**Design and development of Collaborative Robotics cell for 3-d printing**

*Project report submitted*

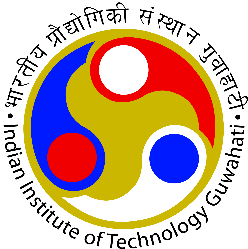
*in partial fulfilment of the requirement for the degree of*

**Bachelor of Technology**

By

Akash Kumar (210103011)

Arushi Madhesiya (210103023)



BTP Supervisor: Dr. Sajan Kapil

**Department of Mechanical Engineering**

**Indian Institute of Technology Guwahati**

**Guwahati – 781039, Assam, India**

**12th Nov. 2024**

# **Acknowledgement**

Firstly, I owe my heartfelt gratitude and deepest regard to my supervisor, Dr. Sajan Kapil, Department of Mechanical Engineering, IIT Guwahati, India, for his relentless guidance, constant support and full cooperation throughout the project. His methods, vision and enthusiasm has always been a constant motivation and inspiration to me. Without his supervision and direction, this project could not have been a success.

I am deeply indebted to IIT Guwahati for providing the necessary resources, facilities, and infrastructure essential for the successful execution of this project.

I am also grateful to my classmates and friends for their encouragement, assistance, and constructive feedback, which have greatly contributed to the refinement of this project.

Thank you to everyone who has played a part, directly or indirectly, in the completion of this project.

Akash Kumar (210103011)

Arushi Madhesiya (210103023)

# **Abstract**

# **Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Akash Kumar (210103011)

Arushi Madhesiya (210103022)

# **List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Caption** | **Page No.** |
| 1.1 | 3 D printing Machine | 12 |
| 1.2 | Robotics Cell for 3 D Printing | 14 |
| 2.1 | Powermill Software | 14 |
| 2.2 | Powershape Software | 15 |
| 2.3 | UltimakerCura |  |
| 2.4 | 3 D printing Process from CAD Model | 15 |
| 2.5 | 3 D printing Process from CAD Model | 15 |
| 2.6 | Slicing of CAD Model | 15 |
| 2.7 | Medial Axis Transform (MAT) of Different Contour | 15 |
| 2.8 | Use of Medial Axis Transform (MAT) Concept |  |
| 2.9 | Trochoidal Path of Tool Tip in 3 D Printing |  |
| 4.1 | CAD Model |  |

**Contents**

Chapter Page No.

[Acknowledgement ii](#_Toc182236260)

[Abstract iii](#_Toc182236261)

[Declaration iv](#_Toc182236262)

[List of Figures v](#_Toc182236263)

[1. INTRODUCTION 1](#_Toc182236264)

[**1.1** **Background** 1](#_Toc182236265)

[**1.2 Need of Project** 1](#_Toc182236266)

[**1.3 Motivation of the Study** 3](#_Toc182236267)

[**1.4 Objective of the present study** 3](#_Toc182236268)

[2. Semesters Work Review 5](#_Toc182236269)

[**2.1 Learnt PowerMill** 5](#_Toc182236270)

[**2.2 Learnt PowerShape** 5](#_Toc182236271)

[**2.3 Learnt Ultimaker Cura** 6](#_Toc182236272)

[6](#_Toc182236273)

[**2.4 Learnt Inkscape** 6](#_Toc182236274)

[**2.5 Learnt Visual Studio** 6](#_Toc182236275)

[**2.6 Learnt Python Tkinter** 7](#_Toc182236276)

[**2.7 Learnt Basics of 3D Printing** 7](#_Toc182236277)

[**2.8 Learnt Slicing of CAD model** 8](#_Toc182236278)

[**2.9 Learnt Tool path Planning** 9](#_Toc182236279)

[**2.10 Learnt Toolpath generation** 10](#_Toc182236280)

[3. Literature Review 14](#_Toc182236281)

[**3.1 Evolution of Collaborative Robotics in Manufacturing** 14](#_Toc182236282)

[**3.2 Current State of 3D Printing Technology** 14](#_Toc182236283)

[**3.3 Collaborative Robotics in 3D Printing** 15](#_Toc182236284)

[4. Methodology 16](#_Toc182236285)

[**4.1 Introduction** 16](#_Toc182236286)

[**4.2 Objectives** 16](#_Toc182236287)

[**4.3 Used Software Overview** 16](#_Toc182236288)

[**4.4 Methodology** 17](#_Toc182236289)

[4.4.1 Methodology for G-code Generation by Powermill 17](#_Toc182236290)

[4.4.2 Methodology for G-code Generation by UltimakerCura 18](#_Toc182236291)

[4.4.3 Methodology for G-code Generation by Inkscape 19](#_Toc182236292)

[4.4.4 Methodology for Getting Medial Axis Transform 20](#_Toc182236293)

[5. Results and Discussion 21](#_Toc182236294)

[**5.1 Result** 21](#_Toc182236295)

[5.1.1 Result of G-code Generation by Powermill 21](#_Toc182236296)

[5.1.2 Result of G-code Generation by UltimakerCura 22](#_Toc182236297)

[5.1.3 Result of G-code Generation by Inkscape 23](#_Toc182236298)

[5.1.1 Result of Medial Axis Transform (MAT) 23](#_Toc182236299)

[**5.2 Discussion** 23](#_Toc182236300)

[6. Conclusion & Future Scope 25](#_Toc182236301)

[**6.1 Conclusion** 25](#_Toc182236302)

[**6.2 Future Plans** 25](#_Toc182236303)

[References 27](#_Toc182236304)

CHAPTER 1

# **1. INTRODUCTION**

## **Background**

3D printing has emerged as a transformational technology in the fast-expanding manufacturing world, enabling new opportunities for design and production. This research focuses on the integration of collaborative robotics with 3D printing to improve manufacturing efficiency, flexibility, and precision. The combination of 3D printing and collaborative robotics has brought about a new wave of innovation in recent years, changing the way that design and manufacture are carried out. Collaborative robotics design and development for 3D printing offers new opportunities for customization, scalability, and real-time flexibility in addition to addressing conventional production difficulties.

