Design and Analysis of a Computer Numerical Control (CNC) Machine for Precision Manufacturing

[1]Shailendra, [2]Deepak Ahirwar, [3]Nitesh Kumar, [4]Lavkush, [5]Anurag Gupta
[1]Student, [2]Student, [3]Student, [4]Student, [5]Assitant Professor
[1]Electronics And Instrumentation Engineering,
[1]Bundelkhand University Jhansi, Jalaun Uttar Pradesh, India

Abstract

Computer Numerical Control (CNC) machines are commonly used in modern manufacturing industries to achieve precision and automation in machining operations. This paper deals with the design and analysis of a CNC system, which can be used in educational and small-scale manufacturing industries. The proposed system makes use of computer-based numerical control to perform machining operations such as cutting, drilling, and engraving. The proposed system makes use of a set of instructions known as G-codes to perform machining operations. The cutting tool moves in multiple directions to perform machining operations. Stepper motors, a controller unit, and mechanical structures are integrated to perform machining operations with precision and accuracy. The proposed CNC system shows the advantages of automation and precision in modern machining operations. The proposed system can be further improvised by implementing intelligent monitoring and control techniques.

Keywords *— Computer Numerical Control (CNC), G-Code Programming, Stepper Motor Control, Automated Manufacturing, CAD/CAM System, Precision Machining.*

Introduction

Computer Numerical Control (CNC) machines have been an integral component of contemporary manufacturing technology due to their ability to produce complex components with a high degree of precision and accuracy. CNC machines utilize a technology that replaces traditional manual machining operations by using computer commands to run machine tools such as milling machines, lathes, and drilling machines. This technology is quite effective in increasing the efficiency of production operations and the quality of products [1].

Conventional machining operations require skilled workers to run machines and perform machining operations. However, manual operations are not only time-consuming but also affect the quality of products to a certain extent. CNC machines have been designed to overcome the disadvantages of manual operations by using a programming language called G-code to run the cutting tools along multiple axes such as X-axis, Y-axis, and Z-axis [2].

Recent research activities have been focused on developing cost-effective and compact CNC machines that can be used in educational institutes and small-scale industries. The use of microcontrollers and stepper motors along with CAD/CAM software has enabled the development of affordable CNC machines that can be used to perform operations such as drilling, cutting, and engraving [3].

This research focuses on the design and analysis of a CNC machine system intended for precision machining and automated manufacturing applications. The study examines the system architecture, working principles, and potential applications of CNC technology in modern industrial environments.

System Design and Architecture

The proposed CNC machine system would utilize a microcontroller-based control unit to enable the automated motion control. The proposed system would comprise hardware elements such as the Arduino Uno controller, stepper motors, motor driver circuits, and mechanical elements such as belts, shafts, and guide rails. The proposed architecture would enable the motion control along various axes to enable the machining or engraving operation.

The proposed architecture of the overall CNC system would comprise various elements such as hardware design, software design, the CNC controller unit, and the motor driver interface [1][2].

Hardware Design

The hardware part of a CNC machine combines mechanical parts with electronic units to provide precise positioning.

Raspberry Pi Pico W (CU – Controller Unit) The Raspberry Pi Pico W microcontroller board is used as a controller unit to control a CNC machine. The microcontroller board receives control signals and sends pulse signals to a motor driver to control a stepper motor. The board is based on an RP2040 microcontroller and has digital input/output pins used to communicate with a motor driver.

Fig 1. Raspberry Pi Pico W

Stepper Motors

The stepper motors are used to control the precise positioning of the CNC machine's axis. The stepper motors differ from the DC motors, as the former rotate the shaft by a specific step. In the proposed system, the stepper motors are connected to the mechanical drive system with the help of belts and shafts to move the tool carriage along the linear guide rails [4].

Fig 2. Stepper Motors

ULN2003 Motor Driver

The ULN2003 driver module is used to interface the stepper motor with the Arduino controller. The Arduino controller is not capable of providing the required current to the stepper motor; hence, the ULN2003 driver is used to amplify the current signals to the motor windings [5]

Fig 3. ULN2003 Motor Driver

Servo Motor

The servo motor, in this case, works as the Z axis, controlling the vertical movement of the pen instead of the horizontal movement, which is controlled by the X axis and the Y axis. The main role of the servo motor is to control the movement of the pen by either moving it upwards or downwards. When a PWM signal is sent to the servo motor by the Raspberry Pi Pico W via the GP20 pin, it rotates to a certain angle depending on the width of the signal. For example, when a short pulse-width modulated signal of about 1ms width is sent to the servo motor, it rotates to an angle that lifts the pen off the paper, which prevents any drawing. On the other hand, when a long pulse-width modulated signal of about 2ms width is sent to the servo motor, it rotates to an angle that puts the

pen down on the paper. This rotation of the servo motor causes the vertical movement of the pen.

Fig 4. Servo Motor

Mechanical Transmission Components

The mechanical motion system includes Timing belts, Shafts, Linear guide rods, Couplers, Base frame.

