

A Methodology for Simulation and Verification of Tool Path Data for 3-Axis and 5-Axis CNC Machining

A Dissertation Submitted
In Partial Fulfilment of the Requirements
for the Degree of

Master of Engineering

in

CAD/CAM Engineering

by

Vishaldeep Singh
801281022



to the

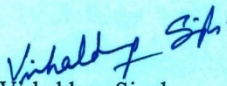
**MECHANICAL ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA**

July, 2014

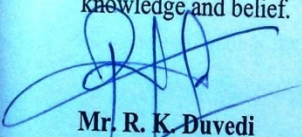
CERTIFICATE

I hereby certify that the thesis entitled "A Methodology for Simulation and Verification of Tool Path Data for 3-Axis and 5-Axis CNC Machining", is an authentic record of my study carried out as requirements for the award of the degree of **Master of Engineering in CAD/CAM Engineering** at **Thapar University, Patiala** under the supervision of **Mr. R. K. Duvedi**, Assistant Professor, Mechanical Engineering Department, Thapar University, Patiala during July, 2013 to July, 2014. The matter embodied in this report has not been submitted in partial or full to any other university or institute for the award of any degree.

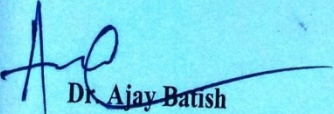
Date: 16/07/2014



Vishaldeep Singh

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.


Mr. R. K. Duvedi
Assistant Professor, MED,
Thapar University, Patiala -147004

Countersigned by:


Dr. Ajay Batish
Professor & Head
Mechanical Engineering Department
Thapar University Patiala - 147004


Dr. S. K. Mohapatra
Dean of Academic Affairs
Thapar University Patiala - 147004

ACKNOWLEDGEMENT

I express my sincere gratitude to **Mr. Ravinder Kumar Duvedi, Assistant Professor, Mechanical Engineering Department, Thapar University, Patiala**, for their valuable guidance, proper advice and constant encouragement during the course of my work in 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**, for their help, inspiration and moral support which went a long way in successful completion of this dissertation.

I used CAD software “**SolidWorks**” provided by the **State Initiated Design Centre (SIDC), Mechanical Engineering Department, Thapar University** for the completion of my dissertation.

(VISHALDEEP SINGH)

ABSTRACT

A lot of research has been going on in the field of simulation of the tool path. The verification of the tool path using simulator is fast process as compared to the traditional method and this is what forms the focus of research.

The dissertation is focused on the development of MACRO in SolidWorks environment for the simulation of tool path data for 3-axis and 5-axis machining. A CAD environment can be programmed using API for CAD automation. The tool path data is generated from the STL file of the original part. The work involves the simulation and verification of the tool path generated. The tool path generated is simulated using the Boolean approach for which a macro for 3-axis and 5-axis tool positioning has been developed. The output of the Boolean operation from the simulator environment has been saved in STL format for easy comparison with the original part.

Two different parts were used to verify the results. The tool path is generated for these two parts and the data thus generated is simulated using the methodology discussed in the dissertation. The results are discussed for the given parts are discussed in chapter 6. The methodology works well for 3-axis and 5-axis machining.

The ultimate aim of the present work is to develop the simulator using CAD environment and verify the results generated from the simulator and validate the results using Ray tracing method to prove that the output generated from the simulator is valid.

TABLE OF CONTENTS

CERTIFICATE	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	x
1. INTRODUCTION	1
1.1 NC tool path data.....	1
1.2 Tool path generation.....	2
1.3 Need for tool path verification	3
1.4 Methods for tool path verification.....	4
1.5 Requirement of NC tool path simulation	4
1.6 Capabilities required in an ideal simulator.....	5
1.7 Commercial solutions.....	6
1.8 Present Work	6
2. LITERATURE REVIEW	8
2.1 Wireframe based NC machining simulation.....	8
2.2 Solid based NC machining simulation.....	8
2.2.1 CSG based NC machining simulation	8
2.2.2 B-rep based NC machining simulation.....	9
2.3 Object space based NC machining simulation.....	10
2.3.1 Z-map method.....	10
2.3.2 Vector method.....	11
2.3.3 Octree based method	11
2.4 Image space based NC machining simulation	12
2.5 Solid modelling based NC simulation of machining	12
2.6 Generation of swept volume	14
2.7 Ray tracing method	15
2.8 Conclusion of literature review	16
3. CONCEPTUAL FRAMEWORK FOR NC SIMULATOR DEVELOPMENT	17
3.1 Need for machining simulator.....	17
3.2 Advantages of machining simulator.....	18
3.3 Parametric CAD software.....	19

3.4 SolidWorks API	19
3.4.1 Benefit of API	20
3.5 Macro in SolidWorks	20
3.5.1 SolidWorks macros	21
3.5.2 General steps for developing macros	21
3.5.3 Record, run, edit and debug the macro	24
3.5.4 Visual basic editor environment	25
3.5.5 Variables	26
3.6 Use of Microsoft Excel for SolidWorks API	26
4. METHODOLOGY	29
4.1 Inputs required for macros NC simulator	29
4.2 Raw stock	29
4.2.1 Pseudo code for generating raw stock	30
4.2.2 Flow chart for macro designed for tool shape selection	30
4.3 Tool shape	31
4.3.1 Pseudo code for generating tool shape	34
4.3.2 Flow chart for macro designed for tool shape selection	35
4.4 Tool path	36
4.5 Type of machining	37
4.5.1 Three-Axis machining	38
4.5.2 Five-Axis machining	38
4.6 Verification	39
4.6.1 Visual verification	40
4.6.2 Analytical verification	40
5. WORKING OF SIMULATOR MACRO	42
5.1 Working of three axis simulator macro	42
5.1.1 Pseudo code for 3-axis simulator macro	44
5.1.2 Cutter motion in space	46
5.1.3 Output	47
5.1.4 Flow chart showing the organisation of design macros for 3-axis simulation	49
5.1.5 Limitations	50
5.2 Working of five axis simulator macro	50
5.2.1 Pseudo code for 5-axis simulator macro	51
5.2.2 Cutter motion in space	53

5.2.3	Output	54
5.2.4	Flow chart showing the organisation of design macros for 5-axis simulation.....	56
5.2.5	Limitations.....	56
6.	RESULTS AND DISCUSSION	57
6.1	Input parameters for simulation.....	57
6.2	Visual verification.....	59
6.2.1	Results of 3-axis machining simulation	59
6.2.2	Results of 5-axis machining simulation	61
6.3	Analytical verification.....	63
6.3.1	Study of verification for 3-axis and 5-axis tool path	64
6.3.2	Study of scallop height in 3-axis and 5-axis tool path.....	66
7.	CONCLUSION AND FUTURE SCOPE.....	72
7.1	Conclusion	72
7.2	Future scope.....	72
ANNEXURE 1	74
ANNEXURE 2	76
REFERENCES	78

LIST OF FIGURES

1.1 Path topologies of the freeform surfaces.	3
1.2 Planning of NC machining simulation.....	5
1.3 Need of developing simulator.....	7
2.1 Octree Model	12
2.2 Simulation of machining process using boolean approach	13
2.3 Removal of tool swept volume from raw stock using boolean subtraction approach .	14
2.4 Swept volume of torus end mill cutter.....	14
2.5 Grazing curve (the thick line) for the tool along tool direction.....	15
3.1 SolidWorks API.....	20
3.2 Macro Toolbar	22
3.3 Output of macro for torus end mill tool.....	24
3.4 Project Viewer	25
3.5 Module Viewer	26
3.6 Excel sheet showing the x, y and z position co-ordinate.....	27
3.7 Output of macro for importing data from excel to SolidWorks.....	28
4.1 Raw stock.....	30
4.2 Flowchart for generation of raw stock	31
4.3 Tool Shapes	32
4.4 Torus end, flat end and ball end mill cutter respectively.....	32
4.5 Sketch to generate torus end mill cutter using macro	33
4.6 Final tool shape of torus end mill cutter	33
4.7 Final tool shape of flat end mill cutter	34
4.8 Final tool shape of ball end mill cutter	34
4.9 Flowchart for the generation of tool shape	35
4.10 CL Data required in Excel File for 3 axis machining.....	36
4.11 Data required in Excel File for 5 axis machining	37
4.12 Two tool positions.....	39
4.13 Ray tracing method.....	40
5.1 User form for 3-axis machining simulator.....	42
5.2 Work piece generated in modeller	43
5.3 Torus end mill cutter generated in modeller	44
5.4 Torus end mill cutter moving through work piece.....	46

5.5 Torus end mill cutter moving through space along CL trajectory	47
5.6 Volume swept by the cutter along curve.....	47
5.7 Output for tool path in Annexure 1 for torus end mill cutter.....	48
5.8 Output for tool path in Annexure 1 for flat end mill cutter	48
5.9 Output for tool path in Annexure 1 for ball end mill cutter.....	48
5.10 Output for tool path in Annexure 1 by Boolean operation	49
5.11 Flowchart explaining the flow of control of macro	49
5.12 User form for 5-axis machining simulator.....	50
5.13 Torus end mill cutter moving through work piece.....	53
5.14 Swept volume of tool for tool	53
5.15 Bottom surface of tool	54
5.16 Torus end mill cutter moving through space along CL trajectory.....	54
5.17 Output for tool path in Annexure 2 for torus end mill.....	54
5.18 Output for tool path in Annexure 2 for flat end mill cutter	55
5.19 Output for tool path in Annexure 2 for ball end mill cutter.....	55
5.20 Output for tool path in Annexure 2 by solid sweep methodology.....	55
5.21 Flowchart explaining the flow of control of macro	56
6.1 Machined part 1	57
6.2 Generation of STL file for part 1	58
6.3 STL for part 1.....	58
6.4 Machined part 2	58
6.5 Generation of STL file for part 2	59
6.6 STL for part 2.....	59
6.7 Output of part 1 using 3-axis machining simulation.....	60
6.8 Visual verification of part 1 using 3-axis machining simulation.....	60
6.9 Output of part 2 using 3-axis machining simulation.....	60
6.10 Visual verification of part 2 using 3-axis machining simulation.....	61
6.11 Output of part 1 using 5-axis machining simulation.....	61
6.12 Visual verification of part 1 using 5-axis machining simulation.....	62
6.13 Output of part 2 using 5-axis machining simulation.....	62
6.14 Visual verification of part 2 using 5-axis machining simulation.....	63
6.15 Ray tracing of part	63
6.16 Comparison of part 1 original machined part and 3-axis simulated part.....	64
6.17 Comparison of part 1 original machined part and 5-axis simulated part.....	64

6.18 Comparison of part 2 original machined part and 3-axis simulated part.....	65
6.19 Comparison of part 2 original machined part and 5-axis simulated part.....	65
6.20 Plot for the scallop height of part 1 at y=10mm for 3-axis simulated part.....	66
6.21 Plot for the scallop height of part 1 at y=20mm for 3-axis simulated part.....	67
6.22 Plot for the scallop height of part 1 at y=30mm for 3-axis simulated part.....	67
6.23 Plot for the scallop height of part 1 at y=10mm for 5-axis simulated part.....	68
6.24 Plot for the scallop height of part 1 at y=20mm for 5-axis simulated part.....	68
6.25 Plot for the scallop height of part 1 at y=30mm for 5-axis simulated part.....	68
6.26 Plot for the scallop height of part 2 at y=10mm for 3-axis simulated part.....	69
6.27 Plot for the scallop height of part 2 at y=20mm for 3-axis simulated part.....	69
6.28 Plot for the scallop height of part 2 at y=30mm for 3-axis simulated part.....	70
6.29 Plot for the scallop height of part 2 at y=10mm for 5-axis simulated part.....	70
6.30 Plot for the scallop height of part 2 at y=20mm for 5-axis simulated part.....	70
6.31 Plot for the scallop height of part 2 at y=30mm for 5-axis simulated part.....	71

LIST OF TABLES

4.1 Parameters used to generate work piece	30
4.2 Parameters used for creating tool	33
6.1 Input parameters for machining simulation	57

CHAPTER 1

INTRODUCTION

In the present day's technology advancement and competition in the machining industries every industry is opting for the implementation of CNC machining and machining simulators all over the world. As present machining scenario demands for high accuracy and reduced machining losses, NC machining is a fundamental and important controlled machining process for the production of parts. The numeric control machine works on an automatic mode. The verification of the tool path is important for the automatic operation of the machine. Therefore, it is very important to verify the NC program for the given tool path before the actual machining. The simulators are the solution for all such problem.

The tool path simulators are developed in which simulation of NC program containing the tool path data is carried out. The simulation of the tool path can be done and the physical shape is verified on the simulator.

For the automation of the machining process, Computer numerical control (CNC) has a great contribution towards it. As the products to be machined are more complex the essential accuracy for machining of products is increasing. In CAD system the surface of the product is formed by the curves. These curves may be B-spline, Bezier or NURBS (Non-Uniform Rational B-Spline) curves. The arrangement of these curves will create surface and the surface is formed from line segments or arcs in CAM systems. The data generated of these segments and the commands thus generated are NC tool path data.

A tool path is a set of commands that explains the location a tool when cutting the part. The tool path is not formatted for the machine. It cannot be directly sent to operate on the machine [31]. The tool path data generated from model cause many problems. The local adjustment or modification of surface is not possible because of the excessive size of NC tool path data. To overcome these problems there is a need of curve or surface interpolator [27] which can interpret or interpolate a surface formed from a curve as a unit.

1.1 NC TOOL PATH DATA

In the present day scenario, the precision and accuracy of machining of products is increasing. The shapes of the products are also increasing along with the increase in complexity of the product. The products to be machined are represented by the surfaces. These surfaces which are to be machined are formed by number of curves. These curves are further divided into lines and arc segments. The data is then generated in the form of cutter

location (CL) points or cutter contact (CC) points. This data generated is NC tool path data. This tool path data indicates the location of the tool with respect to some reference frame during the machining process.

It is required for the generation of commands for the machining. This tool path data is sent to the machines in the form of codes for the cutting operations. Also this data is used for the NC virtual machining process. This data is used as the input for the simulation process and the output generated will be the actual machined surface on the virtual environment. Also the data is used for verification of the machined surfaces.

1.2 TOOL PATH GENERATION

The planning of the tool path for machining of sculptured surfaces is a crucial task. For quality machining there are definite constraints for different machining stages while planning for the tool path generation. In an ideal tool path, while machining the scallops produced must be distributed evenly across the whole surface [14]. The tool path planning for the generation of tool path consists of two parameters. The first is path topology and the second is path parameters. The path topology is defined by the pattern of the tool movement along the path whereas the path parameters method is defined by the tool side step and tool forward step.

While machining the sculptured surfaces, the tool traces a series of cutter contact points; this is called tool path topology. The machining time of the sculptured surfaces is affected by the path topology. An accurate tool path generation can result in the minimum length of the path, minimum recantation and local refinement of the tool path generated so as to match the properties of machining surface [18]. The two path topologies are mostly used for the machining of the freeform surfaces or sculptured surfaces, the first one is direction parallel path topology and second one is contour parallel path topology.

The path segments in the direction parallel path topology are parallel to the defined line as shown in Figure 1.1(a). This line could be either parallel to the axis of a specified co-ordinate system or perpendicular to the surface. The commonly used path for roughing operation in commercially used CAM systems is zigzag path for machining [33].

In case of contour parallel path the boundary curve of the surface is considered to generate the path. The path generated could be connected to each other or could be independent of each other. This independent path must be offset of the boundary curve as shown in Figure 1.1(b) [33].

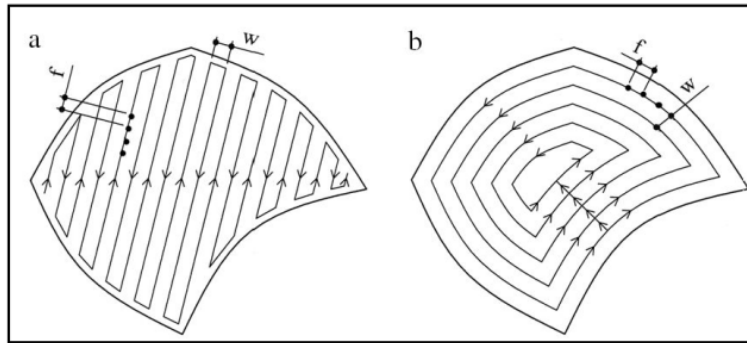


Figure 1.1: Path topologies of the freeform surfaces. (a) Direction parallel. (b) Contour parallel. [33]

The accuracy and time constraints are considered during the development of tool path. To evaluate the tool path generated there are three criteria which are as given below [6]:

Quality: The tool path generated taking care of the constraints must be free of gouge. The scallop height generated between the two passes must be within the specified tolerance.

Efficiency: There are two types of efficiencies that are measured during the tool path generation: (1) efficiency of tool path to develop simulation based on the time taken to simulate and the memory used during simulation in virtual environment, and (2) efficiency in the time of machining process. These efficiencies for tool path generation are achieved by different topologies and parameters; and simulating these with the help of system.

Robust: The system should work on different machine as well as for different surface that is to be machined. The system can work with different tool paths as well.

1.3 NEED FOR TOOL PATH VERIFICATION

After generating the tool path data it is necessary to be verified before sending the data for machining. The machining process is to change the raw stock to the finished part with the given geometry of the part. The surface finish and dimensional accuracy of the machined part is to be maintained. The demand of the quality products at reduced cost is increasing. This can only be achieved by machining with accuracy without wastage; also the machining speed should be more so as to produce maximum number of quality products. The demand of complex shaped parts is increasing, so machining of the complex part is done without the trial cut, and the production of the parts is done in lots and the lot size is smaller [35].

During production, to save time and money by making the products accurately, a proper tool path is required, so that when the raw material is allowed for machining with proper tool path an accurate product is formed in lesser time.

1.4 METHODS FOR TOOL PATH VERIFICATION

There are number of ways to verify the tool path generated for the complex geometries. The most common methods for the tool path verification are as explained below:

1. Dry run method: In this the machine is run without the raw material loaded on the machine. But the worker must be very much experienced to check this method based on his experience whether the part going to be machined will be correct or not.
2. Acid Test: In this method a soft material is loaded on the machine like thermocol or some soft material like sponge and then the machine is made to run. The tool passes over the soft material and forms the same part as that to be made by the raw material. In this way the tool path generated is checked.
3. Physical verification: In this method the surface of the part is machined and this machined surface is measured with the help of coordinate measuring machine (CMM) and compares the dimensions measured by CMM machine for the machined product with actual dimensions on the drawing.
4. Virtual Simulation: The machining of the product and measuring the coordinates of the machined surface is expensive process. All the above stated processes are time consuming so one would prefer to simulate the machining process by simulating the tool path generated from the part to be machined. The errors generated in the machined surface is measured by comparing the simulated part and the original part from which the tool path is generated. This will help to save a lot of time. The graphical user interface allows the system to display the simulation of the tool path that generates the machined part which allows the verification of the tool path. This approach can save a lot of time, which was earlier consumed for machine setup for verifying the tool path [28].

1.5 REQUIREMENT OF NC TOOL PATH SIMULATION

Machining simulation has been developed to reduce the material wastage and machining time by performing physical machining and then verifying the tool path. The verification of the tool path is necessary during machining simulation. The errors in the tool path can be predicted and rectified. The tool path can be modified by removing these errors [40].

Verifying an NC program/tool path for machining parts is often time-consuming and expensive. Tool path can be traced and viewed without doing the actual machining. This can save a lot of time to check the tool path.

In multi-axis NC machining the verification of the tool path of complex geometry is difficult. In multi-axis NC machining along with translatory motion the tool has at least one rotary motion such as rotary table, swivelling head, rotating column, and so on. A complex tool motion is generated when machine is programmed to rotate tool while moving along with the translational direction. To correct the errors in the tool path programming several tryouts for given tool path is performed on CNC machine tool.

1.6 CAPABILITIES REQUIRED IN AN IDEAL SIMULATOR

The simulator is based on 3D representation. The analysis of the accuracy of tool path is achieved by using simulator. The tool path data contains the co-ordinate values of the cutter location (CL) points and the co-ordinate values of cutter contact (CC) points along with the angle of inclination. The simulation of the milling process is used to verify the final output generated. The final work piece generated after the simulation process can be used to check the errors of the actual tool path. This estimation is done before the actual machining of the work piece to reduce the errors and trial cuts to verify the tool path data [19].

The simulator must perform for almost all the type of machine configurations. It must simulate for 5-axis as well as for 4-axis and 3-axis also. The results for tool path must be accurate.

The simulations of the machining process are divided into two categories:

- 1 Geometric Simulation: This approach is used to compare the results graphically. The result shows that whether the tool interferes with the fixtures, work piece, machine, etc. It is also used to check whether the tool cut into the part or leave extra material on the raw stock. Also it provides the entry and exit of the tool with respect to machine centre.
- 2 Physical Simulation: It is used to tell about forces required for cutting operation, vibration produced while machining, roughness of the machined surface, temperature of the machined surface and tool wear produced while machining [29].

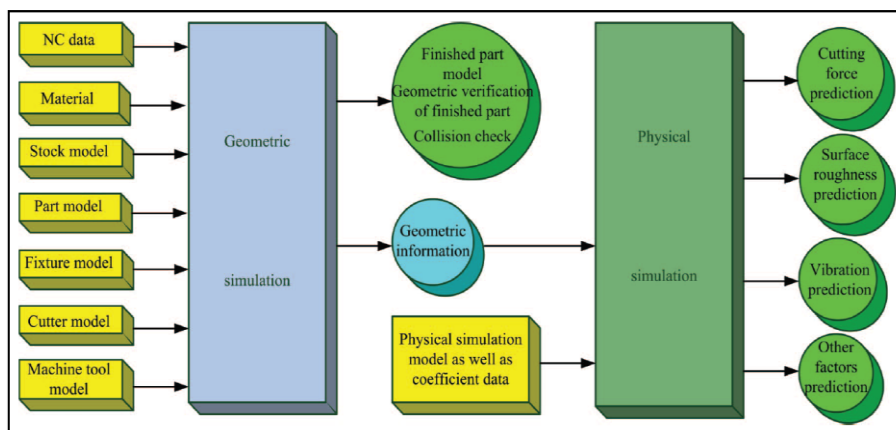


Figure 1.2: Planning of NC machining simulation [29].

The difference between geometric and physical simulation process is as shown in Figure 1.2.

In this report we are concentrating on geometrical simulation.

Certain capabilities that the machining simulator must have are given below:

- 1 Simulator must generate the graphical simulation. It must generate the relative motion of the tool with respect to the work piece.
- 2 Machine part in simulation must be saved in some format.
- 3 Easy comparison between the actual geometry and simulated machined part.
- 4 Errors in tool path should be visible clearly with some specific colour so that it can be easily detected during the run time simulation.

For generating a machining simulator there is a need of CAD package and MACRO proficient CAD package. So MACROS in CAD package can be used for simulating the tool path.

1.7 COMMERCIAL SOLUTIONS

In the present day scenario the CAM systems consists of the software which are used for the verification of tool path. These are basically based on z-buffer algorithm. These types of software's only provide the image of the machining process. For each image there is a need of recomputation. The advantage of the image space based approach is that the quick results are generated. The animation of the tool removing the excess material is possible using solid modelling approach. Some of the commercial packages for simulation of tool path include VERICUT, Predator Virtual CNC, MachineWorks etc. Generally the software's comes in modules. These allow only the graphical visualisation of the machining process. For the verification of undercut and overcut additional modules are available [24]. Some of the commercial solutions are

S.No	Name	Cost	Capability
1.	Vericut	\$7000+	Detects collisions and near-misses between all machine tool components
2.	Pro E	\$4000+	Manufacturing environment in the form of tool design and simulated CNC machining.
3.	Master CAM	\$8000+	
4.	Predator Virtual CNC	\$7000+	Verify CNC process before making scrap.

1.8 PRESENT WORK

A CAD package is required to model a product in the CAD modeller. The tool path is generated from the model generated in the CAD package. There is a need to verify the tool

path. So the big question is that how to verify the tool path. The traditional method is to machine the part using the tool path generated using CAD package. There are lot of errors in this machined part. With the help of this machined part the verification and the comparison is performed between the modelled part and the machined part. This data helps to generate the rectified tool path. The other way to verify the tool path is using NC simulator. There are separate software tools to simulate the tool path which are very costly. There is a lot of research work going on to simulate the tool path and compare it with the CAD model and verify the tool path. Verifying the tool path using NC simulator is fast process as compared to the traditional method as shown in Figure 1.3.

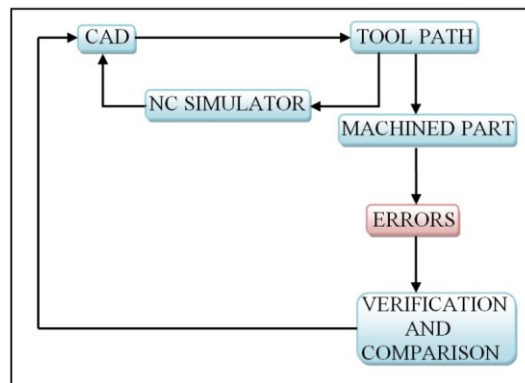


Figure 1.3: Need of developing simulator.

