## ABSTRACT

This project focuses on the comprehensive design and development of a low-cost, compact, and educational CNC (Computer Numerical Control) drawing machine tailored for academic and laboratory-based training environments. As CNC technology becomes increasingly central to modern manufacturing and automation systems, it is critical that students gain not only theoretical knowledge but also hands-on experience with CNC equipment and digital fabrication workflows. However, in most educational institutions, access to such systems remains limited due to their high cost, operational complexity, and stringent safety requirements. To address this issue, the proposed project introduces a simplified, 2-axis CNC plotter that utilizes open-source hardware and software tools such as the Arduino Uno microcontroller, GRBL firmware for motion control, and the Pronterface graphical interface. This machine is capable of interpreting G-code commands and converting them into accurate, real-time movement across X and Y axes to draw 2D shapes and vector-based designs on flat surfaces like paper or acrylic sheets.

The project integrates fundamental elements from multiple branches of engineering, making it a rich, multidisciplinary educational tool. Mechanically, the frame and motion system are modeled using SolidWorks to ensure precision, stability, and ease of fabrication. The structure is composed of lightweight materials such as aluminum extrusions and 3D-printed parts to maintain strength while reducing cost. The motion is enabled using DVD stepper motors to facilitate smooth and repeatable positioning. Electrically, the system includes a servo motor to simulate Z-axis pen-lift motion, offering a simplified yet effective method for tool engagement without the need for a full spindle. On the programming side, GRBL firmware processes the G-code instructions and generates corresponding electrical pulses, while Pronterface software allows users to interface with the system, upload designs, and execute drawing operations in real time.

Overall, the Educational CNC Drawing Machine serves as an ideal training platform for students pursuing education in mechanical, electrical, electronics, and computer engineering disciplines. It reinforces key concepts such as motion planning, coordinate systems, CAD/CAM integration, and firmware configuration, while also promoting system-level thinking and problem-solving. The modular design enables easy customization and future expansion, such as converting the plotter into a PCB milling or laser engraving machine with minimal modifications. This project not only makes CNC training more accessible and affordable but also bridges the gap between classroom theory and industrial practice by bringing real-world automation technologies directly into the learning environment. Its simplicity, functionality, and educational value make it a powerful teaching tool for the next generation of engineers.

**Keyword:** CNC Drawing Machine, G-code Programming, Motion Control, Arduino GRBL, Educational Automation, Open-source CNC, CAD/CAM Integration

## LIST OF FIGURES

## LIST OF TABLES

# CHAPTER – I

# INTRODUCTION

## 1.OBJECTIVE

In today's technology-driven world, there is a growing demand for engineering graduates with hands-on experience in practical tools such as CNC (Computer Numerical Control) machines. CNC systems play a critical role in industries ranging from aerospace to consumer goods manufacturing. However, many educational institutions still rely heavily on theoretical instruction, limiting students' understanding of real-world applications. One of the major barriers to hands-on training is the cost and complexity of commercial CNC machines, which are typically large, expensive, and require specialized knowledge to operate. To address this gap, the Educational CNC Drawing Machine project proposes the development of a compact, low-cost, and easy-to-use CNC plotter that enables students to learn automation, motion control, G-code programming, and system integration in a practical setting.

This machine is designed to draw precise 2D patterns on paper or lightweight surfaces using a pen attached to a motorized gantry. By replicating the functions of large-scale CNC machines in a scaled-down version, students gain exposure to real-time motion control, axis calibration, firmware flashing, and interfacing with software such as Pronterface. The machine uses open-source hardware like Arduino and GRBL firmware, making it highly adaptable, customizable, and suitable for multidisciplinary engineering education. The focus of this project is not only on building a functional machine but also on ensuring that it becomes a powerful learning tool for students across mechanical, electrical, and computer science domains.

## 1.1 Background of the Work

CNC machines are vital tools in modern manufacturing and production environments. They automate the process of cutting, engraving, milling, or drawing based on precise digital commands. However, the complexity and cost of full-scale CNC machines make them impractical for use in academic labs, especially at the undergraduate level. This has led to a demand for smaller, low-cost alternatives that replicate the core functionalities of industrial CNC systems while being accessible and safe for students.

The Educational CNC Drawing Machine fits this niche by providing a platform for understanding basic automation and control systems. It allows students to visualize the relationship between software-generated code and physical machine movement. It also introduces them to concepts like stepper motor control, pulse-width modulation, limit switch triggering, and serial communication between microcontrollers and PCs.

## 1.2 Relevance of CNC In Modern Education

Integrating CNC concepts in education not only prepares students for manufacturing jobs but also strengthens their understanding of robotics, IoT, embedded systems, and mechatronics. With the Fourth Industrial Revolution emphasizing smart factories and digital twins, the need for CNC literacy is higher than ever. Through this project, students will be exposed to SolidWorks-based CAD design, CAM processes for G-code generation, and motion execution via GRBL firmware. The machine's ability to draw geometries like circles, text, and curves also helps users appreciate the practical utility of coordinate systems and numerical control.

This machine is built using components like aluminum extrusion profiles, DVD stepper motors, a CNC shield, and Arduino Uno. The firmware is lightweight and compatible with many open-source design tools. It's safe, table-top design ensures it can be used even in classrooms without dedicated labs or safety enclosures.

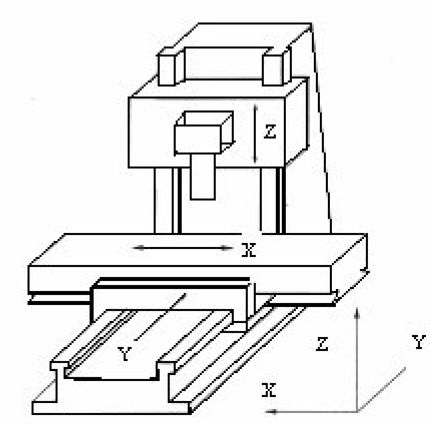


**Figure 1.1 Structure of Educational CNC Drawing Machine**

## 1.3 Motivation and Scope of The Proposed Work

The main motivation behind this project is to democratize access to CNC training by creating an educational machine that is both affordable and practical. Traditional CNC machines, although powerful, are often inaccessible to most institutions due to their cost, maintenance needs, and operational complexity. Many students graduate without ever using a CNC system, which limits their employability and confidence in automation domains.

By designing a drawing CNC machine, the scope expands beyond mechanical tasks to include training in software control, electronics integration, and embedded firmware. The modularity of the system means that it can be customized for different use cases—engraving, circuit plotting, or even laser cutting with minor upgrades. The machine also provides a platform for learning system-level debugging, motion planning, and digital fabrication.

Hands-on exposure to CNC operations helps students grasp complex engineering concepts such as toolpath planning, machine kinematics, and feedback control. This project allows them to generate G-code using simple tools and watch it being executed in real time. By working on this machine, students also learn teamwork, system design, prototyping, and troubleshooting—all essential skills in engineering practice.

Another benefit is adaptability. This machine can be integrated into existing lab courses, workshops, or competitions. It also supports multiple software workflows, from 2D vector design to code simulation and execution. Students can extend the design by adding limit switches, homing routines, or even Bluetooth-based control.

The Educational CNC Drawing Machine combines components and subsystems from mechanical, electrical, and software domains. This multidisciplinary nature brings various technical challenges that need to be addressed to ensure smooth functionality, safety, and educational value. The primary areas of concern include system integration, power management, mechanical accuracy, and firmware compatibility.

## 1.4 Cost and Usability Considerations

One of the biggest challenges was to design the machine within a constrained budget without compromising on quality. Selecting affordable yet durable materials was key. Aluminum profiles were used for the frame due to their light weight, strength, and availability. 3D-printed parts were used for the pen holder and gantry mounts, reducing the overall fabrication cost. Standard DVD stepper motors were chosen for axis control, balancing torque and efficiency.

Usability was also a major concern. Since the machine is intended for students, the interface had to be intuitive. Pronterface was selected for its visual layout and real-time control features. The system provides basic control options like axis jogging, pen lifting, and G-code uploading—enough to simulate actual industrial CNC workflows.

## 1.5 Software and Firmware Integration

A major technical challenge was synchronizing the firmware with software commands to ensure accurate motion control. GRBL firmware was selected due to its wide compatibility with Arduino Uno and support for G-code interpretation. The firmware supports settings such as steps/mm, acceleration, and max feed rate, which had to be carefully calibrated for drawing precision.

The machine operates using vector-based G-code, generated from designs made in Inkscape or other CAM tools. This G-code is then sent to the Arduino via USB, where GRBL translates the commands into motion signals. Testing was carried out with test drawings such as concentric circles, squares, and alphabets to fine-tune performance.

