Spring 2012

5-axis tool path generation with collision detection for finish machining of freeform surfaces

Jomy Francis

Department:

Recommended Citation

Francis, Jomy, "5-axis tool path generation with collision detection for finish machining of freeform surfaces" (2012). *Masters Theses*. 5135.
https://scholarsmine.mst.edu/masters_theses/5135

5-AXIS TOOL PATH GENERATION WITH COLLISION DETECTION FOR FINISH MACHINING OF FREEFORM SURFACES

By

JOMY FRANCIS

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN MANUFACTURING ENGINEERING

2012

Approved by

Dr. Frank Liou, Advisor
Dr. Xiaoping Du
Dr. Joseph W. Newkirk

PUBLICATION THESIS OPTION

This thesis is composed of one paper which was reformatted in the style used by the university.

The first paper presented in pages 4-33 titled “A NOVEL EXTENSION OF BOUNDING BOX FOR COLLISION DETECTION IN 5-AXIS TOOL PATH GENERATION FOR SURFACE FINISH MACHINING OF FREEFORM SURFACES” is intended for submission to INTERNATIONAL JOURNAL OF COMPUTER-AIDED DESIGN AND APPLICATIONS.

ABSTRACT

Research in the field of Tool path generation for freeform surfaces has been done intensively in the past. However, the main challenge that still exists is the computational efficiency related to the tool path generation. Tool path generation for freeform surface involves instantaneous calculation of new tool orientations which does not collide with the neighboring surfaces. Since the collision check of tool and neighboring surface is done repetitively at every instant of the tool, the calculations at every instant are to be computationally as easy as possible.

This thesis is composed of one paper. Paper I presents a novel extension of the Bounding Box technique used for collision detection. This novel method solves the above mentioned challenge of computational efficiency in the field of tool path generation. The new approach that has been implemented in Paper I involves using the simplest computational operators that are comparison operators along with a novel Diagonal Bounding Box technique. This ensures the tool path generation to be less cumbersome computationally.

Furthermore, the boundaries of the proposed machining algorithm in terms of collision correction and the proper application of the machining algorithm have been explored.

ACKNOWLEDGMENT

This research work is a result of some phenomenal help and support extended to me by many individuals at Missouri S&T. I would like to express my sincere gratitude towards my advisor Dr. Frank Liou and Mr. Todd Sparks for their continued guidance, encouragement and co-operation throughout my research work. It has been a pleasure working with them over the past two years and without their help this work would not have been possible. The research assistantship extended to me by Dr. Liou through the Manufacturing Engineering program is also greatly acknowledged.

I would also like to thank my committee members Dr. Xiaoping Du and Dr. Joseph W. Newkirk for their time and advice granted to me during the research. I would like to express my sincere thanks to Jianzhong Ruan, Swati and Divya for helping me during the initial stages of the tool path generation software programming. I sincerely thank all the members of the LAMP lab, especially Sriram Praneeth Isanaka and Sujit Dongre for helping me experiments and providing me with valuable suggestions which have been very critical in completing my research work. I would also like to thank my roommates at Rolla, Mathew and Rana who have made the past two years memorable.

Finally, I would like to express my gratitude towards my parents, Mr. Francis Jacob, Mrs. Annie Francis and my brother Romy for their unconditional love and support and God Almighty for guiding me throughout the various stages of my life.

TABLE OF CONTENTS

Page

SECTION

PAPER

SECTION

LIST OF ILLUSTRATIONS

Figure Page

PAPER I

1. INTRODUCTION

1.1. OBJECTIVE

Research in the field of tool path generation for machining freeform surfaces has one major challenge, which is the computational efficiency of creating the tool paths. The basic idea is to keep the calculation as simple as possible or use computational operators that are the quickest for a computer language. Thus this research aims at solving the major challenge of computational efficiency by proposing and implementing a novel extension of the bounding box technique for calculating collision free tool orientations at every instant.

1.2. BACKGROUND AND PROPOSED TECHNIQUES

Laser Aided Metal Deposition Process creates a product using the concept of additive manufacturing. The main applications of this process are in part repairs and generation of freeform and complex surfaces. Aerospace metals such as titanium are a good example for the explanation of this process. Once titanium alloy powder has been deposited, the final part geometry needs further finish machining operations to have the final customer specified surface finish.

5-axis surface finish machining is used to machine these freeform complex surfaces. Now since the part shapes and the material to be machined involved are complex in geometry and physical property, an optimum collision free tool path has to be generated to achieve the final result of customer specified surface finish with least tool wear or breakage.

In the past, tool path generation for freeform surfaces has been extensively researched. The main challenge in this area though is the computational efficiency of the tool path generation process. The method of scallop height as mentioned in paper I [3] is one of the most popular techniques to generate tool paths. Collision detection of the tool with the neighboring surface has been done using different techniques like C-space method. All these methods aim at solving the same issue of computational efficiency.

This research incorporates the various established techniques of tool path generation and adds to the tool path generation algorithm a new approach of bounding box. This new approach has the main inclination on reducing the overall computational time. This has been done by using the fastest computational comparators that are the “comparison operators (‘<’ & ‘>’)”.

1.3. CONTRIBUTIONS

Listed below are the contributions of Paper-I.

- A novel technique to calculate the initial probable collision points inside the bounding box. This approach is computationally less cumbersome and thus aims at solving the main problem in the field of tool path generation which is computational efficiency.

- A simple integrated approach of machining a freeform surface from b-spline surface generation → optimizing the tool path → collision detection → tool path generation.

PAPER

I. A NOVEL EXTENSION OF BOUNDING BOX FOR COLLISION DETECTION IN 5-AXIS TOOL PATH GENERATION FOR SURFACE FINISH MACHINING OF FREEFORM SURFACES

Jomy Francis[1], Todd E. Sparks[2], Jianzhong Ruan[3] and Frank Liou[4]

[1]Missouri University of Science and Technology, jfb55@mst.edu
[2]Missouri University of Science and Technology, toddesparks@gmail.com
[3]Missouri University of Science and Technology, jzruan@gmail.com
[4]Missouri University of Science and Technology, liou@mst.edu

Abstract

This paper proposes a completely automated and integrated tool path planning software for finish machining of freeform surfaces. This software's capability spans from generation of b-spline freeform surfaces to optimizing the surface finish to Collision Detection to tool path generation. Two scallop height methods have been used to compare the optimal tool path strategy. Collision detection of tool with neighboring surfaces and collision correction for tool are solved by using a novel extension of bounding box which uses body diagonal points for computation. Furthermore, this paper proposes a multiple screening technique to improve the computational efficiency of tool path generation calculations. Final freeform machining has been implemented on wax using Fryer 5X-45 machining center.

Keywords: freeform surface, 5-axis machining, collision, bounding box, tool path.

1. INTRODUCTION

Research in the field of tool path generation for 5-axis machining using ball-end mill has been very extensive. The concept of scallop height introduced [1] by Hsi-Yung Feng has been used in this paper. Furthermore, in the area of collision detection, the concept of treating tool holder as cylinder [2], has been modified to be as cuboids. Also, research related to bounding boxes with regards to text blocks as shown in [4] has been done in the past. However, these bounding box techniques when applied to dynamic tool movement appear to be computationally cumbersome.

Thus, this paper aims at integrating the various existing ideas of tool path generation and collision detection using bounding box. In the process, it also proposes a novel application of bounding box technique to improve the computational efficiency of collision check.

We are using python 2.7 environment with numpy and scipy modules for coding. Matplotlib has been used for 3D-plotting. 5-axis machining has been done on Fryer 5X-45 machining center. Our tool is 0.5” ball-end mill.