The scope of this dissertation work is to create an algorithm in a commercially used CAD package (SolidWorks) using MACRO which can simulate the tool path for the user defined work piece with given parameters and the user defined tool. There are commercial packages for machining simulation but are very costly. Most of the commercial software for the simulation process comes in modules. These allow only the graphical visualisation of the machining process. For the verification of undercut and overcut additional modules are available [24]. So taking care of the cost of the packages and the deficiency of commercially available solutions for simulation of tool path there is a need of developing a simulator. This system will verify the tool path generated for machining and the final output generated after the simulation process can be used to compare with the model from which the tool path is generated.

CHAPTER 2

LITERATURE REVIEW

This literature survey is done to support our methodology and the approach that we are going to use for developing simulator. The following literature review is divided into seven sections. In the section 2.1 NC machining simulation based on wireframe model is discussed. In section 2.2 the discussion is based on solid modeling which is further based on CSG and B-rep technique. In section 2.3 the discussion is based on object spaced which is further divided into Z-map, vector method and octree based. In section 2.4 the discussion is based on image space. In section 2.5 the discussion is based on solid modeling based simulation of machining. In section 2.6 the discussion is based on the generation of swept volume.

2.1 WIREFRAME-BASED NC MACHINING SIMULATION

In this type of simulation, the contour of work piece which is machined and tool path displayed are in the form of wireframe. The simulation based on wireframe model performs the operations of the given data fast and the data structure is simple. This method was widely applied for NC machining simulation. However this model does not generate actual solid model. Also it makes three dimensional objects and its data very complex. Therefore this simulation is useful for generating simple products.

2.2 SOLID-BASED NC MACHINING SIMULATION

Solid modelling generate complete 3D model. The simulation of machining process by this method is helpful for both cases that is geometric and physical simulation. The tool and the work piece can be accurately represented in solid based NC simulation.

2.2.1 CSG based NC machining simulation

1. **Hunt and Voelcker (1982)** stated that the Part and Assembly Description Language (PADL) based system is used for simulation of 2.5-axis machining. They used mathematical models for the geometrical effects. The simulation of the tool and the work piece is based on CSG technique [1].
2. **Spence et al. (1990)** and **Spence and Altintas (1994)** stated that the CSG technique is used for the simulation of 2.5 axis machining. This simulation consists of both geometric and physical simulator. The method used by them for the simulation process is that to directly calculate the intersection of work piece and tool. The

intersection data thus generated is used for CSG part description tree to simulate the machining process [2, 7].

3. **Su et al. (1999)** and **Ho et al. (2001)** proposed the collision detection during the machining process. The methods realised for the efficient and precise collision detection are CSG ‘divide-and-conquer’ concept and the collision detection by the distance aided method for convex bounding volumes. The other method used was heterogeneous representation. In this method CSG was used to represent the cutter and the work piece was represented by the cloud of more than 10,000 points, which was used to detect the penetration depth and the collision of tool in work piece [12, 15].

2.2.2 B-REP based NC machining simulation

1. **Requicha and Rossignac (1992)** stated that the solid modelling is done by B-rep technique. The vertices, edges and surfaces of the part are represented explicitly. They found wide applications in design and manufacturing using the solid modelling based on B-rep technique [4].
2. **O’Connell and Jabolkow (1993)** stated that the simulated machined part formed from the NC program is based on B-rep solid model. NC program contains Cutter Location (CL) data which is used for simulation of the machining process. This simulator is only based on the geometric simulation of the product [5].
3. **Mounayri et al. (1998)** developed a imitation system for 3 axis milling by ball end mill cutter which is based on solid modeller. A part was developed by solid model using B-rep technique. The cubic bezier curves were used to show the cutting edges of the tool. The volume swept by the tool is generated and is intersected with the raw stock for every tool path of one NC block. To generate the tool and work piece immersion geometry, the cutting edge was intersected with the removed volume of the work piece [10].
4. **Imani and Elbestawi (2001)** and **Mounayri et al. (2002)** they developed the geometric simulation as well as the physical simulation. There is good agreement between the simulation of the work piece and experimentally generated results [16, 20]. The swept volume of the cutter is generated using sweeping technique or by using skinning technique. And then Boolean operation is performed on the work piece and the swept volume [16]. A new and general methodology is used for modelling the part for the milling process. The Artificial Neural Networks process is used for

simulation of machining and Particle Swarm Optimization is used for optimization purpose [20].

5. **Fleisig and Spence (2005)** stated that the machining simulation based on B-rep technique is more time consuming because of the increased part complexity. So to overcome this problem parallel processing techniques was used. The total numbers of tool path are grouped into pairs and then the paired tool paths are simulated in parallel [26].
6. **Yip-Hoi and Huang (2006)** used a semi cylinder to represent the tool and the B-rep method is used to model the part. The tool intersection is used to characterise the tool or work piece intersection geometry. The solution based on solid modelling is used for the computation of tool and work piece intersection geometry when multiple tool changes and multiple tool positions are there. The cutter engagement is identified for each step of cutter during machining [30].
7. **Ferry and Yip-Hoi (2008)** continued Yip-Hoi and Huang (2006) research to 5-axis machining. The parallel slicing methodology is used. This method generates the swept volume of the material removed by the cutter from work piece by Boolean operation. The cutter work piece engagement calculation is performed along the tool path for every cutter location and its orientation [36].

2.3 OBJECT SPACE-BASED NC MACHINING SIMULATION

The representation of part in this methodology is done by grouping points which are discrete or certain volume elements or surfaces with vectors. Because of the discretisation of objects, Boolean operations between parts are 2-D or may be 1-D as a result of which simulation computation improves.

2.3.1 Z-Map method

The primary use of the Z-map method is the decomposition of the part into various three dimensional histograms. The height of the part is the initial point of each three dimensional histogram. The height of the histogram is updated by each tool movement, provided it cuts the present height of histogram. As a result of this, the simulation speed by Boolean operation is very fast. This results in the 1-D system which increases the speed of simulation process. The function of 3D histogram is to detect the collision in NC machining by approximating the billet and the cutter assembly shape. If the tool is not perpendicular to the cube, the provision for 4- or 5-axis machining ceases. This simulation method is called Z-map which was used to obtain computation efficiency, better precision and quality.

1. **Lee and Lee (2002)** stated that to obtain a smoother rendering along with a screening ability of the Z-map model in a 3-axis machining simulation. They used a local mesh method for the simulation process. It helps in increasing the real time simulation [21].
2. **Lee and Ko (2002)** stated that to improve the Z-map model, the inclined sampling method was used. This technique is another version of anti-aliasing and it is used to increase the efficiency of simulation algorithm along with mesh rendering in computer graphics [22].

2.3.2 Vector method

In vector method, the facade of the part is represented by points. The direction vectors on these points are perpendicular to the surface. A direction vector keeps on extending till it intersects with another surface of the part or till it reaches the edge of the stock. A calculation of the meeting point of each direction vector and tool movement is made in order to simulate the machining process. If a vector intersects the envelope, then its length is reduced. In the simulation, each direction vector represents the edge of grass growing from the surface and the final length of the vector represents the amount of material removed at that point. Therefore, the vector method helps in judging the undercut and overcut of the part.

2.3.3 Octree-based method

In an Octree scheme, parts are represented in a tree structure. True to its name, in this method, the representation of the model is in the form of a tree consisting nodes. If the base node is partially occupied, these nodes are further subdivided into eight nodes.

1. **Karunakaran and Shringi (2007)** stated that in the machining simulation the formation of the part and amendment in the part was done by octree method. Then for visual verification, optimisation and animation of the machining simulation is done by B-rep method. The objects are stored in the machining environment (in which octree representation is used), a volumetric NC system was developed [32].
2. **Dyllong and Grimm (2008)** stated that the nodes of octree are divided into eight nodes. These nodes are compared with the cuboids. Each node contains the objects of the part. Every node of the octree is checked whether it is fully or partially occupied or the node is empty. If the node is fully occupied or empty it is not further divided but if the node is partially occupied by the objects of the part then the node is further divided into eight nodes. The node is subdivided as shown in Figure 2.1 if it is partially occupied [37].

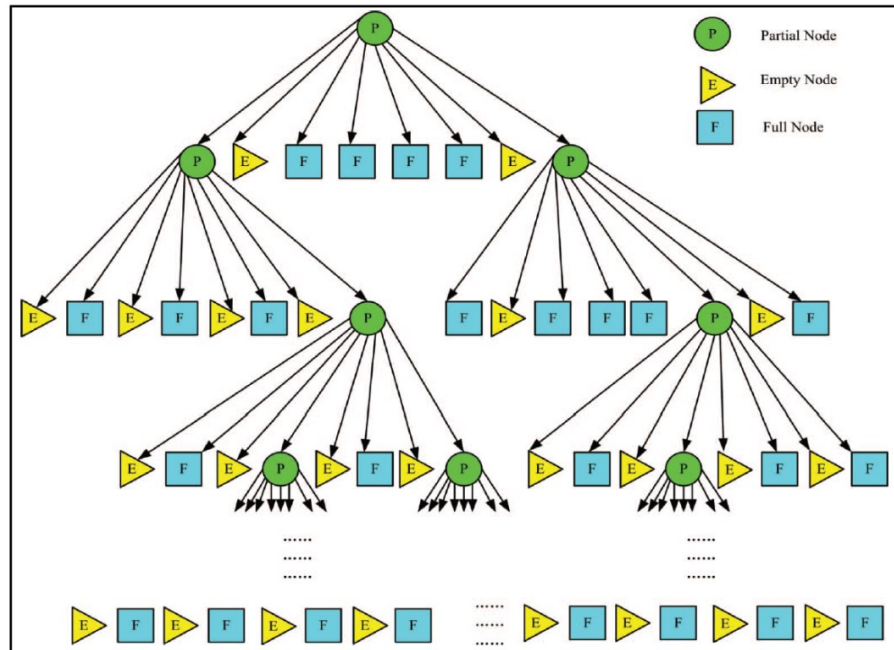


Figure 2.1: Octree Model [37]

2.4 IMAGE SPACE-BASED NC MACHINING SIMULATION

The representation of part in the machining simulation is done by the depth pixel model. This model is also known as dexel. The word “Doxel” is the product of combination of ‘depth’ and ‘pixel’. The comparison between dexel models is used for machining simulation. The simulation based on dexel model is fast. The dexel model is one dimensional because the model is generated with the help of pixels. The Boolean operation performed on model which is in one dimension results in fast simulation process. This approach is used by the Vericut simulator. The simulation process based on image space is faster among the other simulation methods. This model can well be seen as an expansion of Z-buffer which is primarily used for the eradication of unseen surfaces in graphics.

2.5 SOLID MODELLING BASED SIMULATION OF MACHINING

It involves all the geometric entities that are the shape of the part and tool, volume swept by the tool and the cutting edges of the part in machining simulation process. A volume swept by the tool is generated for each implemented tool shape and path motion in this approach. To obtain the machined material, the volume swept by the tool is intersected by the raw stock. And the simulated part is generated by subtracting the volume swept by the tool from the raw stock.

1. **Mounayri et al. (1997)** stated that the simulation of a machining process for a given tool path using a solid modelling approach is shown in Figure 2.2. The model of the tool swept volume is generated and is intersected with the raw stock. The final

simulated part is generated by subtracting the volume swept by the tool from the raw stock [8].

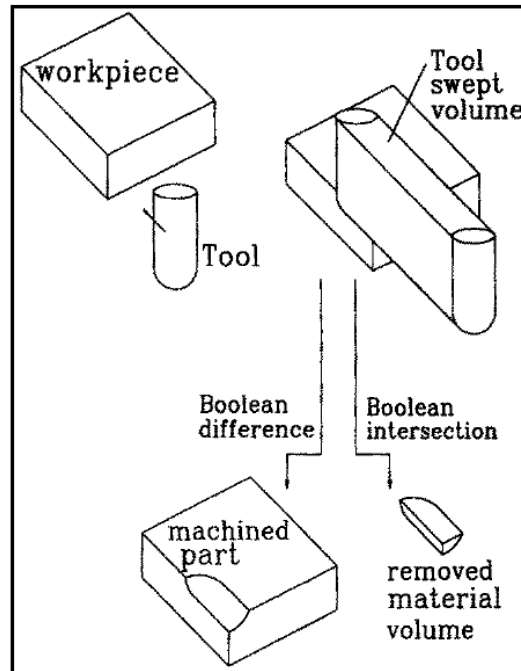


Figure 2.2: Simulation of machining process using boolean approach [8].

2. Lee and Nestler (2008) stated that the machining of the product is the material removal process with a geometry defined cutting tool. In order to verify the tool path, the tool swept volume (SV) is continually subtracted from the raw stock in the virtual environment during simulation.

A series of attempts takes place to get swept volume and to verify the NC milling based on swept volume of a moving tool. By mathematical description and practical implementations, these attempts can be divided into three categories:

- 1) Line based (1-D),
- 2) Surface based (2-D)
- 3) Volume based (3-D)

The function of line based (1-D) method is that it is used to find the intersection between the tool body and rays.

The focus of surface based (2-D) method (based on boundary representation (B-rep) technique) is to find the boundary description of the tool swept volume.

The volume based (3-D) method is based on constructive solid geometry (CSG). This helps in pointing out the quality of reproducing the amount of the geometric appearance and robustness for the Boolean operation [38].

3. Sullivan et al. (2012) stated that the boundary representation (B-rep) is used by the solid modelling based simulators in order to show case the milled work piece and in order to perform the Boolean subtraction operation between a work piece and swept volume of the tool between two positions. These simulators give accurate results for machining simulation. However, its drawback is that requires a high computational cost like data storage, time and complexity, as represented in Figure 2.3 [41].

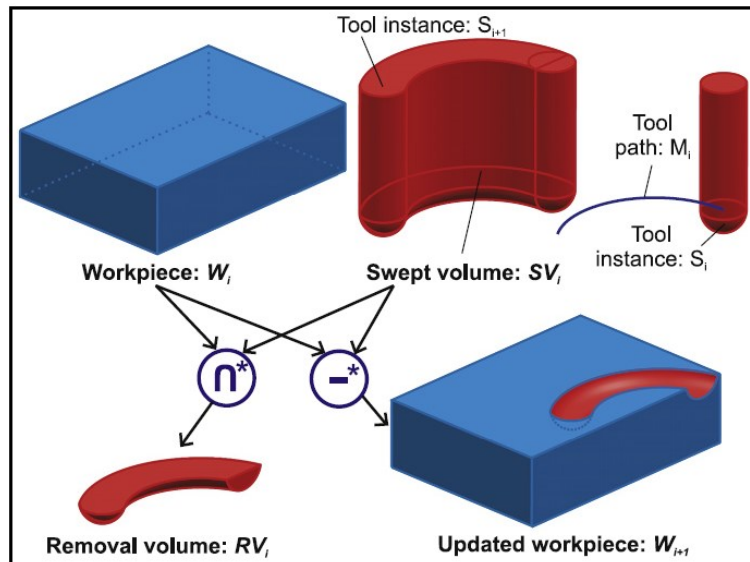


Figure 2.3: Removal of tool swept volume from raw stock using boolean subtraction approach [41].

2.6 GENERATION OF SWEEPED VOLUME

1. Sheltami et al. (1998) stated that the simulator developed use the swept volume of the cutter for the milling operation and then perform the Boolean operation between the work piece and swept volume of the tool.

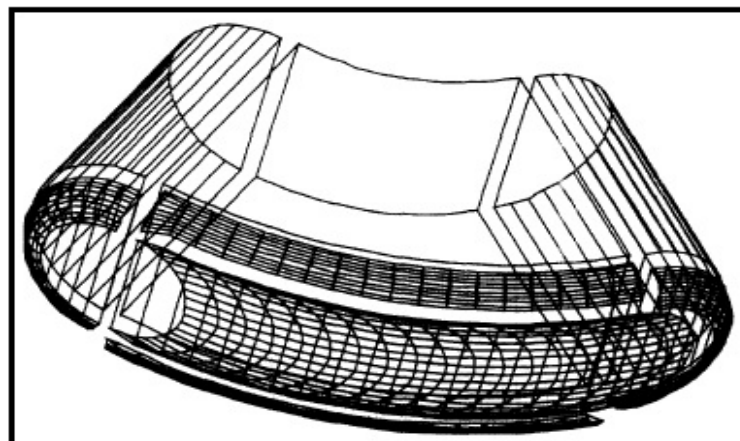


Figure 2.4: Swept volume of torus end mill cutter [11].

This technique is based on recognizing the generating curve along the tool motion, then to make a model from these generating curves to form a swept volume of the

cutter. The swept volume for the torus end mill cutter with the help of generating curve is as shown in Figure 2.4 [11]

2. **Roth et al. (1999)** stated that the tool swept volume is generated using the space curves. A curve is identified on the tool surface for a given tool location. This is the impression of the tool on the surface to be machined. The imprint space curve is identified for each tool location and volume swept by the tool is generated [13].
3. **S. Mann and S. Bedi (2002)** extended the imprint method for finding the swept volume for the surface swept by the tool for the machining of the curved surfaces. The grazing points are calculated on surface of revolution and with the help of these grazing points, grazing curve is generated for the cutter. The grazing curve for the tool is as shown in Figure 2.5 [23]

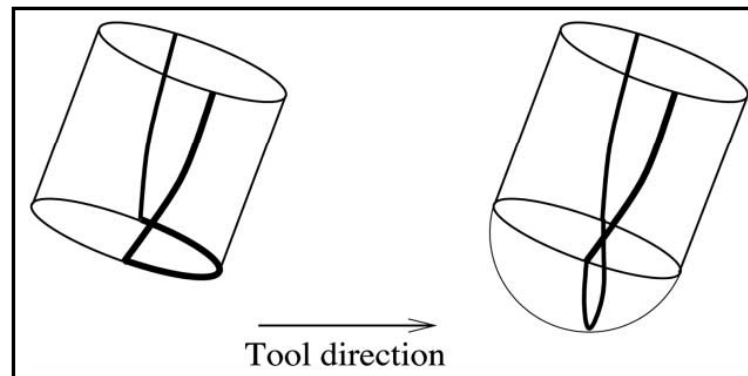


Figure 2.5: Grazing curve (the thick line) for the tool along tool direction [23].

2.7 RAY TRACING METHOD

1. **D. Badouel (1990)** proposed the study to find the intersection between the rays and triangles. In his study he used the barycentric co-ordinates to find the intersection between the line and triangles. The algorithm proposed by him uses the equation of line and triangles and find the intersection of line with each triangle. The limitation of his methodology is that the execution time of the algorithm is not penalized [3].
2. **T. Moller and B. Trumbore (1997)** proposed the study that whether the ray intersects the triangle or not. He extended the Badouel (1990) algorithm by creating the series of triangle transformations. The advantage of this algorithm is that the computation time is less as compared to previous methodology [9].
3. **Segura and Feito (2001)** proposed the algorithm to test the intersection of ray-triangles. They proposed the study in which no errors were involved. They used the sign of triangles in the algorithm to find the intersection. Where there is intersection

the point of intersection can be calculated. The complex calculations are not there in this algorithm, so the results computed are very much precise and accurate [17].

2.8 CONCLUSION OF LITERATURE REVIEW

In the above literature, a brief discussing of papers has been studied with the aim to get customized solutions for developing a simulator to simulate the tool path. A CAD environment can be programmed using API (Application Programming Interface) for CAD automation. This capability has been explored for NC tool path simulator design. But the methods explained in the literature are not universal and do not allow the comparison of machined part geometry with the original model. Thus in the present work an attempt has been made to use API environment of SolidWorks to generate macros for 3-axis and 5-axis NC tool path verification using Boolean and solid sweep operation. The machined/ simulated part is thus available within the CAD modeller and hence can be compared for the verification of NC tool path. The visual and analytical verification of the tool path is possible. The NC tool path has been generated from STL data file of the CAD model. Thus after NC simulation in macro the machined part is also taken in STL format for further analytical verification. The machined part can also be compared for visual verification using SolidWorks modeller environment.

CHAPTER 3

CONCEPTUAL FRAMEWORK FOR NC SIMULATOR DEVELOPMENT

Simulation is the replication of the process of a real-world process or system over time. It is the technique, for representing the real world problems with the help of computer programs. The simulation requires a model which represents the key characteristics of the system which is similar to real life situation. The computer based simulation of a model which is similar to real life situation can be used to study that how system works in real life. The model generated is the system and simulation mimics the operation of the system. By changing the variables in simulation we can predict the behaviour of the system and can make changes to the system as required. It is a virtual tool used to investigate the behaviour of the system under study.

A simulator is a process that runs on a computer to give a better idea for the working of system or machine. In real life, testing under different conditions is not easy, so a computer simulator recreates the real life situation on which the testing is conducted safely. Also a lot of time is saved while working on computer simulator as compared to physical working in the real world. The in-depth testing of the system is also possible.

Machining simulation is the 3D representation of the work piece and the cutting tool. It is used to simulate the tool path for machining process for verifying the tool path against the actual shape required. Machining simulation thus verifies the tool path without the need of actual machining of the work piece. Simulation results are used to make decision for the real life machining process.

3.1 NEED FOR MACHINING SIMULATOR

In the present day life worker's safety is most important during a machining process. As the use of high speed machines is growing very fast, an error in machining NC program can cause serious accident.

The demand for high accuracy and reduced machining losses in the industry is very high. CNC machines have become a necessity for industries to remain in such a competitive market. Therefore every industry is implementing CNC machines and machining simulators all over the world. Although the initial cost of these machines is very high but they have a lot of advantages in terms of accuracy and production rate as compared to the conventional machines.

Further the surface topological information may be of great importance for a particular application which if tested after doing physical machining on the work piece will make the process iterative and costly.

The performance of various tool paths can be checked and one with better results can be chosen for actual machining.

So if you do not have a machining simulator you may not be able to:

- a) Ensure full worker safety.
- b) Ensure high machining accuracy.
- c) Know the surface topological information.
- d) Check the performance of alternative tool paths.

Therefore the use of NC simulation and verification of the tool path is necessary if NC programs are to be run with assurance during automation.

3.2 ADVANTAGES OF MACHINING SIMULATOR

Prevent accident: With the implementation of the machining many accidents due to wrong tool path during machining, wrong tooling selection etc. can be avoided

Fewer Risks: With the use of machining simulators risks of implementing the wrong selection regarding the machining centre, tool or cutter type, cutter diameter can be reduced and thus it can help in reducing the indirect losses that can occur with the actual machining and replacement.

Evaluation of the process: A simulator can help engineers to visually examine the machining process without the need of doing the actual machining. With the visual inspection a lot of errors can be removed and better decisions can be taken.

Better Decision making :As with the implementation of machining simulators, better visualization of the process is possible than its imagination, so better decisions regarding the selection of the tooling, machining centres etc. can be done.

Compare different scenarios to consider all possible angles: With the use of machining simulators all the different alternatives e.g. cutter type in milling, cutter diameter, and different machining parameters can be compared against achieving the required objectives. Thus it helps in achieving optimization.

Test ideas for improvement in a completely risk free environment: With the use of machining simulator all new ideas can be tested without the requirement of implementing it in the actual machining environment and thus it avoids all the risks involved in losses of purchasing the tooling and then to do the trials on the machine.

Product cycle time reduced: With the implementation of the machining simulators overall product cycle of bringing product into the actual machining environment reduces to a great extent because the testing stage is completed by the simulator only without requiring the actual machining of the component and thus speed up the process of implementing the product into the regular production schedule.

Lower Costs : Machining simulators also contribute to lower the cost because losses going to occur due to the wrong tooling purchase; trials over the work piece is totally eliminated and as the product life cycle is shortened it further lowers the cost.

3.3 PARAMETRIC CAD SOFTWARE

Most of the design industries use commercial CAD packages to help them to create the real life situation in the virtual environment. These CAD packages are used to create, modify, analyse and optimize the design. To perform these tasks different kinds of design software packages are available in the market. These CAD packages depend upon the variety of product modelling method such as solid modelling, surface modelling, parametric modelling and knowledge based expert system tools. Some of most commonly used commercial software packages used for the creation and modification of customized products available in market are as listed below.

1. AutoCAD
2. Pro-E
3. CATIA
4. SolidWorks
5. Unigraphics

3.4 SOLIDWORKS API

Most of the commercial CAD packages use API which helps in customization and in extending the design and modelling of 3D models. API stands for Application Programming Interface. Out of the listed CAD packages, SolidWorks API provides an easy access to the internal functions of the package which helps in the creation of the model. API helps in the process of accessing data and control SolidWorks. API allows its developers to build applications which can access SolidWorks and its data. SolidWorks API is standalone application which can be executable by Visual Basic.Net, Visual Basic, Visual C++, Visual C#.

The API works through Visual Basic for Application (VBA) environment. This environment is used for the editing of the internal functions of SolidWorks. The solid model and the

modelling data can be easily accessed, edited and can be modified with the help of VBA environment. The user friendly VBA environment of SolidWorks can be used for writing the macro. The macro generated is modified and is used for automating the 3D design procedure. Thus in the present work macro has been developed in the API environment using VBA editor for the customizing 3D models.

There are functions which are used by the CAD package to perform various operations which are accessed by the API with the help of various programming languages. These functions can directly access the functions of SolidWorks.

3.4.1 Benefit of API

1. It can automate common and repetitive tasks so as to increase productivity, save time and standardize procedures.
2. It can retrieve data or export data that already exist in SolidWorks.
3. It can develop useful tools to meet specific needs.

3.5 MACRO IN SOLIDWORKS

API uses application of VBA environment to develop algorithm into SolidWorks modeller. API allows the use of predefined geometrical entities and operations to form a new algorithm. API can control geometric entities, mathematical entities and user defined entities. These three objects are interlinked with SolidWorks database.