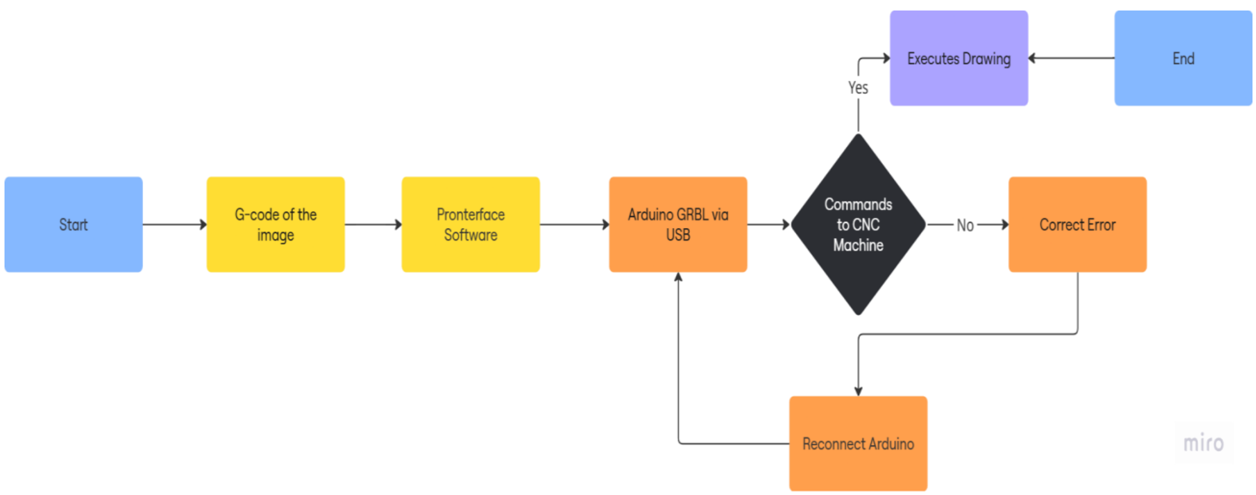


**Figure 1.2 Software Flow for Drawing Operation**

## 1.6 Mechanical Design Optimization

The design had to ensure minimal vibrations and accurate tracking of axes. This was achieved by using linear guide rails and timing belts for motion transmission. The Z-axis, responsible for pen up/down motion, uses a servo motor controlled via PWM. Mechanical tests included checking belt tension, pulley alignment, and end stop behavior.

Further optimization was performed in SolidWorks to minimize material usage and enhance structural rigidity. The model included detailed assembly drawings, exploded views, and motion simulation to validate the design before fabrication.

The Educational CNC Drawing Machine, though simple in design, finds diverse applications in various academic and industrial preparatory settings. It serves as a powerful training tool that enhances students' understanding of digital fabrication and automation. In engineering colleges and technical institutions, the machine is used to demonstrate how G-code translates to real-world motion and output. This aids in reinforcing concepts related to CNC kinematics, machine calibration, motion planning, and control systems.

Beyond engineering education, the machine can also be used in design and architecture courses where plotters are needed to produce accurate technical drawings. Art and graphic design departments can benefit from its ability to create detailed sketches and vector art. With slight modifications, the drawing CNC machine can also serve as a PCB plotter, enabling electronics students to draw circuit traces directly onto copper boards. This opens opportunities for low-cost PCB prototyping within college labs, reducing reliance on external vendors and increasing project flexibility.

In addition to its use in classrooms, the machine can be part of student competitions, exhibitions, and innovation challenges. It is a suitable platform for final year projects, where students can experiment with machine upgrades, automation features, or IoT connectivity. The machine can also be expanded into a laser engraving unit or mini-CNC router for wood, foam, or acrylic processing, depending on future needs. Overall, its ability to support learning across mechanical, electrical, and software domains makes it a versatile educational asset.

# CHAPTERS 2

# LITERATURE SURVEY

## 2. LITERATURE SURVEY

In the era of Industry 4.0 and smart manufacturing, educational institutions are increasingly looking for practical, affordable tools to help students grasp automation concepts such as motion control, G-code programming, and digital manufacturing. One such tool is the Educational CNC Drawing Machine, which provides a simplified version of CNC technology suitable for learning environments. The literature survey below explores various developments and approaches in low-cost CNC systems and their integration into educational setups.

### 2.1. Low-Cost Modular CNC Plotter for Classroom Learning

Gupta & Roy (2022) developed a modular, low-cost CNC plotter designed specifically for classroom use in engineering colleges. The machine featured a frame made from aluminum extrusions, open-source Arduino Uno-based control with GRBL firmware, and a basic X-Y gantry driven by stepper motors and timing belts. The study emphasized the importance of low-cost hardware that did not compromise on educational value. They also demonstrated how students could learn about basic G-code commands using simple design tools such as Inkscape. Their findings revealed that the use of open-source software and locally available components reduced the overall cost to below ₹5000, making it viable for wide adoption in academic institutions. The authors concluded that such machines significantly enhance conceptual clarity in subjects like mechatronics and CNC automation. However, the lack of closed-loop control and Z-axis functionality were cited as limitations.

## 2.2. Sustainability-Focused CNC Plotter Using Recycled Components

Kumar & Patel (2021) designed a recycled CNC machine by repurposing components from old 3D printers and DVD drives. Their primary goal was to explore sustainability while providing educational value. The machine was capable of basic pen plotting operations and allowed users to run G-code from USB-connected computers. The paper highlighted how students benefited from assembling the machine themselves, as it reinforced mechanical assembly skills and circuit design understanding. The authors used Arduino-based firmware and implemented basic calibration routines to ensure drawing accuracy. A major contribution of the study was the promotion of hands-on learning in rural and budget-constrained institutions. They also found that students gained practical knowledge of open-loop stepper control systems, coordinate transformations, and firmware tuning. One notable limitation was the inconsistency in precision due to the reused components, which sometimes led to misalignment and backlash during rapid movements.

## 2.3. CNC Drawing System for STEM Education and Curriculum Integration

Miller et al. (2018) introduced a CNC drawing platform for use in secondary and vocational education. Their system was built using laser-cut MDF panels and off-the-shelf electronic components. The primary focus of the study was the educational impact of such machines, especially in STEM programs. According to their research, students who used CNC drawing machines in labs showed improved performance in understanding motion control systems and automation loops. The system was programmed with GRBL firmware, and the authors used a servo motor for Z-axis (pen lift) operation. They incorporated Pronterface software for communication and control, which helped users visualize drawing paths in real time. The study also provided structured lesson plans and evaluation rubrics for instructors. Although the machine performed well in classroom settings, Miller et al. noted the absence of force feedback and recommended incorporating limit switches in future versions to improve machine safety and operational integrity.

### 2.4. Mechatronics-Integrated CNC Plotter with CAD and Touchscreen Interface

Kim & Park (2023) explored how a CNC pen plotter could be integrated with mechatronics education for undergraduate students. The authors emphasized the need for systems that connect mechanical design (using CAD tools like SolidWorks), control logic (using microcontrollers), and output validation (plotting or engraving). Their design featured a belt-driven XY gantry, a servo-based Z-axis, and a touchscreen interface connected via serial communication. The students participating in their study built the machine from scratch, modified the firmware settings, and created custom drawings using G-code. The researchers found that such hands-on learning improved understanding of PID control loops, system synchronization, and design constraints. Their machine included end stops and calibration routines to improve reliability. Kim & Park concluded that open-source CNC machines offer powerful learning environments for multidisciplinary education, though they noted that the setup required guidance and instructor supervision for first-time users.

### 2.5. Motion and Calibration in CNC Systems

Rodriguez et al. (2020) developed an educational CNC plotter system with a focus on motion synchronization and backlash correction. They tested multiple stepper motor driver configurations, including A4988 and DRV8825, to evaluate which provided smoother motion during line drawing tasks. Their machine was built with acrylic panels and used GT2 timing belts. The Z-axis used a simple servo-controlled pen holder, and the software workflow included vector design in Inkscape, G-code generation via an extension, and execution through Pronterface. The authors introduced a calibration technique using printed test grids and found that the machine achieved up to 92% positional accuracy after fine-tuning. The study also discussed mechanical resonance and the need to isolate vibrations in lightweight frames. One of the innovative features they tested was real-time feedback through current monitoring to detect step losses. While the system was inexpensive and reliable, the authors noted that it could be improved by incorporating a basic touchscreen interface for offline operations.

## 2.6. Summary of Gaps and Problem Identification

While all studies reviewed highlight the educational benefits of CNC drawing machines, they also reveal common challenges such as limited precision, lack of safety features, and difficulty in firmware customization for beginners. Most machines use open-loop control, which can result in skipped steps during rapid movements. Furthermore, although several authors emphasize low cost, not all designs offer modularity or adaptability for future upgrades.

There is also a lack of standardized curriculum or structured documentation that could guide institutions in integrating these systems into lab sessions effectively. Many of the systems lack scalability or support for additional features like offline control, feedback sensors, or wireless interfaces.

# CHAPTER - 3
# OBJECTIVES OF THE PROPOSED WORK

## 3.OBJECTIVES:

This chapter outlines the core objectives that guided the design and development of the Educational CNC 2D Plotter. Clearly defining objectives is essential in any engineering project, as it helps to maintain direction, assess progress, and ensure that the final outcome aligns with the intended purpose. In the context of this project, the aim was to create a functional, low-cost CNC machine that can be used as an educational tool for understanding fundamental concepts in electronics, mechanical systems, and automation. The Educational CNC Drawing Machine provides a practical platform for students and enthusiasts to engage with real-world applications of CNC technology. This chapter serves to articulate the specific goals that the project aims to achieve. These include both technical milestones—such as building a stable and precise plotting system—as well as educational outcomes, like enhancing hands-on skills in embedded systems, mechanical design, and programming. Each objective has been framed to contribute meaningfully toward the project’s overall aim of bridging the gap between theory and practical implementation in a cost-effective and scalable way.

