Journal of Advanced Manufacturing Technology manuscript No.
(will be inserted by the editor)

Tool Positioning in Five Axis Machining on a Tensor Product Surface using Circular Rays

Mukhmeet Singh · Sanjeev Bedi · Stephen Mann

Received: date / Accepted: date

Abstract In this paper, we give a method for positioning a toroidal tool on a tensor product surface with two points of contact. By firing vertical rays and circular rays from the tensor product surface to the toroidal cutting surface, we can find these two points of contact by solving only quadratic equations, whereas previous methods typically solve transcendental equations using iterative methods. The new method is faster than the methods presented in literature and is independent of the method used to define the part surface. In addition to general applicability, test results on three sample surfaces are presented.

Keywords CNC Machining · five-axis machining · multi-point machining · Bézier surface · Circular Rays

1 Introduction

Surface finishing and machining time are inversely proportional to each other: To get better surface finish, a machining toolpath must be fine and closely packed with a smaller side and forward step, which leads to higher machining time and lower material removal rate. To reduce the machining time and increase the material removal rate, the tool can be placed in close proximity to the surface and remove the material using wider strips, which allows for a larger side step. See Li and Jerard's paper for a more in-depth introduction to 5-axis machining and issues with 5-axis machining [1].

In the dies and molds industries tensor product surfaces are used, which can be machined using wider strips on a 5-axis CNC machining center [2–5]. Liang et al. [6] survey a variety of 5-axis techniques; in this paper, we develop what they refer to as a multi-point machining method. One method not mentioned in the Liang et al. survey is that of Kim et al. [8], where they compute the offset of the design surface; then finding a tool position with two points of contact with the design surface reduces to finding orientations of a circle that has two points of tangency with the offset surface; finding such points of tangency requires solving systems of equations; see also Fan and Ball's paper for finding tool orientations with two points of contact between a circle and the design surface [5]. The radiused end mill, also known as toroidal cutter, is used for the toolpath generation with these methods. A toroidal cutter provides suitable variations in

M. Singh
University of Waterloo, Waterloo, ON CANADA
E-mail: mukhmeet.singh@uwaterloo.ca

S. Bedi
University of Waterloo, Waterloo, ON, CANADA
Tel.: +1 519 888 4567 x32178
E-mail: sanjeev.bedi@uwaterloo.ca

Stephen Mann
University of Waterloo, Waterloo, ON, CANADA
Tel.: +1 519 888 4567 x34526
Fax: +1 519 885-1208
E-mail: smann@uwaterloo.ca

curvature that can be matched with the surface geometry, and also gives a generalized tool profile from which other tools can be derived [7].

Multipoint methods give gouge free tool positions with two point of contact but typically require solving complex, non-linear, transcendental equations for tool path generation. Duvedi et al. [10] gave a numerical method for solving these higher order, non-linear, transcendental equations using Newton's method. Newton's method helps to make calculations easy, but it leads to higher computational time; further, sometimes Newton's method did not converge and gave incorrect solutions.

Newton's method helps to solve the higher order non-linear transcendental equations numerically, but gouge checking is still required to ensure that the tool position found is not overcutting. This additional checking for gouging adds to the computational time along with the time taken by the Newton's method for converging to a solution.

Singh et al. [11] omitted the step of using Newton's method and found the first point of contact by just doing a form of gouge checking by firing rays from the tool to the design surface. Singh et al. then did a similar form of gouge checking/ray casting, using binary search to find the second point of contact. However, the ray casting method was based on Newton's method.

In this work a method is developed that is purely based on gouge checking (ray casting) without using Newton's method to position the tool over the Bézier surface, and without using binary search to find the second point of contact. Our method finds the first point of intersection by firing rays from the design surface to the torus (i.e., in the opposite direction of Singh et al. [11]), and then finds the second point of contact by firing "circular rays" from points on the design surface to the torus. While in general, such line/circle-intersect-torus calculations require solving fourth degree equations, in both cases, we choose special rays (both linear and circular) that intersect the torus in at most two locations and result in quadratic equations rather than quartic equations. Such equations have closed form solutions and can be solved fast.

2 Background

A toolpath consists of a sequence of tool positions that describe the location and orientation of the tool over the surface, and that a tool follows to machine a gouge free desired surface. The motion of the tool between the two successive tool positions is determined by linearly interpolating the machine parameters. One approach to machining is to a *footprint* of the tool path to describe the pattern or the sequence of the tool positions that a tool follows while machining the surface. In this work, an XY parallel footprint lying in the XY-plane is used as shown in Figure 1, where the tool path passes in side step direction are separated by $Side_{step}$ and the positions of tool in the feed forward direction are separated by $Forward_{step}$. This type of footprint is a form of an *isoparametric* footprint where the parameter space is embedded in the Euclidean space; see [1] for further discussion of isopametric footprints, side step and feed forward distance.

In the dies and mold industry, sculptured surfaces are commonly defined using parametric surfaces such as Bézier Surfaces, B-Spline Surfaces and Non-uniform rational B-spline surfaces (NURBS). In this work bi-cubic Bézier surfaces are used and the mathematical model for a Bézier surface is given by

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n} P_{i,j} B_n^i(u) B_n^j(v) \tag{1}$$

for $0 \leq u, v \leq 1$, where the $P_{i,j}$ are the *control points* of the surface, n is the degree of the surface, and the $B_n^i(u)$ are the Bernstein functions given by

$$B_n^i(u) = \binom{n}{i} u^i (1-u)^{n-i}. \tag{2}$$

However, the methods developed in this work uses the surface points for the calculation of tool position and are not dependent on the type of the surface. So while we used bi-cubic Bézier surfaces for testing our algorithms, any surface could be used as the design surface.

The most common types of tools used in industry are Ball nose end mills, Flat end mills, and Radiused end mills. In this work, the tool path is generated for the radiused end mill. Geometrically, the radiused end mill can be represented as a torus, as shown in Figure 2.

