RESEARCH PAPER

Design and Development of an Offline Voice-Assisted CNC Plotter Machine

Asuntha A[1]*, Aditya Atal[2], Priyansh Sahu[2], Shrushti Pinate[2], Akash B[2]
[1]*Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India.*
[2]*Department of Automation and Robotics Engineering, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India.*
* Corresponding author's Email: asunthaa@srmist.edu.in

Abstract: This paper presents the development of an offline, speech-driven CNC plotter capable of printing directly from verbal commands without requiring internet connectivity. The system is built around a Raspberry Pi 4, which performs local speech processing using the VOSK engine to recognize spoken input and convert it into text. This transcribed text is then translated into G-code through a custom Python-based algorithm that leverages Hershey vector fonts for precise character generation. Motion control is managed by an Arduino Uno running GRBL firmware, ensuring accurate execution of the generated commands. Experimental evaluation focused on key performance metrics, including latency, speech recognition accuracy, and positional precision. The system achieved a word recognition accuracy between 90% and 95%, with latency ranging from 2.2 to 3.0 seconds (averaging 2.6 seconds), and a positional accuracy of ±0.08 mm. Overall, the proposed approach demonstrates a cost-effective and efficient solution for offline speech recognition integrated with real-time machine control, offering high accuracy and low latency without reliance on cloud-based services.

Keywords: Voice-assisted CNC, Offline speech recognition, Raspberry Pi, G-code generation, CNC automation

How to cite this article: Asuntha A, Atal A, Sahu P, Pinate S, Akash B. Design and Development of an Offline Voice-Assisted CNC Plotter Machine. Int J Drug Deliv Technol. 2026;16(33s):591-599. DOI: 10.25258/ijddt.16.33s.72

I. INTRODUCTION

CNC machines are becoming highly popular in all forms of manufacturing, fabrication, and educational settings because of their capabilities for precision, repeatability, and efficiency. Recent research on CNC tools and educational platforms highlights increasing efforts to improve accessibility and affordability. However, most interactions with CNC machines still use G-code programming, which is an exceedingly complicated process to learn and use for people without formal training or experience using CNCs (Kaarlela & Outeiro, 2025; Soori et al., 2024; Youns et al., 2024; Irwansyah et al., 2024). This issue creates barriers for newly interested users and delays the adoption of CNC systems by individuals and organizations for learning and prototyping how CNC's can be used. While we see an increase in the use of CNC machines, the way that humans interact with machines is changing to a more natural way of communicating with machines. This trend reflects in the increasing adoption of voice-based interfaces that use automatic speech recognition (ASR) technology, which has become increasingly accurate and user-friendly over the last years (O'Shaughnessy, 2024; Chang et al., 2024). Voice-based interaction allows individuals to perform functions on automated systems without requiring any technical expertise, which makes it a desirable option for use in embedded and assistive applications. Many existing speech-based systems move processes to the cloud; if connected, they do not process commands directly and/or do not allow users to customize their commands. As a result, latency issues may arise from processes that occur over the internet and create privacy concerns because sensitive data may pass through multiple servers before reaching the desired destination.

Greater attention has been focused on offline speech recognition as a viable option for embedded systems to move past these limitations. VOSK is one tool that provides offline speech-to-text capabilities on an inexpensive piece of hardware and enables offline operation without needing any internet connection (Soni, 2025), which is especially beneficial for low-latency applications that rely on privacy of data and functionality while operating offline. Furthermore, inexpensive platforms like Raspberry Pi and open-source control systems like GRBL have enabled low-cost and compact development of automated systems (Halfacree, 2025; Apriadi, 2017). Additionally, Bluetooth-audio devices provide flexible methods of voice input in both embedded and robotic applications (Gupta et al., 2025).

However, limited research has been done in the integration of these capabilities into a unified system. Most studies pertain separately to CNC system development, speech recognition, and embedded controllers but few focus on combining offline speech recognition with text processing and automatic G-code generation as a complete system. Regarding CNC plotters, there has also been minimal investigation of completely offline processes that can accept spoken instruction as input to produce G-code for the final product from the spoken instructions received.

