



Title:

Integration of T-spline within STEP-compliant CNC

Authors:

Gang Zhao, zhaog@buaa.edu.cn, Beihang University
 Oleksandr Zavalnyi, sanyzav1992@gmail.com, Beihang University
 Yazui Liu, liuyazui5684845@163.com, Beihang University
 Wenlei Xiao*, xiaowenlei@buaa.edu.cn, Beihang University

Keywords:

T-spline, STEP-NC, STEP-CNC, Freeform surface machining

DOI: 10.14733/cadconfP.2017.37-43

Introduction:

Advanced freeform surface machining usually refers to the recently most used surface representations such as NURBS and T-spline. Their definitions in STEP/STEP-NC provide the possibility to integrate them with the STEP-compliant CNC (STEP-CNC) systems. Due to the advantages of less control points, gap-free model representation capability, and localized refinement operations against NURBS, T-spline models became more and more attractive for manufacturing in the research field. The fact that NURBS surfaces can be converted into T-spline models allows to consider T-spline as an optimal solution for the integration of advanced freeform surface machining technologies within STEP-CNC.

This paper discusses the representation of T-spline within STEP-NC and issues of toolpath generation on T-spline surfaces in context of the machining strategies defined in ISO 14649-11 [2]. Then some results of toolpath generation process are presented. At the end, we show the possibility of integration of T-spline with a prototype STEP-CNC system that provides on-line toolpath generation capability.

Main idea:

Because of their advantages over NURBS [7], T-spline surfaces gained considerable attention of researchers. The capability to represent any complex shape as a one surface, which in turn means no trimming and jointing operations are to be performed, ensures stability of T-spline models in context of manufacturing operations. After STEP-compliant data models for T-spline were proposed in [9], its representation in STEP-NC code should be a logical continuation of the process intending the full integration of T-spline within STEP-CNC systems, wherein the on-line toolpath generation algorithms must be performed.

T-spline within STEP-NC

STEP-compliant CNC can implement much intelligent manufacturing functionality using the abundant and object-oriented manufacturing data carried by the STEP-NC code. The new control type of STEP-CNC systems [5], which demand the on-line toolpath generation capability, has recently encountered a significant challenge in freeform machining. The toolpath generation requires some part of the CAM kernel transferred into a STEP-CNC system. ISO 10303-238 [3] and ISO 14649 provide a new data exchange interface between CAD/CAM and CNC systems. A STEP-compliant CNC system must support this new interface, and be able to exchange all the manufacturing data in STEP-NC format [10]. NURBS is recently the only method used in the STEP/STEP-NC standards to represent freeform surfaces, hence seems to be the only object that can be considered to generate the freeform toolpaths on-line.

However, the real-time and error free performances are hard to be guaranteed, as the trimmed and jointed NURBS surfaces can cost heavy computation load and bring risks of danger. Using the advantages of T-spline, that are single surface representation and free of trimming, the efficiency and stability problems of on-line freeform toolpath generation can be effectively alleviated. In attempt to implement this new idea, the STEP models of T-spline have been proposed in our previous research [9] and an open-source kernel T-SPLINE [8] has been developed accordingly.