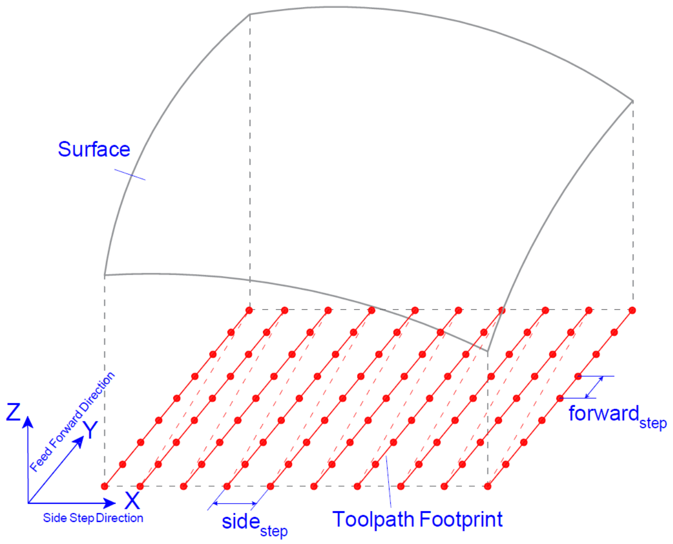


Fig. 1 Tool path footprint with Side and Forward step directions

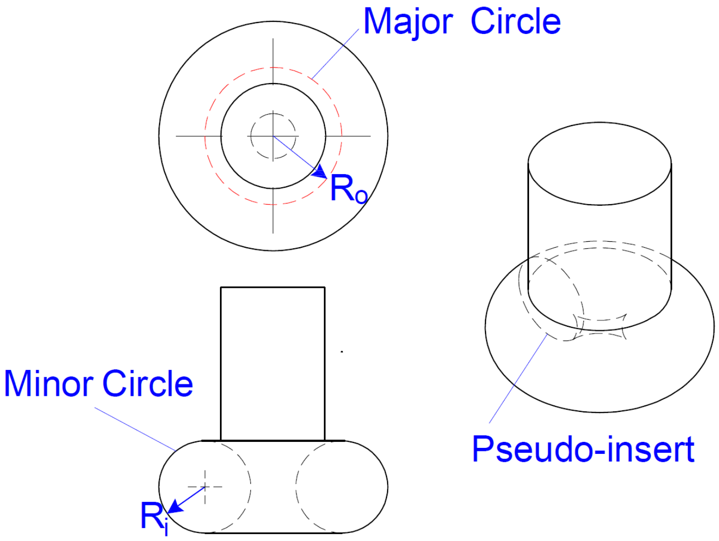


Fig. 2 Geometry of Toroidal Cutter used for Toolpath Generation Toroidal cutters can be represented as a general end mill tool [7]. The flat end mill and ball nose end mill are special cases of the toroidal cutter, with $R_i = 0$ and $R_O = 0$ respectively.

The toroidal cutter is defined with the two radii, R_i, the radius of the minor circle, and R_o, the radius of the major circle. The minor circle with radius R_i represents the *pseudo-insert* that corresponds to the cutting edge on the physical cutter. The pseudo-insert sweeps a torus around the circle with radius R_o known as the major circle.

In this paper, we will have need to distinguish between a *basis*, a minimal set of vectors that span a linear space, and a *coordinate frame*, a basis together with a point; a coordinate frame essentially plays the role of a basis for an *affine space*, with the point of the coordinate frame serving as the origin for the affine space [9].

Finally, in this paper, our focus is on finding tool orientations with two points of contact between a toriodal cutter and a relative flat design surface and limited rotation angles of the tool. We do not consider issues such as interference, and our assumptions of a relatively flat surface and limited tool

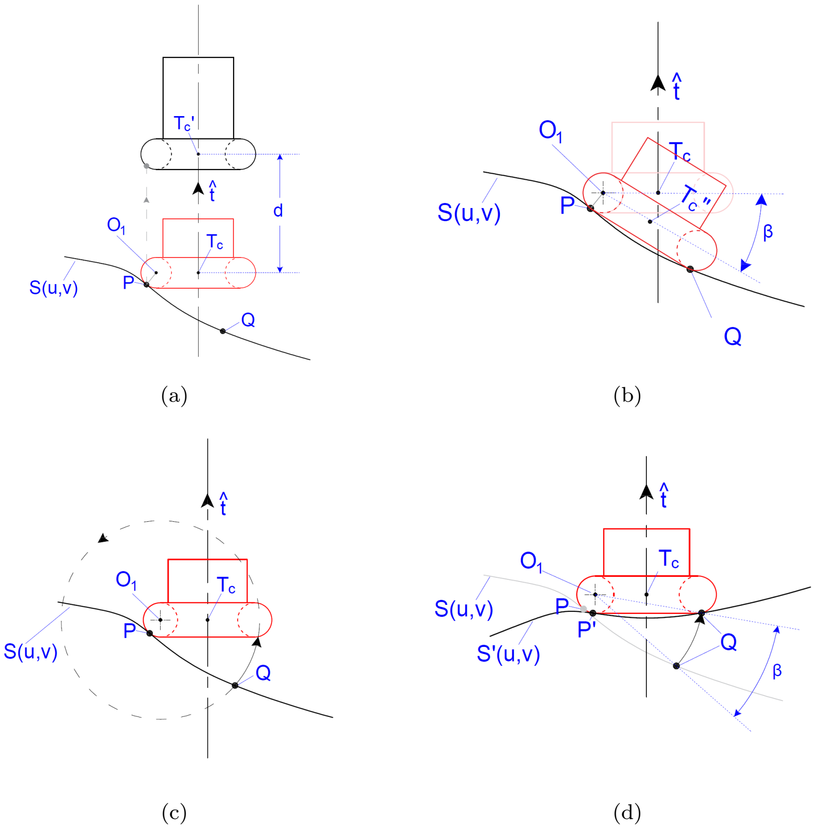


Fig. 3 Different stages shown for getting the points of contact over the Bézier surface using the vertical and circular ray firing method in comparison with tilting the tool. (a) Initial dropping of the torus to find the first point of contact. (b) Rotation of tool to find second point of contact (method used in previous approach). (c) Casting of circular rays to find the second point of contact. (d) Rotation of surface to put tool and surface in contact at two points. In (b) and (d), note that O_1 P, Q, and the circular ray from Q are not necessarily co-planar.

rotations allows us to simplify parts of the algorithm; a further discussion of these issues appears in the conclusions of this paper.

3 General Idea

In MultiPoint machining (MPM) the toroidal cutter touches the surface at two points of contact. Figure 3 shows the stage wise process of finding the two points of contact for our vertical and circular ray firing method, as well as illustrating an earlier method for comparison. The toroidal tool is placed over the Bézier surface at T'_C as the center of the tool and both the points of contact P and Q over the Bézier surface where the tool is supposed to touch the surface without any gouging are shown in the Figure 3(a). To find the first point of contact P with this method, vertical rays are fired from the surface towards the tool; compare this to the method given by Singh et al. [11], in which, the rays are fired from the toroidal surface towards the Bézier surface. After computing the drop distance d, the tool is dropped vertically by the drop distance and touches the Bézier surface at point P giving the location and position of the pseudo-insert.

