The International Journal of Advanced Manufacturing Technology (2019) 102:1–16
https://doi.org/10.1007/s00170-018-3182-4

ORIGINAL ARTICLE

Prospects for using T-splines for the development of the STEP-NC-based manufacturing of freeform surfaces

Gang Zhao[1,2] · Oleksandr Zavalnyi[1] · Yazui Liu[1] · Wenlei Xiao[1,2]

Received: 30 March 2018 / Accepted: 11 December 2018 / Published online: 18 December 20 18

Abstract
For successful development of the intelligent manufacturing of freeform surfaces using STEP-CNC with online toolpath generation capability, it is required to make a choice of the optimal representation of a 3D model which will be used for machining. Traditionally, most existing CAD-CAM systems use NURBS to design freeform surfaces and to perform toolpath generation in order to machine them. The introduction of T-splines to CAD systems and some reported results of using them in manufacturing makes it possible to consider T-splines, or more generally T-NURCCs (Non-Uniform Rational Catmull-Clark Surfaces with T-junctions), as a good solution for the development of the STEP-NC-based manufacturing of freeform surfaces because of their advantages over NURBS. Therefore, this paper gives an overview of the main arguments in favor of choosing the T-spline surface representation for integration within STEP-CNC systems. We examine the prospects for T-splines to become an integral part of modern manufacturing systems, and highlight some important properties of T-splines which are the most beneficial for manufacturing processes. The paper presents the results of the development of a complete T-spline-enabled STEP-CNC system which can strategically support online toolpath generation for three-axis ball end machining of simple T-spline surfaces using four different freeform strategies defined in ISO 14649-11. These results represent the implementation of the first stage of the development process of intelligent STEP-CNC systems, and in the future more research is needed in this direction.

Keywords STEP-NC · STEP-CNC · Toolpath generation · Freeform machining strategy

1 Introduction

Increasing the level of data exchange between CAx-systems remains a priority for the development of the intelligent manufacturing [1]. CAD-CAM systems supporting the ISO 14649-based STEP-NC data format to communicate with STEP-CNC (STEP-compliant CNC) provide a possibility of the feature oriented data representation and bidirectional dataflow [2]. Moreover, the integration of CAM functionality into a STEP-CNC system makes it possible to generate toolpaths online (inside the system) for different machining features, including freeform surfaces [3, 4].

The large amount of research on the STEP-NC-based manufacturing [5–7] opens up the potential for intelligent STEP-CNC systems to be introduced in the foreseeable future. These systems will be expected to perform tool selection, toolpath planning and generation, cutting and control operations autonomously, and human-machine interactions are to be minimized. STEP-CNC systems can be classified into three types: with conventional control, with new control and with new intelligent control [8]. The new intelligent control type of STEP-CNC implies that machining tasks intelligently and autonomously can be performed by a CNC based on the comprehensive information of ISO 14649 [3, 8]. Such systems must be able not only to interpret a STEP-NC program and execute motions (as in the first two types of STEP-CNC), but also comply with the entire set of criteria imposed on the intelligent STEP-CNC systems. These criteria are STEP-NC interpreter, STEP-NC executor; tool path generation; 3D STEP-NC viewer; e-manufacturing ability (enables a distributed and networked manufacturing process); interaction of STEP-NC info (the possibility to edit the machining parameters on a STEP-CNC controller);

✉ Wenlei Xiao
xiaowenlei@buaa.edu.cn

[1] School of Mechanical Engineering & Automation,
Beihang University, Beijing, 100191, China

[2] MIIT Key Laboratory of Aeronautics Intelligent
Manufacturing, Beihang University, 100191, Beijing, China

online closed-loop manufacture (collecting and processing of feedback information on a CNC controller); sensor-based manufacture (sensor feedbacks are supported); open-CNC architecture; and reconfigurable control network [6, 8].

In this paper, the possibility to use such intelligent manufacturing systems for freeform machining has been investigated. And, it was proposed to revise the choice of the optimal freeform surface representation which would be the most appropriate to use in STEP-NC-based manufacturing applications.

When designing freeform surfaces, contemporary CAD systems almost always produce models in which several trimmed surfaces are merged together. Commonly used NURBS representation has proven its reliability for both CAD and manufacturing applications [9, 10]. Many CAM systems are capable to perform toolpath generation for trimmed freeform surfaces using different techniques [11, 12]. However, presence of multiple trimmed NURBS patches makes it possible for cracks and ripples to appear along the edges of adjacent patches. Although in modern CAD systems the computational algorithms can deal with these cracks quite successfully, handling them in CAM environment sometimes may require human-machine interactions, at least for the verification and possible correction. Therefore, it may bring ambiguity to the online toolpath generation process in the systems with CAM functionalities transferred into a STEP-CNC. This can compromise the machining safety of the system.

Unlike the NURBS surfaces, T-splines allow to represent any complex model as a single gap-free surface which does not require trimming operations [13]. Moreover, T-splines have the advantages of fewer control points, gap-free model representation capability and localized refinement operations [13, 14]. At the same time, internally a T-spline surface can be viewed just as a collection of Bézier patches that are joined edge-to-edge in a non-rectangular array (although T-splines can be rational, it is possible to convert them into a set of Bézier patches, if it is necessary for some applications). Therefore, any computational algorithm that works on Bézier patches will work equally well on T-spline surfaces. In view of these, T-splines bring significant advantages to the online freeform toolpath generation, and can be considered as a reasonable choice for the development of the STEP-NC-based manufacturing of freeform surfaces [5, 15]. In this paper, T-spline surfaces are the surfaces that may have extraordinary points (T-NURCCs); whereas, simple T-spline surfaces refer to those without extraordinary points.

The first manufacturing results for simple T-spline surfaces were presented in Gan et al. [15]. The introduction of STEP-compliant data models for T-splines in our previous research [16] made possible their further definition in terms of STEP-NC and, consequently, integration within a STEP-CNC system [5]. Important issues of the efficient and robust toolpath generation for T-spline surfaces using one freeform machining strategy defined in ISO 14649 were investigated, and a prototype T-spline-enabled STEP-CNC system was introduced.

This paper emphasizes the importance of choosing the T-spline surface representation when implementing freeform machining operations within STEP-CNC systems. As the development of such systems with intelligent functionalities is very complicated, it was proposed to distinguish several stages of this process. In this research, we only focus on the implementation of the first stage, that is the development of a real T-spline-enabled STEP-CNC system which can strategically support online toolpath generation for freeform machining of simple T-spline surfaces using different freeform machining strategies. Although the specific approach to generating toolpaths in manufacture of freeform surfaces depends on the selected machining methodology, and there exist many different toolpath generation algorithms [17–19], only four freeform machining strategies defined in ISO 14649-11 have been considered. And, a practical STEP-CNC system with online toolpath generation capability for three-axis ball end machining has been built.

2 Preference for using T-splines in STEP-NC-based manufacturing

This section discusses the main arguments for using T-splines for the development of the STEP-NC-based manufacturing of freeform surface and for integration within STEP-CNC.

2.1 Definition of T-spline

T-splines were first introduced in Sederberg et al. [13, 14], and can be viewed as a modification of the NURBS representation with much more topological flexibility. In comparison to NURBS surfaces, T-splines have significant advantages that mainly are fewer control points, gap-free model representation capability, and localized refinement operations.

As a T-spline is an analytical parametric surface, its representation in three-dimensional space can always be calculated by means of the following point-based equation:

$$\mathbf{S}(s,t) = \frac{\sum_{i=1}^{n} P_i B_i(s,t) w_i}{\sum_{i=1}^{n} B_i(s,t) w_i} \qquad (1)$$

where P_i are the control points, w_i are weights, and $B_i(s,t)$ are the T-spline blending functions given by:

$$B_i(s,t) = N_i(s) \cdot N_i(t) \qquad (2)$$

In Eq. 2, $N_i(s)$ and $N_i(t)$ are B-spline basis functions associated with the individual knot vectors in s and t

directions correspondingly (recently cubic B-spline basis functions are commonly used). Unlike the NURBS, for T-splines the components of the knot vectors, s_i and t_i, for each basis function are deduced from the T-mesh, which is a grid that may have T-junctions [13].

In presence of extraordinary points, T-splines (more precisely, T-NURCCs) employ rules for subdivision surfaces that facilitates the representation of surfaces which are not parametrically rectangular.

2.2 T-spline and STEP-CNC

The development of sustainable STEP-CNC systems for commercial applications is still on a primitive level [20]. However, there is a possibility that the introduction of intelligent functionality to CNC using comprehensive STEP-NC data will gradually attract the attention of not only researchers but also manufacturers.

Implementation of the fully autonomous manufacturing approach to freeform machining using STEP-CNC is quite challenging. For example, as the toolpath computing tasks are carried out online on MMIs of CNC systems, the computing efficiency and robustness of toolpath generation algorithms should at first be considered. Thus, high reliability and flexibility requirements are imposed on the freeform surface representation used for machining. Though the NURBS surface representation has performed quite well in both design and manufacturing applications, we propose to consider T-splines as a better solution for the integration within STEP-CNC due to their advantages which will be discussed later in this section.

