

Efficient 5-axis CNC trochoidal flank milling of 3D cavities using custom-shaped cutting tools

Pengbo Bo^{a,*}, Hongyu Fan^a, Michael Barton^{b,c}

^aSchool of Computer Science and Technology, Harbin Institute of Technology, West Wenhua Str. 2, 264209 Weihai, China

^bBCAM – Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Basque Country, Spain

^cIkerbasque – Basque Foundation for Sciences, Maria Diaz de Haro 3, 48013 Bilbao, Basque Country, Spain

Abstract

A novel method for trochoidal flank milling of 3D cavities bounded by free-form surfaces is proposed. Existing 3D trochoidal milling methods use on-market milling tools whose shape is typically cylindrical or conical, and is therefore not well-suited for meeting fine milling tolerances required for finishing of benchmark free-form surfaces like blades or blisks. In contrast, our variational framework incorporates the shape of the tool into the optimization cycle and looks not only for the trochoidal milling paths, but also for the shape of the tool itself. High precision quality is ensured by firstly designing flank milling paths for the side surfaces that delimit the motion space, in which the trochoidal milling paths are further computed. Additionally, the material removal rate is maximized with the cutter-workpiece engagement being constrained under a given tolerance. Our framework also supports multi-layer approach that is necessary to handle deep cavities. The ability and efficacy of the proposed method are demonstrated by several industrial benchmarks, showing that our approach meets fine machining tolerances using only a few trochoidal paths.

Keywords: 5-axis CNC machining, trochoidal milling, custom-shaped tools, roughing operations, tangential movability, free-form shape manufacturing

1. Introduction

Efficient manufacturing of curved objects is an essential step for many industrial sectors, automotive or aeronautical to name a few. Even though additive technologies like 3D printing are becoming more and more popular [1], there are objects that need to be, e.g. for stiffness reasons, manufactured from a single material block using traditional subtractive approaches and Computer Numerically Controlled (CNC) machining is the leading subtractive technology [2–8].

Typically, a cutting tool is given and one looks for its milling paths such that the motion of the tool approximates the given reference geometry with high precision. Industrial benchmarks like turbine blades, rotors, or blisks, represented by free-form (NURBS) surfaces, however, introduce a challenge for path-planning algorithms due to its complex, doubly-curved, geometries. In particular, an important task is manufacturing of curved 3D cavities, where the tool positioning is limited due to global collision constraints. One needs to navigate the tool to avoid collisions, yet to meet fine manufacturing tolerances. Recently custom-shaped tools have been shown to offer higher approximation quality than classical (conical or cylindrical) tools [9–11]. Our research belongs to this category and we look not only for the milling paths, but also for the optimal shape of the tool.

In conventional flank milling, the cutter is desired to be in tangential contact with the material block. This fact induces excessive cutting force but also accumulates heat and negatively affects the tool wear. This problem can be effectively avoided by trochoidal milling. A trochoidal milling path consists of two parts, the *front-half* path and the *back-half* path. Material is cut in the front-half path and the cutting heat is dissipated as the tool moves in the back-half path [12]. Therefore, in high-speed machining, trochoidal milling is increasingly used in slot milling and is also widely used in cutting hard materials, such as NiTi-based super alloy [13–15].

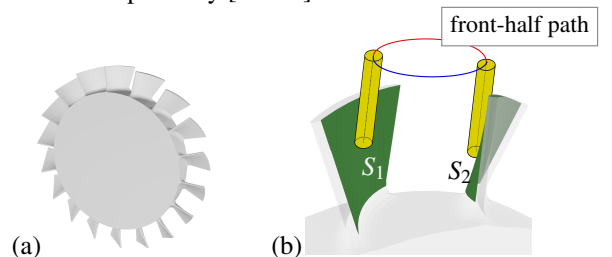


Figure 1: (a) The geometry of a blisk disc. (b) The cavity between two blades of the blisk, delimited by two free-form surfaces. The side surface S_1 (S_2) corresponds to the left (right) side of the cavity (rendered in green). Two positions of the cutting tool during one trochoidal cycle are shown. The path of the tool axis consists of two parts: the front-half path (red) and the back-half path (blue).

Suppose a cavity is delimited by two surfaces S_1 and S_2 , see Fig. 1(b). Our purpose is to determine trochoidal milling paths as well as the shape of a milling tool such that high precision milling is obtained. The cutting tools we consider are not re-

*Corresponding author

Email addresses: pbbo@hit.edu.cn (Pengbo Bo), mbarton@bcamath.org (Michael Barton)

stricted to cylindrical and/or conical, but are general surfaces of revolution. Important machining factors such as Material Removal Rate (MRR) and Cutter-Workpiece Engagement (CWE) are both incorporated in a unified variational framework. While high MRR is correlated with efficient milling, it is in contradiction with CWE, whose high values are closely related to the tool wear, and therefore required to be small. Our optimization framework looks for high MRR while keeping CWE under a given threshold. We refer the reader to [16] for more details on MRR and CWE.

A complete tool path of trochoidal milling is defined by the motion of tool axis which is a ruled surface. This ruled surface is composed of a sequence of cyclic path segments (TR cycles), and each TR cycle contains a *front-half* and a *back-half* path, see Fig. 1(b). The front-half path is also called an active part, along which the cutting tool keeps contacting with the raw material throughout the entire milling, while the back-half path is a transition path from current TR cycle to the next TR cycle. Typically, the cutting tool along a back-half path does not interact with the raw material at most of its positions.

For certain geometries, it is possible to mill the side surface with one single sweep using a large tool. However, for deep cavities using a single sweep, even of a custom-shaped tool, may not be sufficient to meet the fine machining tolerances. Therefore, one typically has to consider multi-layer milling approaches that allow the tool to move in several consecutive layers. In this work, we consider both, a single-layer milling for path generation of a cavity bounded by two surfaces in Sec. 3, and generalize it to multi-layer milling in Sec. 4.

In a summary, the problem to address in the TR path generation is to decide the front-half of every TR cycle such as key constraints are satisfied: 1) the motion of the tool along the path should be tangent to both sides S_1 , S_2 and overcut should be avoided; 2) the MRR should be maximized; 3) the CWE is under a given tolerance. The last requirement comes from the observation that CWE is closely related to the cutting force during manufacturing, therefore, in order to reduce the tool wear, the cutter-workpiece engagement should be constrained to be under a given tolerance. The rest of the paper is organized as follows. Section 2 surveys the related research, Section 3 introduces the algorithm to compute single layer TR paths and Section 4 generalizes it to the multi-layer setup. The numerical examples are shown in Section 5 and the conclusions are drawn in Section 6.

2. Related Work

2.1. The cutting force and stability of trochoidal milling

The tool wear in trochoidal milling is strongly correlated to the cutting force, and there have been a lot of studies on cutting force analysis. Otkur et al. [17] propose a comprehensive analytical model to analyze the tool-workpiece engagement and predict the cutting force. Pleta et al. [13] find that the engagement angle has the highest correspondence with the component perpendicular to the feeding direction of the cutting force. The relationship between the cutting force and the cutting depth in

trochoidal milling is further analyzed in [18]. Wu et al. [19] propose an improved model for analyzing the change of cutting force based on the typical linear milling force model. Niaki et al. [20] analyze the geometry of the in-process workpiece in TR milling in details and give a more accurate cutting force prediction model.

In addition to cutting force, stability is also particularly important in high-speed machining. Kardes et al. [21] analyze the property with the varying cutter immersion condition for suppressing the chatter during TR milling. Yan et al. [22] build a TR milling process stability prediction model. They consider the trochoidal step distance and the spindle speed for analyzing the stability of TR milling. Wang et al. [23] present an adaptive TR tool path generation method, in which machining stability is improved by maintaining the steady radial cutting depth.

2.2. Trochoidal milling of free-form surface

Traditional trochoidal milling methods mainly focus on the cutting force and milling stability of a cutting tool, and are usually used in simple slot milling. In recent years, some works considered the use trochoidal milling for curved slots or even 3D cavities. Xu et al. [24] propose a method based on polynomial curves, which realize the trochoidal machining of arbitrarily curved slot with constant width. They replace the traditional circular paths with polynomial curves as the base TR cycle to fit into the complex curved slots. Li et al. [25] propose an extension of [24], which supports curved slots with varying width by adjusting the polynomial curves. Even though these works can support more complex curved slots, they do not support general free-form cavities. Cavities bounded by free-form surfaces (such as a blisk groove in aerospace industry) are usually manufactured with traditional methods [26] which do not have the benefit of trochoidal milling.

To the best of our knowledge, the closest research to our work is on trochoidal milling of general free-form cavities by Li et al. [16]. They use a tangent sphere to generate the middle surface to design a guiding curve [27] of a TR path. In the latest work, they [28] propose an algorithm of variable-depth multi-layer 5-axis trochoidal milling. By introducing the concept of layered material removal rate (LMRR), the TR milling depth of each layer is optimized. These works apply trochoidal milling to free-form surfaces, but they target rough or semi-finishing operations and the shape of the cutting tool is a fixed input. In contrast, our approach looks also for the optimal shape of the tool.

2.3. High precision 5-axis flank milling

There are a lot of works devoted to high-precision 5-axis flank milling [9, 29–35]. The shape and size of the milling tool are mostly fixed (usually cylindrical or conical [36]). Wang et al. propose a method to compute a composition of discrete ruled surfaces fitting to a given shape using the dynamic programming [37]. Elber et al. [38] approximate general free-form surface with segmented ruled surface by using a truncated conical milling tool. However, Elber et al.'s method requires a lot of subdivisions to well approximate a general free-form shape.

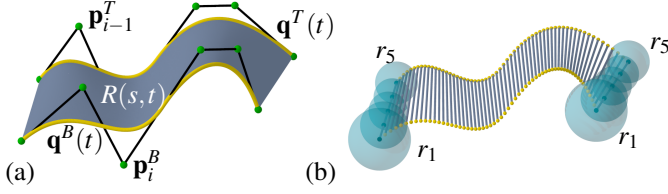


Figure 2: Ruled surface. (a) A ruled surface $R(s, t)$ (blue) is controlled by a pair of rail curves (yellow), represented as B-spline curves. (b) The path of the tool is discretized by $N = 60$ rulings, and the tool is discretized by $M = 5$ spheres (transparent), uniformly distributed along the rulings.