To compute the second point of contact, previous approached tilted the tool about the axis of pseudo-insert until the tool touches the surface at Q, as shown in Figure 3(b). This rotation approach either requires solving non-linear transcendental equations [10], or iteratively rotating the Bézier surface about the axis of the pseudo-insert until the surface touches the toroidal cutter at Q [11].

The key idea in this paper for finding the second point of contact is to keep the tool stationary and to fire a *circular ray* (a ray traveling a circular path) from points on the surface. If the circular ray is fired from the second point of contact Q, the ray will intersect the torus at a right angle, forming an angular arc segment, as shown in Figure 3(c); the angle of this arc segment gives the tilt angle. If a circular ray is fired from any other point R on the surface, it will either miss the tool or the arc from R to the tool will be longer than the arc from the second point of contact Q and the tool. Thus, the circular ray with the shortest arc to the toroidal tool gives the tool rotation for the second point of contact.

Regardless, if we rotate the surface with the tilt angle obtained from the angular arc segment constructed from Q, this places the tool in tangency with S at a second point of contact Q, as shown in Figure 3(d); the tool positioned to have these two points of contact is a tool position used in machining. Figure 3(d) shows the arc formed by the circular ray and the associated tilt angle β, and also shows the rotated Bézier surface $S'(u,v)$ touching the toroidal tool surface at two points of contact P and Q. As the surface is rotated with the tilt angle β about the axis of the pseudo-insert, the first point of contact slides from P to P' around the pseudo-insert.

Both the steps of vertical and circular ray firing involve the solution of algebraic equations of the toroidal cutter and eliminates the use of an iterative process involving non-linear transcendental equations. As described in Section 4, although the algebraic equation for the torus is fourth degree, we only need to solve second degree equations to find the points of contact because of the particular rays (both vertical and circular) being fired. Having to solve only low degree algebraic equations makes the method simpler, faster, and more robust than the earlier approaches.

Note that, while Figure 3(c) makes it appear that O_1, P, Q, and the circular ray from Q lie in a common plane, that is not the case; instead, Q and the circular ray from Q lie in a plane parallel to the plane of the pseudo-insert centered at O_1.

4 Ray Firing Details

The algorithm in the proposed method for finding the first point of contact is based on a set of rays fired from the tensor product surface towards the the tool, and the ray that intersects the tool having the least distance traveled gives the first point of contact. The ray set is construct by discretizing the design surface finely, and the rays are fired in the tool axis direction towards the toroidal surface.

For each footprint point $F_{r,c}$, the surface $S(u,v)$ is re-expressed in the coordinate frame $\mathcal{F}_{r,c} = \{x,y,z,F_{r,c}\}$, giving $S_{F_{r,c}}(u,v)$ (alternatively, the surface could have been translated by $[-x_r,-y_c,0]$, where x_c and y_c are the xy-coordinates of the footprint point $F_{r,c}$). The toroidal surface $T(x,y,z)$ is defined algebraically to be symmetric to the z-axis relative to $\mathcal{F}_{r,c}$:

$$T(x,y,z) = \left(\sqrt{x^2+y^2} - R_o\right)^2 + (z+h)^2 - R_i^2, \tag{3}$$

where h is the height at which the torus is positioned from the xy-plane in the z-direction over the Bézier surface as shown in Figure 4.

Rays are cast from the Bézier surface $S_{F_{r,c}}(u,v)$ in the tool direction $\hat{t} = [0,0,1]^T$ at each parameter (u,v) within the defined grid range, $u_{min} \leq u_i \leq u_{max}$ and $v_{min} \leq v_j \leq v_{max}$. The corresponding intersection point on the toroidal surface with the ray can be obtained by comparing $S_{F_{r,c}}(u_i,v_j) = [S_x,S_y,S_z]$ with T. Since the ray fired is in the tool direction and the point on surface $[S_x,S_y,S_z]^t$ and the intersection point $[T_x,T_y,T_z]^t$ are collinear in the z-direction, as shown in Figure 5, the comparison of $S_{F_{r,c}}(u,v) = [S_x,S_y,S_z]^t$ and $T = [T_x,T_y,T_z]^t$ is given by

$$\begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z - d(u_i,v_j) \end{bmatrix}. \tag{4}$$

Every point within the defined range of the grid on the surface will form a drop distance $d(u,v)$,

$$d(u_i,v_j) = T_z - S_z, \tag{5}$$

where T_z can be obtained from Equation 3, Equation 4 and Equation 5 as solutions to the following equation:

$$(T_z - h)^2 + \sqrt{R_i^2 - \left(\sqrt{S_x^2+S_y^2} - R_O\right)^2} = 0. \tag{6}$$

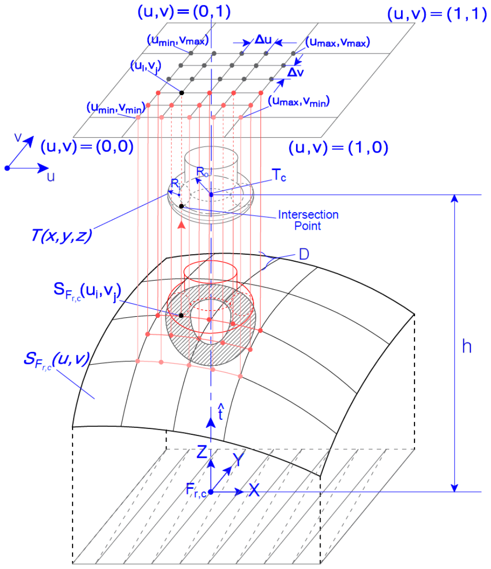

(u,v) = (0,1)
(u,v) = (1,1)
(u_{min},v_{max})
Δu
(u_{max},v_{max})
(u_i,v_j)
Δv
(u_{min},v_{min})
(u_{max},v_{min})
v
u
(u,v) = (0,0)
(u,v) = (1,0)
R_i
R_o
T_c
Intersection Point
T(x,y,z)
D
S_{F_{r,c}}(u_i,v_j)
S_{F_{r,c}}(u,v)
h
t̂
Z
Y
X
F_{r,c}

Fig. 4 Firing the rays from the Bézier surface along tool axis. A patch formed under the shadow of the tool and drop distance set D obtained by intersection of fired rays from the Bézier surface