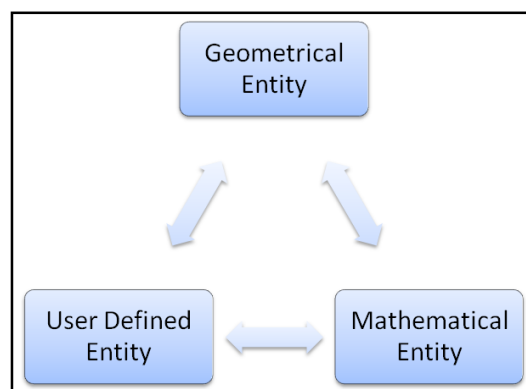


Figure 3.1: SolidWorks API

The geometric entities coming from the SolidWorks database contains the data related to sketch entities and the constraints which makes the sketch fully defined. The basic sketch entities are lines, points, circles, arcs, etc. geometric entities also contain the modelling features like extrude, revolve, sweep, loft etc. If parts are assembled or part is inserted in the modeller then mate and move features are also used. The mathematical entities have concern with the transformations like translation and rotation in the modeller. The mathematical

entities perform faster calculations to project the model from model space to sketch space and vice versa. To project the model basic operations are performed on the vectors of model. The user defined entities are accessed by the API by modifying the geometric entities to perform the functions quickly and repeatedly. All the three objects that is geometric entities, mathematical entities and user defined entities are linked with each other shown in Figure 3.1

Without API we can only use on entity at a time and multiple accessing and looping is not possible. Whereas using API user as access to the basic structure of SolidWorks which helps the user to access the internal entities of SolidWorks and loop them or manipulate them according to his wish. But SolidWorks API does not allow access of all its functions thus limiting the user in some cases [25].

3.5.1 SolidWorks macros

Macros are scripts that help to run operations in the SolidWorks software automatically. A macro can be created and programmed outside the SolidWorks software, or can be recorded in which we can capture a sequence of actions and commands as to perform them in the SolidWorks software. To start programming with SolidWorks API the easy way is to record a macro which can be modified in Microsoft Visual Basic for Applications (VBA) or Microsoft Visual Studio Tools for Application (VSTA). Before recording a macro user must be well aware of the user interface and the environment in which he has to work.

Microsoft VBA record, run and edit macro in SolidWorks. The macro recorded in SW VBA Macros is saved with extension *.swp file format.

Microsoft VSTA is based on VB.net and C#. The macro recorded in SW VSTA VB Macros is saved with extension *.vbproj file format and the macro recorded in SW VSTA C# Macros is saved with extension *.csproj file format.

The basic steps to create API using macro:

1. Plan the user interface in which the user wants to work with.
2. Record the macro step by step as the operations user has to perform.
3. Run the macro to verify the macro that has been recorded by the user.
4. Test the macro whether the output is same or not.
5. Edit the macro; delete the steps which are not required.
6. Debug the edited macro and test the macro again.

3.5.2 General steps for developing macros

While recording a macro each and every operation performed with SolidWorks is recorded in macro. Macro can record the mouse click, operations, features, keystrokes and menu choices.

The macro can be recorded which can be accessed through TOOL menu or through macro toolbar. The macro toolbar contains the shortcuts to record, edit and run macro commands. The macro toolbar is as shown in Figure 3.2.

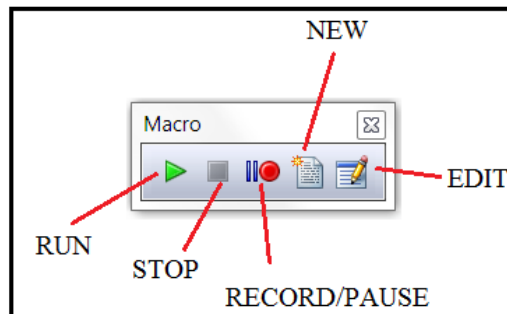


Figure 3.2 Macro Toolbar

1. Record Macro: Operations performed at the user interface are recorded and replayed them with SolidWorks macros.
2. Run Macro: Select Run macro from the Macro toolbar or from the Tools menu. Select a macro that is to be run. Macro will automatically run and perform the task in SolidWorks.
3. Edit Macro: Select a macro that is to be edited. Visual basics editor will launch automatically. Make changes in the macro code debug the code and run it.
4. New Macro: Create a new macro without the use of macro recorder and Visual basic editor will get launched automatically.
5. VBA: It is the tool which runs, records or modifies macros in SolidWorks. And these macros are saved as *.swp format in VBA project files.

The macro starts recording by pressing the record button on the macro toolbar. All the steps are recorded in the macro which is performed in the SolidWorks. By pressing the stop button the recording of macro stops. Save the successfully recorded macro at the desired location. Open the recorded macro by pressing the edit button on toolbar. The macro recorded will open in the VBA environment. The above method can be illustrated with an example as given below:

Example:

Record a macro to make a torus end mill tool with diameter of tool 25.4mm, round of 3mm and length of 70mm. The output file thus generated can save the output at certain location with specific name and then open new blank part in SolidWorks.

Solution:

```
Dim swApp As Object
```

```

Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long
Dim skSegment As Object
Sub main()
Set swApp = Application.SldWorks
Set Part = swApp.ActiveDoc
boolstatus = Part.Extension.SelectByID2("Top", "PLANE", 0, 0, 0, True, 0, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("Right", "PLANE", 0, 0, 0, True, 0, Nothing, 0)
boolstatus = Part.InsertAxis2(True)
boolstatus = Part.Extension.SelectByID2("Top", "PLANE", 0, 0, 0, False, 0, Nothing, 0)
Part.SketchManager.InsertSketch True
Part.ClearSelection2 True
Set skSegment = Part.SketchManager.CreateLine(0#, 0#, 0#, 0.0127, 0#, 0#)
Set skSegment = Part.SketchManager.CreateLine(0.0127, 0#, 0#, 0.0127, -0.07, 0#)
Set skSegment = Part.SketchManager.CreateLine(0.0127, -0.07, 0#, 0#, -0.07, 0#)
Set skSegment = Part.SketchManager.CreateLine(0#, -0.07, 0#, 0#, 0#, 0#)
Part.ClearSelection2 True
boolstatus = Part.Extension.SelectByID2("Line1", "SKETCHSEGMENT", 0#, 0#, 0#, True, 0, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("Line2", "SKETCHSEGMENT", 0#, 0#, 0#, True, 0, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("Fillet<1>", "SKETCHSEGMENT", 0, 0, 0, True, 0, Nothing, 0)
Set skSegment = Part.SketchManager.CreateFillet(0.003, 1)
Part.SketchManager.InsertSketch True
Part.ShowNamedView2 "*Trimetric", 8
boolstatus = Part.Extension.SelectByID2("Axis1", "AXIS", 0, 0, 0, True, 0, Nothing, 0)
Part.ClearSelection2 True
boolstatus = Part.Extension.SelectByID2("Sketch1", "SKETCH", 0, 0, 0, False, 0, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("Axis1", "AXIS", 0, 0, 0, True, 16, Nothing, 0)
Dim myFeature As Object
Set myFeature = Part.FeatureManager.FeatureRevolve2(True, True, False, False, False, False, 0, 0,
6.2831853071796, 0, False, False, 0.01, 0.01, 0, 0, 0, True, True, True)
Part.SelectionManager.EnableContourSelection = False
longstatus = Part.SaveAs3("C:\Users\deep\Desktop\solidworks\to.SLDPRT", 0, 2)
Set Part = Nothing
swApp.CloseDoc ""
Set Part = swApp.NewDocument _
("C:\ProgramFiles\SolidWorksCorp\SolidWorks\lang\english\Tutorial\part.prt", 0, 0, 0)
swApp.ActivateDoc2 "", False, longstatus
Set Part = swApp.ActiveDoc
Dim myModelView As Object
Set myModelView = Part.ActiveView
myModelView.FrameState = swWindowState_e.swWindowStateMaximized
End Sub

```

Output:

All the initial lines declare the variables that are used to run a macro. The main program start after Sub Main(). Then an axis is generated which is formed by the intersection of top and right planes. Then on the top plane four lines are generated to form a rectangle. Then a fillet

of certain radius is generated. Then the revolve command is used. At the end the part formed is saved and new part is opened in SolidWorks. The output is as shown in Figure 3.3.

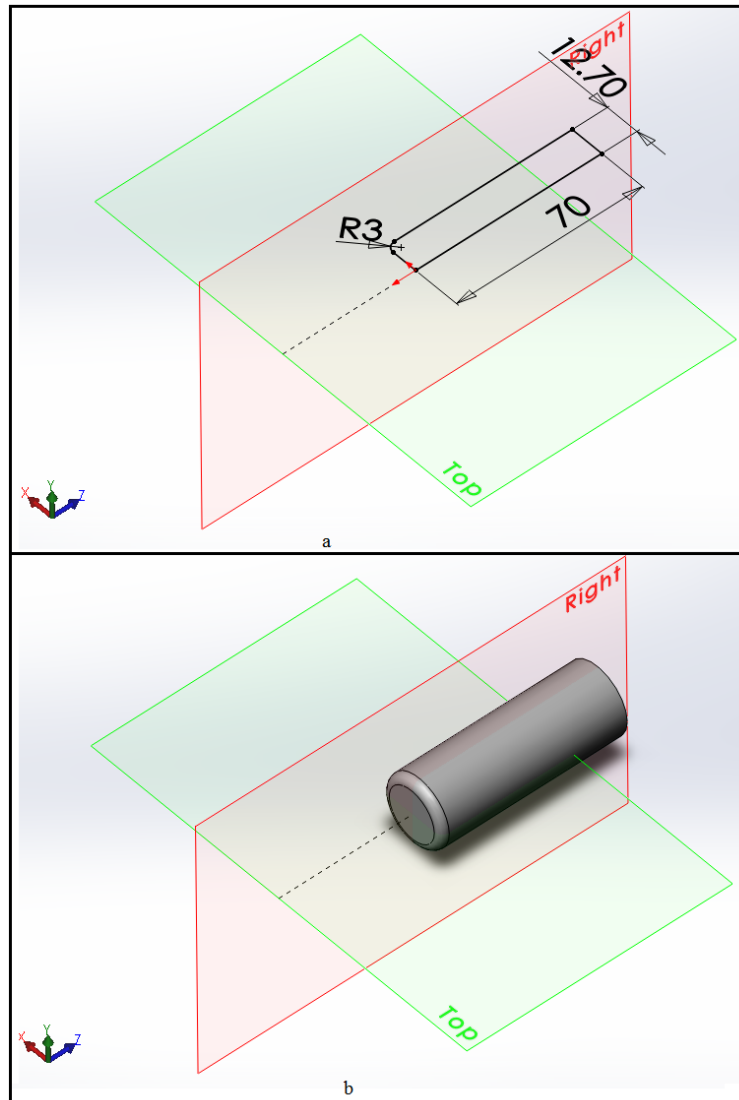


Figure 3.3 Output of macro for torus end mill tool. (a) Sketch formed (b) Revolve command to generate final end product

3.5.3 Record, run, edit and debug the macro

1. Record:

- Click record macro button on the macro toolbar, or go to Menu Bar > Tools > Macro > Record.
- Execute the actions in SolidWorks that you want to record.
- Save the recorded macro with required file name and specific macro type and save.
- While recording click pause button or Tools > Macro > Pause, to pause the macro recording while performing certain operation that one does not want to record.

2. Run:

- a. Click run macro button on the macro toolbar, or go to Menu Bar > Tools > Macro > Run.
- b. A dialog box will open, click the macro you want to run and click open.

3. Edit and Debug:

- a. Click edit macro button on the macro toolbar, or go to Menu Bar > Tools > Macro > Edit.
- b. Select the macro file that you want to edit.
- c. While debugging delete all the lines of code that are there in the code automatically while recording the macro.
- d. Delete the variables not used in the macro.
- e. Delete the lines that are related to the transformations either translation or rotation.
- f. Delete all IModelDocExtension.SelectByID2() appearing immediately before IModelDoc2.ClearSelection2().
- g. Do not delete IModelDocExtension.SelectByID2() appearing immediately after IModelDoc2.ClearSelection2().
- h. Delete all IModelDoc2.ClearSelection2() appearing immediately before IModelDocExtension.SelectByID2().

3.5.4 Visual basic editor environment

1. Projects: The project environment is as shown in Figure 3.4
 - a. SolidWorks objects.
 - b. Definition of SolidWorks objects.

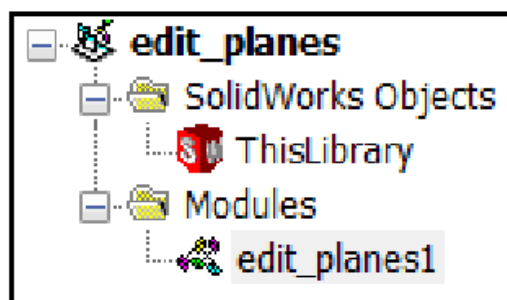


Figure 3.4 Project Viewer

2. Modules: Modules that include code for the macro which is as shown in Figure 3.5

```

' *****
' C:\Users\deep\AppData\Local\Temp\swx1376\Macro1.swb - macro recorded on 11/14/13 by deep
' *****
Dim swApp As Object

Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long, longwarnings As Long

Sub main()
|
Set swApp = Application.SldWorks

Set Part = swApp.ActiveDoc
boolstatus = Part.Extension.SelectByID2("Distance1", "MATE", 0, 0, 0, False, 0, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("D1@Distance1@Part3.SLDPRT", "DIMENSION", 2.76075953465
Dim myDimension As Object
Set myDimension = Part.Parameter("D1@Distance1")
myDimension.SystemValue = 0.05
Part.ClearSelection2 True
boolstatus = Part.Extension.SelectByID2("Distance2", "MATE", 0, 0, 0, False, 0, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("D1@Distance2@Part3.SLDPRT", "DIMENSION", -6.62400763025
Set myDimension = Part.Parameter("D1@Distance2")
myDimension.SystemValue = 0.06
Part.ClearSelection2 True
boolstatus = Part.Extension.SelectByID2("Distance3", "MATE", 0, 0, 0, False, 0, Nothing, 0)
boolstatus = Part.Extension.SelectByID2("D1@Distance3@Part3.SLDPRT", "DIMENSION", 1.245349099153
Set myDimension = Part.Parameter("D1@Distance3")
myDimension.SystemValue = 0.07
Part.ClearSelection2 True
boolstatus = Part.EditRebuild3()
End Sub

```

Figure 3.5 Module Viewer

3.5.5 Variables

1. Local: Values are taken only while in module or procedure. Same variable can be defined locally in multiple modules.
2. Global: Value retained throughout all modules and procedures.
3. Constant: Assign a specific value to a variable name.
4. Variable Types:
 - a. Boolean: True/False.
 - b. Integer: Whole number only.
 - c. Long: Any numerical value.
 - d. String: Any text based value.
 - e. Variant: Can be any type.
 - f. Array(): Spread sheet of data.
 - g. Variant Array(): Spread sheet of data of any type.

3.6 USE OF MICROSOFT EXCEL FOR SOLIDWORKS API

Microsoft excel provides a simple means to create and manage any type of data. Microsoft excel is also accessible by VBA tool to run macro [34]. The VBA tool is automatically installed while installing Microsoft office. This is helpful in writing codes and can be used to operate SolidWorks through excel sheet. Excel is capable of calculations, organizing the data, manipulation of data, arithmetic operations and macro code to execute. VBA allows the user to solve the equations and can also be used as a database.

Microsoft excel helps the user for programming which is called as Application Programming Interface (API). Programmer can write the code in the Visual Basic Editor (VBE)

environment. The VBE environment is capable of writing the code, debugging it. The macro code can perform a particular task repeatedly.

Excel spreadsheet can also be used to import data from excel. The data from currently running excel application can be imported to SolidWorks.

The excel sheet in Figure 3.6 shows the x, y and z position data which can be imported.

	A	B	C
1	0	-0.5	64.75
2	0.521321	1.721482	64.73934
3	0.585258	3.441476	64.65736
4	0.691734	5.158495	64.55407
5	0.840617	6.871058	64.42948
6	1.031728	8.577685	64.28363
7	1.264834	10.27691	64.11653
8	1.539649	11.96726	63.92823
9	1.85584	13.64731	63.71877
10	2.213021	15.31559	63.48818
11	2.610757	16.9707	63.23654
12	3.048564	18.61123	62.9639
13	3.525908	20.23578	62.67033
14	4.042207	21.84299	62.35591
15	4.596833	23.4315	62.0207
16	5.189109	25	61.66481
17	5.818314	26.54717	61.28832
18	6.483683	28.07175	60.89133
19	7.184403	29.57248	60.47396
20	7.919621	31.04815	60.0363
21			

Figure 3.6 Excel sheet showing the x, y and z position co-ordinate.

This can be illustrated by considering an example as given below:

Example:

Import the data from active excel application and plot a point on SolidWorks active part modeller. The x, y and z co-ordinates of point are as shown in excel sheet. The macro should create point until there is an empty cell in the excel sheet.

Solution:

```
Dim swApp As SldWorks.SldWorks
Dim swModel As SldWorks.ModelDoc2
Dim swSketchMgr As SldWorks.SketchManager
Dim swExcel As Excel.Application
Dim exSheet As Excel.Worksheet
Dim skPoint As SketchPoint
Dim i As Integer
Dim xpt As Double
Dim ypt As Double
Dim zpt As Double
Sub main()
Set swApp = Application.SldWorks
Set swModel = swApp.ActiveDoc
Set swExcel = GetObject(, "Excel.Application")
Set exSheet = swExcel.ActiveSheet
Set swModel = swApp.ActiveDoc
Set swSketchMgr = swModel.SketchManager
swSketchMgr.Insert3DSketch (True)
swSketchMgr.AddToDB = True
```

```

i = 1
Do While exSheet.Cells(i, 1).Value <> ""
    xpt = exSheet.Cells(i, 1).Value * 0.001
    ypt = exSheet.Cells(i, 2).Value * 0.001
    zpt = exSheet.Cells(i, 3).Value * 0.001
    Set skPoint = swSketchMgr.CreatePoint(xpt, ypt, zpt)
    i = i + 1
Loop
swSketchMgr.InsertSketch (True)
swSketchMgr.AddToDB = False
End Sub

```

Output:

In the above coding the data in the excel sheet is taken as the x, y and z co-ordinate values. Row 1 gives the 1st point and so on until the empty cell is there in the excel sheet. The column represents the x value of particular point and so on. Initially the variables are declared in the macro to receive data from excel. These variables will help to use the data in macros and perform required tasks in SolidWorks. The macro imports the data from the excel file and create a point in 3D Sketch in SolidWorks. Each of the point has a value that is there in the excel sheet and can be checked by editing the sketch and clicking on particular point and compare the values as are there in the excel sheet. The output of given example is as shown in Figure 3.7.

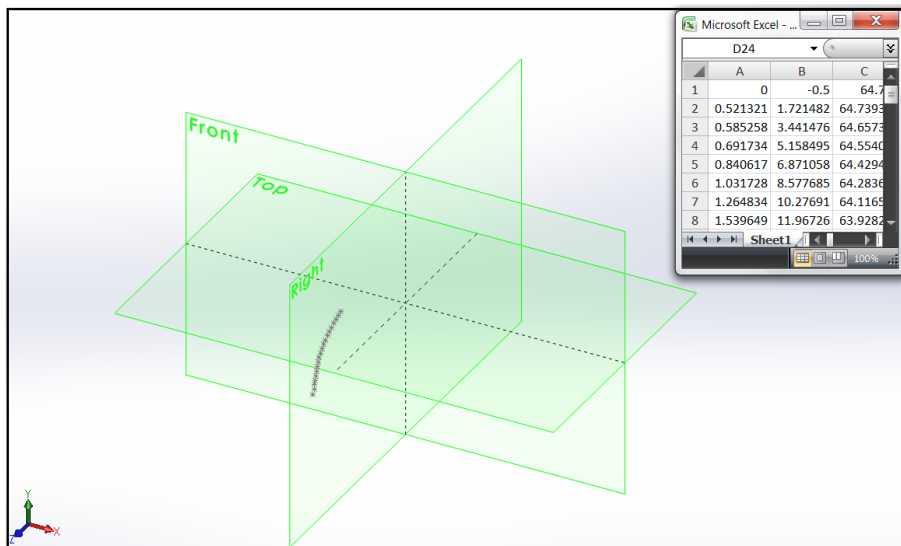


Figure 3.7 Output of macro for importing data from excel to SolidWorks.

CHAPTER 4

METHDOLOGY

A computer program in visual basic for application is created in API SolidWorks and it can be used for the automation to generate the model in SolidWorks. This automation is used for the simulation of tool path in SolidWorks. In this chapter the methodology used for the development of simulator using macros is explained here. The macro developed is to increase the performance, effectiveness and efficiency of the user. Macro is formed with the help of visual basic code and is sequenced to form a code which can represent the output in the modeller of SolidWorks. The macro is modified in such a way such that user is very comfortable to work with SolidWorks. To implement the concept of automation in simulation process the inputs required are discussed in this chapter. Also the user form is discussed to make the work environment easy.

4.1 INPUTS REQUIRED FOR MACROS NC SIMULATOR

The strategies for the developed presented work include the following different components which have been discussed as given below. The inputs required for the macros for NC simulator are

1. Raw stock
2. Tool shape
3. Tool path
4. Type of machining
5. Verification

4.2 RAW STOCK

The raw stock is needed for the simulation process. The orientation of the raw stock in the modeller should be proper. The raw stock generated in the modeller is in positive X and Y direction. The machining process takes place in the negative Z direction. The raw stock generated in the modeller is defined by the user. The parameters required for developing the raw stock are length, width and height.

The parameters used to create the work piece are given in Table 4.1. The work piece is in the shape of cube which will be generated at the origin in positive X, Y and Z direction as shown in Figure 4.1.

Table 4.1: Parameters used to generate work piece

Parameter	Denoted by
Length	L
Width	W
Thickness	T

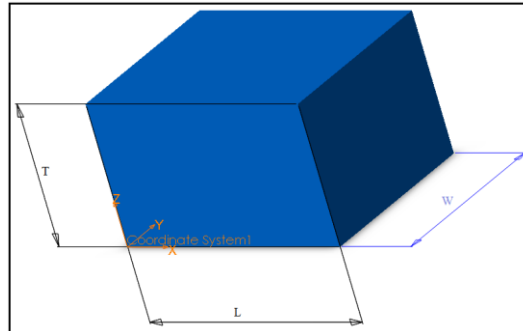


Figure 4.1: Raw stock

The final raw stock generated in the modeller of SolidWorks is saved as *.sldprt file format. This raw stock is oriented and placed at desired position and is then used for the simulation of machining process.

4.2.1 Pseudo code for generating raw stock

```

x1, y1, x2, y2, x3, y3,t = integers      \\ Define variables
Part = swApp.NewDocument                 \\ Open new part
Part = swApp.ActiveDoc
Select "Top", "PLANE"                   \\ Select plane
InsertSketch                             \\ Insert sketch
CreateLine(0, 0, 0, x1, y1, 0)           \\ Shape of raw stock
CreateLine(x1, y1, 0, x2, y2, 0)
CreateLine(x2, y2, 0, x3, y3, 0)
CreateLine(x3, y3, 0, 0, 0, 0)
ExitSketch                               \\ Exit sketch
FeatureExtrusion (t)                    \\ Extrude "t = thickness"
Part.Save                                \\ Save part
CloseDoc ""

```

4.2.2 Flow chart for macro designed for tool shape selection

The procedure explaining the macro designed for raw stock orientation in the form of a flow chart that can help in better understanding. The working and control of the design automation macro using user defined user form is given in Figure 4.2.

The stepwise implementation of SolidWorks macro algorithm for the generation of tools is discussed below:

Step 1: Run the macro.

Step 2: Select the work piece.

Step 3: Enter the parameters required for generating the raw stock.

Step 4: Press the run button. The tool shape generated is saved as *.sldprt format.

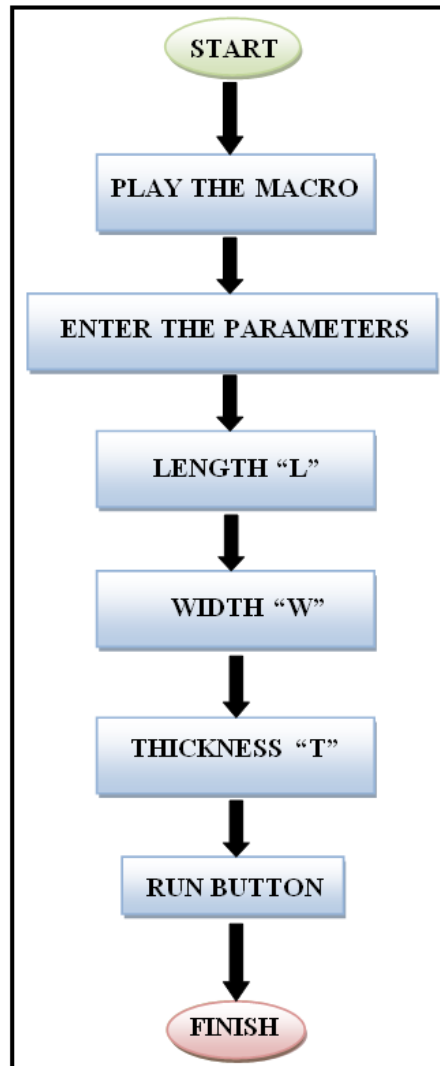


Figure 4.2: Flowchart for generation of raw stock.