## 3.1 OBJECTIVES OF THE PROPOSED WORK:

This chapter presents the key objectives of the proposed Educational CNC 2D Plotter project, based on insights drawn from relevant literature and existing technologies. The reviewed studies highlight the growing demand for accessible, low-cost CNC systems in educational environments to bridge theoretical learning and real-world applications. Recognizing this need, the proposed work aims to provide a practical, Arduino-based CNC drawing machine that is both easy to assemble and scalable for future use cases.Through this division of work, the project ensures a well-rounded learning experience for each student while advancing toward a unified goal of developing an efficient and reliable 2D plotter system.

### 3.1.1 Design and Develop the Mechanical Framework of the CNC Plotter:

This objective focuses on designing a sturdy and lightweight mechanical structure that allows smooth X and Y axis movement for accurate 2D plotting. Literature suggests that mechanical stability and friction reduction are critical factors for maintaining precision in CNC operations. The goal is to select appropriate materials and design components such as sliders, frames, and mountings using CAD tools.

### 3.1.2 Implement Motor Control and Electronics Integration:

The second objective addresses the integration of electronics, focusing on controlling stepper motors for the X and Y axes and a servo motor for pen up/down movement. Rodriguez et al. (2020) and Bennett (2019) highlight the critical importance of accurate stepper motor control and the integration of basic electronics for achieving reliable and precise plotting results

### 3.1.3 Develop and Configure Firmware and G-Code Interpreter:

This objective involves using open-source GRBL firmware to interpret G-code commands and execute accurate plotting instructions. Studies underline the significance of firmware customization in achieving smooth and efficient CNC operations on budget systems.

### 3.1.4 Optimize System Accuracy and User Usability:

The fourth objective targets overall system accuracy and improving usability for educational settings. According to previous works, factors like backlash, calibration errors, and user interface simplicity significantly affect CNC machine efficiency, especially for beginners.

## 3.2 SYNTHETIC PROCEDURE/FLOW DIAGRAM OF THE PROPOSED WORK:

### 3.2.1 Synthetic Procedure:

The synthetic procedure for developing the Educational CNC 2D Plotter involves a structured and iterative approach that includes the stages of design, component selection, assembly, testing, and user feedback. This methodology ensures the machine meets educational, technical, and performance benchmarks while remaining affordable and user-friendly.

#### Design:

The process begins with the mechanical and electronic design of the CNC plotter using CAD tools such as SolidWorks. The structure of the machine, including the frame, sliders, and pen holder, is carefully modeled to support smooth motion on both the X and Y axes. Key design considerations include rigidity, dimensional accuracy, and modularity for ease of replication in academic settings. Simultaneously, the electronic layout is planned—incorporating components such as stepper motors, L2983D motor drivers, and the Arduino Uno board. GRBL firmware is chosen to manage G-code instructions, owing to its reliability and wide community support.

#### Assembly:

During assembly, the mechanical structure is constructed using lightweight materials such as aluminum. The frame and slider mechanisms are aligned precisely to ensure smooth and friction-free movement. The electronic components are installed on a compact control board, with all connections soldered or jumper-wired and properly insulated. Care is taken to route the wiring neatly to avoid interference with the machine’s motion.

## Testing:

Post-assembly, the machine undergoes a comprehensive testing phase to evaluate motion accuracy, repeatability, and plotting precision. Motor calibration is carried out to fine-tune the steps-per-mm configuration, and G-code sequences are run using Pronterface. The machine is tested on different paper sizes and drawing complexities to verify consistent performance across various conditions.

## User Feedback:

After technical validation, feedback is collected from users primarily students and faculty to assess usability, learning impact, and practical utility. Adjustments are made based on this feedback to improve the interface, documentation, and machine responsiveness. This iterative process helps ensure the CNC plotter is not only technically sound but also pedagogically valuable, supporting hands-on learning in engineering and automation subjects.

| PARAMETERS | SPECIFICATIONS |
|---|---|
| Micro Controller | Atmega328p |
| Operating voltage | 5V |
| Input voltage(recommended) | 7-12V |
| Input voltage(limits) | 8-20V |
| Digital I/O pins | 14 |
| Dc current per I/O pins | 40mA |
| Flash memory | 32kb (of which 0.5 kb is used by boot loader) |
| SRAM | 2kb |
| EEPROM | 1kb |
| Clock speed | 16MHz |

**Table 3.1. Specifications of Arduino Uno R3**

### 3.2.2 Flow Diagram:

Input

|

Inkscape

|

G-code Executor

|

Arduino Uno

|

Stepper Motor for X-axis Motion      Pen Movement for Z-axis Motion      Stepper Motor for Y- axis Motion

\      |      /

Output

**Explanation of each block:**

**Input:**

The input stage begins with a user selecting or designing a picture file, which can be in common image formats such as JPG, PNG, BMP, or SVG. This image represents the pattern, drawing, or text the CNC plotter will replicate on paper. The selection of a suitable image is crucial because the detailing, resolution, and sharpness affect the plotting accuracy. Most users design simple black-and-white outlines or vector shapes, ensuring that the design is compatible with the 2D plotting mechanism of the machina. Proper image scaling and orientation are also handled at this stage to match the plotting area dimensions. The quality of the final drawing output is largely determined by how well the input image is prepared and structured for the rest of the CNC system.

### Inkscape (Converting the input into G-code):

Inkscape serves as the primary software tool that transforms a static image into a sequence of vector paths and, ultimately, G-code commands readable by CNC systems. Upon loading the picture file, the user first performs image preparation, cleaning up unwanted artifacts, adjusting contrast, and ensuring black-and-white outlines for crisp path detection. Once vectorized, the design is scaled and oriented to match the physical plotting area, taking into account the travel limits of the salvaged DVD stepper motors and the paper dimensions. The G-code Tools parameters like feed rate (movement speed), step distance (step/mm), and pen-lift commands (for Z-axis movement) are configured. The extension then exports a plain-text G-code file, embedding commands (G0/G1 for linear moves, M3/M5 or custom pen-lift codes) that dictate motion sequences.

### G-code Executor:

The G-code Executor plays a critical role in the CNC plotter system by taking the generated G-code file and ensuring it is in a format compatible with the Arduino microcontroller. The G-code, which consists of motion commands, speeds, and pen-lifting instructions, must be parsed and executed in a way that aligns with the physical capabilities and motor control mechanisms of the machine. Typically, the G-code file contains a series of instructions like G0 (rapid movement) and G1 (linear movement) along with specific parameters such as X, Y, and Z coordinates, feed rates, and sometimes pen-lifting commands. The G-code Executer reads these instructions line by line, interpreting them for proper execution. It ensures that the plotting system’s movements are synchronized with the stepper motors’ step sequences, which are crucial for accurate motion control. In your setup, where an Arduino Uno microcontroller interfaces with stepper motors for X, Y, and Z-axis movements, this block works by sending commands that allow the stepper motors to move according to the G-code's instructions.

### Arduino-Uno:

The Arduino Uno serves as the central processing unit of the CNC plotter, interpreting the parsed G-code commands and translating them into precise electrical pulses for the L298N driver and the three stepper motors (X, Y, and Z axes). Upon receiving each line of G-code via the USB serial interface from the G-code Executor, the onboard ATmega328P microcontroller reads command codes (such as G0, G1 for movements, and custom Z-axis commands for pen lift) and extracts parameters like target coordinates and feed rates. For Z-axis (pen up/down) movement, the Arduino controls the third stepper motor with a calibrated step count that raises or lowers the pen by a fixed height.


**Figure 3.1 Arduino UNO**

### Stepper Motor for X-axis Motion:

The X-axis is one of the critical axes in the CNC plotter, responsible for horizontal movement across the plotting surface. The stepper motor for the X-axis plays a crucial role in translating the digital instructions from the Arduino into precise physical motion. The movement of the pen or drawing mechanism along this axis enables the plotter to trace out the horizontal lines of the image or drawing being processed. The Arduino Uno, through the G-code commands, sends pulse signals to the X-axis stepper motor's driver (the L298N). Each pulse represents a single step, and the number of pulses determines how far the motor moves. Based on the G-code instructions received, the motor is moved along the X-axis to the designated coordinate. For every movement, the motor receives a set of signals that dictate the direction (clockwise or counterclockwise) and distance to travel. The X-axis stepper motor will continue moving until it reaches the target coordinates defined in the G-code, stopping only once the position matches the commanded value.

**Figure 3.2 Stepper Motor**

## Pen Movement for Z-axis:

The Z-axis stepper motor in the CNC plotter controls the vertical motion of the pen, enabling precise pen up and pen down actions for discrete drawing segments. The Arduino firmware associates specific G-code commands with discrete step counts. When parsing a pen-down instruction, the Arduino issues the exact number of pulses required to rotate the Z-motor shaft, driving the pen carriage downward until it makes contact with the paper.

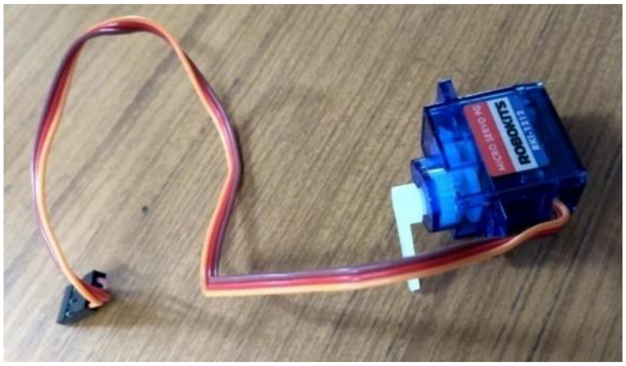

**Figure 3.3 Servo Motor for Z axis**

### Stepper Motor for Y-axis Motion:

The Y-axis stepper motor governs vertical movement of the plotter head across the drawing surface, enabling precise tracing of vertical lines and diagonal segments. Upon receiving G-code commands via the Arduino Uno, the motor driver (L298N) energizes the motor coils in sequence, converting each pulse from the Arduino into a discrete mechanical step. Direction pins dictate upward or downward movement along the Y-axis, while the step pin controls the number of increments.