Figure .1 (3 D Printing Machine) [[1]](#img1)

## **1.2 Need of Project**

- A collaborative robotics cell can be useful in the context of 3D printing for various reasons:

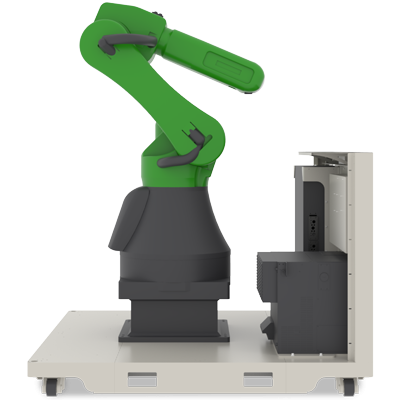


Figure 1.2 (Robotics Cell) [2]

1. **Increased Efficiency:** Collaborative robots can improve overall efficiency by working alongside human operators. They can help with things like loading and unloading materials, setting up print jobs, and post-processing. This collaborative method has the potential to result in a more efficient 3D printing operation.

2. **Precision and Consistency:** Robots can perform extremely accurate and reliable actions. Precision is essential in 3D printing for producing accurate and high-quality prints. Incorporating robots into the process can aid in maintaining a high degree of precision, resulting in improved overall print quality.

3. **Flexibility and Adaptability:** Collaborative robotics are designed to be simply programmed and adaptable to a variety of applications. In a 3D printing setting where jobs may vary depending on the type of print job, having a flexible robotic system enables for rapid reprogramming and reconfiguration to match changing requirements.

4. **Reduced Downtime:** robots that collaborate can work nonstop, cutting down on the downtime that comes with manual labor. Their ability to collaborate with human operators makes it possible for various activities in the 3D printing process to be completed smoothly. A more continuous and effective production cycle may result from this.

5. **Improved Safety:** Collaborative robots are designed to function securely alongside humans while offering no major danger of injury. In a 3D printing setting with hot surfaces, moving parts, or other potential hazards, cobots can help improve safety by taking on activities that would be dangerous for human operators.

6. **Production Capability:** Collaborative robots can operate continuously, enabling for 24-hour manufacturing. This is especially useful in businesses that require rapid production and turnaround periods.

## **1.3 Motivation of the Study**

Based on the literature review it is observed that "The implementation of collaborative robotics within the realm of 3D printing represents a pivotal advancement at the intersection of technology and manufacturing. The motivation behind this project stems from a desire to revolutionize the existing landscape of 3D printing by integrating collaborative robotic systems. Based on the literature review, it is observed that the current studies predominantly focus on the theoretical and isolated applications of collaborative robotics in manufacturing. However, these studies primarily devote their attention to studying specific aspects such as individual robotic movements, safety protocols, or efficiency enhancements. The existing body of research underscores the potential of collaborative robotics in optimizing singular elements within manufacturing processes.

## **1.4 Objective of the present study**

Create a collaborative robotics cell that allows robots to perform 3D printing tasks simultaneously. To maximize robot movement and minimize manufacturing time, incorporate cutting-edge path planning algorithms.

1. **Efficiency Improvement:** Enhance the efficiency of 3D printing processes through automation and robotics, aiming to reduce production time and costs.
2. **Quality Enhancement:** Develop a system that ensures consistent quality in 3D printed objects by integrating robotics for precise control and monitoring of the printing process.
3. **Scalability:** Design a robotics cell that can be scaled up or down based on the requirements of different 3D printing projects, making it adaptable to various industries and applications.
4. **Integration of Technologies:** Explore the integration of different technologies such as robotics, artificial intelligence, and advanced sensors to create an intelligent and versatile 3D printing system.
5. **Customization and Flexibility:** Enable the robotics cell to handle a wide range of materials and geometries, allowing for customized and flexible manufacturing solutions.
6. **Safety and Reliability:** Ensure the safety of operators and reliability of the system by implementing robust safety features and fail-safe mechanisms.
7. **User Interface and Control:** Develop an intuitive user interface for controlling and monitoring the robotics cell, allowing users to easily set up, manage, and track 3D printing jobs.
8. **Environmental Sustainability:** Consider the environmental impact of the 3D printing process and aim to minimize waste generation and energy consumption through optimized robotics control and material usage.
9. **Research and Innovation:** Contribute to the advancement of 3D printing technology through research on novel robotics algorithms, materials, or printing techniques that can further enhance the capabilities of the system.
10. **Demonstration and Validation:** Build a prototype of the robotics cell and conduct thorough testing and validation to demonstrate its feasibility, performance, and potential benefits in real-world manufacturing environments.

CHAPTER 2

# **2. Semesters Work Review**

## A computer screen shot of a machine Description automatically generated**2.1 Learnt PowerMill**

We learnt PowerMill, a powerful software used for programming CNC (Computer Numerical Control) machines. we gained proficiency in creating toolpaths, optimizing machining strategies, and simulating milling operations. This experience equipped us with valuable skills in CAM (Computer-Aided Manufacturing) that are essential for designing and machining complex parts efficiently.

Figure .1 [3]

## **2.2 Learnt PowerShape**

A computer screen shot of a computer program

Description automatically generatedIn addition to PowerMill, we also delved into PowerShape, a CAD (Computer-Aided Design) software widely utilized in the manufacturing industry. Through tutorials, we acquired knowledge of creating and modifying 3D models, preparing geometry for machining, and integrating CAD designs with CAM processes.

Figure 2.2 [4]

## **2.3 Learnt Ultimaker Cura**

Ulti maker Cura is a powerful, open- source slicing software widely used in 3D printing to generate G-code, the machine language that drives 3D printers. It translates a 3D model into a set of instructions that the printer follows to build the object layer by layer.

## 

Figure 2.3 [5]

## **2.4 Learnt Inkscape**

Inkscape is a popular open-source vector graphics editor that is widely used for creating and editing vector images. It can be particularly useful in the context of printing, including applications like 3D printing and traditional 2D printing. Inkscape provides a range of tools for designing and preparing vector graphics for various types of printing. Its vector-based approach ensures that designs are scalable and maintain their quality across different sizes and resolutions.

## **2.5 Learnt Visual Studio**

We learnt Visual Studio, a comprehensive integrated development environment (IDE) for software development. Through guided tutorials and practical coding exercises, we familiarized ourselves with the features and functionalities of Visual Studio, including code editing, debugging, version control, and project management. This was needed for making some useful extensions in the PowerMill and PowerShape.

## **2.6 Learnt Python Tkinter**

We learnt Python Tkinter, a standard GUI (Graphical User Interface) toolkit for Python programming. Through tutorials, we gained proficiency in designing and developing user-friendly interfaces using Tkinter's widgets and layout managers. This would be helpful for creating custom graphical interfaces for our extensions.

