A

DISSERTATION REPORT

on

# DEVELOPMENT OF MACHINING SIMULATOR WITH ERROR AND SURFACE ROUGHNESS DISPLAY

---

**Submitted in partial fulfillment of the requirements for the award of degree of**

**MASTER OF ENGINEERING**
**IN**
**CAD/CAM & ROBOTICS**


**Submitted by:**
**HARSIMRAN SINGH**
**ROLL NO: 801181015**



**Under the guidance of**
**Mr. A. S. JAWANDA**
**Associate Professor, MED**



**MECHANICAL ENGINEERING DEPARTMENT**
**THAPAR UNIVERSITY**
**PATIALA-147004, INDIA**

**JULY- 2013**

# CERTIFICATE

I hereby certify that the dissertation entitled "**DEVELOPMENT OF MACHINING SIMULATOR WITH ERROR AND SURFACE ROUGHNESS DISPLAY**" is an authentic record of my own work carried out in partial fulfillment of the requirement for the award of degree of **Master of Engineering** in **CAD/CAM & Robotics** under the guidance of **Mr. A.S. JAWANDA**, Assistant Professor, MED, Thapar University, Patiala.

Date: - 15/7/2013

Harsimran Singh

**HARSIMRAN SINGH**

**(801181015)**

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

**Mr. A. S. JAWANDA**

**Associate Professor, MED**

Mechanical Engineering Department

Thapar University, Patiala

Countersigned By: -

**Dr. S. K. Mohapatra**

Senior Professor and Dean of Academic Affairs

Mechanical Engineering Department

Thapar University, Patiala

Countersigned By:-

**Dr. Ajay Batish**

**Professor and Head**

**Mechanical Engineering Department**

**Thapar University, Patiala**

# ACKNOWLEDGEMENT

I express my sincere gratitude to my guide **Mr. A. S. Jawanda**, Associate Professor Mechanical Engineering Department, Thapar University, Patiala, for their invaluable guidance, proper advice and constant encouragement during the course of my work in this dissertation report.

I would also express my gratitude to **Dr. Ajay Batish**, Head, Mechanical Engineering Department for his inspirational guidance and wholehearted corporation which was helped me to present this Seminar.

I thank the entire faculty and staff of Mechanical Engineering Department, Thapar University, for their help and moral support.

(HARSIMRAN SINGH)

# ABBREVIATIONS

| | |
|---|---|
| NC: | Numeric Controlled |
| CNC: | Computer Numeric Controlled |
| CAD: | Computer Aided Design |
| CAM: | Computer Aided Manufacturing |
| CL: | Cutter Location |
| CC: | Cutter Contact |
| STEP: | Standard for the Exchange of Product Model Data |
| IGES: | Initial Graphics Exchange Specification |
| STL: | Stereo Lithography |
| TP File: | Toolpath File |

# ABSTRACT

In today's fast growing and competitive world, the machining industries use CNC machines for accuracy, speed , repeatability and flexibility. The cost of these machines is very high. As these are computer operated using very complex , large programs for controlling the tool motion, which are generated using automatic programming tools, there is always an uncertainty of the results and motion of these machines. Therefore there is need for some prediction of the outcome of the machining on a CNC machine.

In this work, a machining simulator is developed for a 3- axis lathe mill using OAC technology. The simulator visually displays the machining being done. It also calculates the error which comes between the actual machined part and required geometry, in STL format which was taken for the tool path generation. The error calculated is normal to the surface of the geometry.

The simulator is based on "mow the grass" technique in which the workpiece is represented by radial grass for lathe mill CNC. The tool position is read from the toolpath file and intersection with grasses is calculated and the radius of grasses is constantly updated which generates an animation of machining environment. The error is measured normal to the direction of machined surface from which conditions like undercutting and overcutting can be estimated. The average of error is also an indicative 'Ra' value of the machined surface.

The machined surface , its error of scallop direction can be seen by OpenGL rendering with mouse based UI for zoom, pan and 3D rotate function with a color legend for knowing the magnitude of error.

# Table of Contents


CHAPTER 1 INTRODUCTION ........ 1

1.1 Simulator ........ 3

1.1.1 Types of simulation ........ 4

1.2 Need of a machining simulator ........ 4

1.2.1 Advantages of a machining simulator ........ 5

1.3 Types of surface topography error ........ 6

CHAPTER 2 LITERATURE SURVEY ........ 9

2.1 Algorithm to find Intersection of a workpiece and Cutter [W5] ........ 11

2.2 Other techniques used for machining simulation ........ 12

2.2.1 Simulation based on STEP-NC [4] [6]: ........ 12

2.2.2 Adaptive NC simulation [8]: ........ 15

2.2.3 Verification of a machine simulator [2] [8] [9] ........ 17

2.3 Conclusions drawn from Literature Survey ........ 18

CHAPTER 3 PROBLEM DEFINITION ........ 20

CHAPTER 4 SIMULATOR DESIGN ........ 22

4.1 Flowchart for proposed simulator ........ 23

4.2 Algorithm to calculate error between machined and actual geometry ........ 24

4.3 Simulator testing ........ 25

4.3.1 Cylinder ........ 25

4.3.2 Variable inclination triangular facet part ........ 29

CHAPTER 5 CONCLUSIONS ........ 32

5.2 FUTURE SCOPE OF WORK ........ 33

REFERENCES ........ 34

WEB REFERENCES ........ 35

# List of Figures Page no.

Figure 1.1 A Machining Simulation Scene [w1 ] ........ 1
Figure 1.2: A Cad & Cam system showing the three phases of CNC machining [1]........ 2
Figure 1.3: A machining simulation scene [w2]........ 3
Figure 1.4: Tool and Workpiece Collision [w3]........ 5
Figure 1.5: General terms of Surface topography [w4] ........ 7
Figure 1.6: A Surface profile with waviness and roughness [1]........ 8
Figure 2.1 Simulation for Three Axis milling [1]........ 9
Figure 2.2: A height field over a cylinder [4] ........ 10
Figure 2.3 A picture from the milling lathe simulator [4] ........ 10
Figure 2.4: Intersection of a line and a sphere [w5] ........ 11
Figure 2.5: A Siemens Machine Having a Step NC Controller [w6] ........ 13
Figure 2.6 Framework of STEP-NC Controller with OMI [6] ........ 14
Figure 2.7: Octree data structure and the associated voxel model [7] ........ 16
Figure 2.8: Stock Representation with a Vertical Height Field [2] ........ 17
Figure 3.1: The general method for predicting machining results........ 20
Figure 4.1: The algorithm for proposed simulator........ 23
Figure 4.2: Algorithm to calculate error between actual and machined geometry........ 24
Figure 4.3: Cylindrical part along X-axis Modeled in CREO 2.0 ........ 25
Figure 4.4: STL representation of Cylindrical Part ........ 26
Figure 4.5: Tool path file used as an input for simulator........ 26
Figure 4.6: Cylindrical part being machined with STL geometry displayed........ 27
Figure 4.7: Final machined part displayed without STL ........ 28
Figure 4.8 Error displayed in Red color........ 28
Figure 4.9 The CAD Model of the part........ 29
Figure 4.10: STL representation of the part........ 29
Figure 4.11 Machining being done on the part........ 30
Figure 4.12: Final machined part........ 30
Figure 4.13: Error displayed in green color........ 31

# CHAPTER 1
# INTRODUCTION

Simulation is the act of imitating the behavior of some situation or some process by means of something suitably analogous. In computers it is the technique of representing the real world by a computer program. A computer based simulation is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. By changing variables, predictions may be made about the behavior of the system .Computer simulation has become a useful part of modeling many natural systems in physics, chemistry and biology, and human systems in economics and social science as well as in engineering to gain insight into the operation of those systems. There are many different types of computer simulation. The common feature they all share is the attempt to generate a sample of representative scenarios for a model in which a complete enumeration of all possible states would be prohibitive or impossible. The purpose of simulation may be training or research.

Machining simulation is the 3D representation of the cutting tool and work piece. It is used to simulate the tool path for machining process for verifying it against the actual shape requirement.

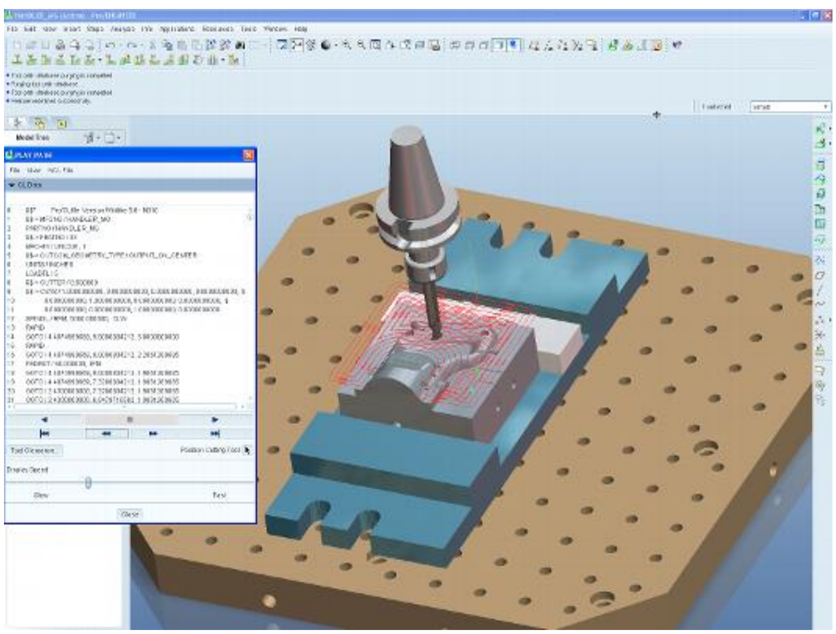


**Figure 1.1 A Machining Simulation Scene [w1 ]**

Machining Simulation thus verifies and analyzes the tool path without any physical machining requirement of the work piece. The idea is to use the simulation results as basis for making decision for the real life machining process. The input parameters can be varied to study the behavior of the real time process.

Complex sculptured surfaces are widely used in many engineering objects such as turbines blades, automobile components, aerospace parts and medical implants. The manufacturing of such custom intricate designs is an expensive and time consuming task. Computer aided design and Computer Aided Manufacturing technology has been around for decades to assist with design and manufacturing of such components. With CAD/ CAM software, Computer Numeric Controlled machines can machine intricate components with highest dimensional accuracy. CAD/CAM software simplifies the programming of CNC machines and helps improve their utilization.