This study presents an offline CNC plotter that converts voice inputs into text using only local processing. The plotter consists of two major components: a Raspberry Pi computer that performs speech processing using the VOSK speech recognition engine and a CNC machine, which in this case consists of an Arduino Uno that runs motion control instructions in the G-code format using the GRBL firmware. A unique Python-based algorithm is designed to convert voice input to G-code, allowing users to create printed text directly from their voices, without using an Internet connection. Unlike other approaches, this study creates a unified system that encompasses voice recognition, G-code generation, and the control of CNC machines all using local processing technologies.

The primary contributions of the study include:

• Development of a fully offline voice-controlled CNC plotter

• Integration of the VOSK speech recognition engine with Arduino-based GRBL motion control.

• Design of a custom algorithm to convert text to G-code.

• Evaluation through extensive experimentation demonstrates the system operates reliably across three performance metrics: accuracy, latency, and print quality.

This paper consists of multiple parts. The methodology and system architecture are covered in Section 2. The workflow and control logic will be explained in Section 3. The workflow is illustrated using a flowchart in Section 4. Experimental analysis will be shown in Section 5. Results and discussion presented in Section 6, concluding with the conclusion present in Section 7

II. MATERIALS AND METHODS

The suggested system is an offline voice controlled CNC plotter. It is built with the combination of embedded hardware and offline speech processing software to design a voice controlled CNC Plotter which does not require an Internet connection to function.

Overall, the system comprises of the following two components:

1. Speech processing system (made up of a Raspberry Pi)

2. Motion controlling system (made up of an Arduino with GRBL firmware).

Raspberry Pi is responsible for speech recognition, command validation, and creation of G-code instructions for the given command by the user whereas the Arduino with GRBL firmware is responsible for processing the G-code instructions to generate motion control signal.

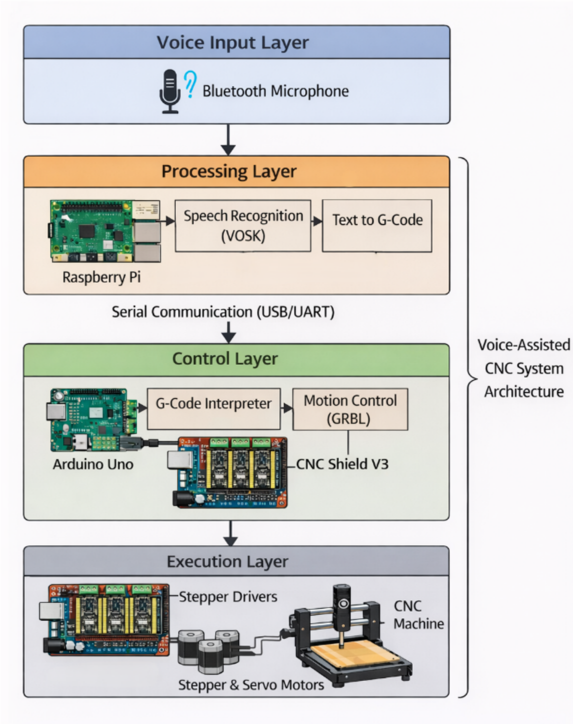

Figure 1. Overall system architecture of the proposed CNC plotter.

2.1. Hardware Subsystem

The hardware elements involved in this system include:

1. **Raspberry Pi 4 Model B (4 GB RAM):** Process unit for voice recognition, command processing and G-code creation (Halfacree, 2025).

2. **Bluetooth Microphone**: Acquires the voice of the user and transmits the audio signal to the Raspberry Pi via Bluetooth.

3. **Arduino Uno:** Functions as the CNC (computer numerical control) controller to control motion through transferring G-code commands from the Raspberry Pi (Allegro MicroSystems, 2024).