## **2.7 Learnt Basics of 3D Printing**



Figure 2.4 [6]

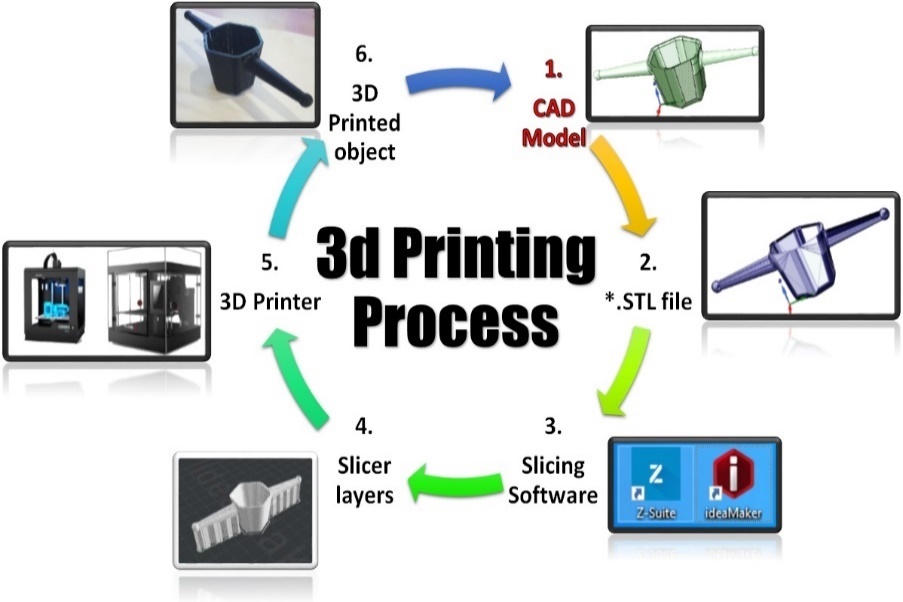
We Learned Visual Studio, a comprehensive integrated development environment (IDE) for software development. Through guided tutorials and practical coding exercises, we familiarized ourselves with the features and functionalities of Visual Studio, including code editing, debugging, version control, and project management. This newfound proficiency in Visual Studio empowered us to write, debug, and deploy software applications efficiently, laying a strong foundation for our journey in software development.

Figure 2.5 [7]

## **2.8 Learnt Slicing of CAD model**

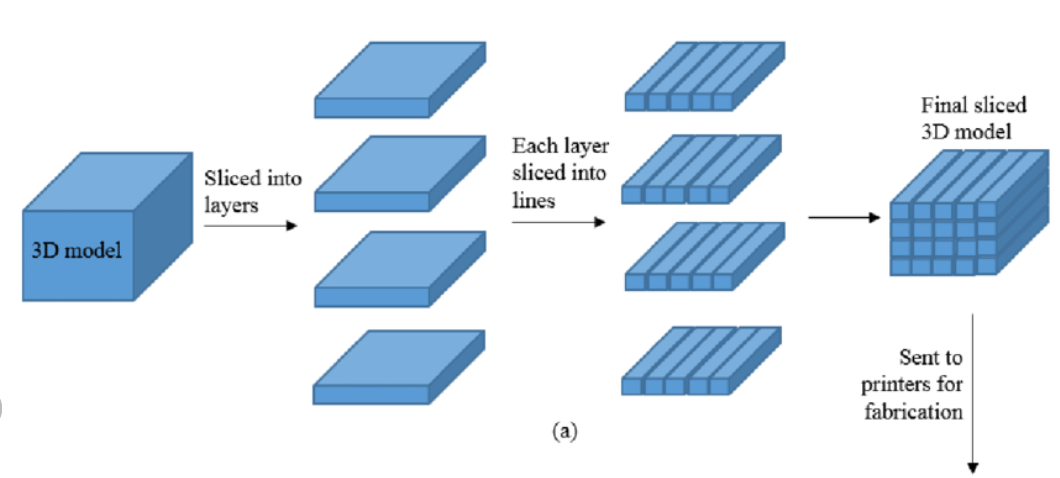


Figure 2.6 [8]

Slicing of CAD models is a fundamental concept in computer-aided design (CAD) and additive manufacturing (AM) processes such as 3D printing. Slicing involves dividing a 3D model into a series of 2D layers or cross-sections that can be printed layer by layer.

**breakdown of the concept:**

**Preparation of CAD Model:** Before slicing, a 3D model is created using CAD software. This model can be designed from scratch or generated from 3D scans or other sources.

**Slicing Software:** Specialized slicing software is used to prepare the CAD model for 3D printing. This software takes into account various factors such as printer specifications, material properties, and desired print quality.

**Layer Thickness:** The user specifies the layer thickness, which determines the height of each individual layer that will be printed. Thinner layers result in smoother surfaces but may increase printing time.

**Slicing Process:** The slicing software analyzes the CAD model and generates a series of horizontal cross-sections (slices) based on the specified layer thickness. Each slice represents a single layer of the final object.

## **2.9 Learnt Tool path Planning**

Toolpath planning for 3D printing involves generating precise instructions for the movement of the printer's nozzle or laser to deposit material layer by layer, ultimately fabricating the desired object.

**Overview of the process:**

**Toolpath Generation:** Toolpath planning involves determining the optimal path for the printer's nozzle or laser to follow when depositing material for each layer. Several factors influence toolpath generation:

**Layer Height:** The layer height determines the thickness of each layer and affects the resolution and surface finish of the printed object. Smaller layer heights result in finer details but increase print time.

**Infill Pattern:** Toolpath planning includes generating paths for infill, which refers to the internal structure of the object. Different infill patterns (e.g., grid, honeycomb) offer varying strength and material usage characteristics.

**Outer Perimeter:** The outer perimeter of each layer is printed first to define the shape of the object. Toolpath planning ensures precise deposition along the perimeter to achieve dimensional accuracy.

**Inner Fills:** After printing the outer perimeter, toolpath planning involves filling the interior of each layer with material according to the specified infill pattern. The toolpath must optimize material usage while maintaining structural integrity.

**Support Structures:** In cases where overhangs or unsupported features exist, toolpath planning includes generating support structures to provide stability during printing. These structures are later removed post-printing.

**Bridging:** Toolpath planning also considers bridging, which involves printing material across gaps between support structures or overhanging features. Proper toolpath generation ensures strong, continuous bridges without sagging or drooping.