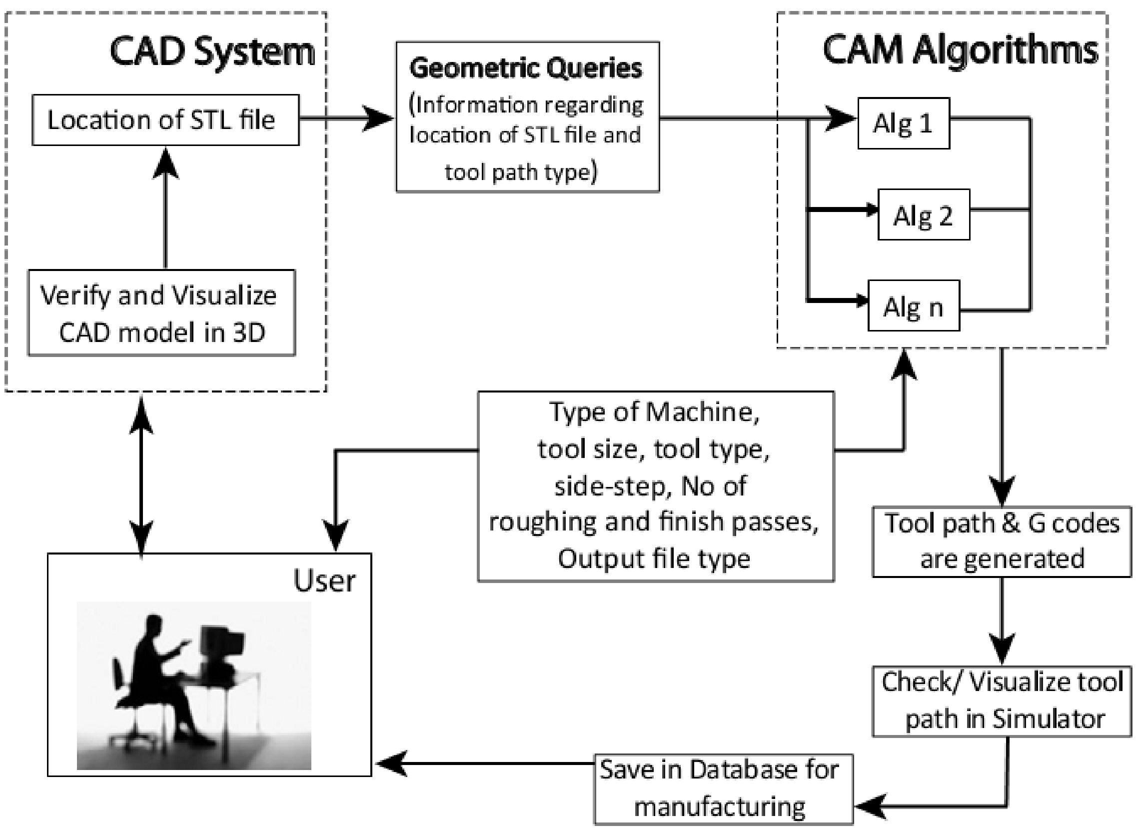



**Figure 1.2: A Cad & Cam system showing the three phases of CNC machining [1]**

The complete CNC machining process is divided into three stages: Design phase, Tool Path Planning stage, Verification and manufacturing stage. In the Design stage, a 3D Computer Aided

Design model for the design surface is created using the modeling capabilities of a CAD packages. In the Tool Path Planning stage, the path for the cutter is planned to produce the component from the CAD model within acceptable dimensional tolerances. In this stage, cutter location data that can be translated into G-Codes that can be understood by CNC machines is generated and then this code will guide the tool to machine the component. In the Dedication stage, the generated tool path is verified and checked in a simulator for machining errors or laws before manufacturing the component. In the Manufacturing stage, the actual component is loaded on the machine and the machine tool will read the G -codes generated during tool path planning stage to produce a final product in its intended application.

## 1.1 Simulator

A computer simulator is a program run on a computer for the purpose of simulating a system, in order to get a better idea how the system functions. At its most fundamental level, a computer simulator is used to recreate a real-life situation where testing many different states wouldn't be easy or safe. In this case, using a computer simulator saves a great deal of time and resources, and may actually allow for in-depth testing that would be physically impossible in the real world. Computer simulations have been used in engineering and science for some time, and in recent years, as hardware have become more robust and software has become more sophisticated, these simulations are able to accomplish more and more.

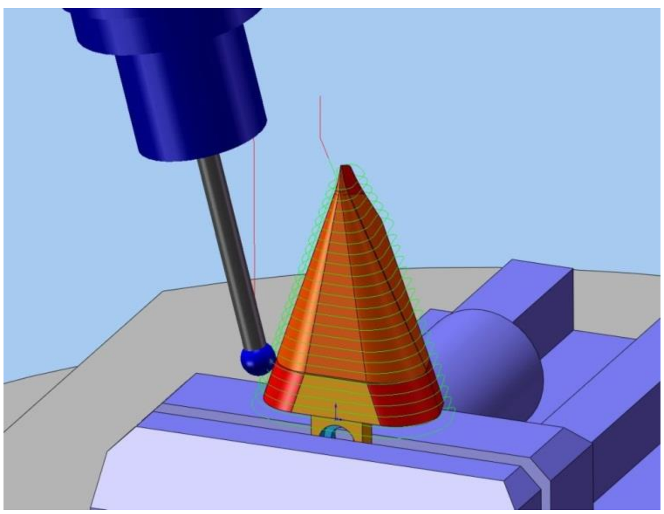


Figure 1.3: A machining simulation scene [w2]

An example of a use of a computer simulator might be an engineering company working on designing a better wing for aircraft. Rather than spending resources building prototypes of each new design, and testing them in wind tunnels or on real aircraft, a simulation can be run instead. If the fluid dynamics of the wind is accurately modeled, any number of variables can be modified to test a virtual wing in a million different scenarios, some of which might be impossible to recreate in the real world. Additionally, if a design turns out not to be ideal, it can be easily modified and retested, without the need for building a new physical object. In this way, hundreds or thousands of designs can be tried in a relatively short period of time, allowing for a much greater degree of fine tuning than would otherwise be possible.

### 1.1.1 Types of simulation

**Continuous-state simulation** is applicable to systems where the notion of state is continuous and typically involves solving (numerically) systems of differential equations. Circuit- level simulators are an example of continuous - state simulation.

**Discrete- event simulation** is applicable to systems in which the state of the system changes at discrete instants of time, with a finite number of changes occurring in any finite interval of time. The simulation in CNC is Continuous state simulation for checking of the tool path and for the machining of the geometry. Manufacturing represents one of the most important applications of Simulation. This technique represents a valuable tool used by engineers when evaluating the effect of capital investment in equipment and physical facilities like factory plants, ware ho uses, and distribution centers.

## 1.2 Need of a machining simulator

A worker's safety should be of utmost importance during a machining process. These days when use of high speed tool is growing very fast, an error in the machining NC program can cause serious accidents. A worker may get seriously injured; it may prove fatal to him.

Also the demands for high accuracy and reduced machining losses is very high, CNC machines have become a necessity for industries to remain in such a competitive market. Therefore every industry is opting for the implementation of CNC machining and machining simulators all over the world. Although the initial cost of these machines is very high, 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 workpiece 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 is essential if programs are to be run with confidence during an unmanned operation. So the various advantages of having a machining simulator are as following:

### 1.2.1 Advantages of a machining simulator

I. **Prevent accident**: With the implementation of the machining many accidents due to wrong tool path during machining, wrong tooling selection etc. can be avoided as can be seen in Figure 1.4 there is collision of the tool holder and the job thus simulation can prevent these type of accidents, also with the selection of the wrong tooling that can cause an accident can be visualized and thus accidents and the losses that can occur due to it can be projected.

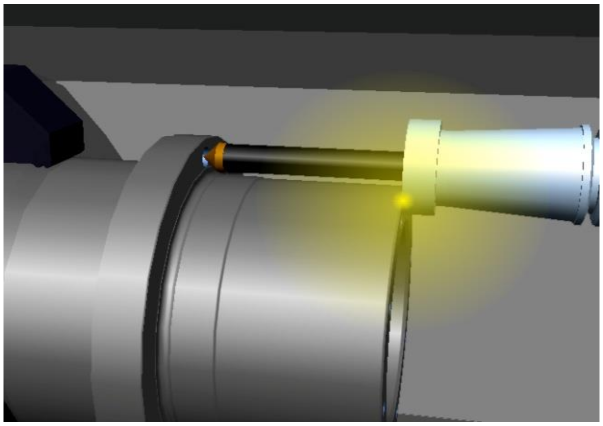


**Figure 1.4: Tool and Workpiece Collision [w3]**

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

III. **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.

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

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

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

VII. **Product cycle time reduced**: With the implementation of the machining simulators overall product cycle of bringing product into the actual machining environment is 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.

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

## 1.3 Types of surface topography error

Geometric errors are those errors that are always existent in a machine on account of its basic design, the inaccuracies built- in during assembly and as a result of the components used on the machine. As such, they form one of the biggest sources of inaccuracy. Geometric errors can be smooth and continuous or they could exhibit hysteresis or random behavior. Geometric errors have various components like linear displacement error, positioning accuracy, straightness and flatness of movement of the axis, spindle inclination angle, squareness error, backlash error etc. Such errors occur during the execution of linear, circular or other types of interpolation algorithms and are more pronounced during actual machining.

Various terms associated with Surface Topography are as discussed below:

- **Flaws**
- **Lay**
- **Surface roughness**
- **Waviness**
- **Flatness**

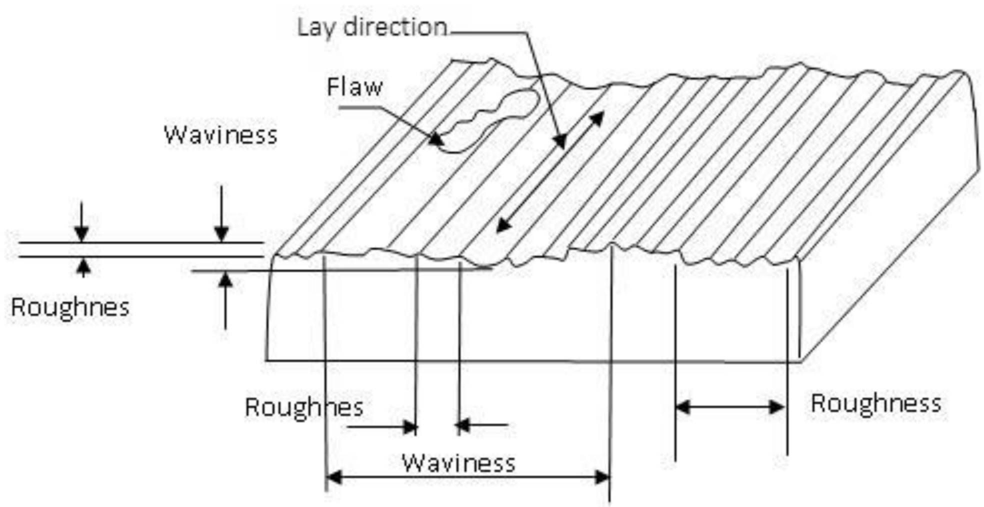



**Figure 1.5: General terms of Surface topography [w4]**

**Flaws** are random irregularities, such as scratches, cracks, holes, depression, seams, tears or inclusions. These defects can be caused during the machining or production process such as moulding.

