TOOL PATH PLANNING FOR 3-AXIS NC-MILLING LATHE AND 3-AXIS NC-VERTICAL MILLING FOR SCULPTURED SURFACES MACHINING USING TRIANGULAR MESH OFFSET

A Thesis
submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
CAD/CAM & ROBOTICS ENGINEERING

Submitted by:
VIVEK PAL SINGH
Roll No: 80781025

Under the guidance of

Dr. Ajay Batish
Associate Professor, MED
TU, Patiala-147004

Mr. A. S. Jawanda
Assistant Professor, MED
TU, Patiala-147004

Mr. Ravinder K. Duvedi
Lecturer, MED
TU, Patiala-147004

MECHANICAL ENGINNERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

2009

CERTIFICATE

This is to certify that the thesis titled, **"Tool Path Planning for 3-Axis NC-Milling Lathe and 3-Axis NC-Vertical Milling for Sculptured Surfaces Machining using Triangular Mesh Offset"**, being submitted by **Mr. Vivek Pal Singh**, Registration No. 80781025, in partial fulfillment of the requirements for the award of degree of **Master of Engineering** in **CAD CAM & ROBOTICS ENGINEERING**, submitted in **Mechanical Engineering Department** of **Thapar University, Patiala**, is a bonafide work carried out by him under our supervision and guidance and no part of this thesis has been submitted to any other university or institute for award of any degree.

Dr. Ajay Batish
Associate Professor, MED
TU, Patiala-147004

Mr. A. S. Jawanda
Assistant Professor, MED
TU, Patiala-147004

Mr. Ravinder K. Duvedi
Lecturer, MED
TU, Patiala-147004

(Dr. S. K. Mohapatra)
Professor & Head, MED
TU,Patiala

(Dr. R. K. Sharma) 2013
Dean of Academic Affairs
TU, Patiala

ACKNOWLEDGEMENT

I express my sincere gratitude to my guide **Dr. Ajay Batish, Associate Professor, Mr. A. S. Jawanda, Assistant Professor and Mr. Ravinder K. Duvedi, Lecturer Mechanical Engineering Department, Thapar University, Patiala,** for their valuable guidance, proper advice and constant encouragement during the course of my work in this thesis.

I do not find enough words with which I can express my feeling of thanks to the entire faculty and staff of **Mechanical Engineering Department, Thapar university, Patiala,** for their help, inspiration And moral support which went a long way successful completion of this thesis.

(VIVEK PAL SINGH)

ABSTRACT

Since the evolution of human civilization, humans are making sculpture on wood, clay and rocks. Even now rural artists like wood maker are making designed sculpture surfaces by hand. Today machining relies heavily on CNC machine for conventional job, but no machines are made for sculptured surfaces. Though there are some CNC machines specially for machining sculptured work their cost are way beyond the limit of artists. The present work involves methodology of tool path generation by surface offset method for cheap and efficient NC-machine which can be used by artists easily without learning the detailed technical.

CNC machines have controllers which use ISO G-code programming language or some machine specific proprietary languages which do not contain any part geometry information. The used NC machine reduces the requirement of costly controller by using computer as a controller. The present work involves **Tool path planning for 3-Axis NC-Milling Lathe and 3-Axis NC-Vertical Milling for sculptured surfaces machining using triangular mesh offset.** The methodology used in this work is for such PC based NC machine which takes the part geometry information as CL point in Cartesian coordinate.

A unique surface offset algorithm is used to generate tool path surface by offsetting the surfaces by tool radius. The center of the ball nosed milling cutter travel along the offset surface where the offset surface acts as a CL point and the center of the ball nosed milling cutter act as CC point. The offsetting is done for both rouging pass and finish pass with different tool radius respectively.

Once the surface is offset then tool path planning for milling lathe and vertical milling machine is performed by ray intersection methodology separately for rouging and finishing pass with variable pitch assuming the milling cutter as a ray which intersect the offset STL surface to generate the cutter location points the CL points are arrange systematically to generate tool path for milling lathe and vertical milling. The computer algorithm has been developed and verified by presenting in Cartesian coordinate format in MATHCAD® graphics representation. The machining operation for various types of models like twisted square, simple cylinder, cube, ring and some design like snake wrapped on cylinder has been performed successfully and presented in the report.

NOMENCLATURE

CC : Cutter Contact point

CL : Cutter location point

Nx, Ny, Nz : Normal vectors

X, Y, Z : Cartesian Coordinates

N_{Vi} : Normal of offset vector

V_i : Vertex

$N_{i,j}$: Facet normal

V_i' : Offset vertex location

α, β : Barycentric coordinates

Xp, Yp, Zp : Point of intersection of triangle with ray

U : Parametric value

$P1$: Source point location

$P2$: Destination point location

$Umax$: Maximum value of U parameter

$Umin$: Minimum value of U parameter

STL : Stereolithography

CAD : Computer aided design

CAM : Computer aided manufacturing

STEP : Standard for the exchange of product model data

IGES : Initial graphics exchange specification

ASCII : American standard code of information interchange

CNC : Computerized numerical controlled machine

INDEX

CONTENTS **Page No.**

CERTIFICATE ii

ACKNOWLEDGEMENT **Error! Bookmark not defined.**

ABSTRACT iii

NOMENCLATURE v

Chapter 1: INTRODUCTION 1-4

1.1 CNC Machining Process 1

1.2 Direct Machining and Controlling (DMAC) 2

1.3 Single Controlled Axis Lathe Mill (SCALM) 3

1.4 Proposed work 3

Chapter 2: LITERATURE REVIEW 5-16

2.1 Tool path generation methods 5
- 2.1.1The Offset Surface method 5
- 2.1.2 The Isoparametric method 5
- 2.1.3 The Cartesian method 6
- 2.1.4 Feed-forward and side-steps planning 6

2.2 Tool positioning methods 6

2.3 Errors in NC tool path planning 8

2.4 Surface offset method 8
- 2.4.1 Offset of faces 8
- 2.4.2 Offset of vertices 9

2.5 Surface offset methods 10
- 2.5.1 Input required for triangular mesh offset 10

2.6 Tool path planning for sculptured surfaces 13

2.7 Problem Definition 16

Chapter 3: METHODOLOGY FOR SOLUTION 17-28

3.1 Tool path planning for sculptured surface for milling lathe 17
  3.1.1 Specification of 3-Axis NC Milling Lathe machine 17
  3.1.2 Parameter required for tool path planning 18
  3.1.3 Surface offset methodology 19
    3.1.3.1 Average surface normal method for vertex offsetting 19
  3.1.4 Ray intersection method 21
    3.1.4.1 Triangle check 21
    3.1.4.2 Edge check 22
    3.1.4.3 Vertex check 22
  3.1.5 Tool path planning for rough and finish pass 23

3.2 Solution procedure for Tool path planning for milling lathe 24

3.3 Tool path planning for sculptured surfaces for 3 Axis NC vertical milling machine 26
  3.3.1 Raster milling 26
    3.3.1.1 Specification of 3-Axis NC milling machine 26
    3.3.1.2 Parameter required for raster milling 27
  3.3.2 Circular milling 27
    3.3.2.1 Specification of 3-Axis NC milling machine 27
    3.3.2.2 Parameter required for tool path planning 28

3.4 Solution procedure for NC tool path planning for milling 28

Chapter 4: RESULTS AND DISCUSSION 30-44

4.1 Validation 30
  4.1.1 Validation for 3 Axis NC milling lathe 30
    4.1.1.1 Validation with CAD model 30
    4.1.1.2 Comparison with actual CAD model 32

4.2 Validation for vertical milling 35
  4.2.1 Validation with CAD model 35
    4.2.1.1 Raster milling 35
    4.2.1.2 Circular milling 36

4.1 Results 38

4.2 Errors and Limitations 40

4.3 Future scope 422

4.4 Conclusion 43

REFERENCES 44

LIST OF FIGURES

Figure 1.1: Conventional method for tool machining 2

Figure 1.2 : CAD drawing of the single controlled axis lathe mill (SCALM) with axes highlighted. 3

Figure 2.1: Overcut and Undercut in NC machining 8

Figure 2.2: (a) Gap between offset surfaces; (b) Intersection between offset surfaces 9

Figure 2.3 : Offsetting vertices 10

Figure 3.1: 3 Axis NC Milling Lathe Machining Coordinates 17

Figure 3.2: Cutter contact and Cutter location path for offset surface 18

Figure 3.3 : .tp file format 18

Figure 3.4: Calculating the averaged normal vectors $\boldsymbol{Nvi}$ at a vertex: (a) Calculating the vertex normal $\boldsymbol{Nvi}$ by averaging the facet normals; (b) Incorrect normal vector $\boldsymbol{Nvi}$ shifted $\boldsymbol{Fi,1}$ and $\boldsymbol{Fi,2}$; (c) Corrected normal vector $\boldsymbol{Nvi}$ by deleting the parallel normal vector 20

Figure 3.5: Barycentrics coordinates of point P 22

Figure 3.6: Tool path for roughing passes with depth of cut 23

Figure 3.7: Flow chart of tool path planning for milling lathe 25

Figure 3.8: 3 Axis NC Raster Milling machine 26

Figure 3.9: 3 Axis NC circular Milling machine 27

Figure 3.10: flow chart for vertical milling 29

Figure 4.1: Actual STL 30

Figure 4.2 : Difference between actual and offset surface of 2mm 31

Figure 4.3: CC surface and CL surface with tool axis 31

Figure 4.4: Roughing and Finishing tool path 31

Figure 4.5: (a) actual STL (b) offset surface with 6.35mm (c) offset surface with .7935mm 32

Figure 4.6: (a) first rough pass (b) third rough pass (c) final rough pass 32

Figure 4.7: (a) rough tool path (b) finish tool path 33

Figure 4.8: (a) and (b) Difference in CL surface for circular portion for finish pass with actual surface with radius 25mm 34

Figure 4.9: comparison of actual CAD with CL surface of square of 60x60mm dimension for finish pass 34

Figure 4.9: Actual STL 35
Figure 4.10: (a) offset surface with 6.35mm radius (b) roughing tool path 35
Figure 4.11: (a) offset surface with .7395mm radius (b) finish tool path 36
Figure 4.12: (a) Roughing passes with depth of cut (b) sectional view 36
Figure 4.13: Actual STL 36
Figure 4.14: (a) offset surface with 6.35mm radius (b) roughing tool path 37
Figure 4.15: (a) offset surface with .7395mm radius (b) finishing tool path 37
Figure 4.16: Sectional view of circular Tool path for all passes 37
Figure 4.17: (a) Actual STL, (b) offset surface with 6.35mm (c) offset surface with .7935mm 38
Figure 4.18: Finishing and roughing tool path 38
Figure 4.19: (a) Finish Part (b) rough Machined part (c) triangular patch 39
Figure 4.20: All rough passes with depth of cut and finish pass tool path 39
Figure 4.21: (a) actual STL (b) machined part (c) finish tool path 40
Figure 4.22: (a) shows actual part undercutting Section view of offsetting a node on a solid edge: (b) convexity; (c) concavity. 41
Figure 4.23: Illustration of vertex splitting method. (a) Normal offset, and (b) offset with vertex splitting 42
Figure 4.24: Generalized APT cutter 43

Chapter 1

INTRODUCTION

Today manufacturing relies heavily on CNC machines. The operations of CNC machines requires modeling a part in the Computer Aided Design (CAD) package, and subsequently planning the tool path by processing the CAD data into a part program with the Computer Aided Manufacturing (CAM) package. The commercial CAM packages use inbuilt tool positioning strategies to generate tool path plans in the ISO G-code programming language or some machine specific proprietary languages. These machine instructions command the tool movements and control all the devices on the CNC machines during the machining process. The program online contains the tool movement information and has no access or information to the part geometry.

The purpose of this chapter is to familiarize the readers with the need and the 3 Axis NC-Milling Lathe for which the tool path planning will be developed.

1.1 CNC Machining Process

The CNC machining paradigm has not changed since its inception. The machine motions are dictated by the part program that contains the tool path. Tool path generation is done prior to the machining process and the tool path is usually generated offline with a commercial CAM package. Different commercial CAM packages use different tool path generation algorithms depending on the final machining accuracy and required surface finish. The CAD model data and the machining parameters, such as the choice of cutting tool, feed rate, and spindle speed, are generally the inputs to the tool path generator and are supplied by the operator. The tool path generator follows the machine tool trajectory and computes a list of tool positions for the machine to interpolate between. Most machines offer linear and circular interpolation capability, however, some new machines offer spline interpolation as well. The tool positions are considered as the instantaneous tool locations with respect to the machine coordinate system or work coordinate system.

Commands that will instruct a specific machine to follow the tool paths that were created during the CAD/CAM phase of design. This is generally a two step process. The first step converts the tool paths to machine independent commands that are stored in a file using the Automatic Programmed Tools (APT) format. This file is then read by a post-processor that converts the APT commands to machine control data for a specific controller. The resulting file consists of geometry and motion commands in the form of line commands commonly known as G and M code.

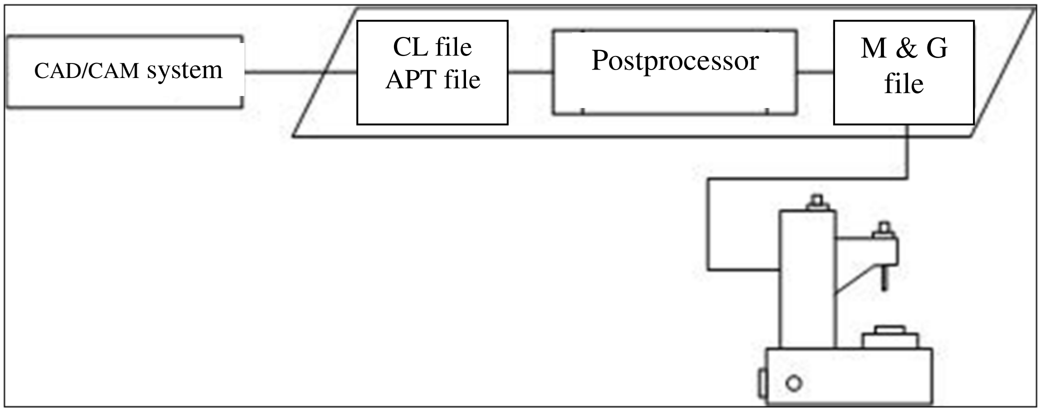


Figure 1.1: Conventional method for tool machining

G&M codes vary slightly among individual types of machine controllers. The variation requires that a post-processor corresponding to each different machine controller be developed. If a manufacturing plant has many different machine tools, a large number of post processors must be made available for use by the process design software.

But generating G & M code for complex sculptured surface will be very difficult for conventional CNC machines. So an efficient but cheap machine which is proposed by Manos et al [6] as a SCALM machine is needed.

1.2 Direct Machining and Controlling (DMAC)

Direct machining and control (DMAC) has provided a unique solution to the shortcomings of typical machine tool controllers described above. The DMAC controller is software-based and very flexible. The first major difference with the DMAC controller is the method of motion input. In place of the usual M&G style commands, the DMAC controller is given entity path information. This can be in the form of position target moves, line moves, arc moves, or NURBS moves. As a result, CAM packages are not required to tessellate complex curves into miniscule arcs and line segments. This results in improved path accuracy, and provides the controller with complex path planning and blending algorithms.

1.3 Single Controlled Axis Lathe Mill (SCALM) [6]

Single Controlled Axis Lathe Mill (SCALM) presents a method for machining complex three-dimensional surfaces using only one axis of controlled motion to position cutting tool on a specially designed numerically controlled (NC) machine. This single controlled axis lathe is configured like a lathe, but is used to produce complex sculptured surfaces out of wood. This is accomplished by mechanically linking two axes of motion to produce a fixed helical footprint of a tool path with constant step-over distance. As the linked axes are rotated, their location is measured by an encoder and passed directly to a personal computer (PC). Software running on the PC determines the depth of the computer controlled axis. The depth information is used to control the depth axis.