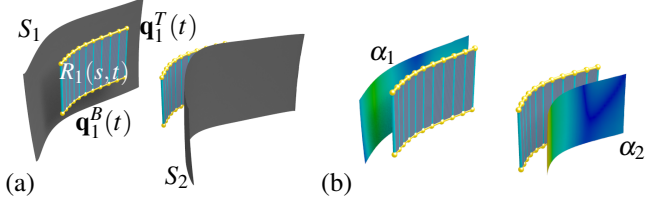


Figure 3: Initial side paths. (a) Initial ruled surfaces R_1 and R_2 that correspond to two motions of a single custom-shaped tool that approximates S_1 and S_2 are shown. (b) Two scalar functions α_1 and α_2 , defined on S_1 and S_2 , and visualized as color-maps, indicate the approximation error to S_1 and S_2 . These functions are used in Eq. (6) to further reduce the approximation error.

Redonnet et al. [39] use a cylindrical cutter for machining of ruled surfaces. They propose a three-tangential arrangements method to optimize the cutter position, which gets high precision compared with standard two-tangential arrangements.

In addition, from the general cutting tool, various methods have been proposed. Senatore et al. [40] analyze the size of a cylindrical cutter, which maximizes the radius while keeping the predefined geometric error. Zhu et al. [41] propose a method based on simultaneous optimization of the tool's motion and shape. Based on this work, Lu et al. [42] consider additional constraints such as the stiffness of a cutter. These two methods focus on improving the stiffness to reduce the deflection and vibration of the tool. Bo et al. [43] propose an alternative formula of the optimization where the shape and motion of tool are both the unknowns and are simultaneously optimized to minimize the approximation error.

3. TR paths for cavities bounded by free-form surfaces

Our research focuses on 3D cavities like the one shown in Fig. 1(b), that typically consist of a bottom surface and two side surfaces. The bottom surface of the cavity is removed from our considerations as one cannot access it by flank-milling due to global collision anyway. We focus on the *side surfaces* which are to-be-milled by one (or several) sweeps of a single custom-shaped tool. In trochoidal milling, the cutting tool touches the side surfaces only at some discrete positions, called contact lines. The whole trochoidal path is divided by the contact lines into a sequence of *TR cycles*. Each TR cycle has a front-half part and a back-half part separated by the contact lines. The contact lines act as extreme positions of the motion space of the cutting tool and the positions of contact lines de facto govern the milling precision of the side surfaces.

It is therefore essential to find accurate positions of contact lines, with higher priority than other (intermediate) positions. Moreover, the motion directions of the cutting tool at the contact lines are also directly related to the finishing quality. However, it is non-trivial to define some contact lines in advance without considering the whole TR paths for milling 3D cavity. To achieve higher milling precision, we consider a variable (custom-shaped) tool which is represented as a one-parameter family of spheres centered on the axis, and whose radii are optimized in our framework. The position of the contact lines, as well as the TR paths inside cavities, are determined using a variational algorithm.

Firstly, the shape of the tool and its paths in the neighborhood of the two reference surfaces are computed. The paths are two ruled surfaces, traversed by the axis of the cutting tool, see Fig. 3(a). These two surfaces delimit the space that is further trochoidal-milled (Sec. 3.2). Secondly, we compute the front-half of the TR cycles one by one (Sec. 3.3) such that path planning will move the tool towards the two limiting surfaces with G^1 continuity, maximizing the MRR while satisfying the CWE constraints. Finally, we generate the back-half paths as a transition between the front-half paths to generate the entire TR cycles (Sec. 3.4).

3.1. Envelope surface fitting

Our objective is to compute an optimal shape of a cutting tool T and its motion paths, i.e., the side paths for flank milling the side surfaces S_1 and S_2 , recall Fig. 1. This goal is realized by an envelope surface fitting method which proceeds as follows. The motion of the axis of the cutting tool is a ruled surface R which is defined as

$$R(s, t) = \mathbf{q}^T(t) \cdot (1 - s) + \mathbf{q}^B(t) \cdot s, \quad s, t \in [0, 1], \quad (1)$$

where $\mathbf{q}^T(t)$ is the top boundary curve and $\mathbf{q}^B(t)$ is the bottom boundary curve, which are both represented by B-spline curves in our work, see Fig. 2. In the following, we review the optimization method for computing R .

It was shown that one can approximate a single free-form surface by a motion of a custom-shaped tool [43]. In our current setup, the situation is a more complicated as we have two reference surfaces, not just one. The cutting tool, a surface of revolution, is conceptualized as a one-parameter family of spheres centered along the tool axis. The behavior of the spheres is described by a radial function $r(s)$, s being the arc length parametrization of the axis. The radial function can be thought of a smooth function, that describes the shape of the tool, but in our discrete optimization-based setup is represented by a discrete set of radii, see Fig. 2(b).

The motion path is obtained by minimizing the distance between the envelope surface of the tool along its paths and the target surfaces S . Due to the nonlinear distance function in surface approximation, an iterative procedure is employed and in each iteration a quadratic approximant function is minimized. We have

$$F_{dist}^1(\mathbf{q}^T, \mathbf{q}^B, \mathbf{r}) = \frac{1}{MN} \sum_k \sum_l^N \|R(s_k, t_l) - \mathbf{f}_{k,l} - r_k \cdot \mathbf{n}_{k,l}\|^2, \quad (2)$$

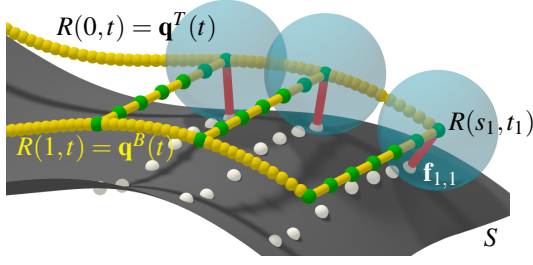


Figure 4: Optimization settings. The initial ruled surface $R(s, t)$, see Eq. (1), is uniformly discretized along the rulings (s -direction) with $M = 7$ samples (green dots). Their distances to S are computed using their footpoints on S (white dots) and are optimized via Eq. (2). The point-footpoint distance must remain constant in the t -direction (red lines); r_1 is visualized using the transparent spheres, but is also a variable in our framework.

where $\mathbf{f}_{k,l}$ is the footpoint of $R(s_k, t_l)$ on the side surface S as shown in Fig. 4, $\mathbf{n}_{k,l}$ is the unit surface normal of S at $\mathbf{f}_{k,l}$ pointing to the machining side of S , and M and N are the number of samples in s and in t directions, respectively. The optimization variables are the control points of the boundary curves \mathbf{q}^T and \mathbf{q}^B and a vector of radii $\mathbf{r} = (r_1, \dots, r_M)$. The stability of the cutting tool during its motion is closely related to the acceleration of the tool's motion, which can be expressed by the fairness of the motion. To control the fairness, we use a standard fairness term defined by the integral of norm of the second derivatives of the boundary curves, enriched by a fairness on the direction of the ruling (bottom term in (3)). Note that two fair rail curves do not imply a fair motion as the line between them may parse in a non-fair fashion. Therefore, we write

$$F_{fair}^1(\mathbf{q}^T, \mathbf{q}^B) = \int \|(\mathbf{q}^T)''(t)\|^2 dt + \int \|(\mathbf{q}^B)''(t)\|^2 dt + \int \|(\mathbf{q}^T)''(t) - (\mathbf{q}^B)''(t)\|^2 dt. \quad (3)$$

In addition, a rigid motion of the cutting tool is required which is guaranteed by a constraint on constant length of the tool axis, denoted by L . This constraint has to be satisfied in every time instant, which in our implementation is controlled at N discrete positions, i.e., we write

$$F_{rigid}^1 = \sum_l^N (\|\mathbf{q}^T(t_l) - \mathbf{q}^B(t_l)\|^2 - L^2)^2. \quad (4)$$

The envelope fitting algorithm minimizes the following function in each iteration

$$F_{prox}(\mathbf{q}^T, \mathbf{q}^B, \mathbf{r}) = F_{dist}^1 + \lambda_1 F_{fair}^1 + \lambda_2 F_{rigid}^1 \rightarrow \min \quad (5)$$

where the optimization unknowns are the control points of the boundary curves of the ruled surfaces R and a vector of radii \mathbf{r} .

3.2. Side path computation for trochoidal milling with controlled overcut

Our distance objective term F_{dist}^1 as defined in (2) minimizes distances to S in the least square sense, which naturally results in errors with both positive and negative signs, and therefore

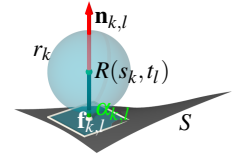
overcutting. To penalize overcutting, we introduce a virtual surface of the side surface as a target, defined by

$$\tilde{S}(s, t) = S(s, t) + \alpha(s, t) \cdot \mathbf{n}(s, t) \quad (6)$$

where \mathbf{n} indicates the surface normal pointing to the inside of cavity, and $\alpha(s, t)$ is a bivariate function that reflects the approximation error from the first optimization cycle achieved by Eq. (2). If $\alpha(s, t) \equiv \text{const.}$, \tilde{S} would be an offset surface of S . However, as $\alpha(s, t)$ varies, see Fig. 3(b), the overcut compensation has to reflect this variance. In our discrete setup, the distance objective function is a sum of distance constraints over the set of samples of $R(s, t)$, and becomes

$$F_{dist}^2(\mathbf{q}^T, \mathbf{q}^B, \mathbf{r}) = \frac{1}{MN} \sum_k^M \sum_l^N \|R(s_k, t_l) - (\mathbf{f}_{k,l} + r_k \cdot \mathbf{n}_{k,l} + \alpha_{k,l} \cdot \mathbf{n}_{k,l})\|^2, \quad (7)$$

where $\alpha_{k,l}$ are the fitting errors at the sample point $R(s_k, t_l)$ from the first optimization cycle. Observe that the optimization in Eq. (2) returns spheres (transparent) with radii r_k that, in the least square sense, approximate best the distances $\|R(s_k, t_l) - \mathbf{f}_{k,l}\|$. This results in defect (signed) distances $\alpha_{k,l}$ which corresponds to overcut or undercut, and is corrected by incorporating $\alpha_{k,l}$ in (7). Note that $\alpha_{k,l}$ changes throughout optimization iterations, and it is therefore updated dynamically in each iteration in our framework.