**Lay** is a measure of the direction of the predominant machining pattern. A lay pattern is a repetitive impression created on the surface of a part. It is often representative of a specific manufacturing operation

**Surface roughness** is defined as closely spaced, irregular deviation on a scale smaller than that of waviness. It is caused by the cutting tool or the abrasive grain action and the machine feed. Roughness may be superimposed on waviness.

**Waviness** is the measure of surface irregularities with spacing greater than that of surface roughness. These usually occur due to warping, vibrations, or deflection during machining. The final surface profile generated after machining is a combination of waviness and roughness as seen in Figure 1.6.

**Flatness** is the tolerance zone defined by the two planes in which the surface lies.

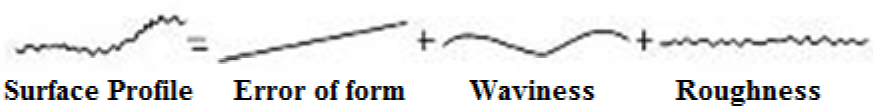



**Figure 1.6: A Surface profile with waviness and roughness [1]**

# CHAPTER 2

# LITERATURE SURVEY

The literature referred on machining simulations has revealed various fields related to the simulators and the techniques used for simulating the machining process. There are various techniques which are used to simulate the machining environment. Every technique has got its pros and cons. These are enlisted below:

**Balram Kalra** [1] developed a simulator for three axis milling. The workpiece was represented by projecting vertical grass (vectors) from the base surface. The grasses were equally spaced by very small interval of the order of 0.5 mm. Thus grass density was high for representing the workpiece. The top surface was represented by the triangles in regular pattern. The machined surface is calculated by considering the intersection of tool swept volume and the vertical grasses. The grass coordinates are constantly updated.
A picture of the interface developed by Balram is as shown in the Figure 2.1. The simulator as developed by him shows a triangulated surface.

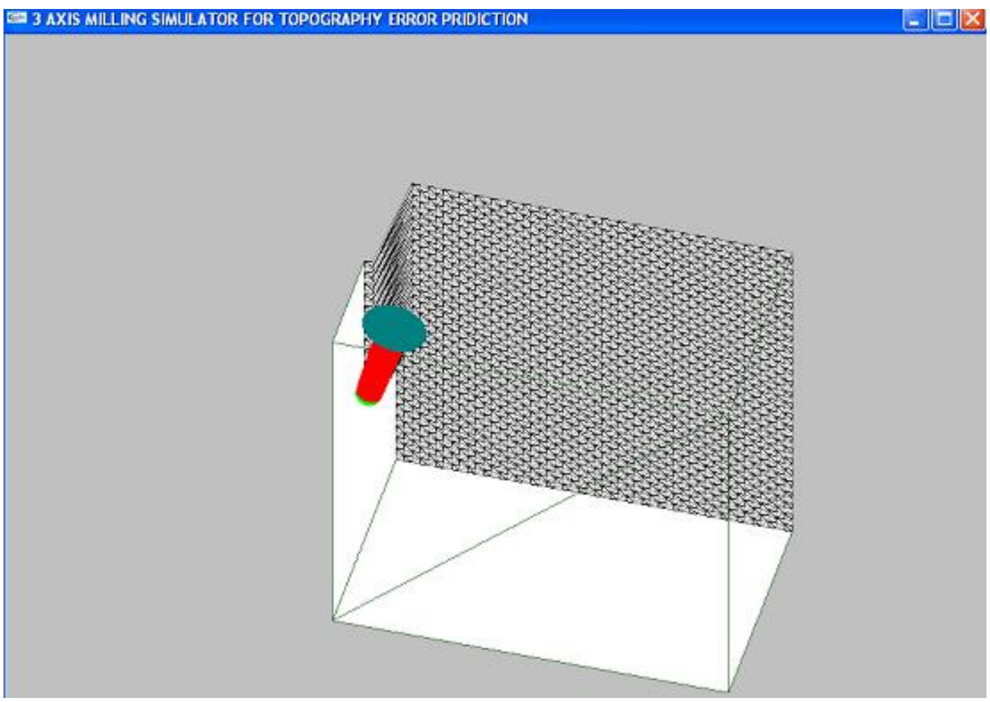



Figure 2.1 Simulation for Three Axis milling [1]

**Kaplan** et al. [4] developed a simulator for the prototype milling lathe. Their simulator is a variation on the height field (“mow the grass") method commonly used for 3-axis simulation. For the milling lathe, however, rather than storing a z -map over a discrete xy-grid, the height field emanates radially from the rotational axis of the lathe as shown in Figure 2.2. Figure 2.2

To simulate the cutting of the cylindrical stock by the tool, computation of a piecewise linear sampling of the grazing curve at each tool position is done, and sweep the grazing curves between adjacent tool positions. This computation gives a set of polygons that is intersected with the radial height field. The tips of the height field are connected to form a triangular mesh, which are render in an OpenGL renderer. The machining simulation is as shown in Figure 2.3

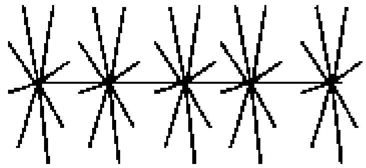


**Figure 2.2: A height field over a cylinder [4]**

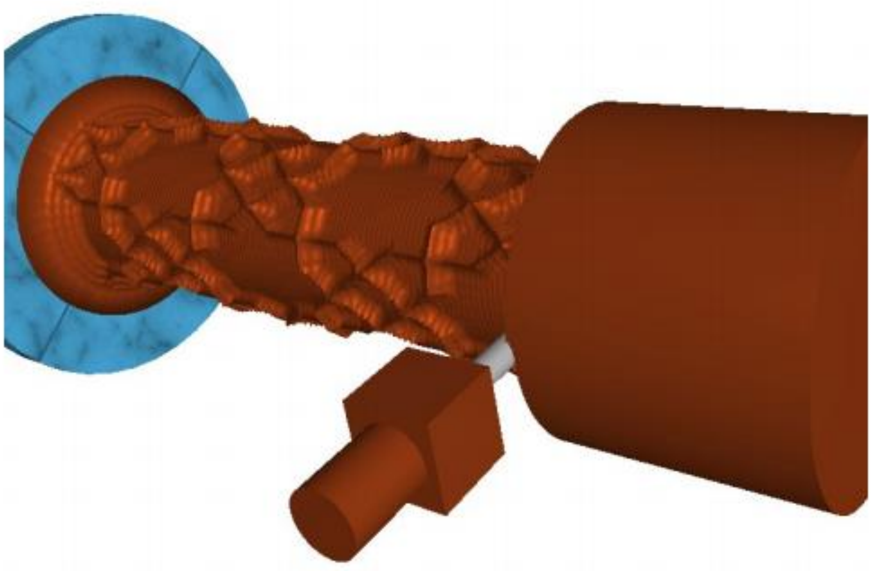


**Figure 2.3 A picture from the milling lathe simulator [4]**

## 2.1 Algorithm to find Intersection of a workpiece and Cutter [W5]

Intersection between a line and a sphere can be found using the algorithm as suggested by Paul Bourke [w5].

Points **P** (x, y) on a line as shown in defined by two points $\mathbf{P}_1$ ($x_1$, $y_1$, $z_1$) and $\mathbf{P}_2$ ($x_2$, $y_2$, $z_2$) is described by the following parametric equation.

$\mathbf{P} = \mathbf{P}_1 + u\ (\mathbf{P}_2 - \mathbf{P}_1)$ or in each coordinate

$x = x1 + u\ (x2 - x1)$

$y = y1 + u\ (y2 - y1)$

$z = z_1 + u\ (z_2 - z_1)$

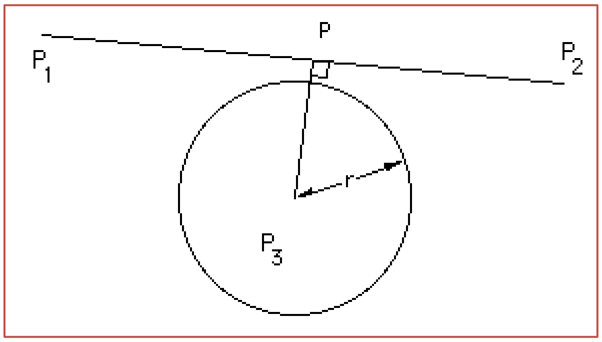



**Figure 2.4: Intersection of a line and a sphere [w5]**

A sphere centered at $\mathbf{P}_3$ ($x_3$, $y_3$, $z_3$) with radius r is described by:

$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = r^2$

Substituting the equation of the line into the sphere gives a quadratic equation of the form $a\ u^2 + b\ u + c = 0$.

Where:

$a = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2$

$b = 2[(x_2 - x_1)\ (x_1 - x_3) + (y_2 - y_1)\ (y_1 - y_3) + (z_2 - z_1)\ (z_1 - z_3)]$

$c = x_3^2 + y_3^2 + z_3^2 + x_1^2 + y_1^2 + z_1^2 - 2[x_3\ x_1 + y_3\ y_1 + z_3\ z_1] - r^2$

The solutions to this quadratic are described by

$$\frac{-b\ \pm\ sqrt(b^2 - 4ac)}{2a}$$

Depending upon the value of $b^2 - 4ac$ the following three cases may arise

a) If this is less than zero then the line does not intersect the sphere.
b) If it equals zero than the line is a tangent to the sphere intersecting it at one point, namely at u = -b/2a.
c) If it is greater than zero the line intersects the sphere at two points.

