Five-axis Tool Path Generation for Sculptured Surface Machining using Rational Bézier Motions of a Flat-end Cutter

W. Zhang[1], Y. F. Zhang[2]and Q. J. Ge[3]

[1]National University of Singapore
[2]National University of Singapore, mpezyf@nus.edu.sg
[3] State University of New York at Stony Brook, Qiaode.Ge@stonybrook.edu

ABSTRACT

This paper presents an efficient approach that uses rational Bézier motions to generate 5-axis tool path for sculptured surface machining (finish cut) with a flat-end cutter. A method is proposed in which dual quaternion is used to represent a spatial displacement. The representation of kinematic motions for the cutter bottom circle of the flat-end cutter is then formulated. Based on that, a new approach for tool path generation using rational Bézier cutter motions is described, in which key issues such as interference avoidance and surface accuracy requirement are addressed.

Keywords: Rational Bézier motion, dual quaternion, sculptured surface machining, 5-axis tool path generation, gouging and collision avoidance.

1. INTRODUCTION

In modern manufacturing, 5-axis machining is commonly used in automotive, aerospace and tooling industries. Compared to 3-axis machining, 5-axis machining offers many advantages, including the ability to manufacture complex parts with free form surface, better material-removal rates, improved surface finish, reduced number of set-ups and thus increased productivity [1]. However, for the moment, 5-axis machining tool path generation remains a difficult task, which is mainly due to the complicated tool movements and the irregular curvature distributions of sculptured surfaces.

In order to overcome the problems of 5-axis tool path generation, different approaches have been developed over the last decade. Dragomatz and Mann [2] provided a classified bibliography of the literature on NC tool path generation. Choi and Jerard [3] also gave an extensive introduction of the 5-axis machining. Some approaches focus on tool path planning strategies, such as iso-parametric machining [4], iso-planar machining [5] and constant scallop height machining [6,7], while some focus on geometric issues, such as the scallop height [8,9] and effective cutting shape [10]. Much efforts focused on obtaining the optimal cutter orientations to improve the efficiency of the tool path generation process [11,12]. Much work also focuses on the detection and avoidance of gouging and collision [13,14,15]. Gouging occurs when portions of the cutter's bottom extends below the part surface, while collision is regarded as the global gouging in which the cylindrical portion of the cutter interferes with the part surface or the machine tool. To date, most of the previous efforts focused on the development of the algorithms for gouging and collision avoidance.

In general, most of the reported tool path generation methods are numerical and discrete in nature [16]. They basically follow a two-step approach:

(1) Given a surface description (either in NURB representation or triangular polyhedral meshes), a set of cutting contact (CC) points is generated based on a machining strategy and the given surface error tolerance.

(2) For each CC point, the cutter location (CL) is determined that avoids gouging and collision and is within the machine's axis limits.

In order to satisfy the surface error tolerance, the number of CC points is generally very large. At the same time, algorithms that search for a feasible CL for a CC point are iterative in nature, which normally leads to extremely long computation time. A further drawback of

this kind of approach is that the complete elimination of gouging or collision between the neighbouring CLs is not guaranteed. Instead of focusing on a particular instant of the tool motion and studying local geometric issues at the instant, tool path can be generated as envelopes of moving cutter. Wang and Joe [17] presented that surfaces can be generated by sweeping a profile curve along a given spline curve. Juttler and Wagner [18,19] proposed a method to generate rational motion-based surface emphasizing the special cases of a moving cylinder of cone of revolution. Ge and Srinivasan [20] presented two algorithms for fine-tuning rational *B*-spline motions suitable for computer aided design. Xia and Ge [21 , 22] provided the representation of the boundary surfaces of the swept surface undergoing rational Bézier and *B*-spline motions and proposed a corresponding method for 5-axis tool path generation. It provides an efficient approach to generate the tool path, and at the same time, allows an accurate representation of the swept surface generated by the cutter. However, their work has not yet explicitly dealt with the issue related to interference detection and avoidance. The work reported in this paper advances their work by developing an algorithm for generating an entire tool path for a given designed surface while avoiding interference.

2. REPRESENTATION OF KINEMATIC MOTIONS FOR THE FLAT-END CUTTER

2.1 Representing a Spatial Displacement with a Dual Quaternion

A dual quaternion is the combination of dual number and quaternion, which has the form of $\hat{\mathbf{q}} = \mathbf{q} + \varepsilon\mathbf{q}^0$. The symbol ε represents the dual unit which has the property $\varepsilon^2=0$, and the real part $\mathbf{q}$ and the dual part $\mathbf{q^0}$ are all quaternions. The quaternion is a hypercomplex number consisting of a real part and 3 imaginery parts:

$$\mathbf{q}=q_1\mathbf{i}+ q_2\mathbf{j}+ q_3\mathbf{k}+ q_4 \quad (1)$$

where q_i (i=1, 2, 3, 4) are real numbers, called the components of $\mathbf{q}$, and four quaternion units 1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ satisfy the relations $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$ and $\mathbf{ij} = -\mathbf{ji} = \mathbf{k}$. A unit quaternion is a quaternion with $\left\|\mathbf{q}^2\right\| = \sum q_i^2 = 1$.