These components convert rotational motion from the motor into linear motion required for CNC machining operations.

Supporting Hardware Components

Additional hardware components used in the system include:

- Breadboard for circuit prototyping
- Connecting wires
- Power supply unit
- Mounting brackets and structural frame

These components support electrical connectivity and mechanical stability of the CNC machine.

Software Design

The software system controls the motion of the CNC machine by converting machining instructions into motor control signals.

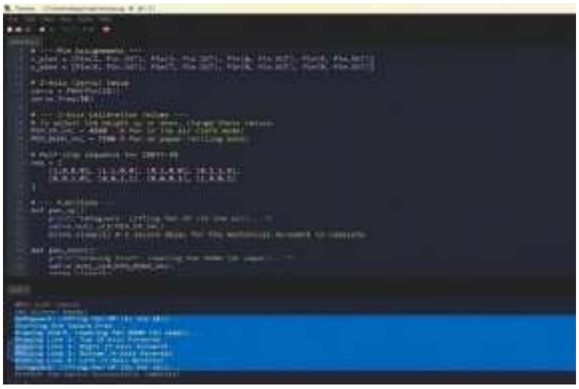


Fig 5. Thonny IDE

Arduino IDE

The Arduino Integrated Development Environment (IDE) is used to program the Arduino Uno microcontroller. The control program defines the motor stepping sequence, speed control, and axis movement required for the CNC machine operation.

G-Code Generation Software

Machining instructions are generated in the form of G-code, which is a programming language used for CNC machines. G-code specifies parameters such as tool movement, speed, and positioning coordinates.

Common G-code generating software are Universal G-Code Sender (UGS), GRBL Controller, Inkscape with G-code plugins (LaserGBRL plugin).

These programs convert CAD designs into G-code commands that control the CNC machine.

Fig 6. UGS Interface (g-code)

CNC Firmware

Firmware such as **GRBL** is used to interpret G-code commands and translate them into stepper motor control signals. The firmware runs on the Arduino microcontroller and ensures synchronized movement of different machine axes [6].

CNC controller

The CNC controller is responsible for coordinating the movement of the machine axes and executing machining instructions. In this system, the Arduino Uno combined with GRBL firmware acts as the CNC controller.

The controller performs the following tasks:

- Interprets incoming G-code instructions
- Generates pulse signals for stepper motors
- Controls motion along X, Y, and Z axes
- Maintains synchronization between machine components

By automating the machining process, the controller improves accuracy, repeatability, and operational efficiency [1].

Fig 7. G-Code Run on UGS

Working Principle

The working principle of Computer Numerical Control (CNC) machines is based on the concept of controlling the machine tools through automated means using numerical instructions. The CNC system is based on the concept of converting the digital design data into machine movements to control the position of the machine tools with high precision. The instructions are generally given in the form of G-code, which includes the coordinates, feed rate, and direction of the machine tools [1].

The working of the proposed CNC machine starts with designing a digital model using Computer-Aided Design (CAD) software. The designed digital model is then converted to machine language using Computer-Aided Manufacturing (CAM) software. The CAM software is used to produce machine language that consists of G-code commands. These G-code commands specify the movement of the tool along various axes of the machine, which include the X-axis, Y-axis, and Z-axis [2].

After generating a G-code program, it is transferred to the CNC controller via a computer interface. In the proposed system, an Arduino Uno microcontroller with GRBL firmware is used as a CNC controller. The CNC controller receives the G-code program and decodes it to convert it into an electric pulse signal to control the movement of the machine via a stepper motor [6].

The reason for the use of stepper motors is the precise incremental motion it provides. With every electrical pulse it receives from the controller, the shaft of the motor turns by a specific angle, referred to as a step. The turning of the shaft is then converted into linear motion of the machine's carriage or the tool holder through mechanical transmission devices like belts, shafts, and guide rails. This process enables the precise positioning of the machine's tool over the workpiece [4].

The ULN2003 driver module is used as the motor driver circuit to control the stepper motors. The driver circuit amplifies the control signals received from the microcontroller. The amplified signals are then used to provide the required current to the motors. By energizing the coils of the stepper motors, the driver circuit enables the precise turning of the shaft of the motor. The turning of the shaft is then converted into the linear motion of the machine's carriage or the tool holder [5].

During operation, the CNC controller continuously interprets the G-code commands and coordinates the movement of the machine axes. The synchronized motion of the motors allows the machine to perform operations such as engraving, drilling, cutting, or milling on the workpiece. This automated control ensures high precision, repeatability, and reduced human error compared to conventional machining processes [1].

Thus, the CNC machine functions as an integrated system where software instructions are translated into controlled mechanical movements through electronic control circuits and mechanical transmission mechanisms.

Experimental Setup

The experimental setup was designed to test the efficiency of the suggested Arduino-based CNC machine prototype. The experimental setup is composed of a computer for creating machining commands, a microcontroller-based controller unit, motor drivers, and stepper motors linked to a mechanical positioning system.