## 2.2 Other techniques used for machining simulation

There are a lot of techniques which are used to perform a computer based machining simulation. Apart from “mow the grass” method some of the commonly used techniques are enlisted below:

### 2.2.1 Simulation based on STEP-NC [4] [6]:

STEP stands for "Standard for the Exchange of Product model data”. It is an ISO 10303 standard for the computer-interpretable representation and exchange of product manufacturing information. ISO 10303 can represent 3D objects. Its official title is: Automation systems and integration — Product data representation and exchange.
The International standard's objective is to provide a mechanism that is capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

Typically STEP can be used to exchange data between CAD, Computer-aided manufacturing, Computer-aided engineering, Product Data Management and other CAx systems. STEP is addressing product data from mechanical and electrical design, Geometric dimensioning and tolerancing, analysis and manufacturing, with additional information specific to various industries such as automotive, aerospace, building construction, ship, oil and gas, process plants and others.
STEP-NC, which is the extension of STEP to the digital manufacturing fields, becomes a new data interface based on STEP between CAD/CAM and CNC. In 2003, the draft International Standard ISO 14649 was developed in Europe. The STEP-NC data model provides a higher level of information in the manufacturing process, enables a bidirectional information flow. STEP-NC data model provides the possibility for OMI being integrated into NC machining. In traditional

closed-loop machining system, the inspecting process was carried out on CMM instead of on NC machine tools.


Figure 2.5: A Siemens Machine Having a Step NC Controller [w6]

Therefore, the workpiece has to be moved from the machine tools to CMM when measurement was necessary and the reposition of workpiece was needed after the inspection process. OMI is the automatic inspecting process on CNC machine tools when the part is still on it. OMI on CNC machine tools has become a key method to improve the quality of products and save lead-time.

Xiaoming Zhu introduced STEP-NC data model （ISO 14649）to the integration of NC machining and On-machine Inspection (OMI) to improve the quality of machined parts. A framework of STEP-NC Controller with OMI is proposed in this paper. As machining and OMI simulation process play important roles in STEP-NC Controller with OMI to verify the machining and probing process, the implementing method of its functional modules including interpreter, machining simulation and inspecting simulation module which are designed by using Java and Java 3D were presented. Then a case study of an example part is also provided focusing on the integration of the machining and the probing processes, to demonstrate the feasibility of the generated machining path and probing path.

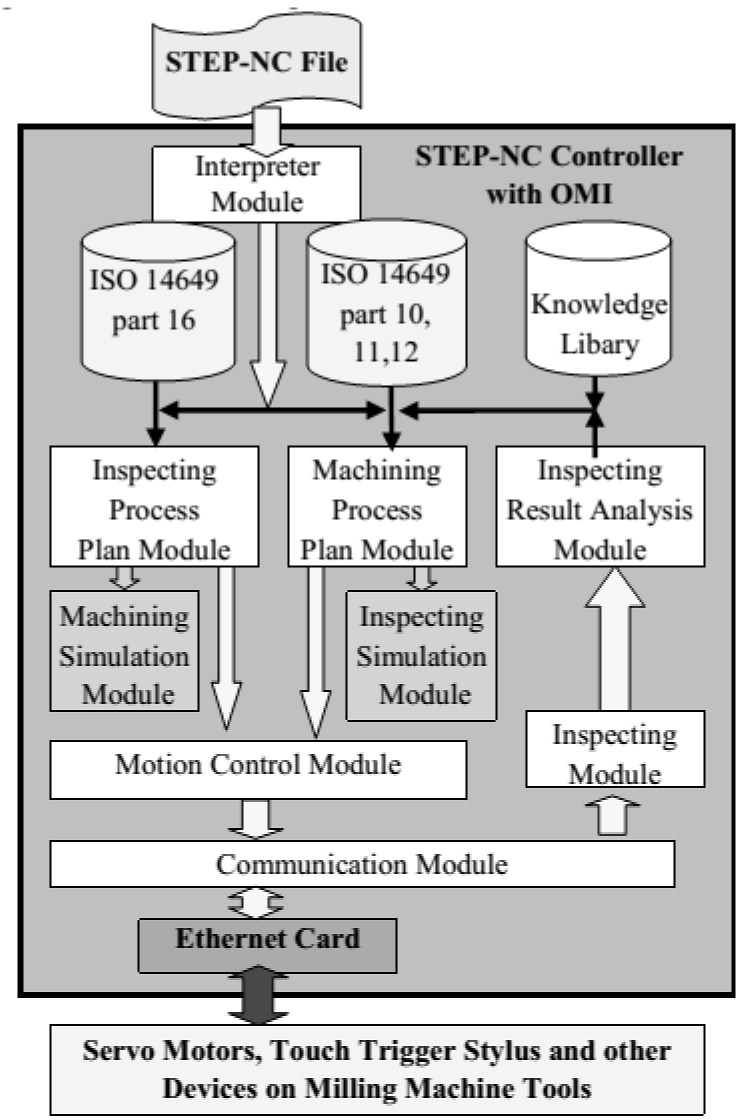



**Figure 2.6 Framework of STEP-NC Controller with OMI [6]**

Figure 2.6 shows the architecture of STEP-NC controller with OMI and the information flow among STEP-NC controller and its peripheral device on machine tools. STEP-NC controller with OMI consists of interpreter, database, machining and inspecting control modules, etc. In this STEP-NC Controller with OMI, both the machining and inspecting process are implemented through the same servo control system. The software structure of STEP-NC Controller with OMI based on Ethernet Field Bus is also shown in Figure 2.6.

Some benefits of using STEP-NC are summarized underneath [6].

1) Both feature description and model structures in ISO14649 are harmonized with ISO10303. So STEP-NC supports a bi-directional information transfer between CAD/CAM and CNC. As a result, modifications for the information about machining tasks and technological data on the shop-floor can be saved and transferred back to the planning department.

2) Post-processors will be eliminated because the interface does not require machine-specific information.

3) Machine tools are safer and more adaptable because STEP-NC is independent from the machine tool vendor.

4) STEP-NC provides a complete and structured data model, linked with geometric and technological information, so that no information is lost between the different stages of the process.

The STEP-NC was not successful because of the too much generalization of the machining process that it would have done. The market does not allow that because then there would not have been competition between various companies of this field.

### 2.2.2 Adaptive NC simulation [8]:

Hong T. Yau et al. [7 ] represented a workpiece using Octree base Voxel model. An adaptive regular mesh algorithm is proposed to achieve high frame rates for three-axis NC milling simulation. This approach can achieve optimization for fast rendering since the total number of triangles for rendering has been reduced significantly and the triangle density can be adjusted. It can also provide the capabilities for sustaining consistent frame rates as the direct relationship between the single threshold and the number of triangles to be rendered exists. Any area of the machined workpiece can be inspected with full resolution, such as an overcut, undercut or error assessment. Moreover, during a milling process simulation, the workpiece can be translated, rotated, zoomed in/out, or measured, which is especially important for virtual reality or augmented reality applications. In a virtual reality or augmented reality environment, the machinist will observe the milling process from different directions, and the distance between the machinist and the workpiece is always varying.

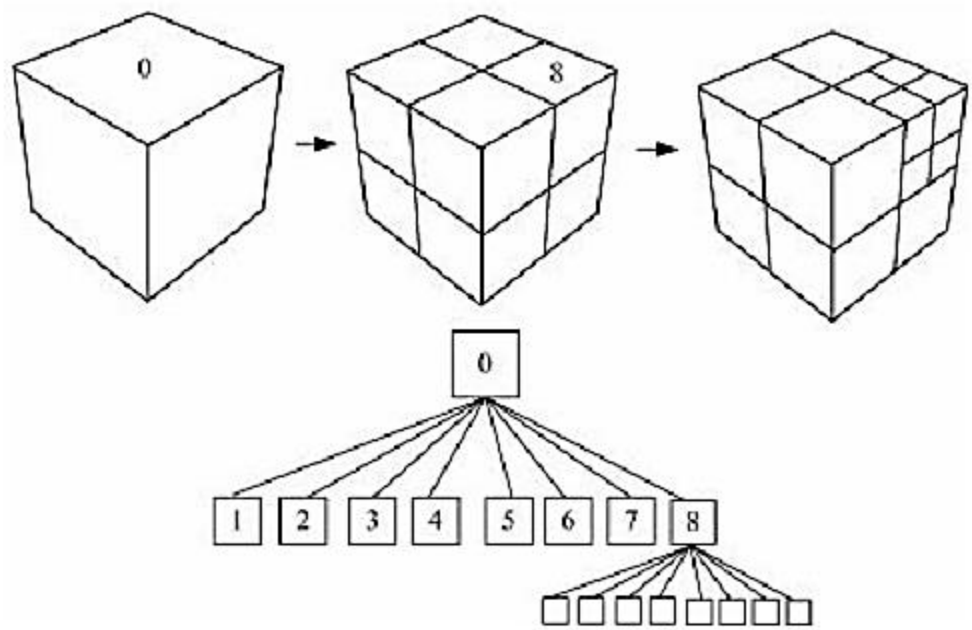



**Figure 2.7: Octree data structure and the associated voxel model [7]**

For the machining of a complicated surface, a large amount of linear NC segments are usually generated to approximate the surface with precision. If inaccurate NC codes are not discovered until the end of cutting, time and expensive material would be wasted. However, accurate and view-independent verification of multi-axis NC machining is still a challenge. This paper emphasizes the use of adaptive octree to develop a reliable multi-axis simulation procedure which verifies the cutting route and the workpiece appearance during and after simulation.

Voxel models with adaptive octree data structure are used to approximate the machined workpiece with specified resolution. Implicit functions are used to represent various cutter geometries for the examination of cutter contact points with speed and accuracy. It allows a user to do error analysis and comparison between the cutting model and the original CAD model. It can also verify the exactness of NC codes before machining is carried out by a CNC machine in order to avoid wasting material and to improve machining accuracy.

### 2.2.3 Verification of a machine simulator [2] [8] [9]

Gilad Israeli et al. presented a technique for the numerical verification of CNC machine simulations. In this field verification of the working of a machining simulator is done. The objective was to develop some tests to gain confidence that the intersection between the stock and the swept surface is implemented correctly. The simulator which is used for testing is ToolSim.

A CNC machine simulator is a complex piece of software. A machine model, a stock model, and a tool model must be designed, and algorithms for intersecting the partially machined stock with the surface swept by the tool must be implemented. Verification that the software is working correctly is critical if the simulator is used for industrial purposes. In this paper, we describe a method to test that the intersection of the tool and the surface swept by a 5-axis machine is correctly generated by a simulator, where the stock is represented as a height field. Further, these tests assist in determining the appropriate values for many of the internal parameters of the simulator.