Traditionally, translation is represented by a vector $\mathbf{d}$ and rotation is represented by an orthogonal matrix [**A**]. Thus, a spatial displacement in Euclidean three-space E^3 is commonly expressed by [**A**] and $\mathbf{d}$ as:

$$\widetilde{\mathbf{P}} = \begin{pmatrix} [\mathbf{A}] & \mathbf{d} \\ 0\ \ 0\ \ 0 & 1 \end{pmatrix}\mathbf{P} \quad (2)$$

where $\widetilde{\mathbf{P}}$ and $\mathbf{P}$ are homogeneous coordinates of a point measured in the fixed and moving reference frames. Dual quaternion can also be used to represent spatial displacement. The 4 components of $\mathbf{q}$ is expressed by the homogeneous Euler parameters of rotation as:

$$\mathbf{q}=(q_1,q_2,q_3,q_4)=(s_1\sin(\theta/2),s_2\sin(\theta/2),s_3\sin(\theta/2),\cos(\theta/2)) \quad (3)$$

where (s_1, s_2, s_3) define the unit vector $\mathbf{s}$ along the axis of rotation and θ denotes the angle of rotation. Eq. (3) contains the information of rotational component of a spatial displacement. The 4 components of the dual part $\mathbf{q}^0$ is another quaternion whose components are defined as:

$$\begin{bmatrix} q_1^0 \\ q_2^0 \\ q_3^0 \\ q_4^0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -d_z & d_y & d_x \\ d_z & 0 & -d_x & d_y \\ -d_y & d_x & 0 & d_z \\ -d_x & -d_y & -d_z & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (4)$$

where $\mathbf{d}=(d_x, d_y, d_z)$ is the translation vector. $\mathbf{q^0}$ includes the information of the translation of a spatial displacement.

Since the real part of $\hat{\mathbf{q}}$ represents the rotation and the dual part represents the translation of a spatial displacement, dual quaternion is capable of representing transformation. A spatial displacement in E^3 is expressed by:

$$\widetilde{\mathbf{P}} = \mathbf{q}\mathbf{P}\mathbf{q}^* + p_4(\mathbf{q}^0\mathbf{q}^* - \mathbf{q}\mathbf{q}^{0*})\mathbf{P} \quad (5)$$

where"*"denotes the conjugate of a quaternion. As with matrices, the multiplication of two dual quaternions $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ using quaternion algebra is still a dual quaternion that composes the two basic transformations.

2.2. Representing Point Trajectory using Rational Bézier Dual Quaternion

Given a moving frame O_M - $X_MY_MZ_M$, a fix frame O_F - $X_FY_FZ_F$, and a point $\mathbf{P}$ in the moving frame, one can get the point trajectory of $\mathbf{P}$ through the motion of the moving frame relative to the fix frame, as shown in Fig. 1. Points $\mathbf{P}_A$ and $\mathbf{P}_B$ are on the point trajectory and represent the intermediate position for point motion. Denoting $\widetilde{\mathbf{P}}$ and $\mathbf{P}$ as homogeneous coordinates of a point measured in the fixed and moving frames, one can obtain the transformation between $\widetilde{\mathbf{P}}$ and $\mathbf{P}$ using dual quaternion $\hat{\mathbf{q}}$ according to section 2.1. Therefore, when point $\mathbf{P}$ moves to point $\mathbf{P}_A$, transformation between $\widetilde{\mathbf{P}}_A$ and $\mathbf{P}_A$ can be represented by $\hat{\mathbf{q}}_A$; similarly, the transformation between $\widetilde{\mathbf{P}}_B$ and $\mathbf{P}_B$ can be represented by $\hat{\mathbf{q}}_B$.

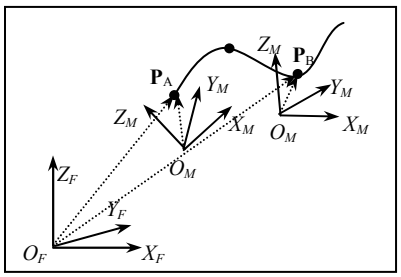


Fig. 1. Point trajectory of **P** generated by the motion of frame O_F-$X_FY_FZ_F$

Given a set of dual quaternion $\hat{\mathbf{q}}_\mathbf{i}$ that represents transformation between point $\widetilde{\mathbf{P}}$ and **P** at different point positions as point **P** is undergoing motions, one can construct a B-spline dual quaternion curve that represents the motions of point **P**. Therefore one can interpolate the transformation between point $\widetilde{\mathbf{P}}$ and **P** at arbitrary position. As a result, according to Eq. (5), one can determine the coordinate of point **P** in the fix frame at arbitrary position.

In order to construct the rational Bézier dual quaternion curve that passes through a set of quaternion $\hat{q}_\mathbf{i}$ (i =1,…n), a set of control dual quaternion $\hat{\mathbf{d}}_j$ (j = 1,…n) needs to be obtained first. Then the rational Bézier dual quaternion curve is given as:

$$\hat{\mathbf{q}}(t) = \sum_{j=1}^{n+1} B_j^n(t)\hat{\mathbf{d}}_j \tag{6}$$

where $B_j^n(t)$ denotes the rational Bézier basis functions.

According to Eq. (5) and (6), we can get the coordinate of point **P** in the fix frame as:

$$\widetilde{\mathbf{P}}(t) = [H^{2n}(t)]\mathbf{P} \quad \text{and} \quad H^{2n}(t) = \sum_{k=0}^{2n} N_k^{2n}(t)[H_k] \tag{7}$$

For details, please refer to [23].

2.3. Representation of Cutter Bottom Circle Undergoing a Rational Bézier Motion

From the rational Bézier representation of point motion, we can determine the representation of a flat-end cutter's bottom circle undergoing a rational Bézier motion (see Fig. 2) by applying the same principle.