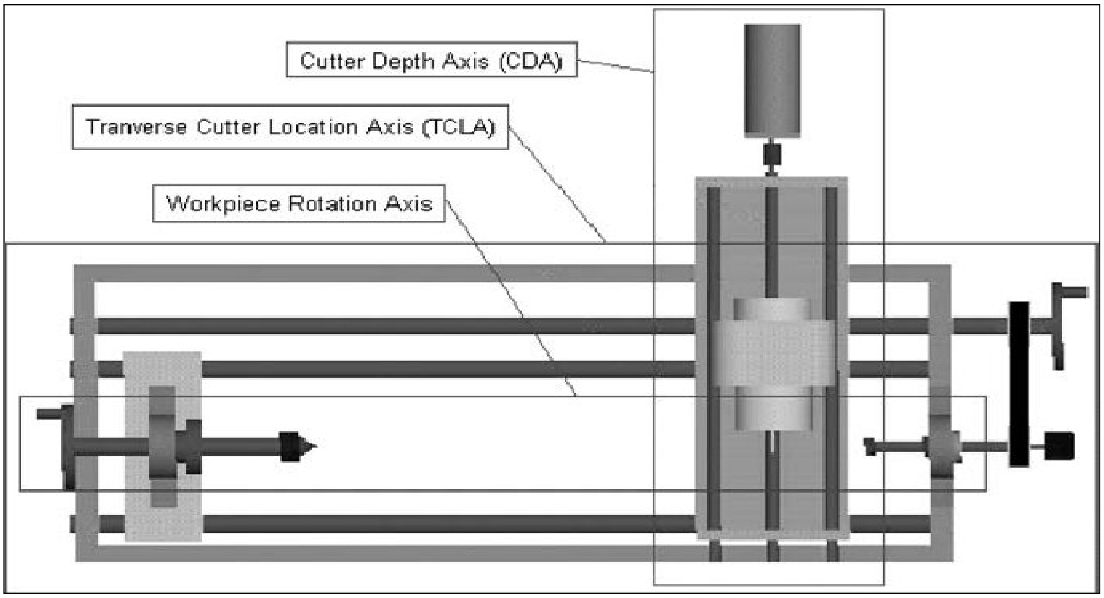


Figure 1.2 : CAD drawing of the single controlled axis lathe mill (SCALM) with axes highlighted.

The tool path generation by Surface Offset methodology has been developed for PBG KW 2048[30]. PBG KW 2048 is the advance version of SCALM with the flexibility of changing the pitch, means the workpiece rotation and the traverse cutter location Axis are not coupled together so time and quality of machining is increased.

1.4 Proposed work

The present work involves **Tool path planning for 3-Axis NC-Milling Lathe and 3-Axis NC-Vertical Milling for sculptured surfaces machining using triangular mesh offset.** The main objective is to study surface offset method for vertex offset and tool path planning methodology for sculptured surfaces that are broadly of two types, one that is machined by milling lathe and other machined by vertical milling machine.

Computer algorithms has been generated in visual C++ for Tool path planning for both kind of machining system and validation has been done by software and actual machining of part on milling lathe, however only software validation has been done on vertical milling machine.

Error and limitation of method is discussed and finally conclusion is made.

Chapter 2

LITERATURE REVIEW

There are many tool path generation methods available which are analyzed below in detail:

2.1 Tool path generation methods

The conventional tool path generation method used in the industry is to specify the cutter contact point on the part surface and then offset that point to yield the cutter location. The cutter contact point (CC) is the location where the tool touches the part surface. The cutter location (CL) is the location of the center of the tool. There exist a number of tool path generation methods that are popular in industry. Some of the common ones are isoparametric, Cartesian, offset surface, feed forward, and side-steps method. In addition to describing a tool's route, the tool path generation methods must also guarantee the deviations between the desired and the part surfaces to be minimal.

2.1.1The Offset Surface method

The offset surface method is conceptually similar to the Cartesian method. The offset method generates the tool path by offsetting the part surfaces by the tool radius. The center of the cutter tool travels along the offset surface to machine the part, and the tool path is calculated by identifying tool passes on the offset surface. However, self intersections tool path that leads to over-cut or cavities of under-cut must be detected and corrected while performing the offset surface algorithm [3]. An advantage of this method is one can find a tool trajectory parallel to X axis in which the tool moves only along the Y and Z axis [3].

2.1.2 The Isoparametric method

The isoparametric method is one of the simplest tool path generation algorithms in which the cutter contact points are specified along isoparametric lines on the part surface. Isoparametric lines are curves of constant parameter value on parametric surfaces. The isoparametric curves are approximated by linear segments. However, if the linear segments are large, it may results in under-cuts on the sculptured surfaces of the model [1].

2.1.3 The Cartesian method

In contrast to determining the tool movements based on the part surface’s parameters, the Cartesian method allow the operators to generate tool paths with respect to the global X, Y, Z Cartesian coordinates. The tool path generator takes the X, Y coordinate of the cutter location as its input and computes the Z-value of the cutter location. The instantaneous tool location is also checked for gouging. This method is more difficult to implement for parametric curved surfaces when compared to isoparametric method. This is due to the complexity of the computational relationship between the cutter contact points in the global X, Y, Z coordinates and the part surface coordinate [2].

2.1.4 Feed-forward and side-steps planning

Feed-forward and side-steps planning are also important in tool path generation as the machine movements is discretized into finite piecewise motions. Huang et al. [4] discussed parameterizing the surface to determine the step size while maintain the machining errors within a desired tolerance. The linear feed-forward tool motion between two tool positions is used to determine the deviation between the actual straight line tool paths and the desired surface. The step-over increments, which are the distance between the adjacent tool paths in the side-step direction, are used to determine the height of cusps that remain after the machining process. Thus, the tool path is basically recognized as a set of tool positions for the tool to traverse and interpolate into a smooth path movement. Hence, the method of determining the tool positions is a factor that can affect the finish quality of the machined parts.

2.2 Tool positioning methods

The CNC machine's tool movement is guided by a series of interpolated tool positions, whereas the parts to be machined could be represented by elements with varying slopes and curvatures. Thus, the method of representing the sculptured surfaces is related to the accuracy of the tool positions and the machined part. CAD systems have adopted parametric methods such as NURBS and B-splines as standards for designing sculptured surfaces. Isoparameteric lines on parametric surfaces can fan out from one end to the other making them convenient for machining as they result in over-machining in some areas [1] and consume unnecessary machining time. An alternate point-by-point based tool positioning approach is commonly used for generating tool positions by offsetting a desired distance along the surface normal at a given location to determine the tool contact point. This method is similar to the offset surface method for tool path generation other than it is applied to a discretized surface. This method is used for tool positioning with ballnosed endmills. A variation of this method is to project the tool normal onto the curve surfaces and determining the tool contact point along the projected surface normal. This method is used for flat end milling cutter. A combination of the two methods is used for toroidal or bullnose endmills. These offset methods are commonly used in determining gouge-free tool positions while the tool is tangent to a point on the surface [5]. The accuracy of the tool positioning methods relies significantly on the part surface representation. In order to develop a tool path independent of surface representation method, a solution adopted in industry is to use triangulated surface representation.

Triangulated surfaces are formed by connecting the discretized points on the part surface with vectors in three-dimensional Cartesian space. The vectors are grouped to form triangular facets to represent an approximation of the desired surface. Increasing the number of points, improves the accuracy in approximating the desired surface. Since all surfaces are linear, tool positioning is simple. This provides a simple and unified approach for machining all type of surfaces with different representations. The use of triangulated surface models in machining is not new: Saito used a computer graphics-based approach for 3-axis surface machining of triangulated surfaces for ballnosed and flat-endmill.

The use of triangulated surface CAD model in tool positioning follows the same steps as any other tool positioning methods. First, the tool path footprint is specified. Next, at each point along the tool path, the triangulated model of the tool is dropped along the tool axis towards the part and the contact point between the tool surface and the triangulated model surface is analytically computed [2]. The direct projection of the tool surface onto the part surface along the tool normal creates an instantaneous tool shadow area where the desired contact point is to be calculated. Under this shadow, the first point of intersection between the tool surface and the part surface is the desired contact point. A series of contact points at regularly pre-determined intervals can produce the tool path. This method is useful in generating the tool positions for three-dimensional models with irregular shapes, and do not require data converters or geometry engines for remodeling the sculptured surface during tool position generation.

2.3 Errors in NC tool path planning

In general, two types of cutting errors may occur in NC tool path generation: **over-cut (gouging)** or **undercut** as shown in Figure 2.1. Among them, undercut is a relatively less difficult problem since it can always be resolved by a smaller cutting tool with extra cutting passes. Gouging is, however, a difficult problem which frequently occurs in the solid-based multiple surface machining environments.

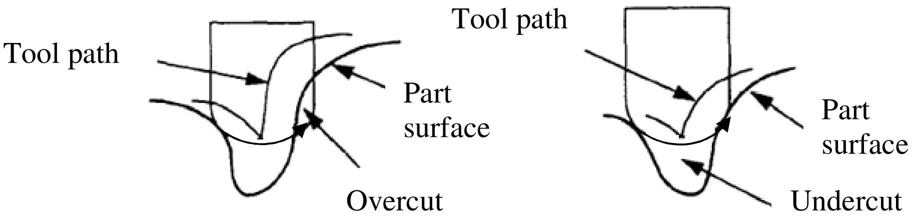


Figure 2.1: Overcut and Undercut in NC machining

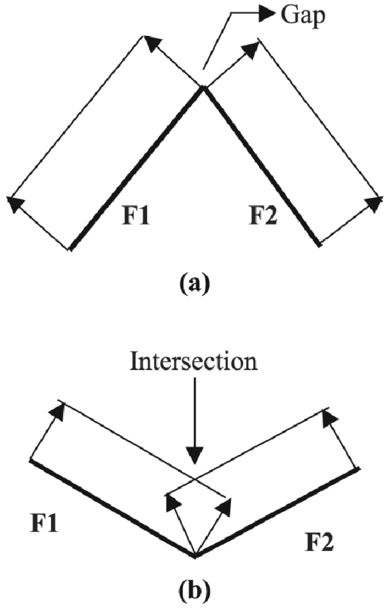
In this thesis, a different approach is provided for solving the gouging problem in multi-surface machining. The scheme proposed is based on the offset surface approach and its scope is limited ball nosed milling cutter NC machining. The heart of the scheme is the offsetting of surface boundary curves. Conceptually, the offset of a 3D curve can be regarded as the envelope of the swept volume generated by moving the inverted tool along the curve. By introducing the offset surfaces around the surface boundaries and incorporate them with the upper envelope operation, this is shown that the resulting tool paths of the offset surface approach are gouge free.

2.4 Surface offset method

Offsets are widely used in many applications. These include tool path generation for 3D NC machining, rapid prototyping, hollowed or shelled model generation, and access space representations in robotics. These are of broadly two types given below:

2.4.1 Offset of faces

The most direct method would be to offset each triangular facet with the given offset distance in its corresponding normal direction. However, this will result in intersections or gaps between the offset surfaces of two neighboring triangles. As shown in Figure 2.2(a), a gap is formed between two offset surfaces F1 and F2 when the angle between them is convex. Conversely, an intersection or overlap occurs between offset surfaces, as shown in Figure 2.2(b), when the angle between them is concave. In order to make closed 3D models from these triangular offset surfaces, it is necessary to identify all the intersections, and then trim the surfaces on the line of intersection, and to identify all the gaps and extend the surfaces to fill them. This can be quite complex, since thousands or millions of triangular facets may exist when representing complex 3D models using the STL format.



(a)

(b)

Figure 2.2: (a) Gap between offset surfaces; (b) Intersection between offset surfaces

2.4.2 Offset of vertices

This problem can be avoided if the vertices, instead of the triangular facets, are offset. As shown in Figure 2.3, when offsetting the vertices the relationship between facets will remain and there is no need to recalculate the triangle intersections. Since a smooth vertex with the single normal vector is moved along one direction, no gap or local interference occurs between the faces that surround the vertex. The challenge when utilizing this method is how to effectively calculate the offset vector for each vertex, taking into account the offset direction and magnitude, from all of its surrounding triangular facets.

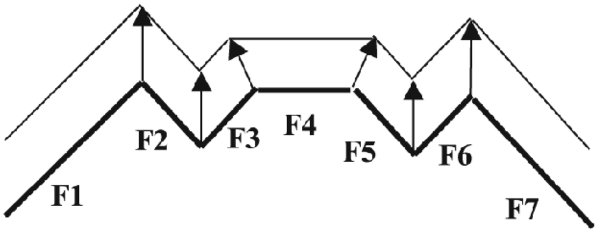


Figure 2.3 : Offsetting vertices

The following section details the literature available and relevant to the present study of surface offset and tool path planning, since using triangular mesh offset for tool path generation for milling lathe is a new area but literature for triangular mesh offset and tool path generation are as follows:

2.5 Surface offset methods

2.5.1 Input required for triangular mesh offset

For surface offset method STL file format is used which can be saved easily from most of the CAD modeling software such as PRO-E, CATIA, and SOIDWORKS etc. Since data is in a definite order as shown below the data extraction is easy as compared to other neutral format like STEP, IGES etc.

STL (stereolithography) format is rather conceptually simple and sufficiently accessible as it repetitively describes every normal and vertex of triangular facets built for object approximation. Facet data are saved in computers in two types of data format: text (ASCII) format and binary format. The text STL format is a set of facet descriptions in the form of ASCII containing its unit normal vector and 3D coordinates of three vertices. An STL file in text format is obviously redundant for computer storage as it records every character and digit of items. Although the content of a text STL file is readable, its file size is so large that it is generally used as a testing tool. The structure of text STL format is as follows:

```
solid solid_name
        <facet list>
        facet normal Nx Ny Nz
        outer loop
                vertex X1 Y1 Z1
```

```
        vertex X2 Y2 Z2
        vertex X3 Y3 Z3
    endloop
endfacet
...
endsolid
```

Where (Nx, Ny, Nz) is a normal vector, and $(X1, Y1, Z1), (X2, Y2, Z2)$ and $(X3, Y3, Z3)$ are coordinates of vertices. Unlike text STL format, binary STL format is more compact and therefore more efficient for data processing because vectors and coordinates are saved as floating-point numbers, each of which occupies four bytes of computer memory. The binary STL format contains file head, facet number and facet list. STL is widely accepted by most commercial CAD software and rapid prototyping equipment due to the obvious advantage of its topologically simple and robust nature. STL format is composed of only one type of element, a triangular facet, which is defined by its normal and three vertices. All the triangular facets described in a STL format file constitute a triangular mesh to approximate modeling surfaces.

However, drawbacks exist in STL format. Flaws may appear in the process of creating triangular facets, although they can be checked and corrected afterwards. Incorrect normal and inconsistent normal are two cases of inconsistency problems in STL. The former problem happens when the facet normal generated by the CAD system is different from that calculated from facet vertices, and the latter is owing to inconsistent orientation of the normals of adjacent triangular facets. Another kind of flaw is malformation, for instance, once an STL triangular facet is too thin to keep its triangular shape, it may collapse to be a gap, crack or hole. Illegal overlap is the third type of problem. When a facet vertex is located on the edge of another facet or when two facets intersect with each other, the two facets are partly overlapping, which breaks the STL rule that each triangular facet must share two vertices with every adjacent facet.

In addition, STL has some disadvantages in both its format and applications. Redundant depiction of geometric elements in STL format, i.e., each vertex of a triangular facet is recorded at least four times brings extra computational memory occupation and time consumption. Another shortcoming is that STL file size is incommensurate with its approximation accuracy. When the required approximation accuracy and pronounced curvature of an object surface increases in case of complex surface and, the size of the generated STL file is dramatically enlarged. Finally, STL format records only the geometry of object surface and lacks object attributes.