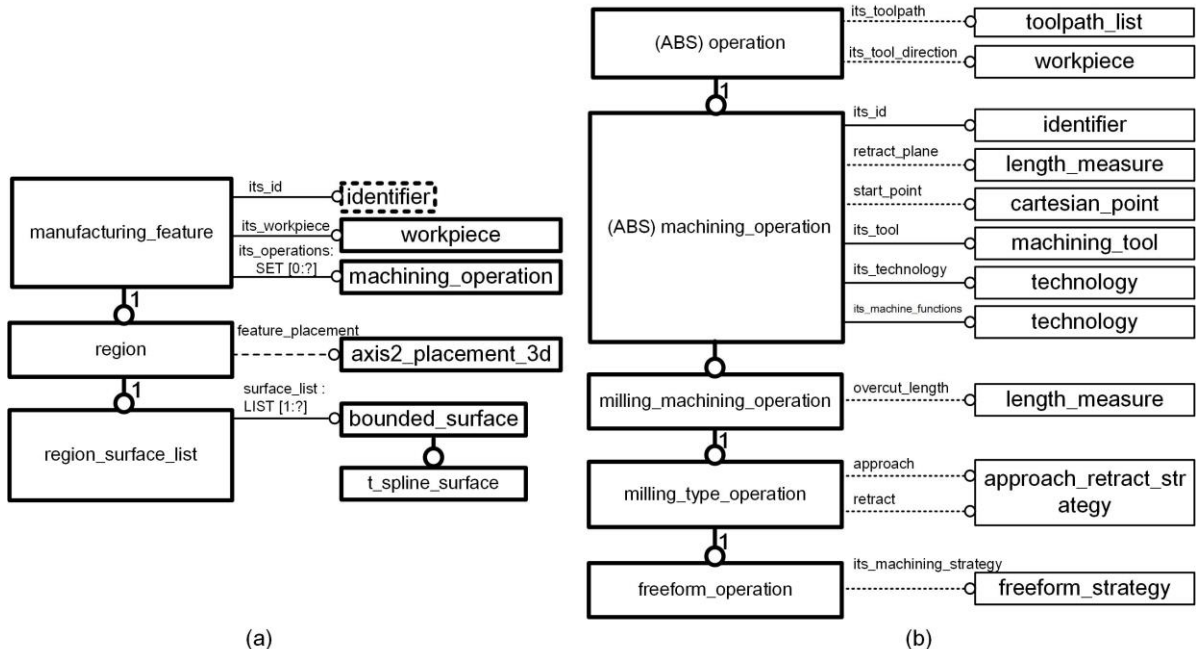


Fig. 1: The EXPRESS-G diagrams of: (a) the entity region_surface_list, (b) the entity freeform_operation.

The manufacturing feature is treated as machining target of STEP-NC, so STEP-compliant T-spline data models have to be linked to the manufacturing feature. Since the entity `t_spline_surface` inherits from `bounded_surface`, it can be contained by manufacturing feature `region_surface_list` defined in ISO 14649-11 (Fig. 1(a)). The freeform operation, is derived from the milling type operation, and is the class of operations for freeform milling [2]. As shown in Fig. 1(b), the entity `freeform_operation` is constituted by attributes of toolpath (optional), tool direction (optional), identifier, retract plane (optional), start point (optional), cutting tool, machining technology, machine functions, overcut length (optional), approach and retract strategies(optional), and machining strategy (optional), which allows the STEP-CNC to perform on-line toolpath generation capability .

Toolpath generation in context of freeform machining strategies

STEP-NC standard (ISO 14649-11) provides definition of machining strategies, and for freeform machining there are four strategies: `uv_strategy` (UV strategy), `plane_cc_strategy` (Plane cutter contact strategy), `plane_cl_strategy` (Plane cutter location strategy) and `leading_line_strategy` (Leading line strategy). Each of the four freeform strategies has the following common attributes: path mode, cut mode, tolerances and stepover (when the explicit definition is needed). The path mode defines whether forward (or unidirectional) or zigzag (or bidirectional) toolpaths will be generated. The cut mode is used to decide whether the climb or conventional milling mode should be used. The distance between two neighboring toolpaths is defined as the stepover. The value can be obtained either by specifying explicitly or calculated in terms of the tolerances, which consist of the chordal tolerance and the scallop height [2].

The four strategies presented in the standard can be classified in terms of the main working domain that is used during the toolpath design process. For the `uv_strategy` the main working domain

is a parametric coordinate system (parametric domain), whereas for the remaining three strategies it is Cartesian space. Thus, *uv_strategy* requires two directions, the forward one (used in the first cut) and the sideward one (in which the second cut is offset from the first), to be specified in order to generate a sequence of cutting points on the surface (Fig. 2(a)). For the *plane_cc_strategy*, *plane_cl_strategy* and *leading_line_strategy*, toolpaths (either the cutter contact paths or cutter location paths) can be obtained out of the interaction of geometrical objects (parallel planes or bounded curves) with the target surface or its offset in 3D space. Detailed treatment of algorithms allowing to calculate the toolpaths for freeform machining can be found in [4],[6]. In this paper we focus on the main implementation concepts in terms of the *uv_strategy* and *plane_cc_strategy* in order to show the flexibility and reliability of the developed T-spline CAM library to handle both approaches to the toolpath design process.