4. **A4988 Stepper Motor Drivers**: Interface between the Arduino and the stepper motor to control its current and micro-stepping capability.

5. **NEMA 17 Stepper Motors**: Precise positioning control on the X-Y axes.

6. **MG90S Servo Motor:** Control the pen lifting and lowering for the plotter.

7. **Limit Switches:** Set the maximum working area of the CNC machine and help prevent any mechanical accidents.

8. **Power Supply Unit**: Provide a voltage range to all electronic units.

2.2. Software Subsystem

The software components are as follows:

1. Raspberry Pi Operating System: A Linux-based operating system that allows the management of hardware components and running of software components.

2. Python Programming Language: Runs the logic for speech processing, command interpretation, and generation of G-code.

3. VOSK Offline Speech Recognition: Transcribes human speech into text without requiring an internet connection.

4. GRBL Firmware: GRBL firmware is used for motion control of Arduino UNO by interpretation of G-code. Serial communication libraries allow streaming of G-Code from Raspberry Pi to Arduino (Apriadi, 2017).

5. Arduino IDE configures firmware and controls the activities of the system.

2.3. Description of Core Functional Modules

i) VOSK Speech Recognition Engine: VOSK is an open source offline speech recognition software engine that is specifically intended for use with embedded devices (Soni, 2025). In this case, the VOSK software engine will transform voice commands into textual inputs on the Raspberry Pi device without using any online services. The benefit of this solution is its low-latency and improved security, among others.

ii) Text to G-code Generator: G-code is the standard computer language that is used to program the CNC machine (Giardini et al., 2025). A text to G-code generator algorithm programmed in Python programming language will take care of generating G-code instructions based on the validated text inputs. The G-code instructions will have information about the position of motors, speed of movement, and other parameters. Thus, this solution will remove the need for manual generation of G-code instructions.

2.4. System workflow

The functioning of the proposed method follows a well-defined logic as depicted in Figure 1 that shows methodology or process of the workflow.

i) Voice Acquirement: The spoken command is captured through a Bluetooth microphone connected to the Raspberry Pi 4 board. The duration of the recording may be set anywhere from 1 second to 120 seconds with a default value of 4 seconds and an ability to terminate the recording early by means of pressing keyboard. The recording is saved in mono, 16kHz, on Raspberry Pi in a local WAV file.

ii) Speech to Text Conversion: The recorded WAV (Waveform Audio Format) file is fed to the offline speech recognition module VOSK (v0.3.45, small English language model) installed on the Raspberry Pi 4 processor itself. It recognizes frames of the audio using Kal-di based acoustic model and generates transcription of the audio in the form of text string written in UTF-8 encoding without any internet connection whatsoever [12]. The whole process takes around 1.8 - 2.4 seconds to complete on average for 4 seconds audio files. Punctuation keyword substitutions are performed on the resulting transcript replacing phrases like 'full stop', 'exclamation mark', and 'question mark' into symbols.

iii) Command Interpretation: The text string, after substitution of punctuation symbols, is then checked by the Python command interpreter. Unacceptable characters that do not exist in the Hershey font dictionary are automatically eliminated. The interpreter restricts the use of A4 boundaries (usable area X=10-225mm and Y=10-165mm). It will determine if the text fits into the available area, depending on the current cursor position. In case the current page has run out of space, the program will wait until the user acknowledges that the page is completed and moves on to the next page. There is no possibility of any illegal or unbounded G-code command as all coordinates have been validated.

iv) Generation of G-code: The validated uppercase text string is then inputted into the user-defined text to G-code generation algorithm. For each character in the text, its corresponding stroke representation is extracted from the Hershey vector font dictionary. Each stroke is defined as a set of (x, y) points in a normalized coordinate system ranging from 0 to 8 units horizontally and 0 to 10 units vertically. Each stroke is scaled by a factor of 0.85 and translated to machine coordinates based on the current cursor location. A constant horizontal distance of 10 mm is assigned to each character (CHAR_WIDTH), while a spacing of 5 mm is maintained between characters (SPACE_WIDTH = CHAR_WIDTH / 2). Line spacing is set at 12.5 mm (LINE_HEIGHT = CHAR_HEIGHT × SCALE + 4). Rapid movements for pen-up actions are carried out with G0 code (3000 mm/min speed), whereas pen-down actions employ G1 linear interpolation commands (1500 mm/min).