STL format is widely used as a de facto industry standard in the rapid prototyping industry due to its simplicity and robustness. However, on account of its shortcomings and inadequacy in applications, many interface alternatives have been brought forward. Wu et al [9] proposed a new scheme to enhance the approximation accuracy and to extend functions of STL by means of introducing additional feature and attribute codes into STL format. The geometry feature code describes a tetrahedron based on the STL triangular facet, which provides better approximation to the object surface covering. The attribute code attaches attributes of object surfaces such as colours and markers to STL triangular facets. Moreover, the enhanced STL also shares the structure of binary STL format by filling feature and attribute codes into its blanks, and therefore is compatible with STL. Compared with STL, the enhanced STL provides not only higher accuracy with the same file size and compatible format, but also colour and marker functions for rapid prototyping.

Literatures available on triangulation mesh offset are as follows:

A shell map [10] is a bijective mapping between shell space (the space between a base surface and its offset) and texture space. It can be used to generate small-scale features on surfaces using a variety of modeling techniques. In this paper, an efficient algorithm which reduces distortion by construction, for the offset surface generation of triangular meshes is given. The basic idea is to independently offset each triangle of the base mesh, and then stitch them up by solving a Poisson equation.

Qu and Stucker [11] presented a new 3D offset method for modifying CAD model data in the STL format. In this method, vertices, instead of facets, are offset. The magnitude and direction of each vertex offset is calculated using the weighted sum of the normals of the facets that are connected to each vertex. To facilitate the vertex offset calculation, topological information is generated from the collection of unordered triangular facets making up the STL file. A straightforward algorithm is used to calculate the vertex offset using the adjoining facet normals, as identified from the topological information. This newly developed technique can successfully generate inward or outward offsets for STL models.

Kim and Yang [12] introduced a new method for offsetting triangular mesh by moving all vertices along the multiple normal vectors of a vertex. The multiple normal vectors of a vertex are set the same as the normal vectors of the faces surrounding the vertex, while the two vectors with the smallest difference are joined repeatedly until the difference is smaller than allowance. Offsetting with the multiple normal vectors of a vertex does not create a gap or overlap at the smooth edges, thereby making the mesh size uniform and the computation time short. In addition, this offsetting method is accurate at the sharp edges because the vertices are moved to the normal directions of faces and joined by the blend surface. The method is also useful for tool path generation if the triangular mesh is tessellated part of the solid models with curved surfaces and sharp edges.

Malosio et al [13] described a new geometric algorithm to offset CAD objects, described as surfaces tessellated with triangular facets, transforming the original geometry in a new smoothed offset model. Different approaches to cope with convex and concave geometries and to prevent overlapping cases are suggested and investigated. Furthermore, an STL-file preprocess algorithm is proposed in order to obtain an errors-free final surface modifying starting tessellated model. The developed algorithm has been named Offset Weighted by Angle (OWA).

2.6 Tool path planning for sculptured surfaces

OuYang and Feng [14] presented a new method to generate iso-planar numerical control (NC) tool paths for the finishing machining of triangular mesh surfaces. One main concern in generating the piecewise linear NC tool paths is to ensure that their resulting machining errors are within the specified tolerance. it is also proposed that the cutter location (CL) tool paths be within the 3D tolerance zone defined by two offset surfaces of the triangular mesh surface. One offset surface corresponds to the under-cut limit and the other offset surface corresponds to the over-cut limit. Also, the scallop-height offset surface is used to facilitate the determination of side steps between adjacent iso-planar tool paths. The common self-intersections in the offset triangles are eliminated using the Z-map approach. The applicability and effectiveness of the presented method was validated through implementations on typical triangular mesh surfaces.

In 3-axis NC (Numerical Control) machining, various cutters are used and the offset compensation for these cutters is important for a gouge free tool path generation. Kim and Yang [15] introduced triangular mesh offset method for a generalized cutter defined based on the APT (Automatically Programmed Tools) definition or parametric curve. An offset vector is computed according to the geometry of a cutter and the normal vector of a part surface. A triangular mesh is offset to the CL (Cutter Location) surface by multiple normal vectors of a vertex and the offset vector computation method.

A key issue in the creation of error-free tool path for numerically controlled (NC) surface machining is gouging (over-cut prevention. In the case of solid-based machining, where the creation of tool paths across several surfaces in a single pass is imperative, the major sources for gouging are the tangent discontinuity (C' discontinuity) and the surface gap (C_0 discontinuity) occurred in the constituent surfaces of the part model. C_1 discontinuity is naturally identified in solid modeling as 'edges' and 'vertices'. C_0 discontinuity occurs, however, in the free edges of surfaces or from errors in surface definition.

Tang et al [16] proposed a system based on offset or upper envelope concept for solving the C_0 and C' discontinuity in 3-axis multi-surface NC machine. Essentially, in addition to offsets of surfaces themselves, offsets are also defined for the boundary curves of these surfaces. By incorporating the upper enveloping operation on the intersection curves between the drive planes and the offsets, it is shown that the resulting tool paths have pleasant features of gouge free, smoothness, and minimal uncut area.

Koca and Leeb [17] presented a new method of using non-uniform offsetting and biarcs offsetting to hollow out solid objects or thick walls to speed up the part building processes on rapid prototyping (RP) systems. Building a hollowed prototype instead of a solid part can significantly reduce the material consumption and the build time. A rapid prototyped part with constant wall thickness is important for many different applications of rapid prototyping. To provide the correct offset wall thickness, we develop a non-uniform offsetting method and an averaged surface normals method to find the correct offset contours of the stereolithography (STL) models. Detailed algorithms are presented to eliminate self-intersections, loops and irregularities of the offsetting contours. Biarcs offsetting is used to generate smooth cross-section boundaries and offset contours for RP processes.

On the basis of STL, Bu et al [18] introduced a new topological structure, and proposed the vertex-based entity offset algorithm in order to realize the rapid generation of roughing/finishing tool path for molar prosthesis. Indicated by simulated machining, the proposed algorithm proves excellent stabilization, fast calculation speed and high machining accuracy.

Kim et al [19] presented a method where the 5-axis tool path that has been generated on the cutter contact (CC) surface is generated on the cutter location (CL) surface, and the CL surface deformation approach that inversely deforms the 3-axis tool path generated on the deformed CL surface to a 5-axis tool path is introduced. The CL point computation and interference check based on the CL surface is faster and more robust than that based on the CC surface. The proposed CL surface deformation approach can be used if the orientation of the cutter is predefined. By the CL surface deformation approach, the 5-axis tool path generation time can be reduced to that of a 3-axis, since the complexity of a CL surface deformation is linear and because the 3-axis tool path generation and gouge removal algorithms are used at the deformed CL surface.

Qu and Stucker [21] presented a paper on an offset-based tool path generation method for STL format three-dimensional (3D) models. The created tool-paths can be effectively used to near-net-shaped parts, in particular those created using rapid prototyping. In their approach the STL model is first offset by the distance of the selected cutter radius using a unique 3D offset method. The intersections between the top facing triangles of the offset model and tool-path drive planes are calculated. The intersection line segments are sorted, trimmed and linked to generate continuous top envelope curves, which represent interference-free tool paths. The developed offset-based algorithm can rapidly and successfully generate interference-free tool paths as continuous lines, instead of a collection of discrete tool location points. The strategy of using adaptive step-over distances based on local geometrical information can significantly increase machining efficiency. Limitations of their work are that the current tool path generation method only works for ball-end mills. The entire surface of the STL model is treated as a single composite surface to be machined using raster milling. To improve machining efficiency, an automatic surface splitting algorithm could be developed to divide the model into several regions based on the characteristics of a group of triangular facets, and then machine these identified regions using different strategies and cutters. The offset-based tool-path generation algorithm from STL models is a unique and novel development, which is useful in the rapid prototyping and computer-aided machining areas.

Moller and Trumbore [25] have presented a clean algorithm for determining whether a ray intersects a triangle. The algorithm translates the origin of the ray and then changes the base of that vector which yields a vector $(t\ u\ v)^T$ where t is the distance to the plane in which the triangle lies and (u, v) represents the coordinates inside the triangle. One advantage of this method is that the plane equations need not be computed on the fly nor be stored, which can amount to significant memory savings for triangle meshes.

Segura and Feito [26] have presented an algorithm to determine the intersection between rays and triangles based on the idea of the study of signs with respect to triangles. One of the advantages of this approach is its robustness due to its lack of trigonometric operations or complex divisions which might alter the result of the calculations. The algorithm is similar (or even better) in time to other existing algorithms, but it is based exclusively on the study of signs, so that the results obtained are more precise. A comparative study of times between the algorithm and other similar algorithms is presented.

Kim and Yang [27] presented the incomplete triangular mesh including reversed faces and gaps between boundary edges are directly offset for NC machining. For three-axis machining, vertical faces are deleted and the face normal vectors directing the lower part of model are changed to upward. The complete mesh is offset by the multiple normal vectors of a vertex, and boundary edges and boundary vertices are offset by the virtual multiple normal vectors of a boundary vertex. For five-axis machining, mesh is offset to both directions by the reversed multiple normal vectors. Using this method, the CL surface is directly obtained from an incomplete triangular mesh.

2.7 Problem Definition

By reading the above literatures it has found that Surface Offset methodology has been used successfully in many published work for tool path generation, however they were used for milling operation with Ball end nosed cutter. Solution for Offset method will be discussed in next chapter.

Tool path generation methods have also been studied and an efficient Ray Intersection method is selected which is used for finding the cutter location point in chapter 3.

Chapter 3

METHODOLOGY FOR SOLUTION

3.1 Tool path planning for sculptured surface for milling lathe

Tool path planning will be done according to specification of 3 Axis NC milling lathe machine which are given below:

3.1.1 Specification of 3-Axis NC Milling Lathe machine [30]

The 3 Axis NC Milling Lathe PBG KW 2048[30] machine used is having one rotational and two translational motions, tool penetrate the workpiece along X axis, the tool traverse along Z axis and workpiece rotate about Z axis. Any pseudosymmetric part (which may or may not have a central axis of symmetry) can be machined by this machine. Since we are using offset technique which already consider tool radius offset for roughing and finishing path, so we assume tool radius as a zero which act like a single point. So whenever a cutter contact (CC) point on an offset surface is found this means cutter location point (CL) is found. The tool path of this machine is helical as shown in figure 3.1. The C axis, which is the rotational axis, gives the location of number of points on the surface along the helical curve. Tool axis is consider as a ray in such condition, which is projecting on a surface from a source point which is at the central axis to destination point which is at a far distance along X axis, in a helical path and each contact point is stored as a cutter location point as shown in figure 3.2.

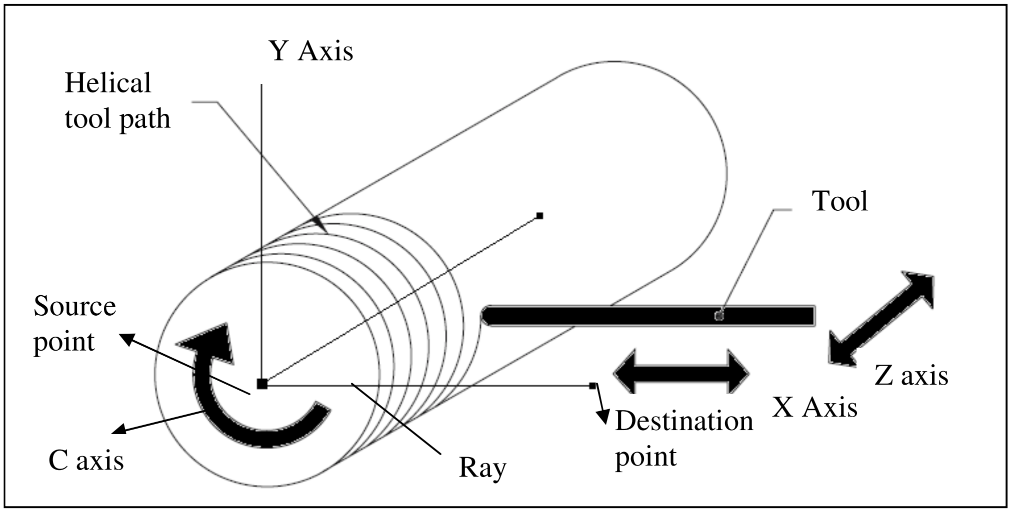


Figure 3.1: 3 Axis NC Milling Lathe Machining Coordinates

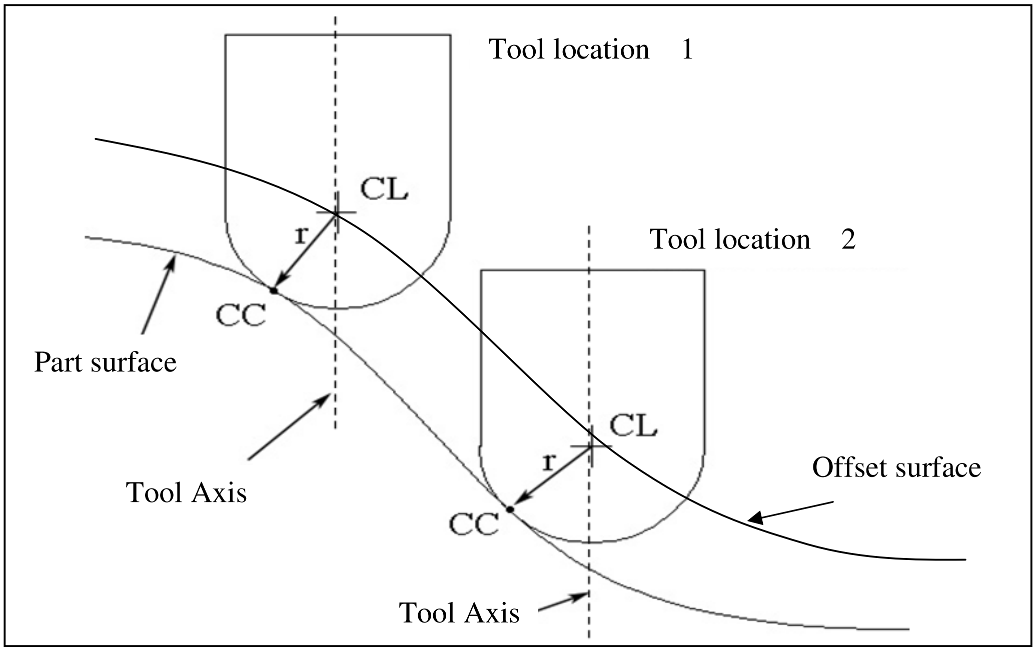


Figure 3.2: Cutter contact and Cutter location path for offset surface

It has a limitation of the work piece size of length along Z axis of 48 inch and diameter of 10 inch, traveling along X axis is limited by 51inch and along Y axis 9.45 inch . The accuracy is of .001 inch and repeatability is of .001 inch. It has least count of 1 degree.

The machine takes input as a .tp file which is as shown in figure below:

```
X30.458130 Y0.000000 Z0.000000
X30.467742 Y0.001102 Z1.440000
X30.496610 Y0.002204 Z2.880000
X30.544823 Y0.003306 Z4.320000
X30.612536 Y0.004408 Z5.760000
```

Figure 3.3 : .tp file format

The first column shows the tool traveling along the X axis; the second column shows the tool travel along the Z axis and the last column shows the angle turned by C axis or workpiece rotational axis.

