A new approach to find gouge free tool positions for a toroidal cutter for Bézier surfaces in five-axis machining

Mukhmeet Singh¹ · Jerry Qu¹ · Ravinder Kumar Duvedi² · Sanjeev Bedi¹ · Stephen Mann¹

Received: 4 March 2021 / Accepted: 5 August 2021 / Published online: 21 August 2021 © The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
We implement and test a multi-point machining tool positioning technique that positions the tool using only a variation on gouge checking. The result is a method that is roughly twice as fast as an earlier method that performed a numerical search to find a tool position with multiple points of contact with the design surface. A GPU implementation provides an additional factor of ten speedup. Verification of the method was done via simulation and machining and measuring physical parts.

Keywords 5-axis machining · Multi-point machining · Gouge checking

1 Introduction

Two types of machines, 3-axis and 5-axis machines, are commonly used in manufacturing industries. 5-axis machining have additional degrees of freedom in rotation and tilting about the z- and x-axes, giving extra flexibility in machine kinematics as compared to 3-axis machines. This enhanced flexibility allows the tool to be positioned in close proximity to the desired surface, which leads to the machining with wider strips in a fewer number of passes [1, 2].

Numerous methods have been proposed to position a tool with a 5-axis machine [3]. The method used in this paper is to position the tool in close proximity to the surface is to inclined the tool to make contact at least at two different points on the surface at the same time. A tool positioning method based on this attribute is known as a Multipoint Method (MPM) [4–9].

The MPM method has evolved to the Drop and Tilt method of Duvedi et al. for both triangulated surfaces [4, 5] and for tensor product Bézier surfaces [11–14]. The implementation of Duvedi et al. [14] involves dropping the toroidal tool onto the surface using Newton’s method to find a point of tangency between the tool and the surface. Since there are multiple points that satisfy the tangency equations between the toroidal tool and the surface, Duvedi et al. perform a gouge check on the resulting tool position to test if the correct solution was found; if the tool gouges the surface, Newton’s method is restarted with a new seed point to find the non-gouging point of contact.

Kim et al. [10] take another approach to finding tool positions with two points of contact with a toroidal cutter. In their approach, they offset from the design surface by the inner radius of the torus. This converts finding two points of contact between the toroidal tool and the design surface into finding two tangential points of contact between a circle and the offset surface. Their approach requires solving systems of algebraic equations.

In this work, a new technique is proposed that eliminates the Newton’s search to find the tangency between the surface and the tool; instead, we find tool positions having two points of contact with the design surface by just using a variation on the gouge detection step alone. The proposed method is implemented in C++ for bi-cubic Bézier surfaces.
The algorithm was tested by simulating three parts having concave and convex regions using the ToolSim machining simulator tool and by machining the parts on a DMU-80P five-axis machine. Our focus is on positioning the tool on the surface with two points of contact, and we do not look at related topics such as selection of tool size or global collision.

In this paper, a brief background on the tools used and a review of the literature is given in Section 2. Our method is described in detail in Section 3. Section 4 discusses testing of the method on several surfaces. The work is concluded in the Section 5 of this paper.

2 Background

To run a CNC machine, a toolpath is given to the controller of the machine. A toolpath is a sequence of tool positions that describe the location and the orientation of the tool. The machine controller moves the tool linearly from one position to the next position of the toolpath.

There are three elements that are required for generating the toolpath: (a) The definition or type of surface to be machined; (b) the geometry of the tool used; and (c) the footprint that the tool should follow.

Sculptured surfaces can be defined using parametric surfaces such as Bézier Surfaces, B-Spline Surfaces and Non-Rational B-spline surfaces. In this work bi-cubic Bézier surfaces are used, although the method should readily extend to the other two classes of surfaces. A Bézier surface is given by

\[ S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{n} P_{i,j} B^n_i(u) B^n_j(v) \]

for \(0 \leq u, v \leq 1\), where \(P_{i,j}\) are the control points of the surface, \(n\) is the degree of the surface, and \(B^n_i(u)\) are the Bernstein polynomials:

\[ B^n_i(u) = \binom{n}{i} u^i (1 - u)^{n-i}. \]

In this work, the toolpath is generated for the radiused end mill, as shown in Fig. 1. The minor circle with radius \(R_i\) represents the pseudo-insert that corresponds to the cutting edge on physical cutter. A pseudo-insert is a circle that is swept around a second circle (with radius \(R_o\) known as the major circle) to generate the torus.

