|\mathbf{m}^{i,j} - \mathbf{m}_{i,j}^{ini}\|^2. \tag{17}$$

Putting all the terms above together, the resulting objective function F to be minimized becomes

$$F(\mathbf{m}) = w_1F_{rigid} + w_2F_{approx} + w_3F_{G^1} + w_4F_{ini} + w_5F_{fair}. \tag{18}$$

Implementation details can be found in Section 7.1.

Remark 4. The highest priority is given to the rigidity constraint with weight $w_1 = 1$ and the least priority is given to the proximity term with $w_4 = 0.0001$ (unless stated otherwise). Other weights are assigned according to the specific priority, e.g., G^1 continuity versus approximation quality; see for instance the example in Fig. 16 for the particular impact of the weights.

7. Examples

We now demonstrate our algorithm on several benchmark industrial examples as well as on synthetic free-form surfaces. The first example shows one flank of a spiral bevel gear approximated by five paths of a given (fixed) conical tool, before and after optimization; see Fig. 13. Observe the improvement in terms of the approximation error in Fig. 13(b–c). The envelopes, Ω_i, $i = 1,\dots,5$, are color-coded by the signed one-sided distance error $\mathrm{dist}(\Omega,\Phi)$ towards the reference surface Φ. The error is computed in a discrete fashion, by sampling the ruled surfaces R_i, computing their footpoints on Φ, and subtracting the corresponding radius of the cone (recall Eq. (11)) as

$$\mathrm{dist}(\Omega_i,\Phi) = \max_{j,k}(\mathrm{dist}(\mathbf{y}_{i,j}^k, \mathbf{y}_{i,j}^{k,\perp}) - r^k). \tag{19}$$

Fig. 15 shows the optimization of three paths with and without the G^1-continuity term; see w_3 in Eq. (18). Again, the paths are color-coded by the distance error towards the reference surface (gear flank) Φ. Observe that when applying the G^1 constraint, the error changes smoothly along the ruling directions, while without it there are clear marks indicating only G^0 continuity. As expected, the approximation error improves when relaxing the G^1-continuity constraint.

Another qualitative comparison showing the trade-off between the G^1 continuity and the approximation quality is depicted in Fig. 16. These two objectives are represented in Eq. (18) by the corresponding terms via weights w_2, and w_3, respectively. Prioritizing the G^1 continuity (see Fig. 16, top row), one obtains

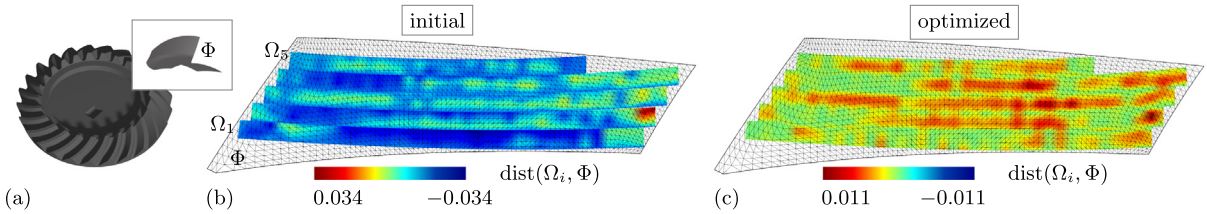

Fig. 13. An industrial benchmark. (a) A spiral bevel gear with a cavity between two teeth (top frame). (b) The right flank Φ is approximated by five initial strips of a conical tool, as described in Section 5. (c) Our optimization (see Section 6) improves the accuracy towards Φ and G^1 continuity of the neighboring strips. In this example, the optimization weights were set as $w_2 = 0.0001$ and $w_3 = 1$; see (18).

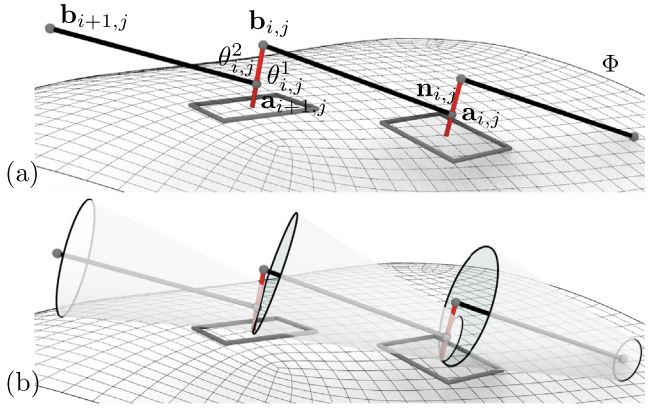

Fig. 14. G^1 continuity. (a) Three consecutive tool axis positions are shown, together with surface normals (red) and tangent planes (gray rectangles) of Φ at the two footpoints. Two neighboring axes span angles θ^1 and θ^2 with the corresponding surface normal. The constraint $\theta^1 = \theta^2$ represents the G^1 continuity of the underlying envelopes. (b) The corresponding conical tools aligned with the reference surface Φ are shown. Note that the axis endpoints are not the centers of the orbital circles but rather the centers of the inscribed spheres. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