3.1.2 Parameter required for tool path planning

Some parameters required for tool path planning for pseudosymmetric sculptured surfaces machining, which is user dependent, are as follows:

1. Diameter for roughing and finishing pass of ball nosed end cutter

2. Depth of cut for roughing pass
3. Finishing allowance for finish pass
4. Pitch of helical path
5. Number of cutter location point per pitch

These parameters are governed by user as per his requirement of quality and time. Surface offset method and tool path generation methodology will be discussed below:

3.1.3 Surface offset methodology

3.1.3.1 Average surface normal method for vertex offsetting [8]

The STL files are generated by tessellating the outside skin of the CAD models. The tessellation (STL) is done by approximating the boundary of the CAD object with triangles. An STL file contains coordinates of the vertices and normal for each facet. Since a STL file does not contain the geometric information of the vertex normal, the normal at each vertex need to be calculated. In this thesis, we use an averaged normal vector method to offset each vertex with the corrected normal direction, as shown in Figure 3.4(c). An offset normal vector at a vertex is calculated by averaging the normals of all the facet that are connected to the vertex. Since, the averaged normal vector method averages the normal vectors of the facets that are connected to the vertex; it approximates the original CAD model closely. However, the accuracy of the method depends on the number of triangles used in the original STL model when approximating the CAD model of the designed part. As shown in Figure 3.4(a), a vertex normal $\overrightarrow{N}_{Vi}$ at vertex V_i, where there are n facets connected to, can be calculated as follows:

$$\overrightarrow{N}_{Vi} = \frac{\sum_{j=1}^{n} \vec{N}_{i,j}}{\left|\sum_{j=1}^{n} \vec{N}_{i,j}\right|} \qquad 3.1$$

Where $\overrightarrow{N_{i,j}}$ are the normals of the facets that are connected to the vertex V_i. Although Eq. (3.1) can work for smooth surfaces, it may still cause problems (for some special cases) if it is used for vertices at sharp corners or flat surfaces. Depending on the triangulations generated in the STL files, the same vertex may have different sets of adjacent triangle facets connected to it. A vertex on a flat surface or on an edge of the flat surface might be connected to several faces with the normals parallel to each other, as shown in Figure 3.4(b), the two facet normals $N_{i,1}$ and $N_{i,2}$ are parallel (i.e. $\vec{N}_{i,2} // \vec{N}_{i,1}$).

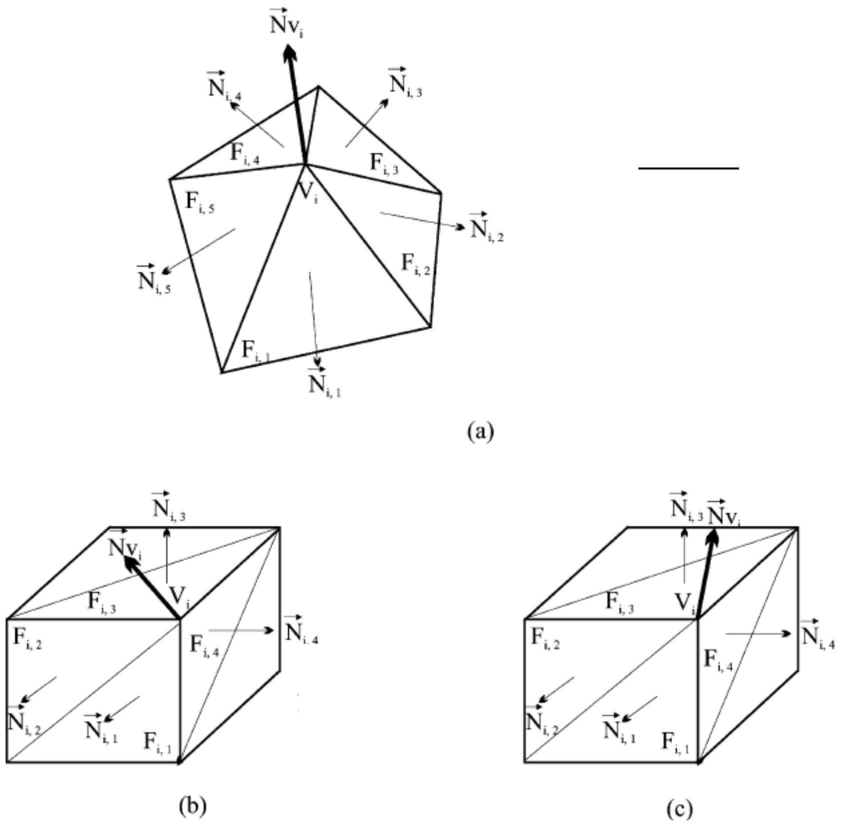


Figure 3.4: Calculating the averaged normal vectors at a vertex: (a) Calculating the vertex normal by averaging the facet normals; (b) Incorrect normal vector shifted and ; (c) Corrected normal vector by deleting the parallel normal vector

The normal vector at vertex is calculated as follows:

3.2

Directly averaging these normals Figure 3.4(b), to calculate the vertex normal N_{Vi} may result in the calculated normal vector shifts towards the faces with parallel facet normals (i.e.) As shown in Figure 3.4(b), the averaged surface normal at the vertex could result in a vector that is closer to the faces $F_{i,1}$ and $F_{i,2}$ due to the fact that these two adjacent faces have the same parallel normals ( ). Figure 3.4(c) shows the corrected normal vector found by eliminating the duplicated parallel normals in the calculation of the averaged normal. In Figure 3.4(c), the corrected normal surface normal vector at the vertex is calculated by averaging all the adjacent facet normals without the duplicated parallel normal as follows:

$$\vec{N}_{Vi} = \frac{\overrightarrow{N_{i,1}} + \overrightarrow{N_{i,3}} + \overrightarrow{N_{i,4}}}{\left|\overrightarrow{N_{i,1}} + \overrightarrow{N_{i,3}} + \overrightarrow{N_{i,4}}\right|} \quad 3.3$$

After the corrected normal vectors $\vec{N}_{Vi}$ at each vertex V_i are found, the offset vertices V_i' can be calculated by offsetting the vertices in their normal directions with a given tool radius offset distance t as follows:

$$V_i' = V_i \pm t\overrightarrow{N_{Vi}} \quad 3.4$$

In Eq. (3.4), the sign (±) depends on whether it is offset outward or inward from the original part surface. Once all the vertices are offset with tool radius 't' then each vertex is placed at the same facet number in new STL file as it was in original STL. Two STL file are made, first for rough pass and other for finish pass, depend upon the tool radius offset distance. In this thesis this whole method is developed in Visual C++ language to generate two STL file for rough and finish tool. Before ray intersection method is used, all the normal which has normal value 0, 0, -1 and 0, 0, 1 are removed because they are parallel to tool axis, once all such normal are removed ray intersection method is applied.

3.1.4 Ray intersection method

In ray intersection method, a ray which is a line (in this case tool axis) with one source point and other destination point is projected on a surface and give intersection point. There are three possible ways a ray can intersect with a triangular facet.

1. Triangle check for point inside the triangle
2. Edge check for point on the triangle
3. Vertex check for point on the triangle vertex

There are many ray intersection algorithm [26] such as MOLLER'S ALGORITHM [25], SNYDER ALGORITHM [28] and SEGURA ALGORITHM [29] but in the present work **BADOUEL'S ALGORITHM [7]** is used to find intersection point of triangle.

3.1.4.1 Triangle check

Badouel's algorithm [7] is based on the study of barycentrics (α, β) coordinates. Let $V0V1V2$ be a triangle. The position of point P inside the triangle can be expressed as shown in Figure 3.5, or either in equation 3.5.

$$\overrightarrow{V_O P} = \alpha\overrightarrow{V_O V_1} + \beta\overrightarrow{V_O V_2} \quad 3.5$$

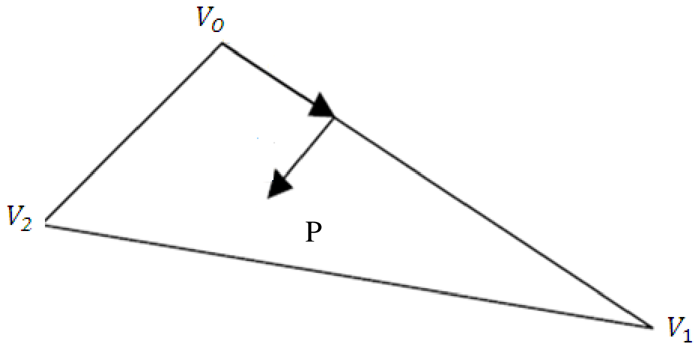


Figure 3.5: Barycentrics coordinates of point P

It is said that point P is inside the triangle if it is true that

3.6

Equation 3.5 is decomposed in a system of three equations, as shown in equation 3.7.

3.7

After finding the coordinates of intersection point, the value of U parameter of a ray equation represent by: is found. Where and are source and destination point respectively. Same is done with all the facets for the best value. If the point is not in the triangle then next check is performed.

3.1.4.2 Edge check

In this check line segments intersection point is find using equation as follows:

3.8

Where , after finding the coordinates of intersection point, the value of U parameter of a ray equation represent by: is found. This check is performed for all the three edges of triangle and with all the facets for the best U value. If the point is not on the triangle edge, next check is performed.

3.1.4.3 Vertex check [6]

In this check ray is pass through point which is at the vertex of triangle then equation used as follows:

3.9

This equation is used for all the three vertices of triangle. If there are no real-valued solutions, then the ray will not touch for any value of . In general, when the ray can touch , there will be two solutions; the solution furthest away from the source point is chosen. A visual C++ programme is made to generate all the tool path cutter location point which is also made for all the rough pass and finish pass, which is used in machine to machine different sculptured part. Once all the cutter location point are calculated tool path planning for rough pass can be done.

3.1.5 Tool path planning for rough and finish pass

Since rough pass are used with large diameter tool so finishing is not the main criterion in such case for sculptured surface, rough pass is perform to minimize time of cutting, so instead of cutting all the sculptured surface at one go using roughing tool, the surface is machined in steps which is defined by user as a depth of cut.

Methodology used has found the value of parameter for all facets and ray projection, from this and the value of value is found. The number of rough pass depends upon depth of cut which are automatically generate from visual C++ programme as shown in figure below:

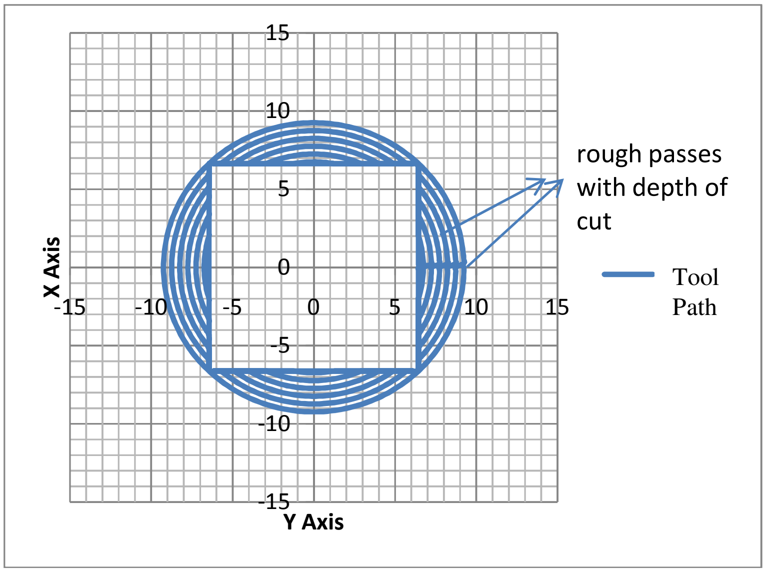
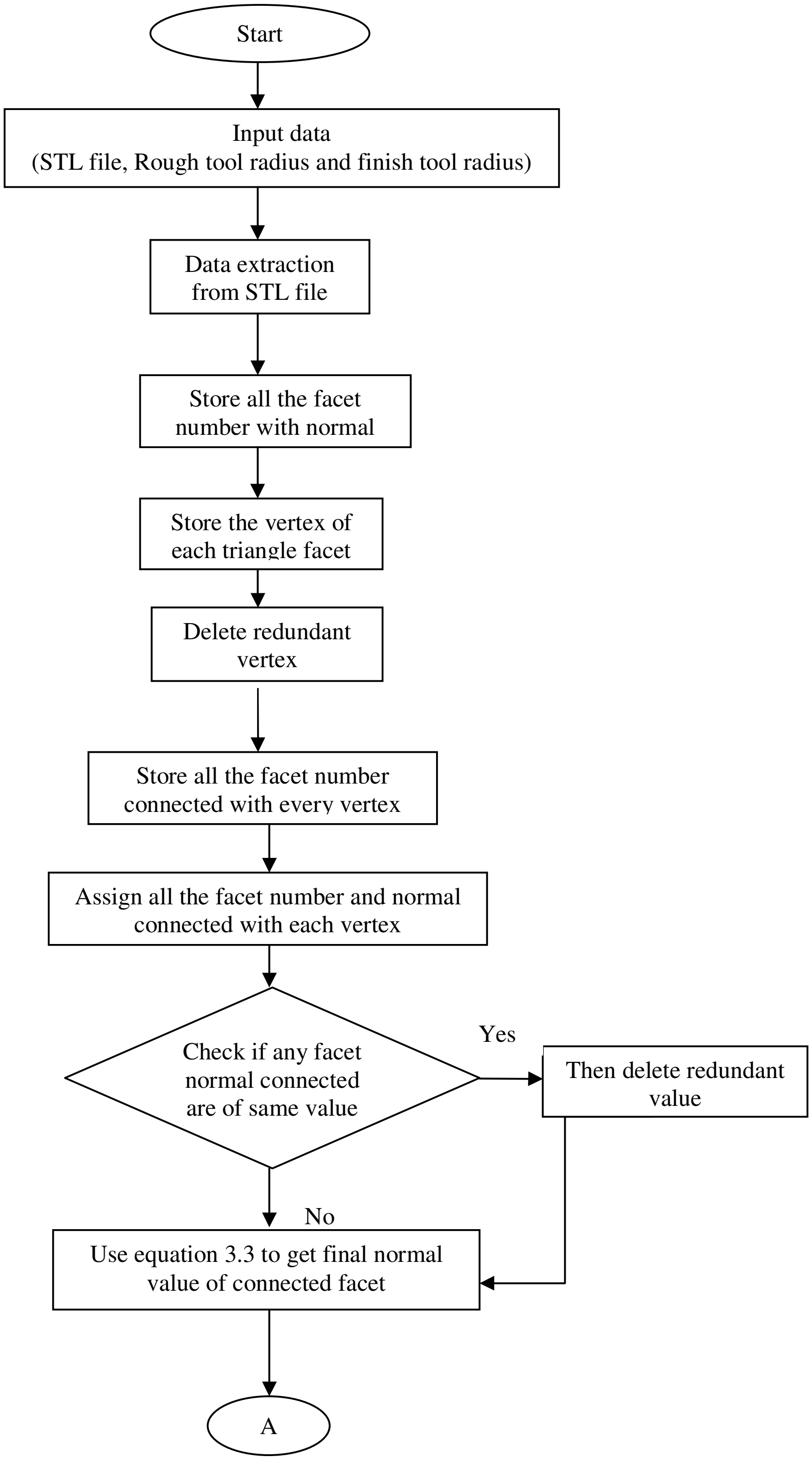


Figure 3.6: Tool path for roughing passes with depth of cut

Finish pass is used with actual value with smallest possible tool for better finishing; the feed rate should be slow to avoid unwanted error.

3.2 Solution procedure for Tool path planning for milling lathe

Based on methodology described in previous sections a computer program has been developed in visual C++. The flow chart of an iterative scheme is shown in figure 3.7.