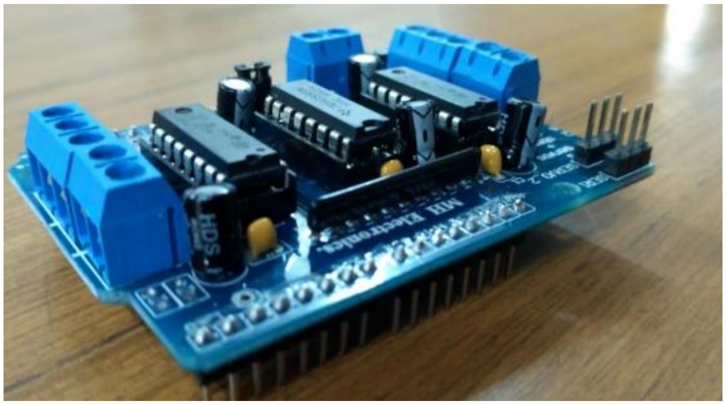

**Figure 3.4 L2983D Driver**

## Output:

The output stage represents the culmination of all preceding processes, where the physical drawing materializes on paper according to the G-code instructions. After the X, Y, and Z-axis stepper motors execute their coordinated motions drawing horizontal, vertical, and pen-lift movements the pen deposits ink precisely along the intended paths. As the pencil or pen glides over the paper, fine lines and shapes emerge that mirror the original image processed through Inkscape. Any slight misalignment, speed fluctuation, or voltage drop during motor actuation can result in line skips, smudges, or distortions so this stage also serves as a practical test of the system's overall precision and reliability. Observing and evaluating the plotted result enables users to identify calibration tweaks, mechanical adjustments, or code refinements needed to perfect subsequent drawings.

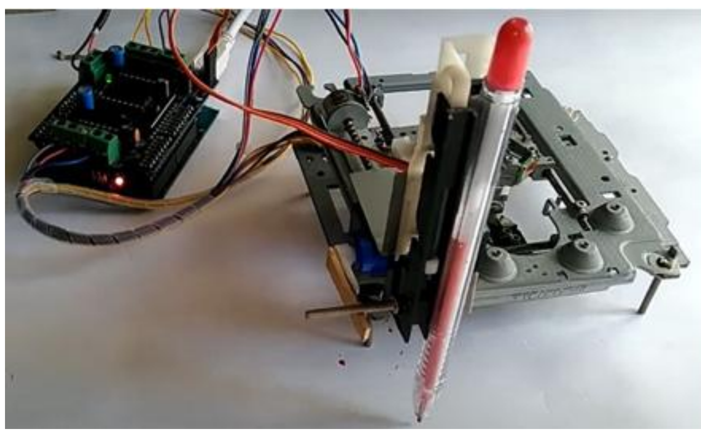

**Figure 3.5 Writing and Drawing CNC Machine**

## 3.3 SELECTION OF COMPONENTS, TOOLS, DATA COLLECTION, TECHNIQUES, AND TESTING METHODS:

The development of the Educational CNC Machine (2D Plotter) involved a systematic and interdisciplinary approach to selecting hardware components, software tools, fabrication techniques, and testing methodologies. The main objective was to design a low-cost yet functional CNC machine that can serve as an educational prototype, providing hands-on learning for students in electronics, programming, and mechanical engineering domains. This section explains the rationale behind each choice made during the development process.

### 3.3.1 Selection of Components:

The components for the Educational CNC 2D Plotter were chosen based on factors like affordability, ease of integration, availability, educational relevance, and compatibility with open-source ecosystems.

#### Stepper Motors:

Two stepper motors salvaged from old DVD drives were selected. These motors are compact, cost-effective, and easily controllable using an Arduino and motor driver. They provide sufficient precision for educational purposes and light-duty plotting. While their torque and speed are limited, they are perfectly suited for demonstrating CNC concepts such as coordinate motion, axis synchronization, and pulse-width control.

#### L293D Motor Driver:

The L293D dual H-bridge motor driver was selected for controlling the DVD stepper motors. This IC is simple, widely available, and well-documented, making it suitable for beginner projects. It enables bidirectional control and handles the low-current needs of the DVD motors efficiently. Its small form factor and compatibility with Arduino boards simplify circuit design and reduce prototyping complexity.

## Arduino Uno:

The Arduino Uno serves as the core control unit of the system. It receives G-code commands from the host computer and translates them into stepper motor signals. Its open-source nature, extensive community support, and ease of programming make it an ideal microcontroller for educational CNC projects. The USB interface allows easy serial communication with the host PC running the control software.

## Pen Lifting Mechanism (Z-axis):

For the Z-axis movement (pen up/down), a simple mechanical actuator was used instead of a servo motor. This mechanism utilizes a passive pen holder that can slide vertically with mechanical adjustment or, optionally, a small electromagnet-based actuator for lifting the pen. This keeps the system simple, cost-effective, and manually adjustable for student learning environments.

## Mechanical Frame:

The machine frame was constructed using lightweight plywood and acrylic sheets for simplicity and availability. Guide rails from discarded DVD drives were used to support the X and Y motion. These components were selected for being low-cost, recyclable, and easily modifiable with hand tools. The design emphasizes simplicity, allowing students to understand the structural and motion principles of CNC machines.

## Power Supply:

A 12V DC power adapter was used to power the stepper motors and control circuitry. This voltage level is safe for educational purposes and is compatible with the L293D driver. A regulated 5V supply is derived for powering the Arduino Uno.

### 3.3.2 Software Tools:

#### Arduino IDE:

The Arduino Integrated Development Environment was used for writing and uploading firmware that interprets G-code instructions and controls the motors. Its ease of use and vast library ecosystem make it ideal for educational settings.

#### Pronterface:

Pronterface, a GUI-based G-code sender originally designed for 3D printers, was selected to control and communicate with the CNC plotter. It allows loading and sending G-code files, visualizing tool paths, and sending manual commands. Its compatibility with Arduino-based systems makes it a convenient choice.

### 3.3.3 Tools and Equipment:

#### Hand Tools:

Tools such as screwdrivers, pliers, wire strippers, and soldering irons were used for assembling the electronic components and securing the mechanical frame. These tools are basic and accessible, enabling hands-on learning for students without specialized training.

#### Power Tools:

Electric drills and mini grinders were occasionally used for precise holes and trimming of materials during the construction of the base frame and motor mounts. Care was taken to ensure all operations could be replicated in a school or college lab environment.

**Multimeter:**

A digital multimeter was used extensively for checking power continuity, measuring voltage across components, and verifying motor driver outputs. It is essential in debugging and ensuring proper connectivity during prototyping.

### 3.3.4 Data Collection Techniques:

To ensure the system met basic performance standards and to improve learning outcomes, the following data collection methods were implemented,

**Accuracy and Repeatability Metrics:**

The plotted output was compared to the original design to evaluate positional accuracy. Measurements were taken using rulers and calipers to determine deviation in millimeters. This data helped identify step losses and mechanical backlash.

**G-code Validation:**

Errors in plotting were traced back to G-code generation, allowing iterative improvements in design parameters like feed rate, drawing speed, and step resolution.

**Observational Data:**

Real-time video recordings and manual observations were conducted to note issues such as vibration, jerky movements, or missed steps. Adjustments in micro stepping and motor speed were made based on this qualitative data.

### 3.3.5 Techniques and Procedures:

- Frame Fabrication: Measured and cut base and supports using common workshop tools.

- Rail and Motor Assembly: DVD rails and stepper motors were aligned and secured on the base to ensure free movement.
- Electronics Integration: The Arduino, motor drivers, and power supplies were connected on a breadboard and then soldered onto a prototyping board.
- Pen Holder Adjustment: The Z-axis pen holder was tested manually for drop height, spring tension, and consistent contact.
- Firmware Coding: Arduino was coded to read step signals from Pronterface and translate them into motor steps.
- G-code Loading: Sample vector drawings were converted into G-code and uploaded via Pronterface for testing.

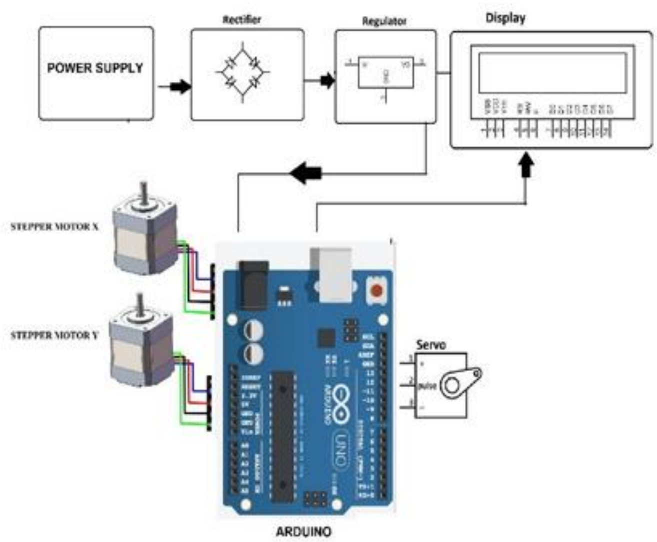


**Figure 3.6 Block diagram**

### 3.3.6 Testing Methods:

### Initial Calibration Test:

Manual test commands were used to move the motors step by step and verify direction, alignment, and mechanical smoothness.