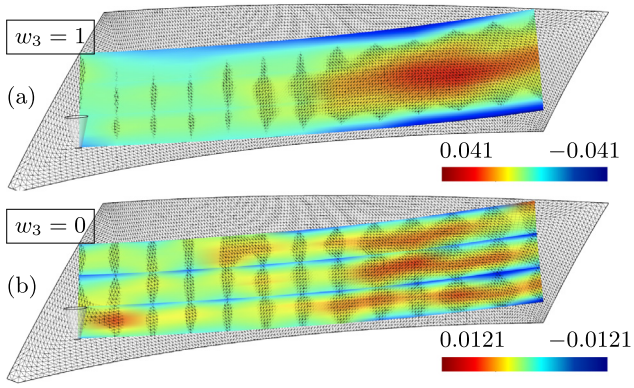

Fig. 15. G^1 continuity. Three optimized patches on the gear flank are shown. In (a), the term corresponding to G^1 continuity of the neighboring strips is applied. Ignoring it in (b), by setting $w_3 = 0$ in Eq. (18), one obtains a lot more accurate approximation. Note that the error bars are set intentionally on different scales for visualization purposes: the same error range as in (a) would result in all three strips to be almost completely green in (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

an approximation with an almost perfect solution in terms of normal continuity among the neighboring envelopes, but with larger distance error. Prioritizing the approximation quality, by setting $w_2 = 1$ and $w_3 = 0.001$ (see Fig. 16, bottom row), the distance error is improved up to the industrial thresholds only by slightly compromising the G^1 continuity among the neighboring strips.

An example of an approximation of a synthetic free-form surface is shown in Fig. 17. The network of 12 guiding curves $\mathbf{g}_i$ generates a set of $M = 11$ initial strips using a fixed conical tool in Fig. 17(b). Observe the clear gaps between the strips of the initial envelopes, that got optimized towards G^1 continuity (weights $w_2 = 0.005$, $w_3 = 1$) for a cost of only a minor decrease of the approximation quality.

Fig. 18 shows the impact of the tool engagement on the approximation quality. When the whole tool is involved (Fig. 18, top), it results in 5 paths, with the error above 5 μm. Using only the upper half of the tool, one needs 10 paths, but they offer a lot better approximation. Observe that the error drops considerably below 5 μm, which is typically the desired error threshold for gear benchmarks.

Yet another example, now on an industrial dataset of a blade of a blisk (aka blade integrated disk), is shown in Fig. 19. The reference geometry, Φ, is the concave side of one blade, and three types of initial paths were experimented with: vertical, horizontal, and diagonal. Our framework does optimize the contact path, but only to a very small extent and thus the paths follow closely the initial guiding curves. The best option are paths called 'vertical' (see Fig. 19, left), where the approximation error, after optimization, is below 25 μm and the maximum kink angle deviation is 1.14°. This fact is in accordance with the geometry of the blade as its shape is close to a parabolic cylinder, and therefore it is expected that the best machining directions would be orthogonal to the ruling directions.

An example of a face mesh from [29] is shown in Fig. 20. We set the guiding direction to be vertical, which resulted in 24 guiding curves and consequently $M = 23$ initial envelopes; see Fig. 20(b). With this complex geometry, the optimization worked only until certain extent (see Fig. 20(c)), as the lengths of the tool (of all positions) did not converge to a constant.

Another comparison with [29] is shown in Fig. 21, where a surface with a convex and a concave bumps is shown. Clearly, such a geometry is not well suited for approximation with conical tools and the approximation errors are above the fine machining tolerances for flank milling. However, our objective is to look for approximations which offer G^1 connections between the adjacent envelopes, and this has been achieved; see Fig. 21(c) with a nicely smooth silhouette of the envelope strips.