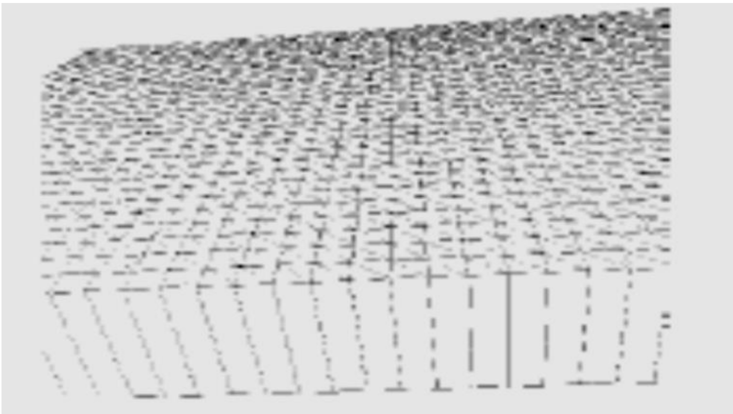


**Figure 2.8: Stock Representation with a Vertical Height Field [2]**

The architecture of ToolSim can be grouped into three major components: the stock, the machine, and the tool path. Together these components are used to compute the parts resulting from milling. The toolpath provides the machine positions used to create the part. The machine is responsible for computing the position and motion of the tool relative to the stock. Finally the stock is responsible for computing the intersection between the machine tool and the stock, as well as storing the result. There are a number of ways to represent the stock. The straightforward way is with a three dimensional matrix, with each entry representing a discreet volume. However there are many challenges to successfully implementing this representation. One problem is that the memory cost of this representation grows quickly with stock density and size. Other problems are that both the intersection of geometric shapes with volumes and the rendering of volumes are nontrivial.

Fortunately, most parts created by CNC machining (even by 5-axis machines) can be represented by a vertical height field on a regular grid. This height field is then tessellated into a surface of triangles to be rendered. Stock density is defined as the amount of height numbers per unit area of the grid. The density can be varied to achieve either faster or more accurate results. For CNC machining simulations, one needs to know the toolpath to simulate the machining. Abstractly, once we have the path, we compute the surface swept by the tool, and merge multiple passes of the tool to compute the machined surface. However, in real machines, the path is specified with g-codes, and the tool position is only specified at discrete locations. The question arises as to what path the tool travels in between these tool positions.

The paper described theoretical error bounds of piecewise linear approximation and showed how to use them to verify that stamping and swept surface intersections of a 5-axis CNC simulator are functioning correctly. Beyond just verifying that the error decreases as a factor of four as the stock density is doubled, we found that other parameters of the simulation also have to be increased to achieve this factor of four error reduction.

## 2.3 Conclusions drawn from Literature Survey

1. The STEP-NC Controller with OMI mentioned above has successfully extracted the machining and inspecting information from STEP-NC program file, tool path and probing path are also produced automatically. In the field of on machine inspection and simulation great effort is still needed to develop a machining module and inspecting module to realize real time machining and inspecting on milling machine tool in the following research work.
2. Adaptive NC simulation can help user to do error analysis and comparison between the cutting model and the original CAD model. It can also verify the exactness of NC codes before machining is carried out by a CNC machine in order to avoid wasting material and to improve machining accuracy. Better algorithms can be designed for faster calculation and better graphical display. The objective of this paper was to use the adaptive Voxel

model to develop a reliable multi-axis simulation procedure which can simulate the cutting route and the workpiece appearance during and after the simulation. It allows the user to do error analysis and comparison between the cutting model and the original CAD model. It can verify the accuracy of NC codes before machining on a CNC machine in order to avoid wasting material and to improve machining accuracy.

3. Numerical verification of the simulator is an essential requirement as to check whether the results displayed by the simulator are correct or not. CNC machine simulators are a useful tool for verifying tool paths without having to machine a test piece. However the simulator itself must be verified as functioning correctly for it to be of use in practice. The tests described were for a particular simulator, these tests should be applicable to other simulators that represent the stock as a height field. Other tests will need to be performed to fully verify the simulator. In particular, tests to verify that the machine has been correctly modeled. At a minimum, a comparison between the simulated surface and the machined surface need to be made. Further, when computing the error between the design surface and the simulated machined surface, a closest point error metric should be employed. However, the tests in this paper provide a way to verify one portion of the code, and should prove useful in locating coding errors and in determining the appropriate settings of the internal parameters of the simulator.

# CHAPTER 3
# PROBLEM DEFINITION

The general method used to predict the machining result is as shown in Figure 3.1. The process is iterative and consumes a lot of time and money.

First the part is modeled in a CAD package. Then the model is saved in a neutral file format and then the CAM module generates the toolpath for machining the part on a CNC machine.

The tool path generated is simulated in standard simulators to visualize the tool path only. A very high skilled operator is required for predicting any kind of accident during the simulation of tool path.

But the surface topography of the simulated surface and the designed surface is still not compared. Fixture and tool set up is done for performing the trial machining .The surface topography of the actual machined surface is measured and compared against the standard required topography.

If the actual machined surface topography is in machining tolerance region then the tool path is accepted and the part is brought to routine production. But if the deviation between the topography is not in tolerance region then refinement of the tool path is done again and the trial machining is repeated. This cycle repeats again and again till the requirements are met. Apparently this method is iterative and takes a lot of time.

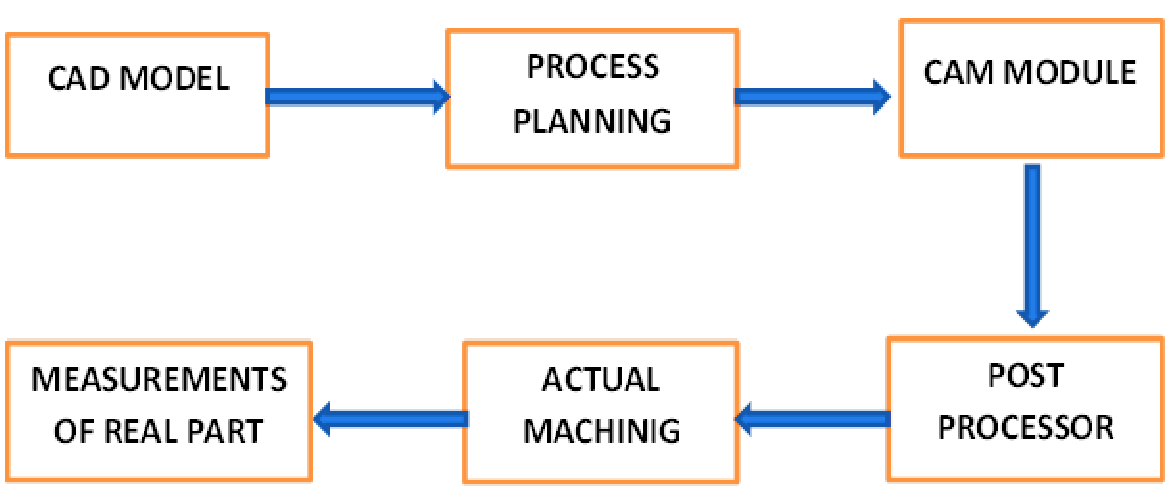



**Figure 3.1: The general method for predicting machining results**

**The literature survey has revealed a lot of other limitations of existing simulators which are as follow**:

- Present day Simulator's shows only machined path graphically without showing any deviation from required geometry.
- The view manipulation functions are not present in these simulators
- Real time feedback of surface topography is not available
- During CAM simulation there is no Comparison of surface topology of simulated surface with the Designed CAD Surface based upon STL geometry.
- Topology errors are found only from physical machined components.
- Iterative process for achieving the required topography.
- Losses in terms of cost involve, labor and time.

# CHAPTER 4
# SIMULATOR DESIGN

The simulator is designed for a 3-axis lathe-mill. The designed simulator takes the CAD geometry in the form of STL data, tool path data of tool location points in sequence as input for simulation and generates the machined surface as output data.

The simulator uses “mow the grass” method for simulating the machining on the lathe mill. The stock is represented by radial grasses projecting radially outwards from an axis.

The simulator reads the Tool Positions from the TP file and calculates the intersection with the grasses which lie inside a projected square. The dimension of the square is equal to the diameter of the tool and its center lies on the tool position. After each calculation the coordinates of the grasses are updated accordingly and a virtual machining scene is created.

The design of the simulator basically requires an algorithm that will incorporate both the machining simulation as well as the measurement of the error between the required and machined geometry. Also, for graphically viewing the simulation, graphical tools are also incorporated. Thus the desired objectives can be met and the gaps defined earlier can be overcome.

The simulator is developed using Microsoft Visual C++ 2010 Express edition which is integrated with OpenGL 4.1 libraries for graphics display.

## 4.1 Flowchart for proposed simulator

The proposed simulator is to be designed to follow the algorithm as shown in Figure 4.1.