It is also desirable that the same tool is used for both side surfaces. Therefore, when the shape of the cutting tool is considered as a variable, both tool paths R_1 , and R_2 are optimized simultaneously with a single cutting tool. That is, the sphere radii r_k are the same for both R_1 , and R_2 , recall Fig. 3(a). The final objective distance function is

$$F_{dist}(\mathbf{q}_n^*, \mathbf{r}) = \frac{1}{MN} \sum_{n=1,2}^M \sum_k^M \sum_l^N \|R_n(s_k, t_l) - (\mathbf{f}_{k,l,n} + r_k \cdot \mathbf{n}_{k,l,n} + \alpha_{k,l,n} \cdot \mathbf{n}_{k,l,n})\|^2, \quad (8)$$

where $\star = T, B$, and the subscript $n = 1, 2$ indicates the left or the right side surface of the cavity. Similarly, energy terms of motion fairness and rigidity are defined respectively by

$$F_{fair}(\mathbf{q}_j^*) = \sum_{j=1,2} \int \|(\mathbf{q}_j^T)''(t)\|^2 dt + \int \|(\mathbf{q}_j^B)''(t)\|^2 dt + \int \|(\mathbf{q}_j^T)''(t) - (\mathbf{q}_j^B)''(t)\|^2 dt \quad (9)$$

and

$$F_{rigid}(\mathbf{q}_j^*) = \sum_{j=1,2} \sum_l^N (\|\mathbf{q}_j^T(t_l) - \mathbf{q}_j^B(t_l)\|^2 - L^2)^2, \quad (10)$$

where again $\star = T, B$, and $j = 1, 2$.

In summary, the computation of the tool shape as well as the tool paths close to S_1 and S_2 is done by iteratively minimizing the objective function

$$\min_{\mathcal{P}, \mathcal{R}} F_{dist} + \lambda_1 F_{fair} + \lambda_2 F_{rigid}, \quad (11)$$

where \mathcal{P} is a set of control points of the two ruled surfaces R_1 and R_2 , \mathcal{R} is the set of sphere radii uniformly distributed along the the tool axis, and λ_1 and λ_2 are scalar weights set

empirically. Unless stated differently in a concrete example, we set $\lambda_1 = 1e^{-6}$ and $\lambda_2 = 1$ in our implementation. We solve the objective function in Eq.(11) with the Gauss-Newton method for the control points of R_1 , R_2 , and the radii values r_k . Note that the footpoints $\mathbf{f}_{k,l}$ and associated surface normals $\mathbf{n}_{k,l}$ on the target side surfaces are updated in each iteration. The solution of Eq.(11) results in two side paths R_1 and R_2 milling the side surfaces S_1 and S_2 , respectively, and an optimal shape of the cutting tool represented by the vector of radii \mathbf{r} . In the following discussions, the tool shape (vector \mathbf{r}) is fixed.

Remark 1. When there is no risk of confusion, from this point on, we omit the variables of the objective functions and list them as sets in the subscript of the minimization symbol as in Eq. (11).

3.3. Front-half paths computation

At this point, our algorithm computed two side paths R_1 and R_2 and a shape of the tool represented by the vector of radii \mathbf{r} . Now we are going to construct trochoidal paths that join R_1 with R_2 , and start with the front-half paths.

Similarly to the side paths, we use B-spline surfaces for their representation. A front-half of a TR cycle, denoted by C_i , is defined by

$$C_i(s, t) = \mathbf{c}_i^T(t) \cdot (1 - s) + \mathbf{c}_i^B(t) \cdot s, \quad s, t \in [0, 1], \quad (12)$$

with the boundary curves of C_i defined as

$$\mathbf{c}_i^*(t) = \sum_{k=0}^m \mathbf{a}_k^* B_{k,d}(t), \quad * \in \{T, B\}, \quad (13)$$

where \mathbf{a}_k^* are the control points and $B_{k,d}(t)$ the B-spline basis functions of degree d . See Fig. 5 for two guiding paths (R_1, R_2) and two front-half paths. Since we need to interpolate boundary rulings, we use clamped uniform knot vectors for the B-spline surfaces. If not said differently, we use $m + 1$ control points, $m = 5$ in our implementation.

The computation of the front-half paths should meet two objectives: milling precision and milling efficiency. For milling precision, the cutting tool should touch the side surfaces with high degree of precision and the overcut to the side surface should meet fine manufacturing tolerances (which are tens of micrometers for objects with tens of centimeters large.) For milling efficiency, the MRR in each TR cycle should be maximized, while keeping the CWE under a pre-defined threshold. We emphasize that the CWE is closely related to physical entities such as cutting force, work load, or tool wear. We do not optimize directly these entities, as this goes beyond the scope of this paper, but we control CWE.

Remark 2. We consider several types of milling paths in our algorithm. They are all represented by ruled surfaces (motions of the milling axis) and parametrized by two variables: s (ruling direction) and t (time aka motion direction). If there is no risk of confusion, we use the same pair of parameter symbols (s, t) for all these ruled surface, but if there are more surfaces involved, each surface has its own parameter domain and these variables are different.

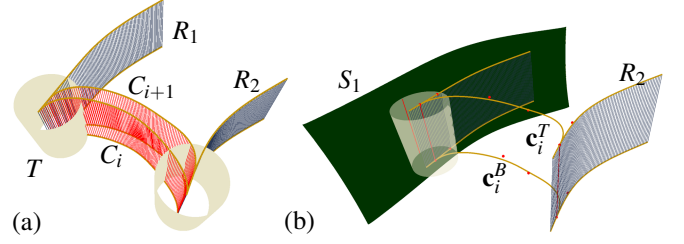


Figure 5: Front-half paths of TR cycles. (a) C_i and C_{i+1} represent the front-half paths of ruled surfaces that join the side ruled surfaces, R_1 and R_2 , in G^1 fashion. (b) This is achieved via the constraints (14) and (16), which is expressed in the terms of the control points of C_i and $R_{1,2}$.

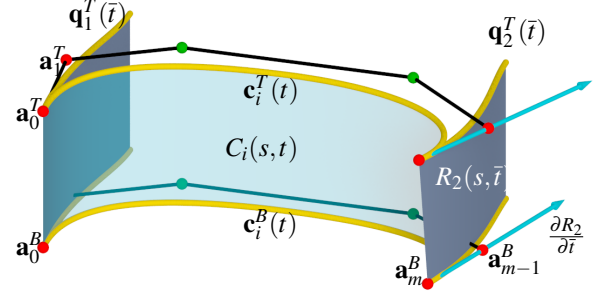


Figure 6: Free and constrained control points of TR motions. A ruled surface C_i (transparent) is determined by the two boundary curves $\mathbf{c}_i^T(t)$ and $\mathbf{c}_i^B(t)$ which satisfy the G^1 constraints with the \mathbf{q} -curves. These curves are already computed and define the side surfaces R_1 and R_2 . Therefore, some control points of C_i are locked (red), while others are being optimized (green).

3.3.1. Milling precision

The side paths R_1 , and R_2 , obtained using the method described in Sec. 3.2, provide essential guiding information for computing $C_i(s, t)$ which is required to touch the side surfaces with high degree of precision with overcut control. The rulings of the side paths define the position and orientation of the cutting tool moving along the side paths. Therefore, it is desired that the front-half path segments (and also the back-half path segments) interpolate some specific rulings of the side ruled surfaces. Moreover, at the contact lines, the motion direction of points on the tool axis should be parallel to the instantaneous motion direction of the same points on the side paths. Therefore we speak about G^1 interpolation of certain rulings of R_1 and R_2 . In addition to these boundary constraints, the whole front-half path should stay inside the space enclosed by the side paths. The considerations on milling precision lead to the following specific constraints in the representation of B-spline surfaces.

Position constraints (G^0 constraints). The end rulings of $C_i(s, t)$, which define the tool positions at both ends of the half path, are required to be some specific rulings of the side paths $R_1(s, t)$ or $R_2(s, t)$, i.e., $C_i(s, 0) = R_1(s, t_{i1})$, $s \in [0, 1]$ and $C_i(s, 1) = R_2(s, t_{i2})$, $s \in [0, 1]$. Using the clamped knot vector in the B-spline surface $C_i(s, t)$, this is easily met if the end control line of $C_i(s, t)$ connects two points lying on the top and bottom boundary curves of the side path at the same parameter t , respectively, see Fig. 6 for an illustration. In particular, we have

$$\begin{aligned} \mathcal{C}_{pos}^{1,*} &\equiv \mathbf{a}_0^* - \mathbf{q}_1^*(t_{i1}) = 0 \\ \mathcal{C}_{pos}^{2,*} &\equiv \mathbf{a}_m^* - \mathbf{q}_2^*(t_{i2}) = 0 \end{aligned} \quad (14)$$

where $* \in \{T, B\}$ with T and B indicate the top curve and the bottom curve of C_i , respectively. Note that the parameters on R_1 and R_2 are in general not identical, i.e., $i1 \neq i2$. The parameters $\mathcal{T} = \{t_{i1}, t_{i2}\}$ are treated as unknowns in our optimization and their computation will be discussed later in Sec. 3.3.3.

Motion direction constraints (G^1 constraints). The instantaneous motion vectors of points at the tool axis, which are the contact lines, should be proportional to the motion vectors of the same points on the side path. This G^1 continuity constraint of the motion of the tool axis is expressed as

$$\begin{aligned} C'_i(s, 0) &= \alpha_1 R'_1(s, t_{i1}), \quad s = 0, 1 \\ C'_i(s, 1) &= \alpha_2 R'_2(s, t_{i2}), \quad s = 0, 1 \end{aligned} \quad (15)$$

where the prime symbol indicates the derivative w.r.t. t . Note that R_1 (and R_2) is a rigid body motion of the tool axis and therefore the instantaneous vector field that moves the points of the ruling is linear in s . Consequently, satisfying the constraints in Eq. (15) at the parameters $s = 0$ and $s = 1$ implies it for all $s \in [0, 1]$. Since we work with clamped B-spline curves, Eq. (15) can be reformulated in terms of control points of \mathbf{a}_i^* as follows

$$\begin{aligned} \mathcal{C}_{tan}^{1,*} &\equiv \|(\mathbf{a}_1^* - \mathbf{a}_0^*) - \alpha_1 \cdot (\mathbf{q}_1^*)'(t_{i1})\|^2 = 0 \\ \mathcal{C}_{tan}^{2,*} &\equiv \|(\mathbf{a}_{m-1}^* - \mathbf{a}_m^*) - \alpha_2 \cdot (\mathbf{q}_2^*)'(t_{i2})\|^2 = 0 \end{aligned} \quad (16)$$

where α_1, α_2 are positive constants and positive sign corresponds to the correct direction of the motion. Note that these constants can be set arbitrarily as they correspond just to a reparametrization of the ruled surface C_i . Since we work with cubic splines, we set $\alpha_1 = \alpha_2 = \frac{1}{3}$ as this corresponds to control point match in the case of the boundary positions of the ruling $R_i(s, 0)$ and $R_i(s, 1)$, $i = 1, 2$, see Fig. 6.

The constraints in Eq. (14) and Eq. (16) concerning milling precision of the side surfaces will also be used later in our algorithm of TR path computation.