## **2.10 Learnt Toolpath generation**

**G-Code Generation:** Once the toolpaths are generated, they are translated into G-code instructions, which provide precise movement commands for the 3D printer's motors. This G-code file controls the printer throughout the printing process.

**Simulation and Preview:** Before printing, it's often beneficial to simulate the toolpaths and preview the print to identify potential issues such as collisions, overhangs, or print failures. This allows adjustments to be made to the toolpaths, if necessary, before actual printing begins. Toolpath generation for 3D printing plays a critical role in achieving accurate, high-quality prints while optimizing print time, material usage, and structural integrity. Advanced slicing software continues to evolve, offering more sophisticated algorithms and customization options to meet the diverse needs of additive manufacturing applications.

**2.11 Learnt Medial Axis Transform (MAT)**

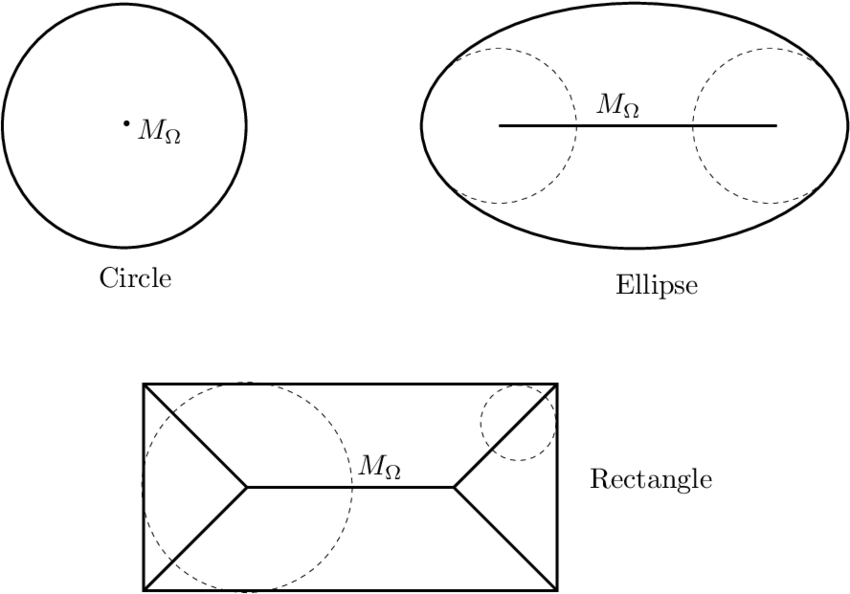
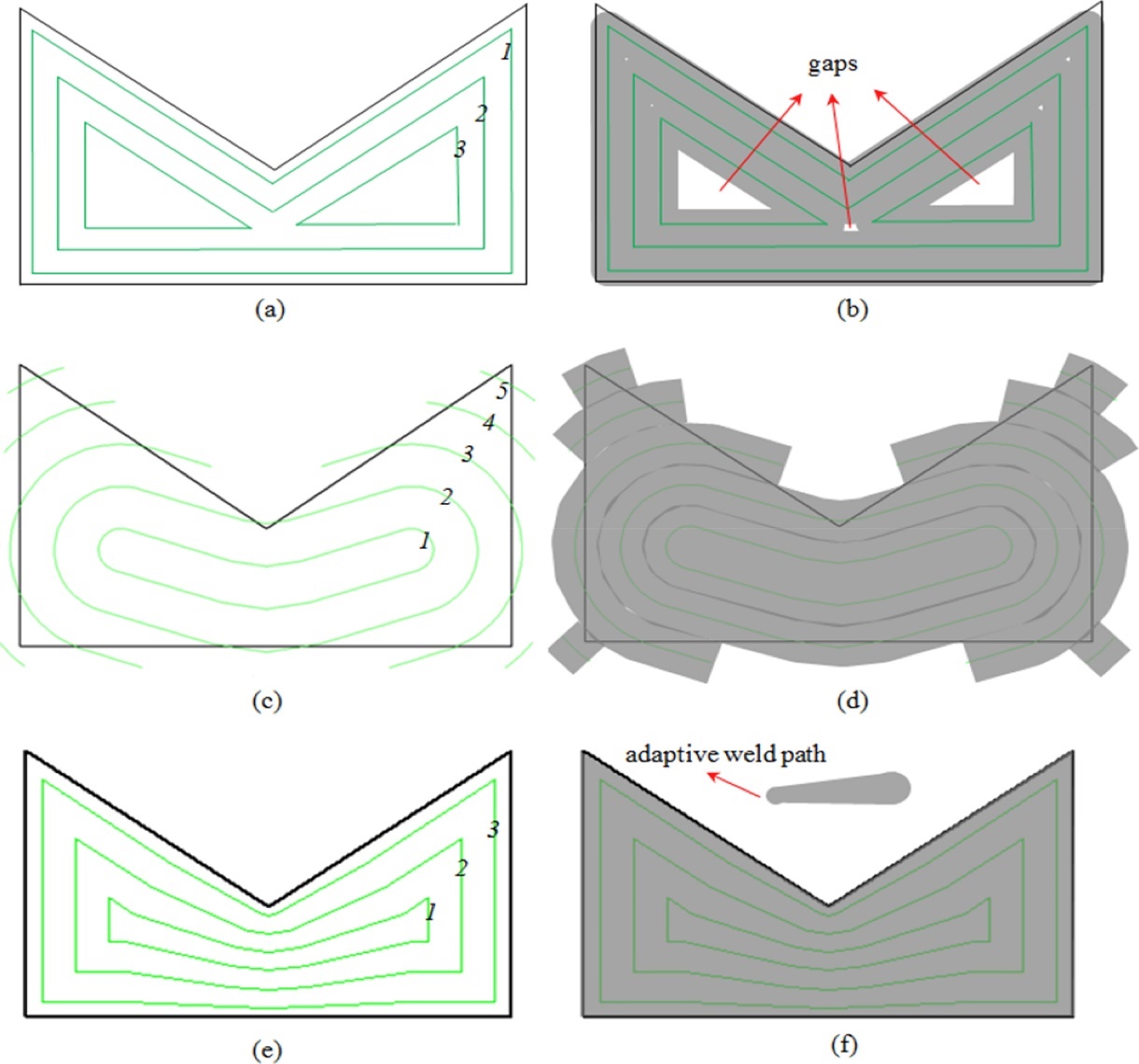
****