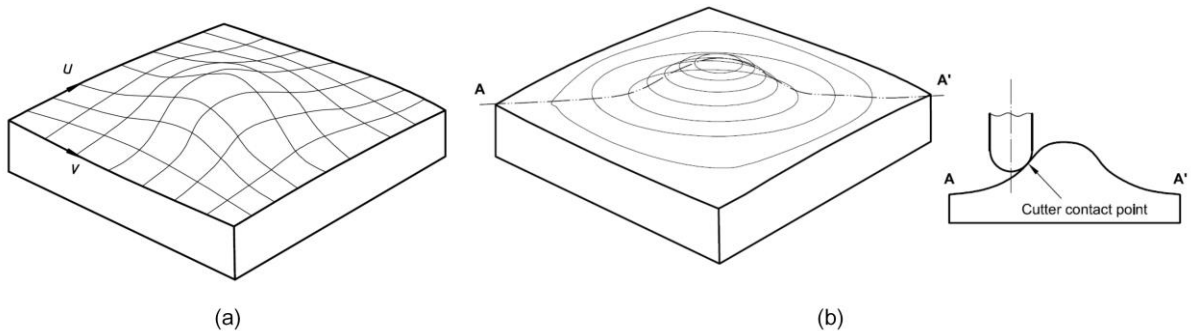


Fig. 2: Definition of the: (a) UV strategy in ISO 14649, (b) Plane cutter contact strategy in ISO 14649.

Toolpath generation under the UV strategy

The UV strategy defines the milling process to follow the lines in the parametric coordinate system. Usually, the T-spline has an irregular parametric domain (i.e. T-image). However, very often it is useful, in order to reduce computing efforts, to represent the T-image as a set of several regular, rectangularly shaped regions (called T-regions). Firstly, derive the rectangular regions from the T-image as big as possible. Then, the toolpath generation process for each T-region has to be performed independently. As each T-region is a rectangle, this process is relatively much simpler. In this paper we present a simple yet robust and effective algorithm to generate the T-regions:

- Convert the T-image with a T-cell matrix. Parametric coordinate (s_i, t_j) of each T-vertex can be used to generate a parametric grid. Horizontal and vertical lines of this grid split the whole parametric domain (including parts where the surface is not defined) into several rectangular cells (called T-cells). The T-cells can be stored in a matrix, so called T-cell matrix. Each T-cell has a pointer to the T-face it covered. When no T-face is covered, the pointer is null.
- Generate T-region strips for each row/column. Depending on the used forward direction in the entity *uv_strategy* (Fig. 2) the row or column T-region strips could be generated. When the forward direction is on the S direction, i.e. equals to $(1.0, 0.0)$, the row T-region strips should be created, and the strips are split by the T-cells with null T-face pointers. The column T-region strips could be created in the other case.
- Merge the possible T-regions. After all the T-region strips are generated as shown in Fig. 3(d), they can be merged together in order to get as less as possible T-regions. The merging process is implemented by comparing the (s_{\min}, s_{\max}) or (t_{\min}, t_{\max}) (determined by forward direction) of the T-regions. Each T-region now covers a sub T-cell matrix.

Fig. 3 gives an example of the T-region generation (the forward direction is S direction), while the T-region generation process when the forward direction is T direction is quite similar. Using the proposed algorithm, allows to avoid the commonly used scanning process, hence higher efficiency could be obtained. Additionally, the algorithm is able to handle both gaps on edges and holes inside.

The identified T-regions could further facilitate the computation of cutting points on the next stage. For each target parametric coordinate (s, t) , the T-cell it located is firstly derived from the sub T-

cell matrix it covers. In order to calculate any cutting point we need to know where the parametric coordinate is located (which T-face it belongs to). As each T-cell contains a T-face pointer, the computation of Cartesian points and normal vectors can be conducted directly using the analytic forms of basis functions [9].

