Journal of Telecommunication, Electronic and Computer Engineering
ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 17 No. 4 (2025) 7-14
jtec.utem.edu.my
DOI: https://doi.org/10.54554/jtec.2025.17.04.002

Embedded Voice-Controlled AI Assistant for Robotic Arm Operation in Industrial Automation

Nurulfajar Abd Manap[1]*, Teow Chean Yang[1] and Azma Putra[2]
[1]*Centre for Telecommunication Research & Innovation, Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), Melaka, 76100, Malaysia*
[2]*School of Civil and Mechanical Engineering, Curtin University, Kent St. Bentley, Australia.*

Article Info
Article history:
Received Aug 20th, 2025
Revised Sep 29th, 2025
Accepted Oct 21st, 2025
Published Dec 24th, 2025

Index Terms:
Natural Language Processing
Voice Control
Industrial Automation
Large Language Model
Robotic Arm

Abstract
The integration of Artificial Intelligence (AI) and Natural Language Processing (NLP) into Human–Machine Interfaces (HMI) has become increasingly significant in advancing Industry 4.0. This paper presents the design and implementation of an embedded voice-controlled AI assistant for robotic arm operation in industrial automation. The system employs a Raspberry Pi as the embedded platform, combined with Google's Gemini Large Language Model (LLM) to interpret voice commands and execute precise movements on a six degrees-of-freedom (6-DoF) robotic arm through Pulse Width Modulation (PWM) control. The assistant architecture integrates speech-to-text conversion, context-aware NLP processing and servo-based actuation, providing a natural and hands-free interaction between humans and machines. Performance evaluation demonstrates a command recognition accuracy of 90% and an average execution time ranging from 3–10 seconds under laboratory conditions. The results highlight the feasibility of deploying LLM-powered voice assistants on embedded hardware for enhanced efficiency and usability in industrial automation. Future work will address robustness against noisy environments, enabling multilingual support and extending applicability to real-world industrial settings.

*Corresponding Author: nurulfajar@utem.edu.my

I. INTRODUCTION

Voice assistants, conversational agents and other embodied Artificial Intelligence (AI) technologies are becoming increasingly prevalent in both daily life and industrial applications [1]. The advent of voice-controlled AI assistants has the potential to transform industrial automation by enabling intuitive interactions for robotic operations, sensor management and workflow optimization. Within the framework of Industry 4.0, the focus has shifted from merely effective production processes to highly efficient and adaptive systems, requiring closer integration between human operators and machines [2]. Traditional Human–Machine Interfaces (HMIs) such as joysticks, remotes and programmed interfaces, often demand technical expertise and physical input, creating accessibility barriers and limiting usability in dynamic environments.

Recent research has explored integrating voice recognition with industrial Programmable Logic Controllers (PLCs) through containerized Internet of Things (IoT) architectures [3], deploying voice assistants for industrial safety systems [4] and leveraging digital assistants in production and logistics for improved efficiency and decision-making [5]. These studies demonstrate the potential of voice-controlled AI systems to enhance safety and streamline operations. However, key challenges remain, including noise sensitivity, limited contextual understanding, hardware dependency and lack of user accessibility without prior training. Addressing these limitations is essential for realizing robust and scalable AI-driven HMIs in industrial settings.

To bridge these gaps, this paper presents the development of an embedded voice-controlled AI assistant for robotic arm operation in industrial automation. The first contribution of this work is the design and implementation of an embedded AI assistant architecture that integrates speech-to-text conversion, Google's Gemini natural language processing model for contextual interpretation and a servo-driven robotic arm controlled via Pulse Width Modulation (PWM). By deploying the system on a Raspberry Pi, the study demonstrates that advanced AI-driven human–machine interaction can be achieved using eokan affordable and portable embedded platform.

The second contribution lies in the system's ability to perform contextual command handling, enabling it to interpret both direct instructions and implied or indirect commands. Leveraging Gemini's Large Language Model (LLM) capabilities, the assistant can, for example, respond correctly to both "*move left*" and "*move to the opposite of left*". This flexibility highlights the importance of contextual understanding in industrial automation, where operators may issue instructions in varied forms. By reducing dependence on fixed command structures, the system enhances accessibility and usability, even for untrained users.

The third contribution is the experimental validation of the proposed system, carried out through laboratory evaluations and user trials. Ten representative voice commands were tested under controlled conditions to assess recognition accuracy, execution time and repeatability. In addition, engineering students participated in user testing to evaluate usability from an operator's perspective. The results demonstrate that the system achieved 90% accuracy in command recognition and execution times between 3 and 10 seconds, confirming the feasibility of real-time robotic control through an embedded voice-controlled AI assistant.

The remainder of this paper is structured as follows. Section II reviews related work on Natural Language Processing (NLP) models, embedded boards, robotic arm control and evaluation methods. Section III describes the methodology for developing and integrating the proposed system. Section IV presents the experimental results and discussion. Finally, Section V concludes with key findings and directions for future work.

II. RELATED WORK

Voice-controlled AI assistants have emerged as an important technology for enabling natural HMI, particularly in the context of Industry 4.0. Their effectiveness depends on four main aspects: the NLP model employed, the choice of embedded hardware, the robotic system being controlled and the evaluation methods adopted to validate performance. This section critically reviews recent developments in these areas.