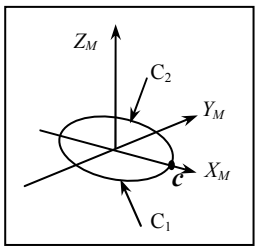


Fig. 2. Position of cutter bottom circle in the moving frame

We first construct a circle in the moving frame. Assume that the axis of the cylinder is along the *Z*-axis and choose $\mathbf{P}_0$=(r,0,0,1), $\mathbf{P}_1$=(0,-r,0,0), $\mathbf{P}_2$=(-r,0,0,1), as the homogeneous coordinates of three Bézier control points, as shown in Fig. 2. The half circle C_1 below *X*-axis can be expressed as:

$$\mathbf{P}(s) = \sum_{i=0}^{2} B_i^2(s)\mathbf{P}_i \tag{8}$$

Similarly, the circular arc C_2 above *X*-axis can be represented by the same formula except $\mathbf{P}_1$=(0,r,0,0). Substitute Eq. (8) into Eq. (7), the circular arc undergoing the rational Bézier motion can be expressed as:

$$\mathbf{P}(s,t) = [H^{2n}(t)]\sum_{i=0}^{2} B_i^2(s)\mathbf{P}_i = \sum_{k=0}^{2n}\sum_{i=0}^{2} N_k^{2n}(t)B_i^2(s)[H_k]\mathbf{P}_i \tag{9}$$

3. THE GEOMETRY OF FIVE-AXIS MACHINING

Here, the designed surface to be machined is a *B*-spline surface. A flat-end cutter is used for machining. Fig. 3 shows the geometry of the surface and cutter in 5-axis machining. In Fig. 3, O_G-$X_GY_GZ_G$ is the global coordinate system, which is fixed on the workpiece. O_L-$X_LY_LZ_L$ is the local coordinate system that is centred at the CC point. The CL point is the centre of the bottom of the cutter, while a CL includes the CL point as well as the orientation of the cutter. X_L is in the direction of the tangent vector at the CC point along current cutting direction. Z_L is in the direction of the normal of the surface at the CC point. O_T-$X_TY_TZ_T$ is the cutter coordinate system that is obtained by first rotating angle λ_L around Y_L and second angle ω_L around Z_L, and then translating $-r$ along X_L, where r is the cutter radius. The transformation between the cutter and global frames can be represented by dual quaternion $\hat{\mathbf{q}}$ as:

$$\hat{\mathbf{q}} = \hat{\mathbf{q}}_{trLG} \cdot \hat{\mathbf{q}}_{r\lambda} \cdot \hat{\mathbf{q}}_{r\omega} \cdot \hat{\mathbf{q}}_{-t\mathbf{r}} \tag{10}$$

where $\hat{\mathbf{q}}_{trLG}$ is the dual quaternion that represents the transformation from the local frame to the global frame; $\hat{\mathbf{q}}_{r\omega}$ is the dual quaternion that represents rotation of angle ω_L around Z_L; $\hat{\mathbf{q}}_{r\lambda}$ represents rotation of angle λ_L

around Y_L; $\hat{\mathbf{q}}_{-tr}$ represents the translation of $-r$ along X_L.

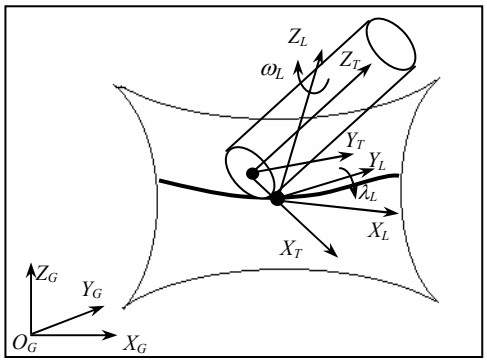


Fig. 3. The geometry of 5-axis milling

4. ISO-PARAMETRIC TOOL PATH GENERATION USING RATIONAL BÉZIER MOTION

Here, we present a method for iso-parametric tool path generation using rational Bézier cutter motion. The rational Bézier dual quaternion curve is constructed to obtain the mathematic forms of the cutter motion along the surface curve; thus, the computation of huge number of discrete CC points is avoided. Since the swept surface of cutter's bottom circle undergoing rational Bézier motion can be expressed in Eq. (9), gouging checking can be considered as finding interference between the swept surface of cutter bottom undergoing rational Bézier motion and the designed surface, while gouging avoidance can be considered as modifying the swept surface of cutter bottom motion so that the interference between the swept surface and the designed surface no longer exists.

The algorithm is implemented in three steps. Given an iso-parametric curve on the design surface, the first step is to find the dual quaternions correspond to several discrete CLs, which is used to construct the motion curve of cutter bottom and consequently construct the swept surface of cutter bottom circle undergoing ration Bézier motion. In the second step, the rational Bézier dual quaternion curve for the cuter motion is constructed and the swept surface of cutter bottom circle is obtained. After that, fitness and interference checking between the swept surface of the cutter bottom and the designed surface is performed and correction of the rational Bézier dual quaternion curve is carried out if the tool path is unsatisfactory. Finally, a satisfactory tool path is obtained.

4.1. Determining Dual Quaternions of CLs on the Tool Path

In order to construct the swept surface of cutter bottom, the control dual quaternions must be determined first. In our application, we calculate the dual quaternion between global coordinate system and the cutter coordinate system (see Fig. 3). To determine the dual quaternions between these two coordinate systems, we must know the position of the point in each of the coordinate system. Since the global coordinate system is fixed, we can determine the dual quaternion when we know the location of the cutter coordinate system. Therefore, the problem of finding the discrete dual quaternions is equivalent to find several CLs on the tool path. This can be solved in two steps: (1) find several CC points on the surface curve, and (2) construct the gouging and collision free CLs associated with the CC points.