Each T-region covers a rectangle on the T-image, thus much fewer computing efforts are required to generate the toolpaths. Nonetheless, to ensure the expected efficiency the algorithms still have to be accurately designed. In terms of steps calculation, two approaches to create the toolpaths are considered in ISO 14649: tolerances and step over guided calculation.

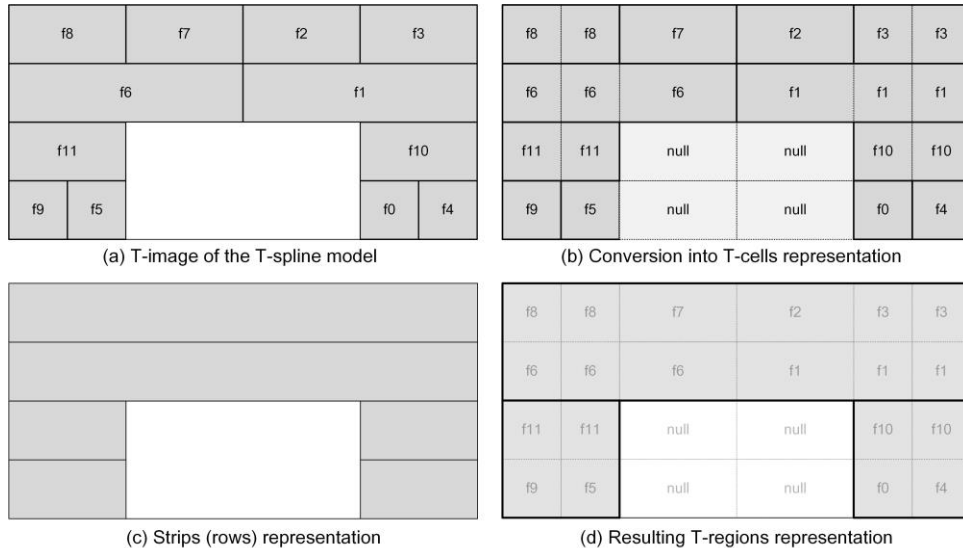


Fig. 3: The process of T-regions generation (forward direction = [1.0, 0.0]).

For the purpose of machining, the toolpath is approximated by a series of line segments. They are defined by the forward-step referring to the distance between two adjacent cutter contact points (CC points). The side-step refers to the distance between two adjacent cutting lines. The values of both of these step types have to be calculated considering the milling errors, the chordal and scallop errors correspondingly. Depending on the parameters given in the freeform strategy, the STEP-NC code can determine which approach needs to be implemented in creating the toolpaths: when the step over is given, the tolerances will be ignored [2]. In this paper we do not present the calculations of the forward-step the side-step because of space limitations, but the references to the used algorithms can be found in [6].

Toolpath generation under the Plane cutter contact strategy

According to the definition of the plane_cc_strategy, the cutter contact paths are generated by intersecting the target surface with parallel planes, as shown in Fig. 2(b) [2]. Surface/surface intersection is a fundamental geometrical problem. There are exist many methods to solve this problem, such as lattice method, subdivision method, marching method [4]. T-spline surface to plane intersection can be viewed as a rational polynomial parametric surface to implicit algebraic surface intersection problem defined as [4]:

$$r = r(u, v) = \left(\frac{X(u, v)}{W(u, v)}, \frac{Y(u, v)}{W(u, v)}, \frac{Z(u, v)}{W(u, v)} \right)^T \cap f(r) = 0, \quad 0 \leq u, v \leq 1. \quad (5.1)$$

