ABSTRACT

In a world where inclusivity and accessibility are critical, individuals with upper limb disabilities still face challenges in tasks as fundamental as writing. This project presents the Voice-Activated Cartesian Bot (VOABT)—a low-cost, assistive robotic system designed to convert spoken language into handwritten text, enabling users to write independently using only their voice.

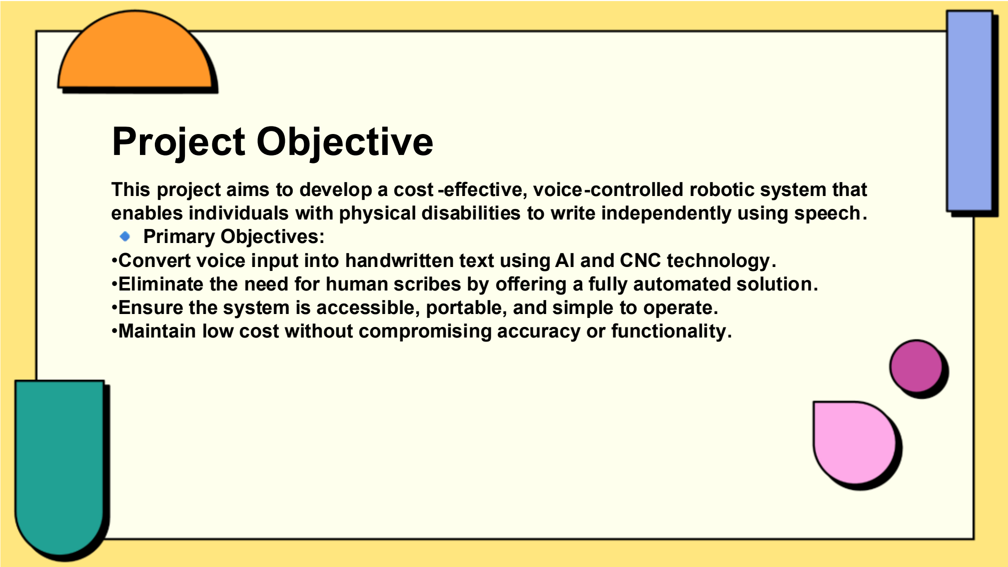
The system utilizes a Cartesian coordinate-based plotting mechanism driven by NEMA 17 stepper motors, controlled via an Arduino UNO with a CNC shield. A Python-based software interface running on a laptop performs live speech recognition using the Whisper AI model, converts transcribed text into G-code using FreeType for font rendering, and communicates directly with the bot over a serial connection to execute precise pen movements.

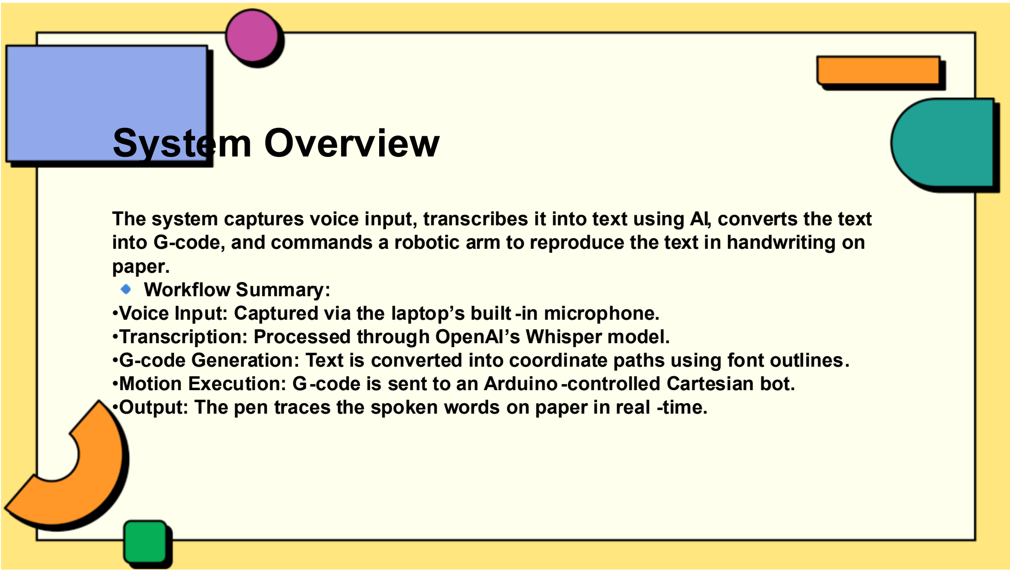
The design emphasizes accessibility, affordability, and ease of use. Voice commands are captured via the laptop's built-in microphone, processed locally, and rendered onto paper using a servo-controlled pen. While physical paper alignment and limit switch integration are still under development, the system has successfully demonstrated full-text transcription and writing capabilities.