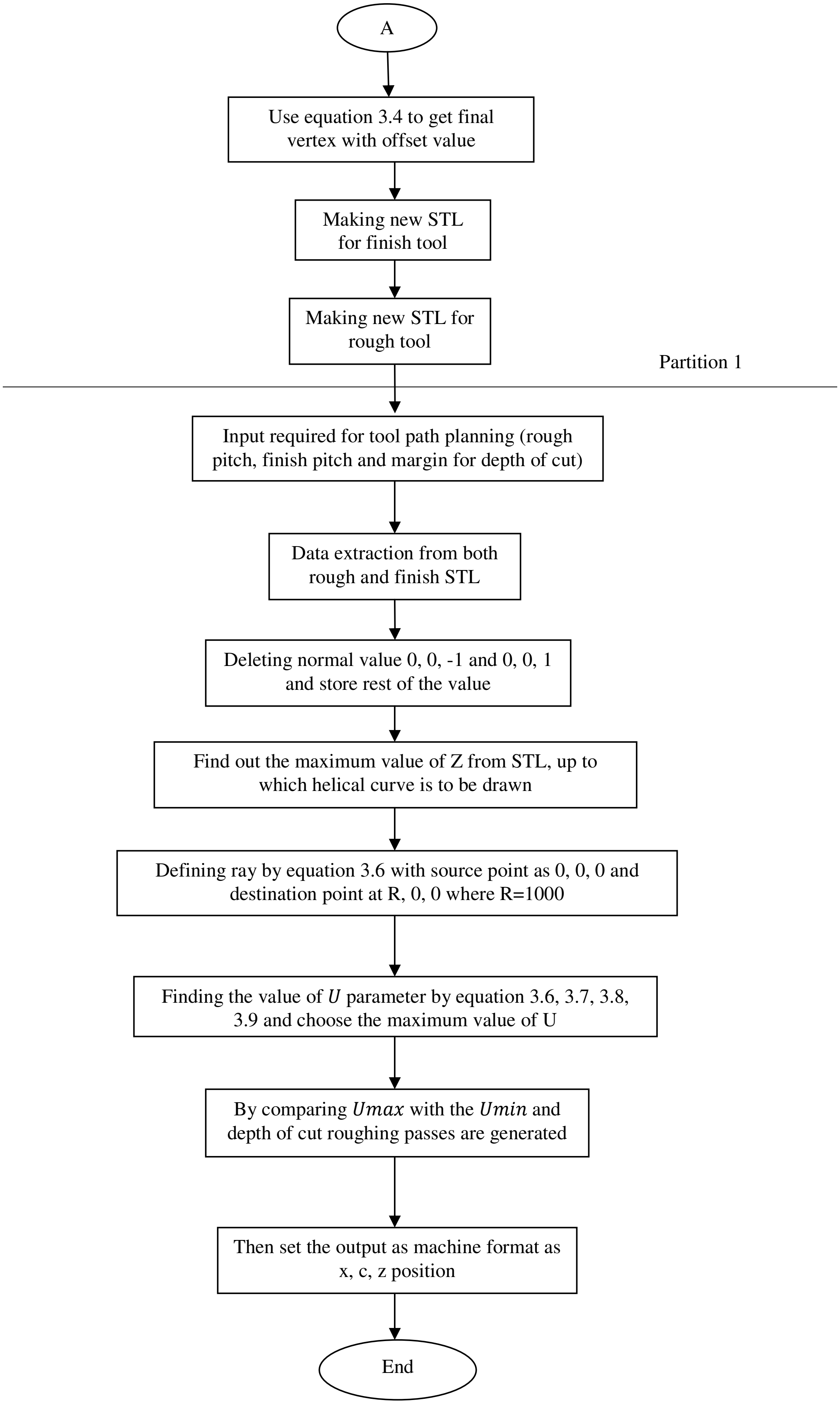


Figure 3.7: Flow chart of tool path planning for milling lathe

3.3 Tool path planning for sculptured surfaces for 3 Axis NC vertical milling machine

Two kind of tool path planning, for raster milling and circular milling, are discussed below:

3.3.1 Raster milling

3.3.1.1 Specification of 3-Axis NC milling machine [21]

The tool planning is done for 3 Axis NC Raster Milling machine is having all three translational motions, tool penetrate the workpiece along Z axis, the tool traverse along Y axis till it reaches to maximum position of it with forward feed and then jump back to X axis with some user defined side feed till the end of part. The part geometry should be made from (0, 0, 0) position in Cartesian coordinates and all X, Y, Z Axis should be in positive direction as shown in figure 3.8.

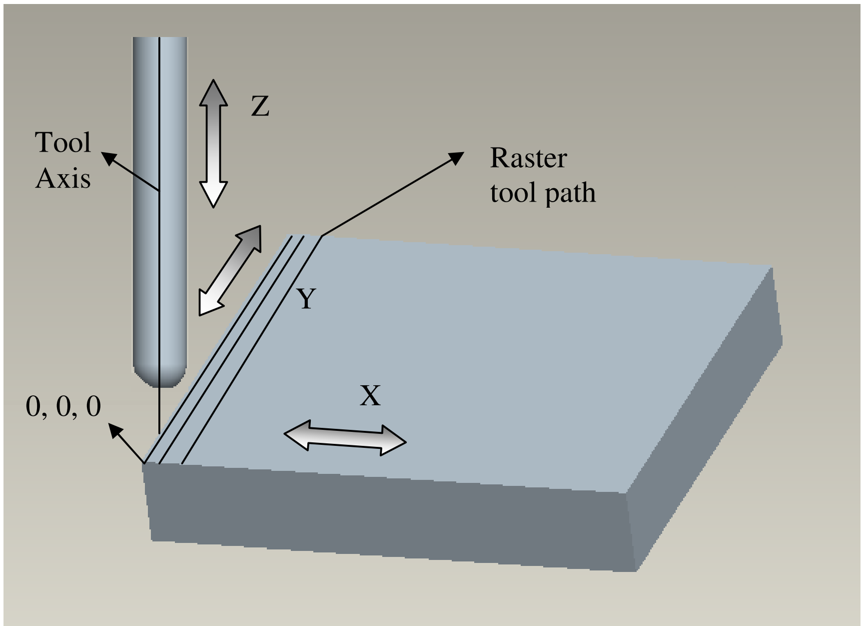


Figure 3.8: 3 Axis NC Raster Milling machine

Any sculptured surface which is not a revolving or round body can be machined by this machine. This methodology is only used by ball nosed milling tool. An offset technique is used which has already been discussed in article 3.1.3. The tool path of this machine is raster tool path for milling machine as shown in figure 3.8.

Before applying ray intersection all the normal with only 0, 0, -1 are removed which are plane surface on the XY Axis or parallel to Tool axis.

3.3.1.2 Parameter required for raster milling

Some parameters required for tool path planning for sculptured surface, which is user dependent, are as follows:

1. Diameter for roughing and finishing pass of ball nosed end cutter
2. Depth of cut for roughing pass
3. Finishing allowance for finish pass
4. Forward feed in Y Axis
5. Side step in X Axis

These parameters are governed by user as per his requirement of quality and time. Tool path planning for rough pass and finish pass, will be as same as discussed in article 3.1.5.

3.3.2 Circular milling

3.3.2.1 Specification of 3-Axis NC milling machine

The tool planning is done for 3 Axis NC Circular Milling machine is circular tool path as shown in figure 3.9. Z Axis is the translational Axis for tool movement, X and Y are the Axis along which tool travels in a circular path. Z Axis is in positive direction whereas X and Y Axis could be anywhere. Initial tool position is on (0, 0, 0) position in Cartesian coordinates. This methodology can only be used by ball nosed milling tool. Tool path is generated by offset technique, which has already been considered in article 3.1.3. The tool path of this machine is raster tool path which is a universal tool path for milling machine as shown in figure 3.9.

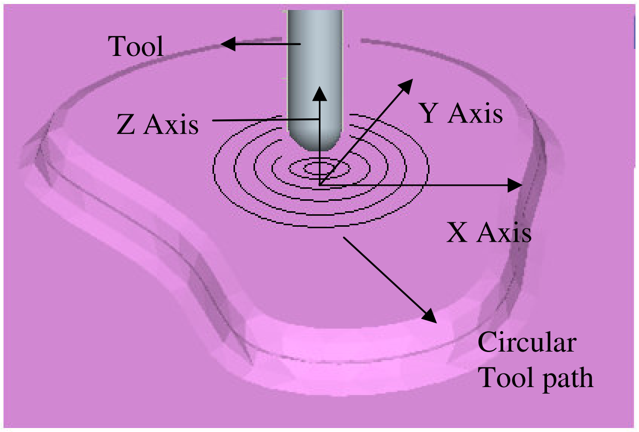


Figure 3.9: 3 Axis NC circular Milling machine

3.3.2.2 Parameter required for tool path planning

Some parameters required for tool path planning for sculptured surface, which is user dependent, are as follows:

1. Diameter for roughing and finishing pass of ball nosed end cutter.
2. Depth of cut for roughing pass.
3. Finishing allowance for finish pass.
4. Difference between two circular paths.
5. Number of turn in one circular path.

These parameters are governed by user as per his requirement of quality and time. Tool path planning for rough pass and finish pass, will be as same as discussed in article 3.1.5.

3.4 Solution procedure for NC tool path planning for milling

Based on methodology described in previous sections a computer program has been developed in visual C++. The flow chart of an iterative scheme is shown with detailed step taken in figure 3.10.

The offset surface method flow chart is same as shown in partition 1 of figure 3.7. The tool path planning for raster and circular milling is shown in flow chart below in figure 3.10.

The difference in vertical milling tool path planning with milling lathe is in ray intersection source and destination point, STL modification and the tool path both machine will work on.

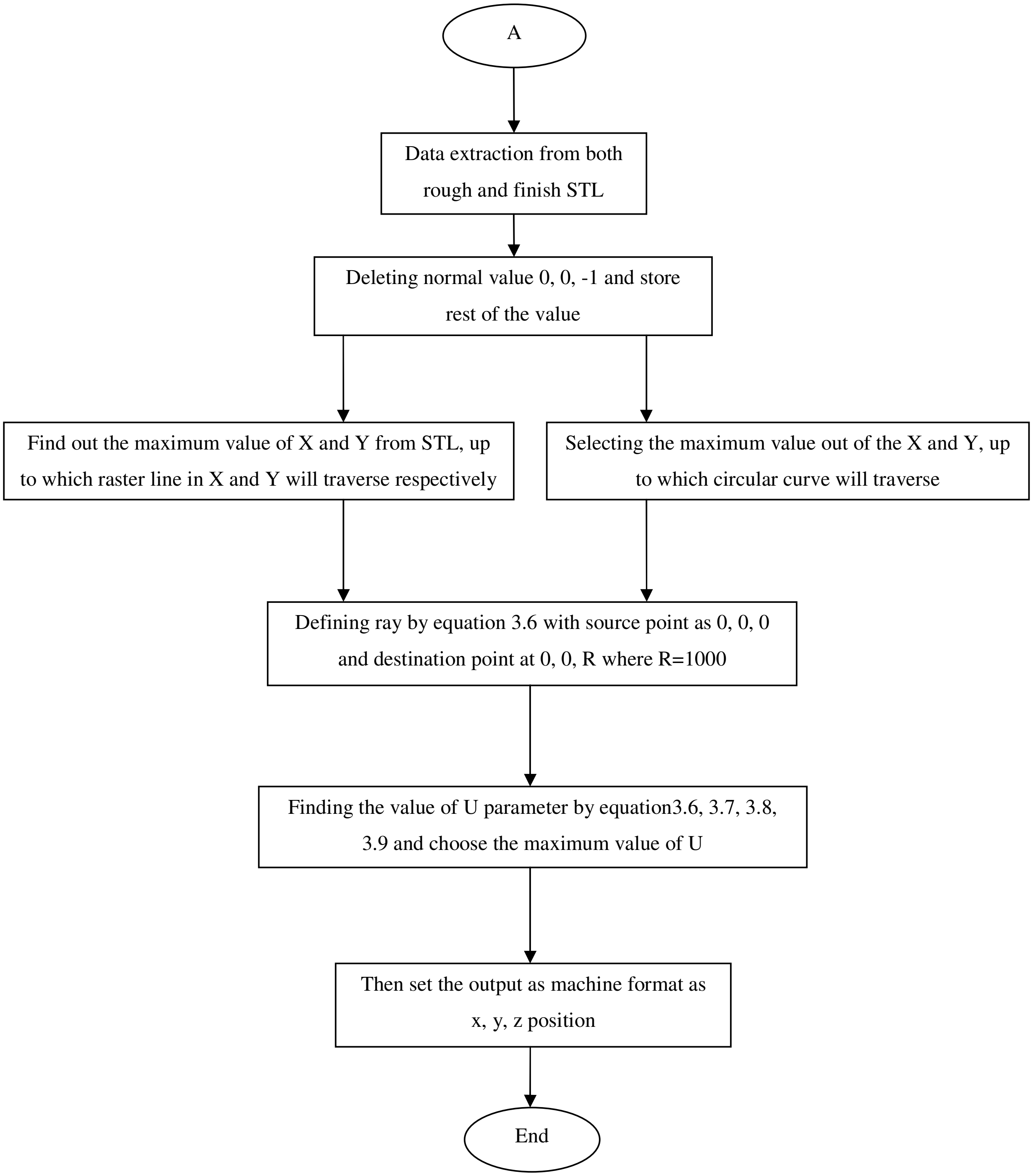


Figure 3.10: flow chart for vertical milling

Chapter 4

RESULTS AND DISCUSSION

The tool path generation and planning which has been done in chapter 3 has been validated and actual machined for pseudosymmetric sculptured part on PBG KW 2048[30] 3 Axis milling lathe. The tool path generation and planning for milling for raster milling and circular milling is validated only on 3D plotter in MATHCAD® software from PTC (Parametric Technology Corporation).

4.1 Validation

Tool path generation and planning has been validated with actual CAD model and actual machining has been done to validate the accuracy of machining. The results is also been validated using 3D plotter in MATHCAD® software from PTC. The point cluster in Cartesian space corresponding to planned tool path for roughing as well as finish passes has been checked

4.1.1 Validation for 3 Axis NC milling lathe

4.1.1.1 Validation with CAD model

Cube of 10x10x10 mm is checked on MATHCAD® to verify surface offset and tool path generation.