To obtain an implicit algebraic curve in (u, v) , we can substitute surface patch equation $r(u, v)$ into plane equation $f(r)$. In terms of Bernstein polynomials this algebraic curve can be represented as:

$$F(u, v) = \sum_{i=0}^M \sum_{j=0}^N c_{ij}^B B_{i,M}(u) B_{j,N}(v) = 0, \quad (u, v) \in [1, 0]^2, \quad (5.2)$$

where $B_{i,M}(u)$ and $B_{j,N}(v)$ are B-spline basis functions. For a plane in an implicit form $ax + by + cz + d = 0$, coefficients c_{ij}^B are:

$$c_{ij}^B = (ax_{ij} + by_{ij} + cz_{ij} + d)w_{ij}, \quad (5.3)$$

where w_{ij} are weights. In order to get all points of the intersection curve, the marching method have to be applied, which requires starting points for every branch of the solution. Starting points could be border points, turning points and singular points. Detailed description of algorithms for calculating these points can be found in [4]. After all the starting points are obtained, we are able to march the curve using differential curve properties (the Runge-Kutta or other method with adaptive step size could be applied).

The calculated intersection curves represent the cutter contact path, and can be further used to generate toolpath on the T-spline surface in efficient way.

Some results of toolpath generation

Implementation of different strategies within STEP-NC definitions, enables a STEP-CNC system to perform on-line toolpath generation process. The choice of the strategy to be applied in each specific case is determined mainly by the surface topology and tool geometry. Some results of toolpath generation are shown in Fig. 4.