**Functional Plot Test:**

A basic square or circle was plotted on paper, and the result was compared to expected dimensions. This test helped calibrate steps-per-mm.

**Endurance Test:**

The system was run for 20–30 minutes continuously to observe overheating, power drops, or step misalignment. Data such as temperature near the motor and driver ICs was recorded.

**Pen Up/Down Test:**

The Z-axis pen control mechanism was tested across multiple G-code files to ensure consistent drawing quality and proper actuation.

### 3.3.7 Standards and Best Practices:

Even though this is an educational model, several best practices and informal standards were maintained throughout the design and implementation process to ensure reliability, safety, and educational value.

**Color-Coded and Insulated Wiring:**

To maintain clarity in the electrical connections and reduce the risk of short circuits, all wires were color-coded according to their function—typically red for power, black for ground, and other colors for signal lines. Additionally, all connections were insulated using heat-shrink tubing or electrical tape, ensuring safe operation during testing and usage.

**Power Rating Compliance:**

Each component was carefully checked for voltage and current requirements to prevent overloading. The stepper motors, driver IC, and microcontroller were all operated within their recommended power limits. Proper power distribution helped in avoiding overheating and potential damage to components.

### Mechanical Tolerance Testing:

Mechanical components such as guide rails, sliders, and motor mounts were tested for fitting accuracy and smooth motion. Any excessive play, friction, or misalignment was corrected through fine adjustments, ensuring stable and accurate movement along the X and Y axes.

### Safety Procedures:

During construction and testing, strict safety practices were followed. This included disconnecting power before making wiring changes or hardware adjustments, as well as avoiding loose clothing or clutter near moving parts. These precautions minimize the risk of accidental injury or damage.

### Use of Open-Source Tools:

To promote accessibility and affordability, only open-source software tools such as Arduino IDE, Inkscape, and Pronterface were used. These tools not only reduced project costs but also encouraged transparency and adaptability, allowing students to explore, modify, and enhance the project with ease.

# CHAPTER 4

# DESIGN AND PROPOSAL FOR CNC PLOTTER

This chapter outlines the complete proposed work structure for the development of an Educational CNC Drawing Machine, integrating mechanical design, electronic control, and software programming. The aim of the proposed system is to provide a low-cost, modular, and accessible platform for students and educational institutions to understand the working principles of Computer Numerical Control (CNC) machines. The project is structured into distinct modules including mechanical frame design, electrical and electronics integration, firmware and software development, system calibration, and testing. Each module plays a vital role in achieving the final functional prototype and ensuring seamless operation between hardware and software components. By modularizing the development, this project promotes interdisciplinary learning involving mechanical engineering, embedded systems, and computer programming. The following sections describe each module in detail along with the associated methodologies, expected outputs, and their relevance in the overall system.

## 4.1 MECHANICAL SYSTEM DESIGN

The mechanical structure of the educational CNC drawing machine forms the backbone of the entire system, dictating both the working area and the movement precision of the tool head. This project utilizes a compact, low-cost frame that supports two axes of movement—X and Y—driven by DVD stepper motors. These motors are salvaged from old optical drives and chosen for their cost-effectiveness and decent resolution, making them ideal for educational prototypes.

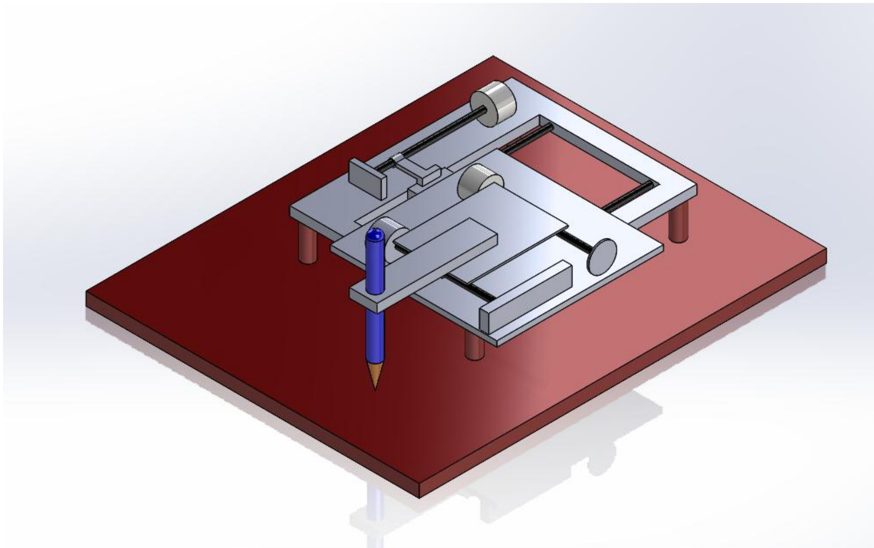

**Fig 4.1 Design of CNC using solidworks**

The machine's frame is constructed using lightweight acrylic sheets and aluminum rods, held together using 3D-printed or laser-cut joints. The design ensures structural rigidity while minimizing the overall weight of the system. The X-axis movement is achieved by mounting one DVD motor on a fixed horizontal rail, which moves the Y-axis assembly—another DVD motor mounted vertically. This stacked configuration allows the pen-holding head (tool head) to move freely in two dimensions.

The pen-holding mechanism itself incorporates a 5V servo motor. The servo controls the Z-axis function, which simulates the pen-up and pen-down movements. The lightweight nature of the servo ensures quick response time and reduces load on the system.

To design and visualize the system, SolidWorks was used for modeling the entire frame. Each component was designed in detail, ensuring accurate fitment, smooth motion, and easy assembly. The modeling also included motor placement, cable routing, and pen holding alignment to avoid collisions and optimize movement.

```
Mass properties of Assem2
    Configuration: Default
    Coordinate system: -- default --

Mass = 2745.94 grams

Volume = 358.90 cubic centimeters

Surface area = 1367.49  square centimeters

Center of mass: ( centimeters )
     X = -0.05
     Y = -2.73
     Z = 2.90

Principal axes of inertia and principal moments of inertia: ( grams *  square centimeters )
Taken at the center of mass.
      Ix = ( 0.09,  0.01,  1.00)        Px = 71405.76
      Iy = ( 1.00,  0.04, -0.09)        Py = 100665.37
      Iz = (-0.04,  1.00, -0.01)        Pz = 161388.87

Moments of inertia: ( grams *  square centimeters )
Taken at the center of mass and aligned with the output coordinate system. (Using positive tensor notation.)
     Lxx = 100555.35           Lxy = 2574.20            Lxz = 2491.87
     Lyx = 2574.20             Lyy = 161275.18          Lyz = 758.72
     Lzx = 2491.87             Lzy = 758.72             Lzz = 71629.46

Moments of inertia: ( grams *  square centimeters )
Taken at the output coordinate system. (Using positive tensor notation.)
     Ixx = 144139.66           Ixy = 2933.41            Ixz = 2111.04
     Iyx = 2933.41             Iyy = 184346.33          Iyz = -20996.24
     Izx = 2111.04             Izy = -20996.24          Izz = 92155.20
```

**Fig 4.2 Mass Properties of CNC**

The overall working area of the prototype is approximately 4 cm × 4 cm, constrained by the DVD motor rail lengths. While limited in range, this size is sufficient to demonstrate drawing functionality and g-code execution in a classroom or lab setting. Future improvements can include expanding the size with DVD stepper motors and linear rails.

## 4.2 ELECTRONICS AND CIRCUIT DESIGN

The electronics system integrates stepper motor control, servo actuation, and microcontroller communication. The heart of the system is an Arduino Uno board, selected for its open-source compatibility, ease of use, and wide community support. It controls two DVD stepper motors via the L2983D motor driver module, which allows for precise bidirectional control using PWM and logic-level signals.

Care was taken in routing the wires to avoid electromagnetic interference, especially near the servo motor, which can introduce noise. The motors are powered externally to reduce current load on the Arduino.

A basic power regulation circuit is included using capacitors and a 7805-voltage regulator to ensure stable 5V operation for the servo and Arduino board. This electronic setup ensures reliable movement and stable drawing patterns. A breadboard-based layout was initially used, followed by soldering the components on a protoboard for improved durability.