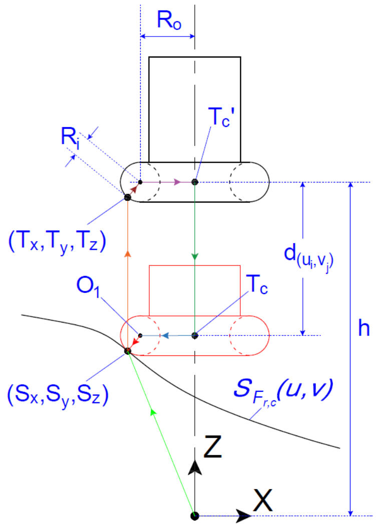

R_o
T_c'
R_i
(T_x,T_y,T_z)
d_(u_i,v_j)
O₁
T_c
h
(S_x,S_y,S_z)
S_{F_{r,c}}(u,v)
Z
X

Fig. 5 Ray fired from Bézier surface and the intersection point on torus showing the collinearity in the XZ plane

While normally the intersection of a ray with a torus would result in a fourth degree equation in the unknowns, the use of a canonical torus and rays parallel to the axis of the torus results in Equation 6 being only quadratic in the unknown T_z, since x, y of Equation 3 (which normally become fourth degree to get rid of the radical) are fixed as S_x, S_y.

The two solutions for T_z to Equation 6 indicate intersections of the ray with the upper surface of the torus and with the lower surface of the torus. As the torus is symmetric about the z-axis and the xy-plane, the lower surface of the torus will make the contact with Bézier surface. Hence the drop distances obtained by the intersection points of fired rays with the lower surface of torus form the drop distance set $\{D\}$. The surface point corresponding to the minimum value of the drop distance d_{min} of the drop distance set $\{D\}$ yields the first point of contact, that the toroidal cutter will make with the Bézier surface.

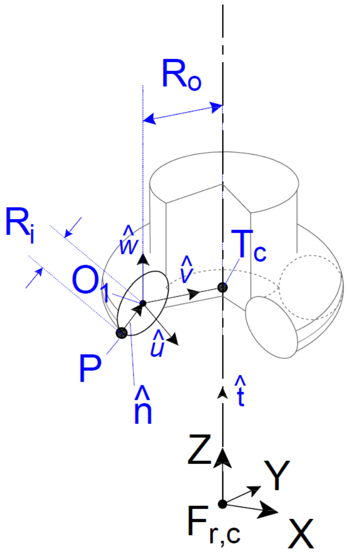


Fig. 6 A new coordinate frame $\mathcal{F} = \{\hat{u}, \hat{v}, \hat{w}, O_1\}$ is generated at the center of the pseudo-insert.

4.1 Circular Ray Intersection for Tilt

Having found one point of contact between the toroidal tool and the design surface, we will find a second point of contact by rotating the tool so as to maintain tangency at the first point of contact, giving us a tool orientation that has two points of contact between the toroidal tool and the design surface.

The first point of contact P lies on one of the pseudo-inserts, as shown in Figure 6. The insert in the physical tool is of the same size of the minor circle with radius R_i. Any rotation about the axis of this pseudo-insert maintains the contact between tool and surface at P. To achieve the rotation about the axis of the pseudo-insert, a basis $\{\hat{u}, \hat{v}, \hat{w}\}$ is defined at the center of the pseudo-insert O_1:

$$\begin{aligned} \hat{w} &= \hat{t} \\ \hat{u} &= \frac{\hat{w} \times (P - T_C)}{|\hat{w} \times (P - T_C)|} \\ \hat{v} &= \hat{w} \times \hat{u}, \end{aligned}$$

where T_C is the tool center. The center of pseudo-insert O_1 is

$$O_1 = T_C - R_o \hat{v}.$$

To obtain the second point of the contact and tilt angle β, circular rays with the circle center on the axis of the pseudo-insert are fired from the surface and the intersection of each ray with the toroidal surface is found. The ray that intersects the torus with the least angle traveled gives the second point of contact and the corresponding angle of arc formed by the circular rays gives the tilt angle required to tilt the tool for machining the surface with the tool touching the surface at multiple points without any gouging.

A plane spanned by $\hat{v}, \hat{w}$ is created at a distance of in S_u in the $\hat{u}$-direction from the origin (O_1). Figure 7 depicts the casting of a circular ray from a point on surface from a plane parallel to the plane spanned by $\hat{v}, \hat{w}$. Note that this circle (lying in a plane parallel to the pseudo-insert and centered at a point on the line through the center of the pseudo-insert in the direction of the normal to the pseudo-insert) intersects the torus in 0, 1, or 2 points, where in the case of two points of intersection neither intersection point is a double root.

The surface point $S(u, v)$ in the coordinate frame $\mathcal{F} = \{\hat{u}, \hat{v}, \hat{w}, O_1\}$ has coordinates

$$S_F(u, v) = S_F = \begin{bmatrix} S_u \\ S_v \\ S_w \\ 1 \end{bmatrix}. \tag{7}$$

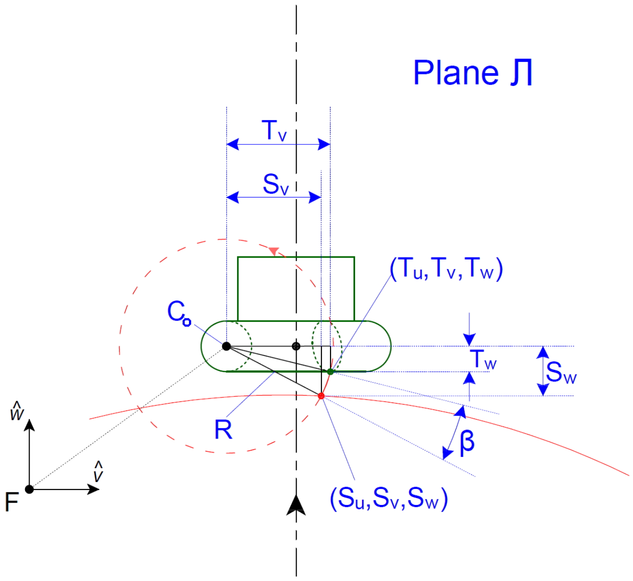


Fig. 7 Planar view of the circular ray (red, dashed circle) fired and giving the required tilt angle

Let L be the line through O_1 in direction $\hat{u}$. We now wish to find a point on the torus, $T = [T_u, T_v, T_w]$, where T is the intersection of the torus with a circle C of radius R that lies in a plane Π parallel to the plane of the pseudo-insert, where Π contains S, and the center of C is the intersection of L and Π.