4.1.1. Determining the cutter contact (CC) points

There are two methods to determine the CC points: one is to find the evenly distributed CC points along the surface curve, and the other is to find the CC points which could reflect the shape of the surface curve. The second approach is employed, i.e., the number and distribution of these CC points depends on the curvature of the surface curve.

Given the current CC point $\mathbf{C}_i$, one need to find the next CC point $\mathbf{C}_{i+1}$ on $\mathbf{S}(u_0,v)$ and to ensure that the fitting error between two neighbouring CC points is within the fitting tolerance τ. In this stage, the fitting tolerance τ should be chosen carefully. To avoid a large number of CC points, we use a rather relaxed bound as the fitting tolerance. In our approach, the CC points chosen in this stage only serve as a starting set and the final surface error will only be accurately checked and ensured after the whole tool path is generated.

Two steps are involved in determining the next CC point. First, a search for the adaptive step size L_i is conducted, which is based on the local curvature of surface curve at the i-th CC point and the pre-defined fitting tolerance τ. After that, the conversion from the step size L_i to the increment of parameter v_i is performed and the next CC point is obtained. L_i can be calculated by [14]:

$$L_i = \sqrt{\frac{8\tau - 4\tau^2 \kappa_n}{\kappa_n}} \tag{11}$$

where κ_n is the curvature of the intersection curve between the surface and the plane containing the surface normal $\mathbf{n}$ and $\mathbf{f}$ [24]. L_i is then converted to Δv_i, and the next CC point is $\mathbf{S}(u_0,v_i+\Delta v_i)$. Δv_i (see Fig. 4b) is determined by [9]:

$$\Delta v_i = L_i / (S_v \cdot \mathbf{X}_L) \tag{12}$$

Δv_i is only an approximated value since the surface curve $\mathbf{S}(u_0,v)$ at the vicinity of $\mathbf{C}_i$ is considered as a circular curve. Therefore, we need to calculate the maximum deviation d_{max} between the points on surface curve $\mathbf{S}(u_0,v)$, where $v_i \le v \le v_i + \Delta v_i$, and the linear segment connected by $\mathbf{S}(u_0, v_i)$ and $\mathbf{S}(u_0, v_i + \Delta v_i)$, as shown in Fig. 4c. If d_{max} is larger than τ, decrease the parameter increment Δv_i, and re-calculate d_{max} again until it is less than τ.

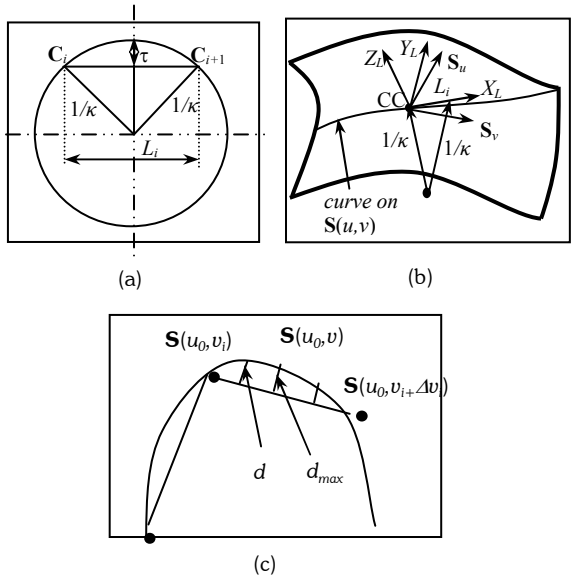


Fig. 4. The geometry of surface curve $\mathbf{S}(u_0,v)$ at the vicinity of $\mathbf{C}_i$

4.1.2. Obtaining the associate gouging-free and collision-free cutter locations (CLs)

Given a set of CC points calculated in section 4.1.1, one can obtain the CLs if the orientations (the inclination angle λ_L and tilting angle ω_L) of the cutter are given at each CC point. In order to have less modification later, gouging and collision are avoided when determining the orientations of the cutter. The proposed method follows a checking-correction approach. The default inclining and tilting angles are assigned, followed by a checking procedure for interference. If interference exists, a correction procedure is applied to change these two angles.

Intersection checking between two surfaces is not a trivial task. We simplify this problem by converting the cutter bottom plane into a limited number of circles (see Fig. 5a), and then checking the intersection between the circles and the designed surface. Similarly, we also simplify the intersection checking between the cylindrical surface of the cutter and the designed surface by converting the cylindrical surface into a limited number of circles (see Fig. 6a), and then checking the intersection between the circles and the designed surface.

To check the possible intersection between a circle and the designed surface, we calculate the minimal distance between the circle and the surface. The Downhill Simplex method [25] is employed. For gouge checking, denote the identified point on the cutter bottom as $\mathbf{Q}$ and the point on the designed surface as $\mathbf{P}$, the existence of gouging is one of the following:

(1) If the minimum distance is closed to 0 ($|\mathbf{P} - \mathbf{Q}| < \rho$, where ρ is a very small value) and $\mathbf{Q}$ is outside the vicinity of CC point, gouging is said to occur.

(2) If $|\mathbf{P} - \mathbf{Q}| \ge \rho$, there are two possibilities. The first is that no interference exists (see Fig. 5b) and the second is that the cutter bottom is completely below the designed surface (see Fig. 5c). To distinguish these two scenarios, we calculate the angle, φ, between vector $\mathbf{PQ}$ and the normal vector at $\mathbf{P}$, i.e., $\mathbf{n}$. If $\varphi \le 90°$, as shown in Fig. 5b, there will be no gouging. If, however, $\varphi > 90°$, gouging will occur.