2. NOMENCLATURE

r = tool radius

h = scallop height

R = radius of curvature of the surface at any given iteration

$P(\mathrm{i}, j)$ = control points for generating b-spline surfaces

u, v = parametric space replacing x,y,z 3d co-ordinate system

k, l = degree of curve along u and v

$N(\mathrm{i}, k)$, $N(j, l)$ = basis functions along u, and v respectively, used to generate the b-spline surfaces

3. REPRESENTATION OF FREEFORM SURFACE

Freeform surfaces are defined by using B-spline surfaces (Fig. 1.).

$$\boxed{P\left(u,v\right)=\sum_{i=0}^{n}\sum_{j-0}^{m}P_{ij}N_{i,k}\left(u\right)N_{j,l}\left(v\right);\quad 0\le u,v\le u_{max},v_{max}} \tag{1}$$

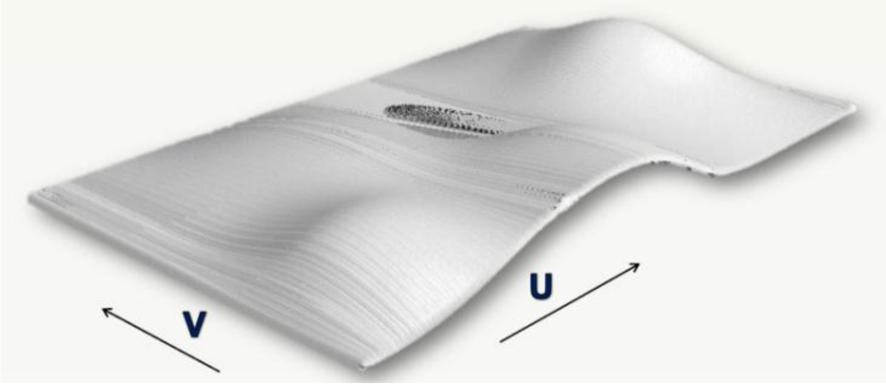


Fig. 1: B-Spline Surface Generation.

Input: Control points, knots & degree of curve along u & v.

Output: B-Spline-surface that can map from u,v → x,y,z

4. TOOL PATH GENERATION

4.1. BACKGROUND & TERMINOLOGIES

Scallop is the amount of material that is intentionally left behind on the surface of the final machined part as shown in Fig. 2. Scallops are formed when the tool steps-over for the next tool path by the calculated tool path interval (TPI) or step-over distance.

The tool is by default aligned with the surface normal of any surface point at any given instant in 5-axis machining. This in turn may result in collision of tool with neighboring surface. It has been explained in detail in Sec. 5.

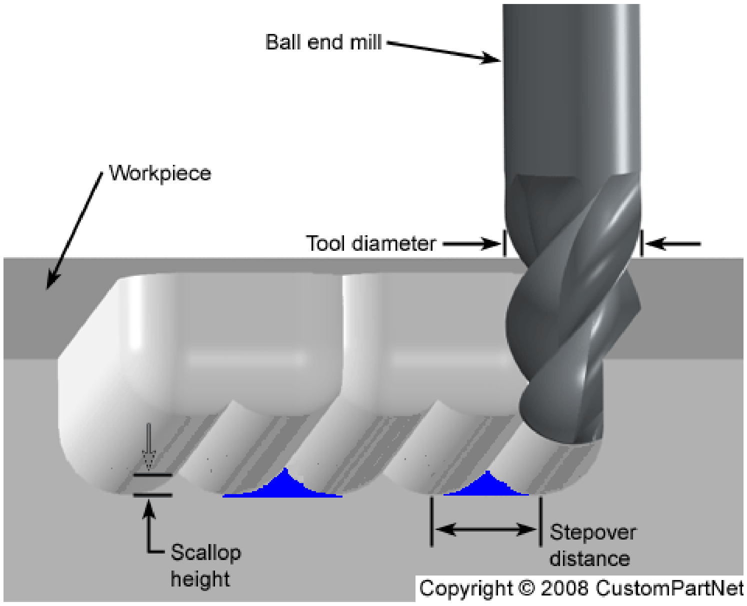


Fig. 2: Scallop Height and Tool Path Interval.

We have selected cutter location (CL) points instead of cutter contact (CC) points as the parameter to create the tool path. This is because, if CC point is the rotation center when collision correction is applied, there will be gouging (also called local interference) as shown in Fig. 3. Thus, we have overcome gouging by pivoting the tool about the CL point when applying collision correction.

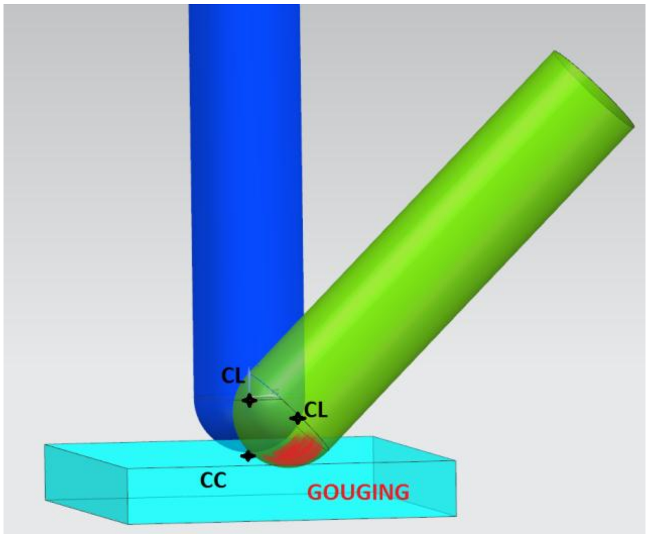


Fig. 3: Gouging when Rotation About CC.

4.2. TOOL PATH INTERVAL & CURVATURE RADIUS

A freeform surface can be generalized as having one of any three contours at any given point. These are Convex, Concave or flat surface curvature. TPI's for these three conditions are shown in Fig. 4. respectively, which incorporate the scallop height as a customer input factor.

The circles represent the ball end of tool when the tool is about to step-over for the next tool path.

Using Eqn. (2.1), Eqn. (2.2) & Eqn. (2.3), for the respective surfaces shown in Fig. 4(a)., Fig. 4(b)., & Fig. 4(c)., TPI's have been calculated.

The consecutive points along the parametric direction perpendicular to the tool travel are taken to calculate the Radius of Curvature of the surface at that instant. For example if the tool travels along *'V'* parametric space then, *'U', 'U+0.02'*, & *'U+0.04* will be considered to calculate the Radius of Curvature at the corresponding next step over point along *'U'* parametric space.*'*

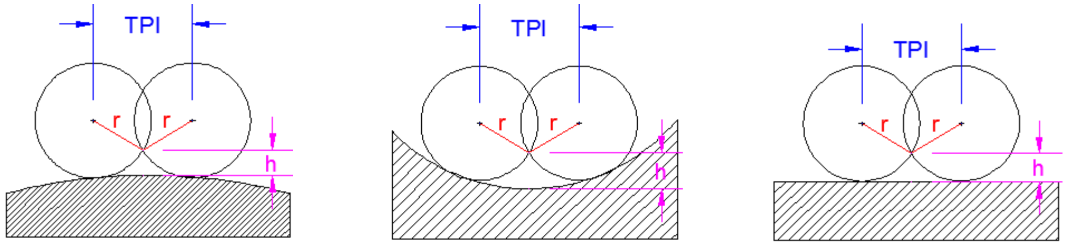


Fig. 4: Scallop Height and TPI. (a) Convex Surface, (b) Concave Surface, (c) Flat Surface.