The sequence for the tool positions in a toolpath is guided by a footprint laid on a plane perpendicular to the tool axis, as shown in Fig. 2. The footprint lies in the \(xy\)-plane in this work with the \(x\)-axis as the side-step direction and the \(y\)-axis as the feed forward direction. The passes in the side-step direction are separated by Side\_step and the tool positions in the feed forward direction are separated by Forward\_step interval as shown in Fig. 2.
3 The method

The tool orientation and location are determined in two mathematical steps, as proposed by Duvedi et al. [11]. The tool is first dropped on the surface to find a point of tangency between the tool and the surface, and next the surface is tilted until a second point of contact between the tool and the surface is found; the resulting position and orientation is one tool position in the toolpath. The key algorithm used in both the steps emulates the dropping of a tool along a specified tool axis. This algorithm is referred to as the tool drop algorithm. The contribution of this work is a tool drop algorithm based solely on a variation of gouge detection. The drop and tilt processes are detailed below.

3.1 Dropping the tool via ray casting

To find a first point of contact between a dropping toroidal tool and a tensor product surface, Duvedi et al. [14] start at an initial seed location on the design surface, and use Newton’s method to find a point of tangency between the tool and the surface (Fig. 3). However, there may be multiple points (at different tool drop heights) that have a point of tangency between the tool and the surface. The point of tangency that has the highest tool position is the desired one, as all the other points of tangency result in the tool gouging the surface. Thus, after finding a point of tangency, Duvedi et al. test if the tool gouges the surface. If the tool does gouge the surface, different seed points are tried until a gouge free solution is found.

Fig. 2  xy-parallel toolpath footprint used for toolpath planning

Fig. 3  From an initial seed point I, Duvedi et al. use Newton’s method to find a point of tangency P between a dropping toroidal tool and a tensor product surface. The green curve illustrates the path taken by Newton’s method from the initial seed point to the point of tangency.
The key idea in this paper is to omit the first step of Duvedi et al.’s algorithm (finding a tangency with the surface), and instead perform a variation on the gouge checking step of Duvedi et al. In particular, rather than test if a tool position gouges the surface, the algorithm proposed in this paper finely discretizes the toroidal surface of tool and fires rays from the tool surface and determines the distance to the intersection with the tensor product surface. The closest point of intersection of these gives us the drop distance where the tool first touches the surface.

The points on the torus from which the rays are cast are given by

\[ T_{tool,k} = T_1 + \hat{i}d_{l,k} + Tor(\theta_l, \phi_k) \]  

(2)

where \( \hat{i} \) is the unit vector \([0,0,1]^T \) and

\[ Tor(\theta_l, \phi_k) = [(R_o + R_i \cos \theta_l) \cos \phi_k, (R_o + R_i \cos \theta_l) \sin \phi_k, R_i \sin \theta_l]^T, \]

with \( \phi_k \) sampled uniformly in \([0, 2\pi]\) and \( \theta_l \) sampled uniformly in \([3\pi/2, 2\pi]\). Note that the tool axis and the minor circle of the torus are coplanar.

From each point \( T_{tool,k} \), a ray parallel to the tool axis is fired towards the Bézier surface. Figure 4a shows a set of rays emanating from points \( T_{tool,k} \) in a direction opposite to \( \hat{i} \). The ray from \( T_{tool,k} \) intersects the surface at \( S_{l,k} \). The projected patch of the toroidal cutter is shown as hatched area on the surface. The intersection point is the solution of

\[ Tool_{l,k} = S_{u,v} \]

(3)

The length of the ray, from the surface to the intersection of the rays with the surface, represents the distance the tool must be dropped to bring that specific surface point in contact with the tool. At each point of the surface this drop distance is different. The drop distance for all the points in the surface array shown in Fig. 4a form a drop distance set. The drop distance set \( D \) is

\[ D = \{d_{l,k}, 1 \leq n, 1 \leq k \leq m\}, \]

where \( d_{l,k} \) is a solution obtained from Eqs. 2 and 3 given by

\[ d_{l,k} = S_{u,v} - (T_1 + Tor(\theta_l, \phi_k)). \]

(4)