Our final example shows a comparison of a given conical versus a given cylindrical tool; see Fig. 22. For a flat geometry, e.g. the flank of the gear, such a comparison would give a marginal difference as the flat tool (conical or cylindrical) can approximate it very well. On a free-form surface, however, one may observe a

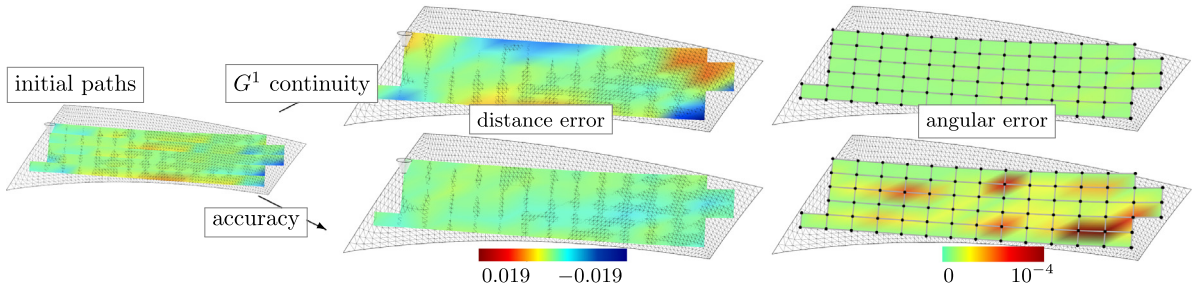

Fig. 16. Accuracy versus G^1 continuity. Left: Another initialization on the gear flank using 5 strips. The optimization towards G^1 continuity (upper row) or accuracy (lower row) is shown. Middle column: the optimized envelopes, color-coded by the approximation error. Right column: the quad structure of the **m**-grid. The quadmesh is color-coded by the angular difference $\left(\cos(\theta^1_{i,j}) - \cos(\theta^2_{i,j})\right)^2$ between the surface normal and the tool axes; recall Eq. (16). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

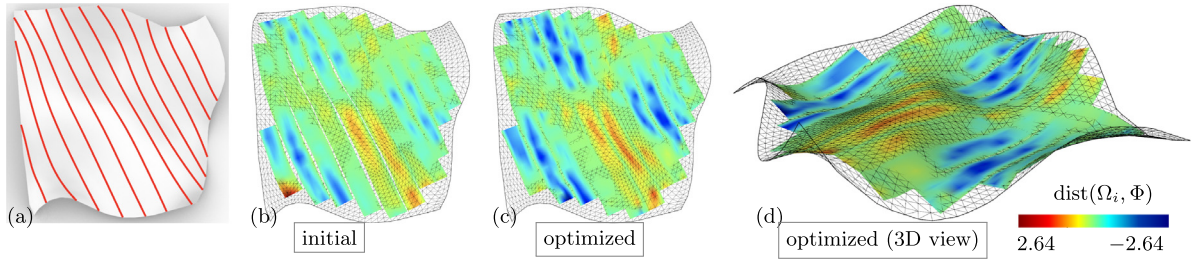

Fig. 17. A general free-form surface. (a) An initial network of the guiding **g** curves is computed as described in Section 5. The initial paths (b) and the optimized counterparts (c), when prioritizing the G^1 continuity, are shown. (d) A 3D view of the surface and the optimized paths.

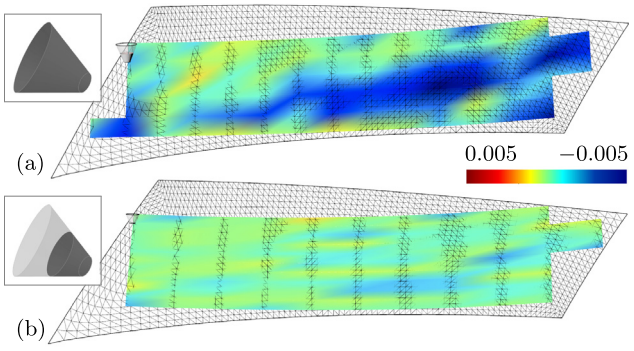

Fig. 18. Tool engagement. The same tool is used, but with different engagements: full in (a) and partial in (b). (a) 5 strips. (b) Using only the upper-half of the tool results in 10 strips, but the approximation error drops considerably below 5 μm.

slightly better performance of the conical tool. This agrees with the theory: while both cylindrical and conical tools have constant (zero) Gaussian curvature, conical tools adapt better towards the given free-form shape.