To find the intersection between the circle C and our torus, we redefine the torus in the coordinate frame $\mathcal{F}$, with an offset of R_O in the $\hat{v}$-direction:

$$T_F(u, v, w) = \left(\sqrt{u^2 + (v - R_O)^2} - R_O\right)^2 + w^2 - R_i^2 = 0. \tag{8}$$

A plane Π parallel to the vw-plane is created at S, which intersects the axis in the u-direction a distance S_u from O_1. In Π, as shown in Figure 7, a circle is created with radius R and its center C_o lying on the axis, i.e., $C_o = [S_u, 0, 0, 1]^t$ relative to the frame $\mathcal{F}$. The implicit equation of the circle C in plane Π relative to coordinate frame $\mathcal{F}$ is given by

$$R^2 = v^2 + w^2. \tag{9}$$

The value of R can be computed from the equation of the circle at the point of intersection with the surface $S(u, v)$:

$$R^2 = S_v^2 + S_w^2. \tag{10}$$

As the circular ray and the point of intersection both lie in a plane parallel to the vw-plane at S_u, the coordinate of the torus at the point of intersection in the u-direction is same as the S_u:

$$T_u = S_u. \tag{11}$$

We wish to find u, v, w such that Equations 8 and 9 are satisfied where $u = S_u$. From Equation 9, we can replace w^2 in Equation 8 with $R^2 - v^2$, giving

$$\begin{aligned} 0 &= \left(\sqrt{S_u^2 + (v - R_O)^2} - R_O\right)^2 + R^2 - v^2 - R_i^2 \\ &= S_u^2 + v^2 - 2vR_O + R_0^2 \;\; - 2\sqrt{S_u^2 + (v - R_O)^2} + R_O^2 + R^2 - v^2 - R_i^2 \end{aligned} \tag{12}$$

The v^2 terms cancel from this equation; taking the radical to one side of the equation and squaring gives an equation that is quadratic in v, our only unknown:

$$8R_O^2\, v^2 - (4aR_O - 8R_O^3)\, v + (a^2 - 4R_O^2[S_u + R_O^2]) = 0, \tag{13}$$

where

$$a = S_u^2 + 2R_O^2 + R^2 - R_i^2.$$

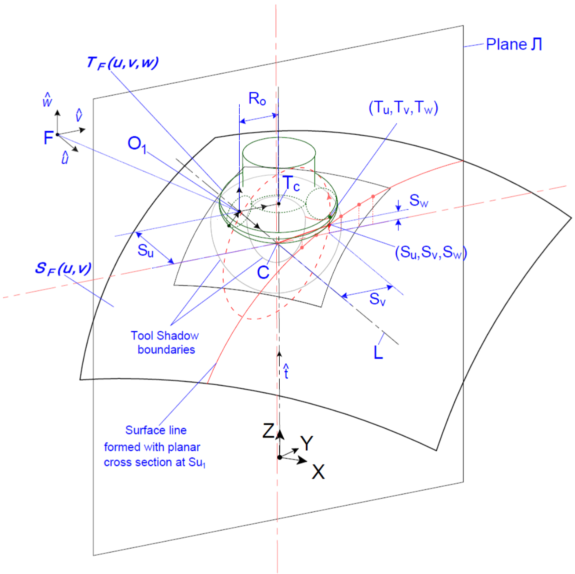


Fig. 8 Circular ray fired from the surface points in the plane Π, intersecting the torus defined in the coordinate frame $\mathcal{F}$.

The roots of Equation 13 give us our solutions for T_v, with Equation 9 giving us T_w and with T_u already known to be $T_u = S_u$. This gives the intersection of the torus and the circle.

Note that it is the cancellation of the v^2 terms in Equation 12 that results in Equation 13 being a quadratic equation rather than a quartic equation, since squaring to get rid of the radical in Equation 12 would result in this v^2 term becoming a v^4 term.

The points $S = (S_u, S_v, S_w)$ and $T = (T_u, T_v, T_w)$ lie on the circle centered at O_1 in the plane Π, of radius R, as shown in Figure 7. The angle β between S and T on this circle is

$$\beta = \tan^{-1}\left(\frac{T_w}{T_v}\right) - \tan^{-1}\left(\frac{S_w}{S_v}\right). \tag{14}$$

Figure 8 shows the three dimensional view of the casting of a circular ray from the surface point (S_u, S_v, S_w) that intersects the torus at (T_u, T_v, T_w).

A grid of surface points is formed beneath the shadow of the tool, similar to the grid formed in the previous section with the limits (u_{min}, u_{max}) and (v_{min}, v_{max}) discretized by $\Delta u, \Delta v$, respectively. Firing of circular rays from all the surface points on the defined grid forms a set of angular arcs $\{\beta\}$. The minimum angle of the set β_{min} gives the required angle for tilting the tool for the machining to be gouge free with two points of contact; and the corresponding values of parameter $[u, v]$ gives the location of the second point of contact. Note that the associated circular ray fired with the minimum tilt angle β_{min} intersects the torus perpendicularly with the tangent formed on the toroidal surface at the point of intersection.

5 Implementation and Verification

We implemented the vertical ray casting algorithm and the circular ray algorithm in C++. Since the tool can be positioned anywhere over the surface, firing the rays from some points $S(u, v)$ will not intersect the tool surface. In particular, rays yielding no intersection with the toroidal surface will give a complex solution to Equation 6 or 13 and can be ignored. To limit the number of rays that do not intersect the tool, the surface patch under the shadow of the tool is found and then rays are cast from the discretized points of that particular patch instead of casting rays from the whole surface. Over the shadow of the tool a parametric grid with the limits (u_{min}, u_{max}) and (v_{min}, v_{max}) is created which is then discretized with δu and δv, as shown in Figure 4. Additional implementation details can be found in [18].

We tested the algorithms on three bi-cubic Bézier surfaces used in another paper [17] to allow for comparison to that work. These surfaces include a convex surface, a concave surface and a saddle surface

Table 1 z-coordinates (in mm) for the three Bi-cubic Bézier Surfaces used for testing the Algorithms

Surfaces	P_{i0}	P_{i1}	P_{i2}	P_{i3}
Convex	80	90	90	80
	90	105	105	90
	90	105	105	90
	80	90	90	80
Concave	80	70	70	80
	70	55	55	70
	70	55	55	70
	80	70	70	80
Saddle	80	65	90	85
	80	85	90	95
	100	105	95	100
	90	100	100	85

Table 2 Drop distance comparison at three x-cross-sections, v parameter (y) in range 0.2 to 0.8. Differences in millimeters. Depth of the parts is approximately 10mm.