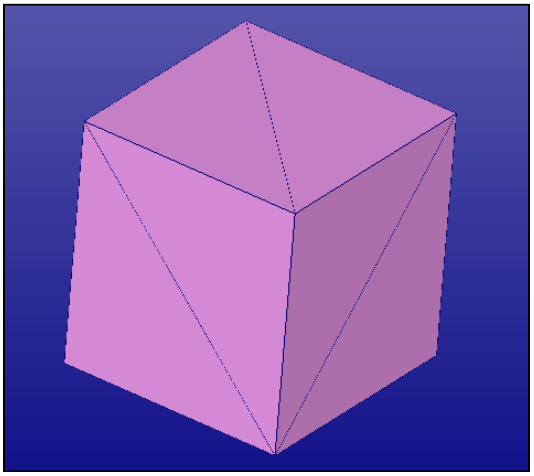


Figure 4.1: Actual STL

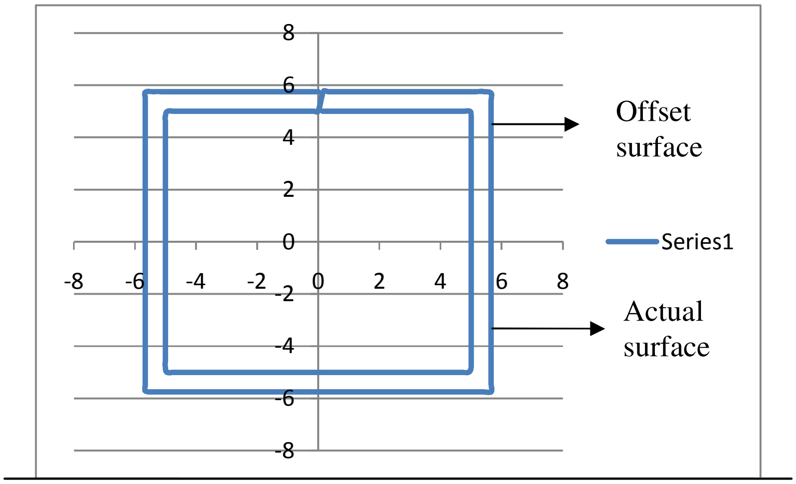


Figure 4.2 : Difference between actual and offset surface of 2mm

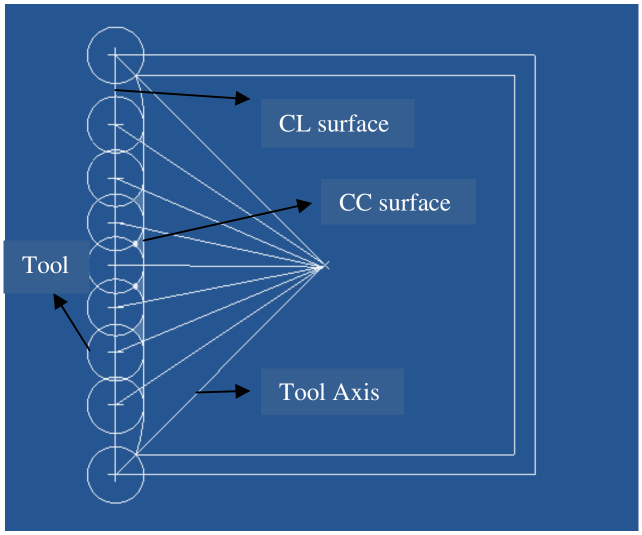


Figure 4.3: CC surface and CL surface with tool axis

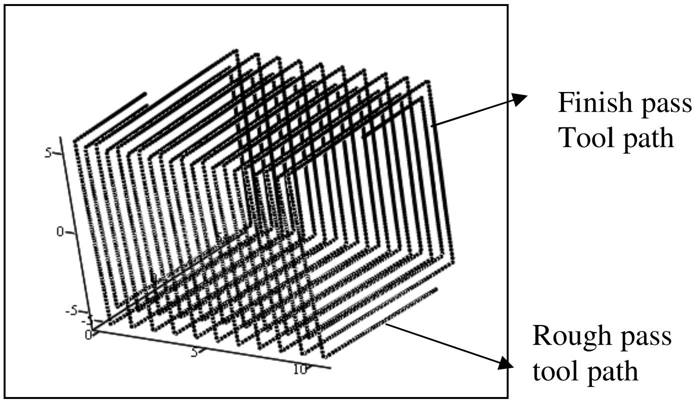
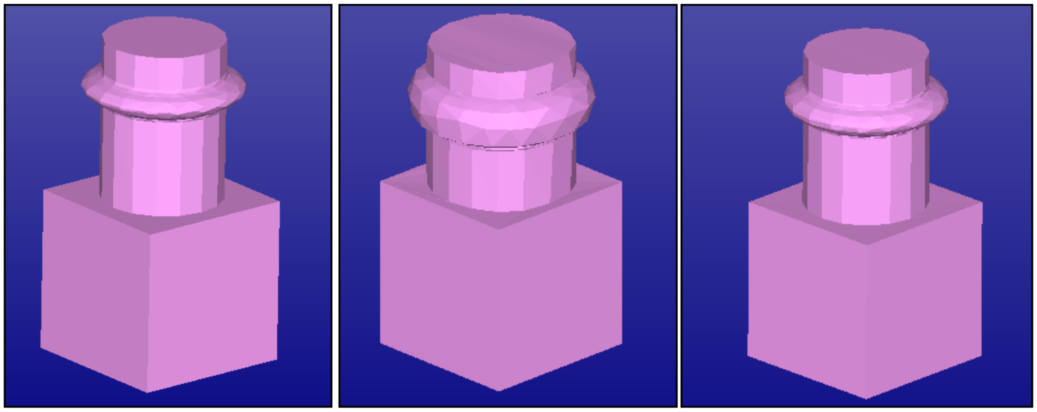


Figure 4.4: Roughing and Finishing tool path

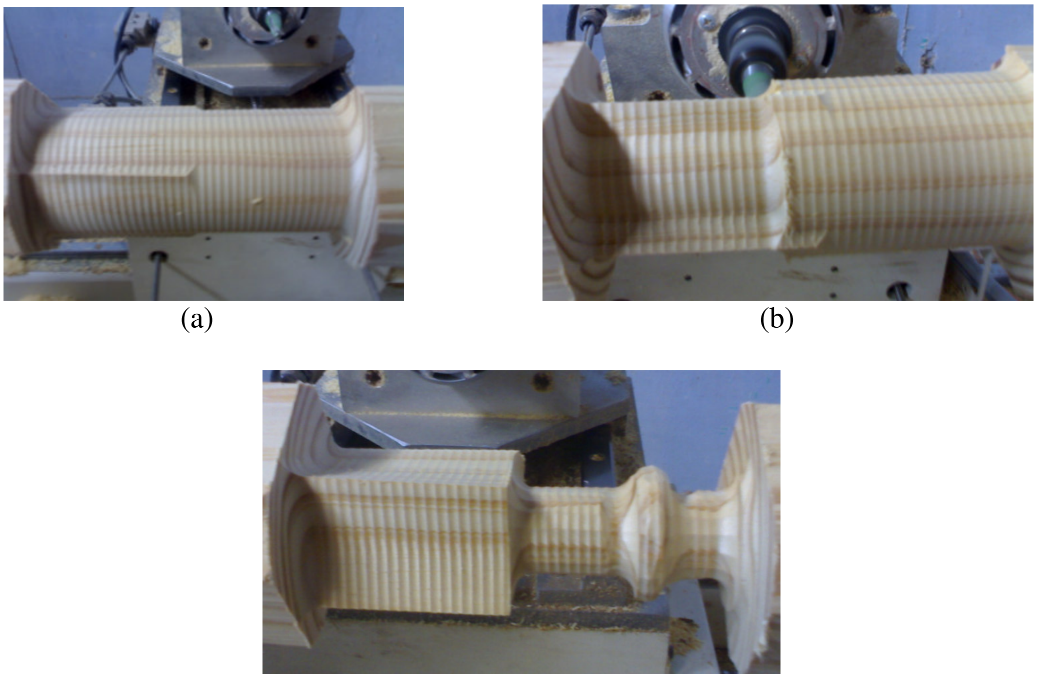
4.1.1.2 Comparison with actual surface model

Following validation are done for a design with square, cylinder and a ring on cylinder as shown in figure 4.5. This part is having dimension of cube as 60x60X60 and cylinder with 45 mm diameter and a ring of diameter 60 mm which is 95 mm far from XY axis.



(a) **(b)** **(c)**

Figure 4.5: (a) actual STL (b) offset surface with 6.35mm (c) offset surface with .7935mm

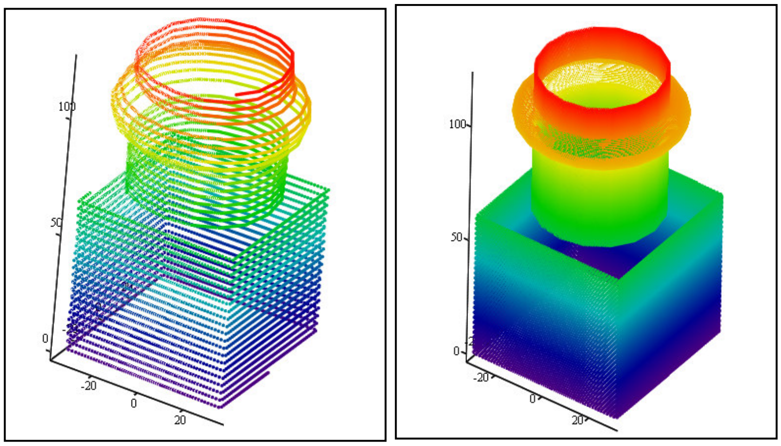


(a) (b)

(c)

Figure 4.6: (a) first rough pass (b) third rough pass (c) final rough pass

MATHCAD generated tool path for roughing and finish pass are shown below:

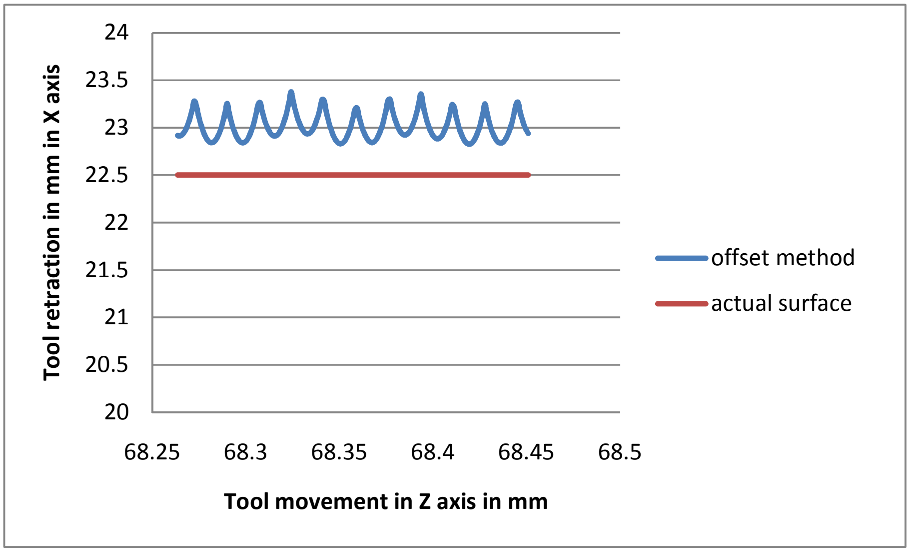


(a) **(b)**

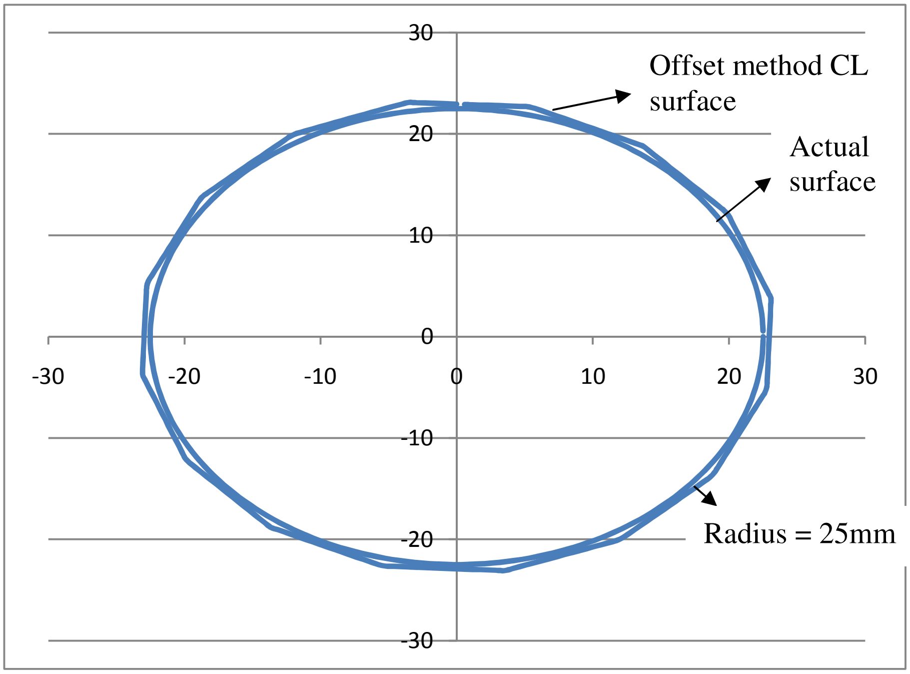
Figure 4.7: (a) rough tool path (b) finish tool path

The following figure shows the difference between tool path generated by surface offset methodology and the actual surface from CAD model.

The figure 4.8 shows the finish pass for cylindrical portion of the part with actual CAD surface for one complete revolution of circular portion. It is clearly shows that STL patches are visible.



(a)



(b)

Figure 4.8: (a) and (b) Difference in CL surface for circular portion for finish pass with actual surface with radius 25mm.

Figure 4.9 shows the CL surface of a square portion of 60x60 mm dimension comparison with actual CAD surface, since square portion STL file have two triangular patches on the surface which are flat or constant surfaces. The portion in the middle of the square is showing some distortion because of the undercut because of lesser tool radius at middle portion to the vertices as shown in figure 4.2 in detail.

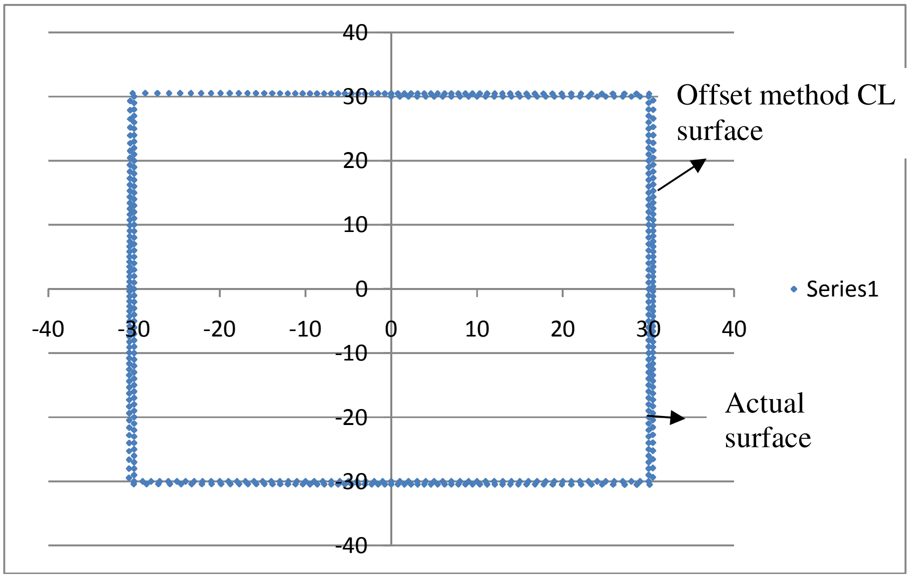


Figure 4.9: comparison of actual CAD with CL surface of square of 60x60mm dimension for finish pass

4.2 Validation for vertical milling

4.2.1 Validation with CAD model

4.2.1.1 Raster milling

A test design was modeled in PRO-E CAD software with square base of 50x50 mm and a cap like design on Z Axis and saved as STL format.