3.3.2. MRR and CWE control

Except for the precision objectives discussed above, it is also desired to maximize MRR while controlling the CWE throughout the milling paths. This goal can be formulated as a constrained optimization problem. Towards this end, we now formally define the MRR and the CWE quantities.

Definition of MRR. The Material Removal Rate (MRR) is defined as the removed volume between two paths C_i and C_{i+1} over a unit of time. Let the front patch of the envelope of the tool motion along C_{i+1} be denoted by E_{i+1} . The volume \mathcal{V}_{i+1} milled by the tool along C_{i+1} is the space enclosed by two consecutive envelope surfaces E_i and E_{i+1} , as shown in Fig. 7. The machining time of C_{i+1} is approximately proportional to the average of the lengths of the top and bottom curves of C_{i+1} , denoted by \mathcal{L}_{i+1} . Therefore, MRR can be approximately represented by

$$\mathcal{M}_{i+1} = a \cdot \frac{\mathcal{V}_{i+1}}{\mathcal{L}_{i+1}}. \quad (17)$$

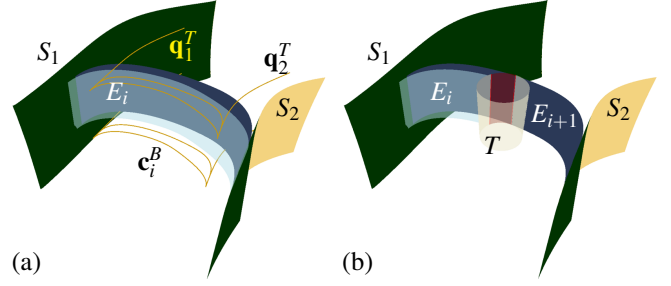


Figure 7: MRR and CWE. (a) E_i and E_{i+1} represent the envelopes generated by T in the i -th and $(i+1)$ -st front-half motion, respectively. MRR is expressed as the volume between E_i and E_{i+1} , milled per unit of time. (b) CWE is the contact area (red) between the cutting tool and the material during the movement, which is bounded by the intersection curves between the cutting tool T , E_i , and E_{i+1} .

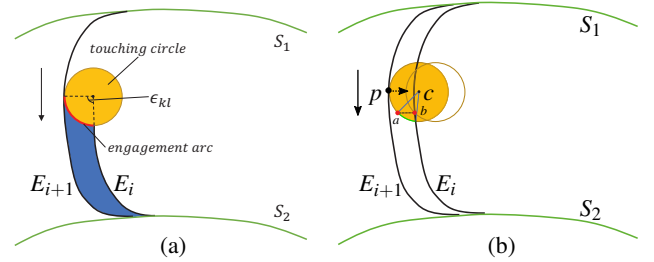


Figure 8: 2D simplification of the CWE. (a) A touching circle is perpendicular to the tool axis and an engagement arc (red) is the part of the circle where it meets the material. ϵ_{kl} is the angle that corresponds to the engagement arc. (b) Touching circle movement. The touching circle is moved in the direction of the surface normal of the envelope E_{i+1} , $-V_{kl} = c - p$, by distance d , where d is decided using the engagement angle ϵ^* , indicated by the green arc.

where a is a constant value that corresponds to the average velocity needed for milling \mathcal{L}_{i+1} . In the end, the units of (17) are mm^3/s (volume per time).

Definition of CWE. For real-life 3D trochoidal milling, the CWE area continuously changes over time in a complicated manner, as shown in Fig. 7(b) and it is a common practice to make the argument easier by considering a 2D simplification, see e.g. [16]. Denote the CWE area by $\Sigma(t)$ which is time-dependent. For a time instant of the cutting tool corresponding to the parameter t_k , $\Sigma(t_k)$ is the contact area of the tool with the raw material. This area on the tool's surface is bounded by two curves: i) the intersection curve of the tool surface with E_i and ii) the characteristic curve on E_{i+1} . To represent this area, we consider the circles on the tool surface which are perpendicular to the tool axis. The axis is parametrized by s and the circle at the parameter s_l of the axis of the tool touching the material is called a touching circle. The part of the touching circle in the engagement area is called an engagement arc. The interior angle of the engagement arc is called an engagement angle which generally varies along the tool axis and can be defined by $\epsilon_{kl} = \epsilon_k(s_l)$, see Fig. 8. In physical machining, ϵ_{kl} is required to be restricted under a specific tolerance, i.e., $\epsilon_{kl} < \epsilon^*$, for all the 2D cuts, i.e., for all l .

Problem formulation. Our aim is to compute a set of front-half paths and we do this in an iterative fashion. That is, once C_i is computed, we construct C_{i+1} with MRR and CWE control,

which is possible because both CWE and MRR depend only on C_i and C_{i+1} . Assuming C_i is computed (and fixed), the computation of C_{i+1} considering the control of MRR and CWE can be formulated as a constrained optimization problem as follows

$$\begin{aligned} & \max_{\mathcal{P}_{i+1}, t_{i1}, t_{i2}} \mathcal{M}_{i+1} \\ & \text{subject to } \begin{cases} \varepsilon_{kl} < \varepsilon^*, \text{ for samples } k, \text{ and } l \\ C_{i+1} \text{ meets the constraints in Eq.(14) and Eq.(16)} \end{cases} \end{aligned} \quad (18)$$

We recall that the parameters t_{i1} , t_{i2} of the end rulings of C_{i+1} , in addition to the control points of C_{i+1} , are also unknowns. The solution of the problem in Eq.(18) defines a ruled surface C_{i+1} which is the path of the tool along which the CWE meets the tolerance constraints. However, due to the complexity of the functions involved, solving Eq.(18) with standard numerical optimization method is difficult and an optimal solution can be hardly found efficiently and robustly. In the following, we propose a practical algorithm to obtain a reasonably good path that is robust and easy to implement.

3.3.3. Optimization algorithm

In order to find a reasonably good front-half path C_{i+1} as an approximate solution of Eq.(18), we have to deal with \mathcal{V}_{i+1} , \mathcal{L}_{i+1} and the CWE generated by C_{i+1} . It is non-trivial to find an optimal C_{i+1} meeting all objectives since the quantities depend on C_{i+1} in a complicated, highly non-linear, manner. Notice that both \mathcal{V}_{i+1} and the CWE are directly defined by E_{i+1} and therefore instead of looking for C_{i+1} , we focus on the corresponding envelope E_{i+1} first, and compute C_{i+1} from it by the surface fitting algorithm. Observe that \mathcal{L}_{i+1} directly depends on C_{i+1} and can be integrated into the envelope surface fitting algorithm with a function term penalizing the length of the path.

Path optimization. Firstly, we propose a sub-algorithm which computes a ruled surface between two rulings on the left and right side path, respectively. This is achieved via envelope fitting with the G^1 constraints defined by Eq.(14) and Eq.(16). Notice that the machining time is proportional to the length of the curve traversed by a particular tool point, assuming a constant speed of the tool. Consequently, in order to reduce the machining time, the length the tool path should be penalized and this is achieved by a boundary curve's penalization term as

$$F_{length} = \int \|(\mathbf{c}_i^T)'(t)\|^2 dt + \int \|(\mathbf{c}_i^B)'(t)\|^2 dt, \quad (19)$$

recall Fig. 6. The algorithm of computing a ruled surface between two side paths is formally defined as follows

$$\begin{aligned} & \min_{\mathcal{P}_{inner}} \beta F_{dist} + \lambda_1 F_{fair} + \lambda_2 F_{rigid} + \lambda_3 F_{length} \\ & \text{subject to } \mathcal{C}_{pos}^{1,*} = 0, \mathcal{C}_{pos}^{2,*} = 0, \mathcal{C}_{tan}^{1,*} = 0, \mathcal{C}_{tan}^{2,*} = 0, * \in \{T, B\} \end{aligned} \quad (20)$$

where the variables \mathcal{P}_{inner} in the optimization are the control points of the ruled surface excluding the end control points. That is, the end rulings of the ruled surface are fixed in the optimization. The lengths of the boundary curves can be controlled with the energy term F_{length} which has the effect of shorting the milling time. This sub-algorithm also works when there is

no reference surface by setting $\beta = 0$ in the objective function (20). For the cases with a reference surface, $\beta = 1$ is used.

Path searching. Assuming C_i being fixed, our method to compute C_{i+1} consists of an initialization stage and an adjustment stage. In the initialization stage, we look for a front-half path that maximizes MRR, by moving a candidate path in the forward direction, until the CWE constraint is violated. In the adjustment stage, we move backwards the path obtained in the initialization step to satisfy the CWE constraint globally.

- *Initialization stage.* We need to determine the two end rulings of C_{i+1} which coincide with some rulings of R_1 and R_2 . First, the end rulings are set to the end ruling lines of C_i , by setting $t_{i+1,1} = t_{i,1}$ and $t_{i+1,2} = t_{i,2}$, where $t_{i,1}$ and $t_{i,2}$ are parameters of R_1 and R_2 , respectively. We then increase $t_{i+1,1}$ and $t_{i+1,2}$ iteratively with identical incremental value Δt_{i+1} for both sides, resulting in a pair of lines $R_1(s, t_{i+1,1})$ and $R_2(s, t_{i+1,2})$ stepping forward.

For every time instant during this iterative process, we construct a ruled surface C_{i+1}^0 between $R_1(s, t_{i+1,1})$ and $R_2(s, t_{i+1,2})$, and check the CWE constraint by measuring ε_{kl} at some sampled points; we sample the ruled surface by a quad mesh with 20×100 quads, 100 in time (t) and 20 in the ruling (s) directions. C_{i+1}^0 is computed by solving the path optimization algorithm (see Eq. (20)) by setting $\beta = 0$.

To maximize the MRR, the inner control points are moved in the direction of the t -derivatives of the side surfaces, i.e., their initial values are set as $\mathbf{a}_{i+1,j}^* = \mathbf{a}_{i,j}^* + \mathbf{v}^*$, for $j = 1, \dots, m-1$, where $\mathbf{v}^* = (\mathbf{a}_{i+1,0}^* - \mathbf{a}_{i,0}^* + \mathbf{a}_{i+1,m}^* - \mathbf{a}_{i,m}^*)/2$, $* \in \{T, B\}$ and moved forward by an iterative process which stops once the CWE constraint is violated. The ruled surface at this moment is denoted by C_{i+1}^1 and is the input for the adjustment stage.