Cross Section	Convex		Concave		Saddle	
X in mm	Max	Avg	Max	Avg	Max	Avg
36	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
72	0.00015	< 0.0001	0.001	< 0.0001	0.0002	< 0.0001
108	< 0.0001	< 0.0001	< 0.0001	< 0.0001	0.0003	< 0.0001

Table 3 First point of contact comparison at three x-cross-sections, v parameter (y) in range 0.2 to 0.8. Differences in millimeters. Depth of the parts is approximately 10mm.

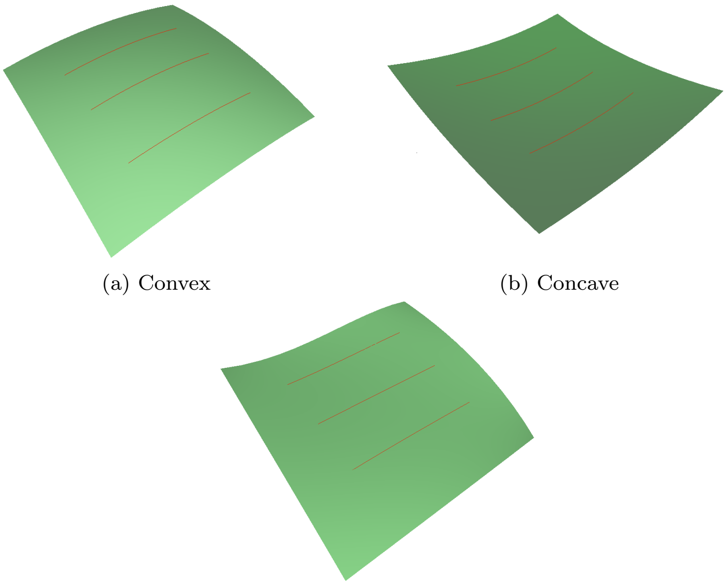
Cross Section	Convex		Concave		Saddle	
X in mm	Max	Avg	Max	Avg	Max	Avg
36	0.00860	0.00258	0.0116	0.00364	0.00581	0.00183
72	0.796	0.0324	0.304	0.0240	0.0934	0.00605
108	0.00798	0.00288	0.130	0.00688	0.173	0.00864

having both concave and convex regions. All three surfaces have a span of 150 × $150mm^2$ in the XY-plane. The x- and y-coordinates of the control points are uniformly distributed with a span interval of 50 starting from the origin. The z-coordinates for the three bi-cubic Bézier surfaces are given in Table 1. The three test surfaces are shown in Figure 14.

In [17], the generated toolpath data was verified by first simulating the toolpath data using Tool-Sim [15], and then physically machining the three surface on aluminum stock using the same tool path with a DMU-80P Hi-Dyn tilt-rotary simultaneous five axis machining center. In the tests run in [17], the maximum error between the simulated machined surfaces and the corresponding designed surfaces was 0.01mm, while the measured error between the actual machined surfaces and the design surfaces was 0.05mm.

The algorithm proposed in this paper should give identical tool positions (with some numerical differences) to the algorithm given by [17] when both methods use the same toolpath footprint. The footprint used in [17] is the XY-parallel toolpath shown in Figure 1, having 10 passes in total with the first 9 passes separated by $Side_{step} = 18mm$ and a tenth pass at $x = 150mm$. Each pass contains 78 tool positions separated by $Forward_{step} = 2.0mm$, out of which the first and last tool position of the pass is used to lift the tool to avoid gouging on the surface as the tool shifts from one pass to the next. The algorithm runs for 760 effective tool positions that physically make contact with the surface to be machined. The toroidal cutter used for machining had a major radius $R_o = 6.7mm$ and minor radius $R_i = 6.0mm$.

We compared the tool positions generated by our algorithm and the algorithm in [17] (DRD) on this toolpath, looking at the the drop distance for the first point of contact, the first point of contact itself, and the tool axis tilt angle. The maximum error of DRD between the simulated machined surface and the design surface on these three examples was 0.01mm. (The maximum error of DRD between the measured machined surface and the design surface on these three examples was 0.05mm, but since the



(a) Convex

(b) Concave

(c) Saddle

Fig. 9 Cubic Surface used to test the algorithms. Red curves are the tool positions used to compare to previous algorithms.

Table 4 Tilt angle comparison at three x-cross-sections, v parameter (y) in range 0.2 to 0.8. Angle differences between old and new (in degrees)

Cross Section	Convex		Concave		Saddle	
X in mm	Max	Avg	Max	Avg	Max	Avg
36	0.0904	0.0516	0.332	0.101	0.235	0.137
72	0.323	0.0920	1.48	0.165	0.421	0.181
108	0.0960	0.0497	1.48	0.254	0.363	0.161

tool positions of the two methods were similar, we did not machine a new set of pieces using the tool paths generated by our new algorithm.)

Note that while the footprint of the DRD and our algorithm is the same, there were significant differences in how the two methods handled tool positions close to the boundaries of the design surfaces. These differences resulted in significantly different tool positions. So for our comparisons, we only compared the tool positions at footprint points on the interior of the surfaces, with the red curves in Figure 9 indicating the footprint points at which we compared the tool positions of the two method.

At all tool positions, the difference in drop distances (given in Table 2) is always less than 0.0003mm. However, while the first point of contact differences (Table 3) and the tilt angle differences (Table 4) were normally small (0.01mm and less than 0.1 degrees), at some tool locations, the first points of contact were as far apart at 0.3mm, and the differences in the tilt angles were as high as 1.48 degrees.

Looking more closely at the tool positions where the first point of contacts are far apart and the tool positions where the tilt angle is large, the larger distance in first point of contacts occurred at the same location as the larger tilt angles. What is happening at these points is that when the tool is in contact with the design surface, a relatively large portion of the tool is close to the surface (as an extreme case, think of dropping the toroidal tool on to a plane that is perpendicular to the tool axis; a circle of points on the tool will be in contact with the surface, with a zone around that circle being a zone in which the tool and surface are numerically close). In such cases, numerical differences between the two methods results in a first point of contact (and tilt angle) that are relatively large, even though the surface machined by the moving tool in the two methods is essentially the same.