Figure 2.7 [9]



*Figure 2.8. Illustrations of different deposition paths. Black lines represent the boundary of the geometry; green lines represent the deposition paths with the numbers representing the order of the deposition paths; grey regions are deposited area by the relevant paths. (a) Contour path patterns; (b) The predicted high accuracy deposition but with internal gaps; (c) MAT path patterns (**[Ding et al., 2015c](https://www.sciencedirect.com/science/article/pii/S0959652616307119" \l "bib8)); (d) The predicted void-free deposition but with extra material deposited along the boundary; (e) Adaptive MAT path patterns with varying step-over distance; (f) The predicted void-free deposition with high accuracy at the boundary through using adaptive MAT path. (**[Ding et al., 2016b](https://www.sciencedirect.com/science/article/pii/S0959652616307119" \l "bib10)). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)* [9]

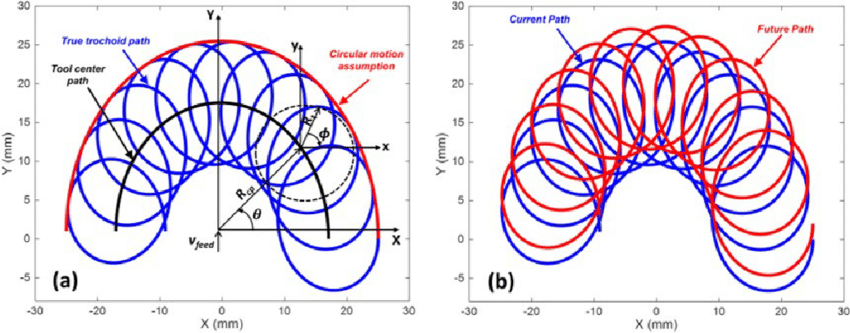
In 3D printing, the Medial Axis Transform (MAT) plays a critical role in optimizing the geometry of an object by identifying its "medial axis" – a set of central points that represent the object’s structure while retaining the overall shape. The MAT is often used to simplify complex shapes and optimize paths for printing by reducing redundant moves and providing structural integrity to the printed parts.

The MAT provides a mathematical representation that converts a shape into a series of skeletal lines or points, equidistant from the shape's boundaries. For a given 2D contour or 3D region, the MAT effectively represents the object's internal structure in a way that preserves its overall topology, allowing for efficient analysis and manipulation.

For our BTech project, we delved into the concept of MAT and developed an algorithm capable of determining the coordinates of the medial axis for any given contour. This algorithm efficiently calculates the medial axis by iteratively sampling points along the boundary of the region and finding equidistant points within the contour. Once we computed the medial axis coordinates, we utilized them to streamline tool paths, enhancing the printing process by minimizing excess movement and ensuring even material distribution.

Our exploration of MAT has underscored its value in 3D printing, where it facilitates intricate designs with minimal material use and increased structural integrity. By applying MAT in pre-processing stages, we’ve enabled more efficient slicing and tool path planning, leading to faster and more accurate print times and supporting the use of diverse materials and complex geometries.

**2.12 Learnt Trochoidal Path**

****

*Figure 2.9 (a) True trochoidal path of the tool tip and comparison of the outer manifold with conventional trochoidal milling; (b) tool tip path of the current and future trochoid.*[10]

In the context of 3D printing, the trochoidal path is an advanced toolpath strategy used to enhance print quality, reduce wear on the printer, and improve overall efficiency. Trochoidal paths involve a series of circular or arc-like motions that overlap to create a smooth and continuous movement pattern. Originally used in high-speed machining, trochoidal tool paths have found application in 3D printing due to their ability to manage tool loads, reduce heat buildup, and facilitate more stable material deposition, particularly in complex or high-density regions.

For our BTech project, we explored the concept of trochoidal paths as a way to improve material deposition strategies in 3D printing. Traditional tool paths often involve linear or raster movements, which can lead to sudden changes in speed and direction, potentially causing irregularities or defects in the printed part. Trochoidal paths, by contrast, create a more fluid motion, allowing for smoother transitions and less abrupt changes in velocity. This is especially beneficial when printing intricate designs or dense areas, as the constant circular movement reduces stress on the printer's extruder and minimizes the risk of clogging or overheating.

In the course of this project, I developed code to implement a trochoidal path for specific 3D printed regions. This path-planning algorithm calculates circular trajectories that intersect at optimal intervals to ensure even material deposition. By doing so, the trochoidal path allows for better material flow, greater control over the extrusion process, and a reduction in print times for complex geometries. Additionally, the continuous flow inherent in trochoidal paths enhances the stability of tall or intricate prints, leading to stronger, more reliable final products.

This experience with trochoidal path generation has provided insights into advanced 3D printing techniques, emphasizing how strategic tool paths can significantly impact the efficiency, durability, and quality of printed components. Implementing trochoidal paths has shown promising results in our tests, setting a foundation for further exploration of optimized tool path strategies in additive manufacturing.

CHAPTER 3

# **3. Literature Review**

## **3.1 Evolution of Collaborative Robotics in Manufacturing**

The evolution of robotics in manufacturing represents a fascinating journey, marked by significant milestones and transformative advancements, particularly with the paradigm shift towards collaborative robotics.

**Early Robotics in Manufacturing:**

The inception of robotics in manufacturing can be traced back to the mid-20th century. Industrial robots were initially introduced for repetitive, high-precision tasks in automotive assembly lines. Unimate, developed by George Devol and Joseph Engelberger in the early 1960s, became the first digitally operated and programmable robot, revolutionizing the manufacturing landscape.

**Shift Towards Collaborative Robotics:**

The significant shift towards collaborative robotics began in the early 21st century. This transition was fueled by advancements in sensor technology, artificial intelligence, and the need for more flexible and adaptive manufacturing systems.

## **3.2 Current State of 3D Printing Technology**

The current landscape of 3D printing, also known as additive manufacturing, is both diverse and dynamic, with a range of applications, materials, and significant impacts across various industries.

**Prototyping and Product Development:**

3D printing is extensively used in the rapid prototyping of new designs and product concepts across industries like automotive, aerospace, and consumer goods. It allows for quick iterations and testing of designs.

**Customized Manufacturing:**

It enables the creation of customized, one-of-a-kind products, from personalized medical devices and dental implants to bespoke fashion and consumer goods.