$$TPI_{convex} = \frac{8hrR}{\sqrt{|R-r|}}_{[3]} \tag{2.1}$$

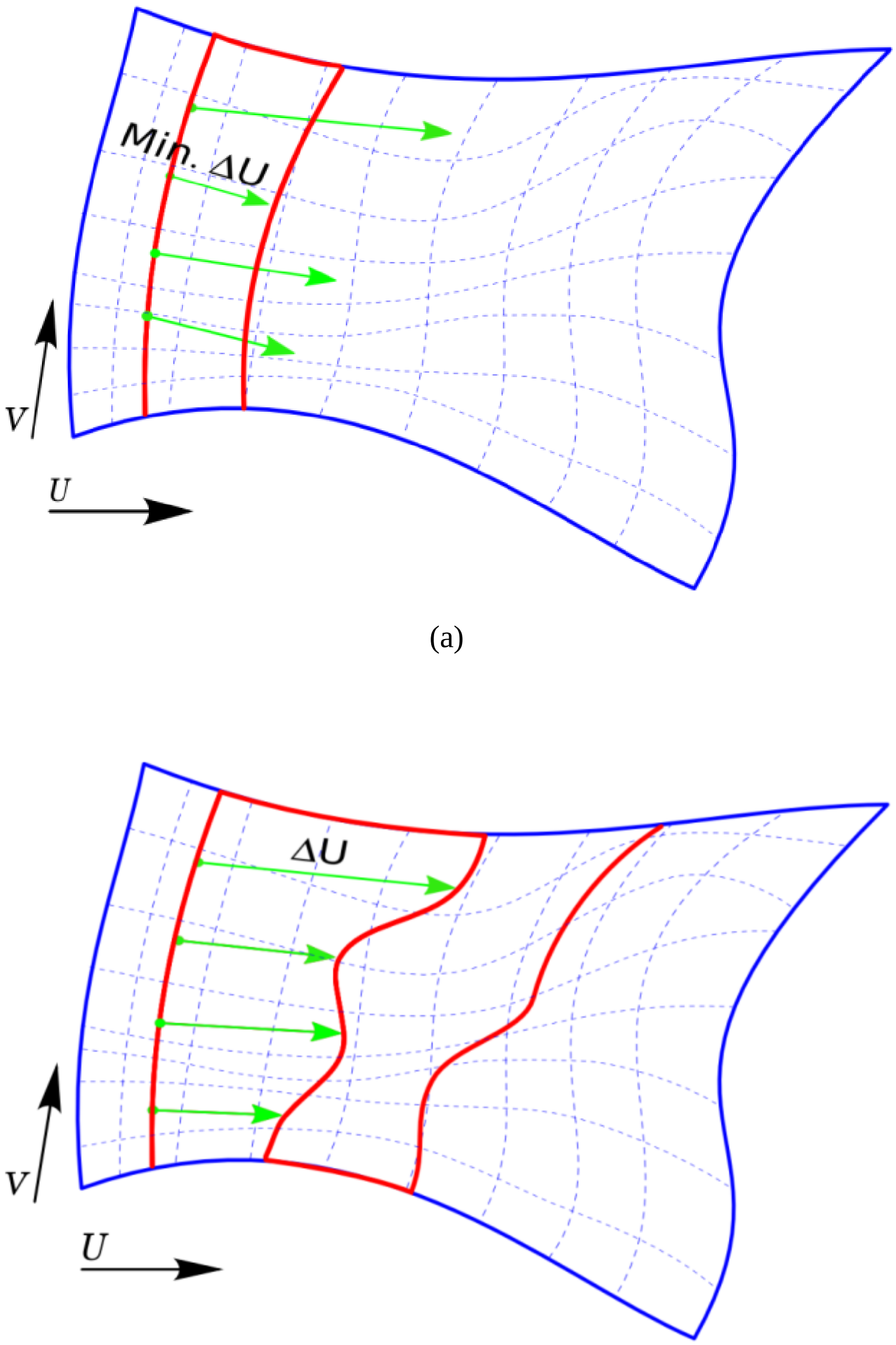
$$TPI_{convex} = \frac{8hrR}{\sqrt{|R+r|}}_{[3]} \tag{2.2}$$

$$TPI_{flat} = 2\sqrt{r^2-(r-h)^2}_{[3]} \tag{2.3}$$

4.3. SCALLOP HEIGHT & TOOL PATH STRATEGY

Two strategies were implemented for sweeping across the free-form surface with constant scallop height (CSH).

4.3.1. Minimum Step-Over Scallop Height Method (MSH). The next TPI along U parametric space is calculated along every movement in V. Then the minimum amongst the set of U is selected as the next TPI. This method gives more tool passes thus giving smoother surfaces but lower machining efficiency. Fig. 5(a).



(a)

(b)

Fig. 5: Tool Path Strategy. (a) Minimum Step-Over Method, (b) Constant Scallop Height Method with Maximum Step-Over.

4.3.2. Constant Scallop Height Method with Maximum Step-Over (CSH). The next TPI along U parametric space is calculated along every movement in V. The corresponding next U for every current U is calculated till any U is greater than Umax. This new calculated set of different U's is used as the next tool path interval along with the constant divisions in V. Refer Fig. 5(b).

Thus, this method sweeps across the finish machining surface in fewer passes as compared to the earlier method. However, the smoothness of the final surface would be less as compare to the earlier method (MSH).

Now that we know the cutter contact (CC) points and tool orientation (default orientation = surface normal at CC) at those points, the next step is to check if collision exists between the tool and the neighboring surface.

5. COLLISION DETECTION

Once the initial tool path is generated, there might be instance where the tool collides with either the current surface or the neighboring surface Fig. 6.

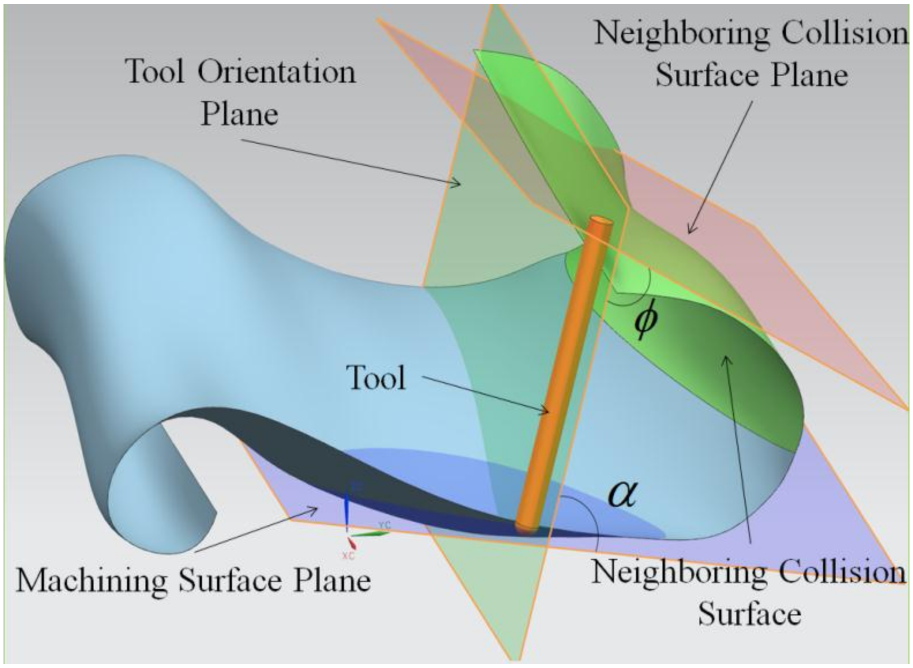


Fig. 6: Tool Collision with Neighboring Surface.

Thus, the goal is to identify the pool of surface points that might collide with the tool at any given instant of tool orientation. This has been achieved in 2 steps by using the concept of Bounding Box (BB).

Bounding box technique has been used for collision detection in game engines and also in tool path generation. However, a novel extension of the bounding box technique has been implemented in this paper. This approach makes tool path generation computationally efficient when a lot of iterations of collision check are done for a huge number of surface points. Section 5.1 & 5.2 further explains the flaws with a simple bounding box and the solution that has been implemented.

5.1. DYNAMIC BODY-DIAGONAL BOUNDING BOX FOR INITIAL COARSE SCREENING

A simple rectangular bounding box (BB) is generally used initially to have better computational efficiency. A simple rectangular BB uses “greater than” & “less than” comparators. These are the fastest in computation. Thus, a rectangular BB helps in quickly sieving out any unnecessary points when collision of the tool at a given cutter contact point is being checked with the entire surface.

The size of the conventional rectangular BB is the “diameter+tolerance” of the tool & the “Length of the tool + tolerance”, Fig. 7(a). However, the conventional BB theory fails to include the whole tool when the tool is tilted as shown in Fig. 7(b). Thus, the bounding box has to dynamically increase according to the tilt in the tool.