4.3 TOOL SHAPE

The cutting tool is modelled as a surface of revolution because the tool rotates around its axis much faster than it travels along the 3-axis/ 5-axis motion [39].

The tools used in our methodology are shown in Figure 4.3. The tools are torus end mill cutter/ bull nose end mill cutter/ radius end mill cutter, flat end mill cutter and ball end mill cutter. In case of torus end mill cutter $R2 > R1$. Whereas in case of ball end mill cutter $R1 \rightarrow R2$ and in case of flat end mill cutter $R2 \gg R1, R1 \approx 0.001 \text{ mm}$. $R2$ is tool radius and $R1$ is fillet radius or radius of pseudo insert circle.

The flat end mill cutter and ball end mill cutter are derivative of torus end mill cutter. If we assume the radius of pseudo insert circle of torus end mill cutter to be very small that is approaching zero then we will get the flat end mill cutter. Similarly if we assume the radius

of pseudo insert circle of torus end mill cutter to be equal to tool radius then we will get the ball end mill cutter. The zoom in sections of the variations of tool is shown in Figure 4.3.

Figure 4.3 (a) shows the torus end mill cutter with tool radius $R2$, fillet radius/ pseudo insert circle radius $R1$ and length L of the tool. Figure 4.3 (b) shows the flat end mill cutter with tool radius $R2$ and length L of tool. The zoom section at the edge shows that the corner of the tool is having the pseudo insert circle with radius approaching to zero and is negligible as compared to $R2$. Figure 4.3 (c) shows the ball end mill cutter with tool radius $R2$ and length L . The zoom section at the tool tip shows that when the radius of pseudo insert circle is assumed to be approaching to tool radius $R1$ then the tip of the tool is flat as shown in the zoom sectioned view.

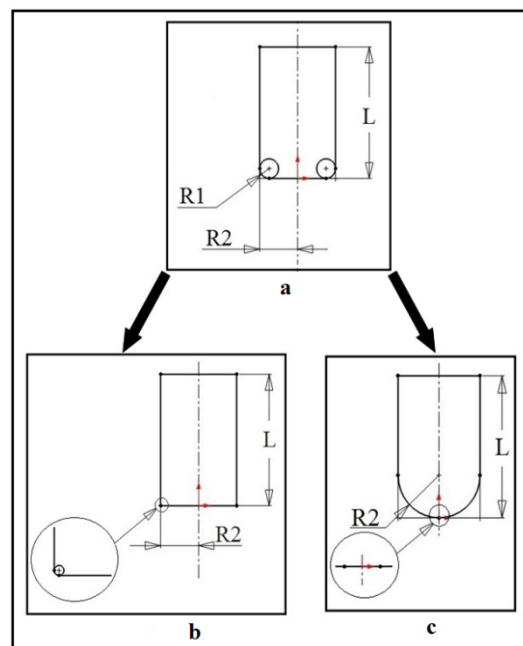


Figure 4.3: Tool Shapes: (a) Torus end mill, (b) Flat end mill and (c) Ball end mill.

The tool shape is sketched as half section in the sketcher of the SolidWorks and is then revolved around the axis. The final outcomes for the various tool shapes generated are shown in Figure 4.4.

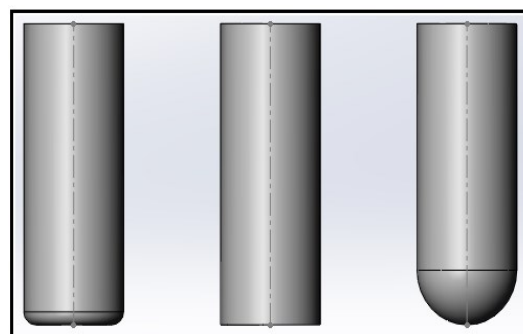


Figure 4.4: Torus end, flat end and ball end mill cutter respectively.

To generate the tool shape the parameters required are the tool radius and tool length for all the three tool shapes. For generating the torus end mill cutter the extra parameter is required to generate the tool shape which is fillet shape. The parameters required for simulation process are as given in Table 4.2.

Table 4.2: Parameters used for creating tool

Parameter	Denoted By
Tool Length	L
Tool Radius	$R2$
Fillet Radius	$R1$

The sketch formed in the sketcher in SolidWorks is shown in Figure 4.5 and the final output generated by revolving the sketch about its axis is shown in Figure 4.6. The user form for generating tool shape is also shown in figure along with the sketch showing the parameters for generating tool shape.

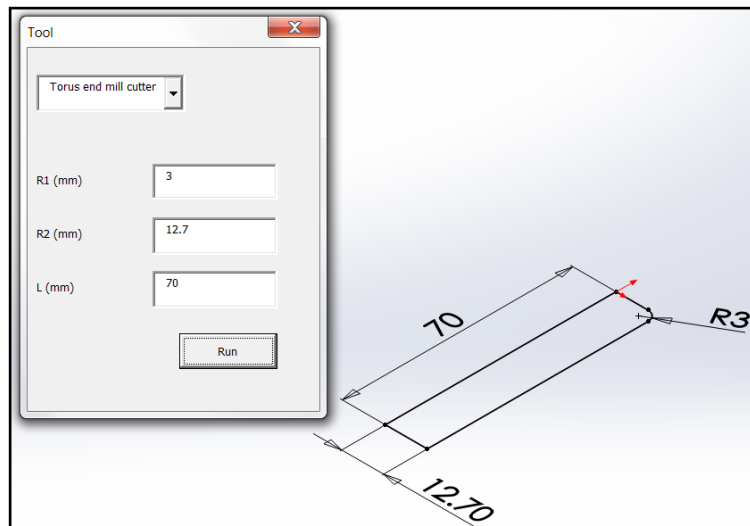


Figure 4.5: Sketch to generate torus end mill cutter using macro.

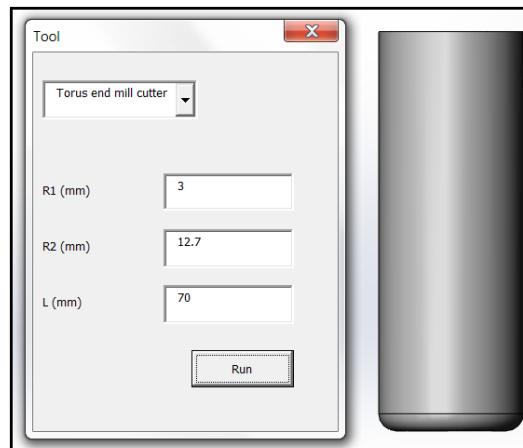


Figure 4.6: Final tool shape of torus end mill cutter.

The final tool shape generated in the modeller of SolidWorks using macros for flat end mill cutter and ball end mill cutter is shown in Figure 4.7 and 4.8 respectively.

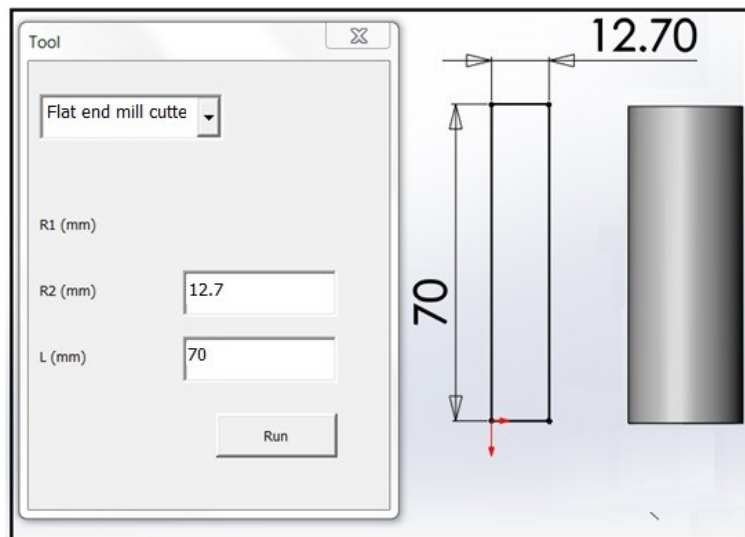


Figure 4.7 Final tool shape of flat end mill cutter.

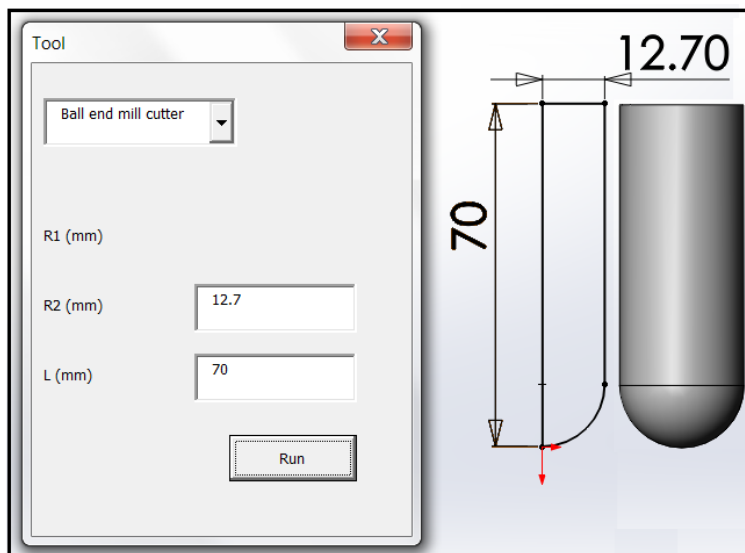


Figure 4.8 Final tool shape of ball end mill cutter.

The final tool shape generated in the modeller of SolidWorks is saved as *.sldprt file format. This tool shape is then used for the simulation of machining process.

4.3.1 Pseudo code for generating tool shape

```

x1, y1, x2, y2, x3, y3, R1 = integers    \\ Define variables
Part = swApp.NewDocument                \\ Open new part
Part = swApp.ActiveDoc
Select "PLANE"                          \\ Select plane
Select "PLANE"                          \\ Select plane
InsertAxis                              \\ Insert Axis
Select "Top", "PLANE"                   \\ Select plane
InsertSketch                             \\ Insert sketch
CreateLine(0, 0, 0, x1, y1, 0)          \\ Shape of tool
CreateLine(x1, y1, 0, x2, y2, 0)       \\ Tool radius R1 and Length L

```

```

Createline(x2, y2, 0, x3, y3, 0)
Createline(x3, y3, 0, 0, 0, 0)
Select Line1, Line2
If "Torus end mill cutter"
Insert Fillet R1                                \\ R1 = radius of pseudo insert circle
ElseIf "Flat end mill cutter" then
Insert Fillet R1 → 0
ElseIf "Ball end mill cutter"
Insert Fillet R1 → R2
ExitSketch
Select "AXIS"
FeatureRevolve (360 * π / 180)
Part.Save                                       \\ Save part
CloseDoc ""

```

4.3.2 Flow chart for macro designed for tool shape selection

The procedure explaining the macro designed for tool shape selection in the form of a flow chart that can help in better understanding. The working and control of the design automation macro using user defined user form is given in Figure 4.9.

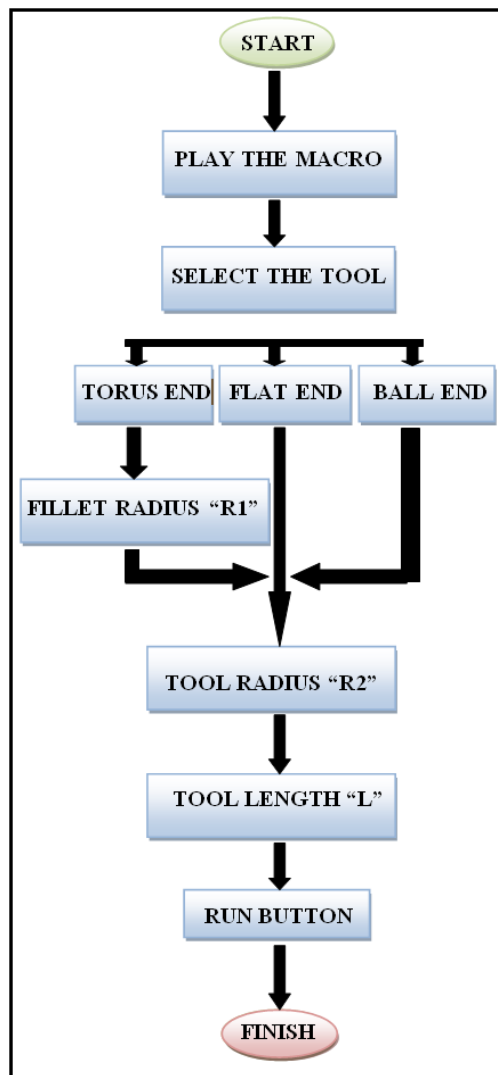


Figure 4.9: Flowchart for the generation of tool shape.

The stepwise implementation of SolidWorks macro algorithm for the generation of tools is discussed below:

Step 1: Run the macro.

Step 2: Select the tool shape.

Step 3: Enter the parameters required for generating the tool shape.

Step 4: Press the run button. The tool shape generated is saved as *.sldprt format.

4.4 TOOL PATH

The data for the relative positioning of the tool and the raw stock is called the tool path data. The tool path data indicates the location of the tool with respect to some reference frame during the machining process. The tool path data is generated using the STL file of the part. The tool positioning points calculated from the STL files are then imported in excel file. This excel file containing the tool path data can then be used by the macro to import these points into the SolidWorks modeller for simulation process.

In case of 3-axis machining, the excel file contains the X, Y and Z co-ordinate values of tool. There are certain limitations of the SolidWorks that we cannot access the cutter location data but we can access the tool centre on the front end of the tool. The cutter location data that is needed in the excel file is as shown in Figure 4.10.

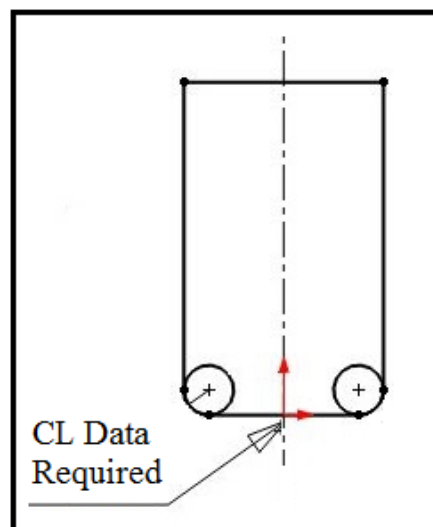


Figure 4.10: CL Data required in Excel File for 3 axis machining.

This CL data are the X, Y and Z co-ordinate values for the particular location of the tool. The path is given in a single direction. If the tool path given is continuous there might be some location where there must be sharp corner. So avoid this corner in the tool path.

The X, Y and Z co-ordinates are in 1st, 2nd and 3rd column of excel sheet. Row 1 will show the 1st tool position with X, Y and Z co-ordinates and so on. The algorithm will run until

there is an empty cell in the excel file. As soon as the empty cell appears in excel file the simulation for certain tool path ends. The co-ordinates of CL points in the excel sheet are in millimetre.

In case of 5-axis machining, along with the tool position on the front end (CL data), the second point is needed which is on the axis of cutter which is taken at the back end of the tool. With the help of these two points it is easy to directly generate the principle axis for placing the tool at particular location. The cutter location data that is needed in the excel file is as shown in Figure 4.11.

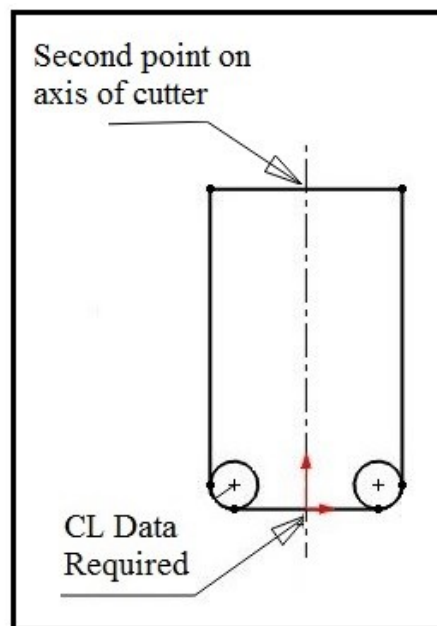


Figure 4.11: Data required in Excel File for 5 axis machining.

This CL data are the X, Y and Z co-ordinate values for the particular location of the tool. The second point on the axis of cutter is also defined by the X, Y and Z co-ordinates in excel file. The tool path for machining is given in the excel file. The CL data having X, Y and Z co-ordinates are in 1st, 2nd and 3rd column of excel sheet and the second point on the axis of cutter having X, Y and Z co-ordinates are in the 4th, 5th and 6th column of excel sheet. Row 1 will show the 1st tool position with X, Y and Z co-ordinates of CL data and X, Y and Z co-ordinates of second point on the axis of cutter and so on. The algorithm will run until there is an empty cell in the excel file. As soon as the empty cell appears in excel file the simulation for certain tool path ends. The co-ordinates of CL points and second point on the axis of cutter in the excel sheet are in millimetre.

4.5 TYPE OF MACHINING

The ability of a CNC machine is to move a part or a tool in different directions simultaneously. Machining simulation is done to reduce the machining time and material

wastage while machining the raw stock physically on the machine. Tool path can be traced and viewed without doing the actual machining. The tool path verification is normally performed during simulation, so that errors can be calculated, based on which the machining process can be rectified [40]. For the simulations of the tool path in SolidWorks using macros the selection of the machining process is necessary. The type of machining has to be selected either 3-axis or 5-axis for the simulation process of the tool path data.

In 3-axis machining, the part moves in three directions that is X, Y and Z. In 3-axis machining the tool or work piece moves about 3 translational axis and there is no rotational axis, therefore tool moves in X, Y and Z direction only. In 5-axis machining centres, the part moves in five directions that is X, Y, Z, A and C. The orientation of the tool is dependent on three translational axis and two rotational axis, therefore tool moves in X, Y and Z direction and two rotary axis A and C. On selecting the type of machining process, the simulation process starts until the tool path ends.

4.5.1 Three-Axis machining: In 3-axis machining simulation process the two methodologies are used. The first methodology is by using the Boolean operation and the second is by using the solid sweep approach.

Boolean approach: The Boolean operation is performed between the tool shape and the work piece. The tool is positioned at the given location according to the CL data. Then the portion of the tool which intersects the work piece is subtracted. This process keeps on repeating for the whole tool path.

Solid sweep approach: The second methodology for the 3-axis machining simulation process for the given tool path is by using the solid sweep operation in SolidWorks. According to this methodology a curve is generated through the given CL data and the tool is allowed to pass through this curve and is subtracted from the work piece. The final output generated from either methodology has been saved in the *.sldprt format. This output can be used to compare the result generated by the tool path and the model from which the tool path was generated.

The solid sweep approach is fast as compared to Boolean approach but at certain locations the solid sweep approach fails because of the sharp corners. So Boolean approach is preferred as compared to other one it doesn't fail.

4.5.2 Five-Axis machining: Similarly in 5-axis machining simulation both the Boolean as well as the solid sweep approach is used for the simulation of the tool path.

Boolean approach: The first methodology is by using the Boolean operation. According to the two points given in the tool path data, the tool positioning axis is generated. The tool is then

positioned and oriented with respect to this tool axis and the Boolean subtraction operation is applied to subtract the portion of the tool intersecting with the raw stock. If the tool path data is in the form of translation points and the orientation angles instead of the two points as explained above, then the tool positioning axis is generated using the given input. This tool positioning axis is generated in the SolidWorks modeller by using reference planes. The generation of axis by using CL point data and angles using reference planes in SolidWorks is little slow process. It takes more time because of the reason that first we have to generate reference planes and then axis, then place the tool on that position and then perform Boolean operation.

Solid sweep approach: The second methodology for 5-axis machining simulation process for given tool path is by using solid sweep operation in SolidWorks. According to this methodology the tool moves from one position to the next position can be assumed to move along a curve. The curve between the two points is considered as a straight line as shown in Figure 4.12. The tool is positioned at the first location and then solid sweep operation up to the next location. The solid sweep thus generated is subtracted from the work piece. The final output generated from either methodology has been saved in the *.sldprt format. This output can be used to compare the result generated by the tool path and the model from which the tool path was generated.

In the 5-axis machining also the Boolean approach is preferred because of the same reason.

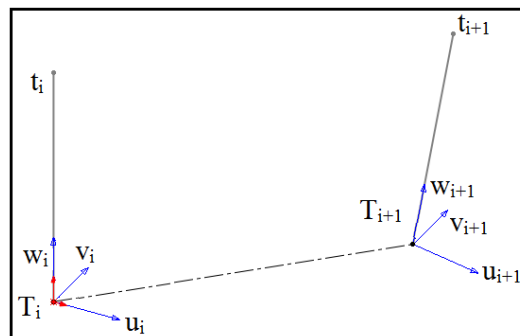


Figure 4.12: Two tool positions.

4.6 VERIFICATION

The most important thing is the verification of the tool path generated from the part and the final output generated by simulation process from this tool path. The visual and analytical verification of the tool path is possible. The final output generated after the simulation process has been saved in the *.sldprt format. This can further be used for the verification of the tool path. The verification methodology is same for the 3-axis and 5-axis machining.

4.6.1 Visual verification: For the visual verification of the simulated part, compare the part formed by the simulation process and the part from which the tool path has been generated. This can be possible by two methods either by using Boolean approach or by changing the colour of two parts.

In the Boolean approach, open a part modeller in SolidWorks. Insert the machined part at the default position in the part modeller from which the tool path data has been generated. Similarly insert the final output generated after the simulation process at the default position. Then perform the Boolean subtraction operation on the simulated and machined part. After the Boolean subtraction operation the volume left can be used to verify the results visually.

The other method is by changing the colour of the machined part and simulated part. In this method for the visual verification change the colour to red for the simulated part and save it. Also change the colour of the machined part to blue. Open the new assembly in SolidWorks. Insert the machined part at the default position in the assembly from which the tool path has been generated. Similarly insert the final output generated after the simulation process at the default position. The output thus can be used for visual verification of the tool path. The dark red/ blue colour will signify that the variation in the tool path. This can be checked by changing the orientation of the final output. If there is dark red colour visible by changing the orientation of the part in assembly, it will signify that there is an overcut. Whereas if there is dark blue colour visible by changing the orientation of the part in assembly, it will signify that there is an undercut.

4.6.2 Analytical verification: The analytical verification of the simulated part is also possible. For the analytical verification of the simulated part is possible by generating the STL file for the simulated part. This is very much similar to the tool path generation method. The tool path of the machined part is generated by using the STL file.

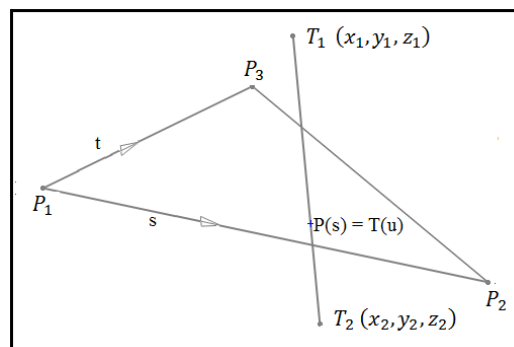


Figure 4.13: Ray tracing method.

The variation in the tool path generated from machined part and the output from the simulated part is compared. The algorithm to find the comparison between the two is as discussed.

1. Equation of line: $T(u) = T_1 + u (T_2 - T_1)$
2. Equation of triangle: $P(s) = P_1 + s(P_2 - P_1) + t(P_3 - P_1)$
3. At the intersection of line and triangular surface, equate the two equations.

$$T_1 + u (T_2 - T_1) - P_1 + s(P_2 - P_1) + t(P_3 - P_1) = 0$$

$$u (T_2 - T_1) + s(P_2 - P_1) + t(P_3 - P_1) = P_1 - T_1$$

$$\text{The above equation can be written as: } u (A) + s(B) + t(C) = \Delta$$

The above equation can be written in matrix form

$$\begin{bmatrix} T_2 - T_1 \\ P_2 - P_1 \\ P_3 - P_1 \end{bmatrix} \begin{bmatrix} u \\ s \\ t \end{bmatrix} = [\Delta]$$

$$\text{Where } T_2 - T_1 = A, P_2 - P_1 = B, P_3 - P_1 = C$$

The above equation can be written as

$$\begin{bmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{bmatrix} \begin{bmatrix} u \\ s \\ t \end{bmatrix} = \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{bmatrix} \quad (1)$$

4. Find u, s and t using equation (1).
5. If $0 \leq u, s, t, s + t \leq 1$ only then the solution is valid.
6. Repeat the above algorithm for all the triangles.
7. The tool path data contains the co-ordinate values of each cutter location for the machined part. Similarly find the co-ordinates for the simulated part. Find the difference in the co-ordinates for each cutter location. If the difference is coming out to be positive then there is an overcut and if the difference is coming out to be negative then there is an undercut. A graph can also be plotted for the difference in the data generated for the machined and simulated part [42].

CHAPTER 5

WORKING OF SIMULATOR MACRO

The simulation of tool path in virtual environment has been developed for reducing machining time and wastage of material due to physical machining. The tool path is generated from STL file from CAD model to be machined; this can be traced and viewed without doing the actual machining. For developing a simulator to simulate the tool path, a CAD environment can be programmed using API for CAD automation. The working of 3-axis and 5-axis simulator is discussed in this chapter.

5.1 WORKING OF THREE AXIS SIMULATOR MACRO

The macro functioning is discussed in the algorithm below. The steps used for the simulation of 3 axis machining for a given tool path is as explained. The user form for the three axis machining simulator is shown in Figure 5.1.