This innovation holds the potential to empower people with disabilities by granting them autonomy in communication, education, and expression.

Table of Contents Page

REPORT


Project Objective
This project aims to develop a cost-effective, voice-controlled robotic system that enables individuals with physical disabilities to write independently using speech.
Primary Objectives:
•Convert voice input into handwritten text using AI and CNC technology.
•Eliminate the need for human scribes by offering a fully automated solution.
•Ensure the system is accessible, portable, and simple to operate.
•Maintain low cost without compromising accuracy or functionality.


System Overview
The system captures voice input, transcribes it into text using AI, converts the text into G-code, and commands a robotic arm to reproduce the text in handwriting on paper.
Workflow Summary:
•Voice Input: Captured via the laptop's built-in microphone.
•Transcription: Processed through OpenAI's Whisper model.
•G-code Generation: Text is converted into coordinate paths using font outlines.
•Motion Execution: G-code is sent to an Arduino-controlled Cartesian bot.
•Output: The pen traces the spoken words on paper in real-time.

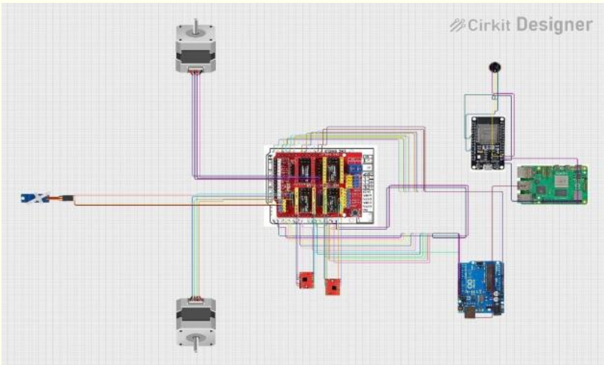
Hardware Components

The system uses accessible, modular hardware commonly found in CNC and 3D printer setups, integrated with a simple robotic frame.

- Core Components:

•Microcontroller: Arduino UNO with CNC Shield
•Motors: 2 × NEMA 17 Stepper Motors
•Motor Drivers: 2 × A4988 Stepper Motor Drivers
•Pen Control: Servo Motor (for pen up/down actuation)
•Mechanical Frame: Threaded rods, smooth rods, bearings, belt, pulleys
•Power Supply: 12V 4A Adapter
•Input Device: Laptop with built-in microphone
•Note: Limit switches are installed but not yet operational; paper is currently held manually.

Circuit Diagram



Software Workflow

The software pipeline handles everything from voice capture to robotic actuation, ensuring a seamless speech-to-writing experience.

- Step-by-Step Process:

1.Voice Capture
Speech is recorded using the laptop's built-in microphone.
2.Speech-to-Text Conversion
Audio is transcribed into text using OpenAI's Whisper model for high accuracy.
3.Text to G-code Conversion
The transcribed text is rendered into vector paths using FreeType, then converted into G-code instructions.
4.Serial Communication
The G-code is sent to the Arduino UNO via PySerial over USB.
5.Motion Execution
GRBL firmware on Arduino interprets the G-code and drives the stepper motors and servo to replicate the text on paper.

Key Features & Achievements

The system successfully integrates voice recognition, motion control, and handwriting reproduction through a modular, efficient pipeline.

🔹 Key Features:

•Fully laptop-driven — no microcontroller required for speech processing
•High-accuracy transcription using Whisper AI
•Supports custom fonts via TrueType (TTF) integration
•Servo-based pen control fully compatible with GRBL
•Modular Python script handling the entire pipeline

🔹 Achievements:

•Voice-to-text functionality tested and validated
•G-code generation from text successfully implemented
•Robotic writing on paper demonstrated
•3D printed components fabricated and integrated

Challenges & Future Improvements

While the core system is functional, certain limitations and enhancements have been identified to improve reliability and usability.