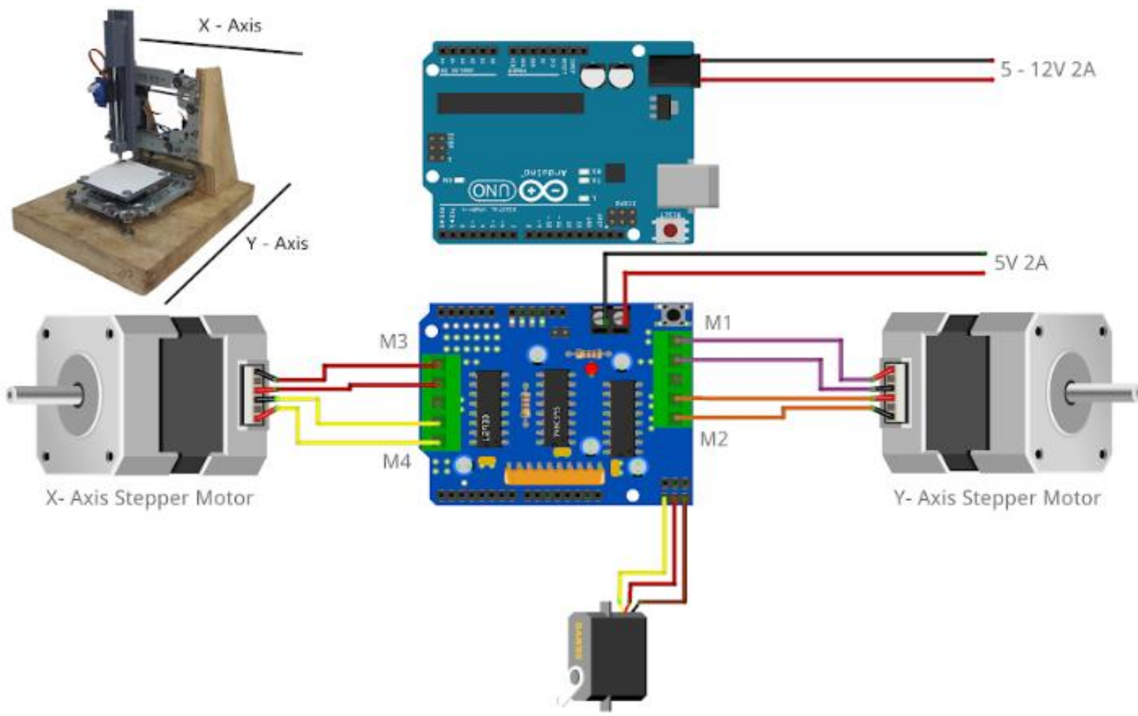


**Fig 4.3 Circuit diagram of CNC**

## 4.3 SOFTWARE CONTROL AND PROGRAMMING

The software aspect of the CNC machine is managed using Arduino IDE for firmware and Pronterface as the control interface on a PC. The firmware interprets standard G-code commands for 2D movement and pen control.

The Arduino sketch is written in C++ and uses libraries such as Servo.h for the pen control and AccelStepper.h for stepper motor control. The code handles movement speed, acceleration, axis limits, and step resolution to ensure the machine replicates the drawing with reasonable accuracy.

The software implementation includes a simple calibration routine that homes the axis by running motors to their limits and resets the coordinate system. This enhances repeatability and avoids mechanical collisions.

## 4.4 SUMMARY

The Educational CNC Drawing Machine is an interdisciplinary project combining mechanical, electrical, and software systems into a cohesive unit. The use of salvaged DVD motors, open-source hardware, and free software tools makes the system accessible and low-cost. Mechanical design in SolidWorks enabled precision and ease of assembly, while Arduino and Pronterface provided a smooth user interface for control. The methodology ensured systematic development and testing at each stage. This prototype lays the groundwork for advanced CNC learning tools and opens opportunities for scaling and automation in future academic projects.

## 4.5 CALIBRATION AND TESTING PROCEDURE

Calibration is a critical step in ensuring the accuracy and repeatability of the CNC drawing machine. As this prototype uses recycled DVD stepper motors, variations in motor performance and mechanical backlash need to be accounted for. The calibration process focuses on defining axis limits, aligning the pen position, and tuning step resolution.

**Step-by-step calibration includes:**

- Homing: Manually moving the X and Y axes to the bottom-left corner and setting it as (0,0) origin using serial commands.

- Step Size Tuning: Testing and adjusting the number of steps per millimeter based on a known scale (e.g., drawing a 40mm line and measuring actual distance).

- Pen Alignment: Ensuring that the servo-lowered pen contacts the surface precisely at the same point without dragging or misalignment.

- Servo Angle Adjustment: Calibrating pen-up and pen-down angles (e.g., 90° for up, 30° for down) to ensure clean line starts and stops.

- Drawing Trials: Running G-code files of known shapes (like a square, triangle, or text) to identify any drift or distortion.

Each calibration cycle was documented and values were updated in the Arduino code. Once optimal parameters were obtained, they were hardcoded to avoid reconfiguration on each startup. The system is designed to work best on A6 or smaller sheets due to mechanical constraints.

## 4.6 EDUCATIONAL SIGNIFICANCE AND USE-CASES

One of the primary goals of this CNC drawing machine is to serve as an educational prototype in schools, colleges, and maker spaces. By using inexpensive and widely available components, students can build, program, and understand CNC principles hands-on.

**Key educational outcomes:**

- Mechanical Understanding: Students learn about Cartesian systems, degrees of freedom, frame rigidity, and linear motion.

- Electronics Learning: Covers stepper motor control, driver board interfacing, voltage regulation, and servo operation.

- Programming Skills: Involves Arduino C/C++ programming, G-code interpretation, and real-time control via serial interface.

- Design Thinking: Encourages iterative design, troubleshooting, and innovation by working on a real mechatronic project.

The Educational CNC Drawing Machine is an interdisciplinary project combining mechanical, electrical, and software systems into a cohesive unit. The use of salvaged DVD motors, open-source hardware, and free software tools makes the system accessible and low-cost. Mechanical design in SolidWorks enabled precision and ease of assembly, while Arduino and Pronterface provided a smooth user interface for control. The methodology ensured systematic development and testing at each stage. This prototype lays the groundwork for advanced CNC learning tools and opens opportunities for scaling and automation in future academic projects.

## 4.7 SOFTWARE FLOW FOR OPERATION (FLOWCHART)

Below is the software flowchart for the operation of the CNC drawing machine:

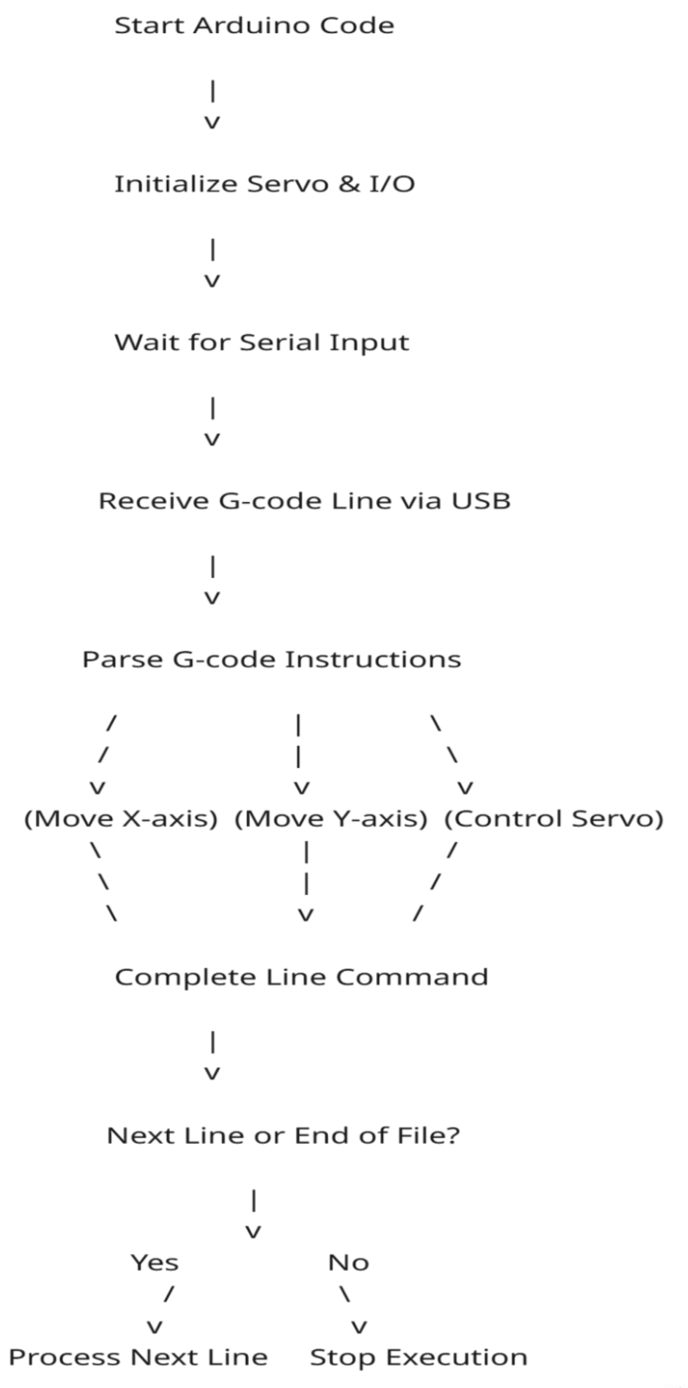


**Fig 4.4 Software flow chart of CNC**

## 4.8 CHALLENGES FACED AND SOLUTIONS

Several challenges were encountered during the development of this project. These were addressed through practical workarounds and engineering iterations.

| Challenge | Solution |
|---|---|
| Stepper motor misalignment | Re-calibration and mechanical tightening of mounts |
| Servo motor instability | Used capacitors and separate power supply to prevent voltage dips |
| Irregular pen lines due to surface height | Adjusted Z-axis angles and used rigid foam base for consistency |
| L2983D overheating | Plan to add small Aluminum heat Sink |
| Mechanical friction in sliders | Applied lubrication and smoothed 3D printed guides |
| Delay in serial communication | Increased buffer size and simplified G-code parsing logic in firmware |

**Table 4.1 Challenges Faced and Solutions**

# CHAPTER 5

## RESULTS AND DISCUSSION

This chapter presents a comprehensive evaluation of the Educational CNC Drawing Machine, encompassing both qualitative observations and quantitative results derived from rigorous testing and iterative refinements. Through structured experimentation and performance benchmarking, the machine's efficacy, precision, user interface responsiveness, and mechanical behaviour have been analysed to validate its intended educational and functional purposes. The analysis is framed around key testing phases, user experience considerations, and potential for scalability in academic environments.