The screenshot shows a software window titled "Machining Simulator". It contains several input fields and buttons. At the top left, there is a dropdown menu set to "3 Axis Simulation" and a text prompt: "Please open Excel File containing Tool Path". To the right, under the heading "Workpiece", there are three input fields: "L" with the value "50", "W" with "30", and "T" with "30". Below these is a button labeled "Workpiece". Under the heading "Tool", there is a dropdown menu set to "Torus end mill cutt...", three input fields: "R1" with "3", "R2" with "12.7", and "L" with "70". Below these is a button labeled "Tool". In the center, there is a diagram showing a 2D cross-section of a workpiece with a tool path, labeled with "R1", "R2", and "L". At the bottom, there is a button labeled "Start Simulation".

Figure 5.1: User form for 3-axis machining simulator.

STEP 1: Open the SolidWorks. The play button is clicked to start the macro. A VB form is popped up on the screen, which enable the user to perform various actions. The form is shown in Figure 5.1.

STEP 2: Select the mode of machining as explained in section 4.4 that is 3 axis simulation. On selecting the 3 axis simulation the form will automatically select the methodology working behind the 3 axis simulation.

STEP 3: Open the excel file containing the tool path for machining. As explained in section 4.4 if solid sweep approach is used then to avoid sharp corners, the “\$” sign is inserted between the path. In the Boolean approach there is no need of the “\$” sign. The algorithm will run until there is an empty cell in the excel file. As soon as the empty cell appears in excel file the simulation for certain tool path ends.

STEP 3: Generate raw stock.

After defining the mode of simulation and opening the excel file containing the tool path, the first frame in the user form is work piece. The form will ask to enter the parameters such as length, width and thickness of the work piece. All the dimensions are in millimetre (mm). Press the work piece button and this will create a geometrical model of the desired parameters as per the values entered by the user. The orientation and placement of the raw stock is explained in section 4.2. The output generated in the geometrical modeller gets saved in *.sldprt format is as shown in Figure 5.2.

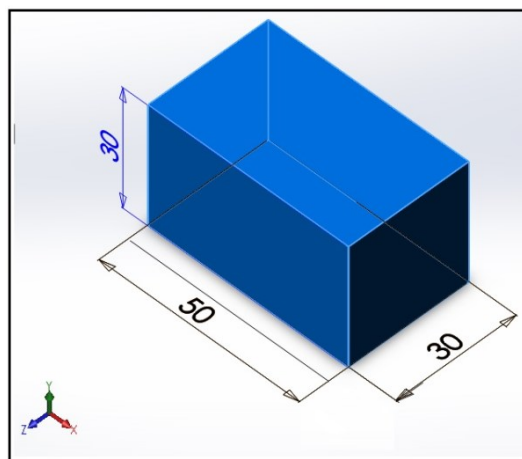


Figure 5.2: Work piece generated in modeller.

STEP 4: Selecting the tool for simulation.

The next frame is selection of tool for simulation. There are three tools used in our methodology as explained in section 4.3. Select the required tool for the simulation process. The tool will ask for the required parameters used to generate the tool for simulation. All the values entered must be in mm. The image of the selected tool is also shown in the user form. The image of the tool shows the various parameters that are required to generate the tool. After pressing the tool button the output is generated in the modeller and the tool gets saved in *.sldprt format which is as shown in Figure 5.3.

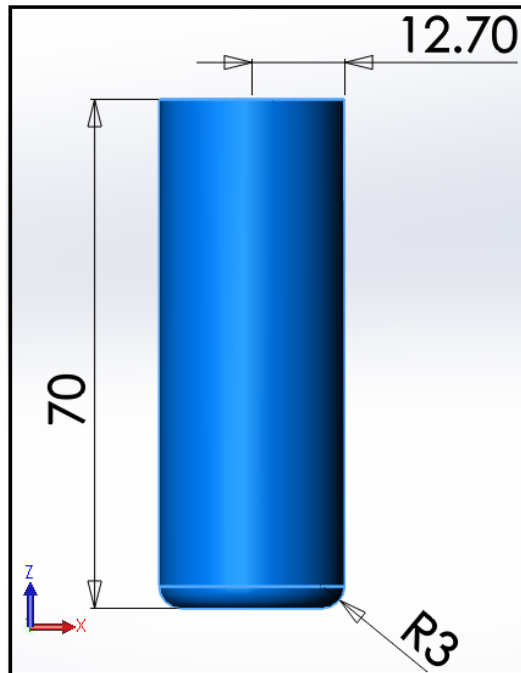


Figure 5.3: Torus end mill cutter generated in modeller.

STEP 5: Click on the Start Simulation Button in User form.

As the desired parameters are entered in the respective text box, the user finally presses the start simulation button. This will invoke the VBA code. The code runs in the background and takes the parameters such as length, width and thickness of the work piece and the tool radius, fillet radius and length of tool. And on the tool path generated from the excel file, the simulation generated is shown in the modeller of SolidWorks. The code will perform different set of steps based upon the inputs.

5.1.1 Pseudo code for 3-axis simulator macro

The pseudo code for the 3-axis simulator macro using solid sweep approach is as discussed.

```

\\ Code for solid sweep approach
i, x1, y1, z1, s = integers           \\ Define variables
Part = swApp.NewDocument              \\ Open new part
Part = swApp.ActiveDoc
Excel.Application                     \\ Open excel sheet
swExcel.ActiveSheet
InsertPart (raw_stock.sldprt)         \\ Insert raw stock
Insert3DSketch                        \\ Make 3D sketch
i = 1
x1 = exSheet (1, 1)                   \\ Import values
y1 = exSheet (1, 2)
z1 = exSheet (1, 3)
CreateLine (x1, y1, z1, x1, y1, z1 + 0.07)
Do While exSheet (i, 1) < > ""
    s = exSheet (i, 1)
    If s = "$" Then

```

```

        x1 = exSheet (i, 1)
        y1 = exSheet (i, 2)
        z1 = exSheet (i, 3)
        Createline (x1, y1, z1, x1, y1, z1 + 0.07)
    End If
    i = i + 1
Loop
Insert3DSketch
CurveFileBegin                \\ Start curve through points
i = 1
Do While exSheet (i, 1) < > ""
    x1 = exSheet (i, 1)
    y1 = exSheet (i, 2)
    z1 = exSheet (i, 3)
    InsertCurveFile (x1, y1, z1)
i = i + 1
Loop
CurveFileEnd
Do While exSheet (i, 1) < > ""
InsertPart (tool.sldprt)      \\ Insert tool shape
AddMate                    \\ Position the tool
Select "AXIS"
Select "LINE"
Coincident                  \\ Coincident constraint between axis and line
Select "PLANE"
Select "POINT"
Coincident
InsertCutSwept (tool, curve) \\ Solid sweep
i = i + 1
Loop
SaveAs (3axis_simulation.sldprt) \\ Save

```

The pseudo code for the 3-axis simulator macro using solid sweep approach is as discussed. The only difference in this methodology is that instead of using solid sweep, use Boolean subtraction operation. The rest of the code is same.

```

\\ Code for Boolean subtraction approach
i, x1, y1, z1 = integers    \\ Define variables
Part = swApp.NewDocument    \\ Open new part
Part = swApp.ActiveDoc
Excel.Application           \\ Open excel sheet
swExcel.ActiveSheet
InsertPart (raw_stock.sldprt) \\ Insert raw stock
Insert3DSketch              \\ Make 3D sketch
Do While exSheet (i, 1) < > ""
    x1 = exSheet (i, 1)
    y1 = exSheet (i, 2)
    z1 = exSheet (i, 3)
    Createline (x1, y1, z1, x1, y1, z1 + 0.07)
i = i + 1

```

```

Loop
i = 1
Do While exSheet (i, 1) < > ""
InsertPart (tool.sldprt)           \\ Insert tool shape
AddMate                             \\ Position the tool
Select "AXIS"
Select "LINE"
Coincident                          \\ Coincident constraint between axis and line
Select "PLANE"
Select "POINT"
Coincident
If i = 1 Then
Select "raw_stock"
Else
Select "CombineName"
End If
Select "tool"
InsertCombine                       \\ Boolean operation
i = i + 1
Loop
SaveAs (3axis_simulation.sldprt) \\ Save

```

5.1.2 Cutter motion in space

A torus end mill cutter is shown in Figure 5.4 as it moves during a 3-axis NC machining simulation. The tool is modelled as a surface of revolution because the tool rotates around its axis much faster than it travels along the 3-axis motion. This is a assumption in any NC machining simulation. The tip of the tool moves along a given cutter location (CL) path as shown in Figure 5.5.

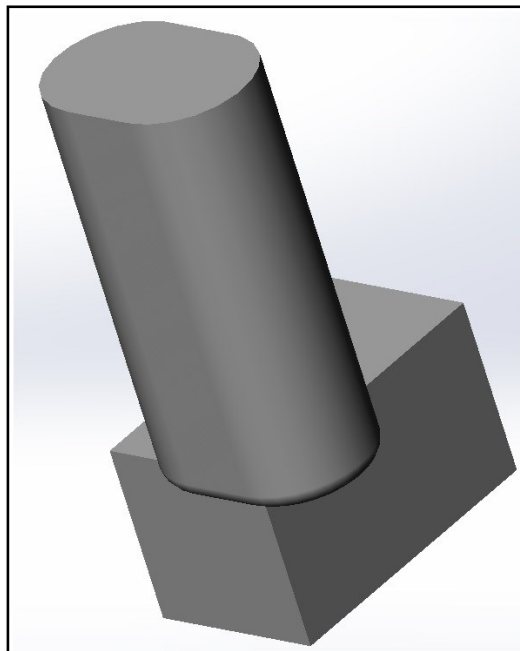


Figure 5.4: Torus end mill cutter moving through work piece.

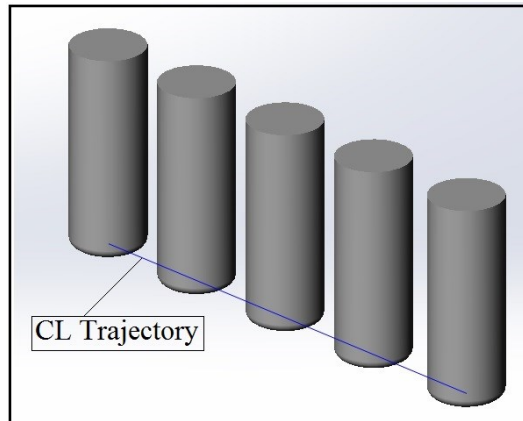


Figure 5.5: Torus end mill cutter moving through space along CL trajectory.

The volume swept by the cutter along the work piece to perform the simulation along the given CL trajectory for given tool path is as shown in Figure 5.6.

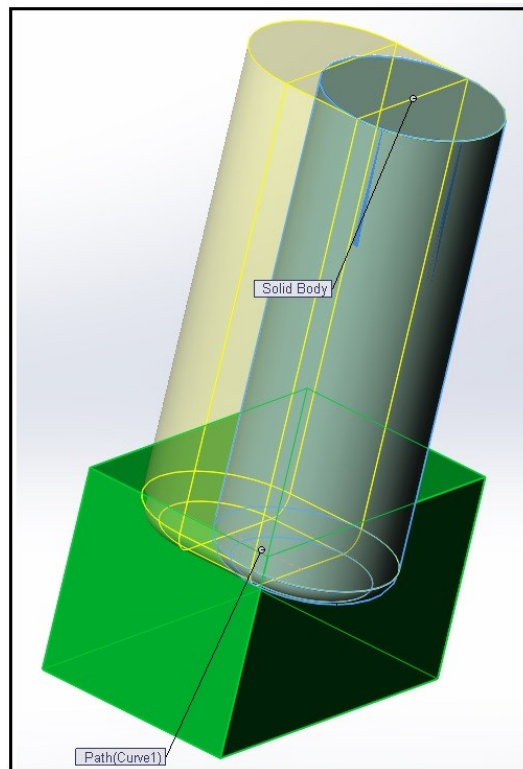


Figure 5.6: Volume swept by the cutter along curve.

5.1.3 Output

The tool path is given in Annexure 1. The final output for the given tool path for torus end mill cutter, flat end mill cutter and ball end mill cutter are shown in Figure 5.7, 5.8 and 5.9 respectively using solid sweep method. The final output for the given tool path for torus end mill cutter using Boolean approach is as shown in Figure 5.10.

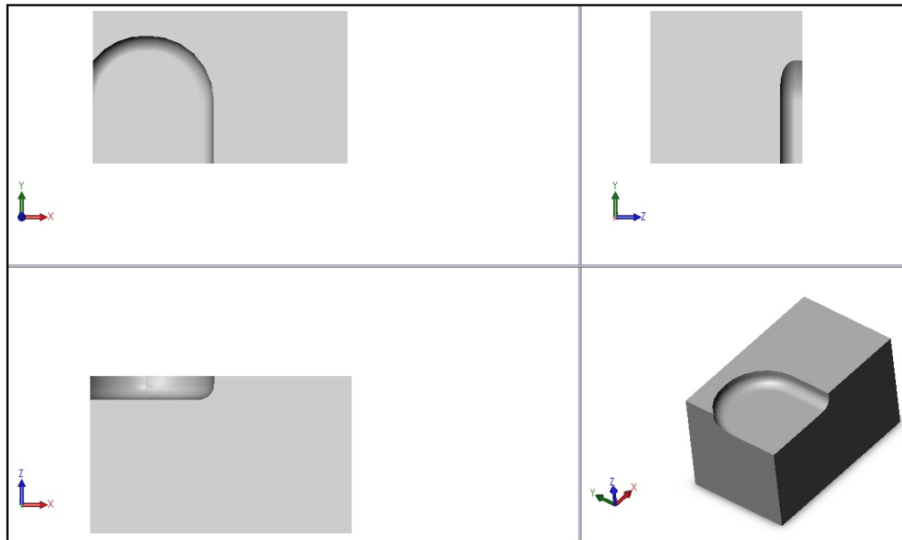


Figure 5.7: Output for tool path in Annexure 1 for torus end mill cutter.

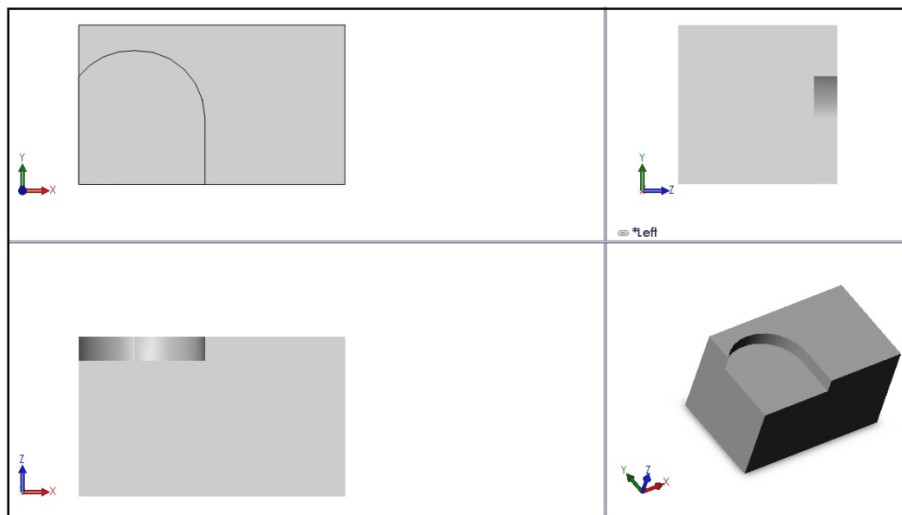


Figure 5.8: Output for tool path in Annexure 1 for flat end mill cutter.

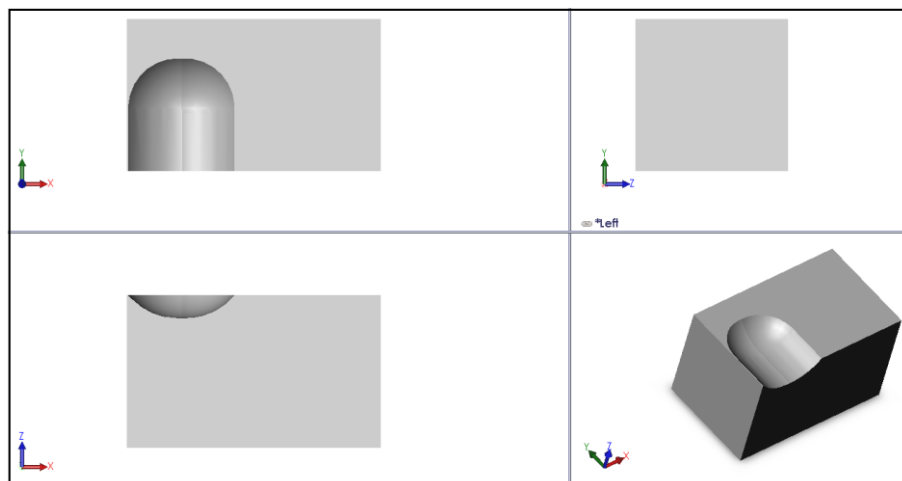


Figure 5.9: Output for tool path in Annexure 1 for ball end mill cutter.

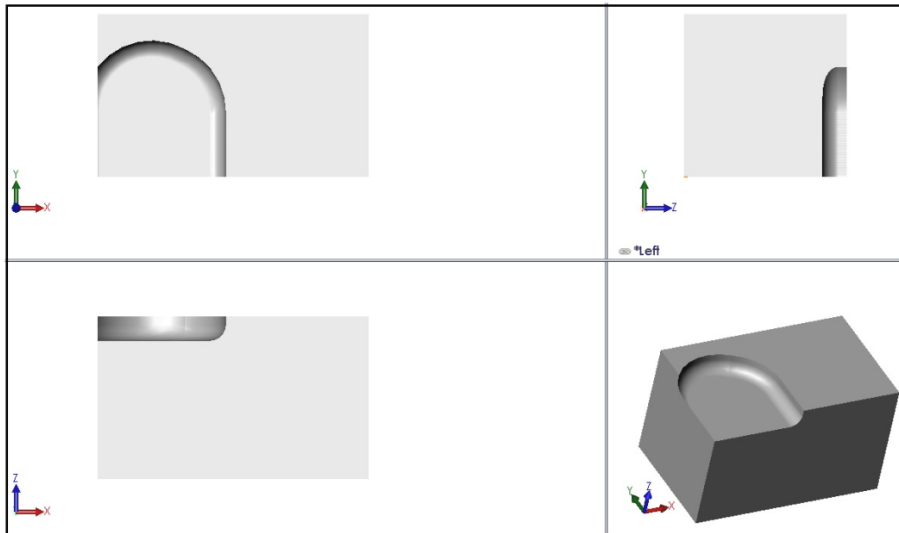


Figure 5.10: Output for tool path in Annexure 1 by Boolean operation.

5.1.4 Flow chart showing the overall organisation of design macros for 3-axis simulation

The procedure discussed above for 3-axis simulation can also be shown as in the form of a flow chart as shown in Figure 5.11.

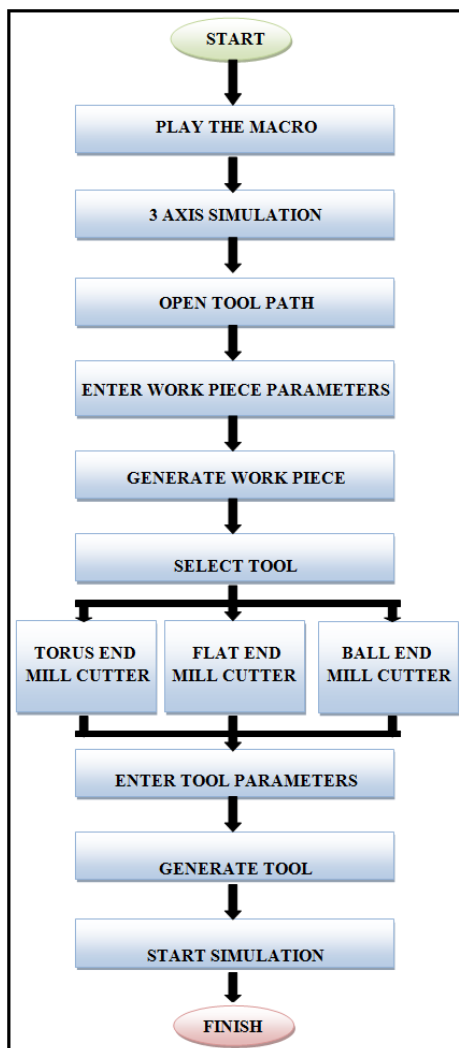


Figure 5.11: Flowchart explaining the flow of control of macro.

5.1.5 Limitations

The above discussed methodologies fail in the case if there is no material to cut. Also the above methodology fails when the two ends do not merge with each other. Also the path for tool movement should be in one direction only. It does not work for sharp corners in tool path. So to overcome such problems we have to take care while generating the tool path.

5.2 WORKING OF FIVE AXIS SIMULATOR MACRO

The macro functioning is discussed in the algorithm below. The steps used for the simulation of 5 axis machining for a given tool path is as explained. The user form for the three axis machining simulator is shown in Figure 5.12.

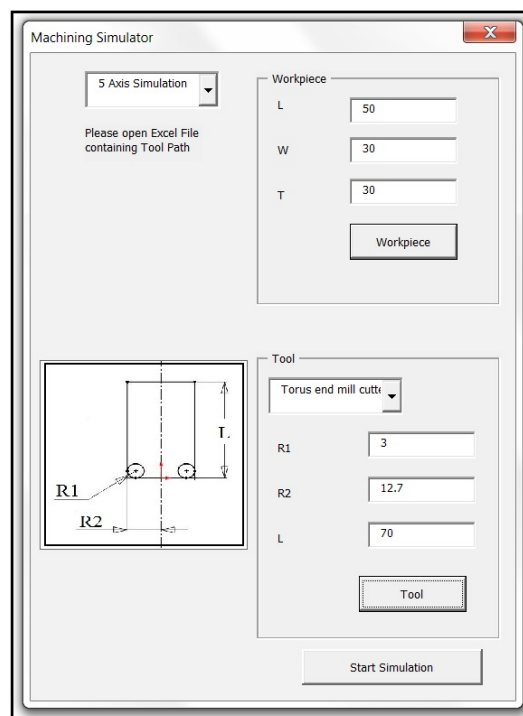


Figure 5.12: User form for 5-axis machining simulator.

STEP 1: Open the SolidWorks. The play button is clicked to start the macro. A VB form is popped up on the screen, which enable the user to perform various actions. The form is shown in Figure 5.12.

STEP 2: Select the mode of machining as explained in section 4.4 that is 5 axis simulation. On selecting the 5 axis simulation the form will automatically select the methodology working behind the 5 axis simulation.

STEP 3: Open the excel file containing the tool path for machining. The algorithm will run until there is an empty cell in the excel file. As soon as the empty cell appears in excel file the simulation for certain tool path ends.

STEP 3: Generate raw stock.

This step is similar as explained in 3-axis simulation.

STEP 4: Selecting the tool for simulation.

This step is also similar as explained in 3-axis simulation.

STEP 5: Click on the Start Simulation Button in User form.

As the desired parameters are entered in the respective text box, the user finally presses the start simulation button. This will invoke the VBA code. The code runs in the background and takes the parameters such as length, width and thickness of the work piece and the tool radius, fillet radius and length of tool. And on the tool path generated from the excel file, the simulation generated is shown in the modeller of SolidWorks. The code will perform different set of steps based upon the inputs.

5.2.1 Pseudo code for 5-axis simulator macro

The pseudo code for the 5-axis simulator macro using Boolean subtraction operation is as discussed.

```
\\ Code for Boolean subtraction approach
i, x1, y1, z1, x2, y2, z2 = integers    \\ Define variables
Part = swApp.NewDocument                \\ Open new part
Part = swApp.ActiveDoc
Excel.Application                       \\ Open excel sheet
swExcel.ActiveSheet
InsertPart (raw_stock.sldprt)           \\ Insert raw stock
Insert3DSketch                           \\ Make 3D sketch
Do While exSheet (i, 1) < > ""
    x1 = exSheet (i, 1)
    y1 = exSheet (i, 2)
    z1 = exSheet (i, 3)
    x2 = exSheet (i, 4)
    y2 = exSheet (i, 5)
    z2 = exSheet (i, 6)
    CreateLine (x1, y1, z1, x2, y2, z2)
    i = i + 1
Loop
i =1
Do While exSheet (i, 1) < > ""
InsertPart (tool.sldprt)                \\ Insert tool shape
AddMate                                 \\ Position the tool
Select "AXIS"
Select "LINE"
Coincident                              \\ Coincident constraint between axis and line
Select "PLANE"
Select "POINT"
Coincident
If i = 1 Then
Select "raw_stock"
```

```

Else
Select "CombineName"
End If
Select "tool"
InsertCombine                \\ Boolean operation
i = i + 1
Loop
SaveAs (5axis_simulation.sldprt) \\ Save

```

The pseudo code for the 5-axis simulator macro using Boolean subtraction operation is as discussed. The only difference in this methodology is that instead of using Boolean operation, use solid sweep. The rest of the code is same.

```

\\ Code for solid sweep approach
i, x1, y1, z1, x2, y2, z2 = integers    \\ Define variables
Part = swApp.NewDocument                \\ Open new part
Part = swApp.ActiveDoc
Excel.Application                       \\ Open excel sheet
swExcel.ActiveSheet
InsertPart (raw_stock.sldprt)           \\ Insert raw stock
Insert3DSketch                          \\ Make 3D sketch
i = 1
Do While exSheet (i, 1) < > ""
    x1 = exSheet (i, 1)                 \\ Import values
    y1 = exSheet (i, 2)
    z1 = exSheet (i, 3)
    x2 = exSheet (i, 4)
    y2 = exSheet (i, 5)
    z2 = exSheet (i, 6)
    CreateLine (x1, y1, z1, x2, y2, z2)
    i = i + 1
Loop
Insert3DSketch
i = 1
Do While exSheet (i + 1, 1) < > ""
    x1 = exSheet (i, 1)
    y1 = exSheet (i, 2)
    z1 = exSheet (i, 3)
    x2 = exSheet (i + 1, 1)
    y2 = exSheet (i + 1, 2)
    z2 = exSheet (i + 1, 3)
    CurveFileBegin                      \\ Start curve through points
    InsertCurveFile (x1, y1, z1)
    InsertCurveFile (x2, y2, z2)
    CurveFileEnd
    InsertPart (tool.sldprt)             \\ Insert tool shape
    AddMate                             \\ Position the tool
    Select "AXIS"
    Select "LINE"
    Coincident                          \\ Coincident constraint between axis and line

```

```

Select "PLANE"
Select "POINT"
Coincident
InsertCutSwept (tool, curve)    \\ Solid sweep
i = i + 1
Loop
SaveAs (5axis_simulation.sldprt)    \\ Save

```

5.2.2 Cutter motion in space

A torus end mill cutter is shown in Figure 5.13 as it moves during a 5-axis NC machining simulation.