A. NLP Models

Voice recognition and speech-to-text (STT) technologies form the foundation of NLP-driven assistants, converting spoken input into structured text for machine interpretation. Early approaches relied on lightweight solutions such as Google Text-to-Speech (gTTS) or Windows Speech Recognition [8,9]. While these methods offered simplicity and no cost, they suffered from slow response times, platform dependency and limited accuracy, making them unsuitable for industrial-grade applications.

More advanced models, such as Aishell2 QuartzNet, provide lower word error rates and allowed fine-tuning [10], though they are often language-specific and lacked the semantic reasoning capabilities of LLMs. Recent work has therefore shifted toward leveraging LLMs such as GPT-4, which not only transcribes speech but also infers intent from ambiguous instructions [11]. However, their deployment often demands significant computational resources, limiting their feasibility for embedded systems.

In contrast, Google's Gemini has shown promise in delivering high accuracy and contextual understanding with relatively faster response times [12]. Its integration into educational and early learning environments demonstrates robustness in handling imperfect speech patterns, suggesting potential suitability for noisy or less controlled industrial contexts. Building on this, the present work adopts Gemini as the NLP engine to achieve both contextual interpretation and embedded deployment efficiency, thereby addressing the limitations of earlier models, which were either too simplistic or computationally demanding.

Table 1 summarizes representative NLP models, highlighting their application domains, strengths and limitations. This comparison illustrates the progression from lightweight, task-specific tools to advanced LLM-driven models and positions Gemini as a balanced option for this study.

Table 1
Comparison of NLP Models for Voice-Controlled Applications

Model	Application Example	Strengths	Limitation
gTTS [8]	PC-based digital assistant	Free, simple, easy to use	Slow response, limited to PC tasks
Windows Speech Recognition [9]	Robot arm control (Windows OS only)	No cost, built-in OS support	Low accuracy, not Linux-compatible
Aishell2 QuartzNet [10]	Voice-controlled elevator (Chinese)	Low word error rate, flexible fine-tuning	No LLM support, language-limited
GPT-based [11]	Robot-assisted assembly planning	High accuracy, contextual reasoning	High computational cost
Gemini [12]	Education & robotics (multi-context)	Fast response, contextual accuracy	Cloud-dependent, cost considerations

B. Embedded Boards

The hardware platform is critical for balancing performance, cost and portability in industrial automation systems. High-end PCs or NVIDIA Jetson platforms provide excellent computational power and AI compatibility [12], [10], but their cost and power consumption restrict scalability in industrial environments. On the other hand, Arduino and STM32 boards are low-cost and energy-efficient [15,16] but lack the processing capacity and library support required for advanced NLP integration.

Table 2
Comparison of Embedded Boards for AI-Assisted Robotic Control

Board Platform	Strengths	Limitation	Suitability for NLP
High-end PC + GPU [12]	Maximum performance, supports heavy AI	Expensive, power-hungry	High (but not portable)
Raspberry Pi [13,14]	Affordable, portable, strong community	Limited computing power	Moderate–High (for lightweight AI)
Arduino [15]	Low-cost, low-power, compact	No advanced AI library support	Low
STM32 [16]	Power-efficient, strong sensor integration	No built-in Wi-Fi, limited AI support	Low
Jetson Nano [10]	Strong edge AI performance	Relatively costly, less community	High

The Raspberry Pi has therefore emerged as a balanced alternative, combining affordability, community support and sufficient computational power for lightweight AI applications [13,14]. Previous works have demonstrated its feasibility in IoT-based voice assistants and robotic arm control, though limited by simpler NLP engines. The present study extends these efforts by deploying Gemini on Raspberry Pi, thereby combining the low-cost portability of embedded boards with the contextual reasoning capabilities of LLMs, a contribution that has not been adequately

explored in prior research. Table 2 highlights commonly used embedded boards and explains why the Raspberry Pi represents a balanced choice for implementing a voice-controlled AI assistant.

C. Robotic Arm Integration

Robotic arms serve as an ideal platform to demonstrate the effectiveness of voice-controlled AI assistants in industrial settings. Commercial arms, such as UFactory Lite 6 [12], offer integrated support and wireless communication, but their cost and proprietary ecosystems limit accessibility for research and prototyping. In contrast, custom-built robotic arms based on servo motors and controllers [17] provide a flexible and affordable testbed, although they require more careful system integration.

Most existing works have focused on either demonstrating voice-triggered actuation or validating robotic precision in simplified tasks. This study advances the field by integrating a custom-built six degrees-of-freedom (6-DoF) robotic arm with Gemini-driven voice interpretation, demonstrating that even low-cost mechanical designs can achieve robust control when coupled with advanced NLP.