The idea is that at the point of intersection, the surface point and the point on the tool surface are same. The difference of both the points gives the deviation of the points and form a set of three equations, derived from Eq. 3,

\[ \begin{bmatrix} \text{Sol}_x \\ \text{Sol}_y \\ \text{Sol}_z \end{bmatrix} = \begin{bmatrix} \text{Point On Surface}_x - Tool_x \\ \text{Point On Surface}_y - Tool_y \\ \text{Point On Surface}_z - Tool_z \end{bmatrix} \]

(5)

Fig. 4 Dropping the tool over the Bézier surface along tool axis \( \hat{i} \). (a) Showing the patch formed and drop distance set \( D \) obtained by intersection of fired rays from the tool surface onto the Bézier surface. (b) Cross-sectional view of tool touching the Bézier surface at \( P \), coplanar to the pseudo-insert.
that must be solved simultaneously for three unknowns $[u, v, d_{i,k}]$, where $u$ and $v$ are the parameters for the surface point and $d_{i,k}$ is the drop distance. The goal is to find the drop distance so that $[Sol_x, Sol_y, Sol_z] = [0, 0, 0]$.

The tool axis $\hat{t}$ is $[0, 0, 1]^T$ and hence Eq. 5 simplifies to

$$J = \begin{bmatrix} S_x(u, v) - (T_1x + Tor_x(\theta_l, \phi_k)) \\ S_y(u, v) - (T_1y + Tor_y(\theta_l, \phi_k)) \\ S_z(u, v) - (T_1z + Tor_z(\theta_l, \phi_k)) \end{bmatrix}$$

(6)

In Eq. 6, only the third equation corresponding to the $z$-direction is a function of the drop distance, $d_{i,k}$. Based on this observation the solution of Eq. 6 is broken into two steps. In the first step, Newton’s method is used to find the values of parameters $u$ and $v$ using the first and second equation of Eq. 6. Given a set of non-linear equations, the iterative step in Newton’s method is

$$x_i = x_{i-1} - [J]^{-1}F(x_{i-1})$$

where $x_i = [u_i, v_i]^T$, $F_i = [Sol_x(u_i, v_i), Sol_y(u_i, v_i)]^T$, and

$$J = \begin{bmatrix} \frac{\partial Sol_x}{\partial u} & \frac{\partial Sol_x}{\partial v} \\ \frac{\partial Sol_y}{\partial u} & \frac{\partial Sol_y}{\partial v} \\ \frac{\partial Sol_z}{\partial u} & \frac{\partial Sol_z}{\partial v} \end{bmatrix}$$

(7)

$J$ is the Jacobian matrix of Eq. 5 and $x_i$ is the vector of unknowns, $u, v$. The initial value of $x_i$ is set to the solution for the previous $Tool_{i,k}$, and the subsequent values of $x_i$ are calculated iteratively until they are within a user specified tolerance. In the second step, the values of parameters $u$ and $v$ are used to solve for drop distance $d_{i,k}$ in the $z$-direction.

The minimum of the drop distance set $d_{min} = \min\{D\}$ gives the distance the tool should be dropped to just touch the surface without any gouging. Any distance greater than $d_{min}$ will result in over-cutting. The surface point $S(u_i, v_i)$ is the first point of contact. Note that the point of contact on the tool may lie on an internal (hyperbolic) point in the case of convex surfaces.

The accuracy with which this shaded area is represented depends on the discretization of the torus surface. We used adaptive discretization to obtain greater accuracy and quick convergence.

After dropping the tool, the first point of contact $P$ lies on one of the pseudo-insert, as shown in Fig. 4b. This minor circle is the same size as the insert in a physical tool and as the tool rotates the physical insert coincides with the minor circle as it touches the surface. The center of the pseudo-insert $O_1$ is given by

$$O_1 = P + \hat{n} \hat{n},$$

(8)

where $\hat{n}$ is the surface unit normal at the point of contact $S(u, v)$ given by

$$\hat{n} = \frac{\partial S(u, v)}{\partial u} \times \frac{\partial S(u, v)}{\partial v}$$

(9)

As mentioned by Duvedi et al. [5, 12, 14] a rotation about the axis of this pseudo-insert maintains the contact between tool and surface at $P$. The earlier works [5, 12] tilted the tool about this axis thereby complicating the resulting drop equations and ray firing equations as transcendental terms due to rotation come into play. In this paper, we use the approach of [14], where the surface is rotated rather than the tool to find the second point of contact and the tool orientation. The tool with the tool axis still aligned to the $z$-axis is subsequently dropped on the rotated surface. This step is repeated within a bisection based algorithm until a second point of contact is found.