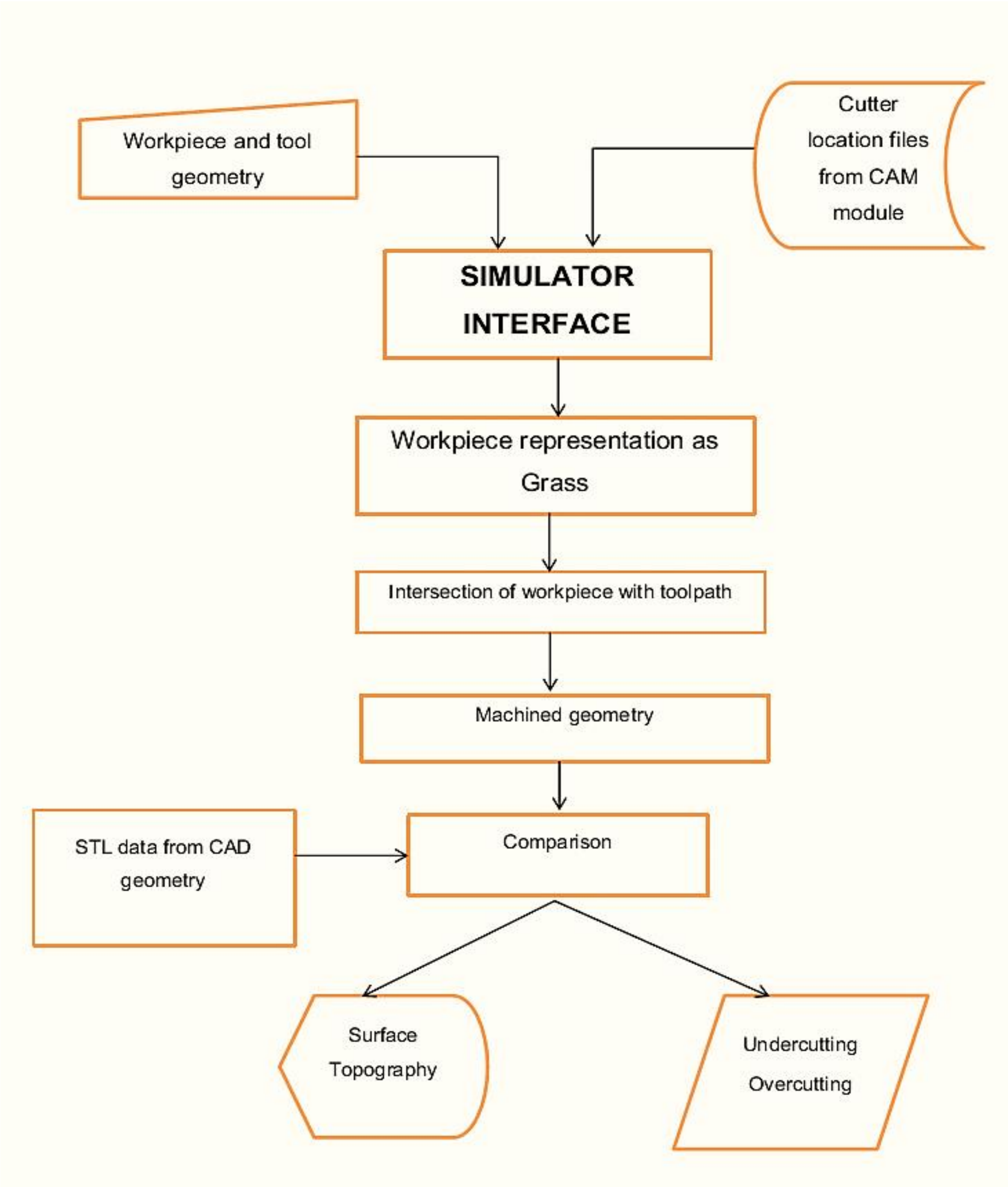



Figure 4.1: The algorithm for proposed simulator

## 4.2 Algorithm to calculate error between machined and actual geometry

The error between the machined part and actual CAD model is to be found. The error is to be calculated in the normal direction to the required surface. Now, from the simulator after finding all the intersection points of the tool tip and the grass, we obtain a point cloud data for the machined surface. A point in the point cloud can either be above or below the actual surface.

Depending on whether the point is above or below the STL geometry we come to know whether the material will be undercut or overcut. The distance between a point and STL geometry is found by calculating the shortest distance between a point and a Triangle. The Algorithm is as shown in following flowchart.

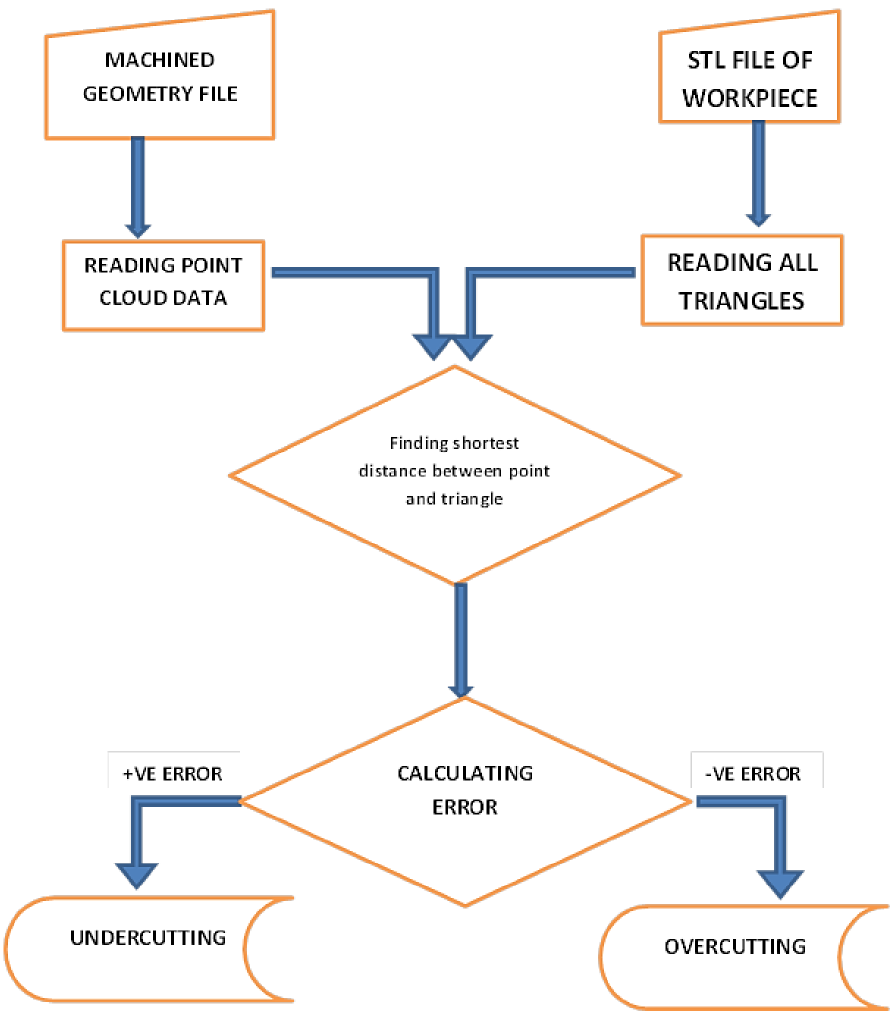



**Figure 4.2: Algorithm to calculate error between actual and machined geometry**

## 4.3 Simulator testing

### 4.3.1 Cylinder

Simulation results are carried out for some selected models. The toolpath for these models is obtained from a CAM package. The toolpath is helical and is generated for a ball nose cutter of diameter 3.175 mm i.e. 1/8$^{th}$ of an inch. A constant side step is taken which is equal to 0.4 mm i.e. 1/64$^{th}$ of an inch.

The first model considered for simulation is a cylinder of diameter 15 mm and length 10 mm. It is modeled in CREO 2.0 as shown in Figure 4.3.The axis of cylinder is along the X-axis.

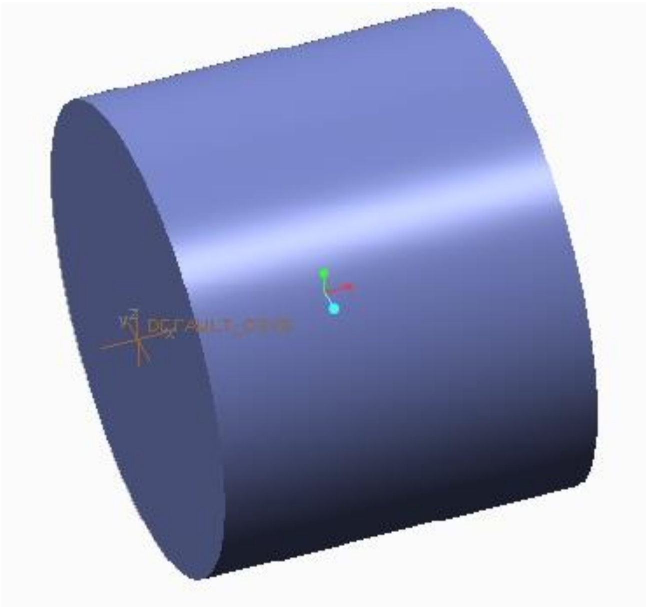


**Figure 4.3: Cylindrical part along X-axis Modeled in CREO 2.0**

The STL data of this cylinder is imported from a CAD package to the CAM package to obtain the toolpath file. The STL representation of the model is as shown in Figure 4.4. STL file can be stored in two formats i.e. ASCII and binary. The ASCII format is as following:

```
solid CYLINDER
  facet normal n1 n2 n3
    outer loop
      vertex v1   v2       v3
      vertex v1   v2       v3
      vertex v1   v2       v3
    endloop
  endfacet
……………
```

end Solid

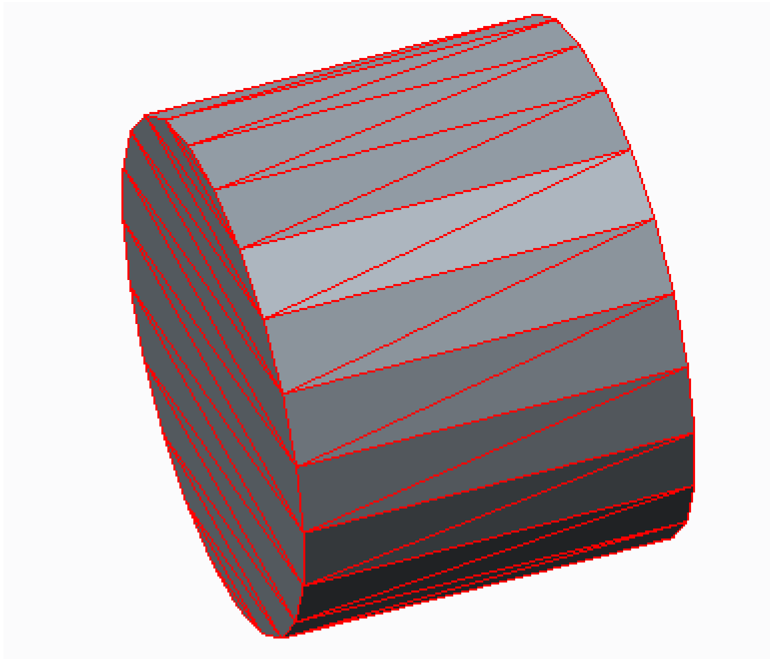
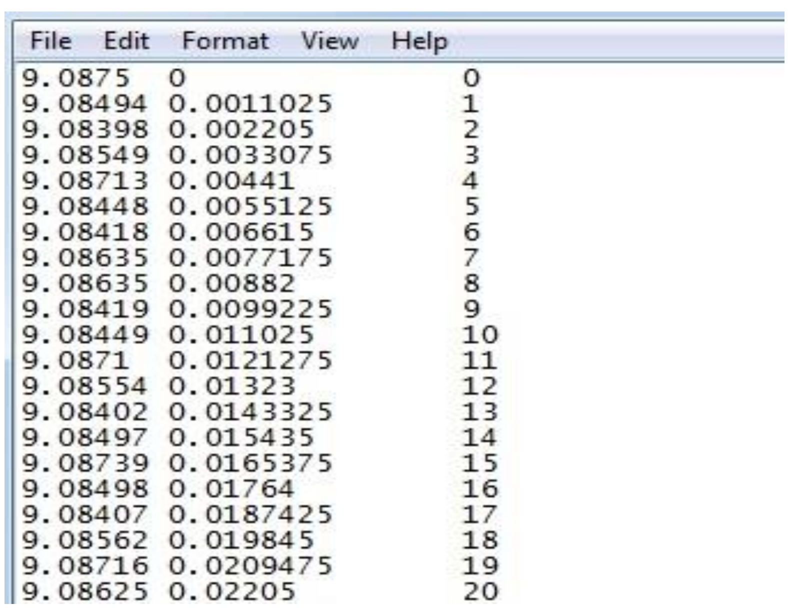


**Figure 4.4: STL representation of Cylindrical Part**

The toolpath file for simulator has the format as shown in Figure 4.5. The first column in the file is the radius of the part to be machined. The second column contains the values of coordinate along the axial direction. The third column has degree the workpiece rotates about the longitudinal axis.