$$CL = CC + r.\hat{n} \tag{3.1}$$

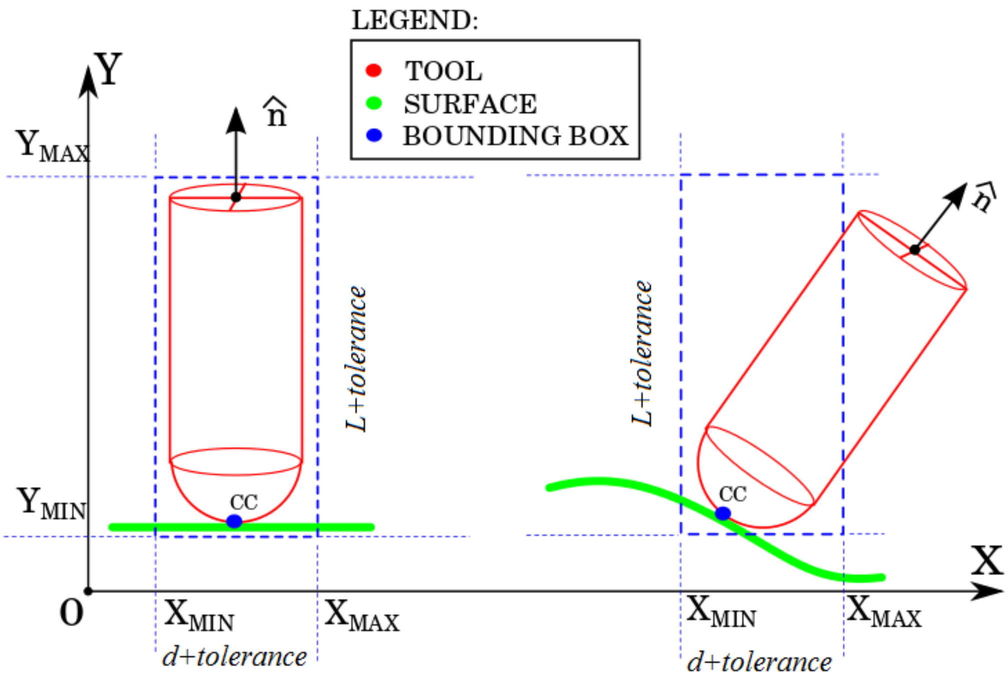
$$A(x_{a,} y_{a,} z_{a,}) = CL + \overrightarrow{K} \tag{3.2}$$

$$B(x_{b,} y_{b,} z_{b,}) = (CL + \hat{n}.L) - \overrightarrow{K} \tag{3.3}$$

$$\overrightarrow{K} = \left(L\left[\hat{n}\cdot\hat{x}\right] + r\right)\hat{x} + \left(L\left[\hat{n}\cdot\hat{y}\right] + r\right)\hat{y} - sign(\hat{n}\cdot\hat{z})r\hat{z} \tag{3.4}$$

K is the vector that is symmetric about the surface normal. This furthermore adds to reduced computation when finding the body diagonal points A & B, refer Fig. 8. Body-Diagonal Rectangular BB serves two purposes:

- It successfully captures the entire tool mathematically at every instant of tool motion by dynamically increasing the bounds of the box.
- It is computationally fast as it adheres to the basic concept of Bounding Box being rectangular.



(a) (b)

Fig. 7: Conventional Bounding Box Flaw. (a) Bounding Box Covering Entire Tool for Flat Surface, (b) Bounding Box Failing to Cover Entire Tool for Curved Surface.

We perform this initial coarse screening of surface points to have a small set probable collision surface points (Fig. 9). Points inside BB (PIBB) are the next input for fine screening. PIBB are the coarsely sieved surface points that might collide with the current tool orientation.

Please refer next page for Fig. 8 and Fig. 9.

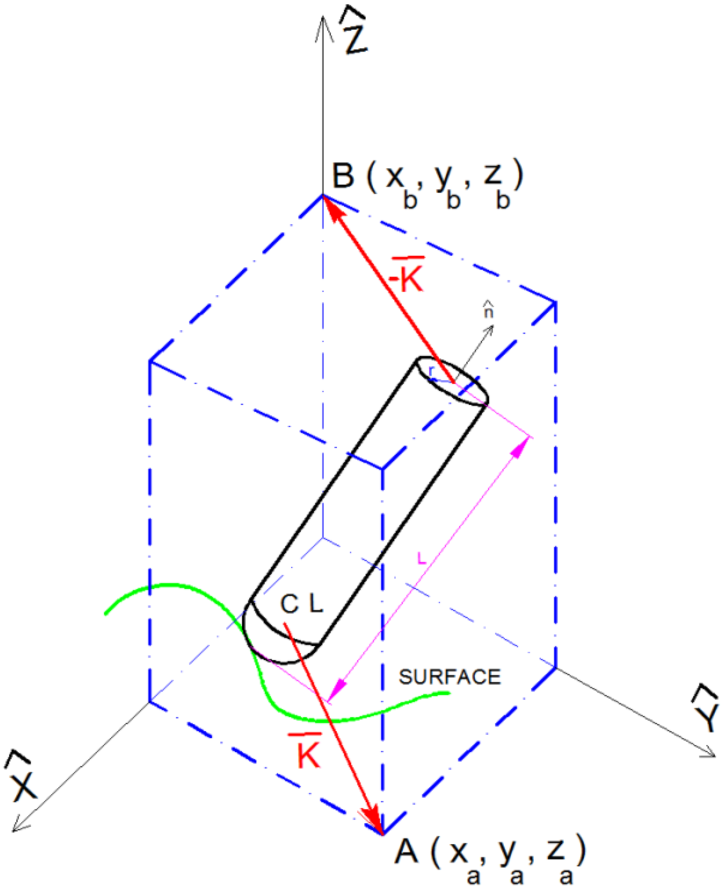


Fig. 8: Initial Bounding Box by Body Diagonal Points.

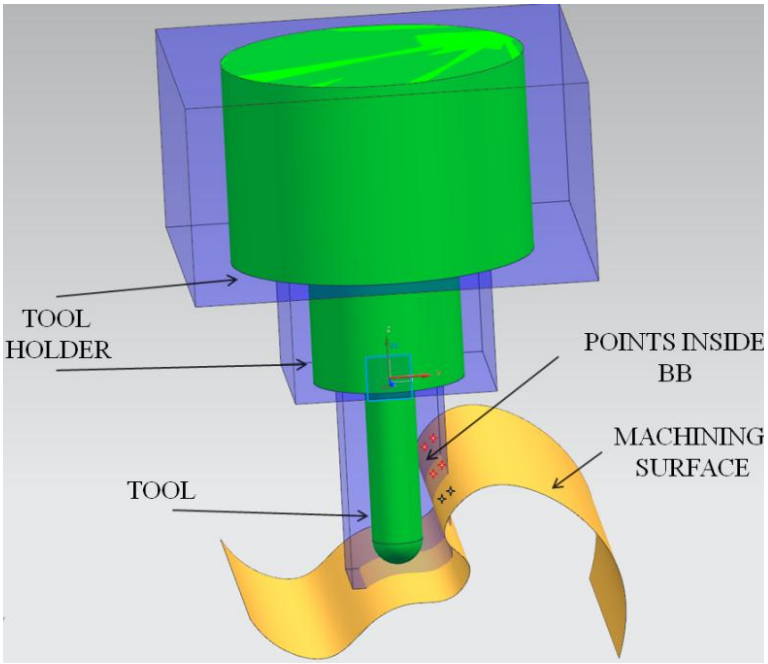


Fig. 9: Probable Collision Points for Fine Screening.

5.2. DETAILED BOUNDING BOX & FINE SCREENING

From the input of PIBB, a new bounding box is formed mathematically which mocks the tool at that instant, refer Fig. 10.

The input of PIBB is further sieved through Eqn. (3.1), Eqn. (3.2), Eqn. (3.3), and Eqn. (3.4) respectively, to mathematically mock the tool position for the current CC point and tool orientation. This then gives us the final set of collision points.

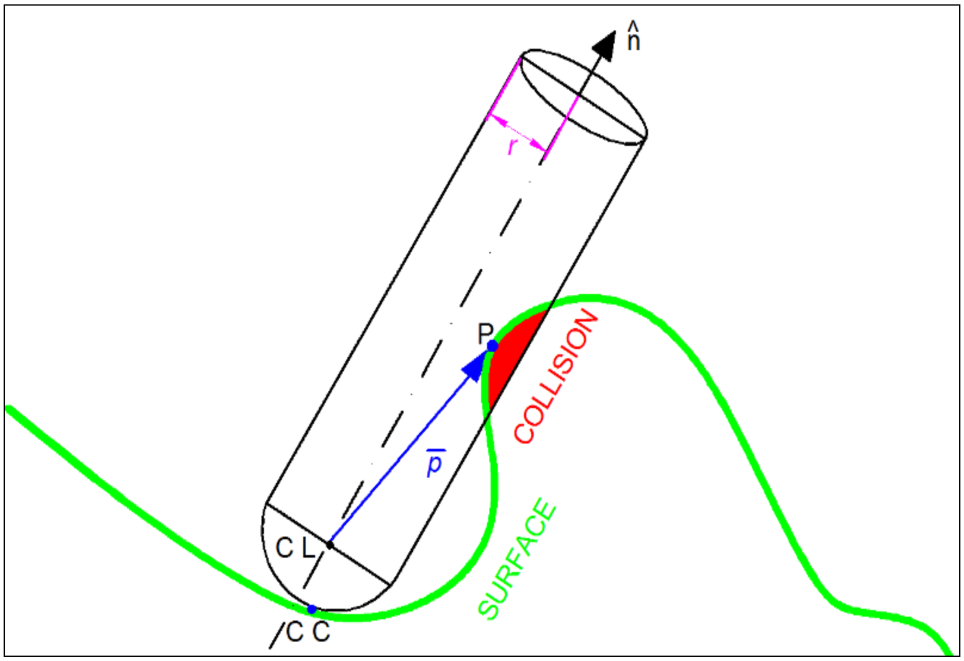


Fig. 10: Detailed Bounding Box and Collision Points.