## **3.3 Collaborative Robotics in 3D Printing**

The integration of collaborative robots (cobots) in 3D printing processes has introduced several advancements in manufacturing and design.

**Enhanced Precision and Flexibility:**

These robots can precisely control the deposition of materials, leading to improved accuracy and consistency in the final printed objects. Their flexibility allows for intricate designs and the creation of complex geometries that might be challenging for traditional manufacturing methods.

**In-Situ Finishing and Post-Processing:**

Another advancement involves the integration of post-processing capabilities within the printing process itself. Collaborative robots can perform in-situ finishing tasks, such as polishing, sanding, or painting, immediately after printing, reducing the need for manual intervention and speeding up the overall production time.

CHAPTER 4

# **4. Methodology**

## **4.1 Introduction**

3D printing has become a cornerstone of modern manufacturing, enabling the production of complex geometries with precision and efficiency.

Generating G-Code, the machine-readable language that controls 3D printers, is crucial for translating CAD designs into physical objects.

## **4.2 Objectives**

To get a systematic methodology for generating G-Code for 3D printing using **PowerShape**, **PowerMill**, **UltimakerCura** and **Inkscape** Software.

To ensure accuracy, reliability, and efficiency in the G-Code generation process.

## **4.3 Used Software Overview**

**PowerShape:** Renowned for its robust surface modeling capabilities, PowerShape is well-suited for creating intricate 3D designs.

**PowerMill:** A CAM software proficient in generating toolpaths and G-Code for various manufacturing processes, including 3D printing.

**UltimakerCura:** Ulti maker Cura is a powerful, open- source slicing software widely used in 3D printing to generate G-code, the machine language that drives 3D printers.

**Inkscape:** Inkscape is a popular open-source vector graphics editor that is widely used for creating and editing vector images.

## **4.4 Methodology**

### **4.4.1 Methodology for G-code Generation by Powermill**

**Step 1: CAD Modeling**

Design the desired object using PowerShape, focusing on precision and manufacturability.

Ensure the CAD model is watertight and free from errors that could affect the printing process.

**Step 2: Export CAD Model**

Export the finalized CAD model in a format compatible with PowerMill, such as STL or OBJ.

A metal piece with screws

Description automatically generated

*Figure 4.1 CAD Model*

**Step 3: Import into PowerMill**

Launch PowerMill and import the CAD model into the software environment.

Verify the imported model to ensure its integrity and suitability for 3D printing.

**Step 4: Toolpath Generation**

Configure the printing parameters, including layer thickness, print speed, and material type.

Utilize PowerMill's advanced toolpath generation features to create efficient toolpaths tailored to the specific 3D printer.

**Step 5: Simulation and Verification**

Simulate the toolpaths to visualize the printing process and identify any potential issues, such as collisions or inadequate support structures. Verify the toolpaths for accuracy and make necessary adjustments to optimize the printing process.

**Step 6: Post-Processing and G-Code Generation**

Finalize the toolpaths and proceed to generate G-Code for the 3D printer.

PowerMill offers customizable options for G-Code generation, allowing users to fine-tune parameters according to printer specifications and material properties.

### **4.4.2 Methodology for G-code Generation by UltimakerCura**

**Step 1: Prepare the CAD Model**

I downloaded the cad model from online source.

I ensured that the model was fully enclosed (watertight) and ready for slicing.

**Step 2: Open Ultimaker Cura and Import the CAD Model**

I launched Ultimaker Cura and selected File Open File(s) to import the STL file of the CAD model.

After importing, the model appeared on the Cura build plate, showing its orientation and position.

**Step 3: Adjust the Model’s Position, Scale, and Orientation**

I adjusted the model to lay flat on the build plate for stability.

Using the Scale tool, I resized the model if necessary to fit the printer’s build area.

I used the Rotate tool to orient the model to minimize the need for support structures.

**Step 4: Configure Print Settings**

I selected the correct printer profile in Cura to ensure compatibility with the 3D printer.

I set the Layer Height to control the print quality (e.g., 0.2 mm for a balance of quality and speed).

Infill Density was set to around 20% for adequate strength without excessive print time.

I adjusted the Print Speed based on the material, aiming for good surface quality.

For overhangs, I enabled Supports and selected a suitable density and pattern to support those areas during printing.

I enabled Adhesion (using a brim or raft) to help the model stick to the bed and avoid warping.

**Step 5: Preview and Slice the Model**

I clicked on the \*Preview\* tab to inspect each layer of the sliced model and ensure there were no issues with the infill, supports, or layer alignment.

After reviewing the preview, I clicked \*Slice\* to convert the model to G-code. Cura displayed an estimated print time and material usage.

**Step 6: Save the G-code**

Once sliced, I saved the G-code file to my computer or directly to a USB/SD card, depending on the 3D printer requirements.

### **4.4.3 Methodology for G-code Generation by Inkscape**

**Step 1: Importing and Preparing the 2D Image**

I started by importing a 2D image (in PNG/JPG format) into Inkscape by selecting File Import.

After positioning the image on the canvas, I adjusted its scale and orientation to match the intended output size for G-code generation.

**Step 2: Converting the Image to a Vector Path**

To create a vector outline from the image, I used the Trace Bitmap tool under Path Trace Bitmap.

I selected settings appropriate for the image (such as edge detection) to capture only the outlines of the design, which would be used to generate the G-code path.

Once the tracing process was complete, I removed the original image, leaving only the vector path.

**Step 3: Optimizing the Path for G-code Generation**

I reviewed and edited the vector path to remove any unnecessary nodes or details that could complicate the G-code generation.

I used Inkscape’s node-editing tools to clean up the path, ensuring smoother transitions and optimized linework, which would result in efficient G-code output.

**Step 4: Generating G-code using the G-code Extension**

After finalizing the path, I accessed the G-code extension in Inkscape.

I configured the tool settings for my 2D drawing, such as specifying the desired units, scaling, and tool parameters (e.g., tool diameter and feed rate).

Finally, I ran the G-code export process, which generated a G-code file based on the vector path.

**Step 5: Saving the G-code**

The G-code file was saved in the appropriate format and location, ready for use in a compatible machine or further editing if needed.