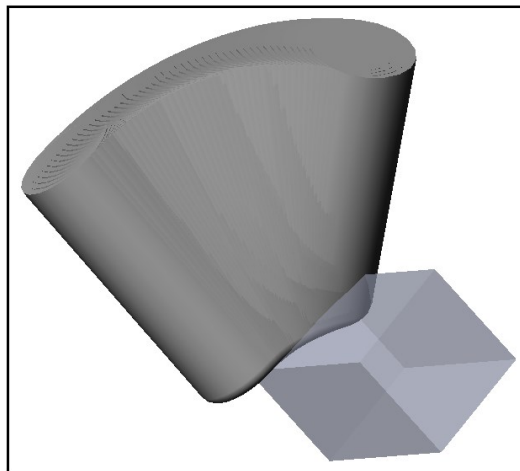


Figure 5.13: Torus end mill cutter moving through work piece.

The surface swept by the tool for same tool path is shown in Figure 5.14. The bottom surface of the tool which is in contact with the work piece and performs the cutting operation is as shown in Figure 5.15.

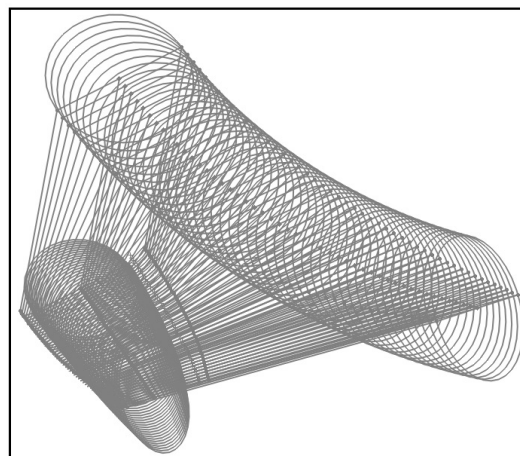


Figure 5.14: Swept volume of tool for tool.

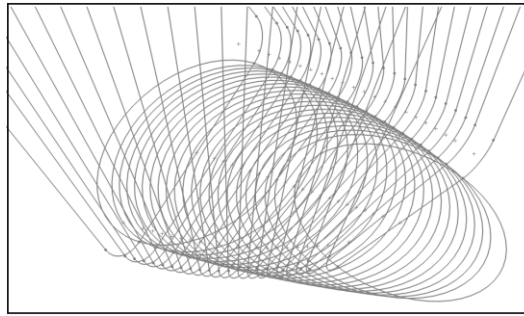


Figure 5.15: Bottom surface of tool.

The tip of the cutter moves along a prescribed cutter location (CL) path and the trajectory of second point on the axis of the cutter as shown in Figure 5.16.

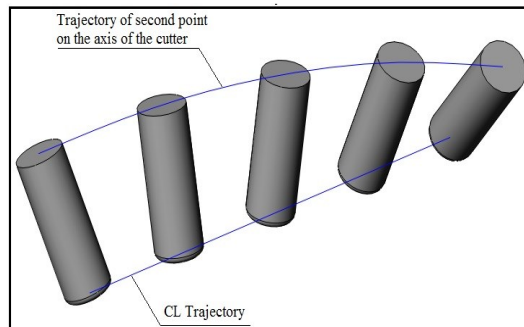


Figure 5.16: Torus end mill cutter moving through space along CL trajectory.

5.2.3 Output

The tool path is given in Annexure 2 for 5-axis machining. The final output for the given tool path for torus end mill cutter, flat end mill cutter and ball end mill cutter are shown in Figure 5.17, 5.18 and 5.19 respectively using Boolean approach. The final output for the given tool path for torus end mill cutter using solid sweep approach is as shown in Figure 5.20.

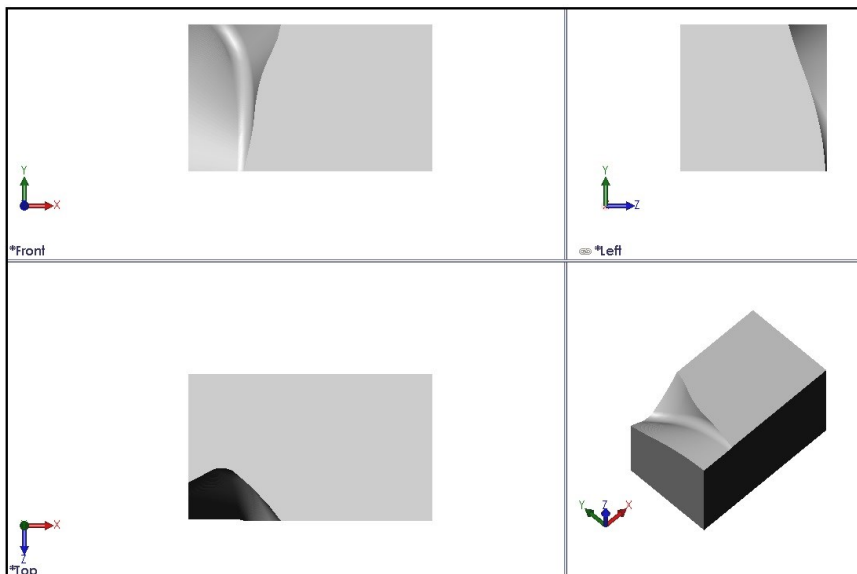


Figure 5.17: Output for tool path in Annexure 2 for torus end mill.

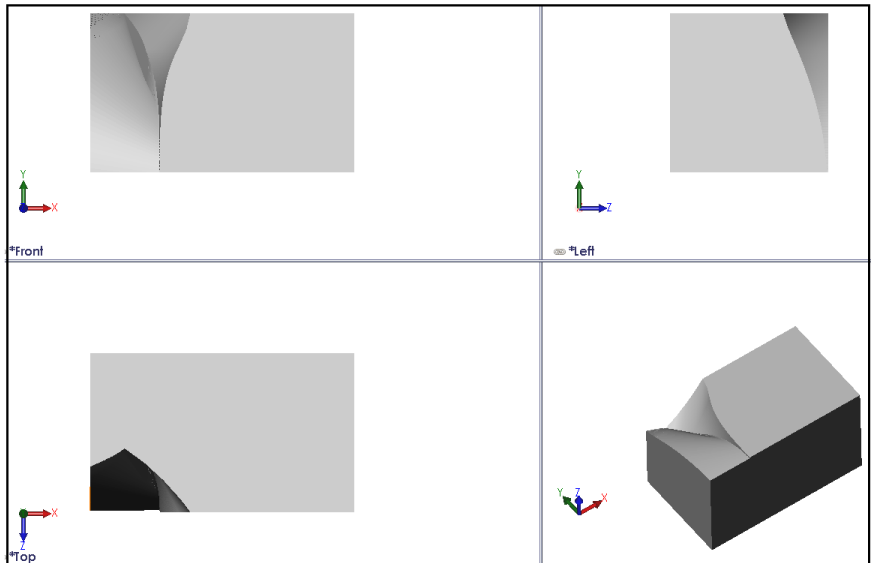


Figure 5.18: Output for tool path in Annexure 2 for flat end mill cutter.

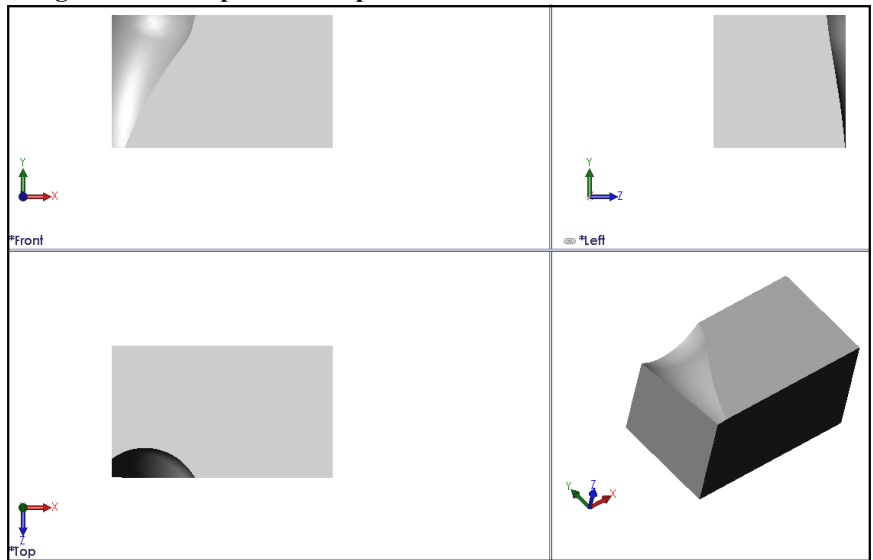


Figure 5.19: Output for tool path in Annexure 2 for ball end mill cutter.

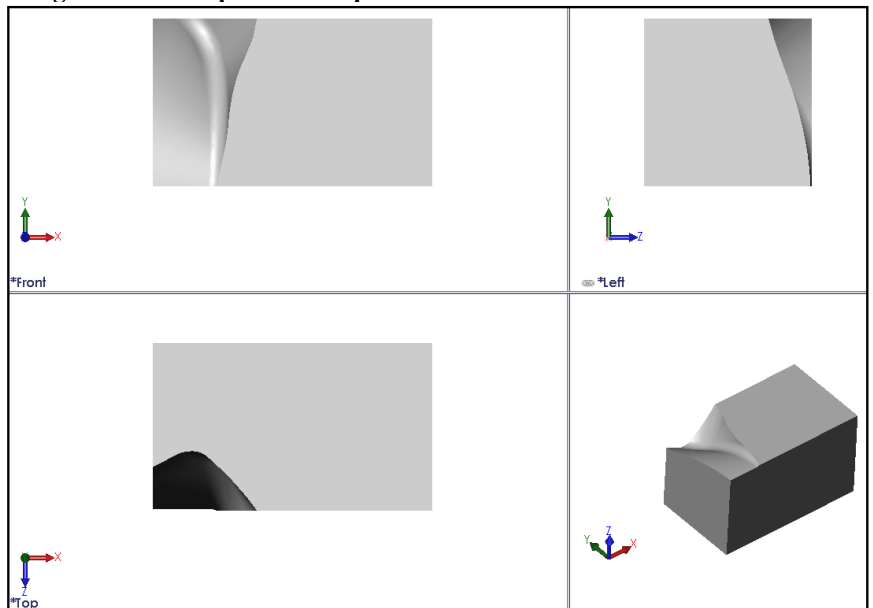


Figure 5.20: Output for tool path in Annexure 2 by solid sweep methodology.

5.2.4 Flow chart showing the overall organisation of design macros for 5-axis simulation

The procedure discussed above for 5-axis simulation can also be shown as in the form of a flow chart as shown in Figure 5.21.

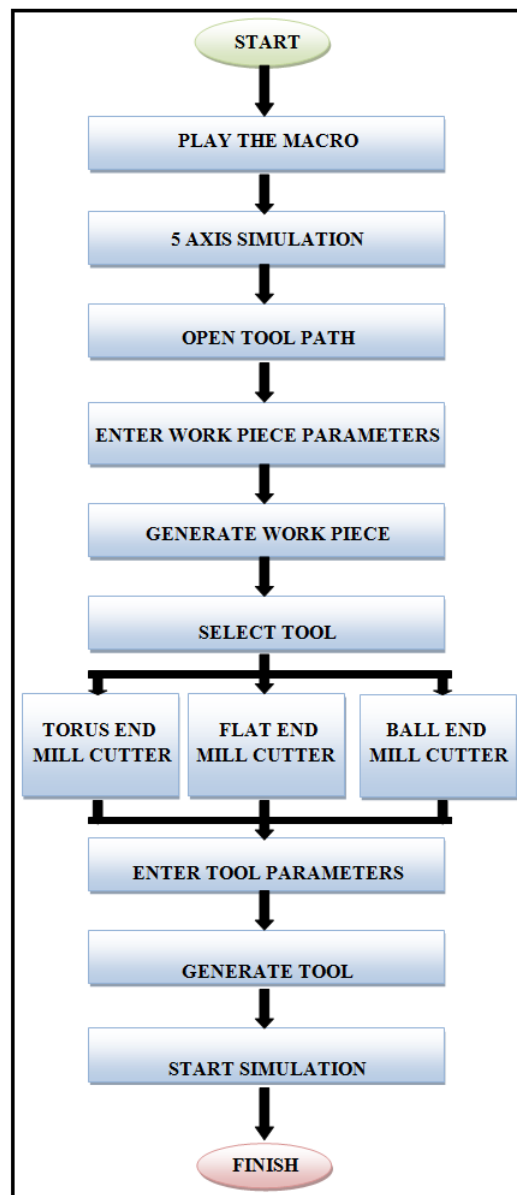


Figure 5.21: Flowchart explaining the flow of control of macro.

5.2.5 Limitations

There is a need of RAM refresh during the simulation process. This can only be possible by closing SolidWorks and then running the macro code from where it has stopped. There are certain free software's like "MACRO RUNNER" which close the solid works after 500 steps and then again start the macro from the same point where it stopped.

CHAPTER 6

RESULTS AND DISCUSSION

The result and validation of the work shown in this chapter is the actual working of the 3-axis and 5-axis machining simulator. The actual tool paths for the two parts are generated from the STL file of the two parts. The tool path is imported into the excel sheet as discussed in section 4.4, which will be the required input for the simulator. The output generated for the given tool path using simulator is saved as *.sldprt. Also generate the *.stl file for the given part. Use the data generated for the visual and analytical verification of the machined part and simulated part.

In this chapter, the working of the whole methodology is presented with the help of following two parts as discussed below. The working of simulator macro for 3-axis and 5-axis as discussed in chapter 5 is used for the two parts.

6.1 INPUT PARAMETERS FOR SIMULATION

The raw stock and tool used for machining simulation is taken as:

Table 6.1: Input parameters for machining simulation

Length of raw stock	25mm
Width of raw stock	50mm
Thickness of raw stock	40mm
Shape of tool	Flat end mill tool
Radius of tool	3mm
Length of tool	10mm

Tool path in excel sheet from STL file.

Part 1: The tool path of given part shown in Figure 6.1 shows the machined part from which the tool path is generated for both 3-axis and 5-axis machining.

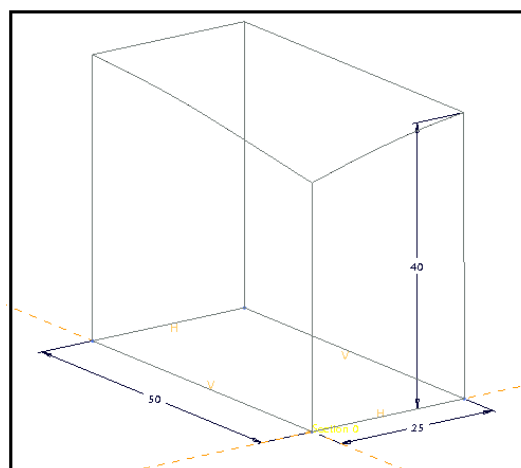


Figure 6.1: Machined part 1.

The STL file in ASCII format for the given part is generated using chord height as 0.05 and angle control as 0.5 shown in Figure 6.2.

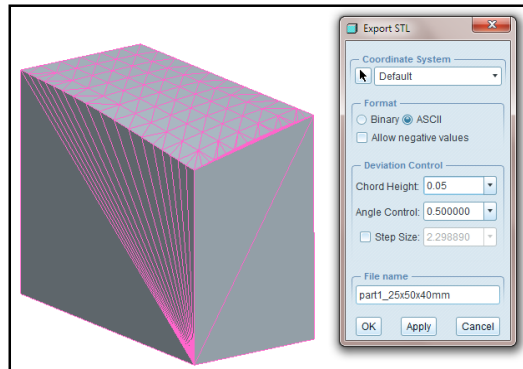


Figure 6.2: Generation of STL file for part 1.

The final STL file generated is shown in Figure 6.3.

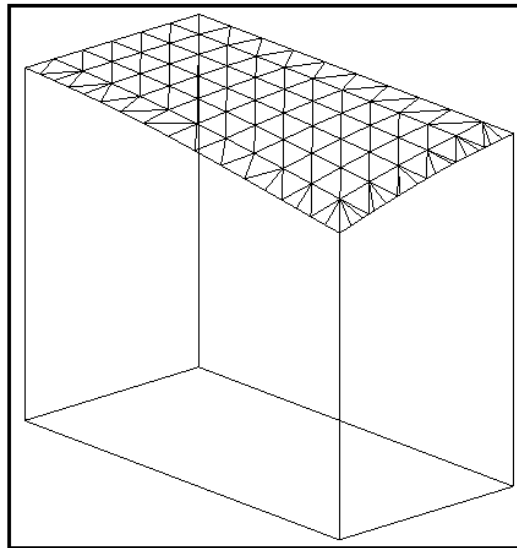


Figure 6.3: STL for part 1.

Part 2: The tool path of given part shown in Figure 6.4 shows the machined part from which the tool path is generated for both 3-axis and 5-axis machining.

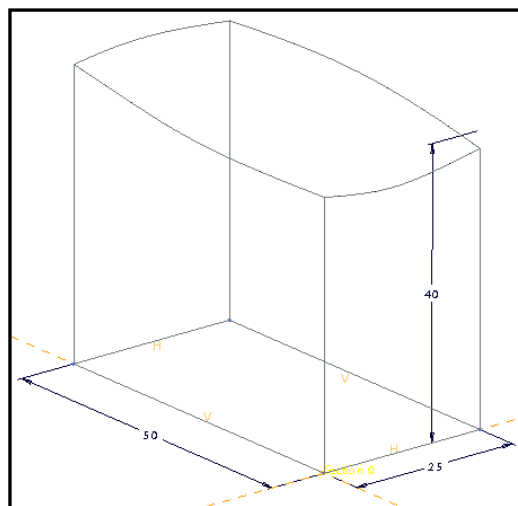


Figure 6.4: Machined part 2.

The STL file in ASCII format for the given part is generated using chord height as 0.05 and angle control as 0.5 shown in Figure 6.5.

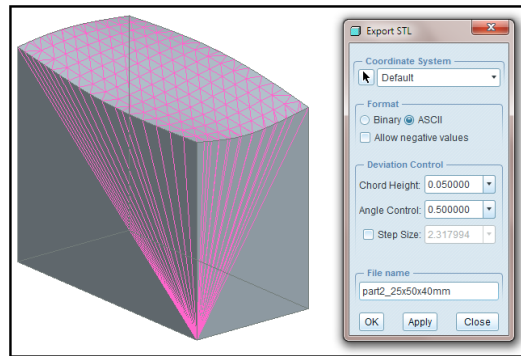


Figure 6.5: Generation of STL file for part 2.

The final STL file generated is shown in Figure 6.6.

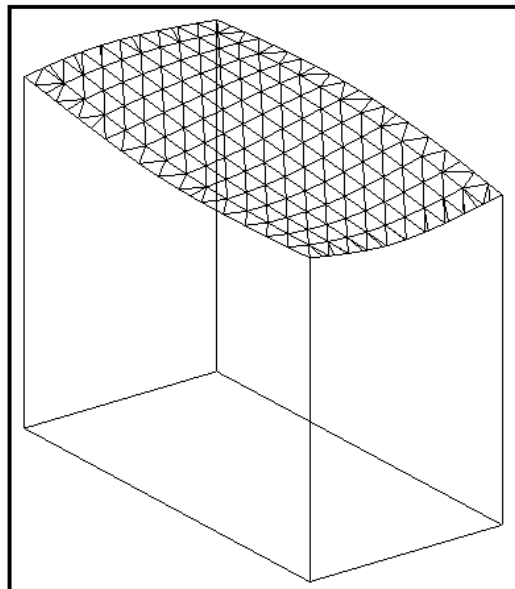


Figure 6.6: STL for part 2.

6.2 VISUAL VERIFICATION

The visual verification of the simulated part is discussed in the section 4.5.1. The output generated for the part 1 and part 2 using the simulator macro. The output generated for the three passes for the given tool path is shown in this section.

The visual verification of part 1 and 2 by the output generated for 3-axis and 5-axis machining is shown below.

6.2.1 Results of 3-axis machining simulation

The visual verification of part 1 and 2 using 3-axis machining data is given below.

Part 1:

The output for the part 1 using 3 axis machining is shown in Figure 6.7.

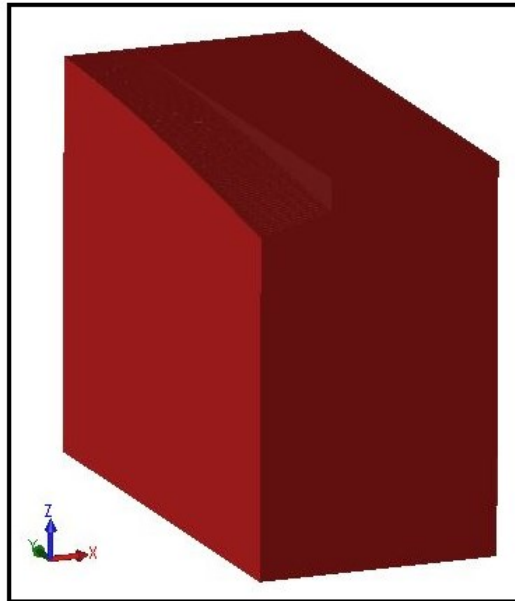


Figure 6.7: Output of part 1 using 3-axis machining simulation.

The visual verification of part 1 is shown in Figure 6.8.

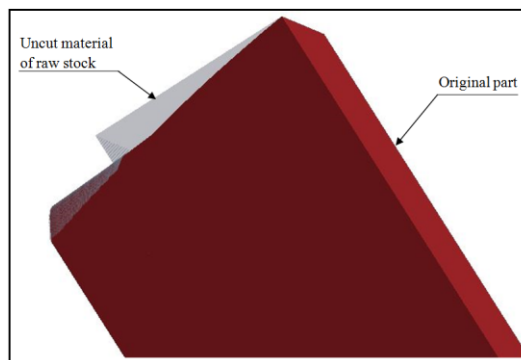


Figure 6.8: Visual verification of part 1 using 3-axis machining simulation.

Part 2:

The output for the part 2 using 3 axis machining is shown in Figure 6.9.

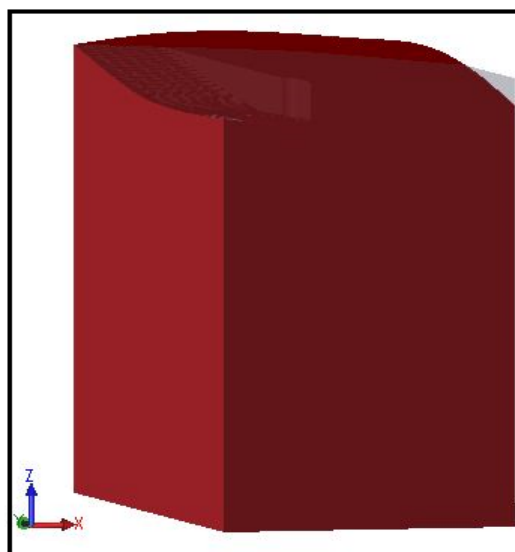


Figure 6.9: Output of part 2 using 3-axis machining simulation.

The visual verification of part 2 is shown in Figure 6.8.

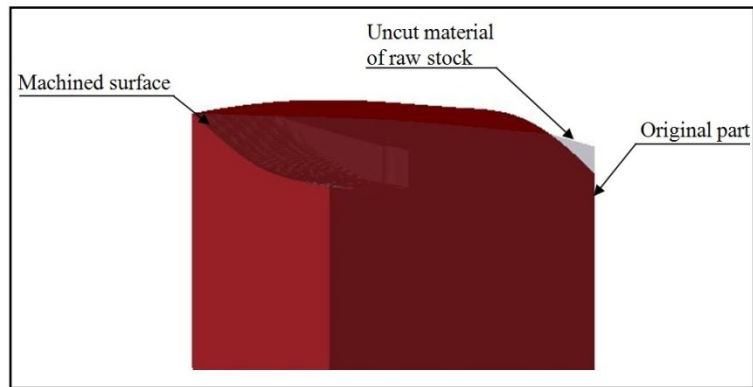


Figure 6.10: Visual verification of part 2 using 3-axis machining simulation.