For collision checking, denote the identified point on the cutter cylindrical surface as $\mathbf{Q}$ and the point on the designed surface as $\mathbf{P}$, the existence of collision is determined in the following scenarios (see Fig. 6b,c,d):

(1) If the minimum distance is closed to zero ($|\mathbf{P} - \mathbf{Q}| < \rho$, where ρ is a very small value) collision is said to occur.

(2) If $|\mathbf{P} - \mathbf{Q}| \ge \rho$, we transform the point $\mathbf{P}$ on the designed surface in the global frame to the cutter frame. If the transformed $\mathbf{P}_T$ is within the volume of the cutter, collision occurs. Otherwise, there is no collision.

If gouging or collision exists, the corresponding CL needs to be adjusted. The strategy for adjusting the inclining and tilting angles is given as follows:

(1) Increase λ_L by a small amount and keep ω_L unchanged. Check the existence of gouging and collision.

(2) If no gouging and collision exists, output λ_L and ω_L, stop. Otherwise, check if the machine limit for λ_L is reached. If so, go to step (3). If not, go back to step (1).

(3) Increase ω_L by a small amount and keep λ_L at its default value. Go back to step (1).

It is worth mentioning that in the above procedure, there is no checking for the machine limit for ω_L. This is based on an assumption that for a given CC point, an interference-free pair of λ_L and ω_L always exists.

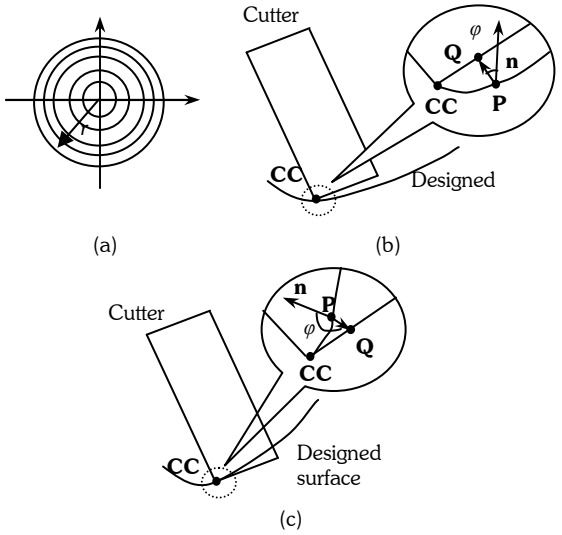


Fig. 5. Check interference between the cutter bottom and the designed surface

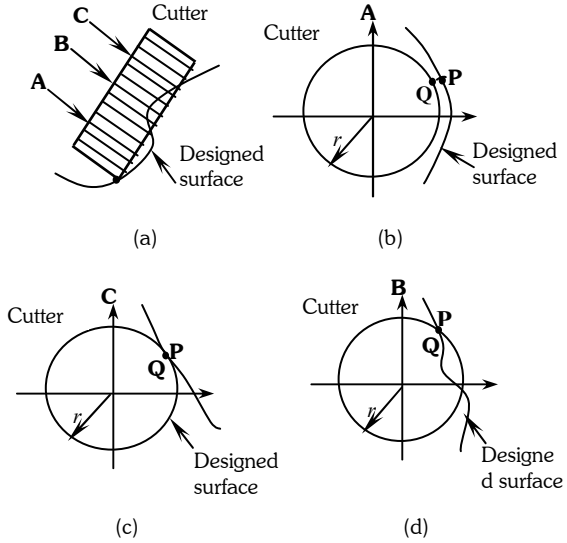


Fig. 6. Interference checking of designed surface and cutter cylindrical surface

4.2. Construct the Dual Quaternion Curve of Cutter Motion for One Tool Pass

Given a set of CLs, their dual quaternion representation $\hat{\mathbf{q}}_i$ can be obtained using Eq. (10). In this section, we construct a rational Bézier motion that interpolates or approximates the arbitrary cutter location on one tool path using the dual quaternion representation $\hat{\mathbf{q}}_i$ of these CLs. It can be considered as a curve construction problem, and there are several methods for curve interpolating and approximation [26]. The problem can be described as follows: Given a set of dual quaternion $\hat{\mathbf{q}}_i\ (0 \le i \le n)$ generated from the CLs for one tool path, and corresponding knot sequence $x_i\ (0 \le i \le n)$, find a segmented cubic rational Bézier dual quaternion curve $\mathbf{Q}$ determined by the knots sequence and a set of control dual quaternion $\hat{\mathbf{d}}_{-1}, \ldots \hat{\mathbf{d}}_{n+1}$ such that $\mathbf{Q}(x_i) = \hat{\mathbf{q}}_i$. We can obtain the relationship between the control dual quaternion $\hat{\mathbf{d}}_i$ and the known dual quaternion $\hat{\mathbf{q}}_i$ using de Casteljau algorithm [26]. For implementation details, please refer to [23].

4.3. Tool Path Verification and Modification

The dual quaternion curve of cutter motion generated represents a complete tool pass. Over this tool pass, however, the cutter positions between the seed neighbouring CLs may cause out-of-bound surface error and/or collision and gouging. Therefore, verification of the complete tool path on surface error, gouging and collision must be carried out. Here, we use the swept surface generated by the cutter bottom and the underlying surface to check the existence of these two problems. The swept surface of the bottom of a cutter undergoing rational motion is considered to be formed by two types of surfaces: one by the rational motion of bottom circle (type- II) and the other by the bottom surface within the bottom circle (type-I). For gouging problem, we need to check whether there is any interference between the swept surface (both type-I and type-II) and the designed surface. For surface error, however, we only need to check the deviation between type-II surface and the designed surface. This is because that the final generated surface must be the swept surface formed by the motion of the bottom circle if the tool path is gouging-free. The surface fitness and gouging checking are carried out on a number of selected CLs. The deviation between type-II surface and the designed surface is checked first. If it is out of tolerance, the respective CLs are recorded. Gouging and collision checking is subsequently carried out over the same set of CLs and the interference CLs are also recorded. These problematic CLs are then modified and the quaternion motion curve is reconstructed. Subsequently, fitness and interference checking are carried out again. This process goes on until a gouging-free and collision-free tool path with a satisfactory fitness is achieved. The algorithms are described in the following sections.