7.1. Implementation details

All the simulations have been performed on a standard laptop with 16 GB RAM and a 2.3 GHz Dual-Core Intel Core i5 processor. The algorithms corresponding to guiding curves generation have been implemented in the virtual Jupyter environment (same as in [10]), and the global tool path optimization framework has been implemented in Python 3.7.0 with the Trimesh, Tensorflow, and Scipy libraries. The average computation took around 30 minutes. The most expensive example was that of Fig. 20, which took 65 minutes. An overview of the optimization parameters of all presented examples is given in Table 1.

7.2. Discussion & limitations

Collision detection. In our implementation, we did not incorporate global collision detection into the path-planning stage. One could compute the bounds on the non-colliding position, e.g. using the approach used in [30]. However, such an approach would be an additional computational burden and more importantly would reduce the search space of all possible positions of the truncated cones. The main challenge of this work lies in connecting the adjacent envelopes in a G^1 fashion and the collision detection is left as a post-process.

Initialization. When initializing the motions of the tool, we compute the guiding curves that are as parallel as possible and their distance is desired to be $d = \sqrt{L^2 - \Delta_r^2}$. This equation assumes a tangential contact at both ends of the tool and a planar surface. Therefore, such an initialization is sufficient only for relatively flat surfaces and/or scenarios when the size of the tool is relatively small compared to the (principal) curvature radii of the surface. For highly curved regions, as shown e.g. in Fig. 17, such an initialization is not sufficient as the above constraint on d is clearly violated. Recall also Fig. 20(d), where the initial guess is poor and the tool axis length is not optimized to a constant, which means that the physical realization would not even be possible (as the tool has to be rigid).

Complex geometries. Our wish to approximate a free-form surface using G^1 continuous envelopes of conical tools is possible only for relatively flat geometries. A clear limitation is showed in Figs. 20 and 21, where the optimization does not manage to approximate the input surface within sufficient accuracy. However, this is not a limitation of the proposed methodology, rather a limitation related to the fact the e.g. convex surfaces cannot be closely approximated by a straight tool's envelopes and over- and/or under-cutting is inevitable.

Error measurement. In the geometric processing literature, there are various ways how to measure the error between two geometric entities, such as meshes. While the error measurement in the terms of the classical Euclidean metric is the most common

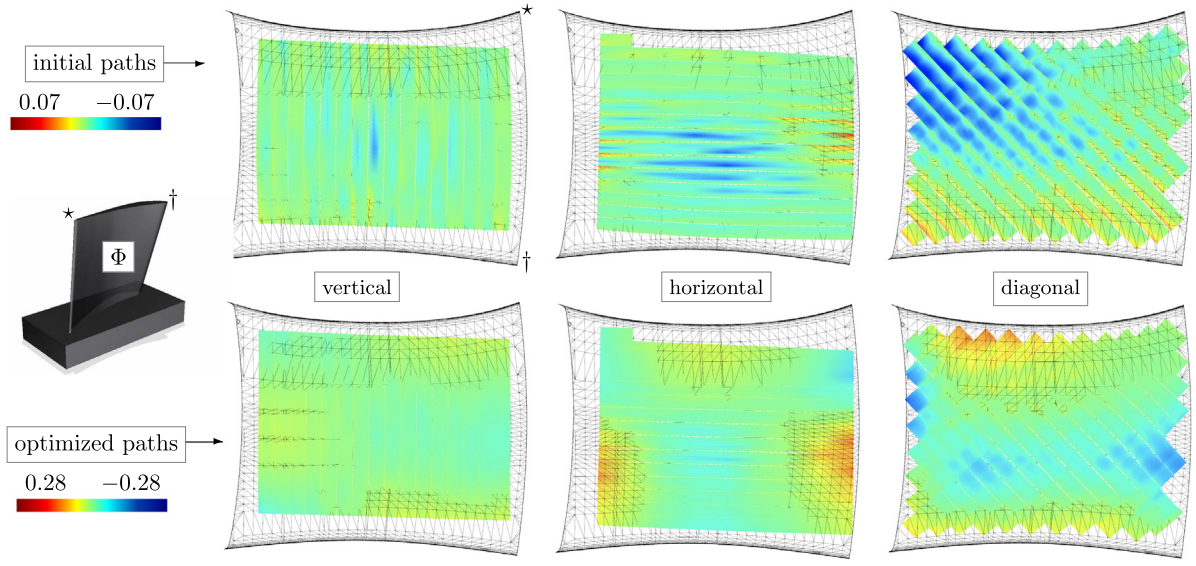

Fig. 19. Path dependence. A blade of a blisk (left) is approximated by three sets of milling paths according to the desired pivotal direction: vertical, horizontal, and diagonal (from left to right). The top row shows the initial paths, and the bottom row shows the optimized paths using $w_2 = 0.001$ and $w_3 = 1$. The envelopes are again color-coded with the distance error to Φ. The orientation of the Φ patch is signaled by the correspondence of its corners using the $\star$ and † symbols. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

