CNC FLAME CUTTING MACHINE

Student: Mara Cristina Urdea

Group: 30432

1. Project Overview

The goal of this project is to develop a CNC[1] Flame Cutting Machine Simulator, a software program that reads a file containing a cutting path and generates commands to simulate the movement of a cutting head along the X and Y axes[2].
The project will offer a valuable tool for testing and optimizing cutting paths before the actual execution, thus reducing material wastage and improving cutting accuracy.

2. Project Objectives

- Develop a user-friendly CNC Flame Cutting Machine Simulator.
- Provide a graphical user interface for user interaction and customization.
- Simulate the movement of the cutting head on a computer screen.
- Read input files containing cutting path information.
- Provide export functionality for generated CNC G-code.
- Allow users to analyze, visualize and adjust cutting paths.

3. Project Plan

1st Meeting – Project theme

- Choose the project theme.

2nd Meeting – Bibliographic study and project planning

- Define the project objectives and scope.
- Write a project plan.
- Identify key stakeholders and gather their requirements.
- Analyze the CNC Flame Cutting Machine.
- Find documentation sources.

3rd Meeting – Skeleton of the application

- Identify essential user interface requirements.
- Make a structure of the project.
- Familiarize with required languages (G-code, Python etc.).
- Start the implementation (skeleton).

---

[1] Computer Numerical Control
[2] Z axe? (depends on simulation)

<u>4th Meeting - Development</u>

- Development work.
- Review the progress.

<u>5th Meeting - Development</u>

- Development work.
- Start writing the documentation.
- Review the progress.

<u>6th Meeting – Final improvement</u>

- Final feedback.
- Define the final shape of the project.

4. Bibliographic study

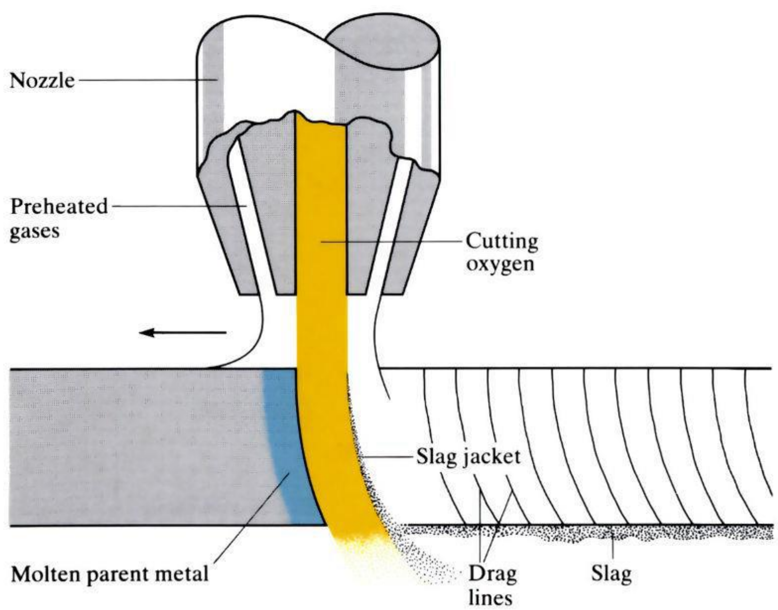

Fig. 1 – Flame Cutting Scheme

For a better understanding of the experiment, the above figure represents the flame cutting machine while cutting a piece of metal. We can observe the oxygen along with the preheated gases that will perform the cutting.

The current project focuses on the algorithm that helps the machine move according to the user-selected coordinates and simulate this movement on the computer's screen.

The working process consists of five important steps:

- Creating the design of the machine and the simulation environment using a GUI.
- Retrieving the movement coordinates from a text file.
- Transmitting these coordinates to the machine.
- Simulating the movement based on the coordinates.
- Generating the G-code for the design, that can be used to set up the actual CNC machine.

The user should be allowed to draw lines by inputting the X and Y coordinates, the start and end of the line. Also, the arcs will be drawn by inputting the start and end of the arc and the radius.

5. Analysis

5.1 G-Code

G-code, short for Geometric Code, is a standard programming language used to control computer numerical control (CNC) machines and 3D printers. It serves as the bridge between computer-aided design (CAD) software and the physical movements of CNC machines, dictating precise instructions on how the machine should operate. G-code is a fundamental component of CNC machining and plays a pivotal role in manufacturing, automation, and various other industries where precision is essential.

G-code is the language that CNC machines understand. It consists of a series of alphanumeric commands and parameters that instruct the machine how to move, what tools to use, and other crucial operational details. These commands control elements such as the tool's position, speed, direction, and tool changes.

Instructions are structured with a letter (often a 'G' for "geometry" or other letters denoting specific functions) followed by a number. For example, 'G00' signifies rapid positioning, 'G01' specifies linear interpolation (movement), and 'G02' or 'G03' instructs the machine to move in arcs. The instruction is often followed by numerical parameters, which provide specific details like coordinates, feed rates, and spindle speeds.