Eqn. (3.1) checks and takes all the points that are above the CC plane. Eqn. (3.2) checks from this new set of PIBB for the points that lie inside the projected cylinder.

$$\hat{n}\cdot\vec{p} > 0 \tag{3.1}$$

$$|\vec{p}-\hat{n}(\hat{n}\cdot\vec{p})| < r \tag{3.2}$$

$$\hat{n}\cdot\vec{p} < r \tag{3.3}$$

$$|P-CL| \le r \tag{3.4}$$

Fine Screening Algorithm

Eqn. (3.1) sieves and takes all the points from PIBB which are above the CC plane.

Eqn. (3.2) forms a projected cylinder along the tool normal and takes all the points lying inside this cylinder

Eqn. (3.3) concentrates the next check just below the CL plane where the projected cylinder needs to be corrected to mathematically mock a hemisphere of ball end mill

Eqn. (3.4) checks sieves and takes all the points from the sieved points that lie inside the ball area of the projected cylinder

This set of final collision points is sent to the next step of collision correction which has been explained in the next section.

6. COLLISION CORRECTION

The aim of collision correction is to

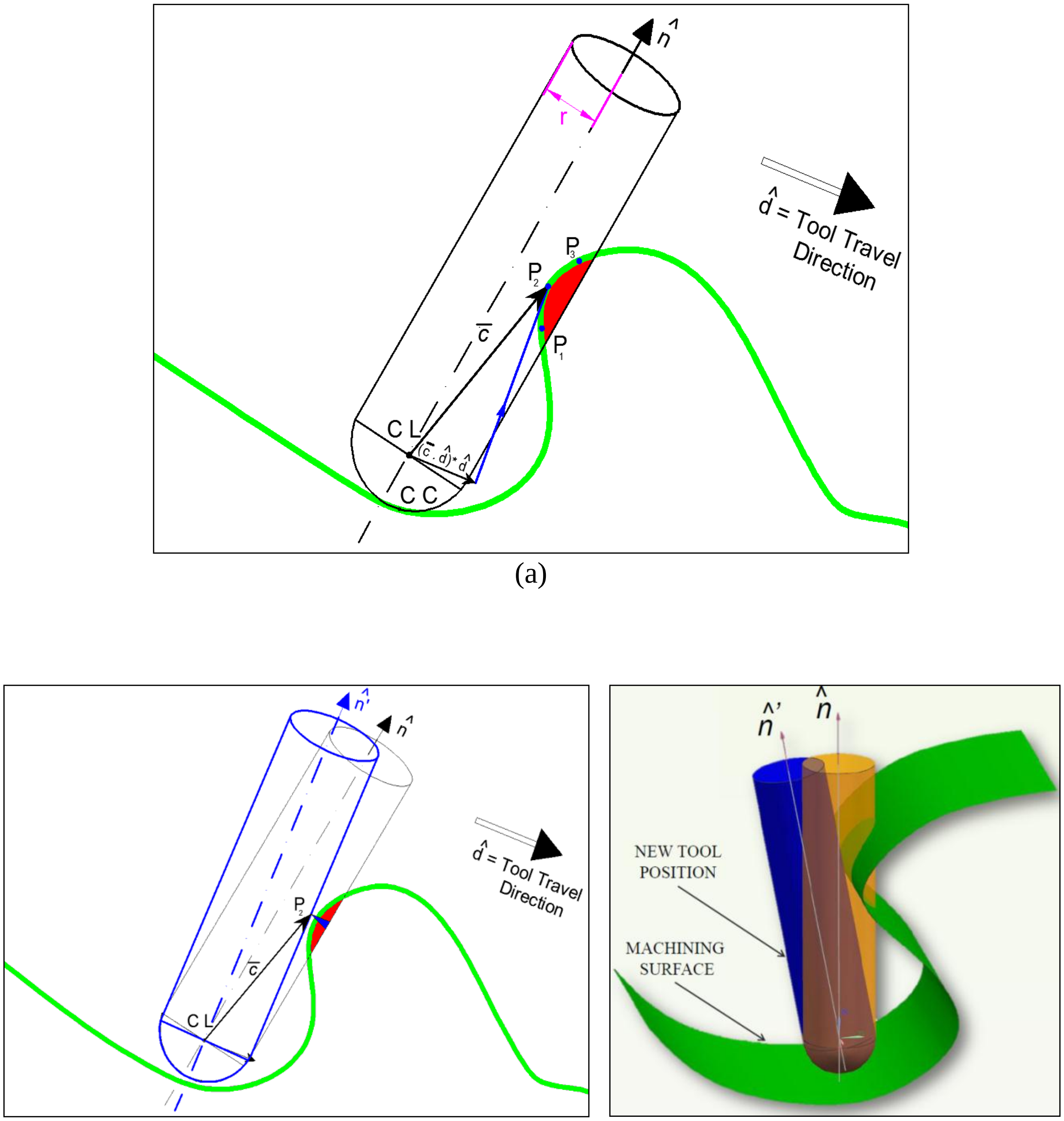
- Firstly, find the point from set of collision points that will first collide with the tool
- Secondly, to find the new tool direction that will be collision free

In Fig. 11(a)., $P_1, P_2 \,\&\, P_3$ are the inputs for collision correction functions. They are the final set of collision points. From this set, the point closest to the tool (i.e. which would first collide with the tool) is selected. This has been achieved by calculating the component of vector c along the tool travel direction d for every collision point. Then, we take $\min(\vec{c}\cdot\hat{d})$. In Fig. 11(a)., P_2 would first collide with the tool.

Once the closest point to the tool that will collide (P_2) is found, we find the new tool direction from Eqn. (4).

$$n' = \frac{\vec{c} - \left(\vec{c}\cdot\hat{d}\right)\hat{d}}{\left|\vec{c} - \left(\vec{c}\cdot\hat{d}\right)\hat{d}\right|} \tag{4}$$

Eqn. (4) calculates the new tool orientation by avoiding collision with the closest point (P_2). Thus, we find the new collision free tool position, Fig. 11(b). & Fig. 11(c). This new tool orientation along with constant CL point is sent back to the collision check algorithm. It checks if this new tool orientation collides with any other surface points. This loop of check keeps on executing till a tool orientation is found which does not have any collision with the neighboring surface.



(a)

(b)

(c)

Fig. 11: Collision Correction. (a) First Point that Collides with Tool, (b) Collision Corrected Tool direction, (c) 3D Representation.

7. ALGORITHM

```
#========================= MAIN PROGRAM ========================#
-------------------- OBJECT CREATION --------------------
creating B-spline_surface CLASS object
creating Tool_data CLASS object
creating Tool_path_ CLASS object

TOOL PATH_as_xyzijk or xyzac = tool_path_oject_call(tool_path_generation)
G&M code for machining = Post_Processor(xyzijk or xyzac)
a,c = rotation about x-axis & z-axis
#=================================================================#

#======================== Classes and Functions =======================#
-------------------- TOOL PATH GENERATION --------------------
Tool_path → tool path generation
-------------------- Final UV List calculations for tool orientation decisions ------------------
        Method 1: Minimum stepover method
                calculating stepover(u,v)
                        3points → bsplinesurface[(u,v), (u,v+0.2h), (u,v+0.4h)]
                        Checking if point(u,v) lies on flat, concave or convex
                        return Stepover
        minimum(stepover) as next stepover
```

```
    Method 2: Max stepover and CSH method
        calculating stepover(u,v)
            3points → bsplinesurface[(u,v), (u,v+0.2h), (u,v+0.4h)]
            Checking if point(u,v) lies on flat, concave or convex
            return Stepover
    Storing every stepover for corresponding 'v'
    Dynamic stepover for every changing 'v' from previously calculated list


-------------------- Loop of collision check & correction--------------------
    I/P: new corrected tool direction□ until no more collision
    Run through the entire Final UV list
        collision_check(CL_point, new_tool_normal)
            collision_points = initial_screening(surface_points)
            refined_points = detailed_screening(points inside BB)

        collision_correction(final_set_of_collision_points)
            closest point to tool(final_set_of_collision_points)
            new corrected tool direction (current tool direction)
            return (CL_points and collision free tool-directions)


   return Final CL points and tool directions [x,y,z,i,j,k] OR
   return Final CL points & angles [x,y,z,a,c]
```

8. RESULTS

The final tool path for one of the free-form surfaces has been shown in Fig. 12(a). & Fig. 12(b). We have used $h=0.125$, just for better representation and tool dia., $d=0.5$”. The plotting has been done in matplotlib.