v) Serial Communication: The G-code that the computer sends to the Arduino Uno through the USB port is done in serial communication mode at 115200 baud using PySerial library software. However, this is only possible once an “ok” signal is received from GRBL.

vi) GRBL Parsing and Motion Planning: The Arduino Uno is fitted with a GRBL firmware of version 0.9 that interprets the G-code and does the planning of motion operations. It includes buffering. Step and direction pulses are transmitted using the firmware to drive the A4988. X and Y axes have been set to 5 steps/mm with direction inversion on. Additionally, homing has been activated, along with the setting of a 2 mm pull-off distance to avoid false triggering. Finally, the maximum travel distances of 235 mm and 175 mm are provided on the X and Y axes, respectively.

vii) Motor Actuation: The A4988 driver changes the control inputs to produce necessary motion of the stepper motor of size NEMA 17 on both X and Y axis. The motion of the pen will be controlled through the MG90S Servo Motor, while pen down would be achieved using M3 (0 degree) and pen up through M5 S90 (90 degrees). The delay period for the servo motor is about 150 ms.

viii) Continuous Processing Loop: Once all the g code instructions are completed, the pen goes up and cursor moves to the next location. At this stage, the entire process starts again, beginning at the stage of inputting voice commands. New set of instructions will be issued for the second row, but cursor would move vertically downward by 12.5 mm. This process would continue until reaching the end of the page and changing it.

2.5. Flowchart

The diagram of the flowchart that illustrates the working operation of the system is shown in Figure 2. It illustrates the entire process starting from its beginning till completion. After powering up all components, it starts with turning on a Raspberry Pi, an Arduino Uno board, a Bluetooth microphone, and CNC equipment. Upon being activated, the microphone captures the user's voice command.

After receiving the voice command from the user, it is sent to a Raspberry Pi computer where it is recognized as speech via the VOSK program using offline speech recognition technology. The speech is transformed to text format and verified for validity. The voice command is considered invalid or if there is unclear speech. In this case, it is ignored, and the system waits for another input signal.

Once the voice command is validated, text-to-G-code transformation takes place using the proprietary Text-To-G-Code software program. The result is sent via serial communication to the Arduino Uno board where it transforms G-code commands into control signals to the A4988 stepper driver, and MG90S servo motor. All parts are united together to carry out movements. While carrying out movements, the system returns to the listening state waiting for another voice command.

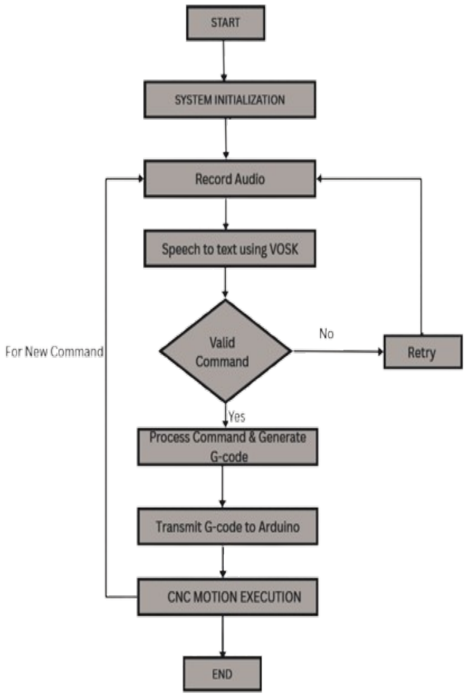

Figure 2. Flowchart of the workflow

III. RESULTS

The experimental verification of the suggested system was performed on four parameters: accuracy of speech recognition, delay time, position error, and the effectiveness of printing. All experiments were conducted in a controlled indoor environment where there was very little noise, with the help of a Bluetooth microphone connected to a Raspberry Pi 4.