Because the development of a T-spline-enabled STEP-CNC system with intelligent functionality is very complicated, it is proposed to distinguish several stages of this process. Possible development stages can be determined as follows:

(1) Developing a real T-spline-enabled STEP-CNC system which can strategically support online toolpath generation for freeform machining of simple T-spline surfaces using different freeform strategies. At this stage, different freeform machining strategies have to be implemented and integrated within a STEP-CNC system, enabling the system to perform online toolpath generation. Obviously, the results of this stage are of limited value for real-life applications because only simple T-spline surfaces were considered, but these results can demonstrate the validity of the idea to use the STEP-CN representation of a T-spline model and manufacturing data in order to generate a toolpath inside the STEP-CNC followed by real cutting.

(2) Developing universal online toolpath generation algorithms that can handle both regular simple cases and the cases when T-spline surfaces have extraordinary points. This implies enabling the system to machine complex surfaces by converting them into a set of more simple regions followed by performing toolpath generation over these regions separately. Different regions can be processed using different machining strategies, and special techniques may be used to deal with the neighborhoods of extraordinary points.

(3) Developing algorithms and methods for the tool selection, tool orientation identification (for five-axis machining), cutting parameters selection, control operations, and status monitoring. All these algorithms have to be integrated within the STEP-CNC system enabling it to perform its functionality based on the information provided by a STEP-NC file and supplementary databases.

(4) Developing a T-spline enabled STEP-CNC system with smart features. This stage implies the implementation of principals of intelligent manufacturing as deep as possible. The implementation results of the previous stages have to be combined together to enable the system to perform all necessary work using its own capabilities, minimizing interactions with a human. Once the system receives the input STEP-NC file which contains the information about the T-spline model and manufacturing data, it can automatically perform feature recognition, detect extraordinary points, if needed, divide the whole complex surface into simple regions depending on the type and location of extraordinary points and surface topology, perform tool selection, toolpath planning and generation, cutting and control, as well as status monitoring and recovery. To improve the decision making abilities of such systems, some features of the artificial intelligence (AI)-based approach may be used. To elevate the performance, the system can be connected to a powerful data center and other systems to utilize their computing resources for computationally expensive operations.

Further in this paper, the implementation results of the first stage will be presented.

2.3 Advantages of the T-spline surface representation

From the point of view of the traditional manufacturing approach, the advantage of T-splines over NURBS mainly relates to the design stage. In terms of manufacturing, localized refinement operations (introduced with T-splines) can be useful for the efficient implementation of subdivision algorithms, in which a complex surface is subdivided into a set of feature-based regions.

However, advantages of the T-spline surface representation can be fully estimated only in the context of the intelligent manufacturing approach. That is, the manufacturer may only need to influence the design stage by defining the surface in terms of the manufacturing features and by specifying all required manufacturing parameters according to the STEP-NC standard. After this, the task of toolpath planning and generation is assigned to the intelligent STEP-CNC system which is able to perform some functions of the manufacturer. So, this approach can be viewed as a goal for future development rather then the current state in manufacturing of freeform surfaces.

Obviously, it is almost unrealistic to expect that all CAD/CAM systems will start using T-splines, but T-spline can be viewed as a superset for several other surface representations, including NURBS. Therefore, by implementing T-splines in STEP-CNC systems, we can solve the problem of freeform surface machining more globally (most probably, models which use other surface representations can be easily converted and represented using T-splines).

Thus, the advantages of using T-splines have been considered in the following aspects:

- convenience and safety
- implementation of the machining process

From the point of view of convenience and safety, T-splines were evaluated as a part of the STEP-NC communication interface between a CAD environment and STEP-CNC. With the increasing number of T-spline applications [21, 22], a question about the method to manufacture new created T-spline models will arise. A possible easy solution is to convert a T-spline surface into a set of NURBS patches and use existing CAM systems to generate toolpaths. But, in terms of the STEP-NC-based manufacturing, we want the STEP-CNC system to receive as complete information about the model as possible. Converting the surface into patches may alter the original explicit representation of some surface features (e.g., after the conversion, no extraordinary points exist in the definition of the surface anymore, yet some manufacturing requirements might be imposed over the neighborhood of extraordinary points).

Potentially, in the future some CAD systems (with STEP compliant CAM functionality) will be enabled to export a STEP-NC file which will contain full information about the model and all necessary manufacturing data. The STEP-NC file can then directly be transferred to the intelligent STEP-CNC system that will perform all the operations online in order to machine the surface. Thus, if the freeform surface is represented in the STEP-NC file by the unambiguously defined T-spline model, it gives more freedom to the STEP-CNC to use different approaches to the machining process. Moreover, it provides more safety as the STEP-NC file at some stage of the life cycle may be subjected to modifications using different software, and the trimmed surfaces may need the additional check for cracks before dispatching to a machine (such checks are unnecessary for gap-free T-spline surfaces).

When implementing the machining process, the T-spline representation ensures more flexibility. If a complex T-spline model has been converted into a set of NURBS patches, it is highly likely that configurations of these patches need to be considered during the toolpath generation. Hence, the methodologies which are used for machining somehow depend on the initial surface representation, and a more general representation of the surface gives more freedom in the design of machining processes. When the surface is represented in the STEP-NC file as a single gap-free T-spline model with extraordinary points, different feature-based regions on the surface can be derived inside the STEP-CNC system exclusively for the purpose of optimizing the machining process.

3 Toolpath generation for freeform machining

The first stage of the development of the intelligent STEP-NC-based manufacturing of freeform surfaces (see Section 2.2) implies that a STEP-CNC system must be enabled to support the online toolpath generation for simple T-spline surfaces using different machining strategies. To machine a freeform surface, many different strategies and methods can be used [9, 15, 18]. This paper focuses on the implementation of four freeform machining strategies defined in ISO 14649-11. These strategies may not be optimal from the point of view of approximation quality or minimizing the error between the tool and the design surface, but the purpose of the current development stage is to implement all the standard strategies. Later, new more advanced machining methodologies [17–19] can be added to the STEP-NC standard and implemented in the system.

3.1 Freeform machining strategies in ISO 14649-11

In ISO 14649-11, the freeform strategy defines the machining strategy used for milling operations. The definition of the milling strategy and the tolerances allow a STEP-CNC system to autonomously calculate the resulting toolpaths out of these values [23]. There are four freeform strategies presented in the standard: UV strategy, Plane Cutter Contact strategy, Plane Cutter Location strategy, and Leading Line strategy.

Under the UV strategy, toolpaths are determined to follow the parameter lines in the local (u, v) coordinate

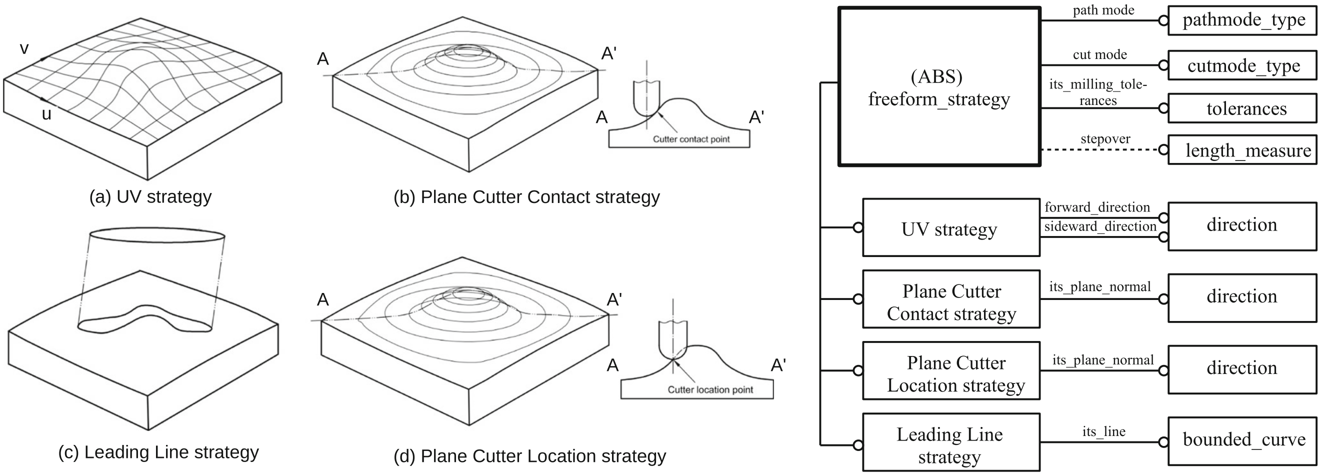


Fig. 1 Freeform machining strategies in ISO 14649-11

system (Fig. 1a). The forward direction (used in the first cut) and the sideward direction (in which the second cut is offset from the first) have to be specified in order to perform the toolpath generation process. According to the definition of the Plane Cutter Contact strategy, the cutter contact paths are generated by intersecting the target surface with a family of parallel planes, as shown in Fig. 1b. Unlike the previous strategy, the Plane Cutter Location strategy defines cutter location paths as a result of the intersection of the target surface, offset by the cutter radius, with planes (Fig. 1d). Both methods require the normal of the planes used for intersection with the target surface to be specified. According to the Leading Line strategy, toolpaths are obtained by projecting a curve on the workpiece surface along the z-axis of the local coordinate system (Fig. 1c). The curve is given as an attribute.