- *Adjustment stage.* In this stage, we aim to move back some parts of the path C_{i+1}^1 obtained in the initialization stage where the CWE is violated. It is difficult to directly modify C_{i+1}^1 to ensure its envelope E_{i+1}^1 meeting the CWE constraint (due to the fact that the envelope is a one parameter family of characteristics that change dynamically in time). Therefore, we approach the problem in the reverse order, i.e., we update the envelope first, such that the CWE constraint is met, and consequently compute C_{i+1}^1 via envelope fitting. Since the CWE is directly related to the envelope surface, modifying the envelope to satisfy the CWE constraint is much easier than modifying the path. To modify the envelope, we measure again ε_{kl} at the samples and update E_{i+1}^1 where $\varepsilon_{kl} < \varepsilon^*$ is violated. The process iteratively executes two steps as follows. See Fig. 8(b).

(1) Let p be a contact point of the envelope E_{i+1}^1 and the tool T , and c be the center of the circle of the tool touching p . Then the vector $V_{kl} = p - c$ is perpendicular to both E_{i+1}^1 and T which is the direction along the fastest increase of the step-over distance. Therefore, if the engagement angle associated with p is larger than the tolerance, we move back the point p on E_{i+1}^1 by $-d \cdot \frac{V_{kl}}{\|V_{kl}\|}$ to decrease

the CWE. To decide d , we first find the position a on the circle whose engagement angle is equal to ε^* . Let b be the intersection point of the line defined by a and V_{kl} and the envelope in previous cycle E_i . Then d is the distance between a and b , i.e. $d = \|a - b\|$.

(2) After moving back all sample points on E_{i+1}^1 violating the CWE constraint, a ruled surface C_{i+1}^2 is computed whose envelope fits to the updated envelope \hat{E}_{i+1}^1 . This is done by solving Eq.(20) with $\beta = 1$. Then, the envelope of C_{i+1}^2 , E_{i+1}^2 , is generated and tested again.

The two above described steps are wrapped into a single loop of a process to update E_{i+1}^q with q being the number of iterations. One loop of this procedure is shown in Fig. 9. The maximum number of iterations is set $q_{max} = 10$, but this limit has not been reached in any of our examples. The pseudocode of path searching is given in Algorithm 1.

Algorithm 1 Path Searching

```

1: procedure ALGORITHM( $C_i, C_{i+1}$ )
2:   Initialize  $t_{i+1,1}$  and  $t_{i+1,2}$ 
3:   repeat ▷ Initialization
4:     Increase  $t_{i+1,1}$  and  $t_{i+1,2}$  by  $\Delta t_{i+1}$ 
5:     Generate  $C_{i+1}^0$  via path optimization
6:     Measure  $\varepsilon_{kl}$  at some samples
7:      $\Delta t_{i+1} := \Delta t_{i+1} \cdot (1 - \max\{\varepsilon_{kl}\} / \varepsilon^*)$ 
8:   until  $\varepsilon_{kl} > \varepsilon^*$  for some sample
9:    $C_{i+1}^1 := C_{i+1}^0$ ,  $q := 1$ 
10:  while  $\max\{\varepsilon_{kl}\} > \varepsilon^*$  and  $q \leq q_{max}$  do ▷ Adjustment
11:    for all sample points  $p$  do
12:      if  $\varepsilon_{kl}$  associated with  $p$  is bigger than  $\varepsilon^*$  then
13:        Move back the point on  $E_{i+1}^q$  by  $-\frac{V_{kl}}{\|V_{kl}\|} \cdot d$ 
14:      end if
15:    end for
16:     $q := q + 1$ 
17:    Generate  $C_{i+1}^q$  with adjusted envelope as reference
18:  end while
19:   $C_{i+1} := C_{i+1}^q$ 
20: end procedure

```

First front-half path. The above procedure for computing a sequence of front-half paths C_i depends on C_0 , which is the front-half path of the first TR cycle. In our implementation, C_0 is generated by the path optimization algorithm with $\beta = 0$, with the initial control points uniformly distributed. The bisector surface H of S_1 and S_2 is computed and fitted with a ruled surface $R^H(s, t)$ as in [9] with s being the parameter in ruling direction and t in motion direction. To compute the bisector surface, we seek 3D points \mathbf{x} that are equally distant from the side surfaces, therefore we compute the signed minimal distances to S_1 and S_2 and define their difference as

$$\varepsilon(\mathbf{x}) = d(\mathbf{x}, S_1) - d(\mathbf{x}, S_2) \quad (21)$$

which measures the deviation of \mathbf{x} from H . We look for \mathbf{x} which satisfy $\varepsilon(\mathbf{x}) = 0$ and use a variant of the marching cubes algorithm which returns a triangular mesh that approximates H . To

define the end rulings of C_0 , the starting ruling line of R^H is projected to R_1 and R_2 , and the end rulings of C_0 are defined to be the best fitting lines to the projection points on R_1 and R_2 , respectively.

3.4. Back-half path computation and entire path generation

Once the front-half paths of the TR cycles are computed, the back-half paths are constructed to form the transition between two consecutive front-half paths. The construction of the back-half paths is easier than the computation of the front-half paths since the cutting tool does not touch the raw material, except for the region near the end of paths where the tool touches the side surfaces. A back-half path D_i is defined as a B-spline ruled surface

$$D_i(t, s) = \mathbf{d}_i^T(t) \cdot (1 - s) + \mathbf{d}_i^B(t) \cdot s, \quad t, s \in [0, 1], \quad (22)$$

where $\mathbf{d}_i^*(t) = \sum_{k=0}^m \mathbf{b}_k^* B_{k,d}(t)$, $* \in \{T, B\}$ are B-spline curves. In order to make D_i the transition from the last ruling of C_i to the start ruling of C_{i+1} , $D_i(t, s)$ is required to connect two lines $C_i(1, s)$ and $C_{i+1}(0, s)$. Analogously to Eq. (14), the G^0 connection constraints are defined as

$$\begin{aligned} \mathcal{C}_{pos}^{1,*} &\equiv \mathbf{b}_0^* - \mathbf{q}_1^*(t_{i1}) = 0 \\ \mathcal{C}_{pos}^{2,*} &\equiv \mathbf{b}_m^* - \mathbf{q}_2^*(t_{i2}) = 0 \end{aligned} \quad (23)$$

In addition to the connection constraints, the motion of the tool along the back-half paths should also join the side surfaces tangentially (in G^1 fashion) to maintain a high precision milling error guaranteed by the construction of R_1 and R_2 . These G^1 -continuity constraints are formulated analogously to Eq.(16) as

$$\begin{aligned} \mathcal{C}_{tan}^{1,*} &\equiv \|(\mathbf{b}_1^* - \mathbf{b}_0^*) + \lambda_1 \cdot (\mathbf{q}_1^*)'(t_{i1})\|^2 = 0 \\ \mathcal{C}_{tan}^{2,*} &\equiv \|(\mathbf{b}_{m-1}^* - \mathbf{b}_m^*) + \lambda_2 \cdot (\mathbf{q}_2^*)'(t_{i2})\|^2 = 0 \end{aligned} \quad (24)$$

where $* \in \{T, B\}$ with T and B indicate the top or the bottom curve, and λ_1 , and λ_2 are positive constants. Note that the velocity vectors have opposite signs than their counterparts on the ruled surfaces R_1 and R_2 due to the fact that D_i are the back-half paths.

4. Multi-layer paths for deep cavities

For trochoidal milling of deep cavities bounded by free-form surfaces, it is typically not possible to finish the milling with one single sweep because a long cutting tool is not practical, e.g. for the chattering reasons. In such a case, a multi-layer milling strategy is needed which divides the side surfaces into layers and each layer is processed with one trochoidal milling path. The milling regions of different layers may have overlapping patches, but the composition of milled regions should cover the whole side surfaces. The milling procedure starts from the top layer from where the cutting tool removes material from the material block, and processes the other layers from the top of the cavity to the bottom. It is also desired that the same tool is used for all the layers for the sake of time needed for the tool exchange and the cost of custom-shaped tools.

4.1. Side path initialization

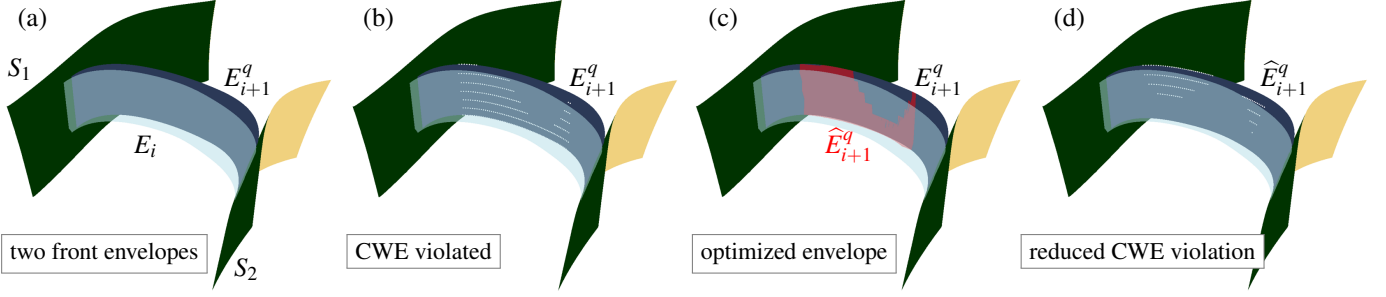
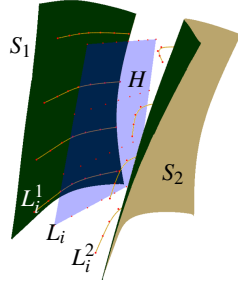


Figure 9: One iteration of the front envelope adjustment. (a) Two consecutive front envelopes E_i and E_{i+1} , connecting two side surfaces S_1 and S_2 , are shown. E_i is taken for granted and E_{i+1} gets optimized; the q -th iteration E_{i+1}^q is shown. (b) E_{i+1}^q violates the CWE test at 268 points (white dots). (c) \hat{E}_{i+1}^q gets adjusted (see Section 3.3.3) and becomes closer to E_i (red envelope) to reduce the CWE violation. (d) \hat{E}_{i+1}^{q+1} represents the optimized envelope, now the CWE is violated at only 127 points (white dots).

For the initialization of the side paths, we use a bisector surface H of the two side surfaces S_1 and S_2 . We further define a scalar function $f(\mathbf{x}) = d(\mathbf{x}, S_1)$, $\mathbf{x} \in H$. To compute a boundary curve of a ruled surface, an iso-line $f(\mathbf{x}) = \text{const.}$ on H is extracted and projected to the side surface S_i ($i = 1, 2$). The iso-lines are approximately the iso-parametric lines at some particular parameter s of the ruled surface $R^H(s, t)$ fitting to H . The projected curve is then moved along the surface normal of S_i ($i = 1, 2$) in a direction towards the inside of the cavity with a certain distance, which is decided by the preferred tool size which must be less than half of the width of the cavity. In order to create multiple paths, we uniformly select several iso-lines L_i on H and generate boundary curves L_i^1 and L_i^2 associated with S_1 and S_2 , respectively. The distance between iso-lines is decided assuming the tool length is roughly known. This construction forces the tool to have the first and the last radius of the same size (for a tool with two different limit radii, see discussion later in Section 5.4).