### 3.2 Rotating the surface

To rotate the surface, a new coordinate frame $\{\hat{u}_1, \hat{v}_1, \hat{w}_1\}$ is created at $O_1$, the center of the pseudo-insert at $P$, the first point of contact, where

$$\hat{w}_1 = \hat{t}, \quad \hat{u}_1 = \frac{(T_c - P) \times \hat{w}_1}{|T_c - P| \times \hat{w}_1|}, \quad \hat{v}_1 = \hat{w}_1 \times \hat{u}_1$$

(10)

The surface is defined in this coordinate frame and is then rotated about $\hat{u}_1$ by an angle $\beta$ to form a new rotated surface $S'(u, v)$. This rotation leads to the formation of third rotated coordinate frame $\{\hat{u}_2, \hat{v}_2, \hat{w}_2\}$ as shown in Fig. 5, where

$$\hat{u}_2 = \hat{u}_1, \quad \hat{v}_2 = \cos \beta \hat{v}_1 + \sin \beta \hat{w}_1, \quad \hat{w}_2 = -\sin \beta \hat{v}_1 + \cos \beta \hat{w}_1$$

To rotate a tensor product surface, only the control points needed to be rotated. The surface is rotated initially with a minimum rotation angle $\beta$ and the rotation is incrementally increased until the second gouge free point of contact $Q$ is found. This results in two points of contact $P$ and $Q$, at which the surface and toroidal surface of tool are tangent to each other.

As the surface is tilted in $[\hat{u}_1, \hat{v}_1, \hat{w}_1]$, the pseudo-insert maintains its contact with the surface; however, the position of first point of contact on the pseudo-insert will change, sliding along the pseudo-insert from $P$ to $p'$, as shown in Fig. 5b. The surface $S(u, v)$ will stay tangent to the toroidal surface of the tool, as the rotation is about the center of pseudo-insert $O_1$.

The bisection method is used to determine the rotation angle that gives the second point of contact $Q$. In the bisection method, the surface is rotated by an angle $\beta$ and the tool is dropped over the rotated surface $S'$ in the same manner as the first point of contact. If the point of contact of the dropping tool is the same as the first point of contact the surface is tilted further for increasing $\beta$ otherwise the
rotation angle is halved to $\beta/2$ and the process is continued until second point of contact is found.

The condition for the second point of contact $Q$ to be gouge free is checked by comparing the two heights of drop, i.e., $d_{\text{min}}$ and $d'_{\text{min}}$ at every stage of rotation. The rotation angle where both the heights are equal, gives the second point of contact. Again, note that in the case of convex surfaces, after rotation, the points of contact may be on interior (hyperbolic) points on the toroidal tool.

Additional implementation details and pseudo-code for the method may be found in [15].

4 Experimental results

The algorithm from Section 3 was implemented in C++ and tested on three bi-cubic Bézier surfaces: a convex surface, a concave surface and a saddle surface. All three surfaces have a span of $150 \times 150$ in the $xy$-plane. The $x$- and $y$-coordinates 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 Fig. 6. The surfaces were machined using an $xy$-parallel toolpath as shown in Fig. 2 having 10 passes in total with the first 9 passes separated by $\text{Side step} = 18.0\text{mm}$ and the 10th pass at $x = 150\text{mm}$. Each pass contains 78 tool positions separated by $\text{Forward step} = 2.0\text{mm}$, 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 another. The algorithm runs for 760 effective tool positions that are physically making contact with the surface to be machined. The toroidal cutter used for machining had a major radius of $R_o = 6.7\text{mm}$ and minor radius of $R_i = 6\text{mm}$.

The convex and concave surfaces were used to check the accuracy of the algorithm and the saddle surface (which includes both the convex and concave regions) was used to check the behavior of algorithm as it transits from concave to convex and vice versa.

| Convex | Concave | Saddle |
|--------|---------|--------|
| 80     | 90      | 90     |
| 90     | 105     | 105    |
| 90     | 105     | 105    |
| 80     | 90      | 90     |

Table 1 Z-Coordinates (in mm) for the three bi-cubic Bézier surfaces used for testing the algorithm

Fig. 5 (a) Tilting the surface with around O1 about axis. (b) Cross-sectional view coplanar to the pseudo-insert after tilting the surface giving the first and second point of contact, P and Q respectively
Toolpath data for all three surfaces was generated for every tool position of all the passes of the toolpath footprint. The generated toolpath data was verified by first simulating the toolpath data using the ToolSim simulator [16], and then physically machined on aluminum stock using the same toolpath with a DMU-80P Hi-Dyn tilt-rotary simultaneous five-axis machining center. The results of the physical machining and the simulator match the design surfaces as shown in Fig. 7, which gives the side-by-side representation of the simulated part surface and machined part surface.