Because all of these strategies are subtypes of the freeform strategy, they have the following common attributes: the path mode type, cut mode type, tolerances, and stepover (the stepover can either be explicitly specified or be calculated out of the values of the tolerances). The path mode specifies the milling direction. This can be forward (or unidirectional) milling or zigzag (or bidirectional) milling. The cut mode is used to decide whether the climb or conventional milling mode should be used. The components of the tolerances are the chordal tolerance and the scallop height (Fig. 2). These values are necessary for the forward-step and side-step calculations which were discussed in detail in our previous research [5]. The stepover defines the distance between two neighboring toolpaths.

Toolpath generation processes under the definitions of all strategies are performed for three-axis ball end machining.

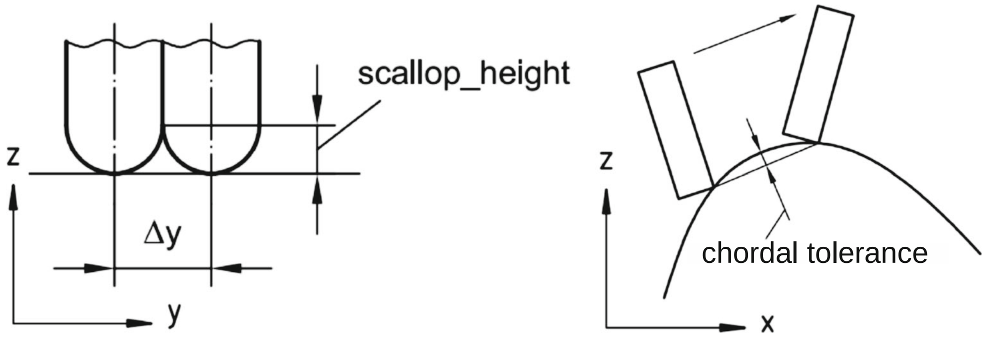
3.2 UV strategy

UV strategy defines the milling process as following iso-parametric lines of the parametric domain. A detailed survey on the toolpath generation in terms of UV strategy has been presented in our previous research [5]. The algorithm consists of the following main stages:

- Identification of rectangular T-regions.

  Because T-spline can be irregular in the parametric domain (i.e., T-image), it is often useful, in order to reduce computing efforts, to represent the T-image as a set of several regular, rectangularly shaped regions

Fig. 2 Definitions of the tolerances in ISO 14649 [23]

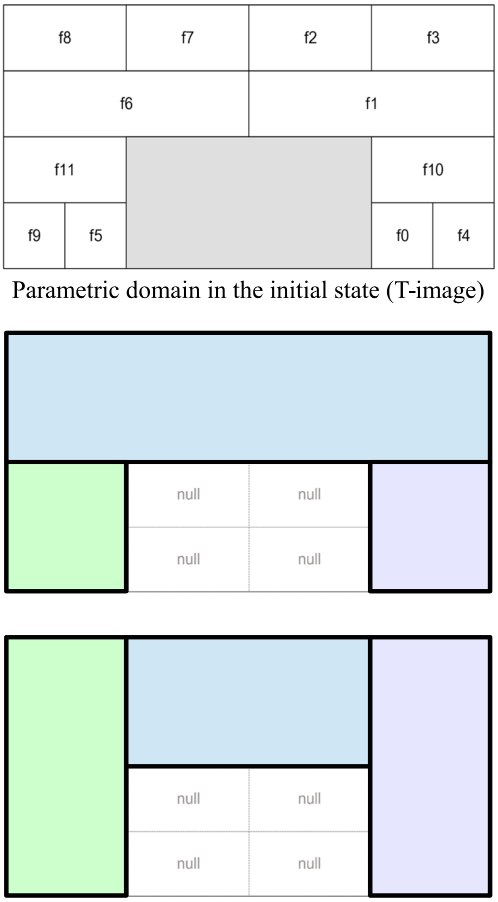


(called T-regions) [5]. T-region represents a collection of T-faces, and it can be viewed as a regular tensor product patch. Thus, the entire T-image is divided into a set of rectangular regions (as big as possible) using the special algorithm of T-regions generation [5]. Figure 3 shows the result of the T-regions generation for a given parametric domain.

- Toolpath generation for each T-region.

  All T-regions are parametrically rectangular, therefore toolpath generation is relatively simple and can be performed for each T-region independently.

This approach is not necessarily optimal for manufacturing, but in many cases it may significantly reduce the complexity of the problem which is often very important in terms of online toolpath generation. Therefore, it can be considered as a legitimate option. The resulting toolpaths (sequence of CC points) can be generated after specifying all the necessary parameters and performing forward-step and side-step calculations. Calculated toolpaths for the mouse model are shown in Fig. 4a. In order to perform real machining, the location of the cutting tool can be derived at each CC point through tool offsetting.



Parametric domain in the initial state (T-image)

Two possible variants of covering parametric domain by three T-regions

Fig. 3 The result of T-regions generation process

3.3 Plane Cutter Contact strategy

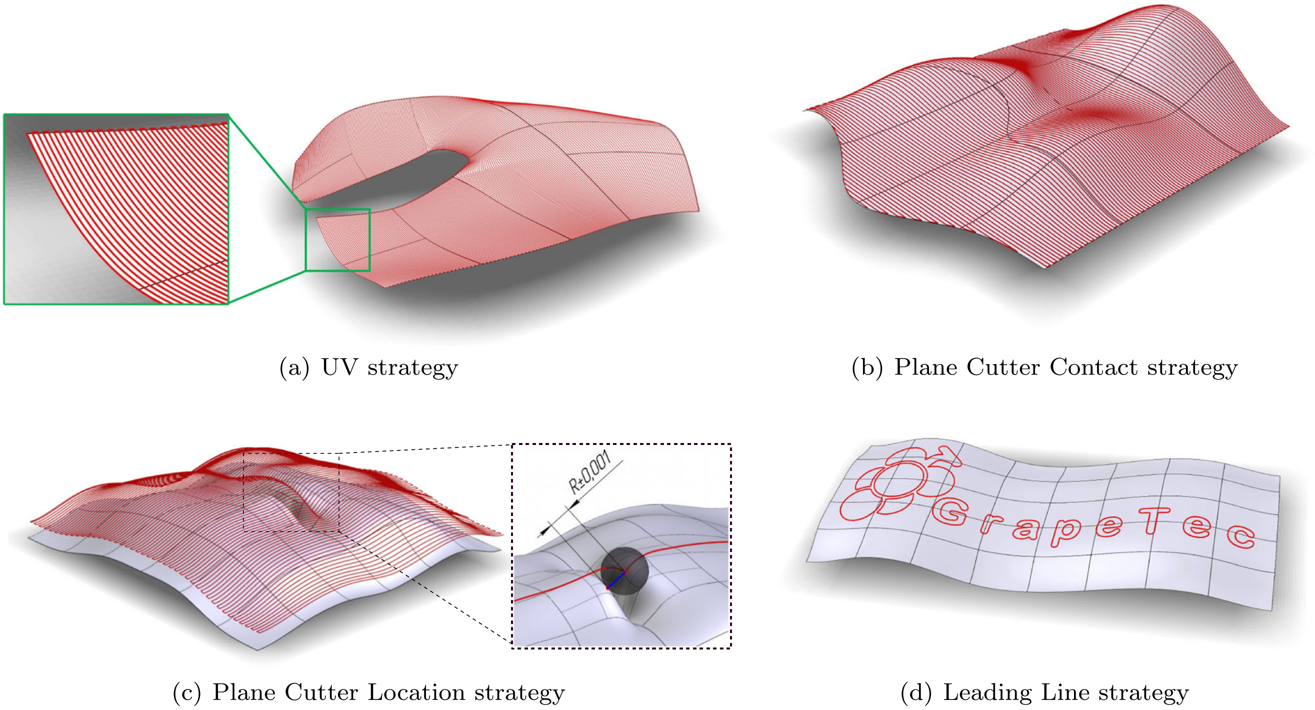
The cutter contact paths under the definition of Plane Cutter Contact strategy are generated by intersecting the target T-spline surface with parallel planes. Thus, the distance in which the next cutting plane is offset from the previous one (side-step) has to be determined in order to design the toolpaths. Moreover, the distance between two adjacent CC points (forward-step) must be calculated to ensure machining accuracy. Toolpath generation is carried out based on the parameters provided by STEP-NC code.

3.3.1 T-spline surface-plane intersection

Surface-surface intersection is a fundamental geometrical problem. Generally, there are many different approaches to solve this problem, e.g., by using analytic, lattice, marching, or subdivision methods [24]. It is often a good idea to formulate the general surface-surface intersection problem as a root-finding problem of an underconstrained piecewise polynomial system. Recently, a method to solve such systems with decomposition and a subdivision-based solver was presented in [25]. Bartoň et al. [26] proposed a method that can guarantee univariate solutions of underconstrained polynomial systems via no-loop and single-component tests. Both of these methods enable the efficient detection of the correct topology of the intersection curve, and can be implemented to solve the toolpath generation problem for T-spline surfaces in the future. In the context of Plane Cutter Contact strategy, the problem is formulated as T-spline surface-plane intersection problem. For each individual T-face (a face by face intersection approach has been used), resulting intersection curves will be topologically simple in the most cases. Moreover, as the toolpath generation problem is to be solved in terms of three-axis ball end milling, it is assumed that the complexity of the problem is not very high. Therefore, a more simple and easy to implement method can be used.