The two different tool path strategies of Minimum Step-over (MSH) & maximum step-over constant scallop height (CSH) were compared. From Fig. 12(a). & Fig. 12(b). respectively, we have,

- No. of tool-passes in MSH = 8; Better Surface finish
- No. of tool-passes in CSH = 7; Better Machining Efficiency

Maximum Step-over Constant Scallop Height (CSH) method has been selected for real life machining. The machining has been implemented using the Fryer 5X-45 5-axis machining center at Missouri University of Science of Technology; Rolla, MO. Wax block has been used as the work piece material. Fig. 13(a). & Fig. 13(b)., shows CSH machining using scallop height, $h=0.125$” and $h=0.005$”.

Refer the next two pages for the figures of generated tool paths and machined tool paths for surface finish machining.

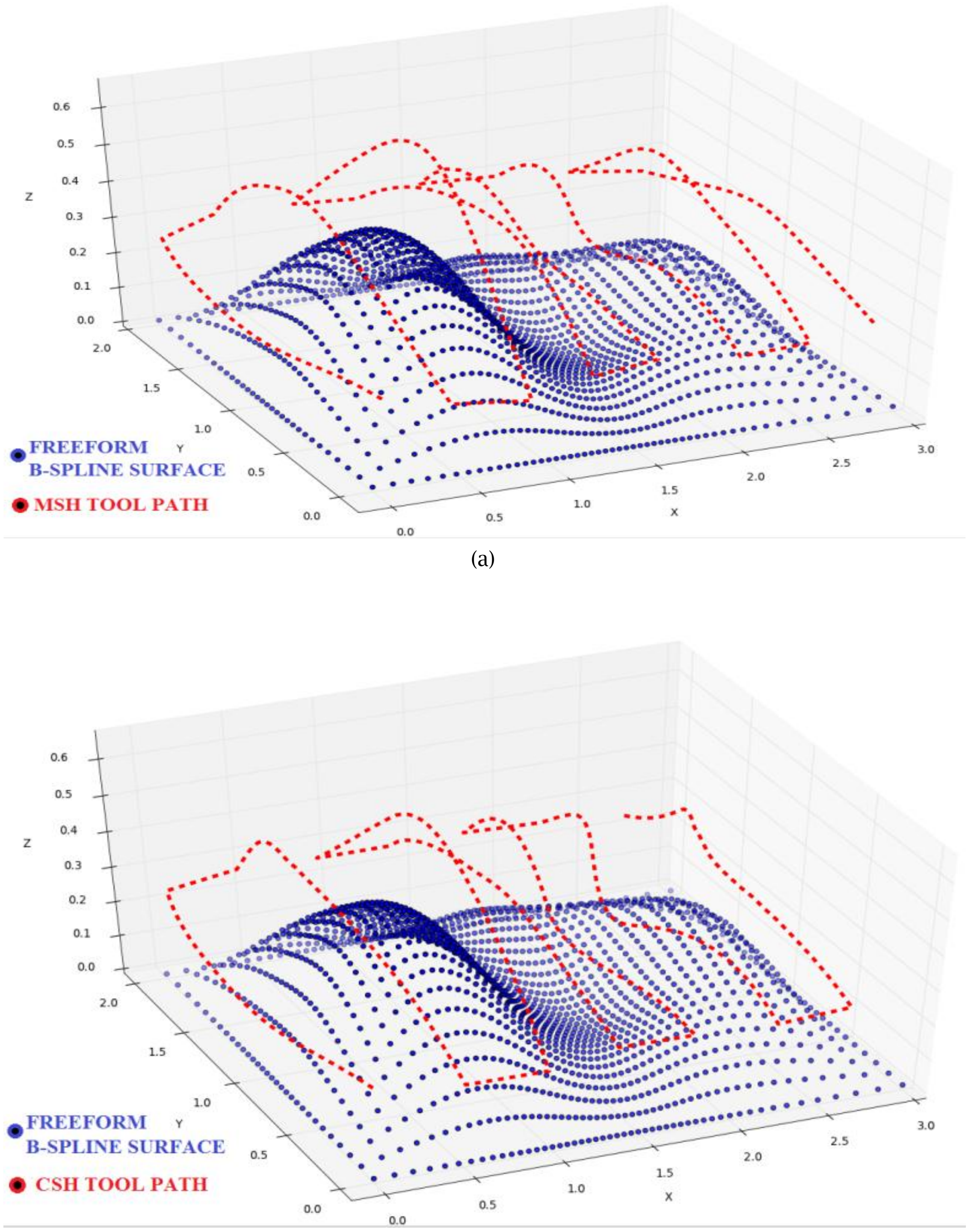


Fig. 12: Generated Tool Paths with h=0.125”. (a) MSH Tool Path with 8 Tool Passes, (b) CSH Tool Path with 7 Tool Passes.

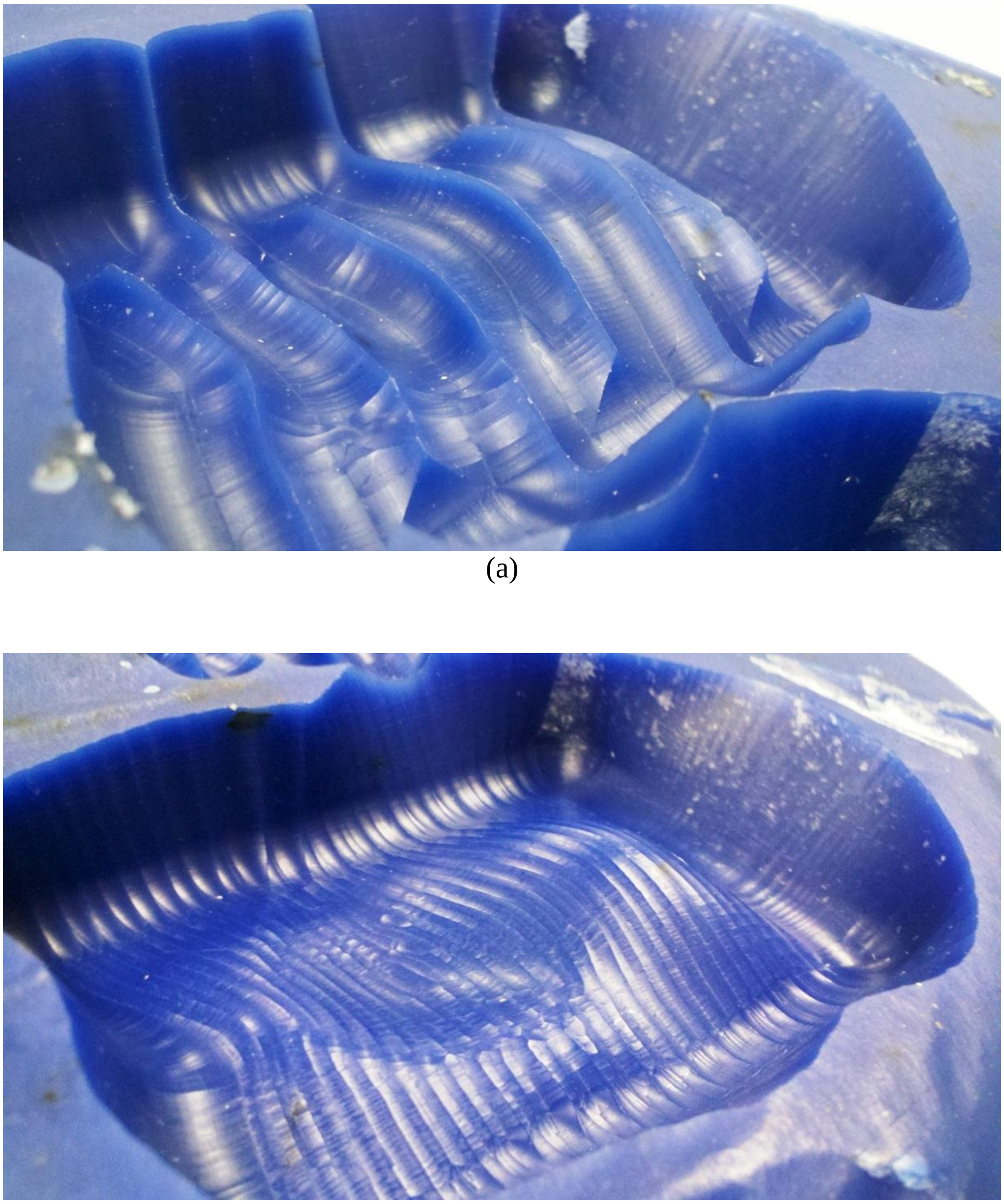


Fig.13: CSH Machined Tool Path. (a) Scallop Height = 0.125”, (b) Scallop height = 0.005”.