4.3.1. Fitness and interference checking

The deviation between the type-II surface and designed surface at a given CL can be defined as the minimal distance between the cutter bottom circle and designed surface. Here, we have chosen a sampling method based on the change of curvature of the type- II surface, i.e.,

the part of surface with large curvatures is sampled densely; while the part of surface with small curvatures is sampled sparsely. In implementation, we use the curve of the CC point undergoing the same rational Bézier motion as an approximation. First, we obtain the equation of this curve. As shown in Fig. 2, since point $\boldsymbol{c}$ (CC point) is the start point of two circular arcs expressed in Eq. (8) and $s = 0$. Thus, according to Eq. (9), the curve generated by point $\boldsymbol{c}$ undergoing the rational Bézier motion is given by $\mathbf{P}(0,t)$ as:

$$\mathbf{P}(0,t) = [H^{2n}(t)] \sum_{i=0}^{2} B_i^2(0)\mathbf{P}_i = \sum_{k=0}^{2n} \sum_{i=0}^{2} B_k^{2n}(t) B_i^2(0) [H_k]\mathbf{P}_i \tag{13}$$

The next step is to generate a set of CC points from $\mathbf{P}(0, t)$. For implementation, the method for CC-point generation described in section 4.1.1 is adopted, in which the designed surface $\mathbf{S}(u,v)$ and surface curve $\mathbf{S}(u_0,v)$ are replaced by the swept surface $\mathbf{P}(s,t)$ and the curve $\mathbf{P}(0,t)$. The given surface error tolerance is used as the fitting tolerance for CC-point generation. The generated CC-point set corresponds to a set of parameters $\{t_i, i=1, 2, \ldots, K\}$, which further corresponds to a set of CLs whose CC points can be represented as $\{\mathbf{P}(0, t_i), i=1, 2, \ldots, K\}$. The calculation of the minimum distance between a CC point at t_i and the underlying curve on the designed surface can be performed using the Downhill Simplex algorithm. The output gives a set of K points on $\mathbf{S}(u_0,v)$, $\{\mathbf{S}(u_0, v_i), i=1, 2, \ldots, K\}$, that correspond to the minimum distances at the K CLs. If the minimal distance at a CL is found to be larger than the surface error tolerance, the corresponding point, which belong to $\{\mathbf{S}(u_0, v_i), i=1, 2, \ldots, K\}$, that yield this distance is recorded into a *supplementary CC-point set*. The CLs that satisfy the tolerance are to be used for gouging and collision checking.

Next, interference detection needs to be carried out at instant CLs. For simplicity, the instant CLs for interference detection are the same set of CLs determined in the process of fitness checking. At these CLs, interference detection is similar to the gouging and collision detection method introduced in section 4.1.2. However, there is still a little difference between them. The adjustment of the incline angle of cutter at instant location is no longer needed at this stage, since the adjustment will be done later by modifying the rational Bézier dual quaternion curve. Note that the minimal distance calculated for interference detection is between the cutter and the designed surface as a whole. If interference occurs at a CL (t_i), the corresponding point on $\mathbf{S}(u_0,v)$ should be found and added to the supplementary CC-point set. The requirement for this point is that by using it as a CC point, a feasible CL can be found to avoid interference. Therefore, there should be more than one solution to this point. Here, we use the point on $\mathbf{S}(u_0,v)$ that corresponds to the minimal distance between $\mathbf{P}(0, t_i)$ and $\mathbf{S}(u_0,v)$, i.e., from $\{\mathbf{S}(u_0, v_i), i=1, 2, \ldots, K\}$, which is obtained during the above fitting checking procedure. Finally, the supplementary CC-point set is completed, which will be used to modify the curve that defines the cutter motion.

4.3.2. Modification of the rational Bézier dual quaternion curve

Having found the problematic CLs (on the dual quaternion curve), a direct modification approach is to use the points in the supplementary CC-point set to determine their CLs that are interference-free and have satisfactory fitting error. These CLs, together with the existing CLs, are then used to re-construct the dual quaternion curve.

For implementation, the points in the supplementary CC-point set are considered as new CC points, and the corresponding cutter postures (CLs) at these positions are then obtained based on gouging and collision avoidance described in section 4.1.2. The newly generated CLs are converted to dual quaternion representation. Thus the control quaternions are modified and a new quaternion motion curve is generated from these modified quaternions. The swept surface of cutter undergoing the new quaternion motion will have more contact points with the underlying curve. It is therefore expected that the new tool path have less problems in terms of fitting error, gouging and collision. Fitness, gouging and collision checking will be carried out on the new tool path. This checking-modification-checking process is repeated until fitting test, gouging and collision test are satisfactory.