After machining, all three machined surfaces were geometrically measured using the Coordinate measuring machine (CMM) at a cross section taken in the side-step direction at \( Y = 27.00\, mm \) in the \( xy \)-plane with a probe of diameter 4\,mm. Another set of geometrical data points was taken from the machined stock produced in the ToolSim simulator. The vertices for the cross section at \( Y = 27.00\,mm \) in the \( xy \)-plane are found and stored separately for the comparison. The obtained geometrical data from machined geometry and the simulated surface was compared.
then compared with the similar cross section taken from the modeled surface. Figure 8 shows the graphical comparison of all the three surfaces, Convex, Concave and Saddle, for the three sets of geometrical data obtained from Simulated surface, Machined surface and Modeled surface.

It can be seen from the graphical representation of the surfaces in Fig. 8 that the surface data obtained from machined and simulated geometries for the three parts is similar and overlaps with each other, whereas deviation of the machined and simulated surfaces from the modeled surface in the form of scallops can be seen. The toolpath was designed with a fairly large side-step to intentionally produce scallops as a way to assist in aligning the CMM data with the simulated data; these scallops can be eliminated by reducing the side-step value as per the required surface finish.

Table 2 gives the minimum and maximum values of deviation on the machined and simulated geometries from the modeled geometry for all the three test surfaces. Maximum deviation gives the value of maximum scallop height which is higher in case of the Concave surface as compared to the Convex and Saddle surfaces. The scallops can be eliminated by reducing the side step and having 12 tool passes instead of 10; as an indication of the accuracy of the method, we have also listed in Table 2 the maximum deviation of the machined and simulated surface.

Fig. 8 Graph Showing the comparison of Machined and Simulated data with the Modeled Surface at a cross-section taken in the xy-plane at \( y = 27.0mm \) for (A) Convex Surface, (B) Concave Surface and (C) Saddle Surface
Table 2 Minimum and maximum deviation (in mm) of the machined and simulated geometries from the modeled geometry

| Surface | Machined | Simulated |
|---------|----------|-----------|
|         | Min      | Max       | Max over [70, 75] | Min      | Max       | Max over [70, 75] |
| Convex  | $8.8 \times 10^{-5}$ | 0.57 | 0.0468 | $2.9 \times 10^{-5}$ | 0.54 | 0.00912 |
| Concave | $1.8 \times 10^{-2}$ | 1.00 | 0.0513 | $1.4 \times 10^{-3}$ | 0.92 | 0.00901 |
| Saddle  | $7.6 \times 10^{-5}$ | 0.86 | 0.0398 | $2.4 \times 10^{-5}$ | 0.79 | 0.0102 |

in a region away from the scallops (in particular, the range $x \in [0.7, 0.75]$). From the table, we see that the error in the simulated surface is about 10 micrometers, while the error in the measured, machined surface is higher (40 to 50 micrometers) as expected.

Table 3 gives a timing comparison of the current method with an earlier method (SR) of Duvedi et al. [14]. As seen in the table, the new method (DRC, CPU) is roughly a factor of two faster than the earlier method. Further, the new method lends itself to a GPU implementation.

We implemented two variations of a GPU implementation. Both implementations render the surface and the tool using an orthographic projection (in the direction of the tool axis) and using the z-buffer of these renderings to determine the drop distance. In the first variation of the GPU algorithm, we used a $500 \times 500$ pixel grid where the outline of the torus just fit within this pixel grid. This method gave timing numbers similar to the CPU implementation.

In the second version of the GPU algorithm, we used a 2-level algorithm, finding an initial drop distance and point of contact $P_i$ using a $50 \times 50$ grid (again where the outline of the torus just fit within this pixel grid), and then redid the drop computation again using a $50 \times 50$ pixel grid but only in a small region around $P_i$. This second version of the GPU algorithm gave us tool paths that were numerically the same as the CPU version of the DRD algorithm, with a roughly factor of 10 improvement in performance.