9. EXPLORATION AND LIMITATIONS

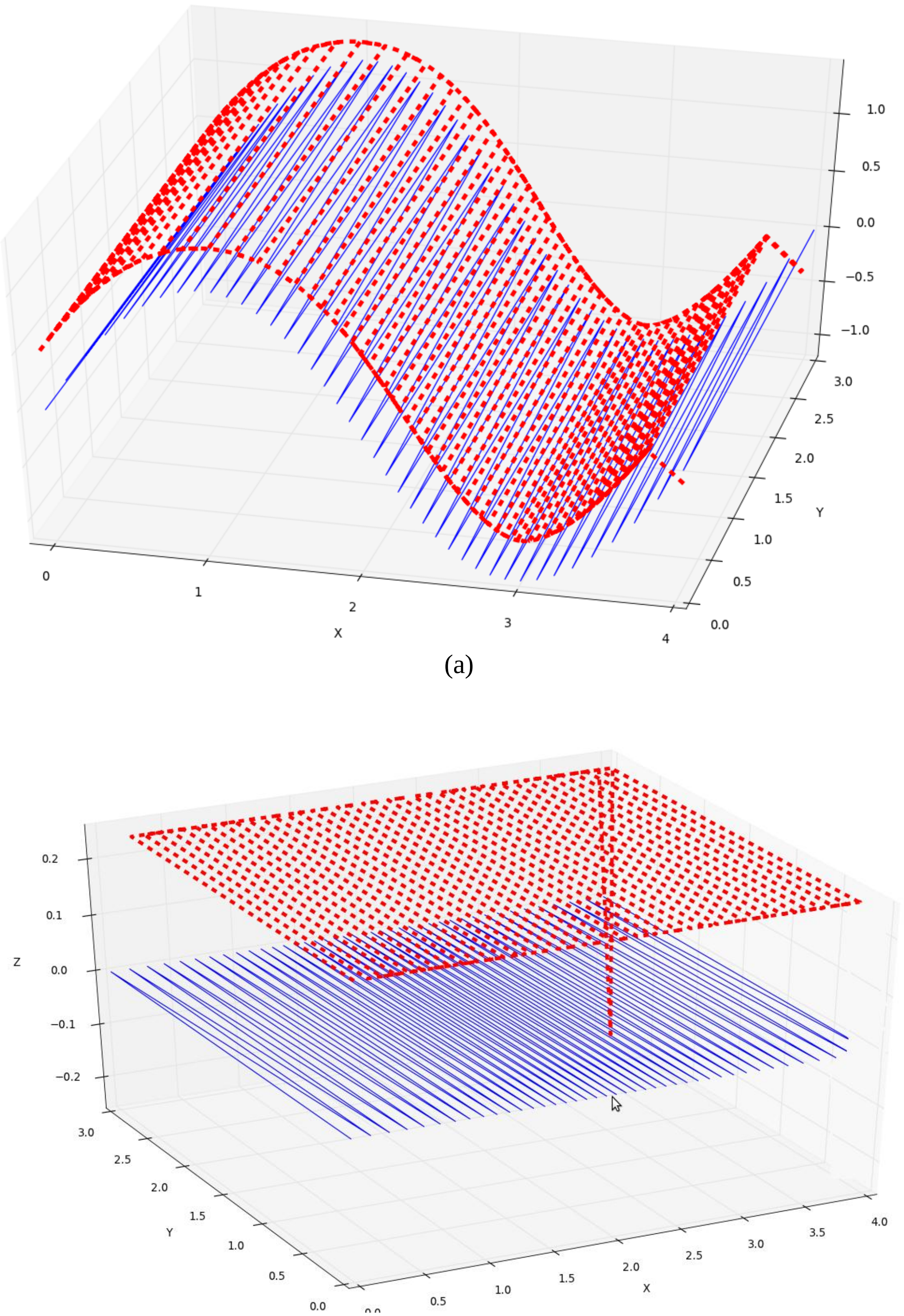
This section is aimed at shedding some light on the proper application and the limitations of this algorithm. This has been achieved by making three classification; Superfluous, Efficient and Extreme. Below mentioned are their respective explanations.

9.1. SUPERFLUOUS

These are surfaces which can be machined using a general 3-axis machining algorithm or surfaces which do not have extreme curvatures or surfaces that do not need collision detection. For example, flat surfaces, sine curves, general geometric shapes such as rectangles, triangles etc. Fig. 14. shows an example of this where our machining algorithm has been used.

9.2. EFFICIENT

These are surfaces which are apt for being machined using our machining algorithm. Examples of this would involve surfaces that have curvatures that would make the tool collide with the part geometry and thus would require collision correction. Fig. 15. shows the various shapes and the corresponding successful collision free tool paths generated using our proposed machining algorithm. The b-spline freeform surface has been shown in blue and red color represents the collision free tool path. Scallop height of 0.1” has been used to show that the proposed algorithm is capable of generating collision free tool paths for complex surfaces (Fig. 15).



(a)

(b)

Fig.14: Superfluous Tool Paths for Simple Surfaces. (a) Sine Curve Surface, (b) Flat Plane.