After we have a sequence of lines L_i^1 and L_i^2 ($i = 0, \dots, k$) on S_1 and S_2 , respectively, each pair of neighbouring lines can be used as boundary curves to define the initial paths. In order to avoid gaps between neighboring paths, the corresponding lines lying on the ruled surfaces are scaled up to either direction by a certain factor, which is set 5% in our implementation.

Special treatment of the top layer. For the top layer, it is common that a part of the tool lies outside the cavity. The sampling points on the tool axis which is located outside the cavity needs to be ignored during the side path calculation. These points \mathbf{p} can be recognized by checking their closest points on the free-form side surface. If the closest point of \mathbf{p} is located on the boundary of the side surface, \mathbf{p} is considered to be outside the cavity and does not contribute to the side path calculation. Note that for the part of the tool with the top layer laying outside the cavity, its top boundary curve cannot be found using the above discussed iso-line method. To resolve this issue, in our implementation, we modify the top ruled surface generated with our iso-line method by moving its top boundary curve in the direc-

tion of the tool axis to the outside of the cavity. However, we conclude that handling perfectly boundaries is a separate issue in many CNC projects, and goes also beyond the scope of this paper.

4.2. Side path computation

Once the side surface is divided into layers, we generate side paths for all layers. Since it is required that all layers are processed with the same cutting tool, we look for the motion paths for all layers simultaneously with shape parameters of a single cutting tool. Let F_{dist}^k and F_{fair}^k be the distance function term and fairness term, respectively, corresponding to the layer k . We solve the following minimization problem for side path computation for multi-layer milling

$$\min_{\mathcal{P}^1, \dots, \mathcal{P}^K, \mathbf{r}} \sum_{k=1}^K F_{dist}^k + \lambda_1 \sum_{k=1}^K F_{fair}^k + \lambda_2 \sum_{k=1}^K F_{rigid}^k, \quad (25)$$

where K is the number of layers, \mathcal{P}^k is a set of control points of the ruled surfaces in the k -th layer, and \mathbf{r} is the vector of radii set uniformly distributed along the tool axis. Note that only one single tool shape is used for all layers. F_{dist}^k and F_{fair}^k are defined similarly to Eq.(8) and Eq.(3), respectively. To avoid problems with surface patch boundaries and to make the motion path computation highly accurate, whole side surfaces are used as reference instead of their layer counterparts.

4.3. Trochoidal path computation for multi-layer milling

Once the side paths are obtained for all layers and the tool shape is determined, we can compute trochoidal path for each layer. We start with the top layer and process the remaining layers one by one. Each layer can be regarded as an independent TR cycle problem and all the front-half and the back-half paths of the single layer can be generated by the methods described in Sec. 3.3 and Sec. 3.4.

5. Experimental results

In this section, we verify the proposed 5-axis trochoidal milling methods described in Sec. 3 and Sec. 4 by performing experiments on three test cases. The algorithms are implemented in C++ language. The experimental environment is a desktop with CPU i7-10700K 3.80 GHz and 16G RAM.

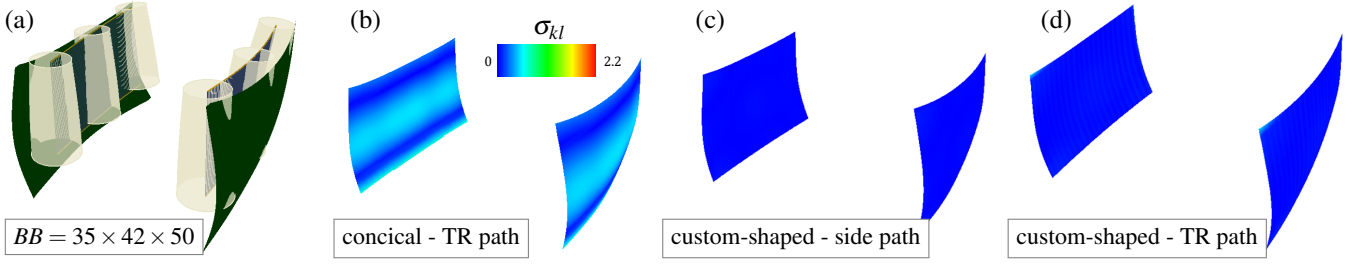


Figure 10: Trochoidal milling with conical vs. custom-shaped tools. (a) In the optimization of the side paths, we constraint the meridian (radial function) to be linear for a pair of side surfaces with a bounding box BB . (b) This results in the best conical cutting tool; its envelopes are color-coded by the error σ_{kl} , see Eq. (26), with the maximum error $\sigma_{max} = 0.6493$. (c) The envelopes of a single (side) path using the custom-shaped tool shown in Fig. 11(b), color-coded by σ_{kl} with $\sigma_{max} = 0.0869$. (d) The complete trochoidal paths, with almost the same error as for the side surface.

We measure the error of the envelope surface generated by cutting tool and the target surface in a discrete way, at samples of the ruled surfaces. The error σ_{kl} is the larger of the distance error between the points on envelope surface and the target side surfaces S_n , $n = 1, 2$, that is

$$\sigma_{kl} = \max_{n=1,2} |dist(R(t_k, s_l), \phi_{kl}) - r_l| \quad (26)$$

where $R(t_k, s_l)$ are the samples on the ruled surface, ϕ_{kl} are the corresponding footpoints on the target surface, and r_l are the radii of cutting tool corresponding to the l -sample in the ruling direction. All the color-coded examples shown in the paper reflect this absolute (non-signed) error. Finally, the total error is defined as

$$\sigma_{max} = \max_{k,l} \sigma_{kl}. \quad (27)$$

5.1. Single-layer TR paths for two free-form surfaces

We start with a test case where we consider a single-layer trochoidal milling, described in Sec. 3, for two free-form side surfaces. The side surfaces form a synthetic (symmetric) cavity that admits highly accurate approximation. The parameters in Eq.(16) that control the overcut direction are all set to 0.1.

Fig. 11 shows the optimization results. The error of the initial side path and the optimized side path are shown in Fig.11(a). For a cavity with a bounding box $BB = 35 \times 42 \times 50$, the error σ_{max} is optimized from the initial $\sigma_{max}^{ini} = 2.1950$ to $\sigma_{max}^{opt} = 0.0869$, which meets the accuracy required for semi-finishing operations. Fig. 11(c)-(d) show the final envelopes of the trochoidal paths computed by our method. A total of 22 trochoidal cycle paths are generated between the two free-form surfaces.

To give comparisons with traditional tools, we firstly give a comparison with conical tools. We employ our method with an additional constraint enforcing the tool to be conical. Fig.10 shows the shape and error of the conical tool where the machining error of the optimized paths reaches $\sigma_{max}^{opt} = 0.6493$, which is by order of magnitude worse than using a custom-shaped tool shown in Fig. 11.

Another comparison is made against a fixed barrel tool. The TR paths are optimized with our method, however, the shape of the tool stays constant. The barrel tool is defined by the radius function R_ϕ [44]

$$R_\phi = \frac{R_t - R_e(1 - \cos \phi)}{\cos \phi}, \phi \in [-\arcsin(L_e/2R_e), \arcsin(L_e/2R_e)],$$

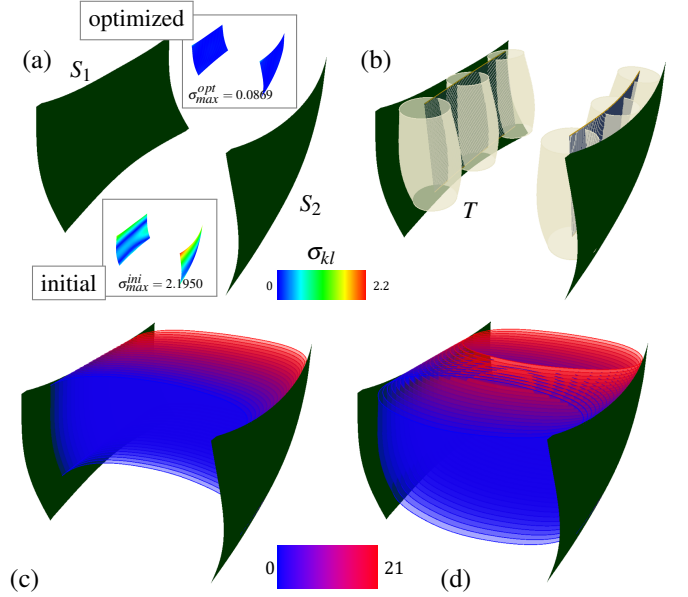


Figure 11: (a) Two free-form surfaces S_1 and S_2 define a cavity and their approximation by the envelopes of a custom-shaped tool T , before and after optimization, are shown framed. (b) The motion of T ; side paths and the tool are optimized by Eq.(11). (c) Front-half envelopes. There are 22 front-half envelopes computed by the approach described in Sec. 3.3 and color-coded by the number of the TR cycle. (d) The whole cavity filled by the TR envelopes of T .

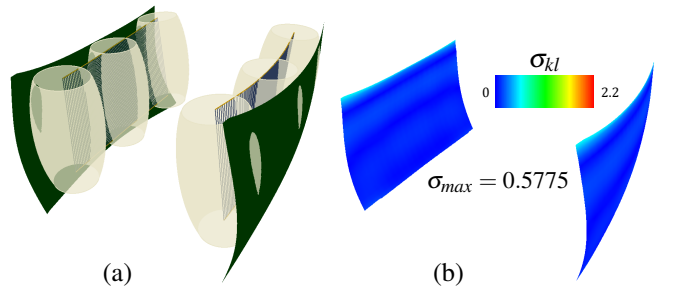


Figure 12: Trochoidal milling with a fixed barrel tool. (a) TR milling with a fixed barrel tool where only the milling path is optimized. The color coding of machining error is shown in (b). While the maximum error is better than for the best conical tool, Fig. 10(b), it is more than six times worse than using a custom-shaped tool, see Fig. 10(d).

where ϕ is the angle of a circular arc generatrix. We choose reasonable shape parameters for the barrel tool suggested by the side surfaces to be milled, i.e. $R_e = 15$, $R_t = 6$. The length of the tool axis is set $L_e = 16$. Fig.12 shows the results where

Model	# vertices	BB	σ_{max}^{ini}	σ_{max}^{opt}	min:sec
Fig. 11	1470	$35 \times 42 \times 50$	2.1950	0.0869	18:20
Fig. 15	1903	$25 \times 200 \times 200$	1.0434	0.0751	41:44
Fig. 17	3073	$35 \times 82 \times 100$	3.3358	0.1340	130:03

Table 1: The statistics of the TR algorithm. In turn, we report the number of the mesh vertices ($S_1 \cup S_2$), the size of the bounding box of $S_1 \cup S_2$, the approximation error before and after optimization, and the computation time.

the maximal error is $\sigma_{max} = 0.5775$, which is more than six times more than using a custom-shaped tool, c.f. Fig. 11(a). and Fig. 10(d).