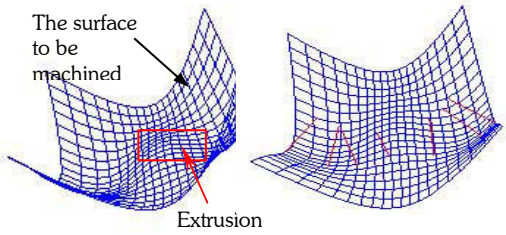
5. COMPUTER IMPLEMENTATION AND AN APPLICATION EXAMPLE

The proposed method has been implemented on PC using VC++ and OpenGL. In this section, an example is used to illustrate the capability of the developed algorithm. The example surface is a rational Bézier surface with control points:

$$\begin{bmatrix} (-2,0,0) & (-0.5,0,-2) & (0,0,-2) & (0.5,0,-2) & (2,0,0) \\ (-2,1,-1) & (-0.5,1,-3) & (0,1,-2) & (0.5,1,-3) & (2,1,-1) \\ (-2,2,-2) & (-0.5,2,-4) & (0,2,2) & (0.5,2,-4) & (2,2,-2) \\ (-2,3,-2) & (-0.5,3,-4) & (0,3,2) & (0.5,3,-4) & (2,3,-2) \\ (-2,4,-1) & (-0.5,4,-3) & (0,4,-2) & (0.5,4,-3) & (2,4,-1) \\ (-2,5,0) & (-0.5,5,-2) & (0,5,-2) & (0.5,5,-2) & (2,5,0) \end{bmatrix}$$

Fig. 7a shows the surface patch geometry. It can be seen that the patch is concave in general with an extrusion in the centre, which will cause the change of the tool postures in order to avoid interference. A flat-end cutter with the radius of 0.3 is used here. The position of the single tool pass is at $u = 0.3$.

First, a set of CC points were generated on the surface curve $\mathbf{S}(0.3, v)$, assigning the searching tolerance $\tau = 0.05$. Fig. 7b shows the normal vectors of the CC points generated under the surface error tolerances.

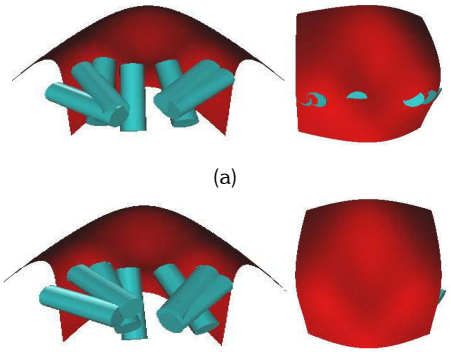


(a) (b)

Fig. 7. (a) The designed surface patch (b) CC points with τ =0.05

Second, the associated CLs associated to the CC points are generated. The default inclining and tilting angles were chosen as λ_L=5° and ω_L=0°. Fig. 8a shows the CLs before gouging avoidance (the left image shows the cutter postures and the right one shows the view from the back of the surface patch), from which we can see that there was some interference between the cutter and the designed surface. Fig. 8b shows the CLs after gouging avoidance, from which we can see that the interference no longer existed.

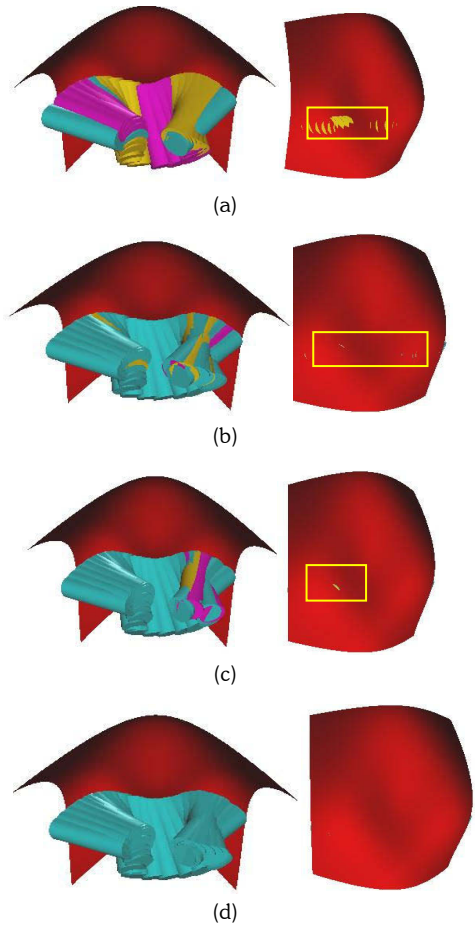
After that, the rational Bézier dual quaternion curve of cutter motion was generated and then went through the tool path verification and correction process. The cutter undergoing the initially generated rational Bézier motion is shown in Fig. 9a and some interference exists. The cutters in *yellow* indicate the CLs where there is gouging problem; while the cutters in *pink* indicate the CLs with fitting problem. Fig. 9b and 9c show the resulted tool path after the first and second modification of the rational Bézier dual quaternion curve. The gouging and fitting problem abated as opposed to that in Fig. 9a. The tool path after the third modification is shown in Fig. 9d, in which we can see that neither gouging nor fitting problems were detected and, therefore, the whole tool path was generated. The final tool path has 24 CLs that are used to construct the rational Bézier motion of the cutter.



(a)

(b)

Fig. 8. The cutter locations: (a) before gouging avoidance; (b) after gouging avoidance



(a)

(b)

(c)

(d)

Fig. 9. The result of the cutter undergoing the rational Bézier motion
(a) The initial tool path; (b) The 1st modified tool path;(c) The 2nd modified tool path; (d) The 3rd modified tool path

6. CONCLUSIONS

In this paper, a new method to generate 5-axis finishing tool path for sculptured surface is presented. The method employs the recently developed rational Bézier motions for NC tool path generation. Details on dual quaternion representation of a spatial displacement and the mathematic form of cutter bottom circle undergoing the rational Bézier motion are described. Critical issues in 5-axis tool path generation such as gouging avoidance and accuracy requirement are discussed. The major advantages of rational Bézier motions for tool path representation include (a) the entire tool path can be represented using a much more compact set of control positions of the motion as opposed to a huge data set of discrete cutter positions; (b) Since the cutter motion representation is analytic, it provides an exact representation of effective cutting shape so that tool path verification can be carried out accurately. An algorithm for tool path generation, verification, and modification has been developed. Compared with conventional 5-axis tool path generation algorithms, our algorithms has the following advantages: (a) fewer CC points are involved thus less computation required; (b) since the tool path is represented analytically, complete elimination of interference and surface accuracy, to a large extent, can be guaranteed between neighbouring CLs. Furthermore, in our method, the effective cutter shape used for the further generation of the cutter locations in the next tool path can be represented exactly. This could lead to an accurate computation of scallop height between two neighbouring tool paths. Currently, study on generating neighbouring tool path based on constant scallop height is underway.