5 Discussion

The method presented in this paper finds multi-point tool positions by dropping a toroidal tool onto a Bézier surface and finding the highest point of contact. By just using this variation of gouge detection from an earlier algorithm, a factor of two speed-up was obtained over numerically solving for tool positions with two points of contact. While the presented method tests for the highest point of contact at a discrete set of points, locally refining the grid in the neighborhood of the initial solution for the closest point of contact yields solutions comparable to the earlier method. Further, the discrete grid of rays cast to find the point of contact resulted in an efficient GPU implementation.

When dropping the toroidal tool on a convex surface, the first point of contact will be on the interior of the torus if the surface normal is close to aligned with the z-axis (the tool drop direction). After rotation, both points on contact can both lie on the interior of the torus. If the entire interior of the torus is not a cutting surface, a flat disk may need to be inserted inside the interior of the torus at the limiting point of the cutting edge for the interior, and rays cast from this disk. When dropping, if the closest point lies on this disk, then a rotation about this closest point in a plane containing the tool axis will yield a second point of contact on the torus. However, since this disk is a non-cutting surface, the resulting tool position would have a single point of contact.

Funding This research was supported by NSERC.

Availability of data and material Some of the surfaces used in this work are given in the paper. The remaining surfaces and other data are available on request.

Code availability Pseudo-code for the work in this paper appears in the appendix; the actual code is available on request.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Roth D, Ismail F, Bedi S (2005) Mechanistic modelling of the milling process using complex tool geometry. Int J Adv Manuf Technol 25(1–2):140–144
2. Bedi S, Ismail F, Mahjoob MJ, Chen Y (1997) Toroidal versus ball nose and flat bottom end mills. Int J Adv Manuf Technol 13(5):326–332
3. Liang F, Kang C, Fang F (2020) A review on tool orientation planning in multi-axis machining, International Journal of Production Research. Int J Prod Review, online 1–31
4. Duvedi RK, Bedi S, Batish A, Mann S (2014) A multipoint method for 5-axis machining of triangulated surface models. CAD Comput Aided Des 52:17–26
5. Duvedi RK, Bedi S, Batish A, Mann S (2015) Numeric implementation of drop and tilt method of 5-axis tool positioning for machining of triangulated surfaces. Int J Mach Tools Manuf 78:1677–1690
6. Duvedi RK, Bedi S, Batish A, Mann S (2016) The edge–torus tangency problem in multipoint machining of triangulated surface models. Int J Adv Manuf Technol 82(9–12):1959–1972
7. He Y, Chen Z (2014) Optimising tool positioning for achieving multi-point contact based on symmetrical error distribution curve in sculptured surface machining. Int J Adv Manuf Technol 73(5–8):707–714
8. Warkentin A, Ismail F, Bedi S (2000) Comparison between multi-point and other 5-axis tool positioning strategies. Int J Mach Tools Manuf 40(2):185–208
9. Warkentin A, Ismail F, Bedi S (2000) Multi-point tool positioning strategy for 5-axis machining of sculptured surfaces. Comput Aided Geomet Des 17(1):83–100
10. Kim Y.-J, Elber G, Barton M, Pottmann H (2015) Precise gouging-free tool orientations for 5-axis CNC machining. Comput Aided Des 58(1):220–229
11. Duvedi RK, Bedi S, Mann S (2017) Drop and tilt method of five-axis tool positioning for tensor product surfaces. Int J Adv Manuf Technol 93:617–622
12. Duvedi RK, Bedi S, Mann S (2018) Numerical implementation of drop and tilt method of five-axis tool positioning for tensor product surfaces. Int J Adv Manuf Technol 95(1–4):219–232
13. Duvedi RK, Bedi S, Mann S (2018) An efficient multipoint 5-axis tool positioning method for tensor product surfaces. Int J Adv Manuf Technol 97:279–295
14. Duvedi RK, Singh M, Bedi S, Mann S (2020) Multipoint tool positioning of a toroidal end mill for five-axis machining of generalized tensor product b´ezier surfaces. Int J Adv Manuf Technol 111:495–503. https://doi.org/10.1007/s00170-020-06006-1
15. Singh M (2020) Fast and robust approach to find the gouge-free tool position of the Toroidal cutter for the b´ezier surface in five axis machining. Master Thesis, University of Waterloo
16. Israeli G (2006) Software simulation of numerically controlled machining. Masters Thesis, University of Waterloo

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