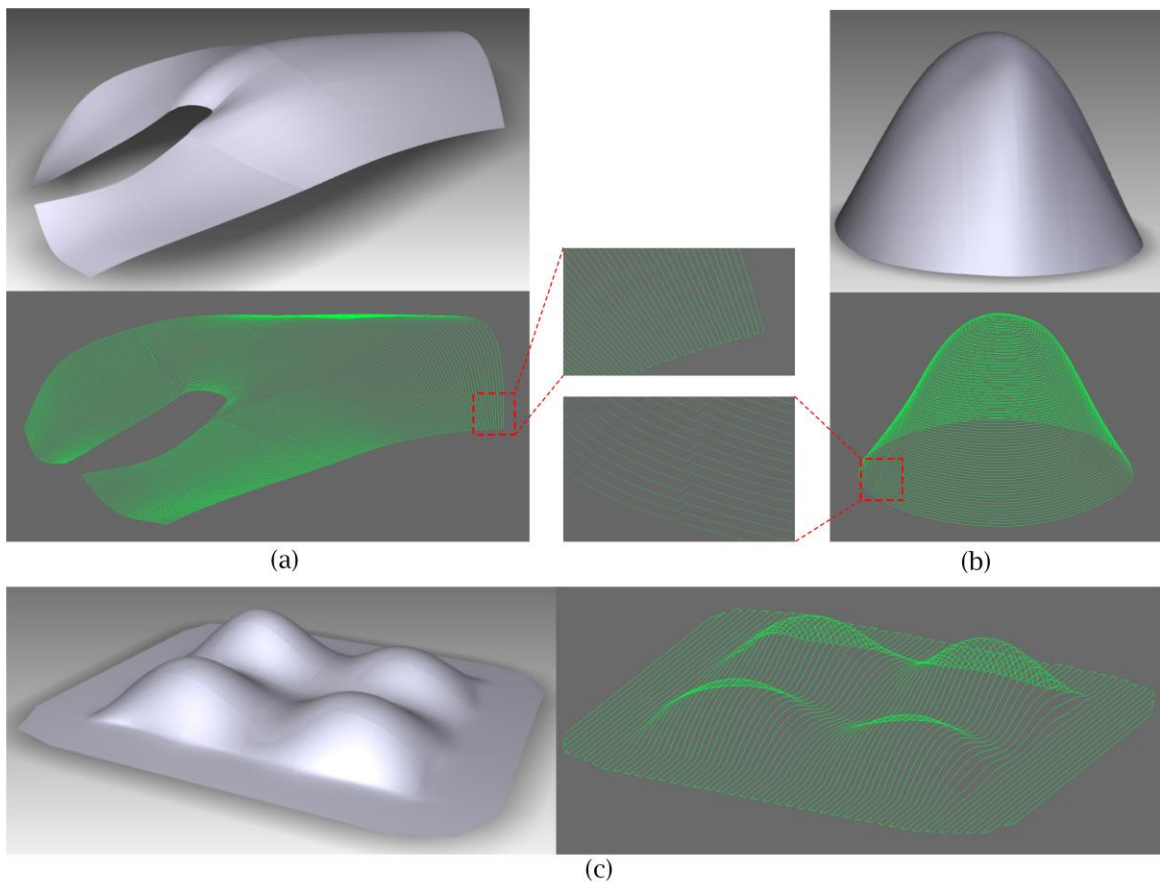


Fig. 4: Toolpath generation results for different T-spline models using: (a) UV strategy, (b) and (c) Plane cutter contact strategy.

A prototype STEP-CNC system enabled by T-spline

A practical STEP-CNC system supporting the T-spline freeform machining functionality has been developed. The STEP-CNC system is based 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. 5.

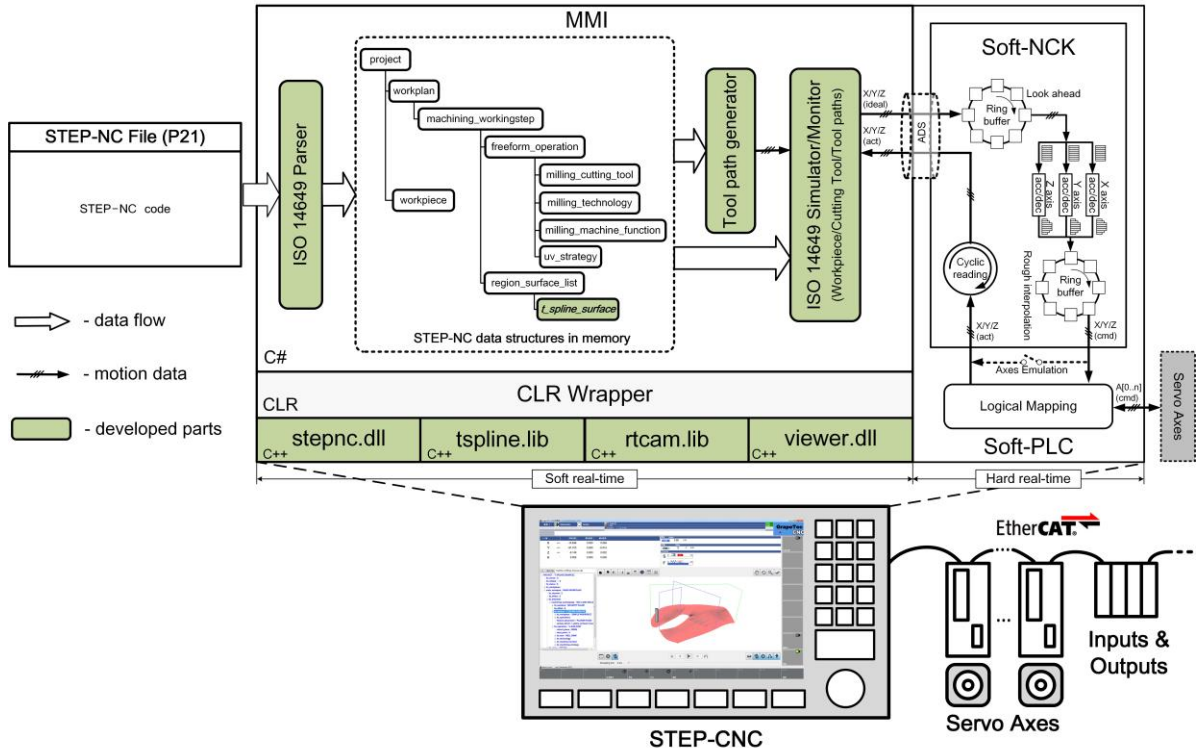


Fig. 5: System architecture, data flows and motions flows of the STEP-CNC system.

In order to integrate the T-spline with the STEP-CNC most modifications have been carried out on the MMI. In particular, a STEP-NC parser, an on-line 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 is parsed into memory and passed to the toolpath generator, then obtained toolpaths, after simulation and verification, are sent to the NCK via ADS (Automation Device Specification).