Fig. 13 shows the change of the average cutter-workpiece engagement angle with position in the front-path movement of the 1st, 7th, and 14th trochoidal cycle. Fig. 14(a) shows the maximum value of CWE in each TR cycle, controlled by our algorithm below the given tolerance $\epsilon_{kl}^* = 60^\circ$. Fig. 14(b) shows the material removal rate in each TR cycle and Fig. 14(c) depicts the correlation of the optimization costs and computational timings in each iteration.

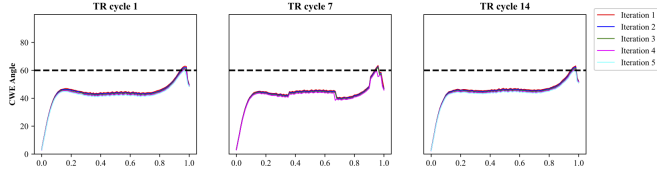


Figure 13: CWE optimization. Cutter-workpiece engagement angles for three front-half paths. The figures show the optimization process of the 1st, 7th, and 14th rounds of the TR cycles, respectively. Each round of TR cycle is generally generated by 3 ~ 7 optimizations. The tolerance ϵ_{kl}^* was set to 60° .

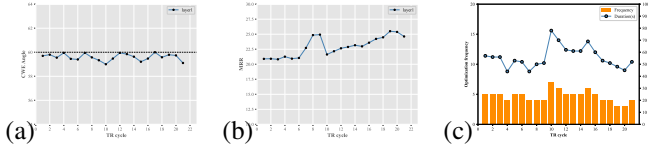


Figure 14: (a) The maximum CWE angle of the cutting tool with front-half movement of each TR cycle. The CWE angle of each iteration is smaller than the tolerance ϵ_{kl}^* . (b) The MRR information of each TR cycle. (c) The polyline shows the cost time of each round path generation. The histogram shows the number of optimizations for each iteration, which is positively correlated with the cost time of each iteration.

5.2. Multi-layer TR paths for the industrial blisk model

The second experiment is the cavity of the blisk model shown in Fig 1. Due to the depth of the cavity, we adopt the multi-layer TR milling strategy described in Section 4 and consider three layers. We use Eq. (25) to optimize the cutter shape (represented by the vector of radii \mathbf{r}), simultaneously for the three-layer side paths. The weights of F_{fair} and F_{rigid} in Eq. (25) are set to $1.0 \times e^{-6}$ and 0.1, respectively. We chose 10 control points for the single-side motion path of each layer, and the number of sampling points on the tool axis is set to 21. The optimization results of the side path and tool shape are shown in Fig.15. In this experiment, the weights of F_{fair} and F_{length} in

Eq. (20) were both set to $1.0 \times e^{-5}$ and the weight of F_{rigid} was set to 0.1. The parameters in Eq.(16) were all set to $1/3$.

Fig. 15 shows the results of the multi-layer approach, compared with a single layer; in both approaches the shape of the tool is computed (\approx optimized). The multi-layer approach generates in turn (from the bottom to top of the cavity) 17, 19, and 21 trochoidal paths. The difference is related to the fact that the size of the cavity is smallest at the bottom. The multi-layer approach returns finer approximation, see the color-maps in Fig. 15(a) and (b). The optimal tool is close to a conical, yet it is curvature-varying, see Fig. 15(d) top-framed.

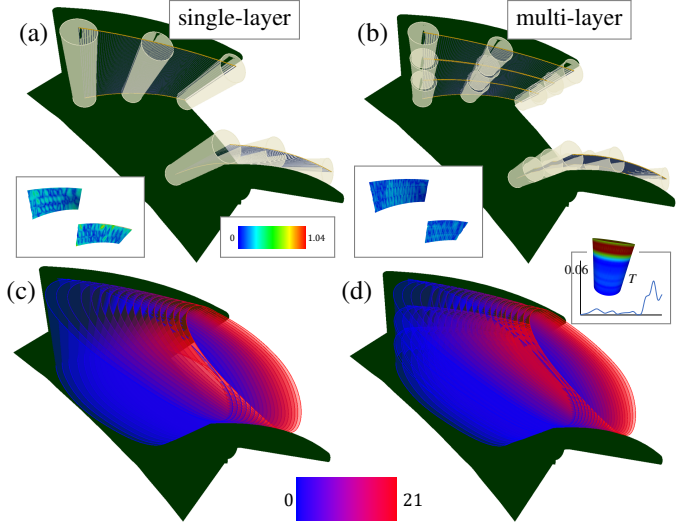


Figure 15: Layer refinement. Single-layer (left) vs. multi-layer (right) trochoidal milling are compared. The side paths (framed) are color coded by the distance error σ_{max} . While the error in the case of a single layer $\sigma_{max} = 0.3837$, using three layers the error gets reduced to $\sigma_{max} = 0.0751$. (c-d) The final TR paths of our algorithm. The color coding here shows the number of trochoidal cycle. (d) top-framed. The optimized T color-coded by the Gaussian curvature and the curvature plot of the meridian curve of T .

Fig.16(a) shows the maximum value of the CWE in each round of front-path in each layer of milling and Fig.16(b) shows the MRR of the front-path of each round of the TR cycle.

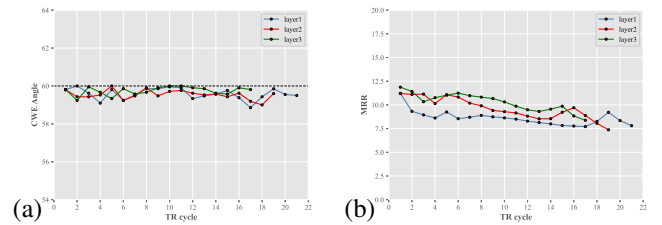


Figure 16: (a) CWE angle and MRR for a TR path with 3 layers, c.f. Fig. 15(b).

5.3. Multi-layer TR paths for second real blisk model

The third test example uses another industrial blisk dataset. The optimization parameters and the number of control points were set equally to those in Sec. 5.2. Since this cavity is deeper, we chose five instead of three layers, each one generated 27 ~ 31 rounds of trochoidal paths. The side paths are

shown in Fig.17(a) and the complete TR envelopes are shown in Fig.17(b). The successive optimization process is visualized in Fig.17(c+d) and a comparison against a single-layer approach is shown in Fig.17(e). Again, the errors are by the order of magnitude better in favor of the multi-layer strategy. The computation times and other statistics of all three test cases are listed in Table 1.

5.4. Discussion & limitations

Optimal tool selection. We initialized the tool such that its side (flank) motion fits the side surfaces, however, its size, in the terms of thickness, remained uncontrolled in our optimization and one could also look for the thickest tool that fits the cavity, as this should further reduce the machining time. We also slightly restricted the space of tools as we unified the L_i paths in the initialization (in Section 4.1), which resulted in equal limit radii $r_1 = r_M$. However, these radii were not constrained to be equal in the optimization stage.

G^1 connectivity between strips. In the multilayer approach, the neighboring layers are constructed to avoid gaps between neighboring layers, however, there might be a small overlap between neighboring layers and one could further optimize the layers to match better, e.g., in a G^1 fashion.

Global collision detection. Our algorithm tests only local collision, i.e., the collision of the custom-shaped tool and the side surfaces. As the tool is typically mounted on a handler/shank, there could be, however, collision of that part with the cavity. We checked the global collisions of the tools' axis as a post-process, but a more thorough global collision test shall be done in the case of physical experiments.

Global optimization. Our approach looks for a tool that minimizes the distance error between the motion of the tool and two side surfaces of the cavity. The problem is formulated as an optimization problem and we look for a minimizer. There is no guarantee that our method finds the global minimizer (which may not be unique), however, our results show that for cavities of industrial benchmarks, such as the blisk geometry, sufficiently accurate approximation (tens of micrometers) exists.

Existence of an exact tool. An exact tool exists iff the two side surfaces are exact envelopes of a motion of such a tool. One can construct a (counter-)example of a cavity where a single tool cannot give sufficiently good results; e.g. such a cavity can be formed by one convex elliptic side surface and other one concave elliptic.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62072139), the Spanish Ministry of Science, Innovation and Universities, grant No PID2019-104488RB-I00, BCAM Severo Ochoa accreditation (SEV-2017-0718), the BERC 2022-2025 program, and by the European Unions Horizon 2020 program under grant agreement No 862025. Michael Bartoń was supported by the Ramón y Cajal fellowship RYC-2017-22649.

6. Conclusions

In this work, a method for trochoidal milling of 3D cavities bounded by free-form surfaces has been proposed. Our method computes not only the milling paths, but also the shape of the cutting tool itself, both in a single- and multi-layer setup. Material removal rate and cutter-workpiece engagement are also incorporated inside our variational framework and easy to control. The proposed method is validated on both synthetic and industrial benchmarks, and returns highly accurate milling paths that meet fine machining tolerances.

