Computer-Aided Design 56 (2014) 120–132

Contents lists available at ScienceDirect

Computer-Aided Design

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

A generic uniform scallop tool path generation method for five-axis machining of freeform surface ☆

Zhiwei Lin, Jianzhong Fu *, Hongyao Shen, Wenfeng Gan

The State Key Lab of Fluid Power Transmission and Control, Department of Mechanical Engineering, Zhejiang University, Hangzhou, 310027, China

HIGHLIGHTS

- Uniform scallop tool path has been generated via cutting simulation.
- Grass/CC rings are calculated both in parametric and 3D Euclidean space.
- Optimized methods are used to fast calculate the grass/CC ring.
- The method is free of local geometry assumptions; thus is more precise.
- The method is generic for any cutter, parametric surface and tool path pattern.

ARTICLE INFO

Article history:
Received 7 April 2013
Accepted 18 June 2014

Keywords:
Five-axis machining
Uniform scallop tool path
Cutting simulation
Ring

ABSTRACT

In this paper, a generic uniform scallop tool path generation method for five-axis machining is presented. Unlike the conventional methods which are based on the local surface geometry assumptions, this method is inspired by cutting simulation. Initially, the designed surface is planted with dense grasses. If a cutter is put onto the surface, the affected grasses will be cut short. All the affected grasses form a grass ring on the surface. When the cutter moves along the previous tool path, the envelope of the grass rings will form a machining band. Based on the machining band, cutter contact points can be found on the surface to ensure that the cutting edge touches exactly on the side of the band. These cutter contact points are fitted to construct the next tool path. In this way, all the tool paths can be generated recursively. An optimization is also developed to improve the computing efficiency of the path generation process. The proposed uniform scallop tool path generation method is generic. It can be popularized to (1) any kind of end mill with various sizes, (2) any kind of parametric surface and (3) directional- or contour-parallel tool path topologies. Another salient feature of this method is that it is free of local surface geometry assumptions, so the obtained tool paths are more precise. The proposed method is implemented and evaluated with several freeform surface examples. The feasibility of the method is also verified by actual cutting experiment.

1. Introduction

Freeform surfaces are widely used in various industry applications. It is defined in a parametric way as the form $S(u, v)$, in which u and v are two parameters. Usually, freeform surfaces are sent to CAM systems before they can be machined on three- or five-axis machine tools. The main task of the CAM systems is to plan tool paths on the input surface. The tool path can be classified into roughing and finishing according to different machining stages. In our previous work [1], we were able to generate contour-parallel roughing tool path. In this work, we put our focus on five-axis finishing tool path generation.

A tool path defines the feed directions of the cutter on the surface. There are two basic error types involved in generating the finishing tool path: the chord error and the scallop error. The chord error is defined in the tool feed direction along the tool path, which is used to evaluate the cutter trajectory precision between two continuous cutter contact (CC) points; while the scallop error is defined in the cross section of the two adjacent tool paths, which is used to evaluate the surface roughness after machining. In the tool path generation strategy presented by Lo and Lin [2], these two errors are integrated to get the finishing tool path. In this paper, however, the term tool path refers to an accurate curve on the designed surface. So in the tool path planning stage, the chord error is not taken into consideration.

☆ This paper has been recommended for acceptance by Kin-Chuen Hui.

* Corresponding author. Tel.: +86 13819193086.
E-mail address: fjz@zju.edu.cn (J. Fu).

http://dx.doi.org/10.1016/j.cad.2014.06.010

The quality of the finishing tool path has a direct influence on the final surface precision and the machining efficiency. For a high-quality tool path, the machining strip should be maximized while the geometric errors should be minimized [3]. Among the available tool path planning strategies, the iso-parametric strategy seems to be the most straightforward. In an iso-parametric path, the cutter moves along constant parametric curves. The side step is constrained by maximum scallop between two adjacent parametric curves. Obviously, the maximum scallop must be smaller than the scallop allowance. Another strategy which is often used is the iso-planar strategy. The only difference between the iso-parametric and iso-planar strategies is that for the latter, tool path curves are obtained by intersecting planes with the designed surface. Both the iso-parametric and iso-planar strategies belong to the category of constrained scallop strategy, which is characterized as low efficiency as it might have overlapped or redundant tool paths. The uniform scallop (iso-scallop) strategy finds a balance between the machining precision and efficiency. It attracts much research attention in this area since it was first proposed probably by Suresh and Yang [4]. However, it has never been an easy task to generate the iso-scallop tool path due to the irregular curvature distributions of freeform surfaces [5], especially for five-axis machining, which has two additional motion degrees and is considered to be collision-prone.

