A new approach to find gouge free tool positions for a toroidal cutter for Bezier surfaces in Five axis machining

Mukhmeet Sing

University of Waterloo

Jerry Qu

University of Waterloo

Ravinder Kumar Duvedi

Thapar Institute of Engineering and Technology

Sanjeev Bedi

University of Waterloo

Stephen Mann (✉ smann@uwaterloo.ca)

University of Waterloo

Research Article

Keywords: 5-axis machining, Multi-point machining, Gouge checking

Posted Date: March 17th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-300596/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on August 21st, 2021. See the published version at <https://doi.org/10.1007/s00170-021-07849-y>.

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

Mukhmeet Singh · Jerry Qu · Ravinder Kumar Duvedi · Sanjeev Bedi · Stephen Mann

Received: date / Accepted: date

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

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

J. Qu
University of Waterloo, Waterloo, ON CANADA
E-mail: j32qu@uwaterlo.ca

R. K. Duvedi
Thapar Institute of Engineering and Technology, Patiala, Punjab, INDIA
E-mail: rduvedi@thapar.edu

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

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

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) [5–10].

The MPM method has evolved to the Drop and Tilt method of Duvedi et al. for both triangulated surfaces [5, 6] and for tensor product Bézier surfaces [14–17]. The implementation of Duvedi et al. [17] 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.

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.

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

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^n P_{i,j} B_i^n(u) B_j^n(v) \quad (1)$$

Fig. 1 Geometry of Toroidal Cutter used for the machining

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 $B_i^n(u)$ are the Bernstein polynomials:

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}.$$

In this work, the toolpath is generated for the radiused end mill, as shown in Figure 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 Figure 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 Figure 2

3 The Method

The tool orientation and location are determined in two mathematical steps, as proposed by Duvedi et al. [14]. 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

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

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. [17] 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 (Figure 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.

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

$$Tool_{l,k} = \mathbf{T}_1 + \hat{t}d_{l,k} + Tor(\theta_l, \phi_k) \quad (2)$$

where $\hat{t}$ is the unit vector $[0, 0, 1]^T$ and

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

From each point $Tool_{l,k}$, a ray parallel to the tool axis is fired towards the Bézier surface. Figure 4(a) shows a set of rays emanating from points $Tool_{l,k}$ in a direction opposite to $\hat{t}$. The ray from $Tool_{l,k}$ 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

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 Figure 4(a) form a drop distance set. The drop distance set D is

where $d_{l,k}$ is a solution obtained from equations 2 and 3 given by

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 Equation 3,

$$\begin{bmatrix} Sol_x \\ Sol_y \\ Sol_z \end{bmatrix} = \begin{bmatrix} PointOnSurface_x - Tool_x \\ PointOnSurface_y - Tool_y \\ PointOnSurface_z - Tool_z \end{bmatrix} \quad (5)$$

Fig. 4 Dropping the tool over the Bézier surface along tool axis $\hat{t}$. (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 , co-planar to the pseudo-insert.

that must be solved simultaneously for three unknowns $[u, v, d_{l,k}]$, where u and v are the parameters for the surface point and $d_{l,k}$ 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 Equation 5 simplifies to

$$\begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} = \begin{bmatrix} S_x(u, v) - (T_1x + Tor_x(\theta_l, \phi_k)) \\ S_y(u, v) - (T_1y + Tor_y(\theta_l, \phi_k)) \\ S_z(u, v) - (T_1z + Tor_z(\theta_l, \phi_k)) \end{bmatrix} \quad (6)$$

In Equation 6, only the third equation corresponding to the z -direction is a function of the drop distance, $d_{l,k}$. Based on this observation the solution of Equation 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 Equation 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

$$J = \begin{bmatrix} \frac{\partial Sol_x}{\partial u} & \frac{\partial Sol_x}{\partial v} \\ \frac{\partial Sol_y}{\partial u} & \frac{\partial Sol_y}{\partial v} \end{bmatrix}. \quad (7)$$

J is the Jacobian matrix of Equation 5 and x_i is the vector of unknowns, u, v . The initial value of x_0 is set to the solution for the previous $Tool_{l,k}$, 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 $d_{l,k}$ 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 $S(u_1, v_i)$ is the first point of contact.