Three test scenarios were created, increasing in difficulty from a multiple letter technical term to the famous sentence, “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.” The experiment was conducted for each scenario using the same conditions for a single user.

3.1 Speech Recognition Accuracy

The measure for the accuracy of words was conducted based on the comparison of the words said versus the number of total words said. The correct recognition of a word was determined when there is an exact match to the intended word once the target word was capitalized. In the indoor testing, the accuracy rate for recognition ranged somewhere between 90% to 95%. There was also a test to determine how accurately the punctuation keywords could be recognized so that instead of saying “exclamation point,” the system would be able to translate it into "!".

3.2 End-to-End Latency

Latency was measured from the moment the command stopped being spoken until the first movement of the pen onto the surface of the paper. Speech-to-text conversion was the dominating factor affecting the latency. Average VOSK transcription time varied between 1.8 sec and 2.4 sec (mean value – 2.1 sec ± 0.3 sec standard deviation).

The G-code was generated in less than 50 milliseconds, due to effective memory management. Latency associated with serial communications and GRBL response times amounted to 0.2 - 0.5 sec. Thus, the latency of the entire system fell into 2.2 – 3.0 sec (mean value – 2.6 sec ± 0.4 sec standard deviation), excluding the initial homing process lasting up to 12 seconds.

Performance of the proposed solution is inferior to cloud services, possibly experiencing additional network-related delay.

Total latency of the system may be calculated using equation:

$$T_{total} = T_{STT} + T_{Gcode} + T_{serial} + T_{execution} \quad (1)$$

Where T_STT is speech-to-text processing time, T_Gcode is the G-code generation time, T_serial is serial communication latency, and T_execution is motion activation time. As seen from the results, latency mostly depends on the speech recognition part.

3.3 Positional Deviation

The positional deviation test was conducted by considering two aspects: distance between lines on the Y-axis and distance between characters on the X-axis. Standard line height is calculated as follows:

$$LINE\_HEIGHT = CHAR\_HEIGHT \times SCALE + 4 = 10 \times 0.85 + 4 = 12.5 \text{ mm} \quad (2)$$

Printing of the repeated “GOOD MORNING” phrase showed deviation of around ±0.08 mm with occasional deviations up to ±0.1 mm. Such accuracy is acceptable considering that the stepper motor works at the rate of 5 steps/mm or 0.2 mm/step. Inaccuracies observed could have been caused by mechanical failures such as the flexibility of the belt during direction change and roughness of the paper surface. Character spacing did not vary much while using short sentences along the X-axis. With the increase in the length of sentences, backlash was observed during reverse motion.

3.4 Print Success Rate

Trial was considered successful when: (a) voice input was accurately detected and transcribed, (b) all characters fell in the A4 range and were not truncated, and (c) all characters remained readable on the printed output. Each of the three trials ran successfully, resulting in a 100% print success rate in all controlled tests. No problems arose with software, G-code generation, or serial port communication during any of the trials conducted.

Three representative samples were selected to illustrate how well the system performs on various levels of complexity. Trial 1 used the sample sentence "AUTOMATION AND ROBOTICS" (23 characters, 3 words) to demonstrate correct multi-word transcription and appropriate character printing. Trial 2 used the repetition of "GOOD MORNING" five times in a row to measure the consistent behaviour of the line spacing functionality. Finally, Trial 3 used the full pangram phrase "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG" (43 characters, 9 words, including all 26 letters). Physical outputs of each of these three trials are presented below (Figures 3–5)