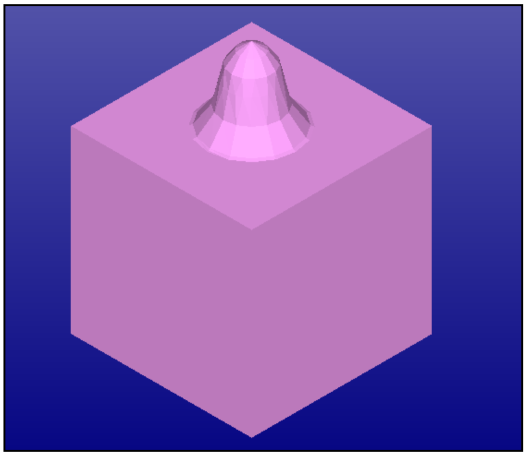
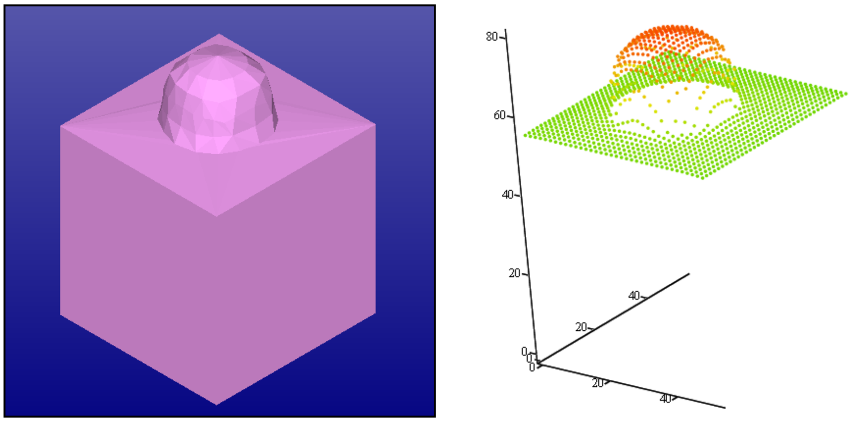
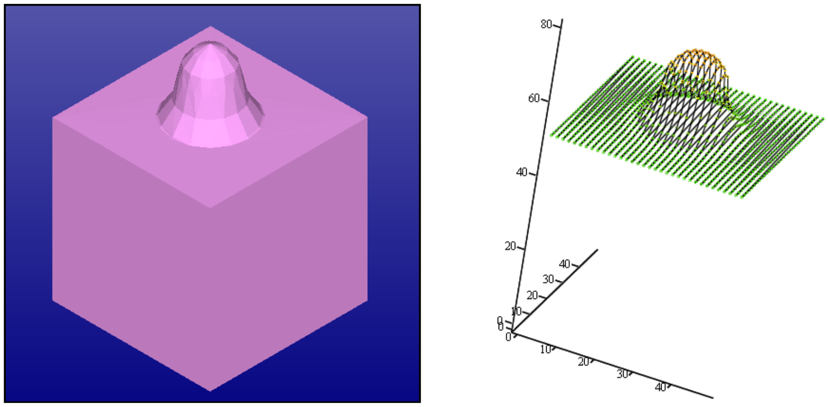


Figure 4.10: Actual STL



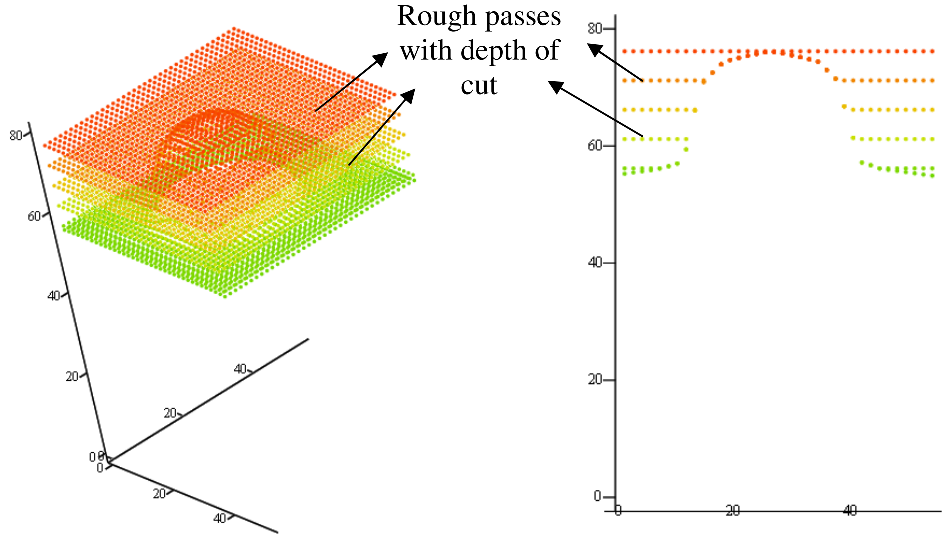
(a) (b)

Figure 4.11: (a) offset surface with 6.35mm radius (b) roughing tool path



(a) **(b)**

Figure 4.12: (a) offset surface with .7395mm radius (b) finish tool path



(a) **(b)**

Figure 4.13: (a) Roughing passes with depth of cut (b) sectional view

4.2.1.2 Circular milling

A test part was modeled in PRO-E CAD software with 50x50 mm square base and a snow cap like design on Z Axis.

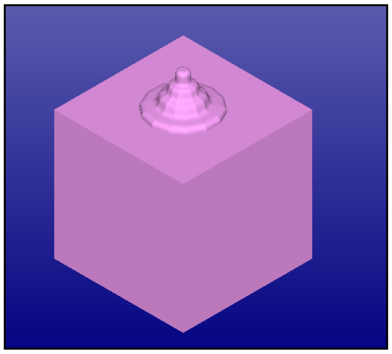
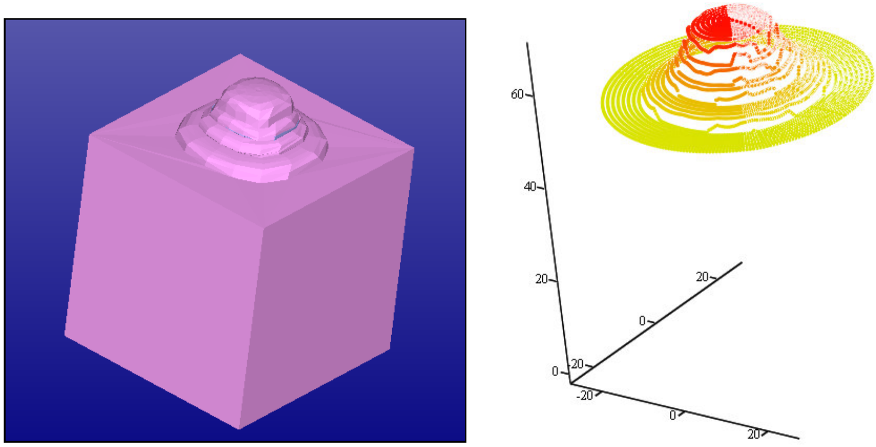
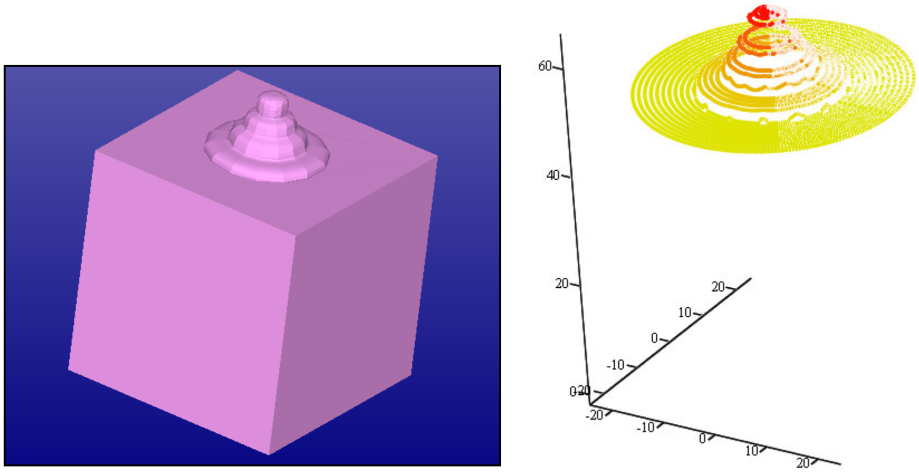


Figure 4.14: Actual STL



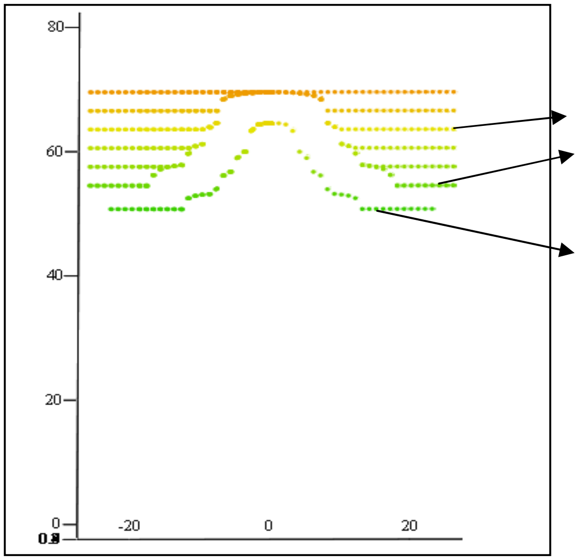
(a) **(b)**

Figure 4.85: (a) offset surface with 6.35mm radius (b) roughing tool path



(a) **(b)**

Figure 4.96: (a) offset surface with .7395mm radius (b) finishing tool path



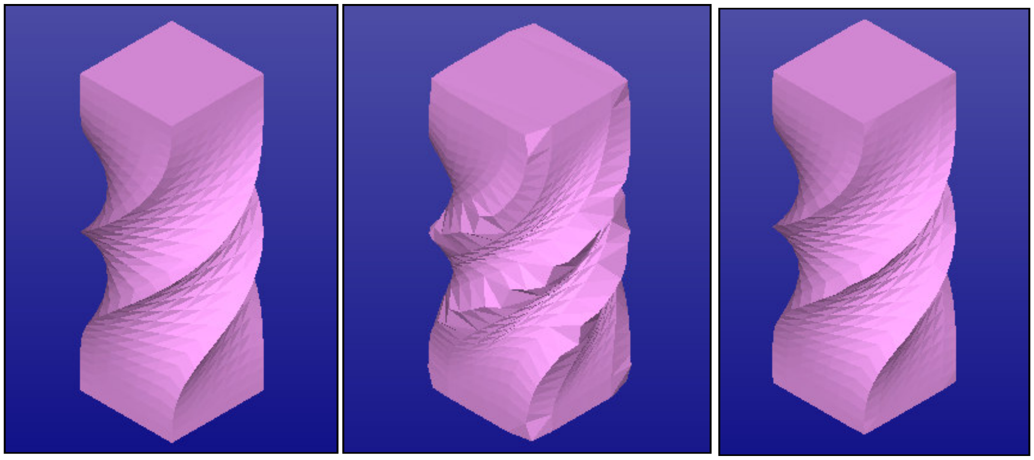
Rough passes with depth of cut 5mm

Finish pass

Figure 4.17: Sectional view of circular Tool path for all passes

4.1 Results

A twisted square of 50x50x150 mm dimension is modeled in PRO-E CAD software with 180 degree twist and saved as .STL file format with 1604 triangle facet.



(a) **(b)** **(c)**

Figure 4.18: (a) Actual STL, (b) offset surface with 6.35mm (c) offset surface with .7935mm

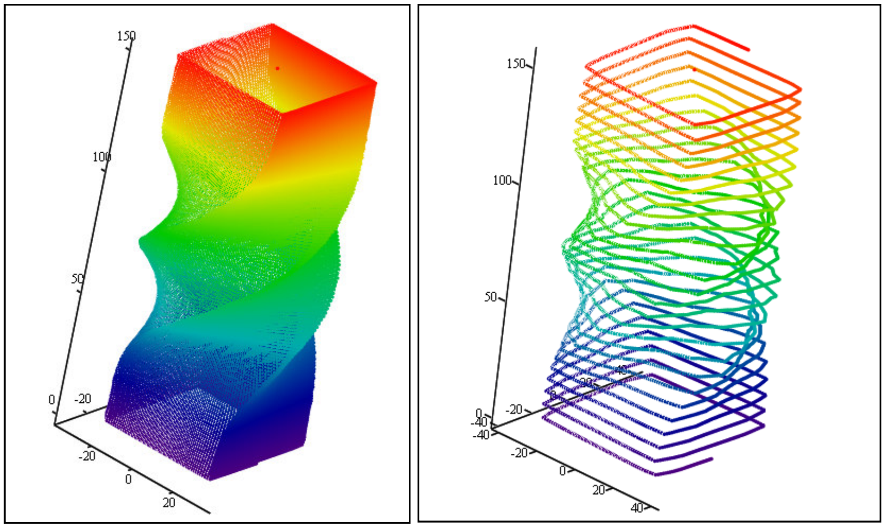
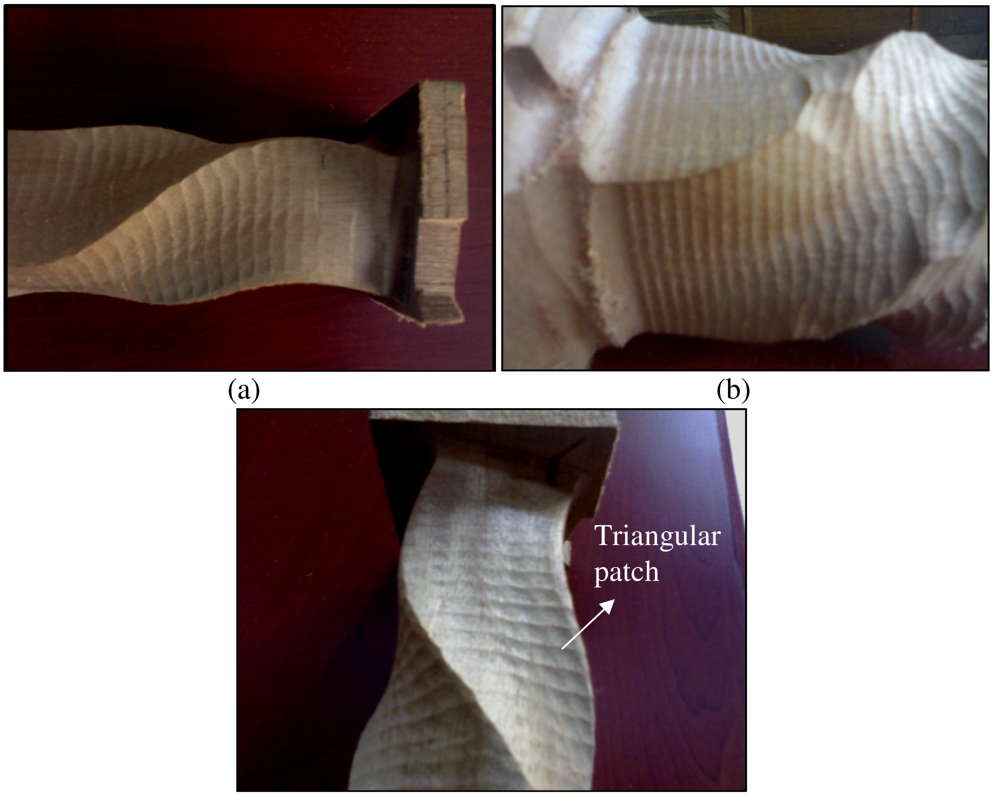


Figure 4.19: Finishing and roughing tool path



(a) (b)

(c)

Figure 4.20: (a) Finish Part (b) rough Machined part (c) triangular patch

The final machined part has been measured by and 50x50mm cross section twisted square part has found to be of 51.5x51.5mm after machining and the twisted portion has been found out to be of 72.7mm whereas the actual part is having 70.71mm.The percentage error found to be 2.9 for square cross section and 2.7 for diagonal value error on twisted portion.