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 Figure 4(b). 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

$$O_1 = P + R_i \hat{n}, \quad (8)$$

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

$$\hat{n} = \frac{\frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}}{\left| \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v} \right|}. \quad (9)$$

As mentioned by Duvedi et al. [6,15,17] a rotation about the axis of this pseudo-insert maintains the contact between tool and surface at P . The earlier works [6,15] 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 [17], 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

$$\begin{aligned} \hat{w}_1 &= \hat{t} \\ \hat{u}_1 &= \frac{(T_c - P) \times \hat{w}_1}{|(T_c - P) \times \hat{w}_1|} \\ \hat{v}_1 &= \hat{w}_1 \times \hat{u}_1 \end{aligned} \quad (10)$$

The surface is defined in this coordinate frame and is then rotated about $\hat{u}_1$ by an angle β 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 Figure 5, where

$$\begin{aligned} \hat{u}_2 &= \hat{u}_1 \\ \hat{v}_2 &= \cos \beta \hat{v}_1 + \sin \beta \hat{w}_1 \\ \hat{w}_2 &= -\sin \beta \hat{v}_1 + \cos \beta \hat{w}_1 \end{aligned}$$

To rotate a tensor product surface, only the control points needed to be rotated. The surface is rotated initially with a minimum rotation angle β and the rotation

Fig. 5 (a) Tilting the surface with around O_1 about axis , (b) Cross sectional view co planar to the pseudo-insert after tilting the surface giving the first and second point of contact, P and Q respectively.

is incrementally increased until the second gouge free point of contact Q is found. This results 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 Figure 5(b). 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 β 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 β 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_{min} and d'_{min} at every stage of rotation. The rotation angle where both the heights are equal, gives the second point of contact.

Additional implementation details and pseudo-code for the method may be found in [18].

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

Convex				Concave				Saddle			
80	90	90	80	80	70	70	80	80	65	90	85
90	105	105	90	70	55	55	70	80	85	90	95
90	105	105	90	70	55	55	70	100	105	95	100
80	90	90	80	80	70	70	80	90	100	100	85

Fig. 6 Cubic Surface used to test the Algorithm

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 150×150 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 Figure 6. The surfaces were machined using an xy -parallel toolpath as shown in Figure 2 having 10 passes in total with the first 9 passes separated by $Side_{step} = 18.0mm$ and the 10th 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 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 $R_o = 6.7mm$ and minor radius of $R_i = 6mm$.

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.

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 [19], 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 Figure 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 $Y = 27.00mm$ in the xy -plane with a probe of diameter $4mm$.

Fig. 7 Three surfaces tested and verified by simulator under the section Simulated surface; and by machining with DMU-80P Hi-Dyn tilt-rotary simultaneous Five axis machining center under the section Machined surfaces

Another set of geometrical data points was taken from the machined stock produced in the ToolSim simulator. The vertices for the cross section at $Y = 27.00mm$ in the xy -plane are found and stored separately for the comparison. The obtained geometrical data from machined geometry and the simulated surface was 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 Figure 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

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

scallop 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.

Table 3 gives a timing comparison of the current method with an earlier method (SR) of Duvedi et al. [17]. 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.

Table 2 Minimum and Maximum deviation (in mm) of the Machined and Simulated geometries from the Modeled geometry

Surface	Machined		Simulated	
	Min	Max	Min	Max
Convex	8.8×10^{-5}	0.57	2.9×10^{-5}	0.54
Concave	1.8×10^{-2}	1.00	1.4×10^{-3}	0.92
Saddle	7.6×10^{-5}	0.86	2.4×10^{-5}	0.79

	SR	DRD (CPU)	DRD 500×500	DRD 2-level 50×50
Convex	498	180	153	14
Concave	476	256	157	15
Saddle	491	193	140	13

Table 3 Timing comparison of new method (DRD) to earlier method of Duvedi et al. [17]. All values are in seconds.

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 500×500 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 50×50 grid (again where the outline of the torus just fit within this pixel grid), and then redid the drop computation again using a 50×50 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 should lend itself to an efficient GPU implementation.

6 Declarations

6.1 Funding

This research was supported by NSERC.