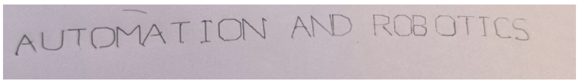

Figure 3. Trial 1 — "AUTOMATION AND ROBOTICS", validating multi-word rendering

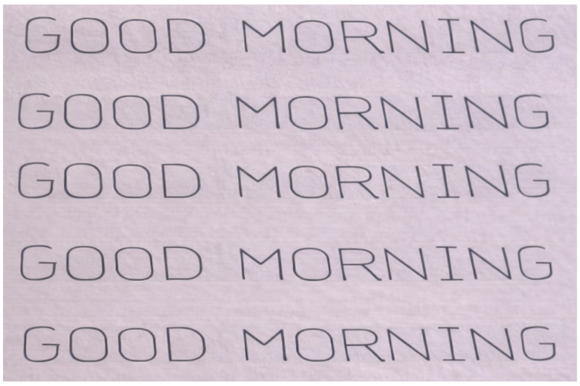

Figure 4. Trial 2 — "GOOD MORNING" repeated five times, demonstrating inter-line spacing consistency of 12.5 mm

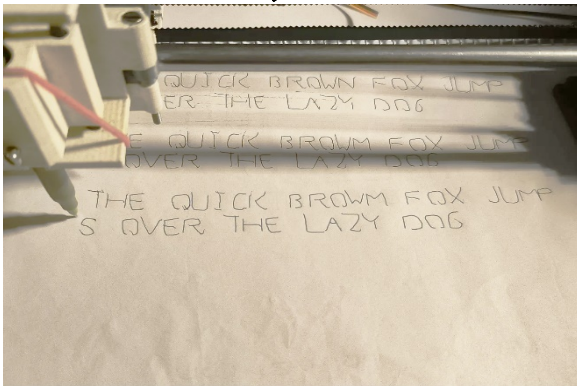

Figure 5. Trial 3 — pangram "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG" across three successive runs, validating the full 26-letter character set and automatic line-wrapping.

IV. DISCUSSION

The above formal experiment gives evidence that the performance of the offline voice-assisted CNC machine under test is both reliable and measurable using the four key metrics specified above. Through the offline voice engine known as VOSK, we observe a word recognition accuracy ranging from 90% to 95% within controlled indoor environments. Translating the recorded speech into written text takes approximately 2.1 seconds (with a standard deviation of ±0.3 s) in a Raspberry Pi 4. While comparative tests with cloud-based systems were not conducted, it is generally known from literature that the average response time in such cases is approximately between 3 to 6 seconds. Therefore, offline solutions perform competitively in terms of latency compared to cloud-based ones.

Regarding the overall command latency, the offline CNC machine demonstrated values from 2.2 to 3.0 seconds with an average latency of 2.6 seconds (with a standard deviation of ±0.4 s). Positional accuracy in terms of deviations from ideal line spacing on the Y axis is measured in ±0.08 mm, indicating repeatable positioning accuracy of ±1.0 mm.

There was no decrease in the print success rate across all tests, except for some trivial typing errors when longer word sequences were typed. Throughout the entire experiment period, there were no malfunctions within the software; there were no G-code malfunctions; and there were no communication problems between the subsystems of the CNC machine.

The findings prove the correctness of the developed G-code and demonstrate reliability of the embedded system control. Communication between the Raspberry Pi board and the Arduino Uno board over the serial connection worked perfectly fine through many operation cycles, without losing any data. The use of homing with the help of limit switches allowed bringing the machine to a definite start point before performing a new operation.

Therefore, positioning errors could not pile up, which meant that outputs were always reproducible. Stepper motors, A4988 drivers, and GRBL firmware provided smooth paths and exact placing of the pen on both X and Y axes.

The proposed system looks like an efficient option for educational, assistive, and fast prototyping applications in cases when the internet is not available at all. Comparison between the proposed system and cloud-based speech recognition services is provided in Table I. Cloud services described in Table I were validated by previous studies (Xu et al., 2021), with their high efficiency and vulnerability to connectivity and surrounding noise mentioned among other factors.