The red part shown in Figure 6.8 and 6.10 shows the actual machined part from which the tool path is generated. The transparent part shows the uncut material of the raw stock. The machined part from the raw stock and the original part appear to be almost same.

6.2.2 Results of 5-axis machining simulation

The visual verification of part 1 and 2 using 5-axis machining data is given below.

Part 1:

The output for the part 1 using 5 axis machining is shown in Figure 6.11.

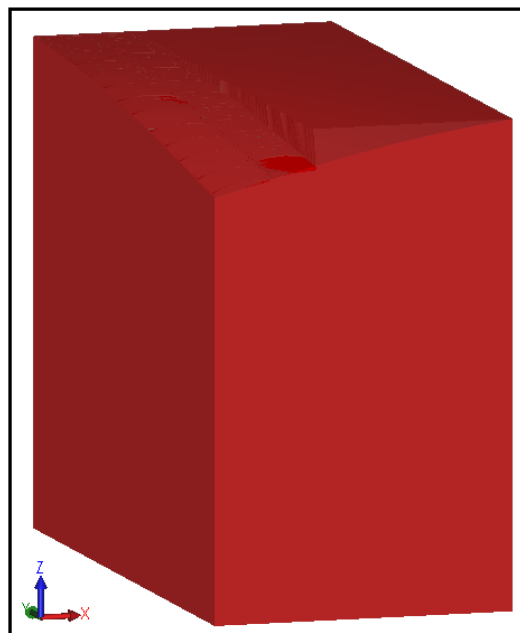


Figure 6.11: Output of part 1 using 5-axis machining simulation.

The visual verification of part 1 is shown in Figure 6.12.

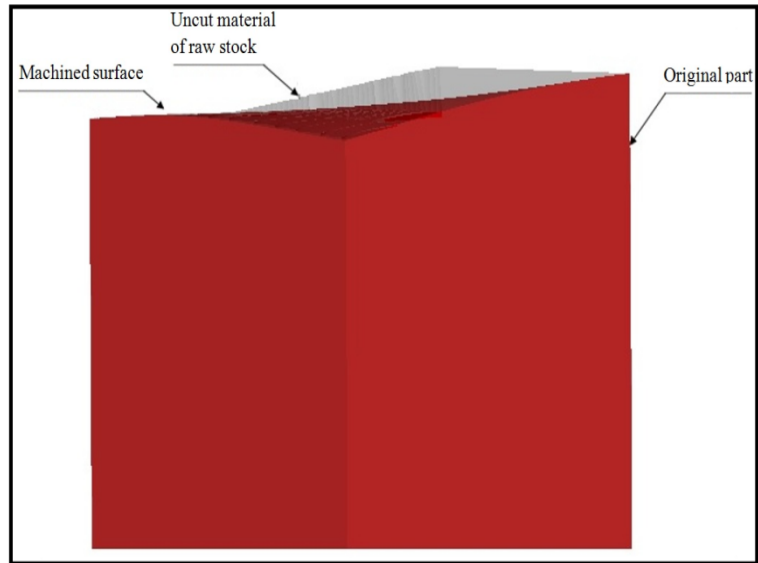


Figure 6.12: Visual verification of part 1 using 5-axis machining simulation.

Part 2:

The output for the part 2 using 5 axis machining is shown in Figure 6.13.

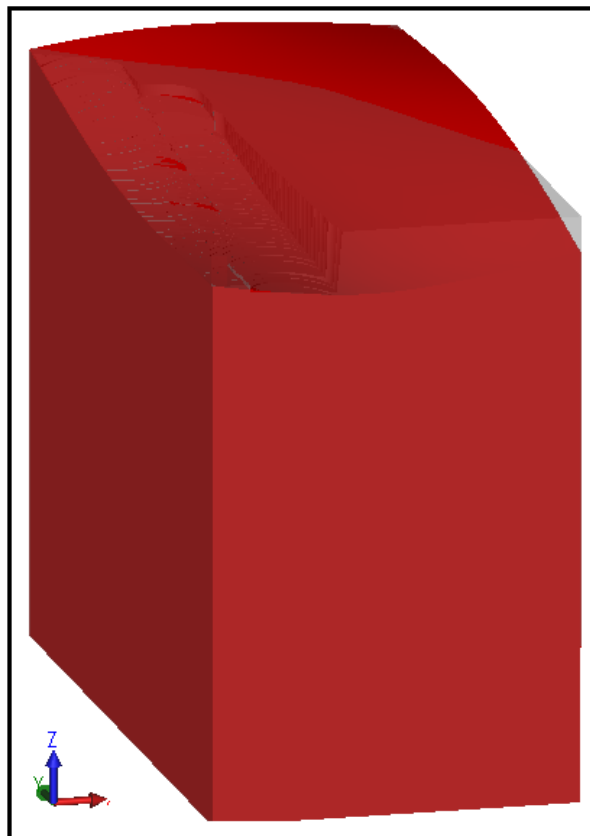


Figure 6.13: Output of part 2 using 5-axis machining simulation.

The visual verification of part 2 is shown in Figure 6.14.

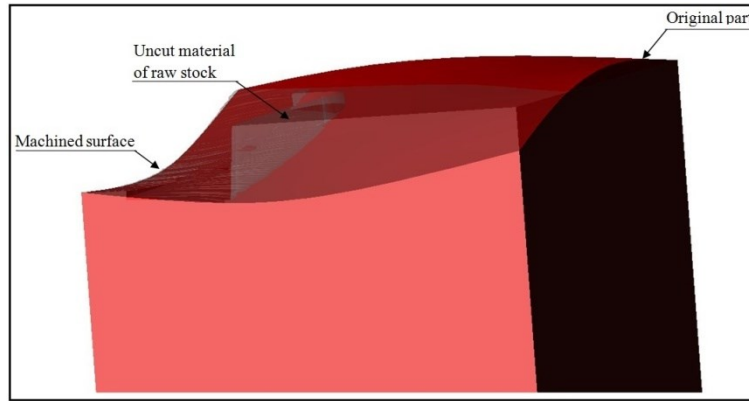


Figure 6.14: Visual verification of part 2 using 5-axis machining simulation.

The red part shown in Figure 6.8 and 6.10 shows the actual machined part from which the tool path is generated. The transparent part shows the uncut material of the raw stock. The machined surface of the original part and the simulated part are almost same.

6.3 ANALYTICAL VERIFICATION

The analytical verification for the two parts using the ray tracing methodology is discussed here. The ray intersection data (x, y and z) are calculated in the feed forward/ tool motion direction with the given increment and in the side step direction with the given increment. The ray is traced from 0 to 4mm in x direction and from 10 to 30mm in y direction. The ray tracing of the part with the given parameters is shown in Figure 6.15. The 3D plots for 3-axis and 5-axis machining are formed which is used to represent the undercut/ overcut of the simulated part. The yellow colour represents the undercut; red colour represents the overcut and green colour within the tolerance region of the simulated part. The tolerance acceptance region for the simulation is taken from -0.1mm to +0.1mm. The 2D plots are formed for the study of scallop height in 3-axis and 5-axis tool paths.

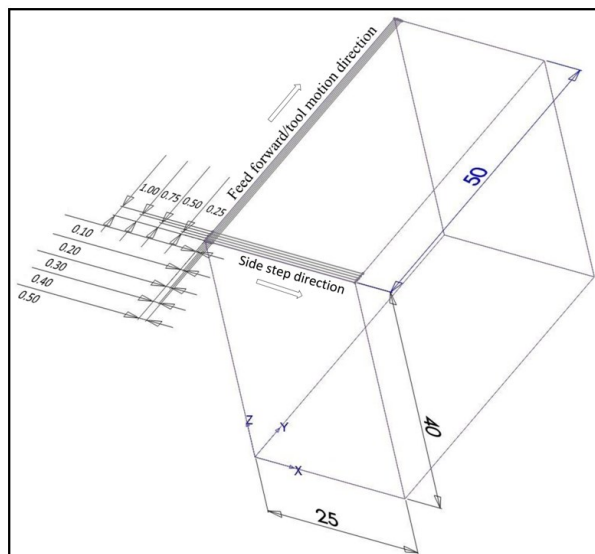


Figure 6.15: Ray tracing of part.

6.3.1 Study of verification for 3-axis and 5-axis tool path

3D plots for part 1 and part 2 for 3-axis and 5-axis machined surface and simulated parts are given below. The ray is traced for the original part and the simulated part with the ray intersection data for $x=0$ to 4mm and $y=10$ to 30mm. The feed forward step increment is taken as 0.1mm and the side step increment is taken as 0.25mm.

Part 1:

Comparison between the original machined part and 3-axis machined simulated part.

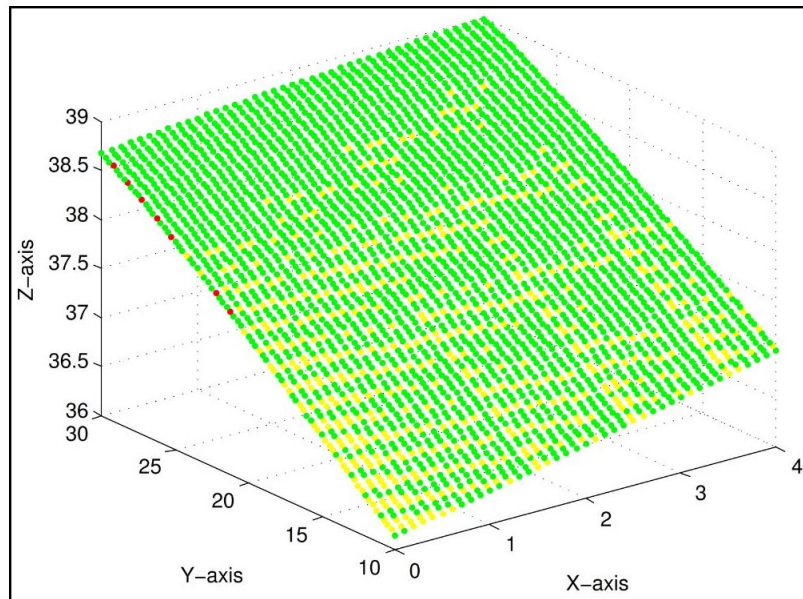


Figure 6.16: Comparison of part 1 original machined part and 3-axis simulated part.

Comparison between the original machined part and 5-axis machined simulated part.

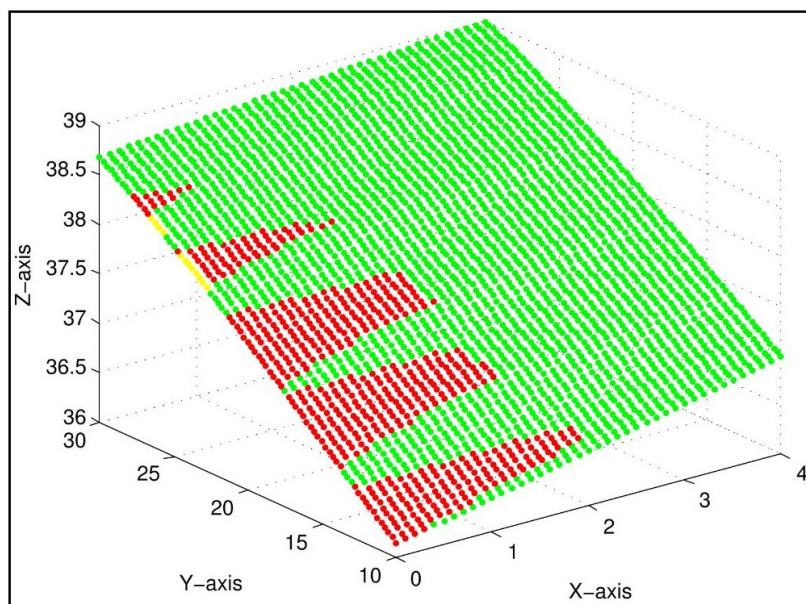


Figure 6.17: Comparison of part 1 original machined part and 5-axis simulated part.

Part 2:

Comparison between the original machined part and 3-axis machined simulated part.

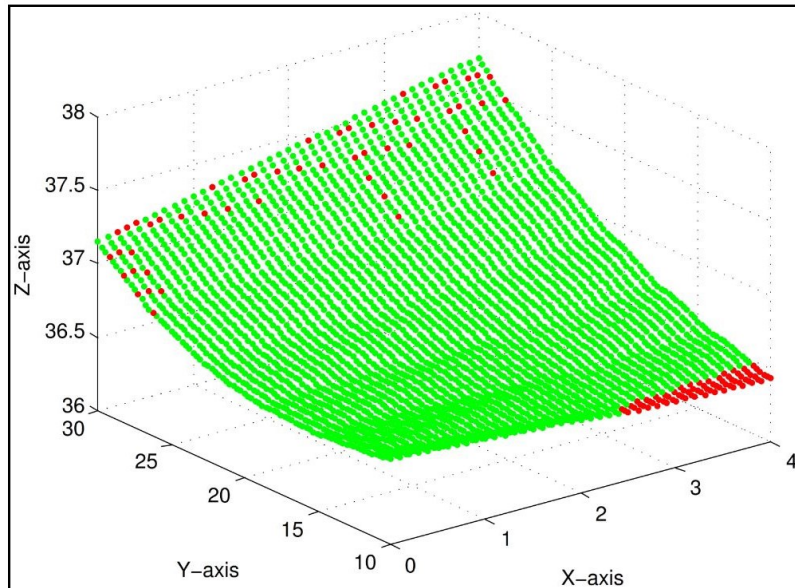


Figure 6.18: Comparison of part 2 original machined part and 3-axis simulated part.

Comparison between the original machined part and 5-axis machined simulated part.

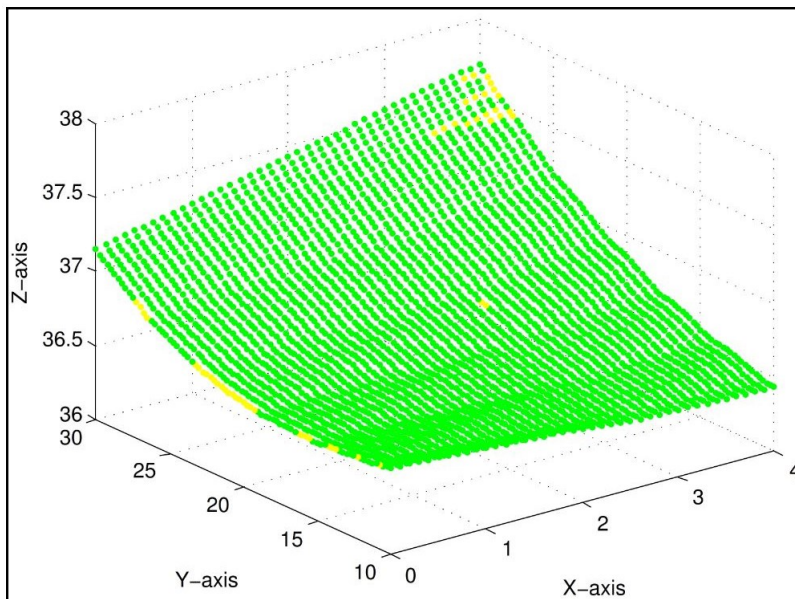


Figure 6.19: Comparison of part 2 original machined part and 5-axis simulated part.

From the above graphs, the yellow colour is for overcut and red colour is for undercut and the green colour is for machining in tolerance region. These variations are because of that the tool path is generated for 1mm feed forward and 1mm side step distance for 3axis machining and 1mm feed forward and 3.5mm side step distance for 5-axis machining. There are some stairs cases which will be generated from one tool position to next tool position whereas in actual machining those steps are not there. In this overcut values are there is primarily

because of Boolean operation and left out stair steps further because we have tried to compare the original STL model with the output of simulated part again taken in STL format with 0.03mm chord height. So there can be some errors found using ray tracing comparison at given x y location for z height of actual part and simulated part. As the chord height for original part is 0.05mm and for the simulated part the chord height is taken as 0.03mm. The errors in the tool positioning with respect to the original part is of the magnitude $\pm 0.05\text{mm}$, they are expected. Similarly in ray tracing of simulated part the errors in the tool positioning is of magnitude $\pm 0.03\text{mm}$ is expected.

6.3.2 Study of scallop height in 3-axis and 5-axis tool path

The 2D plots are formed to study the scallop height. The plots are formed for given y values against the x and $(z_2 - z_1)$, where z_2 is the z values of the ray traced for the simulated part and z_1 for the original part. The ray is traced for the original part and the simulated part with the ray intersection data for $x=0$ to 4mm and $y=10$ to 30mm. The feed forward step increment is taken as 0.1mm and the side step increment is taken as 10mm. The plots for the part 1 and part 2 are given below.

1. Plot between the original part and 3-axis simulated part at $y=10, 20$ and 30mm in Figure 6.20, 6.21 and 6.22 respectively for part 1.

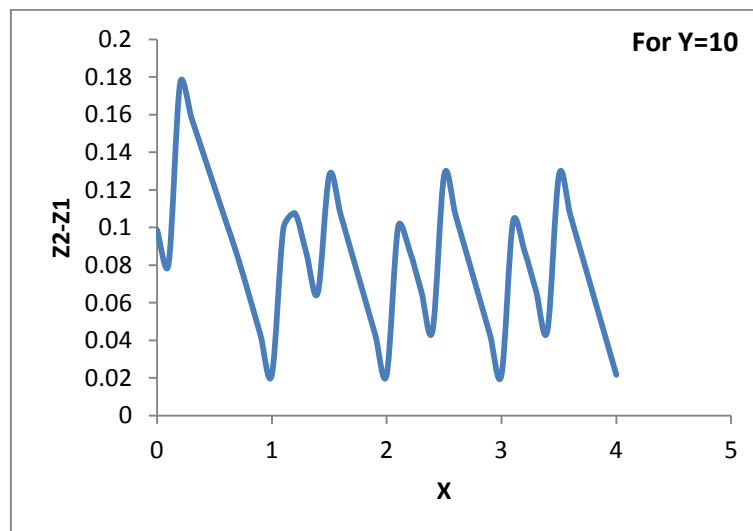


Figure 6.20: Plot for the scallop height of part 1 at $y=10\text{mm}$ for 3-axis simulated part.

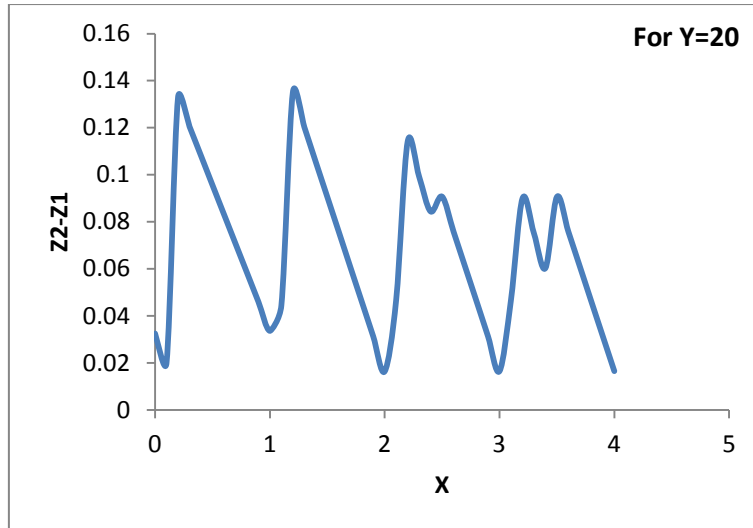


Figure 6.21: Plot for the scallop height of part 1 at y=20mm for 3-axis simulated part.

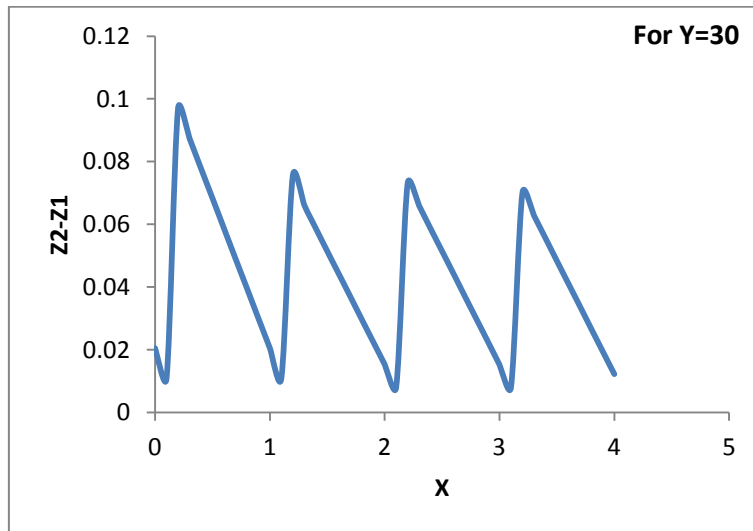


Figure 6.22: Plot for the scallop height of part 1 at y=30mm for 3-axis simulated part.

The graphs shown in Figure 6.20, 6.21 and 6.22 have been plotted for x-axis direction (perpendicular to across the tool path direction, which is y-axis). A good estimate of scallop height can be made from the uncut material by comparing the side step direction distance with the difference in z height of original STL file and simulated STL file. The errors are there for each side step as explained in section 6.3.1. The graph shows that the error for each side step is not more than the 0.2mm which is in the permissible range.

2. Plot between the original part and 5-axis simulated part at y=10, 20 and 30mm in Figure 6.23, 6.24 and 6.25 respectively for part 1.

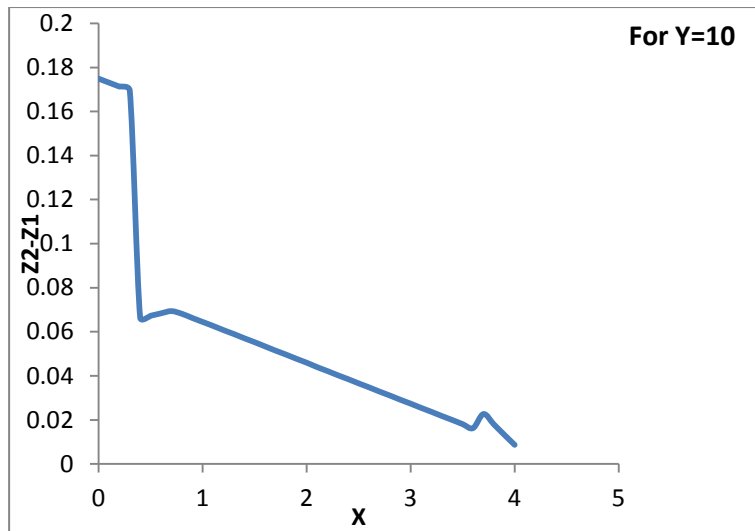


Figure 6.23: Plot for the scallop height of part 1 at y=10mm for 5-axis simulated part.

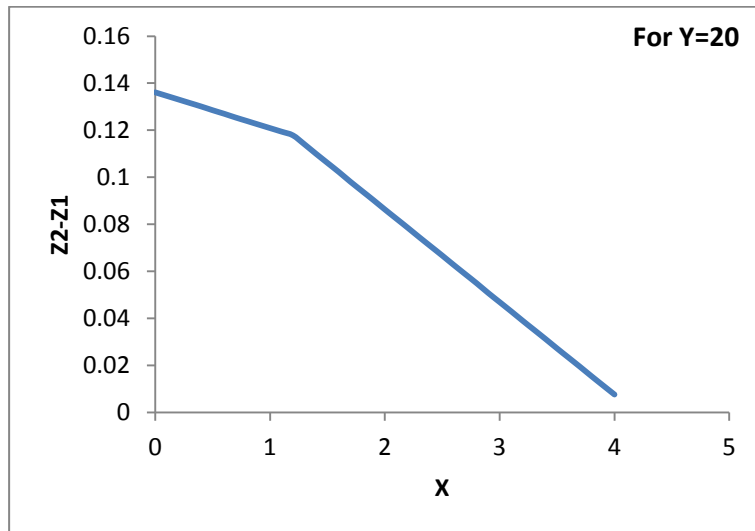


Figure 6.24: Plot for the scallop height of part 1 at y=20mm for 5-axis simulated part.

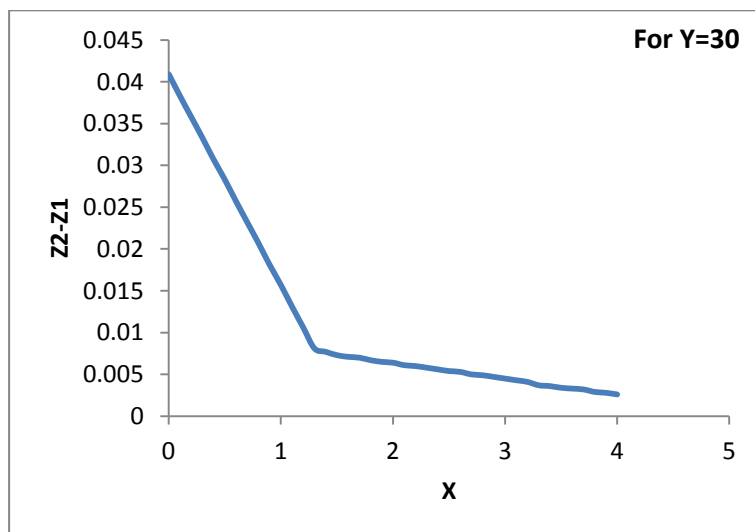


Figure 6.25: Plot for the scallop height of part 1 at y=30mm for 5-axis simulated part.