6.2 Conflicts of interest/Competing interests

The authors have no conflicts of interest concerning this work.

6.3 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.

6.4 Code availability

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

References

1. 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.
2. 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.
3. 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.
4. N. Rao, F. Ismail, and S. Bedi, "Tool path planning for five-axis machining using the principal axis method," *Int. J. Mach. Tools Manuf.*, vol. 37, no. 7, pp. 1025–1040, 1997.
5. R. K. Duvedi, S. Bedi, A. Batish, and S. Mann, "A multipoint method for 5-axis machining of triangulated surface models," *CAD Comput. Aided Des.*, vol. 52, pp. 17–26, 2014.
6. R. K. Duvedi, S. Bedi, A. Batish, and S. Mann, "Numeric implementation of drop and tilt method of 5-axis tool positioning for machining of triangulated surfaces," *Int. J. Mach. Tools Manuf.*, vol. 78, pp. 1677–1690, 2015.
7. 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.
8. Y. He and Z. Chen, "Optimising tool positioning for achieving multi-point contact based on symmetrical error distribution curve in sculptured surface machining," *Int. J. Adv. Manuf. Technol.*, vol. 73, no. 5–8, pp. 707–714, 2014.
9. A. Warkentin, F. Ismail, and S. Bedi, "Comparison between multi-point and other 5-axis tool positioning strategies," *International Journal of Machine Tools and Manufacture*, vol. 40, no. 2, pp. 185–208, 2000.
10. A. Warkentin, F. Ismail, and S. Bedi, "Multi-point tool positioning strategy for 5-axis machining of sculptured surfaces," *Computer Aided Geometric Design*, vol. 17, no. 1, pp. 83–100, 2000.
11. A. Warkentin, F. Ismail, and S. Bedi, "Intersection approach to multi-point machining of sculptured surfaces," *Computer Aided Geometric Design*, vol. 15, no. 6, pp. 567–584, 1998.
12. P. J. Gray, F. Ismail, and S. Bedi, "Graphics-assisted Rolling Ball Method for 5-axis surface machining," *CAD Computer Aided Design*, vol. 36, no. 7, pp. 653–663, 2004.
13. H. T. Yau, C. M. Chuang, and Y. S. Lee, "Numerical control machining of triangulated sculptured surfaces in a stereo lithography format with a generalized cutter," *Int. J. Prod. Res.*, vol. 42, no. 13, pp. 2573–2598, 2004.
14. R. K. Duvedi, S. Bedi, and S. Mann, "Drop and tilt method of five-axis tool positioning for tensor product surfaces," *Int. J. Adv. Manuf. Technol.*, vol. 93, pp. 617–622, 2017.
15. 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, no. 1–4, pp. 219–232, 2018.

16. 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.
17. R.K. Duvedi, M. Singh, S. Bedi, and 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, pg 495–503 (2020). <https://doi.org/10.1007/s00170-020-06006-1>
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," Master Thesis, University of Waterloo, 2020.
19. G. Israeli, "Software Simulation of Numerically Controlled Machining," Masters Thesis, University of Waterloo, 2006.

Figures

Figure 1

Geometry of Toroidal Cutter used for the machining

Figure 2

xy-parallel toolpath footprint used for toolpath planning

Figure 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.

Figure 4

Dropping the tool over the Bezier surface along tool axis $\hat{t}$. (a) Showing the patch formed and drop distance set D obtained by intersection of fired rays from the tool surface onto the Bezier surface, (b) Cross-sectional view of tool touching the Bezier surface at P , co-planar to the pseudo-insert.

Figure 5

(a) Tilting the surface with around O_1 about axis , (b) Cross sectional view co planar to the pseudo-insert after tilting the surface giving the first and second point of contact, P and Q respectively

Figure 6

Cubic Surface used to test the Algorithm

Figure 7

Three surfaces tested and verified by simulator under the section Simulated surface; and by machining with DMU-80P Hi-Dyn tilt-rotary simultaneous Five axis machining center under the section Machined surfaces

Figure 8

Graph Showing the comparison of Machined and Simulated data with the Modeled Surface at a cross-section taken in the xy-plane at $y = 27.0\text{mm}$ for (A) Convex Surface, (B) Concave Surface and (C) Saddle Surface