Table 1: Comparison of proposed system with cloud-based speech systems

System	Latency (s)	Internet Requirement	Accuracy
Proposed System	2.2–3.0 (avg 2.6)	No	100%

			(controlled conditions)
Google Speech-to-Text	3–5 (approx)	Yes	95–99%
Amazon Transcribe	3–6 (approx)	Yes	94–98%

However, even as the technology has proven itself in controlled circumstances, there are certain limitations worth noting. For one, the tests were performed by a single individual speaking within a relatively silent environment, and this may affect its ability when there are background noise or other accents. Secondly, the system needs to be able to transcribe relatively clear speech at an average speed; otherwise, this will negatively affect accuracy. Moreover, at this point, the technology supports only the English language.

V. CONCLUSION

The paper proposes a complete offline-based voice command-driven platform integrating speech recognition, text-to-code conversion, and motion control within an embedded device. The experimental findings indicate excellent accuracy and timely performance with minimal latency. Since no cloud computing resources are employed, and responses are delivered promptly, the proposed platform appears appropriate for educational settings and fast prototyping applications. Future work may focus on improving noise robustness and extending support for multiple languages.

VI. ACKNOWLEDGEMENTS

The authors would like to thank the Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Kattankulathur, India, for providing laboratory facilities and technical support for the development and testing of the voice-assisted CNC plotter system.

VII. CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

Allegro MicroSystems 2024, A4988 DMOS microstepping driver with translator and over-current protection datasheet.

Apriadi, ML 2017, ‘Perancangan mesin CNC (computer numerical control) router dengan aplikasi GRBL 0.9 control 3 axis sistem X, Y, dan Z (hardware)’, Doctoral dissertation, Politeknik Negeri Sriwijaya.

Chang, X, Yan, B, Choi, K, Jung, JW, Lu, Y, Maiti, S & Wang, HH 2024, ‘Exploring speech recognition, translation, and understanding with discrete speech units: a comparative study’, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 11481–11485.

Giardini, C, Quarto, M & Pellegrini, G 2025, ‘Improving quality control in 3D printing: a method for G-code-based geometry reconstruction in material extrusion’, Advances in Materials and Processing Technologies, pp. 1–17.

Gupta, S, Mamodiya, U, Hossain, AZ & Al-Gburi, AJ 2025, ‘A framework for Bluetooth-based real-time audio data acquisition in mobile robotics’, Signals, vol. 6, no. 3, p. 31.

Halfacree, G 2025, The official Raspberry Pi beginner’s guide: how to use your new computer, Raspberry Pi Press.

Ibraheem, SM & Adrees, S 2024, Embedded systems for engineers and students, Sheikh Muhammad Ibraheem.

Irwansyah, D, Asnawi, A, Ula, M, Sahputra, I, Sahbana, BN & Arrahman, F 2024, ‘Design and building CNC engraving router machine using Arduino Uno with GRBL control system’, Edelweiss Applied Science and Technology, vol. 8, no. 6, pp. 4570–4584.

Kaarlela, T & Outeiro, J 2025, ‘A cyber-physical machine tool concept for education and workforce training in CNC machining’, Manufacturing Letters.

O’Shaughnessy, D 2024, ‘Trends and developments in automatic speech recognition research', Computer Speech & Language, vol. 83, 101538.

Soni, AA 2025, 'Improving speech recognition accuracy using custom language models with the Vosk toolkit', arXiv preprint, arXiv:2503.21025.

Soori, M, Jough, FKG, Dastres, R & Arezoo, B 2024, 'Robotical automation in CNC machine tools: a review', Acta Mechanica et Automatica, vol. 18, no. 3.

Youns, SS, Anwer, LJ & Ramadhan, MA 2024, 'Design and implementation of a CNC plotter machine using Arduino and CNC shield', in Proceedings of the IEEE International Multi-Conference on Systems, Signals & Devices (SSD), IEEE, pp. 631–636.