The second main architectural component PLC is developed according to the IEC 61131-3 standard using the soft-PLC technology [1]. The soft-PLC kernel provides the logical mapping functionality to Ether-CAT connected devices, such as servo axes, inputs & outputs, etc.

NCK consists of a look ahead buffer, which allows to reduce the real-time requirement on MMI, an accelerator/decelerator to generate the gently changed TCP speeds, and a rough interpolator providing a capability to smooth the axis motions.

To demonstrate the STEP-CNC operations an exemplary P21 data file is introduced. It contains a T-spline surface with its freeform machining operation. As shown in Fig. 5, all the STEP-NC data structures in the memory are kept in the project. Freeform machining operation is contained in a single machining working step presented within the workplan.

Conclusions:

The integration of T-spline within STEP-CNC systems is an important attempt towards the development of intelligent manufacture. The on-line toolpath generation algorithms implemented in this paper prove the possibility and efficiency to use T-spline as the freeform machining feature under the STEP-NC architecture. The prototype system can be further developed to be an intelligent STEP-CNC system, since the on-line toolpath generation is necessary to the on-line machining adjustment. The further STEP-CNC system can automatically check the status of machine tool (from intelligent sensors) and the workpiece to be machined (from STEP-NC codes), and generate the most appropriate toolpaths using some intelligent methods, so that different strategies and tools can be chosen automatically. In consideration of these requirements, a robust and efficient toolpath generator, especially for freeform machining, has become more and more necessary.

Acknowledgements:

This research was supported by the Chinese National Natural Science Foundation through grants 61572056,31327901

References:

- [1] Huan, J.; Jing, Y.; Xiao, W.: Industrial control programming language based on IEC61131-3 CNC system software design, 2011.
- [2] ISO 14649-11. Industrial automation systems and integration - physical device control - data model for computerized numerical controllers - part 11: Process data for milling, 2004.
- [3] ISO 10303-238. Industrial automation systems and integration Product data representation and exchange - part 238: Application protocol: Application interpreted model for computerized numerical controllers, 2007.
- [4] Patrikalakis, N. M.; Maekawa, T.; Cho, Wonjoon: Shape interrogation for computer aided design and manufacturing, Hyperbook edition, 2009. <http://web.mit.edu/hyperbook/Patrikalakis-Maekawa-Cho/mathe.html>
- [5] Suh, S. H.; Lee, B. E.; Chung, D. H.; Cheon, S. U.: Architecture and implementation of a shop-floor programming system for STEP-compliant CNC, Computer-Aided Design, 35(12), 2003, 1069-1083. [http://dx.doi.org/10.1016/S0010-4485\(02\)00179-3](http://dx.doi.org/10.1016/S0010-4485(02)00179-3).
- [6] Suresh, K.; Yang, D. C. H.: Constant scallop-height machining of free-form surfaces, Journal of engineering for industry, 116(2), 1994, 253-259. <http://dx.doi.org/10.1115/1.2901938>
- [7] Sederberg, T. W.; Zheng, J.; Bakenov, A.; Narsi, A.: T-splines and T-NURCCs, ACM Transactions on Graphics, 22(3), 2003, 477-484. <http://dx.doi.org/10.1145/882262.882295>
- [8] T-spline kernel, <http://sourceforge.net/projects/tspline> & <http://github.com/GrapeTec/T-SPLINE>, An Open-source T-spline kernel, 2016.
- [9] Xiao, W.; Liu, Y.; Li, R.; Wang, W.; Zheng, J.; Zhao, G.: Reconsideration of T-spline data models and their exchanges using STEP, Computer-Aided Design, 79, 2016, 36 - 47. <http://dx.doi.org/10.1016/j.cad.2016.06.004>
- [10] Xu, X.; Newman, S. T.: Making CNC machine tools more open, interoperable and intelligent - a review of the technologies, Computers in Industry, 57(2), 2005, 141-152. <http://dx.doi.org/10.1016/j.compind.2005.06.002>