### **4.4.4 Methodology for Getting Medial Axis Transform**

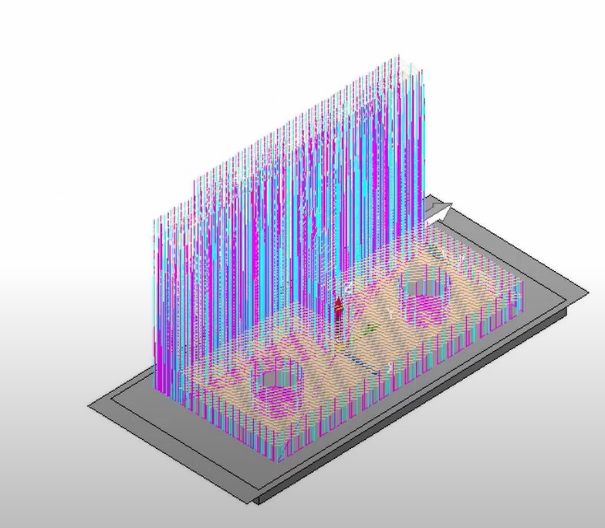
CHAPTER 5

# **5. Results and Discussion**

## **5.1 Result**

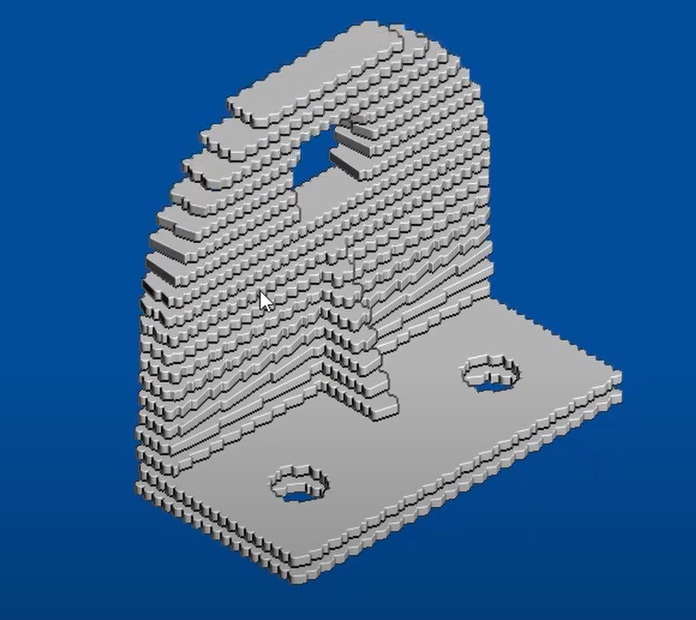
### **5.1.1 Result of G-code Generation by Powermill**

A metal piece with screws

Description automatically generated 

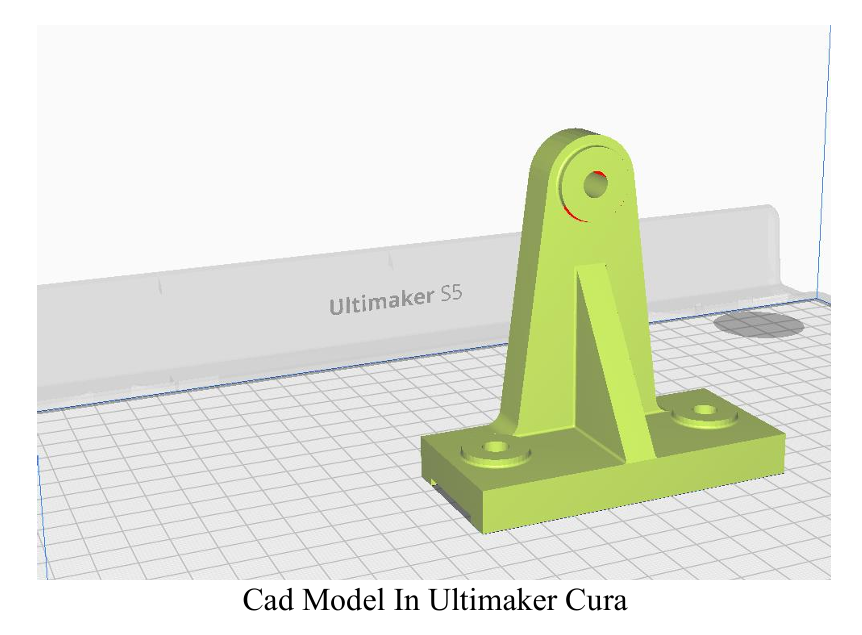
*Figure 5.2*

*Figure 5.1*

****

*Figure 5.3*

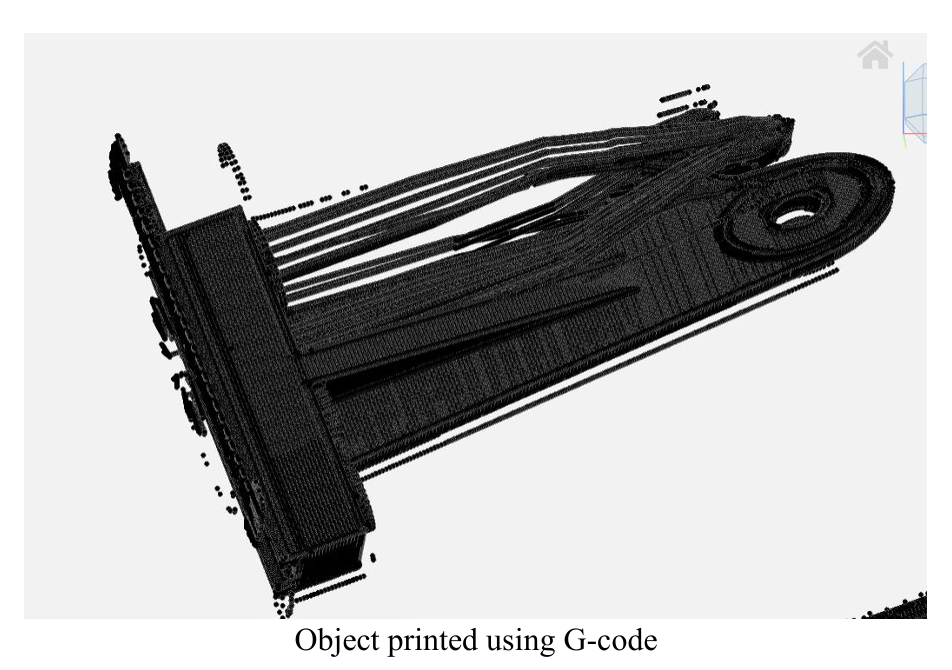
### **5.1.2 Result of G-code Generation by UltimakerCura**