🔹 Current Challenges:

•Limit switches are installed but not yet operational.
•Paper holding is done manually; there's no automated clamping.
•Requires user intervention for initiating voice input.

🔹 Future Improvements:

•Integrate fully functional limit switch logic for motion safety.
•Design a paper clamping mechanism to automate alignment.
•Add voice-controlled commands (e.g., “start”, “clear”, “home”).
•Develop a graphical user interface (GUI) for better control.
•Implement multi-language transcription and text formatting options.

Conclusion

The Voice-Activated Cartesian Bot demonstrates how artificial intelligence and robotics can be combined to create impactful, inclusive solutions for individuals with physical disabilities.

🔹 Key Takeaways:

•Enables independent writing through voice input
•Merges AI transcription with CNC-based actuation
•Cost-effective and accessible design
•Fully modular and customizable for future expansion

“A voice can now leave a mark on paper.”

Conclusion

The Voice-Activated Cartesian Bot represents a fusion of advanced speech recognition, mechanical automation, and inclusive design aimed at addressing a real-world problem—writing accessibility for individuals with upper limb disabilities. This project offers a transformative solution that replaces traditional, human-dependent methods like scribes with a completely voice-driven, automated system that can write on behalf of the user.

By leveraging OpenAI's Whisper model, the system achieves highly accurate speech-to-text transcription, even in real-time. The transcribed text is further processed using the FreeType library to convert it into vector-based G-code, which is then transmitted to a microcontroller-controlled Cartesian robot using PySerial. The Arduino UNO running GRBL firmware interprets these commands and moves a pen mounted on a CNC frame to recreate the spoken words as legible handwriting on paper.

One of the project's key strengths lies in its modular architecture. The entire workflow—from voice input to mechanical output—is handled through a single Python-based script that is both adaptable and scalable. The use of standard hardware components like NEMA 17 stepper motors, A4988 drivers, and a CNC shield makes the system cost-effective and easy to replicate or upgrade.

While the current prototype demonstrates a complete and functional speech-to-handwriting pipeline, certain limitations such as the non-functional limit switches and manual paper alignment highlight areas for further development. Future improvements will focus on integrating safety features, adding GUI-based user interaction, supporting multiple languages, and enabling real-time voice commands for system control.

Overall, this project stands as a testament to how artificial intelligence and mechatronics can be combined to create practical, socially responsible technologies. It not only solves a problem but also enhances the dignity and independence of its users—offering them a powerful tool to communicate, learn, and express themselves freely. The Voice-Activated Cartesian Bot is more than just a machine—it is a step toward a more inclusive and empowering future.

References

1. OpenAI. (2022). *Whisper: Robust Speech Recognition via Large-Scale Weak Supervision*. Retrieved from https://github.com/openai/whisper

2. Python Software Foundation. (2024). *SpeechRecognition Library Documentation*. Retrieved from https://pypi.org/project/SpeechRecognition/

3. GRBL CNC Firmware. (2023). *GRBL: An open-source, high-performance G-code parser*. Retrieved from https://github.com/gnea/grbl

4. FreeType Project. (2024). *FreeType 2 Documentation*. Retrieved from https://freetype.org/freetype2/docs/

5. Arduino. (2024). *Arduino UNO Reference Guide*. Retrieved from https://docs.arduino.cc/hardware/uno-rev3

6. PySerial Developers. (2024). *pySerial: Python Serial Port Access Library*. Retrieved from https://pythonhosted.org/pyserial/

7. A4988 Stepper Motor Driver Datasheet. (2018). Allegro MicroSystems. Retrieved from https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf

8. NEMA 17 Stepper Motor Datasheet. (2020). Wantai Motor. Retrieved from https://www.wantmotor.com/ProductsView.asp?id=155&pid=75&sid=82

9. Universal G-code Sender. (2023). *UGS Platform*. Retrieved from https://winder.github.io/ugs_website/

10. CNC Shield Documentation. (2022). Retrieved from https://protoneer.co.nz/arduino-cnc-shield/