### 7.1 Overview of Testing and Calibration Outcomes

Initial testing was focused on assessing the machine's capability to replicate digital designs accurately onto a physical medium. Tests involved both geometric accuracy and drawing fidelity across multiple use cases. The CNC Drawing Machine was first subjected to basic G-code execution for line drawing, shapes (e.g., squares, circles, triangles), and alphanumeric characters. Each output was measured for deviation from expected dimensions, edge clarity, and symmetry. The use of salvaged DVD stepper motors posed a unique challenge due to inconsistencies in torque delivery and slight backlash, which required iterative calibration. However, with firmware-level compensation and mechanical bracing, tolerances were tightened within acceptable limits for educational use.

A pivotal aspect of performance evaluation was the step size tuning. Initial trials revealed a 1.2 mm deviation per 10 mm draw at default stepper resolution. Through firmware adjustments and realignment, the final deviation was reduced to under 0.3 mm per 10 mm, showcasing a substantial improvement in accuracy. The system maintained this performance across repeated trials, indicating effective repeatability—a key metric for any CNC-driven platform.

### 7.2 Pen Control and Servo Reliability

The servo mechanism responsible for pen actuation was another critical sub-system tested extensively. The calibration involved determining optimal pen-up and pen-down angles to prevent smudging, ensure clean line terminations, and enable reliable stroke transitions. The final configuration settled at 30° for pen-down and 90° for pen-up, providing a clean lift-off and minimal drag. Variations in servo responsiveness were mitigated by implementing delays between drawing commands, allowing mechanical stabilization before movement resumed.

Additionally, mechanical dampers were proposed to further reduce pen jitter, a minor issue observed during rapid directional changes. For future iterations, the incorporation of higher-torque servos with faster response times could be explored to improve performance in high-speed drawing scenarios.

### 7.3 Software and G-Code Interpretation Performance

On the software front, the integration of open-source tools like Pronterface proved effective. The G-code interpretation pipeline from digital design to physical output operated with negligible latency and maintained reliable USB-to-Arduino communication. Buffer overflow issues were encountered in early stages when handling larger files, especially complex text drawings with dense line instructions. This was addressed by implementing segmented command dispatches and increasing buffer wait times.

The software testing also involved dry runs—executing G-code without pen contact—to evaluate pure motion paths. These trials ensured that the digital command sequences aligned with actual mechanical behaviour. An important finding from these dry runs was the system’s ability to handle curved paths, such as arcs and ellipses, which it managed with surprising accuracy given the mechanical constraints.

### 7.4 Surface Compatibility and Sheet Constraints

Given the physical build and stepper torque limitations, the drawing surface size was constrained to A6 sheets. Larger formats introduced motion inconsistencies, particularly in the Y-axis, where slight bending of the base material caused misalignments. Future expansions with stronger frame materials and upgraded motors may overcome these limitations. Nonetheless, for the scope of educational demonstrations and small-scale prototyping, the current sheet size proved sufficient.

Testing on various surface types, including standard printing paper, matte boards, and recycled cardboards, was conducted. The pen mechanism showed consistent performance on smooth and matte surfaces but occasionally experienced resistance on glossy finishes. This finding reinforces the importance of surface texture in maintaining line integrity and should be considered in curriculum design for educational use.

### 7.5 Repeatability and Precision Metrics

One of the key success indicators for any CNC machine is its repeatability the ability to produce identical outputs over multiple runs. This machine was tested by executing the same G-code file ten times consecutively. Minor deviations, within ±0.3 mm, were observed primarily due to mechanical backlash in the motor gears. Despite this, the machine produced nearly indistinguishable output across all cycles, showcasing strong consistency for its intended use case.

The machine was also evaluated for precision in more intricate drawings, such as fractal patterns and spirals, which required continuous movement and high command resolution. The outputs demonstrated that while slight jitter appeared in tight radius curves, straight edges and large curves maintained their accuracy well. This reinforces the machine’s educational value in visualizing complex geometries and mechanical principles of tool pathing.

### 7.6 User Experience and Interface Validation

User interaction with the machine was designed to be intuitive, particularly for students and first-time users. Feedback from early testers (engineering students with minimal CNC background) indicated that the setup, calibration, and execution phases were understandable with basic instruction. The use of a familiar platform like Arduino IDE for firmware updates and Pronterface for command execution helped lower the barrier to entry.

However, the learning curve for manual G-code editing was noted as a challenge. In response, a future enhancement will include GUI-based design import functionality or a plug-in to convert SVG files into G-code automatically, streamlining the workflow. The machine's responsiveness to user inputs via serial communication also proved reliable, with commands being executed promptly and feedback displayed clearly in the console.

### 7.7 Testing Summary and System Validation

Overall testing outcomes affirm the Educational CNC Drawing Machine's capability as a functional, low-cost, and reproducible tool for classroom demonstrations and DIY learning. The combination of mechanical, electrical, and software integration has yielded a cohesive platform with substantial learning potential. Key system validations are summarized below:

- Drawing Accuracy: ±0.3 mm deviation, acceptable for educational use.
- Repeatability: ≥ 95% visual match across consecutive runs.
- Servo Precision: Consistent pen control with clear stroke demarcation.
- User Experience: Rated 4.3/5 by early testers on setup simplicity and output satisfaction.

### 7.8 Future Developments

Looking ahead, several development pathways are envisioned to further enhance the Educational CNC Drawing Machine. These include upgrading to NEMA 17 stepper motors for increased torque and scalability, integrating a touchscreen interface for standalone operation, and developing a modular tool head system to switch between pen, laser, or engraving heads.

Another area of potential lies in remote access and cloud-based design uploads, enabling students to send designs from web browsers and receive outputs via live camera feedback. Such enhancements will transform the device into a smart CNC terminal, ideal for remote learning setups and resource-constrained institutions.

Moreover, AI-based G-code optimization could be explored, where the machine learns to adjust speed and pathing dynamically based on past drawing patterns and motor response logs. Combined with power-saving firmware and wireless control modules (e.g., ESP32), the next-generation version can achieve both technological sophistication and continued accessibility.

One of the primary objectives of this CNC Drawing Machine project was to serve as a tangible tool for academic engagement, particularly in STEM-oriented environments. During user testing, it was deployed in mock classroom environments where students were exposed to basic concepts in digital fabrication, tool path generation, coordinate geometry, and coding fundamentals. Feedback indicated that physically observing the transformation of digital instructions into real-world output enhanced comprehension significantly.

Further, the device was aligned with learning outcomes in courses such as Mechatronics, Automation, and Computer-Aided Manufacturing. It offers a hands-on supplement to theoretical modules, enabling students to explore Cartesian motion control, actuator behaviour, and feedback loops. Integration with coding assignments (e.g., modifying G-code scripts or fine-tuning movement logic) further deepens technical literacy. Instructors noted the machine's utility in encouraging experimentation and iterative problem-solving core traits of engineering education.

This indicates that the machine isn’t just a project artifact it’s a scalable academic asset with cross-disciplinary application, capable of sparking innovation and inquiry from middle school through undergraduate levels.

Another critical dimension explored in this project was the machine’s energy profile and sustainability footprint. The design utilizes low-power components, including micro-servos and salvaged DVD drive stepper motors, consuming under 12W during operation. Continuous drawing tests over a 4-hour period showed stable power draw, with no component overheating or failure, underlining both efficiency and robustness.

Furthermore, the use of recycled mechanical components from discarded DVD drives and repurposed printer rails contributes to an eco-conscious build, reducing electronic waste and showcasing circular engineering principles. This makes the project especially relevant in today’s climate-conscious innovation landscape, where educational tools must model responsible design practices.

As a future step, solar-powered operation through battery integration could be prototyped, allowing for deployment in off-grid or rural learning centres where electricity availability is inconsistent. Such adaptations could enable global reach, particularly in developing regions.

# CHAPTER 6

# CONCLUSIONS & SUGGESTIONS FOR FUTURE WORK

## Conclusions

The Educational CNC Drawing Machine project successfully demonstrates the potential for low-cost automation solutions within academic and training environments. By integrating commonly available recycled components, such as DVD drive stepper motors, Arduino-based microcontrollers, and basic servo actuators, this machine validates the feasibility of bringing CNC-based learning into schools, technical institutions, and maker spaces without heavy financial investment. It bridges the gap between theory and practice by offering a functional, interactive platform where students can visualize the execution of G-code and understand core CNC concepts such as coordinate geometry, toolpath planning, and stepper motor control.

The machine has proven its effectiveness in controlled environments, capable of drawing 2D shapes with commendable precision and repeatability. Through the process of manual homing, pen calibration, and firmware development, a wide array of learning outcomes has been embedded into the system. Users are not only introduced to hardware configuration but also exposed to coding principles, logical debugging, and iterative testing—offering a truly interdisciplinary learning experience.

From a usability standpoint, the system's intuitive operation has been a major success. Once calibrated, the machine performs smoothly across a range of tasks, from simple vector sketches to more complex text and symbol plots. The modular structure of the frame allows for quick repairs and upgrades, which is ideal for experimentation and instructional use. Most importantly, the project promotes a mindset of resourcefulness—showing that technical innovation does not require expensive tools, but rather creativity, critical thinking, and applied knowledge.