Five-axis machining has brought in many advantages including faster material-removal rates, improved surface finish, and the elimination of hand polishing [6], as well as new problems such as tool positioning and collisions. In five-axis machining, the tool positions including its orientations should be carefully adjusted as to achieve the maximum machining strip. Methods such as curvature matching machining [7,8], principle axis method [9], multi-point machining [10,11] and arc-intersect method [12] are interesting. For collision detection and avoidance issue, it has been a significant research topic in five-axis machining during the last decades [13–18]. In our previous work presented in [19], we aimed to efficiently calculate tool admissible area for any surface point, using the idea of sample admissible areas interpolation. In this work, the collision problem is left as an interface so it is not discussed at present. Also, this interface is compatible with the tool positioning problem.

The main difficulty in generating the five-axis iso-scallop tool path is finding the corresponding CC point of the next tool path for a current CC point of the current tool path. On one hand, the surface curvature in the vicinity of the current CC point is irregular, making it nearly impossible to precisely formulize the position of the next iso-scallop CC point; on the other hand, as the tool orientation of the next point is unknown, the effective cutting radius (for flat- or fillet-end mills) is unknown, either. Meanwhile, one has to be careful to avoid the possible gouging or collision problems when adjusting the tool orientations.

The purpose of this work is to build a framework for generating five-axis iso-scallop tool paths via cutting simulation. Many techniques can be used to do cutting simulation [20–24], in which a well-known method is the normal vector method (NVM) [9]. This method can be vividly described as mowing the lawn on the surface. At first, grasses (considered as straight) with appropriate lengths are planted perpendicularly onto the surface. Then a cutter is forced to move on the surface along a tool path. Any grass that makes contact with the cutting edge of the cutter is cut short at the intersecting point. As the cutter reaches the end of the path, the resulting grasses can be used to approximate the machined surface. The precision of the simulated surface is controlled by the density of the grasses on the surface. In this work, the concept of NVM is adopted to calculate the grass curve for the current tool path. And then the next iso-scallop tool path is calculated based on the grass curve. The benefit of using this method is that no surface curvature assumptions are needed, thus cutters with larger radii can be used. The fillet-end mill is used in this work. The advantage of fillet-end mills over ball- or flat-end mills is that it inherits the merits of the other two cutters and produces smaller scallops across the feed direction and low roughness along the feed direction [25].

The rest of the paper is organized as follows. Section 2 studies the backgrounds of iso-scallop tool path. In Section 3, some preliminary definitions and operations throughout the paper are given. The intermediate grass curve is generated in Section 4. Based on the grass curve, the tool path curve is generated in Section 5. Section 6 optimizes the proposed tool path generation method. Some iso-scallop tool paths examples including a real cutting test are given in Section 7. The last section concludes this paper.

2. Backgrounds of iso-scallop tool path

The iso-scallop tool paths are usually generated recursively [26]: initially, a primary tool path from which the other paths are constructed is selected, usually, from the surface boundary curves. Then, all the CC points of the next tool path are calculated from the corresponding CC points of the current path, such that the scallop between the two paths remains at constant height. Theoretically, if these newly generated CC points are dense enough, they can be used directly as a tool path. Practically, these points cannot be too dense in order to save computation. They are usually fitted (interpolated or approximated) to a cubic spline, either in 3D physical space or 2D parametric domain. In practice, 2D parametric fitting is superior to 3D physical fitting in terms of accumulation errors as the former resulting splines remain staying always on the designed surface. The secondary path is then used as a primary path for generating another tool path. This process keeps recursively running until the stop condition is met.

The real challenge here is to compute the next CC point according to the current one in the path interval direction. In three-axis machining, as ball-end mills are used, things are relatively easier because one only needs to handle the irregular freeform surfaces. For five-axis machining, flat-end or fillet-end mills are often used. In this case, the challenge is even bigger, because one has to deal with the irregular cutter surfaces, too.

Suresh and Yang [4] made probably the first attempt to plan tool path with constant scallop criterion for three-axis machining with ball-end mills. In their work, the classic scallop models on flat, convex and concave spherical surfaces (with regular curvature) are built (as shown in Fig. 1) and formulated as:

$$g = \frac{2a}{1 + \text{sign} \cdot r/R}, \tag{1}$$

in which g is the path interval in physical space, r is tool end radius, R is the surface curvature, and $2a$ (as AB in Fig. 1) is the distance between two adjacent tool centers, as shown in Fig. 1. *sign* is an operator whose value is 0 for flat surface, 1 for convex surface and −1 for concave surface. In Eq. (1), the solution of a requires a tedious iterative process. Local surfaces are approximated in the neighborhood by spheres (referred as sphere assumption) so that the path interval can be calculated with Eq. (1). This path interval is converted (referred as conversion assumption) to parametric interval with the first order Taylor expansion, thus the next CC point can be located on the surface. Lin and Koren [27] concerned that the first order Taylor expansion is only valid when the converted parametric interval is very small. They introduced a more accurate conversion technique based on both the second order Taylor expansion and an error compensation method. Besides, they further developed Eq. (1) and gave a much more neat expression of the path interval, as:

$$g = \sqrt{\frac{8hrR}{R + \text{sign} \cdot r}}. \tag{2}$$