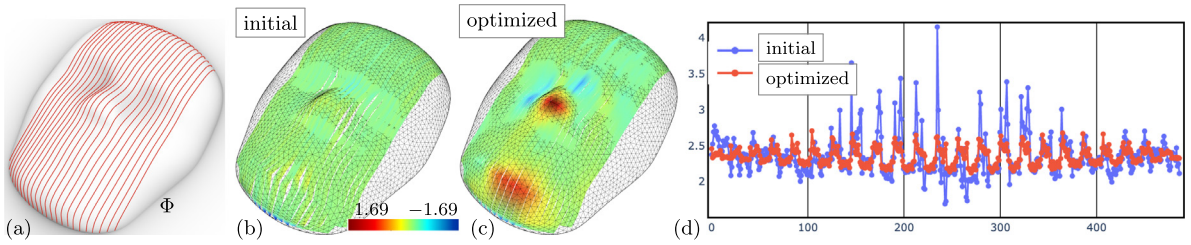

Fig. 20. A complex free-form surface. (a) An example of a face mesh Φ from [29] with the initial guiding curves. (b) The initial set of $M = 23$ envelopes and (c) the optimized counterparts, both color-coded by the distance to Φ. (d) The histogram of the tool's length of 506 axis positions. The optimization failed to make the axis length constant, cf. Fig. 11. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

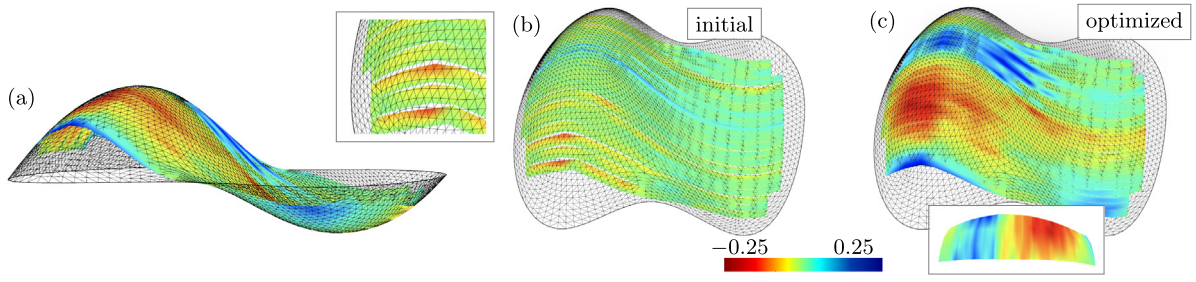

Fig. 21. A convex–concave surface. (a) Another challenging geometry taken from [29]. The initial (b) and the optimized (c) envelopes, color-coded by the distance to the input surface, are shown. While the initial envelopes have visible gaps (top frame), the optimization makes the result close to G^1, at the cost of a slightly higher approximation error. Observe the smooth silhouette (bottom frame) of the approximation of the convex part. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

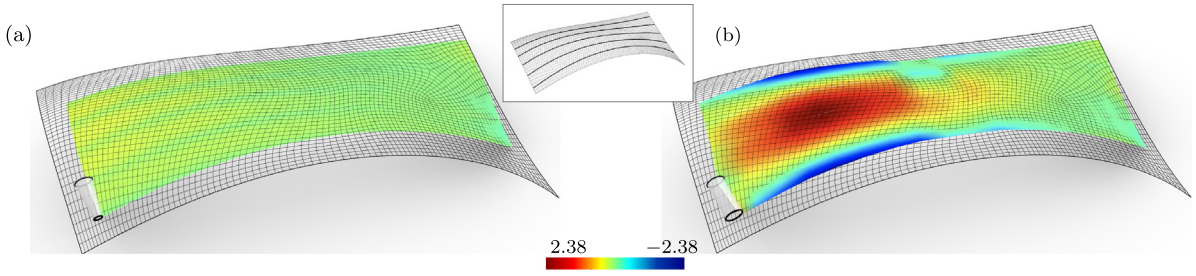

Fig. 22. A conical versus a cylindrical tool. Using the same set of guiding curves (top frame), a synthetic free-form surface is approximated by four envelopes of a conical (a) and cylindrical (b) tool.