Thus, the solution for the rational polynomial parametric surface-implicit algebraic surface intersection problem can be used [27]. This problem is defined as:

$$S(s,t) = \left(\frac{X(s,t)}{W(s,t)}, \frac{Y(s,t)}{W(s,t)}, \frac{Z(s,t)}{W(s,t)} \right)^T \cap \ f(r) = 0, \quad 0 \le s, t \le 1 \tag{3}$$



(a) UV strategy

(b) Plane Cutter Contact strategy

(c) Plane Cutter Location strategy

(d) Leading Line strategy

Fig. 4 Results of toolpath calculations using the proposed algorithms for different simple T-spline models and strategies

where r denotes coordinates in Euclidean space. The implicit algebraic surface $f(x, y, z) = 0$ of total degree m can be represented as:

$$f(x, y, z) = \sum_{i=0}^{m}\sum_{j=0}^{m-i}\sum_{k=0}^{m-i-j} c_{ijk}x^i y^j z^k \tag{4}$$

By substituting $x = \frac{X(s,t)}{W(s,t)}$, $y = \frac{Y(s,t)}{W(s,t)}$, $z = \frac{Z(s,t)}{W(s,t)}$ into Eq. 4 and multiplying by $W^m(s, t)$ we can obtain an algebraic curve equation $F(s, t) = 0$.

Consider a plane equation in an implicit form:

$$ax + by + cz + d = 0 \tag{5}$$

Then, in terms of T-spline surface definition (1), the intersection curve equation can be represented as:

$$F(s, t) = \frac{\sum_{i=1}^{n} c_i^B B_i(s, t)}{\sum_{i=1}^{n} B_i(s, t) w_i} = 0, \tag{6}$$

where $c_i^B = (ax_i + by_i + cz_i + d)w_i$, and x_i, y_i, z_i and w_i are Euclidean coordinates and weights of the control points, respectively.

Once the intersection curve equation is obtained, the problem of intersection is reduced to the problem of the curve tracing (marching). A somewhat complex intersection curve, $F(s, t) = 0$, may involve various branches (from border to border), internal loops, and singular points [27], as shown in Fig. 5. Consequently, so-called characteristic points may occur: border points, turning points, or singular points correspondingly.

Efficient detection and calculation of turning and singular points is a quite extensive problem, and it can hardly be covered within the scope of this paper. Patrikalakis et al. [27] provides a detailed survey on calculation of the characteristic points. As an example, the calculation of a border point (a point of the intersection curve that lays on any of the borders of the (s, t) parametric domain) can be considered. It involves the solution of an univariate

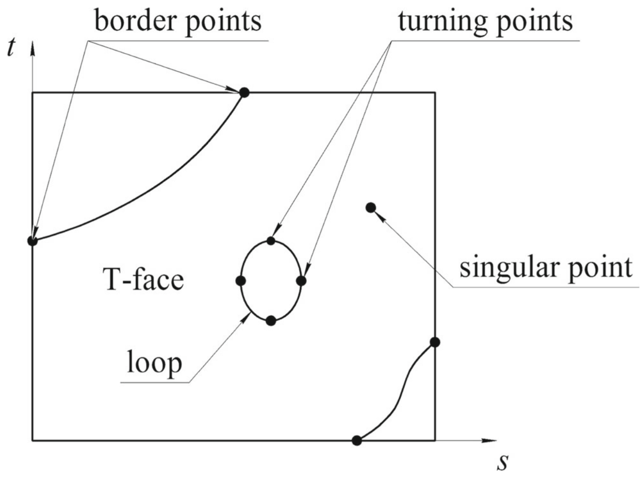


Fig. 5 Algebraic intersection curve in parametric space

polynomial equation. Thus, using Eq. 6, the equation for the border along $s = 0$ is:

$$F(0,t) = \frac{\sum_{i=1}^{n} c_i^B B_i(0,t)}{\sum_{i=1}^{n} B_i(0,t) w_i} = 0 \qquad (7)$$

Newton-Raphson method along with adaptive subdivision techniques is used to find all the local solutions on each border of the domain.

3.3.2 Domain decomposition

Having found all characteristic points of the intersection curve, the domain decomposition has to be performed for every face involved in the intersection [28]. The algorithm allows to trace the intersection curve in an efficient way without missing any important curve features. Assuming there are only two border points inside a domain (no loops inside), these points belong to the same component of the intersection curve. Moreover, there is exactly one component of the intersection curve inside this domain. Hence, the algorithm is designed to subdivide the original domain into smaller subdomains such that each subdomain contains exactly one curve component (Fig. 6).

Krishnan et al. [28] presented a detailed description of the domain decomposition process. In this paper, the main steps of the algorithm are briefly reviewed:

1. Given a set of points, S, on the intersection curve inside a domain. The cardinality of S is checked. If the cardinality is two, then there is a single curve component inside the domain, and the decomposition terminates.
2. If the cardinality is greater than two, the domain is subdivided along isoparametric lines determined by the parametric values of the points of S. Intersections of these isoparametric lines with the intersection curve define new points inside the domain. These points are further inserted into the appropriate subdomains.
3. If needed, the domain decomposition is applied recursively to each subdomain. The minimal size of subdomains can be limited by specifying a tolerance value.

3.3.3 Tracing of the intersection curve

Once domain decomposition is completed, the tracing process can be performed for each component of the intersection curve in order to generate all the points of this curve. Any of the two border points can be used as a starting point.

One possible solution is to trace the algebraic curve $F(s,t) = 0$ using differential curve properties [27]. Another approach is to march the intersection curve as a curve on the two intersecting surfaces [29]. Patrikalakis et al. [29] presented a robust marching method to trace an intersection curve of two parametric surfaces. The proposed method can be used to march a T-spline-plane intersection curve as a special case. In this paper, we present the basic equations used to calculate the intersection curve in terms of the proposed method.

A curve $s = s(u)$, $t = t(u)$ in the (s,t) plane defines a curve $c(u) = S(s(u), t(u))$ on a parametric surface $\mathbf{S}(s,t)$. The first derivative of the intersection curve, $c'(u)$, can be derived from a curve on the parametric surface using the chain rule:

$$c'(u) = S_s s' + S_t t', \qquad (8)$$

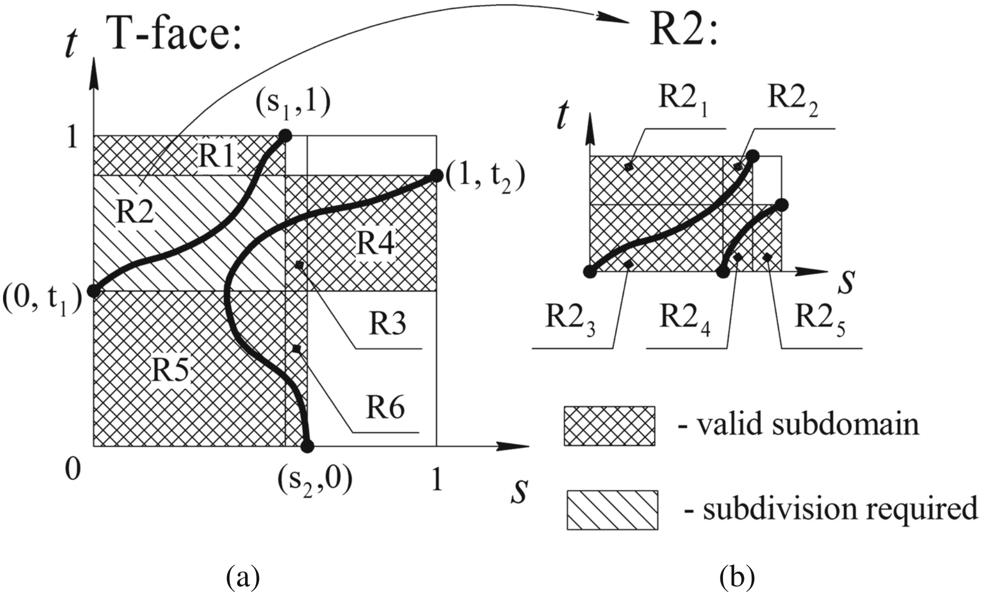


Fig. 6 T-face domain decomposition. **a** First level. **b** Second level

where $S_s = \frac{\partial \mathbf{S}}{\partial s}$ and $S_t = \frac{\partial \mathbf{S}}{\partial t}$. By taking the inner product on both sides of the Eq. 8 with S_s and S_t, the values of s' and t' can be evaluated as:

$$\begin{cases} s' = \frac{\det(c', S_t, \mathbf{N}(s,t))}{\mathbf{N}(s,t)\cdot\mathbf{N}(s,t)} \\ t' = \frac{\det(S_s, c', \mathbf{N}(s,t))}{\mathbf{N}(s,t)\cdot\mathbf{N}(s,t)}, \end{cases} \tag{9}$$

where det denotes the determinant and $\mathbf{N}(s,t) = S_s \times S_t$ is the normal vector of $\mathbf{S}(s,t)$. The Runge-Kutta method is employed to compute the points of the intersection curves by integrating the initial value problem for a system of nonlinear ordinary differential equations (9).

The marching direction when two surfaces intersect transversally can be obtained as follows:

$$c'(u) = \frac{\mathbf{N}(s,t) \times \mathbf{N_p}}{|\mathbf{N}(s,t) \times \mathbf{N_p}|}, \tag{10}$$

where $\mathbf{N_p}$ is a normal vector of the intersecting plane.