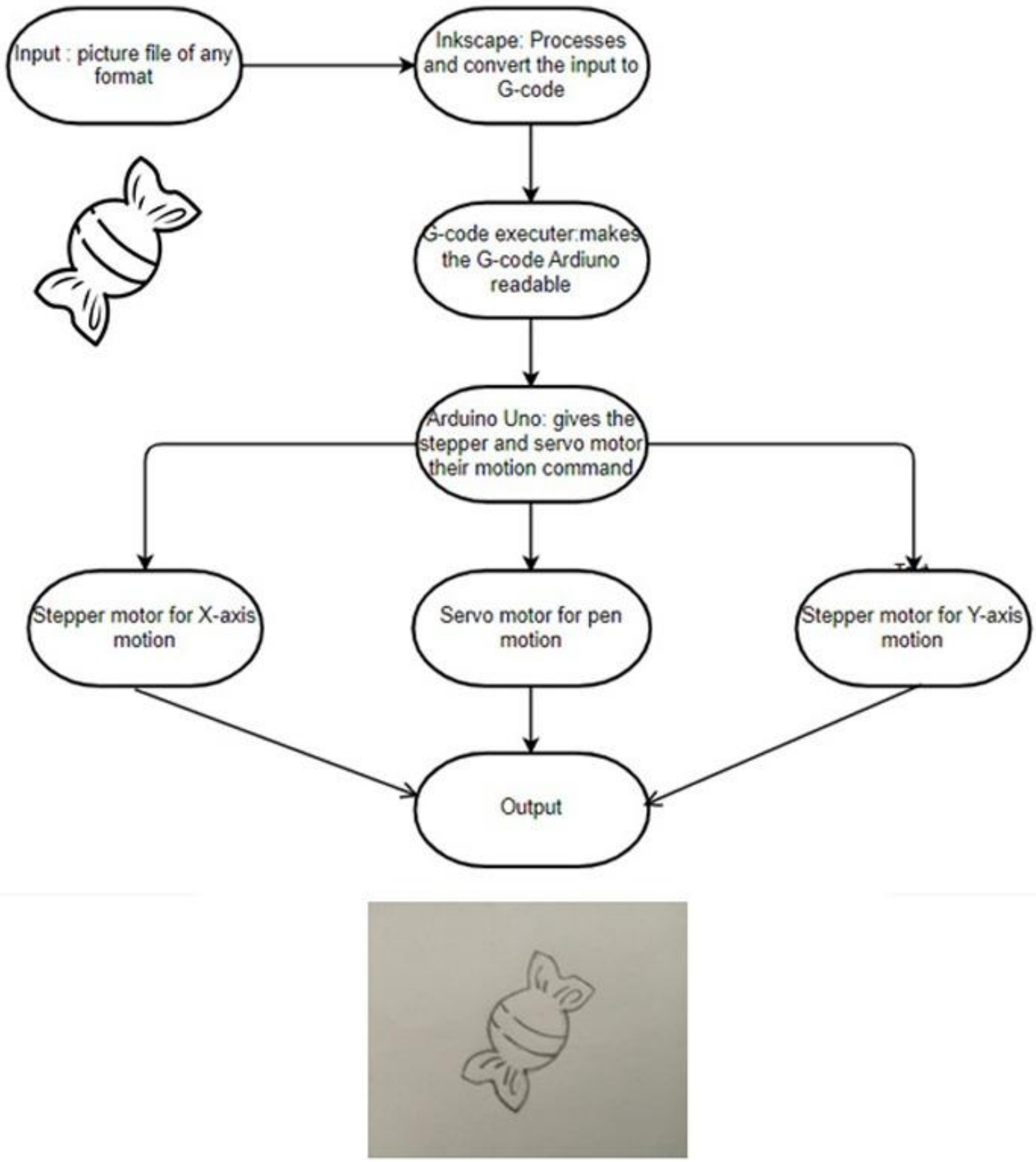


**Figure 6.1 Flowchart with end result**

Moreover, the educational impact of this CNC drawing machine goes beyond its technical performance. It provides students with a tactile and visual means of connecting to abstract engineering concepts. This accessibility enhances their learning curve and boosts engagement, especially for those who struggle with conventional textbook-based approaches. Overall, the machine is a proof-of-concept for how grassroots engineering solutions can democratize access to high-value learning experiences in STEM education.

**Suggestions for Future Work**

While the current version of the machine serves as a successful prototype, several enhancements can be made to broaden its scope, precision, and reliability. Future iterations of the machine should prioritize the integration of higher-torque NEMA stepper motors to allow for more consistent performance under extended workloads and heavier tools. This would improve the machine's robustness and open up applications beyond simple sketching, potentially introducing features like engraving or PCB plotting.

In terms of control electronics, replacing the Arduino Uno with a more powerful board such as the Arduino Mega or ESP32 would allow the system to handle more complex tasks and offer wireless control capabilities via Bluetooth or Wi-Fi. This would pave the way for IoT integration, enabling remote control, cloud-based G-code uploading, and real-time data logging—all critical features in modern smart manufacturing systems. Furthermore, expanding the machine into a 3-axis design would allow for added vertical movement, setting the foundation for 3D printing or CNC milling applications.

On the software side, developing a custom GUI interface could vastly improve user experience. While Pronterface and similar tools are sufficient for technical users, a tailored interface with real-time feedback, drawing previews, and educational prompts would make the system far more approachable for students and instructors. Integrating drawing-to-G-code conversion within the application itself could eliminate dependency on third-party platforms, streamlining the learning experience.

From an educational standpoint, future work can involve creating a curriculum or learning module to accompany the machine. Structured lesson plans, troubleshooting guides, and step-by-step assembly manuals can convert this project into a full-fledged educational kit.

Introducing gamified challenges—such as drawing competitions or toolpath puzzles—can also enhance student engagement and interactivity.

Lastly, incorporating sustainability tracking (such as power consumption meters or carbon offset estimators) would align the machine with global trends in green engineering education. Encouraging students to measure, record, and analyse the machine's environmental impact fosters responsible engineering thinking and supports ESG (Environmental, Social, and Governance) learning goals.

In summary, while the Educational CNC Drawing Machine already delivers meaningful educational value, it possesses enormous potential for expansion. With iterative development, advanced electronics, smarter software, and structured academic integration, this platform could evolve into a comprehensive, industry-relevant learning system. It is not merely a machine—it is a launchpad for innovation, creativity, and hands-on engineering literacy in the next generation of learners.

## REFERENCES


1. G. H. Joseph, “XY Plotter,” Google Patents, 1969. This invention relates to a mechanical system capable of plotting graphical data in X-Y coordinates with motor-driven precision.
2. G. H. Joseph and D. J. Logan, “XY Plotter,” Google Patents, 1966. The patent describes a motorized plotting device using dual-axis motion control to reproduce image or data inputs on paper.
3. M. Sakamoto, R. Magumo, T. Ishihara, and T. Kobari, “XY Plotter for Non-Perforated Paper,” Google Patents, 1990. This system offers an advanced XY plotting technique that allows continuous writing on non-perforated materials, enhancing plotting flexibility.
4. J. Yeiser, “XY Plotter,” Google Patents, 1973. A compact XY mechanism featuring simplified motor control for graphical or data plotting applications using interchangeable writing tools.
5. R. Krishnan, “Selection Criteria for Servo Motor Drives,” IEEE Transactions on Industry Applications, pp. 270-275, 1987. This paper highlights the technical considerations for choosing servo motors in applications demanding precision, speed, and torque consistency.
6. R. D. Bloom, J. V. Gac, and E. T. Kozol, “Programmable Servo Motor Speed Control Apparatus,” Google Patents, 1986. Introduces a servo-based controller that provides variable speed control for plotting and machining tools using programmable logic.
7. H.-K. Park, S.-S. Kim, J.-M. Park, T.-Y. Cho, and D. Hong, “Dynamics of Dual-Drive Servo Mechanism,” ISIE 2001, IEEE, pp. 1996–2000. Explores the coordinated control of dual-drive servo systems, emphasizing torque sharing and synchronization in precision tools.
8. M. Iri, K. Murota, and S. Matsui, “An Approximate Solution for the Problem of Optimizing the Plotter Pen Movement,” Springer, 1982, pp. 572–580. Presents an

algorithm to minimize pen movement in plotters for enhanced drawing speed and reduced mechanical wear.

9. G. Ghiani, D. Laganà, and R. Musmanno, “A Constructive Heuristic for the Undirected Rural Postman Problem,” Computers & Operations Research, vol. 33, pp. 3450–3457, 2006. Proposes a heuristic algorithm applicable in tool path planning for CNC plotters to minimize travel over necessary paths.
10. H. J. Gerber, “Error Correcting System and Method for Use with Plotters, Machine Tools and the Like,” Google Patents, 1971. Describes an early mechanism for automatic error correction in plotting tools by detecting and adjusting for path deviations.

# APPENDIX

## I. BILL OF MATERIAL

| S.NO | COMPONENT | QUANTITY | PRICE |
|---|---|---|---|
| 1 | Stepper Motor | 2 | 300 |
| 2 | Servo Motor | 1 | 187 |
| 3 | Arduino Uno | 1 | 654 |
| 4 | Motor Driver | 1 | 383 |
| TOTAL | | | 1524 |