File Edit Format View Help

| | | |
|---|---|---|
| 9.0875 | 0 | 0 |
| 9.08494 | 0.0011025 | 1 |
| 9.08398 | 0.002205 | 2 |
| 9.08549 | 0.0033075 | 3 |
| 9.08713 | 0.00441 | 4 |
| 9.08448 | 0.0055125 | 5 |
| 9.08418 | 0.006615 | 6 |
| 9.08635 | 0.0077175 | 7 |
| 9.08635 | 0.00882 | 8 |
| 9.08419 | 0.0099225 | 9 |
| 9.08449 | 0.011025 | 10 |
| 9.0871 | 0.0121275 | 11 |
| 9.08554 | 0.01323 | 12 |
| 9.08402 | 0.0143325 | 13 |
| 9.08497 | 0.015435 | 14 |
| 9.08739 | 0.0165375 | 15 |
| 9.08498 | 0.01764 | 16 |
| 9.08407 | 0.0187425 | 17 |
| 9.08562 | 0.019845 | 18 |
| 9.08716 | 0.0209475 | 19 |
| 9.08625 | 0.02205 | 20 |

**Figure 4.5: Tool path file used as an input for simulator**

Simulation of cylindrical part being machined at various stages is shown in the following figures. The STL geometry is also displayed for visually comparing with the machined geometry. The theoretical Ra value for this geometry is calculated for all the triangle of the STL. The Ra value ranges from about 10 microns to 200microns.

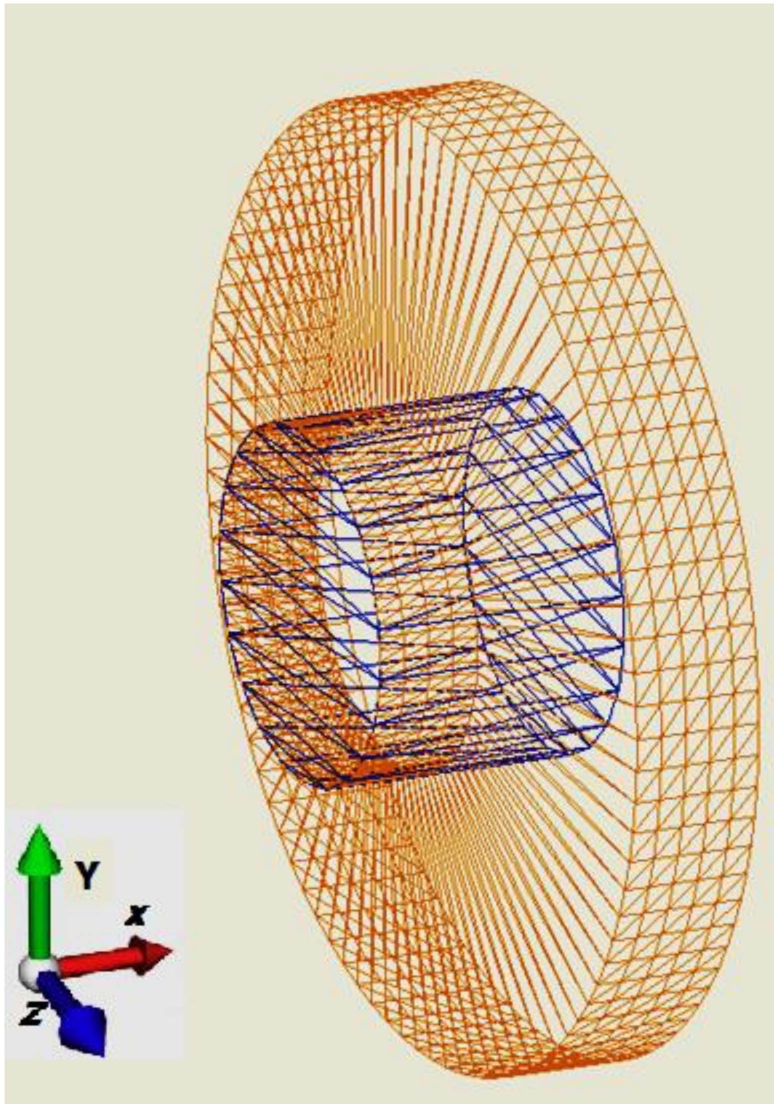



**Figure 4.6: Cylindrical part being machined with STL geometry displayed**

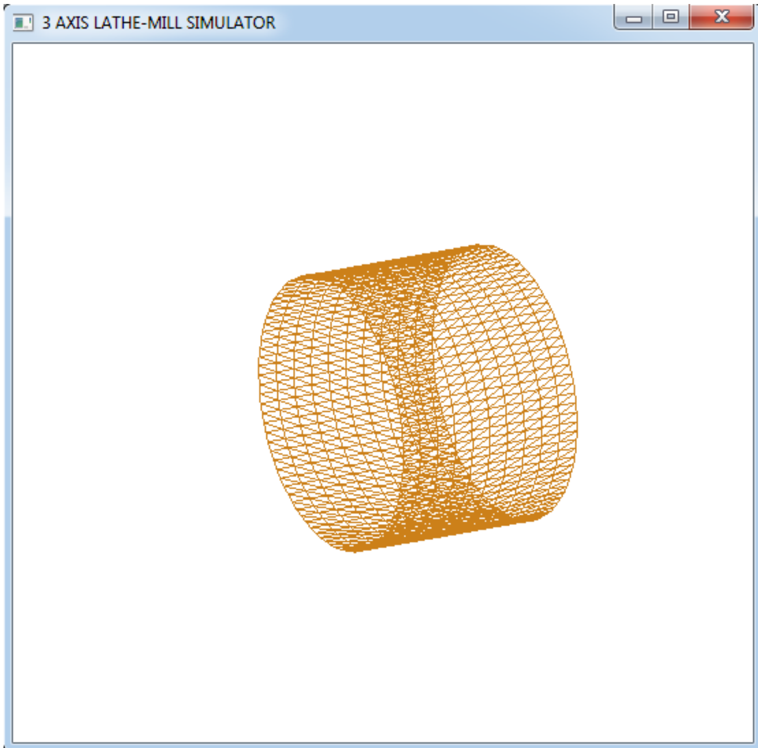



**Figure 4.7: Final machined part displayed without STL**

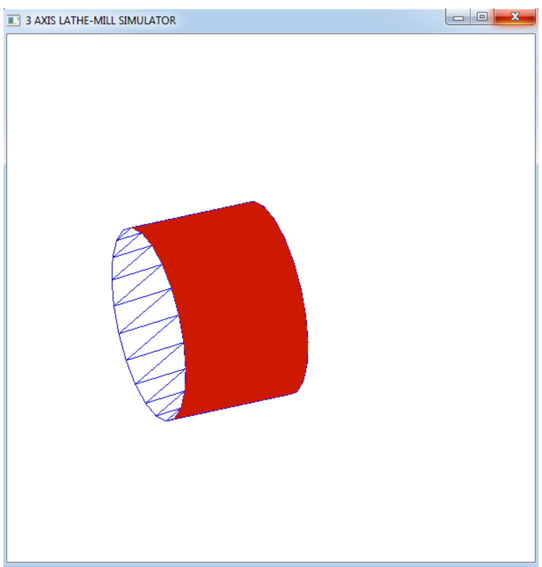



**Figure 4.8 Error displayed in Red color**

The error shown in the above Figure 4.8 in red color is for undercutting. As the whole cylinder has a red surface this shows that the whole cylinder will be undercut.

### 4.3.2 Variable inclination triangular facet part

The second model selected is having a triangular faceted part. First there is square cross section having 20 cm * 20 cm dimension which is 8 cm long. Then there are four triangular facets having variable inclination. Facet 1 has $\theta_1$ equal to 40$^o$ about y axis. Facet 2 has $\theta_2$ equal to 50$^o$ about z axis. Facet 3 has $\theta_3$ equal to 45$^o$ (CW) about y axis. Facet 4 has $\theta_1$ equal to 40$^o$ (CW) about z axis. It is blended to from a quadrilateral closed by two triangles boundary blend as shown in the Figure 4.9 .