A metal piece with screws

Description automatically generated

*Figure 5.5*

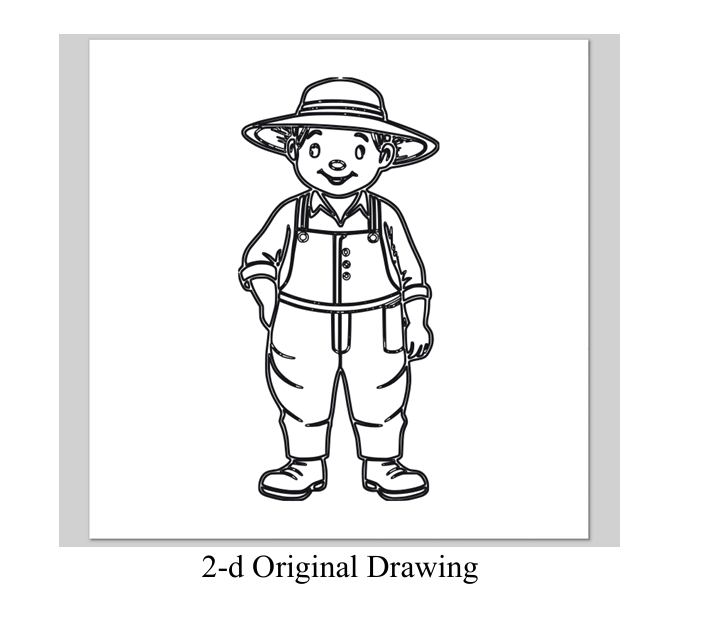
*Figure 5.4*

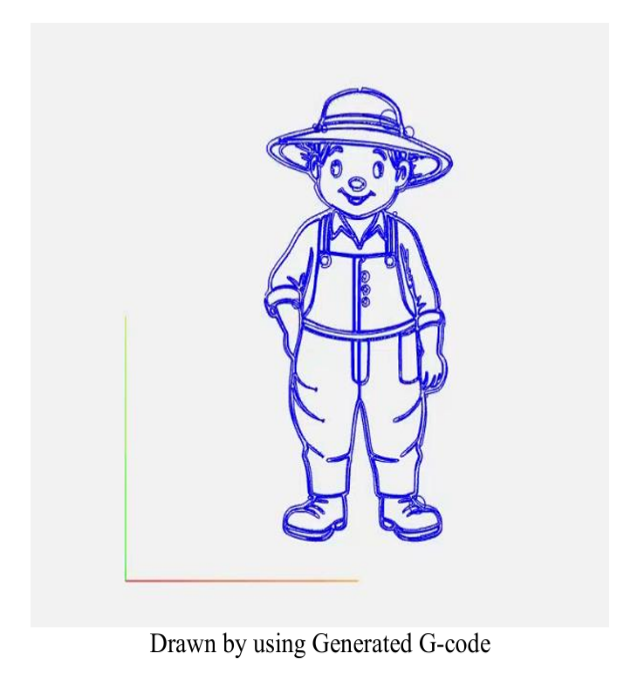


*Figure 5.6*

Printed by Ultimaker Cura Software

### **5.1.3 Result of G-code Generation by Inkscape**



*Figure 5.7*

*Figure 5.6*

\\\\\\\\\\\\\\\

Drawn by Inkscape Software

### **5.1.4 Result of Medial Axis Transform (MAT)**

## **5.2 Discussion**

Generating G-code from a CAD model using PowerMill is a critical process in modern manufacturing, enabling the translation of digital designs into machine-readable instructions for CNC (Computer Numerical Control) machines.

**Here’s a discussion on this process:**

1. PowerMill offers advanced algorithms for generating toolpaths with high accuracy and precision. However, achieving optimal accuracy also depends on factors such as machine calibration, tool quality, and material properties.
2. PowerMill aims to streamline the CAM (Computer-Aided Manufacturing) workflow by offering automated features for toolpath generation, simulation, and optimization.
3. The quality of toolpaths generated by PowerMill significantly impacts machining results. Smooth, collision-free toolpaths ensure optimal material removal and surface finish.
4. Simulation plays a crucial role in verifying the machining process and detecting potential errors or issues before actual machining begins.
5. PowerMill generates G-code that is compatible with a wide range of CNC machines. However, post-processing may be required to fine-tune the G-code for specific machine configurations.
6. As manufacturing technologies evolve, there’s a continuous need for software improvements and new features to meet industry demands.

CHAPTER 6

# **6. Conclusion & Future Scope**

## **6.1 Conclusion**

1. Learnt PowerMill Software
2. Learnt PowerShape Software
3. Learnt Basics of 3-D printing
4. Learnt Visual Studio
5. Learnt Python Tkinter
6. Learnt Slicing of CAD Model
7. Learnt Toolpath Planning
8. Learnt Toolpath Generation

## **6.2 Future Plans**

Leveraging the gained proficiency in PowerMill, PowerShape, 3D printing basics, Visual Studio, Python Tkinter, the possibilities for future contributions to the field are abundant.

Next semester’s work is the creation of an extension for PowerMill aimed at streamlining the slicing process and other related operations.

**Utilizing Learned Technologies:**

1. **PowerMill and PowerShape:** The foundation for the extension will be built upon the advanced capabilities of PowerMill and PowerShape, harnessing their robust CAD/CAM functionalities to ensure seamless integration with existing workflows and processes.
2. **3D Printing Basics:** Understanding the fundamentals of 3D printing will be instrumental in designing features that cater to the unique requirements of additive manufacturing processes. This knowledge will inform decisions regarding slicing algorithms and optimization techniques tailored specifically for 3D printing applications.
3. **Visual Studio:** The development environment provided by Visual Studio will serve as the primary platform for coding and debugging the extension. Its rich set of tools and functionalities will expedite the development process and facilitate efficient collaboration among team members.
4. **Python Tkinter:** Tkinter will be employed for creating the graphical user interface (GUI) of the extension, offering users an intuitive and user-friendly interface for interacting with the various features and functionalities. The versatility of Tkinter will enable the design of visually appealing and responsive interfaces that enhance the user experience.

# **References**

[1] https://www.ankermake.com/blogs/guides/3d-printing-software-for-beginners

[2]

[3]

[4]

[5]

[6]

[7]

[8]

[9]

[10]