The most common coordinate system used in CNC machining is the Cartesian coordinate system, where X, Y, and Z axes represent three-dimensional space. These axes dictate the tool's position and movement in relation to the workpiece.

A line of G-code has the following structure:

G## X## Y## Z## F##

Example: G01 X247 Y11 Z-1 F40

- First is the G-code command and in this case that's the G01 which means "move in straight line to a specific position". (See Annex 1)

- We declare the position or the coordinates with the X, Y and Z values.

- Lastly, with the F value we set the feed rate, or the speed at which the move will be executed.

To wrap up, the line G01 X247 Y11 Z-1 F40 tells the CNC machine to move in a straight line from its current position to the coordinates X247, Y11 and Z-1 with speed of 40 mm/min.

5.2 G-Code from Cartesian coordinates

The algorithm for generating G-Code from Cartesian coordinates will take as inputs the X and Y, the feed rate (the rate at which the CNC machine should move) and the name of the G-Code file to be generated (for example “output.gcode”). The algorithm starts by creating an empty list to store the G-Code commands (‘G21’: sets the units to millimeters, ‘G90’: sets the positioning to mode absolute, ‘G1 F{feed_rate}’: sets the feed rate for movement). Then, it iterates through the provided Cartesian coordinates and generates G-Code for each point. For each coordinate pair (X, Y), it appends a G-Code command to move the CNC machine to that position using ‘G1 X{x} Y{y}’. Optionally, the program can add a G-Code command to return the CNC machine to the origin (X=0, Y=0) using ‘G0 X0 Y0’. Then the algorithm opens a file in write mode and writes the generates G-Code commands to that file.

5.3 Line Generation

6. Design

The main structure of the system is presented in Figure 2.