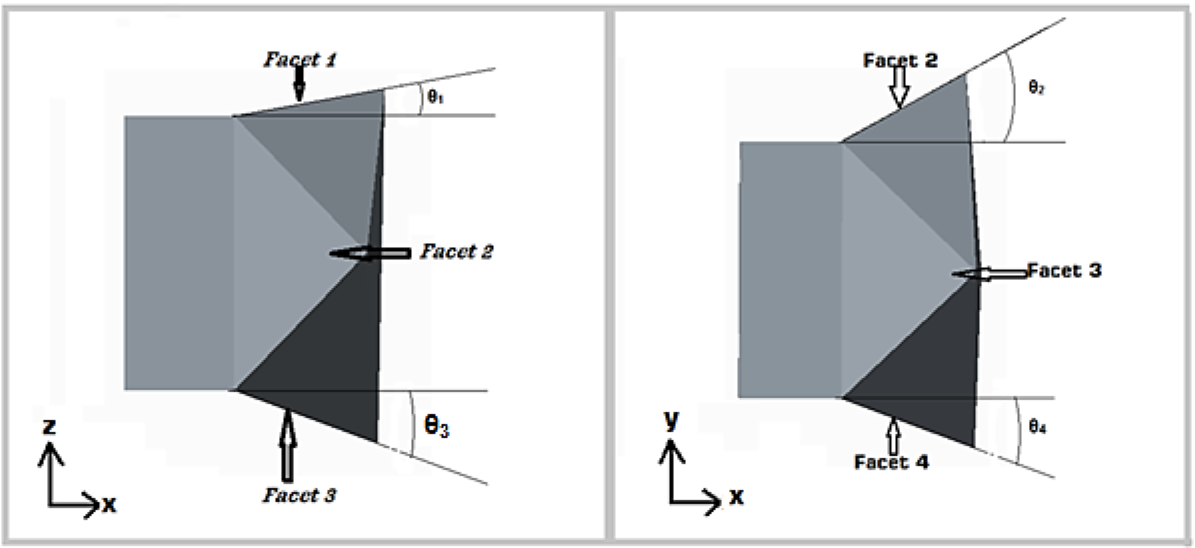



**Figure 4.9 The CAD Model of the part**

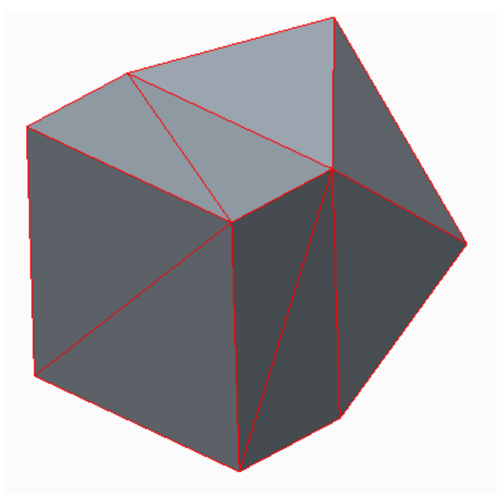


**Figure 4.10: STL representation of the part**

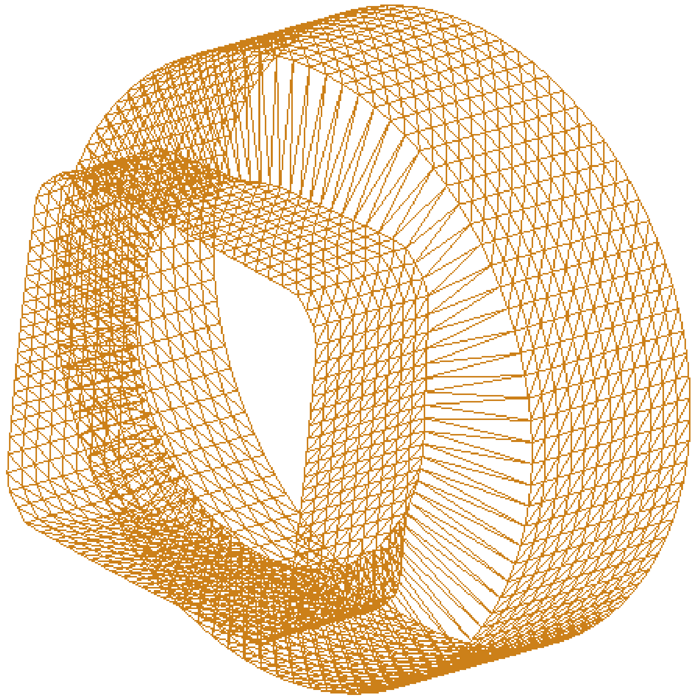


**Figure 4.11 Machining being done on the part**

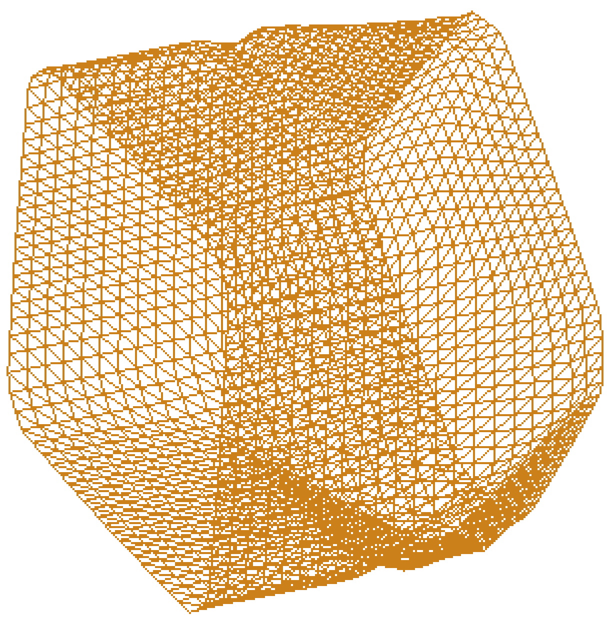


**Figure 4.12: Final machined part**

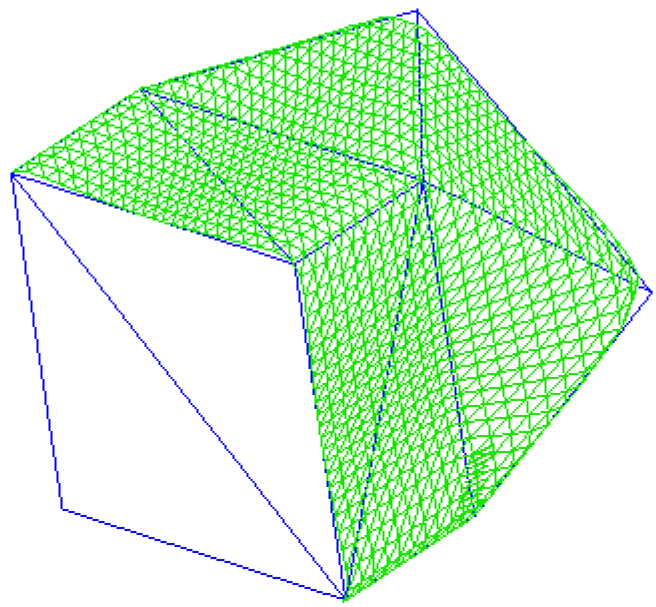


**Figure 4.13: Error displayed in green color**

The part is enveloped by a green triangulated surface. The green color shows that the envelope is above the required STL geometry. So the whole part will be overcut. The Ra value is calculated for all the triangles in the geometry. The theoretical Ra value ranges from 10micron to 200 micron.

# CHAPTER 5

# CONCLUSIONS

After performing some sample simulations on different models and visualizing the results of the simulator the achievements and the limitation of the proposed simulator is discussed in detail below.

1. The simulator successfully displays the machining being done.
2. Its results are consistent i.e. for same tool path result will be always similar.
3. It only requires Finish TP file for simulating the machining process.
4. The TP file needed has a very simple sequential XYZ points coordinates of cutter location.
5. The simulator has view manipulation functions like zoom, pan, rotate which can be used to visualize some specific portion of the geometry and the machined surface with error.
6. The simulator can show wrong machining which may be due to toolpath error.
7. It also displays the error between the required STL geometry and the actual obtained geometry.
8. The data of machined part surface can be saved in a file for further analysis.

The limitations of the proposed simulator are discussed below:

1. The proposed simulator uses the STL geometry for Toolpath generation and error calculation. The STL file itself contains some error, so some deviation from actual results is going to come.
2. It can simulate even a wrong toolpath, so it does not do error detection in toolpath. A user himself needs to compare the results with requirements.
3. It does not do toolpath refinement for compensating for the error between the actual and machined geometry.

## 5.2 FUTURE SCOPE OF WORK

1. Proposed simulator can be integrated with other modules and algorithms for predicting the other geometric errors measurement than contribute to machining inaccuracies.
2. Tool path file refinement module can also be integrated with the proposed simulator for achieving the desired topography.
3. Since this simulator is for three axis lathe milling, for higher axis machining and training centers new algorithm can be developed to incorporate the machining.
4. Speed of the simulator can also be increased by implementing the new advanced hardware & software/programming techniques.
5. Physical verification of simulation results needs to be done by machining standard geometry.
6. This simulator is using discrete tool position for cutting the ‘grass” while the actual tool motion between the points need to be taken for a continuous ‘tool path’ which can be approximated by a cylinder for small steps and further incorporate machine kinematics to get intermediate path

# REFERENCES


1. Balram Kalra, “A Thesis Report “Simulated Topographic Error Prediction for 3-axis Milling”, Thapar university, Patiala http://dspace.thapar.edu:8080/dspace/handle/10266/1598.
2. Gilad Israeli, Stephen Mann, Sanjeev Bedi and Ajayinder Singh Jawanda, “Numerical Verification of CNC Machine Simulations”, Computer aided design and Applications.
3. Hong T. Yau1, Lee S. Tsou2 and Yu C. Tong, “Adaptive NC Simulation for Multi-axis Solid Machining” ,Proceedings of Computer-Aided Design & Applications, Vol. 2, Nos. 1-4, pp 95-104(2005).
4. Craig S. Kaplan, Sanjeev Bedi, Stephen Mann, Gilad Israeli, Gilbert Poon, “A New Paradigm for Woodworking with NC Machines” http://www.cgl.uwaterloo.ca/~smann/Papers/burninator.pdf
5. Xiaoming Zhu, Yongzhang Wang and Hongya Fu. “A 3-D Simulation System for Milling Machining Based on STEP-NC”, Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China
6. Junzhe Tan, Chengrui Zhang, Xihui Liang, “Research on STEP-NC based Machining and on machine Inspecting Simulation System”, ieeexplore.ieee.org. E-ISBN : 978-1-4244-4507-3
7. Hong T. Yau1, Lee S. Tsou2 and Yu C. Tong, “Adaptive NC Simulation for Multi-axis Solid Machining” ,Proceedings of Computer-Aided Design & Applications, Vol. 2, Nos. 1-4, pp 95-104(2005)
8. Mann,S.;Bedi,S.;Israeli,G.; Zhou, X.: Machine models and tool motions for simulating five-axis machining,Computer-Aided Design, 42, 2010, 231-237.DOI:10.1016/j.cad.2009.11.005
9. Mann,S.;Bedi,S.: Generalization of the imprint method to general surfaces of revolution for NC machining, Computer Aided Design, 34(5), 2002, 373-378.DOI:10.1016/S0010-4485(01)00103-8

## WEB REFERENCES


1. http://pfiles1.ptc.com/WCMS/images/108442/108442en_image1_1268259795.png
2. http://www.moduleworks.com%2Fimages-controlled%2Fstudies-examples%2FEXAPT_Trichternase_1024.jpg
3. http://www.binaryspaces.de/images/content/008_kollision/018.jpg
4. http://2.bp.blogspot.com/-OYGH14G9wTA/ThVIhddsHmI/AAAAAAAAAAo/W3K4rCBfa_M/s1600/surfacetext
5. http://paulbourke.net/geometry/circlesphere/
6. http://en.wikipedia.org/wiki/File:STEP-NC_control_scania.jpg