3.3.4 Toolpath generation results

The normal of the plane (its_plane_normal) is used to construct an oriented bounding box of the surface in order to generate the toolpaths. Hence, the location of the first cutting plane can be obtained. After performing the intersection of the surface with the first cutting plane, a side-step value is calculated, and the next cutting plane can be derived.

The algorithm has been tested on different T-spline models and has performed stable results for different sets of the input parameters. Toolpath generation results for zigzag milling of a simple T-spline surface are presented in Fig. 4b.

3.4 Plane Cutter Location strategy

The definition of the Plane Cutter Location strategy implies the need to obtain a surface offset from the target surface by the tool radius and further perform the intersection of this surface with a set of parallel planes.

An offset surface is a surface whose points are equidistant from the specified surface, and it can be used to generate gouge-free toolpaths [30]. Offset surfaces of rational polynomial parametric surfaces (including T-splines) cannot be represented exactly within the same class of functions describing the progenitor surface [31]. Therefore, if the offset surface has to be represented in the same form as the progenitor surface, approximation is required. Even if the progenitor is regular, its offset may have cusps and self-intersections. Thus, the loops that are resulted by self-intersections must be trimmed off.

Much research work has been done on the approximation of offset surfaces [32], and most of the well known algorithms could be applied for T-splines. However, in the presence of self-intersections, complex and time consuming calculations are required in order to detect and remove them. Therefore, it has been proposed to use an alternative and more simple method, which can be referred to as the direct positioning approach [33]. The implementation of the proposed algorithm requires approximating the T-spline surface using triangular mesh surface representation. Toolpath generation results obtained by using this method can guarantee good machining accuracy results when the sufficiently fine triangular mesh approximation of the surface is provided, and our developed T-spline kernel can generate triangular mesh approximations online for any given accuracy.

3.4.1 Alternative algorithm

According to the proposed algorithm, the location of a Cutter Location (CL) point is obtained by moving a sphere with the center in this point towards the target surface until it "touches" the surface, as shown in Fig. 7. As the result, the obtained CL point will be located above the target surface,

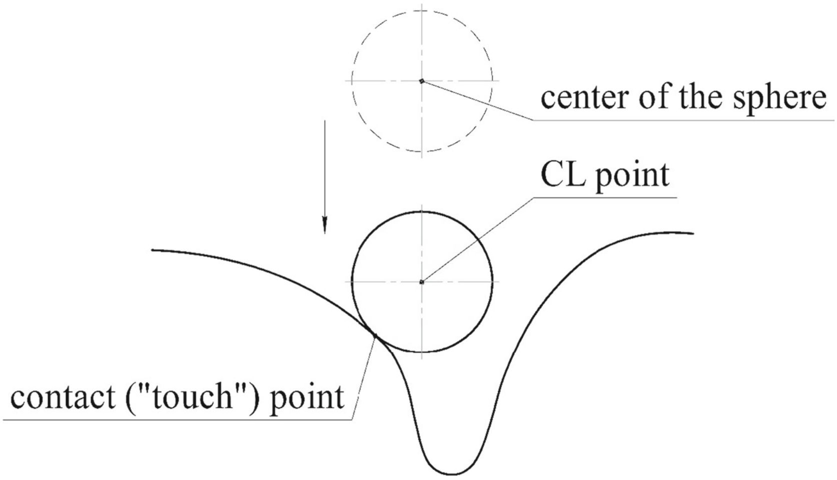


Fig. 7 The process of obtaining a CL point

offset by the radius of the sphere. By imposing constraints on the point movements (to move only along the line which is parallel to the z-axis and belongs to the intersecting plane), the algorithm can generate CL points which form the "intersection curve" of the offset surface with the plane (Fig. 8). The proposed algorithm can handle the cases when self-intersections of the offset surface may occur.

Once a T-spline surface is represented using the triangular mesh approximation, the AABB (axis-aligned bounding box) tree [34], a component of the very powerful library (CGAL) containing various geometric algorithms written in C++, can be used to perform efficient intersection and distance queries. The initial location of a CL point and the tool radius are given as an input for the algorithm.

To adjust the location and to define approximate boundaries in which the z value of the CL point may vary (the boundaries can be extended, but some initial guess is very useful), five ray-to-surface intersections are performed using AABB tree algorithms. The CL point with coordinates (x, y, z) and four points with coordinates $(x + r, y, z)$, $(x - r, y, z)$, $(x, y{+}r, z)$ and $(x, y{-}r, z)$ define the corresponding rays that are parallel to the z-axis and directed towards the surface. Thus, the corresponding intersection points on the surface provide a good initial guess about the boundaries (see Fig. 9).

To calculate the final location of the CL point, a black-box optimization problem has to be solved, and it is defined as:

$$|f(z) - r| = 0 \tag{11}$$

where z is the variable, $f(z)$ is a current distance from the CL point and the surface, and r is the tool radius.

The problem is solved using BOBYQA algorithm [35] realized in the NLopt library (an efficient library for non-linear local and global optimization written in C++) [36]. This algorithm is used to perform derivative-free bound-constrained optimization using an iteratively constructed quadratic approximation for the objective function. At every iteration, $f(z)$ value is calculated using AABB tree algorithm for the closest point and distance evaluation. The iteration process terminates when specified accuracy results are achieved (in our experiments, the tolerance value was set to 0.001). With this approach, a global solution for the given problem can be calculated very efficiently.

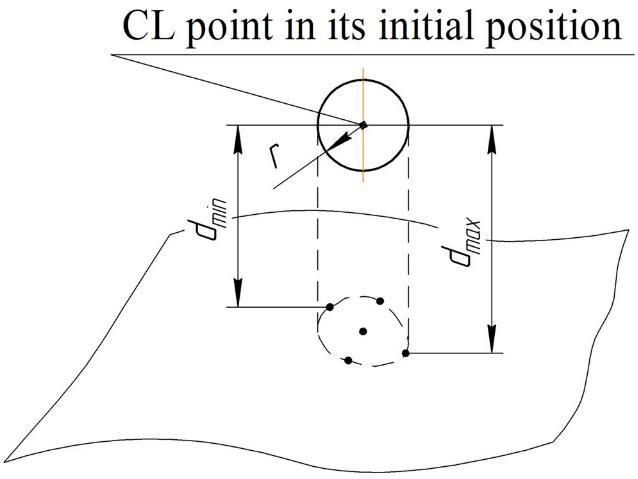


Fig. 9 Initial location of a CL point

As the result, after computing final locations of each CL point with a given accuracy, the set of all CL points may be viewed as the discrete (pixel) approximation of a virtual offset surface. Due to the discrete nature of this method, it is best suited for semi finishing operations (though it can also be successfully used for finishing operations). The algorithm demonstrates good performance results, which is very important in terms of the online toolpath generation that has to be performed inside a CNC system, using its CPUs. Moreover, concurrent computing techniques can be utilized to elevate the performance of the algorithm.

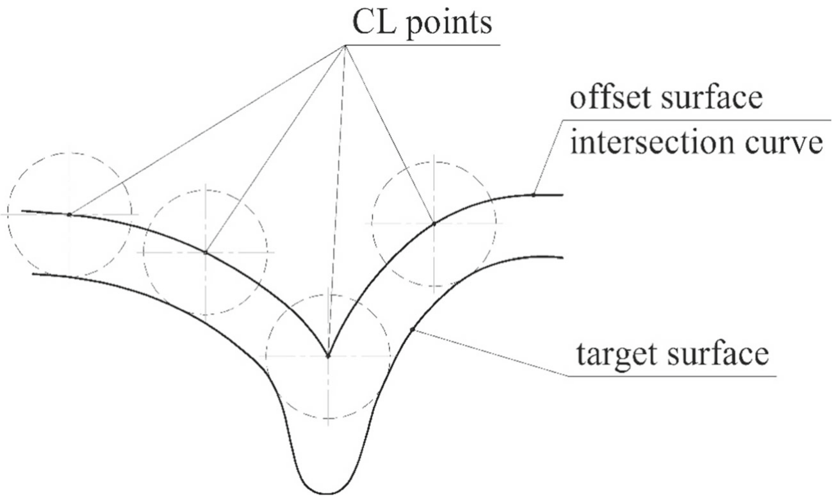


Fig. 8 Intersection curve, cross-section of the offset surface

If for some reason, one needs to return from the triangular mesh approximation to the original T-spline surface representation and to get parametric coordinates of the closest points (footpoints), the projection and distance detection problem has to be solved as described in Section 3.4.2. The initial value for the Newton iteration can be derived as follows: the calculated final location of a CL point defines the corresponding footpoint on the triangular mesh and the triangle to which this point belongs; provided that the parametric coordinates of all three vertices of the triangle are known, the approximate parametric coordinates of this footpoint can be obtain. It is assumed that the computation of projection points using the Newton-like method will provide global solutions, because the initial values are derived based on the results of the previous global optimization. Otherwise, some global solvers have to be used to find the closest point globally, such as in van Sosin et al. [25].

Thus, using the algorithm in Section 3.4.2, the footpoints on the T-spline surface for all CL points can be found, and the corresponding distances can be calculated. Usually, toolpath generation results obtained using a triangular mesh surface approximation are good enough, so there is no practical need to adjust the results using the original T-spline surface representation.