7. REFERENCES

[1] Vickers, G. W. and Quan, K. W., Ball-mills versus end-mills for curved surface machining, *Journal of Engineering for Industry*, Vol. 11, 1989, pp 22–26.

[2] Dragomatz, D. and Mann, S., A classified bibliography of literature on NC milling path generation, *Computer-Aided Design*, Vol. 29, No. 3, 1998, pp 239-247.

[3] Choi, B. K. and Jerard, R. B., *Sculptured surface machining*, Dordrecht: Kluwer Academic, 1998.

[4] Elger, G. and Cohen, E., Tool path generation for freeform surface models, *Computer-Aided Design*, Vol. 26, No. 6, 1994, pp 490-496.

[5] Rao, N., Bedi, S. and Buchal, R., Implementation of the principal-axis method for machining of complex surface, *International Journal of Advanced Manufacturing Technology*, Vol. 11, 1996, pp 249-257.

[6] Suresh, K. and Yang, D. C. H., Constant scallop height machining of free-form surfaces. *ASME Journal of Engineering for Industry*, Vol. 116, No. 5, 1994, pp 253-259.

[7] Lo, C. C., Efficient cutter-path planning for five-axis surface machining with a flat-end cutter, *Computer-Aided Design*, Vol. 31, No. 6, 1993, pp 557–566.

[8] Lee, Y. S. and Chang, T. C., Two-phase approach to global tool interference avoidance in 5-axis machining, *Computer-Aided Design*, Vol. 27, No. 27, 1995, pp 715–29.

[9] Lee, Y. S., Non-isoparametric tool path planning by machining strip evaluation for 5-axis sculptured surface machining, *Computer-Aided Design*, Vol. 30, No. 7,1998, pp 559–570.

[10] Sarma, R., Flat-ended tool swept sections for five-axis NC machining of sculptured surfaces, *ASME Journal of Manufacturing Science and Engineering*. Vol. 122, No. 2, 2000, pp 158-165.

[11] Choi, B. K., Park, J. W., and Jun, C. S., Cutter-location data optimisation in 5-axis surface machining, *Computer-Aided Design*, Vol. 25, No. 6, 1993, pp 377–386.

[12] Rao, A. and Sarma, R., On local gouging in five-axis sculptured surface machining using flat-end tools, *Computer-Aided Design*, Vol. 32, No. 5, 2000, pp 409–420.

[13] Li, S. and Jerard, R. B., 5-Axis machining of sculptured surfaces with a flat end cutter, *Computer-Aided Design*, Vol. 26, No. 3, 1994, pp 165–178.

[14] Pi, J., Red, W. E., and Jensen, C. G., Grind-free tool path generation for five-axis surface machining, *Computer Integrated Manufacturing System*, Vol. 11, No. 4, 1998, pp 337-350.

[15] Lee, Y. S., Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining, *Computer-Aided Design*, Vol. 29, No. 7, 1997, pp 507–521.

[16] Rao, N., Ismail, F., and Bedi, S., Tool path planning for five-axis machining using the principle axis method, *International Journal of Machine Tools and Manufacture*, Vol. 37, No. 7, 1997, pp 1025–1040.

[17] Wang, W. and Joe, B., Robust computation of the RMF for swept surface modeling, *Computer-Aided Design*, Vol. 29, 1997, pp 379-391.

[18] Juttler, B. and Wagner, M. G., Computer-aided design with spatial rational B-spline motions, *ASME Journal of Mechanical Design*, Vol. 118, No. 2, 1996, pp 193-201.

[19] Juttler, B. and Wagner, M. G., Rational motion-based surface generation, *Computer-Aided Design*, Vol. 31, 1999, pp 203-213.

[20] Ge, Q. J. and Srinivasan, L. N., Fine tuning of rational *B*-Spline motions, *ASME Journal of Mechanical Design*, Vol. 120, No. 3, 1998, pp 46-51.

[21] Xia, J. and Ge, Q. J., On the exact representation of the boundary surfaces of the swept surface undergoing rational Bézier and *B*-spline motions, *Proc. 1999 ASME Design Automation Conf, Las Vegas*, DETC99/DAC-8607.
[22] Xia, J. and Ge, Q. J., An exact representation of effective cutting shapes of 5-axis CNC machining using rational Bézier and *B*-spline tool motions, *Proc. 2001 IEEE ICRA*, Korea, pp 342-347.
[23] Zhang, W., Five-axis tool path generation using piecewise rational Bézier motions of a flat-end cutter, *M.Eng Thesis*, National University of Singapore, 2003.
[24] Faux, I. D. and Pratt, M. J., *Computational geometry for design and manufacturing*, New York: Wiley, 1981.
[25] Nelder, J. A. and Mead, R., A Simplex Method for function minimization, *Computer Journal*, Vol. 7, 1965, pp 308–313.
[26] Piegl, L. and Tiller, W., *The NURBS book*. Springer-Verlag, 1995.