Table 1
An overview of the simulation parameters for different examples used throughout the paper and the resulting errors. We report the length L of the axis, the two radii, the number of sampling points sampled on the quad structure, the final error after optimization, and the mesh size. All lengths are given in millimeters.

Example	L	r_1	r_2	# points $N \times M \times K$	Error	Mesh size (mm)
Gear Fig. 13(c)	1.97	0.2988	0.986	$40 \times 5 \times 40$	0.01	
Gear Fig. 15(a)	2.71	0.3	1	$14 \times 3 \times 40$	0.041	
Gear Fig. 15(b)	2.71	0.3	1	$14 \times 3 \times 40$	0.0121	
Gear Fig. 16	1.71	0.3	1	$15 \times 5 \times 40$	0.019	$41 \times 31 \times 16$
Gear Fig. 18(a)	1.717	0.3	1	$15 \times 5 \times 40$	0.0051	
Gear Fig. 18(b)	0.85	0.3	0.65	$15 \times 10 \times 40$	0.0051	
Blisk Fig. 19 (vertical)	2.46	1	0.3	$10 \times 16 \times 30$	0.28	
Blisk Fig. 19 (horizontal)	2.10	1	0.3	$10 \times 15 \times 30$	0.28	$12 \times 47 \times 37$
Blisk Fig. 19 (diagonal)	2.83	1	0.3	$10 \times 19 \times 30$	0.28	
Face Fig. 20(b)	2.13	0.3	1	$23 \times 23 \times 30$	1.69	$67 \times 90 \times 27$
Concavity Fig. 21(b)	2.64	0.3	1	$21 \times 20 \times 20$	0.25	$80 \times 80 \times 30$
Free-form Fig. 17	23.85	3	10	$20 \times 11 \times 30$	2.64	$280 \times 290 \times 100$
Free-form Fig. 22(a)	18.62	2	4	$20 \times 4 \times 40$	2.38	$108 \times 253 \times 55$
Free-form Fig. 22(b)	18.67	4	4	$20 \times 4 \times 40$	2.38	

one, one may consider the deviation of the normal fields as well [44]. Our algorithm looks for solutions that minimize both the distance between the envelopes and the reference surface, and the deviation of the normals between the neighboring strips. On a conceptual level, it is a step towards a Sobolev metric H^1 that combines the deviation of distances (points) and angles (normals), in our case blended with the corresponding weights; recall Eq. (18).

Optimization. We incorporated the hard constraint on the tool axis length as a soft constraint with a high weight. The optimization worked well this way, recall Fig. 11, except for complex geometries (Fig. 20(d)) that contradict good G^1 approximation. We used gradient-based methods to optimize the initial positions. While these methods work very well once a good initial guess is provided, these methods typically tend to converge to local minimizers in scenarios where the initial guess is not good enough. One could experiment with stochastic and/or machine learning methods to improve the optimized solution. Such investigations fall beyond the scope of the current paper.

Fixed versus variable tool. In most of the examples, we set the radii r_1 and r_2 as fixed, and optimize the length L of the tool. We incorporated r_1 and r_2 as optimization variables in Fig. 13, but the error improvement was marginal compared to fixed radii ($r_1 = 0.3$, $r_2 = 1$). In contrast, fixed length turned out to be a more significant factor and therefore we optimized mainly for L in our examples.

Overlapping. One could allow the tool to "overlap", i.e., when constructing the envelopes to allow tangential gliding such that only a subpart of the tool would be involved/engaged. This engagement would change along strips and also dynamically in time. Such an approach might improve the approximation error. However, it would require a very different conceptual approach and would be computationally more expensive as the beginning and end of the engagement would have to be incorporated as extra variables.

8. Conclusion

We have introduced a new path-planning algorithm for 5-axis flank CNC machining of free-form surfaces using conical and/or cylindrical tools. Given an input free-form surface and a pivotal machining direction given by a set of guiding curves or just a single tangent vector, our framework computes the milling paths of a conical tool, fixed or variable, such that the tool moves close to the given guiding curves, and, among other standard objectives such as the motion smoothness or approximation quality, optimizes the motions such that the neighboring paths are G^1 continuous. As this constraint is too restrictive for complex geometries, our optimization-based framework offers a trade-off between the approximation quality (machining accuracy) and G^1 continuity of the neighboring paths, controlled by the corresponding weights in the our optimization.