Figure 4c shows the toolpath generated by using the proposed algorithm on a simple T-spline model which has a groove in the middle (the offset magnitude here significantly exceeds the principal radii of curvature of the surface). The generated toolpath does not contain self-intersections, therefore gouge-free milling is guaranteed. The distances between every CL point and the surface are equal to $r \pm 0.001$ mm.

3.4.2 Projection and distance detection problem

This section discuses the problem of projecting a point in Euclidean space onto a T-spline surface in order to calculate a minimal distance between this point and the surface.

Piegl et al. [37] addresses this problem for NURBS curves and surfaces. The Newton iteration method is used to find a point on the surface so that the distance between point $\mathbf{P}$ and the surface is minimal, as shown in Fig. 10. A start value, (s_0, t_0), for Newton iteration is given as an input for this algorithm.

Form the vector function:

$$\mathbf{r}(s,t) = \mathbf{S}(s,t) - \mathbf{P} \tag{12}$$

and two scalar equations [37] (see Fig. 10):

$$\begin{aligned} f(s,t) &= \mathbf{r}(s,t) \cdot S_s(s,t) = 0 \\ g(s,t) &= \mathbf{r}(s,t) \cdot S_t(s,t) = 0 \end{aligned} \tag{13}$$

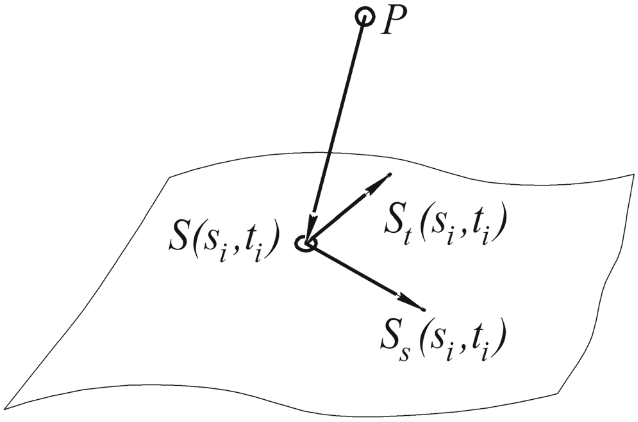


Fig. 10 Point projection

To solve Eq. 13 define:

$$\delta_i = \begin{bmatrix} \Delta s \\ \Delta t \end{bmatrix} = \begin{bmatrix} s_{i+1} - s_i \\ t_{i+1} - t_i \end{bmatrix}$$

$$J_i = \begin{bmatrix} f_s & f_t \\ g_s & g_t \end{bmatrix} = \begin{bmatrix} |S_s|^2 + \mathbf{r} \cdot S_{ss} & S_s \cdot S_t + \mathbf{r} \cdot S_{st} \\ S_s \cdot S_t + \mathbf{r} \cdot S_{ts} & |S_t|^2 + \mathbf{r} \cdot S_{tt} \end{bmatrix}$$

$$k_i = -\begin{bmatrix} f(s_i, v_i) \\ g(s_i, v_i) \end{bmatrix}$$

where $S_{ss} = \frac{\partial^2 \mathbf{S}}{\partial s^2}$, $S_{tt} = \frac{\partial^2 \mathbf{S}}{\partial t^2}$ and $S_{st} = S_{ts} = \frac{\partial^2 \mathbf{S}}{\partial s \partial t}$. All the functions in the matrix J_i are evaluated at (s_i, t_i). At the ith iteration, a (2×2) system of linear equations with the unknown δ_i must be solved. The system is given as:

$$J_i \delta_i = k_i \tag{14}$$

From δ_i we can derive:

$$\begin{aligned} s_{i+1} &= \Delta s + s_i \\ t_{i+1} &= \Delta t + t_i \end{aligned} \tag{15}$$

Zero tolerance (a zero cosine measure), ϵ_1, can be used to indicate convergence. Thus, convergence criteria is given as:

$$\begin{aligned} \frac{|S_s(s_i,t_i) \cdot (\mathbf{S}(s_i,t_i) - \mathbf{P})\,|}{|S_s(s_i,t_i)||\mathbf{S}(s_i,t_i) - \mathbf{P}|} &\le \epsilon_1 \\ \frac{|S_t(s_i,t_i) \cdot (\mathbf{S}(s_i,t_i) - \mathbf{P})\,|}{|S_t(s_i,t_i)||\mathbf{S}(s_i,t_i) - \mathbf{P}|} &\le \epsilon_1 \end{aligned} \tag{16}$$

If any of these conditions is not satisfied, a new value (s_{i+1}, t_{i+1}) has to be computed using Eq. 15. Then, the new value is verified to ensure that the parameters stay in range.

Additionally, the value can be checked, whether parameters do change significantly or not, using the following condition:

$$|(s_{i+1} - s_i)S_s(s_i,t_i) + (t_{i+1} - t_i)S_t(s_i,t_i)| \le \epsilon_2 \tag{17}$$

where ϵ_2 denotes a zero measure of Euclidean distance.

Iteration terminates if any of the conditions, Eq. 16 or Eq. 17, is satisfied.

3.5 Leading Line strategy

The definition of the Leading Line strategy requires the projection of a given curve (a sequence of discrete points) onto a T-spline surface along the z-axis of the local coordinate system in order to generate the toolpaths. The curve (leading line) is defined in the STEP-NC code by means of the its_line attribute.

To project a point of the given leading line onto the surface, let two non parallel vectors, $\overrightarrow{X}$ and $\overrightarrow{Y}$, be perpendicular to the z-axis. Then, for any given point, $\mathbf{P}$, one can find a point on the surface $\mathbf{S}(s_i, t_i)$ at which the vector $\overrightarrow{PS}$ is perpendicular to the vectors $\overrightarrow{X}$ and $\overrightarrow{Y}$ (if such a point exists) using Newton iteration method, as shown in Fig. 11. The start value, (s_0, t_0), for Newton iteration is the point on the surface closest to the line parallel to the z-axis and passing through the point $\mathbf{P}$.

Form the vector function

$$\mathbf{r}(s,t) = \mathbf{S}(s,t) - \mathbf{P} \tag{18}$$

and the two scalar equations:

$$\begin{aligned} f(s,t) &= \mathbf{r}(s,t) \cdot \overrightarrow{X} = 0 \\ g(s,t) &= \mathbf{r}(s,t) \cdot \overrightarrow{Y} = 0 \end{aligned} \tag{19}$$

The calculation sequence is the same as described in Section 3.4.2, except that the matrix, J_i, is defined as:

$$J_i = \begin{bmatrix} f_s & f_t \\ g_s & g_t \end{bmatrix} = \begin{bmatrix} \overrightarrow{X} \cdot S_s & \overrightarrow{X} \cdot S_t \\ \overrightarrow{Y} \cdot S_s & \overrightarrow{Y} \cdot S_t \end{bmatrix}$$

and, to indicate convergence, zero tolerances, zero measure of Euclidean distance, ϵ_1, and zero cosine measure, ϵ_2, have to be used. Thus, the first and second convergence criteria are given as:

$$|\mathbf{S}(s_i, t_i) - \mathbf{P}| \leq \epsilon_1, \tag{20}$$

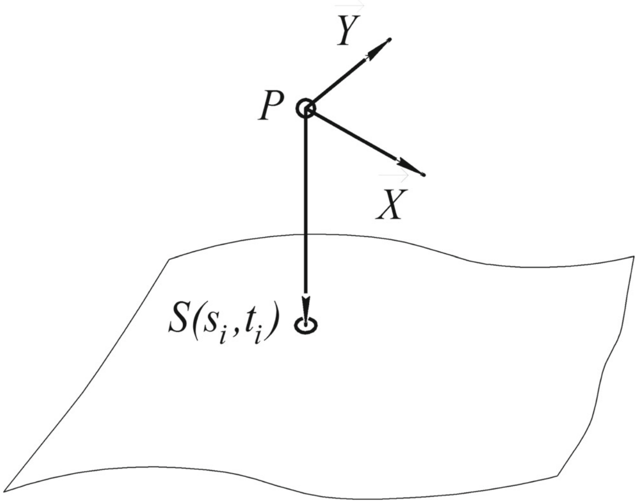


Fig. 11 Point projection along z-axis

and

$$\begin{aligned} &\frac{|\overrightarrow{X} \cdot (\mathbf{S}(s_i,t_i) - \mathbf{P})|}{|\overrightarrow{X}||\mathbf{S}(s_i,t_i) - \mathbf{P}|} \leq \epsilon_2 \\ &\frac{|\overrightarrow{Y} \cdot (\mathbf{S}(s_i,t_i) - \mathbf{P})|}{|\overrightarrow{Y}||\mathbf{S}(s_i,t_i) - \mathbf{P}|} \leq \epsilon_2 \end{aligned} \tag{21}$$

If at least one of the two conditions, Eq. 20 or Eq. 21, is satisfied, the iteration terminates, otherwise a new value (s_{i+1}, t_{i+1}) has to be computed.

The selected vectors $\overrightarrow{X}$ and $\overrightarrow{Y}$ have to ensure convergence of the method (their directions may be determined by projecting the vectors S_s and S_t on the x-y plane).

The toolpaths are generated by performing projection of each point of the given leading line, and the result is obtained as a sequence of CC points.