7. References

- [1] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri, "The status, challenges, and future of additive manufacturing in engineering," *Computer-Aided Design*, vol. 69, pp. 65–89, 2015.
- [2] B. H. Kim and B. K. Choi, "Guide surface based tool path generation in 3-axis milling: an extension of the guide plane method," *Computer-Aided Design*, vol. 32, pp. 191–199, 2000.
- [3] J. Senatore, S. Segonds, W. Rubio, and G. Dessein, "Correlation between machining direction, cutter geometry and step-over distance in 3-axis milling: Application to milling by zones," *Comput. Aided Des.*, vol. 44, pp. 1151–1160, 2012.
- [4] J. Senatore, F. Monies, Y. Landon, and W. Rubio, "Optimising positioning of the axis of a milling cutter on an offset surface by geometric error minimisation," *Int. J. Adv. Manuf. Technol.*, vol. 37, pp. 861–871, 2008.
- [5] S. Bedi, S. Mann, and C. Menzel, "Flank milling with flat end milling cutters," *CAD*, vol. 35, no. 3, pp. 293–300, 2003.
- [6] C. Y. Wu, "Arbitrary surface flank milling of fan, compressor, and impeller blades," *Transactions on ASME. Journal of Engineering for Gas Turbines and Power*, vol. 117, pp. 534–539, 1995.
- [7] K. Xu, J. Wang, C.-H. Chu, and K. Tang, "Cutting force and machine kinematics constrained cutter location planning for five-axis flank milling of ruled surfaces," *Journal of Computational Design and Engineering*, 2017, in press.
- [8] D. Roth, S. Bedi, F. Ismail, and S. Mann, "Surface swept by a toroidal cutter during 5-axis machining," *Computer-Aided Design*, vol. 33, no. 1, pp. 57–63, 2001.
- [9] P. Bo, H. González, A. Calleja, L. N. L. de Lacalle, and M. Bartoń, "5-axis double-flank CNC machining of spiral bevel gears via custom-shaped milling toolspart I: Modeling and simulation," *Precision Engineering*, vol. 62, pp. 204–212, 2020.
- [10] M. Bizzarri and M. Bartoń, "Manufacturing of screw rotors via 5-axis double-flank CNC machining," *Computer-Aided Design*, vol. 132, p. 102960, 2021.
- [11] G. G. Escudero, P. Bo, H. González-Barrio, A. Calleja-Ochoa, M. Bartoń, and L. N. L. de Lacalle, "5-axis double-flank CNC machining of spiral bevel gears via custom-shaped toolspart II: physical validations and experiments," *The International Journal of Advanced Manufacturing Technology*, pp. 1–12, 2021.
- [12] N. Kardes and Y. Altintas, "Mechanics and Dynamics of the Circular Milling Process," *Journal of Manufacturing Science and Engineering*, vol. 129, pp. 21–31, 01 2006.
- [13] A. Pleta and L. Mears, "Cutting force investigation of trochoidal milling in nickel-based superalloy," *Procedia Manufacturing*, vol. 5, pp. 1348–1356, 2016.
- [14] A. Pleta, D. Ulutan, and L. Mears, "Investigation of trochoidal milling in nickel-based superalloy inconel 738 and comparison with end milling," in *International Manufacturing Science and Engineering Conference*, vol. 45813, p. V002T02A058, American Society of Mechanical Engineers, 2014.
- [15] B. H. Wu, C. Y. Zheng, M. Luo, and X. D. He, "Investigation of trochoidal milling nickel-based superalloy," in *Materials Science Forum*, vol. 723, pp. 332–336, Trans Tech Publ, 2012.
- [16] Z. Li, L. Chen, K. Xu, Y. Gao, and K. Tang, "Five-axis trochoidal flank milling of deep 3d cavities," *Computer-Aided Design*, vol. 119, p. 102775, 2020.

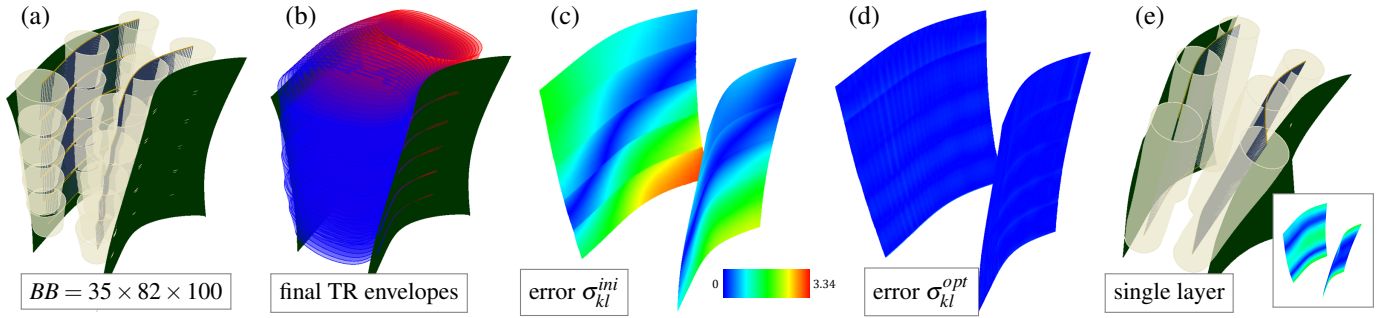


Figure 17: Another blisk geometry. (a) The cavity is filled by 5 layers of a custom-shaped tool. (b) Envelopes of the TR path. The approximation error before the optimization $\sigma_{max}^{ini} = 3.335$ (c) gets reduced to $\sigma_{max}^{opt} = 0.1340$ (d). (e) The same cavity approximated with a single-layer approach with less accurate approximation $\sigma_{max}^{opt} = 2.45$ (framed).

- [17] M. Otkur and I. Lazoglu, "Trochoidal milling," *International Journal of Machine Tools and Manufacture*, vol. 47, no. 9, pp. 1324–1332, 2007.
- [18] A. Pleta, F. A. Niaki, and L. Mears, "Investigation of chip thickness and force modelling of trochoidal milling," *Procedia Manufacturing*, vol. 10, pp. 612–621, 2017.
- [19] B. Wu, X. Yan, M. Luo, and G. Gao, "Cutting force prediction for circular end milling process," *Chinese Journal of Aeronautics*, vol. 26, no. 4, pp. 1057–1063, 2013.
- [20] F. A. Niaki, A. Pleta, and L. Mears, "Trochoidal milling: investigation of a new approach on uncut chip thickness modeling and cutting force simulation in an alternative path planning strategy," *The International Journal of Advanced Manufacturing Technology*, vol. 97, no. 1, pp. 641–656, 2018.
- [21] N. Kardes and Y. Altintas, "Mechanics and dynamics of the circular milling process," *Journal of Manufacturing Science and Engineering*, vol. 129, pp. 21–31, 01 2006.
- [22] R. Yan, H. Li, F. Peng, X. Tang, J. Xu, and H. Zeng, "Stability prediction and step optimization of trochoidal milling," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 9, 2017.
- [23] Q.-H. Wang, S. Wang, F. Jiang, and J.-R. Li, "Adaptive trochoidal tool-path for complex pockets machining," *International Journal of Production Research*, vol. 54, no. 20, pp. 5976–5989, 2016.
- [24] K. Xu, B. Wu, Z. Li, and K. Tang, "Time-efficient trochoidal tool path generation for milling arbitrary curved slots," *Journal of manufacturing science and engineering*, vol. 141, no. 3, 2019.
- [25] Z. Li, K. Xu, and K. Tang, "A new trochoidal pattern for slotting operation," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 5, pp. 1153–1163, 2019.
- [26] J. Ren, C. Yao, D. Zhang, Y. L. Xue, and Y. S. Liang, "Research on tool path planning method of four-axis high-efficiency slot plunge milling for open blisk," *The International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1-2, pp. 101–109, 2009.
- [27] D. J. Sheehy, C. G. Armstrong, and D. J. Robinson, "Shape description by medial surface construction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 1, pp. 62–72, 1996.
- [28] L. Zhaoyu, H. Pengcheng, X. Fubao, and T. Kai, "A variable-depth multi-layer five-axis trochoidal milling method for machining deep freeform 3d slots," *Robotics and Computer-Integrated Manufacturing*, vol. 68, p. 102093, 2021.
- [29] C. Li, S. Bedi, and S. Mann, "Flank milling of a ruled surface with conical tools – an optimization approach," *Int. J. Adv. Manuf. Technol.*, vol. 29, pp. 1115i–1124, 2006.
- [30] C. Li, S. Bedi, and S. Mann, "Flank millable surface design with conical and barrel tools," *Computer-Aided Design and Applications*, vol. 5, pp. 461–470, 2008.
- [31] K. Tang and C. Wang, "Modeling developable folds on a strip," *Journal of Computing and Information Science in Engineering*, vol. 5, no. 1, pp. 35–47, 2005.
- [32] C. Wang and K. Tang, "Optimal boundary triangulations of an interpolating ruled surface," *Journal of Computing and Information Science in Engineering*, vol. 5, no. 4, pp. 291–301, 2005.
- [33] J. Machchhar, D. Plakhotnik, and G. Elber, "Precise algebraic-based swept volumes for arbitrary free-form shaped tools towards multi-axis CNC machining verification," *Computer-Aided Design*, vol. 90, pp. 48–58, 2017.
- [34] A. Calleja, P. Bo, H. González, M. Bartoñ, and L. N. L. de Lacalle, "Highly accurate 5-axis flank CNC machining with conical tools," *The International Journal of Advanced Manufacturing Technology*, vol. 97, no. 5, pp. 1605–1615, 2018.
- [35] M. Skopenkov, P. Bo, M. Bartoñ, and H. Pottmann, "Characterizing envelopes of moving rotational cones and applications in CNC machining," *Computer Aided Geometric Design*, vol. 83, p. 101944, 2020.
- [36] P. Bo, M. Bartoñ, and H. Pottmann, "Automatic fitting of conical envelopes to free-form surfaces for flank CNC machining," *Computer-Aided Design*, vol. 91, pp. 84–94, 2017.
- [37] C. Wang and G. Elber, "Multi-dimensional dynamic programming in ruled surface fitting," *Computer-Aided Design*, vol. 51, pp. 39–49, Feb. 2014.
- [38] G. Elber and R. Fish, "5-axis freeform surface milling using piecewise ruled surface approximation," *Journal of Manufacturing Science and Engineering*, vol. 119, pp. 383–387, 08 1997.
- [39] J.-M. Redonnet, W. Rubio, and G. Dessein, "Side milling of ruled surfaces: Optimum positioning of the milling cutter and calculation of interference," *The International Journal of Advanced Manufacturing Technology*, vol. 14, no. 7, pp. 459–465, 1998.
- [40] J. Senatore, Y. Landon, and W. Rubio, "Analytical estimation of error in flank milling of ruled surfaces," *Computer-Aided Design*, vol. 40, no. 5, pp. 595–603, 2008.
- [41] L. Zhu, H. Ding, and Y. Xiong, "Simultaneous optimization of tool path and shape for five-axis flank milling," *Computer-Aided Design*, vol. 44, no. 12, pp. 1229–1234, 2012.
- [42] Y. Lu, Q. Bi, and L. Zhu, "Five-axis flank milling of impellers: Optimal geometry of a conical tool considering stiffness and geometric constraints," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacturing*, vol. 230, no. 1, pp. 38–52, 2016.
- [43] P. Bo, M. Bartoñ, D. Plakhotnik, and H. Pottmann, "Towards efficient 5-axis flank CNC machining of free-form surfaces via fitting envelopes of surfaces of revolution," *Computer-Aided Design*, vol. 79, pp. 1–11, 2016.
- [44] F.-J. Meng, Z.-T. Chen, R.-F. Xu, and X. Li, "Optimal barrel cutter selection for the CNC machining of blisk," *Computer-Aided Design*, vol. 53, no. Complete, pp. 36–45, 2014.