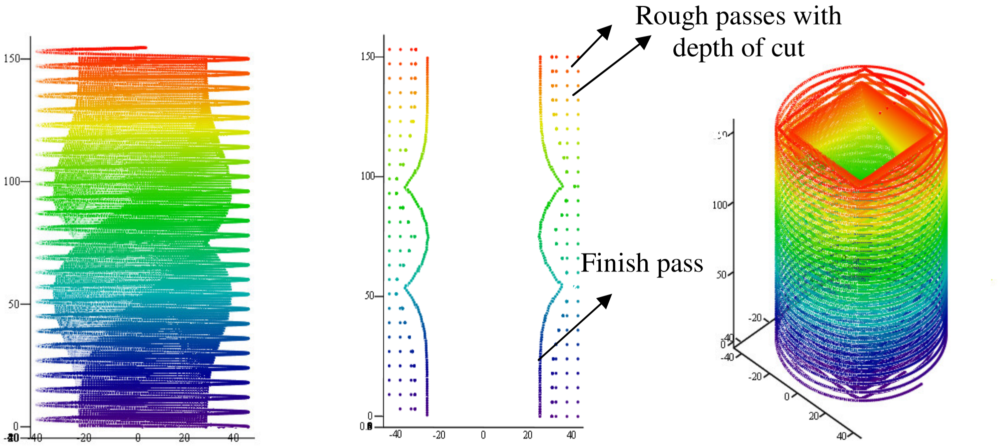
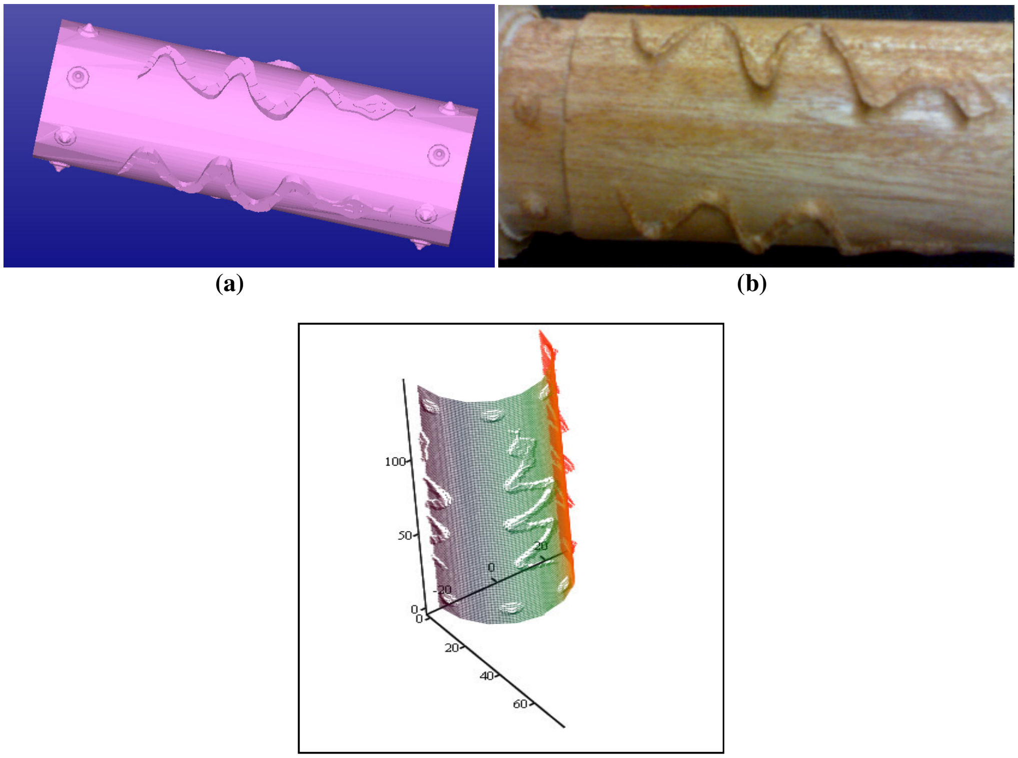


Figure 4.101: All rough passes with depth of cut and finish pass tool path

A new design of four snake around a cylinder of 50x150 mm with four snake extrude of 3 mm is modeled and saved as STL file with 17000 triangle facet since tool diameter was 1.57mm and a part has intricate structure with .2mm and .1mm that is the reason The machining is showing undercut and overcut.



(a) (b)

(c)

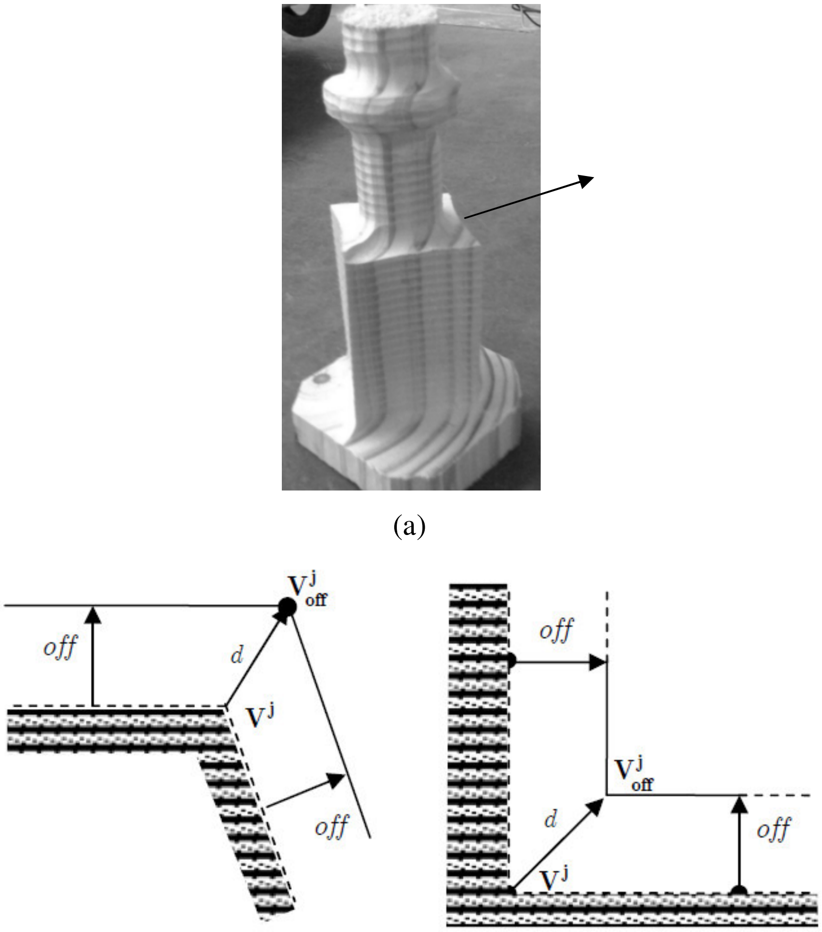
Figure 4.22: (a) actual STL (b) machined part (c) finish tool path

Actual dimension of snake cylinder is 50mm and machined part dimension found out to be 51.1mm. The percentage error is found to be 2.1.

4.2 Errors and Limitations

The offset methodology used in this thesis has focused on just solving the problem of offsetting all individual vertices of an STL model. It works well for small offset values, where local and global self-intersections normally do not happen or is not a main issue. To handle the self-intersection problem while offsetting the model with relatively large values, some post processing needs to be done.

Some problems can occur if the vertex V^j is translated along it and the geometry is consequently modified. As shown in Figure 4.23, the distance j off d = is different from the offset distance off by imposing a planar translation to the surfaces. So if there is too sharp edge like in Figure 4.23, offset method will give undesirable result.



(a)

(b) (c)

Figure 4.23: (a) shows actual part undercutting Section view of offsetting a node on a solid edge: (b) convexity; (c) concavity.

Since the offset vector for each vertex is calculated using a averaged normal of those triangular facets connected to each vertex, in some situations, as shown in Figure 4.23(a), the length of the offset vector is much larger than one, and the vertex offset value becomes very large compared to the offset distance d Offset. This is obviously undesirable. In order to solve this problem, a vertex splitting technique can be used. For this method, the original vertex is first divided into two vertices by adding a zero area triangular facet, then two vertices which include two different groups of triangular facets are offset individually using the above-mentioned method. Figure 4.23(b) shows the offset results by applying this technique to the geometry shown in Figure 4.23(a).

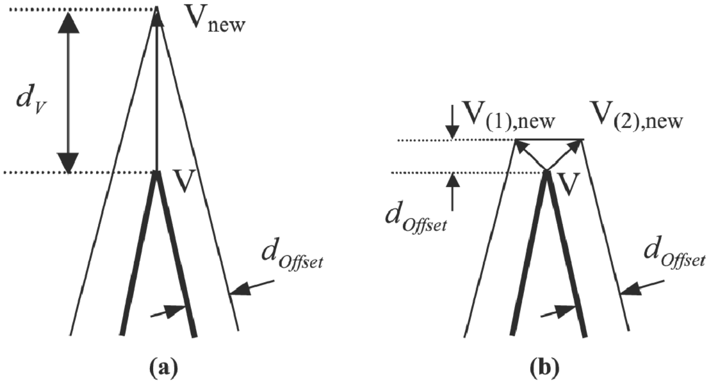


Figure 4.24: Illustration of vertex splitting method. (a) Normal offset, and (b) offset with vertex splitting

The methodology used in this thesis works well for small offset values, where local and global self-intersections normally do not happen or is not a main issue. To handle the self-intersection problem while offsetting the model with relatively large values, some post processing needs to be done.

4.3 Future scope

Error and limitation of this thesis work can further be investigated and improved by method such as:

Self-intersection of offset vertex can be avoided by some methods like try and remove self-intersections in 2D space on a slice-by-slice basis, which is a suitable method for a layer based manufacturing process. A significant amount of work has been done in the area of removing self-intersections from 2D offset curves. [22][23][24]

Since STL contain error, the triangle patches will be shown in machined part even if the STL is highly refined and meshed. Many neutral format file like STEP and IGES can be used in place of STL. These format got advantage of better approximation of actual surface.

A new offset algorithm could be used on STL format that can be used with generalized cutter [15]. The vertex could be offset as per the distance of APT cutter (Automatically Programmed Tools) center as shown in figure 4.25. One can also use surface offset method to 5 Axis NC machine to machine intricate parts which cannot be machined by 3 Axis machine.

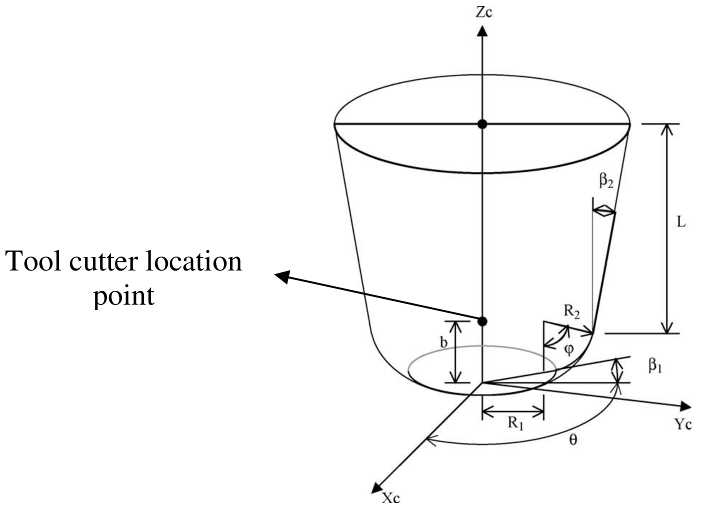


Figure 4.25: Generalized APT cutter

4.4 Conclusion

Based on the results presented in section 4.1, the following conclusions have been drawn.

The Triangular mesh offset methodology discussed above works well for sculptured surfaces however those surfaces which have sharp edges or in STL format which contains two or more than two triangular patch perpendicular to each other shows errors, as explained in section 4.2. By using mathematical equation and CAD techniques as discussed in section 3.1.3, a computer algorithm is developed and result has been analyzed both physically and graphically. After validating the analysis it is found that on vertices of triangular patch the algorithm gives the desired result but on the edge some undercut is developed, the extent of undercut depend upon the tool radius as the tool radius increases the depth of undercut increases, so for efficient use of methodology the tool radius should be small.

The tool path planning by Triangular mesh offset method developed for 3 Axis milling lathe and 3 axis vertical milling works only with Ball nosed cutter however it shows good results only when the radius of the tool is small whereas for larger tool radius undesired result or distorted surface geometry can be found as discussed in section 4.2 in detail.

REFERENCES

---

[1] P. Broomhead and M. Edkins, "Generation NC data at the machine tool for the manufacture of free-form surfaces", International Journal of Production Research, Vol. 24, No. 1, 1986, pp. 1-14.

[2] J. E. Bobrow, "NC machine tool path generation from CSG part representations", Computer Aided Design, Vol. 17, No. 2, 1985, pp. 69-76.

[3] Y.J. Chen and B. Ravani, Offset Surface Generation and Contouring in Computer-Aided Design, ASME, Journal of Mechanisms, Transmissions & Automation in Design, Vol. 109, March 1987, pp.132-142.

[4] Y. Huang and J. H. Oliver, "Non-constant Parameter NC Tool Path Generation on Sculptured Surfaces", ASME Computer in Engineering 1992, Vol. 1, pp 411-419.

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

[6] Nick P. Manos, Sanjeev Bedi, Dan Miller and Stephen Mann, " Single Controlled Axis Lathe Mill", August 11, 2005

[7] F. Badouel, "An efficient Ray-Polygon intersection", Graphic Gems, Academic Press, 1990, pp: 390-393.

[8] B. Koc and Y.S. Lee, (2002), "Non-uniform offsetting and hollowing objects by using biarcs fitting for rapid prototyping processes", Computers in Industry, Vol. 47 No. 1, pp. 1-23.

[9] Tong Wu, Edmund H and M. Cheung "Enhanced STL" Int J Adv Manuf Technol (2006) 29, pp. 1143–1150.

[10] S. D. Porumbescu, B. Budge, L. Feng, and K. I. Joy. "Shell maps" ACM Trans. on Graphics, 23(3) Aug. 2005, pp. 626-633,

[11] Xiuzhi Qu and Brent Stucker, "A 3D surface offset method for STL-format models", Rapid Prototyping Journal Volume 9, Number 3, 2003, pp. 133–141.

[12] Su-Jin Kim and Min-Yang Yang, "Offset Triangular Mesh Using the Multiple Normal Vectors of a Vertex", Journal of the Korean Society of Precision Engineering, Vol. 22, No. 9, September 2005.

[13] Matteo Malosio, Nicola Pedrocchi and Lorenzo Molinari Tosatti, “Algorithm to Offset and Smooth Tessellated Surfaces”, Computer-Aided Design and Applications 2009, pp. 351-363.

[14] Daoshan OuYang and Hsi-Yung Feng, “Computer-Aided Design and Applications”, Computer-Aided Design & Applications, 5(1-4), 2008, pp. 254-265.

[15] Su-Jin Kim and Min-Yang Yang, “Triangular mesh offset for generalized cutter”, Computer-Aided Design 37 (2005), pp. 999–1014.

[16] Kai Tang, Charles C Cheng and Yakove Dayan, Computer-Aided Design, Vol. 27 (1995), No. 12, pp. 915-927.

[17] Bahattin Koca and Yuan-Shin Leeb, “Non-uniform offsetting and hollowing objects by using biarcs offsetting for rapid prototyping processes”, Computers in Industry 47 (2002), pp. 1-23.

[18] Yunfeng Bu, Quanping Sun, Xiaogang Chen, Qianliang Chen, Xiang Zhang, “Research on the Algorithm of Tool Path Rapid Generation for NC Machining of Molar Prosthesis”.

[19] Computer-Aided Design and Computer Graphics, 2007 10th IEEE International Conference on 11/2007.

[20] Su-Jin Kim, Dong-Yoon Lee, Hyun-Chul Kim, Sung-Gun Lee and Min-Yang Yang, “CL surface deformation approach for a 5-axis tool path generation”, Int J Adv Manuf Technol (2006) 28, pp. 509–517.

[21] Xiuzhi Qu and Brent Stucker, “Raster milling tool-path generation”, Rapid Prototyping Journal, Volume 12, Number 1, 2006, pp. 4–11

[22] Bala, M. and Chang, T.C. (1991), “Automatic cutter selection and optimal cutter path generation for prismatic parts”, International Journal of Production Research, Vol. 29 No. 11, pp. 2163-76.

[23] R.F. Rohmfeld, “IGB-offset for plane curves – loop removal by scanning of interval sequences”, Computer Aided Geometric Design, Vol. 15 No. 4, 1998, pp. 339-75.

[24] W. Tiller and E.G. Hanson, “Offset of two dimensional profiles”, IEEE Computer Graphics and Applications, Vol. 4, 1984, pp. 36-46.

[25] Tomas Moller and Ben Trumbore, “Fast, Minimum Storage Ray/Triangle Intersection”, Journal of Graphics Tools, Volume 2 , Issue 1 (1997), pp. 21 – 28

[26] Rafael J. Segura1 and Francisco R. Feito, “algorithms to test ray-triangle intersection comparative study”, WSCG 2001.Conference Proceedings. Pilsen. 2001.

[27] Su-Jin Kima and Min-Yang Yang, “Incomplete mesh offset for NC machining”, Journal of Materials Processing Technology 194 (2007), pp. 110–120.

[28] M. Snyder and A. H. Barr, “Raytracing complex models containing surface tessellations”, Proceedings of the 14th annual conference on Computer Graphics, Vol.21, 1987, No.4, pp.119-128.

[29] R. J. Segura and F. R. Feito, “An algorithm for determining intersection segment-polygon in 3D”, Computer & Graphics, Vol.22, 1998, No.5, pp.587-592.

[30] http://visor.uwaterloo.ca/~krmoule/webcad,