To show the flexibility of the used algorithm, Fig. 4d presents the result of projecting a set of curves onto the T-spline surface (curves form the emblem of our research group which is a bunch of grapes and the word "GrapeTec").

4 Development of the STEP-CNC system

This section presents the results of the development of a practical T-spline enabled STEP-CNC system with online toolpath generation capability. Also, issues of integrating the toolpath generation algorithms within the STEP-CNC system are discussed.

4.1 Representation of T-spline models and manufacturing data within STEP-NC

All necessary information for a STEP-CNC system that must be provided in order to machine a T-spline surface has to be represented in a STEP-NC file. This means that the T-spline model itself and all manufacturing data are to be defined by means of the STEP-NC code. The information within the STEP-NC code is represented as the object-oriented data, therefore various intelligent manufacturing functionalities can be implemented based on the provided abundant information.

The manufacturing feature is treated as a machining target of STEP-NC, therefore STEP-compliant T-spline data models have to be linked to the manufacturing feature. STEP-compliant data models for T-splines were introduced in our previous research [16]. In terms of STEP-CN, the entity t_spline_surface is inherited from the bounded surface, so that it can be contained by the manufacturing feature region_surface_list defined in the standard [23], as shown in Fig. 12.

To represent manufacturing data in terms of STEP-NC, the entity freeform_operation inherited from the abstract

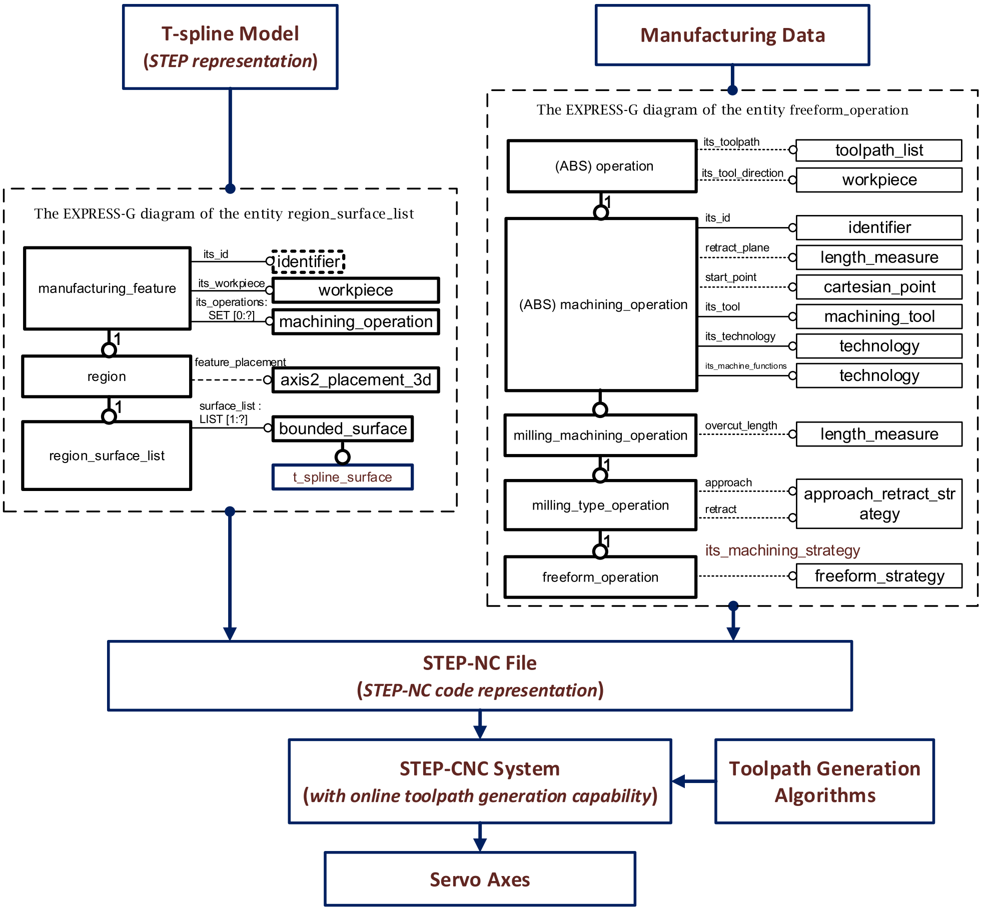


Fig. 12 General scheme of main components and data flows of the STEP-CNC-based manufacturing of simple T-spline models

supertype machining_operation has to be defined, as shown in Fig. 12. The definition implies that all attributes of the operation (toolpath, tool direction, start point, retract plane, cutting tool, machining technology, machine functions, approach, and retract strategies, etc.) have to be determined. Most of the attributes are optional, and their combinations depend on the selected operation setup. Thus, in the case of online toolpath generation, the attribute its_machining_strategy determines which of the four freeform strategies defined in the standard will be used in order to perform toolpath generation. In this case, some other attributes (e.g., the its_toolpath attribute that may contain a list of explicitly specified toolpaths) will be ignored.

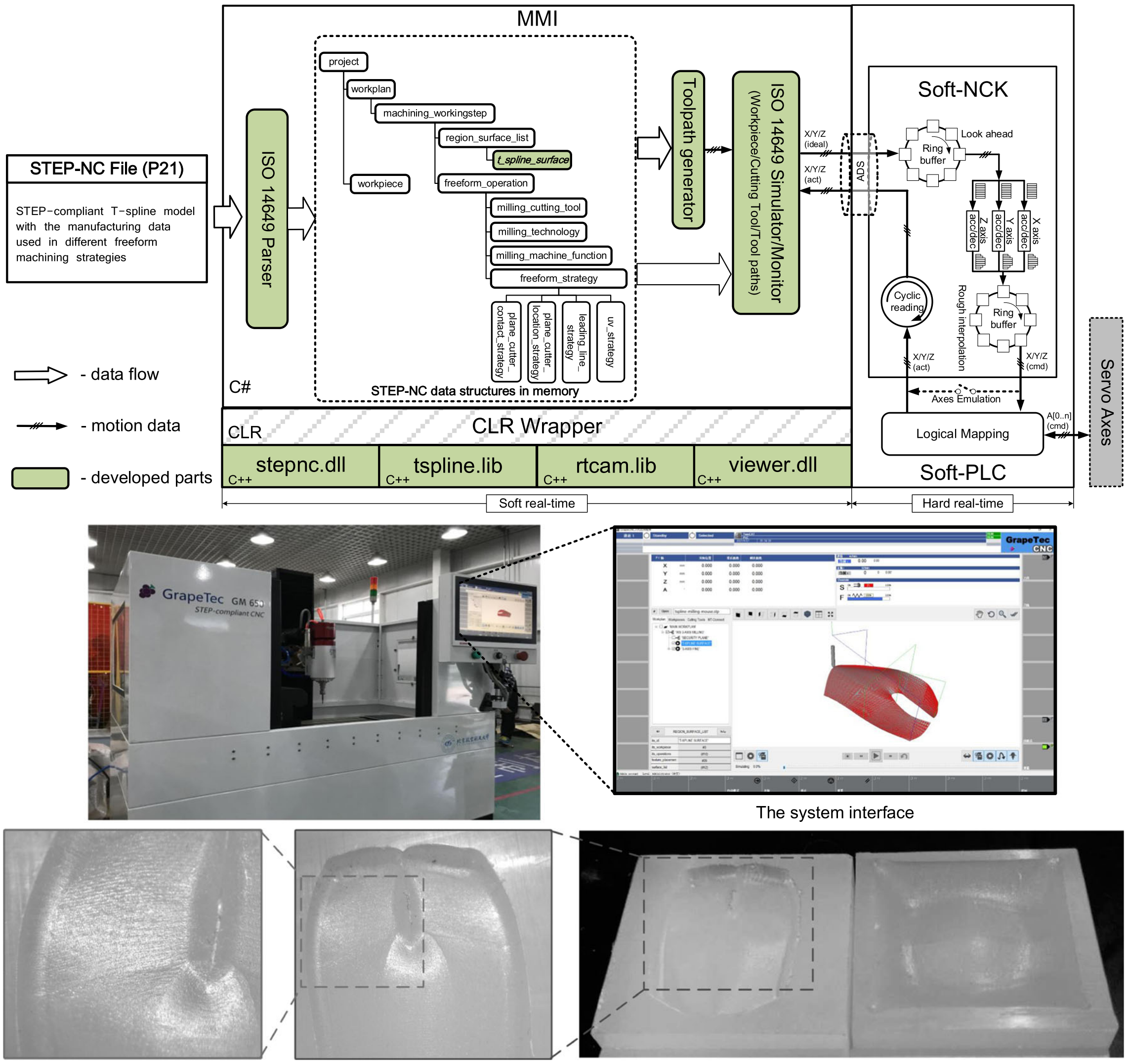
4.2 Development results

The introduction of the STEP-compliant CNC system supporting the surface direct interpolation technology for machining of simple T-spline surfaces can be considered as an important step towards the development of intelligent manufacturing, and it represents the implementation of the first stage of this process.

A prototype STEP-CNC system supporting the T-spline freeform machining functionality has been introduced in our previous research [5]. This paper presents the development results of the first practically built T-spline enabled STEP-compliant CNC system with online toolpath generation capability. The system is able to perform three-axis ball end machining of simple T-spline surfaces in terms of the four different strategies defined in the standard [23].