Our algorithm has been validated on several industrial benchmark datasets as well as on synthetic free-form geometries. We have shown that our results meet the fine industrial tolerances of a few micrometers. In regions where G^1 continuity and high accuracy cannot be met simultaneously, e.g. on highly convex patches, one may consider curved (aka custom-shaped) tools, which remains an interesting direction for future research, as well as physical validations of this methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the Spanish Ministry of Science, Spain, Innovation and Universities, Spain, grant No PID2019-104488RB-I00, BCAM "Severo Ocho" accreditation (SEV-2017-0718), and by the European Union's Horizon 2020 program under grant agreement No 862025. Michael Bartoň was supported by the Ramón *y* Cajal, Spain fellowship RYC-2017-22649.

References

[1] Calleja A, Bo P, González H, Bartoň M, López de Lacalle LN. Highly accurate 5-axis flank CNC machining with conical tools. Int J Adv Manuf Technol 2018;1–11.
[2] Choi BK, Jerard RB. Sculptured surface machining: Theory and applications. Kluwer; 1998.
[3] Elber G, Fish R. 5-Axis freeform surface milling using piecewise rule surface approximation. ASME J Manuf Sci Eng 1997;119(3):383–7.
[4] Zheng G, Bi Q-Z, Zhu L-M. Smooth tool path generation for five-axis flank milling using multi-objective programming. Proc Inst Mech Eng B 2012;226(2):247–54.
[5] Wang C, Elber G. Multi-dimensional dynamic programming in ruled surface fitting. Comput Aided Des 2014;51:39–49.
[6] Wang C, Tang K. Optimal boundary triangulations of an interpolating ruled surface. J Comput Inf Sci Eng 2005;5(4):291–301.
[7] Li C, Bedi S, Mann S. Flank millable surface design with conical and barrel tools. Comput-Aided Des Appl 2008;5:461–70.