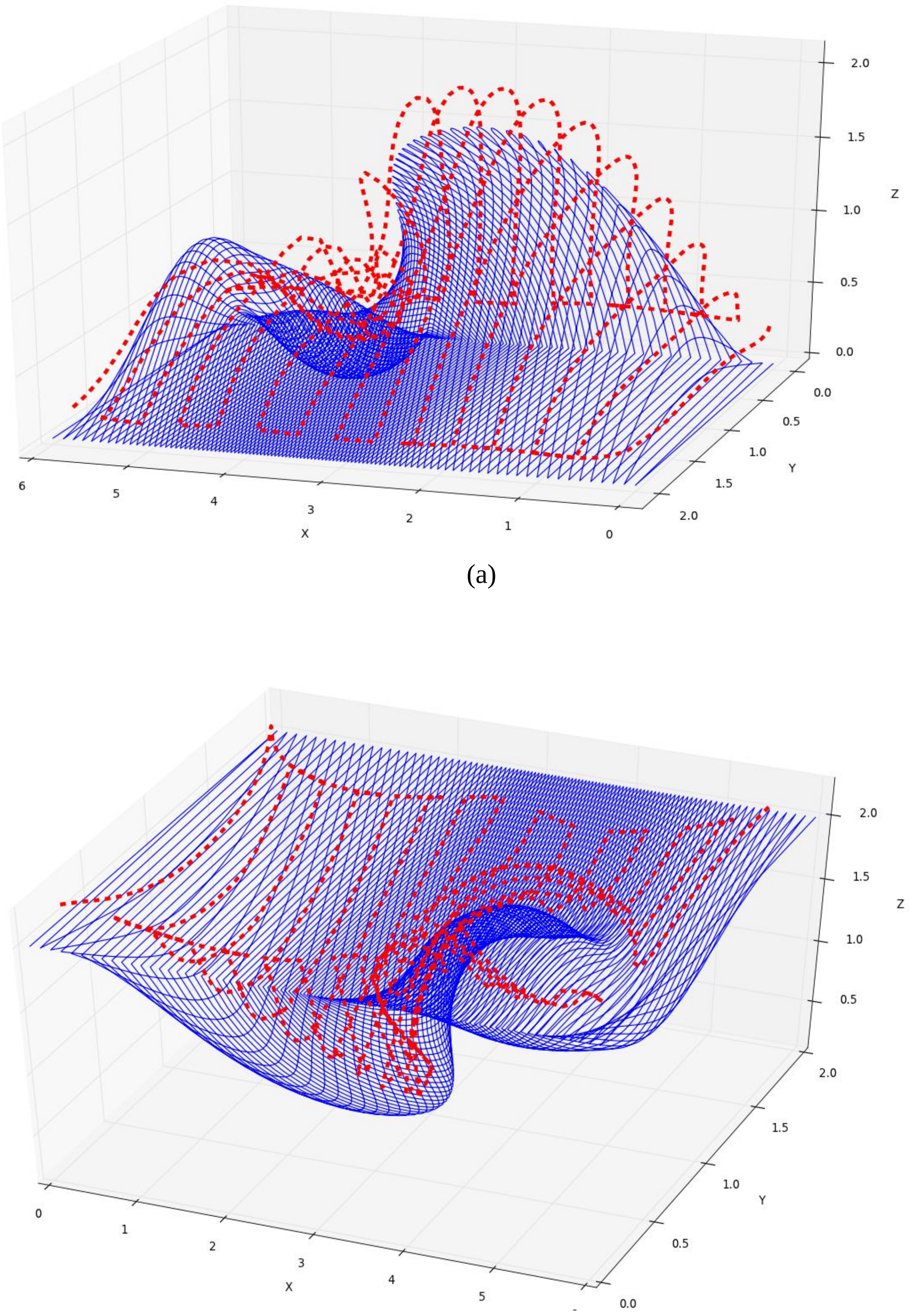
9.3. EXTREME

These are surfaces which have extreme curvatures. Our machining algorithm strategy would not fail, but would end in an infinite loop trying to find the collision free tool paths for such surfaces. Fig. 16. depicts one such example of a very complex freeform surface.

As shown, the tool will initially be in position 1 and would try to correct itself to avoid collision-1 area shown by red. After applying collision correction several times (represented by tool transition positions 2 and 3), the tool would achieve the collision free tool position 4.

However, in tool position 4, area collision-2 takes place. Thus, the collision correction will again be applied and now the tool will correct from tool position 4 to position 1.This is a typical case of Type I Error. Here the proposed algorithm succeeds in detecting collision and applying collision correction but, fails to decipher a finite solution.

This has its advantages in the fact that we avoid Type II Error, where the proposed algorithm would fail to detect the collision in such a situation. However, the limitation as mentioned earlier is that the machining algorithm would not be able to compute a finite solution for the tool orientation in such a case.



(a)

(b)

Fig.15: Efficient Tool Paths for Complex Surfaces. (a) Surface 1, (b) Surface 2.

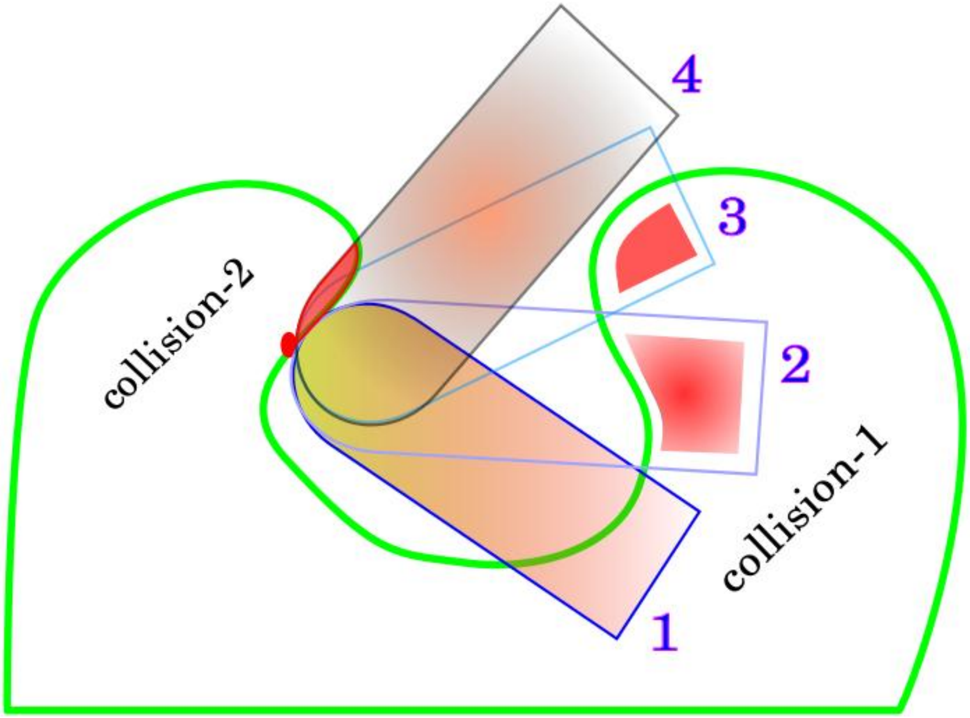


Fig.16: Tool Path Limitation for Extreme Surfaces.

10. CONCLUSION

This paper proposes a complete integration of various existing ideas such as tool path generation with scallop height method and collision detection using bounding box method.

Furthermore, this paper proposes a new approach to bounding box calculations to reduce the computational time. Thus, complete tool path planning software has been developed which incorporates a novel initial coarse & final fine screen technique of quick collision check along with the existing tool path planning strategies.

Thus, the contribution of this research to the existing body of knowledge is the novel idea of body diagonal bounding box which increases the computational efficiency in selecting the probable collision points in the initial stages of collision detection.

11. ACKNOWLEDGEMENT

This research is a result of the phenomenal support and guidance extended to me by various individuals at Missouri University of science & Technology. I would like to express my sincere thanks to my advisor, Dr. Frank Liou who has been a constant source of encouragement and guidance for me.

Secondly, I would like to thank Mr. Todd Sparks for his innovative ideas and programming skills that he shared with me during the course of my research. Also, I would like to thank Dr. Jianzhong Ruan for reviewing the various logic errors in my tool path planning software.

Finally, I would like to thank National Science Foundation and Intelligent Systems Center for supporting this research financially.

12. REFERENCES

[1] Feng, Hsi-Yung; Li, Huiwen: Constant scallop-height tool path generation for three-axis sculptured surface machining, Computer-Aided Design, 34(9), 2002, 647-654, doi:10.1016/S0010-4485(01)00136-1.

[2] Misra, Debananda; Sundararajan, V.; Wright, Paul K.: ZigZag Tool Path Generation for Sculptured Surface Finishing, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 2003.

[3] Lee, S. G.; Yang, S. H.: CNC Tool-Path Planning for High-Speed High-Resolution Machining Using a New Tool-Path Calculation Algorithm, International Journal of Advanced Manufacturing Technology, 2002.

[4] Yuan, Bo; Kwoh, L. K.; Tan, C. H.: Finding the Best-Fit Bounding Boxes, National University of Singapore, 2006, http://www.comp.nus.edu.sg.

2. CONCLUSION

A novel idea of Body diagonal Bounding Box was proposed and applied in this research. The final tool path generated is more efficient conceptually in terms of computational efficiency. This new idea has been implemented on wax blocks for final 5-axis machining using a ball-end mill. In future, this tool path generation software will be integrated with MAPS software which generates the G&M code for laser aided metal deposition system. This research has solved the problem of machining freeform surface and collision of tool with the neighboring surface. Future scope of the research can involve optimization of the process of selection of surfaces to be machined.

Furthermore, this research was successful in exploring the possible applications of this machining algorithm. It has been successful in identifying and classifying the various surfaces based on the complexity of machining to determine the feasibility of applying this algorithm.

VITA

Jomy Francis, son of Annie Francis and Francis Jacob, was born in Mumbai, India on January 19th, 1987. He received his B.E. in Mechanical Engineering from Mumbai University, Maharashtra, India in July 2008. In August 2009, he joined Missouri University of Science and Technology to pursue his Masters in Manufacturing Engineering. Jomy worked as a Graduate Research Assistant in the Laser Aided Manufacturing Processes Lab at Missouri S&T. His research was focused on tool path generation for 5-axis freeform surface machining. Jomy got his Masters in Manufacturing in May 2012. During his Masters, he interned with Torotel Products, Inc at Olathe, Kansas during the summer of 2010. He was employed with Green Bay Packaging, Inc, at Morrilton, Arkansas as a Mechanical Project Engineer Co-op during the spring and summer of 2011.

Jomy has published a conference paper in the proceedings of the Solid Freeform Fabrication Symposium in summer 2010. He was awarded I place for Research Poster Presentation during the Intelligent Systems Center Research Showcase in fall 2011.