The system is established on an open-architecture CNC platform TwinCAT, promoted by BECKHOFF GmbH, Germany. It contains three main architectural components: MMI (Man Machine Interface), PLC (Programmable Logical Control), and NCK (Numerical Control Kernel), as shown in Fig. 13.

In order to integrate the T-spline within the STEP-CNC, most modifications have been carried out on the MMI. In particular, a STEP-NC parser, an online toolpath generator, and a STEP-NC simulator were embedded in the system. The data flow within MMI has the following order: the STEP-NC file (which contains the attribute of the freeform operation its_machining_strategy allowing to select one of the four freeform strategies) is parsed into memory and passed to the toolpath generator, then obtained



The system interface

Fig. 13 System architecture, data flows, general appearance of the developed STEP-CNC system and cutting results

toolpaths, after simulation and verification, are sent to the NCK via ADS (Automation Device Specification). The toolpath generator is responsible for performing the toolpath generation process according to the definitions of the selected strategy. The developed toolpath generation algorithms are integrated into to the system by means of the real-time CAM library rtcam.lib.

The second main architectural component PLC is developed according to the IEC 61131-3 standard using the soft-PLC technology [38]. The soft-PLC kernel provides the logical mapping functionality to Ether-CAT connected devices, such as servo axes, inputs and outputs, etc.

NCK contains the following functional components: a look ahead buffer which allows to reduce the real-time requirement on MMI, an accelerator/decelerator responsible for generation of the gently changed TCP speeds, and a rough interpolator providing a capability to smooth the axis motions.

5 Discussion and conclusion

This paper shows a real possibility of the successful development of a T-spline enabled STEP-CNC system capable of strategically supporting online toolpath generation for freeform machining. To comply with the requirements of the first stage of the intelligent manufacturing development process, such systems must support different freeform machining strategies (at least the four basic freeform machining strategies defined in ISO 14649-11). Therefore, the paper surveys algorithms and methodologies of toolpath generation in terms of all the four freeform strategies. Finally, a practical STEP-CNC system has been built, the toolpath generation algorithms have been implemented inside the system, and real cutting of simple T-spline surfaces has been performed.

Though the results presented in this paper are of limited value for real-life applications, because only simple T-spline surfaces have been considered, these results demonstrate the validity of the idea to use T-splines in the STEP-NC-based manufacturing of freeform surfaces. Thus, the focus of further research has to be the implementation of the next development stages of a T-spline-enabled STEP-CNC system with intelligent functionalities (as discussed in Section 2.2).

Acknowledgements The authors appreciate BECKHOFF China for their kind support of the open-architecture TwinCAT-CNC platform. Special Program of Ministry of Industry and Information Technology, China.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Xu XW, Wang H, Mao J, Newman ST, Kramer TR, Proctor FM et al (2005) STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration. Int J Prod Res 43(17):3703–3743
2. Newman ST, Allen RD, Rosso RSU Jr (2003) CAD/CAM solutions for STEP-compliant CNC manufacture. Int J Comput Integr Manuf 16(7–8):590–597
3. Suh SH, Lee BE, Chung DH, Cheon SU (2003) Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. Comput Aided Des 35(12):1069–1083
4. Nittler KM, Farouki RT (2016) A real-time surface interpolator methodology for precision CNC machining of swept surfaces. Int J Adv Manuf Technol 83(1):561–574
5. Zhao G, Liu YZ, Xiao WL, Zavalnyi O, Zheng LY (2018) STEP-compliant CNC with T-spline enabled toolpath generation capability. Int J Adv Manuf Technol 94(5–8):1799–1810. https://doi.org/10.1007/s00170-017-0253-x
6. Xiao WL, Lianyu Z, Huan J, Pei L (2015) A complete CAD/CAM/CNC solution for step-compliant manufacturing. Robot Comput Integr Manuf 31:1–10
7. Hu P, Han Z, Fu H, Han D (2016) Architecture and implementation of closed-loop machining system based on open step-nc controller. Int J Adv Manuf Technol 83(5):1361–1375
8. Suh SH, Kang SK, Chung DH, Stroud I (2008) Theory and design of CNC systems, 1st edn. Springer, Berlin. ISBN 1848822111, 9781848822115
9. Lasemi A, Xue D, Gu P (2010) Recent development in CNC machining of freeform surfaces: a state-of-the-art review. Comput Aided Des 42(7):641–654
10. Zhou K, Wang GJ, Jin HZ, Tan ZY (2008) NURBS interpolation based on exponential smoothing forecasting. Int J Adv Manuf Technol 39(11):1190–1196
11. Yang DCH, Chuang JJ, OuLee TH (2003) Boundary-conformed toolpath generation for trimmed free-form surfaces. Comput Aided Des 35(2):127–139
12. Li CL (2007) A geometric approach to boundary-conformed toolpath generation. Comput Aided Des 39(11):941–952
13. Sederberg TW, Zheng J, Bakenov A, Nasri A (2003) T-splines and T-NURCCs. ACM Trans Graph 22(3):477–484
14. Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T (2004) T-spline simplification and local refinement. ACM Trans Graph 23(3):276–283
15. Gan WF, Fu JZ, Shen HY, Chen ZY, Lin ZW (2014) Five-axis tool path generation in CNC machining of T-spline surfaces. Comput Aided Des 52:51–63
16. Xiao WL, Liu YZ, Li R, Wang W, Zheng JM, Zhao G (2016) Reconsideration of T-spline data models and their exchanges using STEP. Comput Aided Des 79:36–47
17. Wang X, Li Y, Chosh S, Wu X (1993) Curvature catering—a new approach in manufacture of sculptured surfaces (part 2. Methodology). J Mater Process Technol 38(1):177–193
18. Kim YJ, Elber G, Barton M, Pottmann H (2015) Precise gouging-free tool orientations for 5-axis cnc machining. Comput Aided Des 58:220–229. Solid and Physical Modeling 2014
19. Bo P, Bartoň M, Plakhotnik D, Pottmann H (2016) Towards efficient 5-axis flank cnc machining of free-form surfaces via fitting envelopes of surfaces of revolution. Comput Aided Des 79:1–11
20. Latif K, Yusof Y (2016) New Method for the development of sustainable STEP-Compliant Open CNC System. Procedia CIRP 40:230–235
21. Sederberg MT, Sederberg TW (2010) T-splines: a technology for marine design with minimal control points. In: 2nd Chesapeake power boat symposium 2010
22. Bazilevs Y, Calo VM, Cottrell J, Evans J, Hughes T, Lipton S et al (2010) Isogeometric analysis using t-splines. Comput Methods Appl Mech Eng 199(5):229–263. Computational Geometry and Analysis
23. ISO 14649-11 (2004) Industrial automation systems and integration—physical device control - data model for computerized numerical controllers—part 11: process data for milling
24. Patrikalakis NM, Prakash PV (1990) Surface intersections for geometric modeling. ASME J Mech Des 112(1):100–107
25. van Sosin B, Elber G (2017) Solving piecewise polynomial constraint systems with decomposition and a subdivision-based solver. Comput Aided Des 90:37–47. https://doi.org/10.1016/j.cad.2017.05.023; sI:SPM2017
26. Bartoň M, Elber G, Hanniel I (2011) Topologically guaranteed univariate solutions of underconstrained polynomial systems via no-loop and single-component tests. Comput Aided Des 43(8):1035–1044. https://doi.org/10.1016/j.cad.2011.03.009
27. Patrikalakis NM, Maekawa T (2009) Shape interrogation for computer aided design and manufacturing, 1st edn. Springer, Berlin. ISBN 364204073X, 9783642040733
28. Krishnan S, Manocha D (1997) An efficient surface intersection algorithm based on lower-dimensional formulation. ACM Trans Graph 16(1):74–106
29. Patrikalakis NM, Maekawa T, Ko KH, Mukundan H (2004) Surface to surface intersections. Comput Aided Des Appl 1(1–4):449–458
30. Tang K, Cheng CC, Dayan Y (1995) Offsetting surface boundaries and 3-axis gouge-free surface machining. Comput Aided Des 27(12):915–927
31. Patrikalakis NM, Prakash PV (1988) Free-form plate modeling using offset surfaces. ASME J Offshore Mech Arctic Eng 110(3):287–294
32. Chen YJ, Ravani B (1987) Offset surface generation and contouring in computer-aided design. ASME J Mech Transm Autom Des 109(1):133–142

33. Choi BK, Kim DH, Jerard RB (1997) C-space approach to tool-path generation for die and mould machining. Comput Aided Des 29(9):657–669
34. Alliez P, Tayeb S, Wormser C (2018) 3D fast intersection and distance computation. In: CGAL user and reference manual, 4.13 edn. CGAL Editorial Board
35. Powell MJD (2009) The BOBYQA algorithm for bound constrained optimization without derivatives. Tech. Rep. NA2009/06; Department of Applied Mathematics and Theoretical Physics; Cambridge
36. Johnson SG (2018) The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt
37. Piegl L, Tiller W (1997) The NURBS Boo, 2nd edn. Springer, New York. ISBN 3-540-61545-8
38. Huan J, Jing Y, Xiao WL (2011) Industrial control programming language based on IEC61131-3 CNC system software design, 1st edn. Beijing University of Aeronautics and Astronautics Press, Beijing. ISBN 7512404859, 9787512404854