A laptop was used to develop the machining pattern and produce the required G-code commands. A graphical pattern was designed using Inkscape software to produce G-code commands required for CNC machining. The G-code commands were transmitted to the CNC controller using Universal G-Code Sender software.

The CNC controller unit of the suggested system is implemented using an Arduino Uno microcontroller unit. The Arduino Uno microcontroller unit uses GRBL firmware designed for 28BYJ-48 stepper motors and servo motors, which can interpret the G-code commands and translate them into commands for the stepper motors [7].

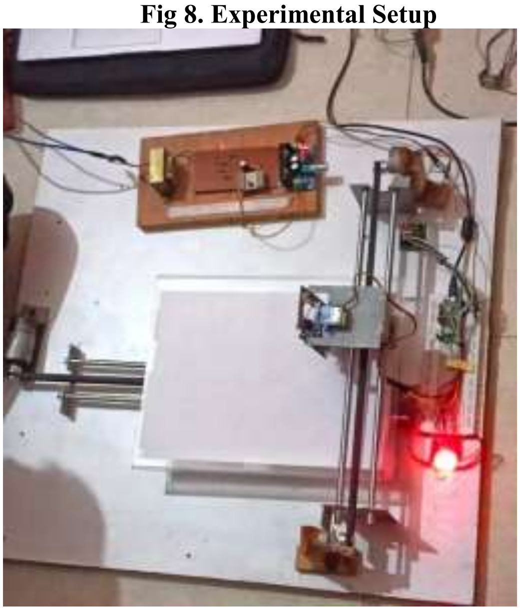
Stepper motors were used to manage the movement of the system along the X-axis and Y-axis. The stepper motors were linked to the Arduino Uno microcontroller unit using ULN2003 motor driver modules to amplify the current required to drive the stepper motors.

The experimental hardware configuration therefore includes:

- Laptop computer for design and control
- Inkscape software for generating tool paths
- Universal G-Code Sender for G-code transmission
- Raspberry Pi Pico W microcontroller as CNC controller
- GRBL-based firmware for motion control
- ULN2003 motor driver modules
- 28BYJ-48 stepper motors
- Breadboard and connecting wires
- Belt and shaft transmission mechanism for linear motion

This integrated system allows automated control of tool movement based on the generated G-code instructions.

Fig 8. Experimental Setup



Results and Discussion

To check the functionality of the developed CNC machine prototype, a graphical pattern was created using Inkscape software. The created graphical pattern was then converted into G-code. The functionality of the created G-code was verified by checking the preview of the created tool path. The preview verified that the created G-code correctly represented the created graphical pattern.

The created G-code was then sent to the CNC machine controller using the Universal G-Code Sender interface. The created G-code was then interpreted by the GRBL firmware running on the Arduino microcontroller. The microcontroller was able to generate step signals to control the rotation of the stepper motors during execution. The motors were able to move sequentially along the X-axis and Y-axis.

The experimental results showed that the CNC machine prototype was capable of:

- Correctly interpreting G-code commands
- Generating motion control signals through the Arduino controller
- Driving stepper motors using ULN2003 motor drivers
- Executing tool paths generated from CAD designs

During the testing process, a soft limit alarm error was experienced while carrying out the execution of the command. This error occurs when the executed command exceeds the axis limits that were predefined in the GRBL configuration file. After resetting the configuration and making the required changes in the configuration parameters, the execution of the program was carried out normally.

As can be noted from the experimental results, the proposed system can be effectively used for the execution of automatic CNC machining operations. The system can be improved in various ways in the future, such as making the system more rigid and adding more axes and using advanced motor drives for the system.

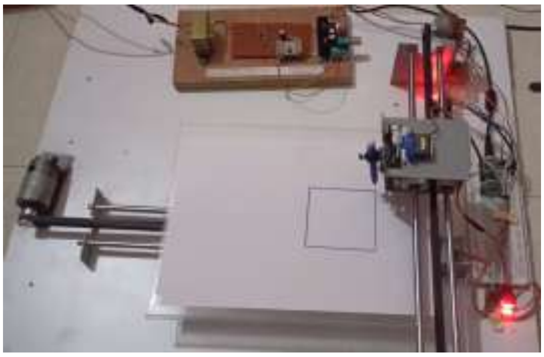


Fig 9. Experimental Setup Output

References

[1] Y. Altintas, Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design. Cambridge University Press, 2012.

[2] Y. Koren, Computer Control of Manufacturing Systems. McGraw-Hill, 2008.

[3] B. Griffiths, Manufacturing Surface Technology. Wiley-ISTE, 2014.

[4] M. P. Groover, Automation, Production Systems, and Computer-Integrated Manufacturing. Pearson Education, 2015.

[5] Texas Instruments, ULN2003A Darlington Transistor Array Datasheet, Texas Instruments, 2018.

[6] S. Jeon, GRBL: An Open Source Embedded CNC Controller, GitHub Documentation, 2020.

[7] Ruizivo, GRBL-28byj-48-Servo Firmware for Arduino CNC Systems, GitHub Repository. Available: https://github.com/ruizivo/GRBL-28byj-48-Servo

Copyright & License:
© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.