Similarly the graphs shown in Figure 6.23, 6.24 and 6.25 have been plotted for x-axis direction (perpendicular to across the tool path direction which is y-axis). The side

step in 5-axis is more as compared to the tool path for 3-axis machining. In this case, tool permits to be close to the surface of work piece so scallop height is same. The scallop is much smooth and this is not more than 0.2mm.

3. Plot between the original part and 3-axis simulated part at $y=10, 20$ and 30mm in Figure 6.26, 6.27 and 6.28 respectively for part 2.

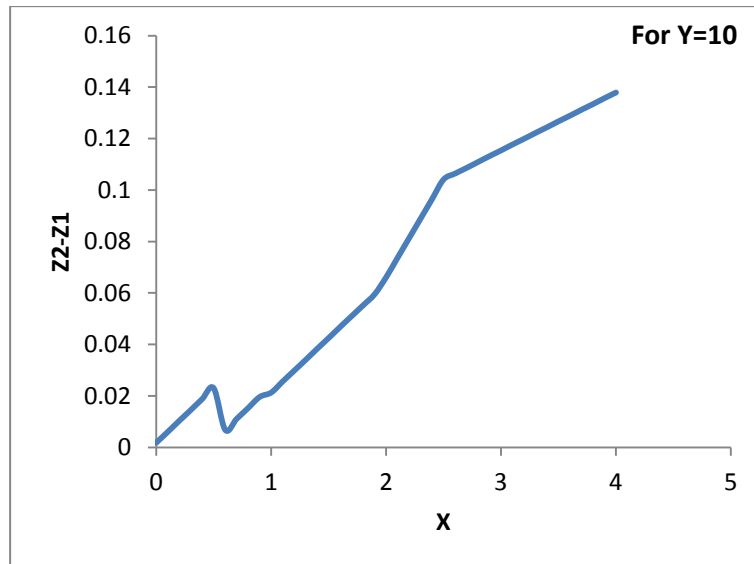


Figure 6.26: Plot for the scallop height of part 2 at $y=10\text{mm}$ for 3-axis simulated part.

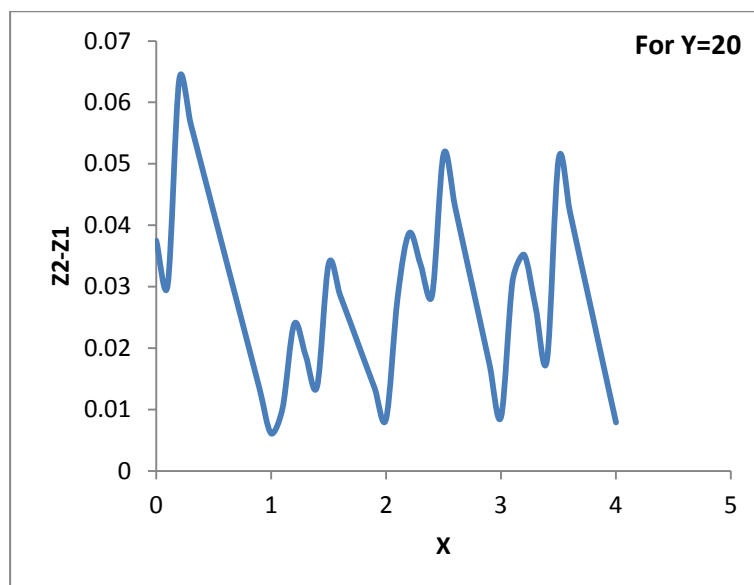


Figure 6.27: Plot for the scallop height of part 2 at $y=20\text{mm}$ for 3-axis simulated part.

Similarly as explained for part 1 for 3-axis machining, the Figure 6.26, 6.27 and 6.28 shows that the scallop is same for each step and is within the range of 0.2mm.

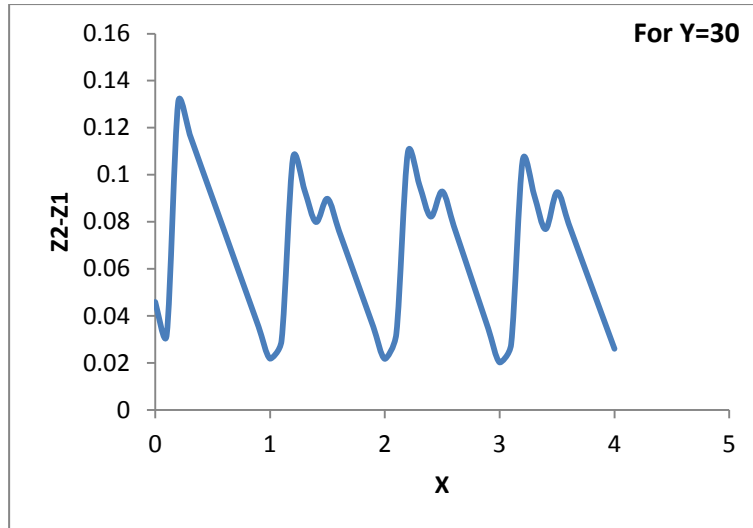


Figure 6.28: Plot for the scallop height of part 2 at y=30mm for 3-axis simulated part.

4. Plot between the original part and 5-axis simulated part at y=10, 20 and 30mm in Figure 6.29, 6.30 and 6.31 respectively for part 2.

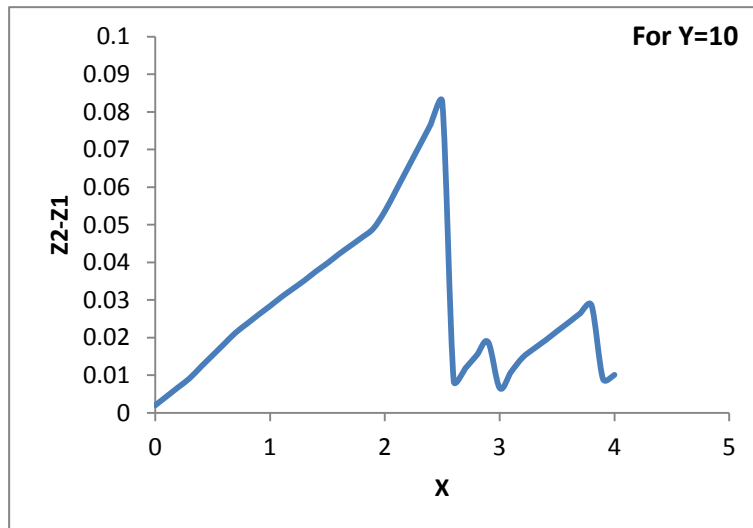


Figure 6.29: Plot for the scallop height of part 2 at y=10mm for 5-axis simulated part.

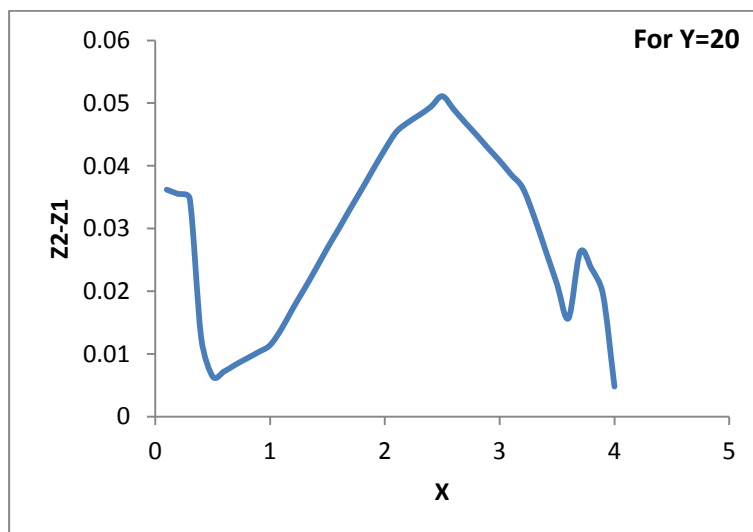


Figure 6.30: Plot for the scallop height of part 2 at y=20mm for 5-axis simulated part.

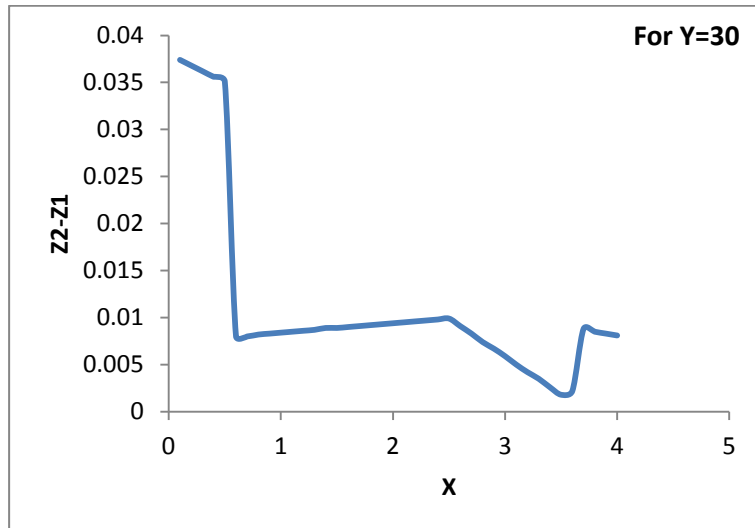


Figure 6.31: Plot for the scallop height of part 2 at y=30mm for 5-axis simulated part.

Similarly as explained for part 1 for 5-axis machining, the Figure 6.29, 6.30 and 6.31 shows that the scallop is same for each step and is within the range of 0.1mm.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

The conclusion and future scope for the dissertation work has been discussed in this chapter.

7.1 CONCLUSION

In the present work the details about various procedures for development of automatic macro for the simulation of tool path for 3-axis and 5-axis machining with the help of SolidWorks API has been discussed. The Boolean approach and solid sweep approach are used for developing the macro for simulation of tool path. The Boolean approach is slower compared to solid sweep approach. But the solid sweep fails if there are sharp corners in the tool path in two consecutive tool locations. Also if there is self intersection of the tool the solid sweep fails. Thus for the consistent working of macro Boolean approach is used for obtaining simulated work piece geometry. The output file generated after the simulation process is saved as *.sldprt format and *.stl format. The verification of the tool path is done by visual and analytical approaches.

In the present work the simulation is intended to verify the input part geometry in STL format. The verification of the tool path is possible without the loss of data. In the developed macros without incurring the additional errors accumulated because of actual machining and inherent errors of error measurement procedure. But there are some other formats of inaccuracies found during our process which is like stair steps, they show the overcut.

For the verification of tool path, the tool path for 3-axis and 5-axis machining has been developed using algorithm developed by Duvedi et al. and the data has been provided by the authors [43]. The STL file for the machined part and the simulated part is generated. The visual verification is possible in CAD package. For the analytical verification the ray tracing methodology is used for the verification of simulated part with the original part. The comparison of simulated part in STL format with the original part in STL format is done using the user defined tolerance. The results for the tool path verification are discussed in previous section.

7.2 SCOPE FOR FUTURE WORK

The following activities can be undertaken for extending the work presented in this thesis work.

1. The computational speed can be increased by finding the solution for refreshing the RAM during simulation process.

2. The simulation methodology can be extended for different type of machining like 3-axis, 3.5-axis, 4-axis and 4.5-axis.
3. The cylindrical shaped raw stock can be used for the simulation process.

ANNEXURE 1

Tool path for 3 axis simulation.

S.No.	X	Y	Z
1	10	0.5	-9.5
2	10	1	-9.5
3	10	1.5	-9.5
4	10	2	-9.5
5	10	2.5	-9.5
6	10	3	-9.5
7	10	3.5	-9.5
8	10	4	-9.5
9	10	4.5	-9.5
10	10	5	-9.5
11	10	5.5	-9.5
12	10	6	-9.5
13	10	6.5	-9.5
14	10	7	-9.5
15	10	7.5	-9.5
16	10	8	-9.5
17	10	8.5	-9.5
18	10	9	-9.5
19	10	9.5	-9.5
20	10	10	-9.5
21	10	10.5	-9.5
22	10	11	-9.5
23	10	11.5	-9.5
24	10	12	-9.5
25	10	12.5	-9.5
26	11	0.5	-9.5
27	11	1	-9.5
28	11	1.5	-9.5
29	11	2	-9.5
30	11	2.5	-9.5
31	11	3	-9.5
32	11	3.5	-9.5
33	11	4	-9.5
34	11	4.5	-9.5
35	11	5	-9.5
36	11	5.5	-9.5
37	11	6	-9.5
38	11	6.5	-9.5
39	11	7	-9.5
40	11	7.5	-9.5
41	11	8	-9.5
42	11	8.5	-9.5

43	11	9	-9.5
44	11	9.5	-9.5
45	11	10	-9.5
46	11	10.5	-9.5
47	11	11	-9.5
48	11	11.5	-9.5
49	11	12	-9.5

ANNEXURE 2

Tool path for 5 axis machining

S.No	CL data			Second point on the axis of cutter		
	X	Y	Z	X1	Y1	Z1
1	0.5	0.50	-5.25	0.52132105	1.72148213	64.73933867
2	0.5	1.00	-5.3	0.58525821	3.44147607	64.65735791
3	0.5	1.50	-5.35	0.69173358	5.15849546	64.55406747
4	0.5	2.00	-5.4	0.84061745	6.87105753	64.42948359
5	0.5	2.50	-5.45	1.03172842	8.57768497	64.28362898
6	0.5	3.00	-5.5	1.26483366	10.27690769	64.11653284
7	0.5	3.50	-5.55	1.53964915	11.96726462	63.92823083
8	0.5	4.00	-5.6	1.85584008	13.6473055	63.7187651
9	0.5	4.50	-5.65	2.21302122	15.31559263	63.4881842
10	0.5	5.00	-5.7	2.6107574	16.97070263	63.23654315
11	0.5	5.50	-5.75	3.04856404	18.61122818	62.96390337
12	0.5	6.00	-5.8	3.52590774	20.23577972	62.67033269
13	0.5	6.50	-5.85	4.04220693	21.84298717	62.35590528
14	0.5	7.00	-5.9	4.59683258	23.43150156	62.0207017
15	0.5	7.50	-5.95	5.18910896	24.9999967	61.66480883
16	0.5	8.00	-6	5.81831448	26.5471708	61.28831984
17	0.5	8.50	-6.05	6.48368254	28.07174804	60.89133418
18	0.5	9.00	-6.1	7.18440251	29.57248013	60.47395755
19	0.5	9.50	-6.15	7.91962065	31.04814783	60.03630186
20	0.5	10.00	-6.2	8.68844122	32.49756244	59.57848519
21	0.5	10.50	-6.25	9.48992754	33.91956726	59.10063177
22	0.5	11.00	-6.3	10.3231031	35.31303894	58.60287191
23	0.5	11.50	-6.35	11.18695283	36.67688895	58.08534203
24	0.5	12.00	-6.4	12.08042424	38.01006481	57.54818452
25	0.5	12.50	-6.45	13.00242879	39.31155142	56.99154778
26	0.5	13.00	-6.5	13.95184316	40.58037231	56.41558614
27	0.5	13.50	-6.55	14.92751062	41.81559077	55.82045981
28	0.5	14.00	-6.6	15.92824248	43.01631106	55.20633484
29	0.5	14.50	-6.65	16.9528195	44.18167946	54.57338307
30	0.5	15.00	-6.7	17.9999934	45.31088532	53.92178208
31	0.5	15.50	-6.75	19.06848835	46.40316205	53.25171511
32	0.5	16.00	-6.8	20.15700255	47.45778806	52.56337104
33	0.5	16.50	-6.85	21.26420983	48.47408761	51.85694432
34	0.5	17.00	-6.9	22.38876122	49.45143167	51.13263491
35	0.5	17.50	-6.95	23.52928662	50.38923868	50.3906482
36	0.5	18.00	-7	24.68439649	51.28697524	49.63119498
37	0.5	18.50	-7.05	25.85268351	52.14415677	48.85449136
38	0.5	19.00	-7.1	27.03272428	52.96034808	48.0607587
39	0.5	19.50	-7.15	28.22308113	53.73516396	47.25022354
40	0.5	20.00	-7.2	29.42230377	54.46826959	46.42311755
41	0.5	20.50	-7.25	30.62893115	55.15938096	45.57967745

42	0.5	21.00	-7.3	31.84149317	55.80826522	44.72014493
43	0.5	21.50	-7.35	33.05851252	56.414741	43.84476657
44	0.5	22.00	-7.4	34.27850644	56.97867856	42.95379379
45	0.5	22.50	-7.45	35.49998856	57.5	42.04748277
46	0.5	23.00	-7.5	36.7214707	57.97867935	41.12609434
47	0.5	23.50	-7.55	37.94146466	58.41474259	40.18989394
48	0.5	24.00	-7.6	39.15848408	58.80826761	39.23915151

REFERNCES

- [1] W.A. Hunt and H.B. Voelcker, "An exploratory study of automatic verification of programs for numerically controlled machine tools," Production Automation Project Tech. Memo No. 34. Rochester. New York: University of Rochester, 1982.
- [2] A.Spence, Y. Altintas and D. Kirkpatrick, "Direct calculation of machining parameters from a solid model," *Computers in Industry*, 14 (4), 271–280,1990.
- [3] D. Badouel, "An efficient ray-polygon intersection," In *Graphics gems*, pages 390–393." Academic Press Professional, Inc., 1990.
- [4] A.A.G. Requicha and J.R. Rossignac, "Solid modelling and beyond," *IEEE Computer Graphics and Applications*, 12 (5), 31–44, 1992.
- [5] J.M. O'Connell and A.G. Jablokow, "Construction of solid models from NC machining programs," *ASME Production Engineering Division (Publication)*, 64, 157–166, 1993.
- [6] S.X. Li, R.B. Jerard, "5-axis machining of sculptured surfaces with a flat-end cutter," *Computer-Aided Design*;26(3):165-78, 1994.
- [7] A.D.Spence and Y. Altintas, "A solid modeler based milling process simulation and planning system," *ASME Journal of Engineering for Industry*, 116, 61–69, 1994.
- [8] H.El-Mounayri, M.A.Elbestawi, A.D.Spence, and S.Bedi, "General geometric modelling approach for machining process simulation," *The International Journal of Advanced Manufacturing Technology*, 13(4):237–247, 1997.
- [9] T. Moller and B. Trumbore, "Fast, minimum storage ray-triangle intersection," *Journal of graphics tools*, 2(1):21–28, 1997.
- [10] H. El-Mounayri, A.D. Spence and M.A. Elbestawi, "Milling process simulation—a generic solid modeller based paradigm," *Journal of Manufacturing Science and Engineering*, 120 (2), 213–221, 1998.
- [11] K. Sheltami, S. Bedi, and F. Ismail, "Swept volumes of toroidal cutters using generating curves," *International Journal of Machine Tools and Manufacture*, 38(7):855–870,1998.
- [12] C.J.Su, F.H.Lin and L.Ye, "A new collision detection method for CSG-represented objects in virtual manufacturing," *Computers in Industry*, 40 (1), 1–13, 1999.
- [13] D. Roth, S. Bedi, and F. Ismail, "Generation of swept volumes of toroidal endmills in five-axis motion using space curves," In *Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 306–311. ACM, 1999.

- [14] A. Warkentin, P. Hoskins, F. Ismail, S. Bedi, "Computer aided 5-axis machining," In: Computer aided design, engineering and manufacturing: system techniques and applications. CRC Press; 2001 [chapter 3].
- [15] S.Ho, S.Sarma and Y.Adachi, "Real-time interference analysis between a tool and an environment," Computer-Aided Design, 33 (13), 935–947,2001.
- [16] B.M.Imani and M.A.Elbestawi, "Geometric simulation of ball-end milling operations," Journal of Manufacturing Science and Engineering, 123 (2), 177–184, 2001.
- [17] R. Segura and F. Feito, "Algorithms to test ray-triangle intersection. comparative study," In WSCG (Short Papers), pages 76–81, 2001.
- [18] B.H. Kim, B.K. Choi, "Machining efficiency comparison direction-parallel tool path with contour-parallel tool path," Computer-Aided Design; 34(2): 89-95, 2002.
- [19] M. Munlin, "Tool path simulation using a virtual 5-axis milling machine," In Industrial Technology,2002. IEEE ICIT'02. 2002 IEEE International Conference on, volume 1, pages 193–198. IEEE, 2002.
- [20] H.El-Mounayri, H.Kishawy and V.Tandon, "Optimized CNC end-milling: a practical approach," International Journal of Computer Integrated Manufacturing, 15 (5), 453–470, 2002.
- [21] S.H.Lee, and S.K.Lee, "Local mesh decimation for view-Independent three-axis NC milling simulation," International Journal of Advanced Manufacturing Technology,19 (8), 579–586, 2002.
- [22] S.K.Lee and S.L.Ko, "Development of simulation system for machining process using enhanced Z map model," Journal of Materials Processing Technology, 130–131, 608–617, 2002.
- [23] S. Mann and S. Bedi, "Generalization of the imprint method to general surfaces of revolution for NC machining," Computer-aided design, 34(5):373–378, 2002.
- [24] E.L.J. Bohez, N. Thi Hong Minh, B. Kiatsrithanakorn, P. Natasukon, H. Rueil-Yun, and Le T. Son, "The stencil buffer sweep plane algorithm for 5-axis cnc tool path verification," Computer-Aided Design, 35(12):1129–1142, 2003.
- [25] E. Pezzuti, G. Piscopo, A. Ubertini, P.P. Valentini, M. Milana, R. Di Leginio, R. Una metodologia per l'analisi e l'archiviazione di reperti archeologici basata sul rilievo mediante scanner laser tridimensionali a non contatto, Archiviazione e Restauro di Reperti Archeologici Mediante Tecniche CAD-RP (in italian), Napoli, 2004.
- [26] R.V. Fleisig and A.D. Spence, "Techniques for accelerating B-Rep based parallel machining simulation," Computer-Aided Design, 37 (12), 1229–1240, 2005.

- [27] R.S. Lin, "Real-time Surface Interpolator for 3-D Parametric Surface Machining on 3-Axis Machine Tools," *International Journal of Machine Tools and Manufacture*, Vol. 40, pp. 1513-1526, 2006.
- [28] T. Chang, R.A. Wysk, and H. Wang, "Computer-aided manufacturing," 3rd ed. Upper Saddle River: Person Prentice Hall, 2006.
- [29] Ph. Lorong, J. Yvonnet, G. Coffignal, and S. Cohen, "Contribution of computational mechanics in numerical simulation of machining and blanking: State-of-the-art," *Archives of Computational Methods in Engineering*, 13(1):45–90, 2006.
- [30] D. Yip-Hoi and X.M. Huang, "Cutter/workpiece engagement feature extraction from solid models for end milling," *Journal of Manufacturing Science and Engineering*, 128 (1), 249–260, 2006.
- [31] S. Sukkromsai, "The Integrated Simulation and Verification of a Specific 5-axis Machine," *The IE Network Conference 2007*.
- [32] K.P. Karunakaran and R. Shringi, "Octree-to-BRep conversion for volumetric NC simulation," *International Journal of Advanced Manufacturing Technology*, 32 (1–2), 116–131, 2007.
- [33] W. Anotaipaiboon, S.S. Makhanov, "Curvilinear space-filling curves for five-axis machining," *Computer-Aided Design*, 40(3):350-67, 2008.
- [34] M. Spens, "Automating Solidworks 2009 using Macros," SCHROFF development Corporation, Third edition, 2008.
- [35] T. Moriwak, "Multi-functional machine tool," *CIRP Annals-Manufacturing Technology*, 57(2):736–749, 2008.
- [36] W. Ferry and D. Yip-Hoi, "Cutter-workpiece engagement calculations by parallel slicing for five-axis flank milling of jet engine impellers," *Journal of Manufacturing Science and Engineering*, 130 (5), 051011.1–051011.12, 2008.
- [37] E. Dyllong and C. Grimm, "A reliable extended octree representation of CSG objects with an adaptive subdivision depth," *Lecture Notes in Computer Science*, 4967, 1341–1350, 2008.
- [38] S.W. Lee and A. Nestler, "5- Axis milling simulation using a swept volume via gauss map," In 16th WSCG International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, Czech Republic, 2008.
- [39] H. Erdim and H. T. Ilies, "A comparison of sampling strategies for computing general sweeps," *Computer-Aided Design*, 42(8):657–669, 2010.

[40] J. Zhang, Soh-Khim Ong, and Y.C. Nee Andrew, “A multi-regional computation scheme in an AR-assisted in situ CNC simulation environment.” *Computer-Aided Design*, 42(12):1167–1177,2010.

[41] A. Sullivan, H. Erdim, R.N. Perry, and S.F. Frisken, “High accuracy nc milling simulation using composite adaptively sampled distance fields,” *Computer-Aided Design*, 44(6):522–536, 2012.

[42] R. K. Duvedi, S. Bedi, A. Batish, and S. Mann, “A multipoint method for 5-axis machining of triangulated surface models,” *Computer-Aided Design*, 52:17–26, 2014.

[43] R. K. Duvedi, S. Bedi, A. Batish, and S. Mann, “Scallop Height of 5-axis Machining of Large Triangles with a Flat End Mill,” Annual International CAD conference 2014, Hong Kong University of Science and Technology, China, June 2014.

NOTE: The document writing has done using the IEEE format.