[8] Roth D, Bedi S, Ismail F, Mann S. Surface swept by a toroidal cutter during 5-axis machining. Comput Aided Des 2001;33(1):57–63.
[9] Machchhar J, Plakhotnik D, Elber G. Precise algebraic-based swept volumes for arbitrary free-form shaped tools towards multi-axis CNC machining verification. Comput Aided Des 2017;90:48–58.
[10] Bartoň M, Bizzarri M, Rist F, Sliusarenko O, Pottmann H. Geometry and tool motion planning for curvature adapted CNC machining. ACM Trans Graph 2021;40(4):1–16.
[11] Chu C, Huang W, Hsu Y. Machining accuracy improvement in five-axis flank milling of ruled surfaces. Int J Mach Tools Manuf 2008;48(7):914–21.
[12] Pottmann H, Peternell M. Envelopes-computational theory and applications. In: Proceedings of spring conference on computer graphics. 2000, p. 3–23.
[13] Flöry S, Pottmann H. Ruled surfaces for rationalization and design in architecture. In: Proceedings of the 30th annual conference of the association for computer aided design in architecture. 2010, p. 103–9.
[14] Li C, Bedi S, Mann S. Flank milling of a ruled surface with conical tools – an optimization approach. Int J Adv Manuf Technol 2006;29:1115–24.
[15] Sprott K, Ravani B. Cylindrical milling of ruled surfaces. Int J Adv Manuf Technol 2008;38:649–56.
[16] Chen HY, Pottmann H. Approximation by ruled surfaces. J Comput Appl Math 1999;102:143–56.
[17] Gong H, Li-Xin C, Jian L. Improved positioning of cylindrical cutter for flank milling ruled surfaces. Comput Aided Des 2005;37:1205–13.
[18] Senatore J, Landon Y, Rubio W. Analytical estimation of error in flank milling of ruled surfaces. Comput Aided Des 2008;40:595–603.
[19] Zhu L, Zheng G, Ding H, Xiong Y. Global optimization of tool path for five-axis flank milling with a conical cutter. Comput Aided Des 2010;42(10):903–10.
[20] Lu Y, Bi Q, Zhu L. Five-axis flank milling of impellers: Optimal geometry of a conical tool considering stiffness and geometric constraints. Proc Inst Mech Eng B 2014;228:1226–36.
[21] Bo P, Bartoň M, Pottmann H. Automatic fitting of conical envelopes to free-form surfaces for flank CNC machining. Comput Aided Des 2017;91:84–94.
[22] Skopenkov M, Bo P, Bartoň M, Pottmann H. Characterizing envelopes of moving rotational cones and applications in CNC machining. Comput Aided Geom Design 2020;83:101944.
[23] Bo P, Bartoň M. On initialization of milling paths for 5-axis flank CNC machining of free-form surfaces with general milling tools. Comput Aided Geom Design 2019;71:30–42.
[24] Zhu L, Lu Y. Geometric conditions for tangent continuity of swept tool envelopes with application to multi-pass flank milling. Comput Aided Des 2015;59:43–9.
[25] Bo P, Bartoň M, Plakhotnik D, Pottmann H. Towards efficient 5-axis flank CNC machining of free-form surfaces via fitting envelopes of surfaces of revolution. Comput Aided Des 2016;79:1–11.
[26] Bo P, González H, Calleja A, de Lacalle LNL, Bartoň M. 5-axis double-flank CNC machining of spiral bevel gears via custom-shaped milling tools—Part I: Modeling and simulation. Precis Eng 2020;62:204–12.
[27] Escudero GG, Bo P, González-Barrio H, Calleja-Ochoa A, Bartoň M, de Lacalle LNL. 5-axis double-flank CNC machining of spiral bevel gears via custom-shaped tools—Part II: physical validations and experiments. Int J Adv Manuf Technol 2021;1–12.
[28] Bizzarri M, Bartoň M. Manufacturing of screw rotors via 5-axis double-flank CNC machining. Comput Aided Des 2021;132:102960.
[29] He D, Li Z, Li Y, Tang K. Quasi-developable and signed multi-strip approximation of a freeform surface mesh for efficient flank milling. Comput Aided Des 2021;140:103083.
[30] Ezair B, Elber G. Using curvature bounds towards collision free 5-axis tool-paths. Graph Models 2019;103:101022.
[31] Sun Y, Sun J, Li J, Li W, Feng B. Modeling of cutting force under the tool flank wear effect in end milling Ti6Al4V with solid carbide tool. Int J Adv Manuf Technol 2013;69(9–12):2545–53.
[32] Larue A, Altintas Y. Simulation of flank milling processes. Int J Mach Tools Manuf 2005;45(4):549–59.
[33] Sonthipermpoon K, Bohez E, Hasemann H, Rautenberg M. The vibration behavior of impeller blades in the five-axis CNC flank milling process. Int J Adv Manuf Technol 2010;46(9–12):1171–7.
[34] Zhang X, Zhang J, Pang B, Zhao W. An accurate prediction method of cutting forces in 5-axis flank milling of sculptured surface. Int J Mach Tools Manuf 2016;104:26–36.
[35] Xu K, Wang J, Chu C-H, Tang K. Cutting force and machine kinematics constrained cutter location planning for five-axis flank milling of ruled surfaces. J Comput Des Eng 2017 4(3):203–17.
[36] De Lacalle LL, Lamikiz A, Sanchez J, Salgado M. Toolpath selection based on the minimum deflection cutting forces in the programming of complex surfaces milling. Int J Mach Tools Manuf 2007;47(2):388–400.
[37] Beudaert X, Lavernhe S, Tournier C. Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G^1 tool path. Int J Mach Tools Manuf 2012;57:73–82.
[38] De Lacalle LL, Angulo C, Lamikiz A, Sanchez J. Experimental and numerical investigation of the effect of spray cutting fluids in high speed milling. J Mater Process Technol 2006;172(1):11–5.
[39] Sorby K, Tonnessen K, Torjusen JE, Rasch FO. Improving high speed flank milling operations in multi-axis machines. CIRP Ann-Manuf Technol 2000;49(1):371–4.
[40] Pechard PY, Tournier C, Lartigue C, Lugarini JP. Geometrical deviations versus smoothness in 5-axis high-speed flank milling. Int J Mach Tools Manuf 2009;49:453–61.
[41] Harik RF, Gong H, Bernard A. 5-axis flank milling: A state-of-the-art review. Comput Aided Des 2013;45(3):796–808.
[42] Rajain K, Sliusarenko O, Bizzarri M, Bartoň M. Curve-guided 5-axis CNC flank milling of free-form surfaces using custom-shaped tools. Comput Aided Geom Design 2022;102082.
[43] Bizzarri M, Lávička M, Kosinka J. Skinning and blending with rational envelope surfaces. Comput Aided Des 2017;87:41–51.
[44] Cohen-Steiner D, Alliez P, Desbrun M. Variational shape approximation. In: ACM SIGGRAPH 2004 papers. 2004, p. 905–14.