Thus, away from the boundaries, the tool positions computed by the method described in this paper are the same (up to numerical differences) as those computed in [17]. However, the new method is computationally faster. Table 5 gives timing information, comparing our VCRF with four other methods: the Tool-tilt Direct method of Ravinder et al. [10]; the Tool-tilt bisection method of Ravinder et al. [13];

	Direct	Bisection	SR	DRD	**VCRF**
Convex	27	7.0	498	180	10
Concave	2.1	6.1	476	256	9
Saddle	104	2.1	491	193	10

Table 5 Timing comparison of new method (VCRF) to previous work. All values are in seconds.

the Surface Rotation (SR) method of Ravinder et al. [16], and the the Drop, Rotate, and Drop (DRD) method of Singh [18]. All timings are on bicubic surfaces, with the Tool-tilt Direct and Bisection methods being run on a 2.7GHz machine, while the DRD and VCRF methods being run on a 2.8GHz machine.

The significantly faster times for the Direct and Bisection methods (compared to the Surface Rotation and Drop, Rotate, Drop methods) are because the Direct and Bisection methods assume the xy-values of the control points are linear functions of the uv-domain values; while this assumption simplifies the search for a solution, it also restricts the types of surfaces that can be machined. The other three methods (SR, DRD, VCRF) can produce tool paths for a more general class of surfaces that are required by industry.

The key observation is that with the VCRF method, we obtain times similar to Direct and Bisection methods while being able to create tool paths for a more general class of surfaces.

6 Discussion

In our examples, we used fairly flat surfaces and had somewhat limited rotations. Such restrictions match the types of surfaces and machining done in some industries. For example in a ship propellor, the blades have a curved surface that is flat with slight curvature. Similarly, the body panels of a automobile also share the same characteristics. For such surfaces, the uniform sampling in parameter space that we used to determine the origins of the linear and circular rays worked well. However, with larger rotation angles or with surfaces with more variation, a more sophisticated scheme to determine the origins of the rays would be required.

Similarly, while our restriction of the origin of the rays to being in the “shadow of the tool” works well with the relatively flat surfaces that we considered, for surfaces with more variation and with larger rotation angles, a different approach to limiting the region used for the origin of the rays would be required.

Further, while our method obtains two points of contact at each tool position, to maximize material removal, ideally these two points of contact would be located far from one another on the tool, and the line between these two points would be close to perpendicular to the direction of motion of the tool. However, as seen in Figure 10, the two points of contact computed with our method do not always have these properties.

The left column of Figure 10 shows three passes in each of the three surface. For each pass, the first and the second points of contact are shown in a different color. Ideally, the line connecting the first and second points of contact in this figure would be close to horizontal (to make it perpendicular to the direction of the tool motion). Looking at the start and ends of each of the nine passes shown in the figure, we see that while sometimes this line between the first and second points of contact is reasonably close to horizontal (saddle surface, $x = 36$mm), at other times this line is close to vertical (concave survface, $x = 72$mm).

Also, the change in the gap between the points of contact is visibly obvious in the left column of Figure 10. The distance between the two contact points is computed and is shown as a histogram on the right of Figure 10. While the spacing changes with the surface, numerous tool positions have the two points of contact in close proximity to one another.

Thus, the fixed tool footprint that we used in this paper leads to sub-optimal solutions. An alternative approach is to treat the contact path as an unknown. We chose the simpler, fixed toolpath footprint to better illustrate our tool drop ideas, but using a non-isoparametric toolpath that obtains two points of contact perpendicular to the direction of motion should give a more optimal result, a topic that we are currently investigating. For a further discussion of known verses computed contact paths, the interested reader should see, for example, the work of Barton et al. [12].

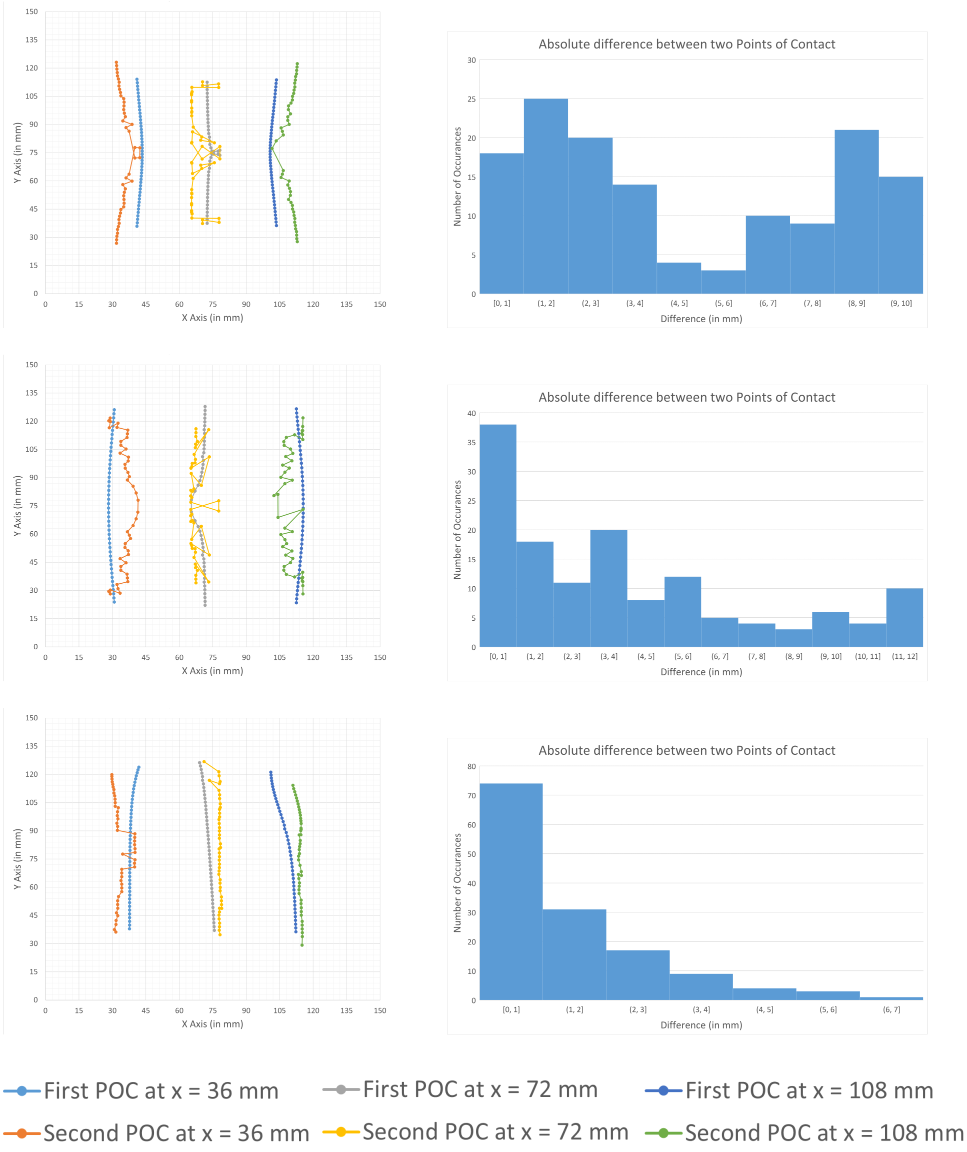


Fig. 10 Location of points of contact (left) and histograms of distances between points of contact. Top row: convex surface; middle row: concave surface; bottom row: saddle surface.