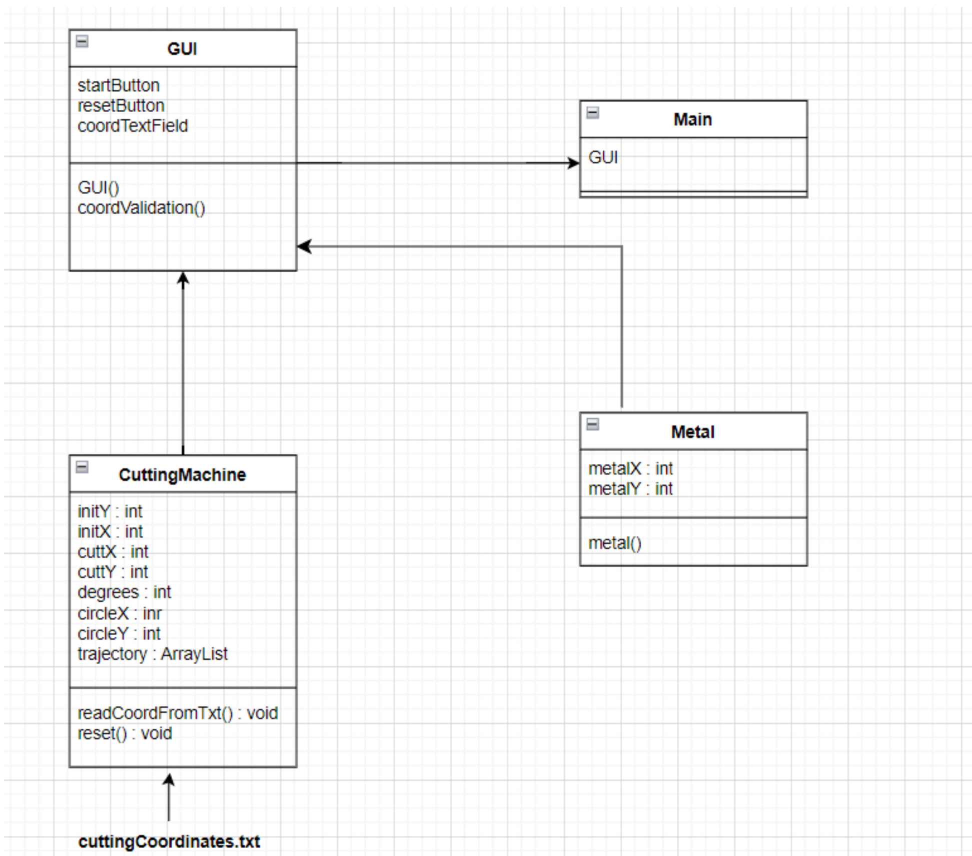

Fig.2 – Class Diagram

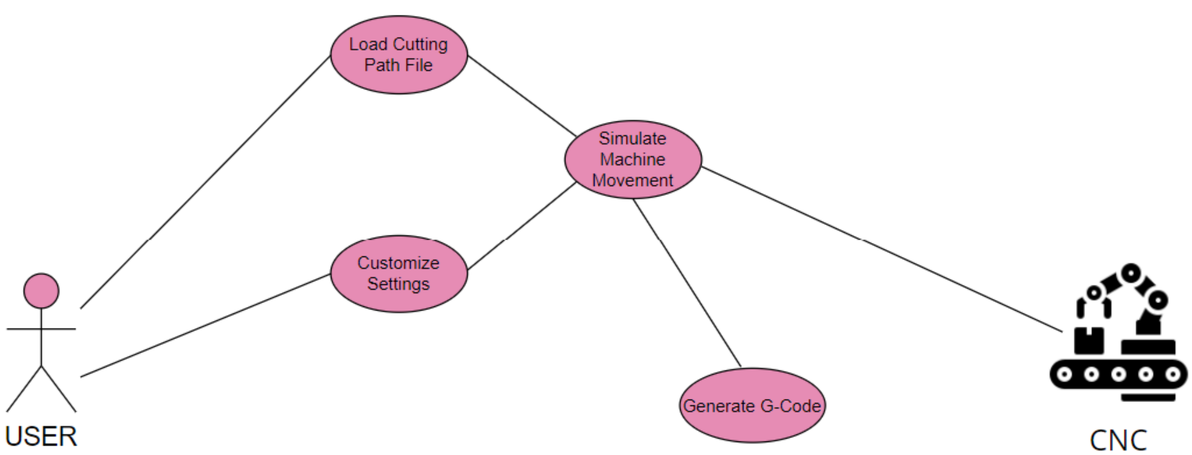

Fig.3 – Use Case Diagram

References

- [1] “G-code Explained | List of Most Important G-code Commands”, [Online]. Available: https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/
- [2] “Learn”, [Online]. Available: https://www.learnpython.org/
- [3] “What Is CNC Machining? An Overview of the CNC Machining Process”, [Online]. Available: https://astromachineworks.com/what-is-cnc-machining/
- [4]“Bresenham’s circle drawing algorithm”, [Online]. Available: https://www.geeksforgeeks.org/bresenhams-circle-drawing-algorithm/
- [5] “Flame cutting”, [Online]. Available: https://www.open.edu/openlearn/science-maths-technology/engineering-technology/manupedia/flame-cutting
- [6] “CNC G-Code”, [Online]. Available: https://gcodetutor.com/cnc-machine-training/cnc-g-codes.html
- [7]“Bresenham’s Line Generation Algorithm”, [Online]. Available: https://www.geeksforgeeks.org/bresenhams-line-generation-algorithm/
-

Annex 1. G-Code Reference List

G-Code Reference List

G Code	Function
G00	Positioning at rapid travel;
G01	Linear interpolation using a feed rate;
G02	Circular interpolation clockwise;
G03	Circular interpolation, counterclockwise;
G04	Dwell
G17	Select X-Y plane;
G18	Select Z-X plane;
G19	Select Z-Y plane;
G20	Imperial units;
G21	Metric units;
G27	Reference return check;
G28	Automatic return through reference point;
G29	Move to a location through reference point;
G31	Skip function;
G32	Thread cutting operation on a Lathe;
G33	Thread cutting operation on a Mill;
G40	Cancel cutter compensation;
G41	Cutter compensation left;
G42	Cutter compensation right;
G43	Tool length compensation;
G44	Tool length compensation;
G50	Set coordinate system (Mill);
G50	Maximum RPM (Lathe);
G52	Local coordinate system setting;
G53	Machine coordinate system setting;
G54	Set Datum;
G55	Set Datum;
G56	Set Datum;
G57	Set Datum;

G Code	Function
G58	Set Datum;
G59	Set Datum;
G70	Finish cycle (Lathe);
G71	Rough turning cycle (Lathe);
G72	Rough facing cycle (Lathe);
G73	Chip break drilling cycle;
G74	Left hand tapping (Mill);
G74	Face grooving cycle;
G75	OD groove pecking cycle (Lathe);
G76	Boring cycle (Mill);
G76	Screw cutting cycle (Lathe);
G80	Cancel cycles;
G81	Drill cycle;
G82	Drill cycle with dwell;
G83	Peck drilling cycle;
G84	Tapping cycle;
G85	Bore in, bore out;
G86	Bore in, rapid out;
G87	Back boring cycle;
G90	Absolute programming;
G91	Incremental programming;
G92	Reposition origin point (Mill);
G92	Screw thread cutting cycle (Lathe);
G94	Per minute feed;
G95	Per revolution feed;
G96	Constant surface speed (Lathe);
G97	Constant surface speed cancel;
G98	Feed per minute (Lathe);
G99	Feed per revolution (Lathe);

GCodeTutor.com