7 Conclusions

In this work, we developed a Vertical and Circular Ray Firing (VCRF) method for tool path generation, which we implemented and tested. Three bi-cubic Bézier surfaces, consisting of concave, convex and saddle regions, were used for the toolpath generation and machined on DMU-80P Hi-Dyn tilt-rotary simultaneous five axis machining center using toroidal cutter.

The VCRF method eliminates the use of Newton's method for the convergence of the solution. The VCRF method uses the implicit equations of the toroidal surface defined in Cartesian coordinate frames located at either toolpath footprint points or at the center of pseudo-inserts. In this method, vertical rays are fired from the surface towards the toroidal cutter and the ray intersecting the toroidal surface of the tool traveling the shortest distance gives the location of the first point of contact. After getting the first point of contact, then circular rays are fired from the surface. The circular ray that intersects with the torus traveling the shortest angle gives the required tilt angle and the associated second point of contact.

Tool paths were generated by implementing the VCRF algorithm in C++, generating tool paths for three surfaces. These toolpaths were compared to those of an earlier method and found to match to within numerical tolerances, but the new method computed the toolpaths two orders of magnitude faster than earlier equally general algorithms, and at similar speeds to some more specialized algorithms.

8 Declarations

8.1 Funding

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

8.2 Conflicts of interest/Competing interests

The authors have no conflicts or competing interests.

8.3 Availability of data and material

Available on request.

8.4 Code availability

Available on request.

8.5 Authors' contributions

The authors contributed the work in this article.

8.6 Additional declarations for articles in life science journals that report the results of studies involving humans and/or animals

Not applicable

8.7 Ethics approval (include appropriate approvals or waivers)

Not applicable

8.8 Consent to participate (include appropriate statements)

Not applicable

References

1. S. X. Li and R. B. Jerard, "5-axis machining of sculptured surfaces with a flat-end cutter," Comput. Aided Des., vol. 46, no 3, pp. 165–178, 1994.
2. A. Lasemi, D. Xue, and P. Gu, "Recent development in CNC machining of freeform surfaces: A state-of-the-art review," Comput. Des., vol. 42, pp. 641–654, 2010.
3. K. Patel, G. Salas Bolaños, R. Bassi, and S. Bedi, "Optimal tool shape selection based on surface geometry for three-axis CNC machining," Int. J. Adv. Manuf. Technol., vol. 57, no. 5–8, pp. 655–670, 2011.
4. D. Roth, F. Ismail, and S. Bedi, "Mechanistic modelling of the milling process using complex tool geometry," Int. J. Adv. Manuf. Technol., vol. 25, no. 1–2, pp. 140–144, 2005.

5. J. Liang and A. Ball, “Flat-end cutter orientation on a quadric in five axis machining,” Comput. Aid. Des., vol 53, pg 126-138, 2014.
6. F. Liang, C. Kang, and F. Fang, “A review on tool orientation planning in multi-axis machining, International Journal of Production Research,” Int. J. Prod. Review, online, pp. 1-31, 2020.
7. S. Bedi, F. Ismail, M. J. Mahjoob, and Y. Chen, “Toroidal versus ball nose and flat bottom end mills,” Int. J. Adv. Manuf. Technol., vol. 13, no. 5, pp. 326–332, 1997.
8. Y-J. Kim, G. Elber, M. Barton, H. Pottmann, “Precise gouging-free tool orientations for 5-axis CNC machining,” Computer-Aided Design, vol. 58, no. 1, pp 220-229, 2015.
9. R. Goldman, “An Integrated Introduction to Computer Graphics and Geometric Modeling,” CRC Press, 2009.
10. R. K. Duvedi, S. Bedi, and S. Mann, “Numerical implementation of drop and tilt method of five-axis tool positioning for tensor product surfaces,” Int. J. Adv. Manuf. Technol., vol. 95, pp. 219–232, 2018.
11. M. Singh, J. Qu, R. K. Duvedi, S. Bedi, and S. Mann, “A new approach to find gouge free tool positions for a toroidal cutter for Beézier surfaces in Five axis machining,” in the Int. J. Adv. Manuf. Technol., Vol 117, No 9–10, pg 3053–3062. 2021.
12. M. Barton, M. Bizzarri, F. Rist, O. Sliusarenko, and H. Pottmann, “Geometry and tool motion planning for curvature adapted CNC machining,” ACM Trans. Graph., vol 40, no. 4, 2021.
13. R. K. Duvedi, S. Bedi, and S. Mann, “An efficient multipoint 5-axis tool positioning method for tensor product surfaces,” Int. J. Adv. Manuf. Technol., vol. 97, pp. 279–295, 2018.
14. R. K. Duvedi, S. Bedi, A. Batish, and S. Mann, “The edge–torus tangency problem in multipoint machining of triangulated surface models,” Int. J. Adv. Manuf. Technol., vol. 82, no. 9–12, pp. 1959–1972, 2016.
15. G. Israeli, “Software Simulation of Numerically Controlled Machining,” MSc Thesis, University of Waterloo, 2006.
16. R. K. Duvedi, M. Singh, S. Bedi, S. Mann, “Multipoint tool positioning of a toroidal end mill for five-axis machining of generalized tensor product Bézier surfaces,” Int. J. Adv. Manuf. Technol., vol. 111, pp. 495–503, 2020. DOI https://doi.org/10.1007/s00170-020-06006-1
17. M. Singh, J. Qu, R.K. Duvedi, S. Bedi, S. Mann, “A new approach to find gouge free tool positions for a toroidal cutter for Bézier surfaces in five-axis machining,” Int. J. Adv. Manuf. Technol. vol. 117, pp. 3053–3062 (2021). DOI https://doi.org/10.1007/s00170-021-07849-y
18. M. Singh, “Fast and Robust Approach to Find the Gouge-free Tool Position of the Toroidal Cutter for the Bézier Surface in Five Axis Machining,” MSc Thesis, University of Waterloo, 2020.