D. Evaluation Methods

Evaluation plays a central role in determining the effectiveness and practicality of voice-controlled AI assistants. Prior works have adopted diverse strategies, often focusing narrowly on speech recognition accuracy while overlooking broader system-level performance. For example, Sikorski et al. [18] assessed accuracy by testing a limited set of predefined commands, reporting intent interpretation rates of around 90% but providing little insight into execution time or user adaptability. Similarly, Koc et al. [19] and Xu et al. [20] concentrated primarily on recognition robustness under varying noise levels and signal-to-noise (SNR) ratios, achieving accuracies between 87% and 94%. While valuable, these evaluations provide only a partial view of real-world usability, as industrial environments require not only recognition robustness but also timely and repeatable execution of machine actions.

Other recent studies involving GPT-based robotic assistants [11] demonstrated the importance of measuring execution time alongside accuracy. Forlini et al. [11], for instance, reported that assembly tasks could take up to 20 seconds due to delays in object recognition and contextual reasoning, highlighting the trade-off between intelligence and speed. Aguilera et al. [12] extended this by incorporating repeatability testing with 311 voice instructions, showing that the gTTS model achieved faster runtimes than GPT-based methods, but with less semantic flexibility. These works highlight the challenge of balancing recognition accuracy, execution time, and contextual robustness in NLP-powered robotics.

Building on these insights, the present study adopts a dual-phase evaluation strategy that combines quantitative and user-centered assessments. In Phase 1, ten representative voice commands of varying complexity were tested repeatedly to evaluate both recognition accuracy and execution time, ensuring that performance metrics are systematically benchmarked. In Phase 2, user trials with engineering students are conducted to assess how untrained operators interacted with the system in practice. This approach not only captures traditional metrics such as accuracy and latency but also provides insights into system usability and accessibility, which are often neglected in prior evaluations. By integrating both controlled testing and user-based validation, the study establishes a more comprehensive framework for assessing embedded voice-controlled AI assistants in industrial automation.

Table 3 summarizes representative evaluation approaches in the literature, highlighting their focus areas and limitations. While these studies have advanced the field by examining recognition accuracy, noise robustness, or execution time, most fall short of providing a comprehensive view of system usability, particularly in industrial settings where reliability and operator adaptability are crucial. In contrast to these prior work, the present study adopts a dual-phase evaluation strategy that systematically measures recognition accuracy, execution time, and user usability. By combining controlled command-based testing with trials involving untrained operators, this approach goes beyond the scope of existing evaluations, offering a more holistic framework for assessing embedded voice-controlled AI assistants in industrial automation.

Table 3
Comparison of Evaluation Methods in Prior Works

References	Evaluation Focus	Strengths	Limitations
Sikorski et al. [18]	Intent interpretation (limited commands)	Demonstrated command accuracy (~90%)	No execution time or usability testing
Koc et al. [19]	Noise robustness (SNR variation)	Assessed recognition under noise	No execution time or real-world testing
Xu et al. [20]	Noise interference (siren test)	Reported high accuracy (94% at 0–15 dB)	No task-level validation
Forlini et al. [11]	Assembly tasks with GPT-based assistant	Measured accuracy & execution time	Long delays (up to 20s), high resources
Aguilera et al. [12]	Repeatability & child usability testing	Evaluated contextual robustness	High-end setup, not embedded

In summary, prior studies have made significant progress in advancing voice-controlled AI assistants, but important gaps remain across multiple dimensions. Existing NLP models either offer limited functionality, as seen in lightweight engines such as gTTS or Windows Speech Recognition [8,9], or demand high computational resources, as in GPT-based systems [11]. While embedded boards like Raspberry Pi have been explored for affordability and portability [13,14], most implementations used simple NLP models, thereby limiting contextual command interpretation. Similarly, robotic arms have been employed as test platforms, but many works relied on costly proprietary systems [12] or simplified mechanical designs [17] without fully integrating advanced NLP capabilities. Finally, evaluation methods often emphasized recognition accuracy alone [18,19,20], neglecting execution time and usability, which are essential for practical deployment in industrial automation.

Addressing these limitations, the present study combines Gemini's LLM with a Raspberry Pi embedded platform to deliver contextual command interpretation and real-time control of a 6-DoF robotic arm through PWM. Furthermore, the evaluation strategy extends beyond recognition accuracy by incorporating execution time and user trials, offering a more comprehensive measure of system effectiveness. This

positioning highlights the novelty of the proposed work: demonstrating that advanced NLP capabilities can be effectively deployed on low-cost embedded hardware to achieve intuitive, efficient, and practical human–machine interaction for industrial automation.

III. METHODOLOGY

The methodology of this project is structured around six sequential stages to guide the design, integration and validation of the embedded voice-controlled AI assistant. The process began with requirements analysis and system design, where both hardware and software specifications were defined, and an overall system architecture developed. This was followed by the integration of Google's Gemini model, which involved configuring speech-to-text transcription and natural language processing capabilities on the Raspberry Pi platform. The third stage focused on constructing a 6-DoF robotic arm, which served as the primary hardware for demonstrating voice-based control. Once the arm was assembled, the fourth stage involved Gemini context training, where varied command phrasings were mapped to specific robotic functions to improve contextual interpretation. The fifth stage, system integration, combined the audio processing, NLP and robotic control components into a unified workflow, illustrated through a block diagram. Finally, the sixth stage covered performance evaluation and fine-tuning, with testing under both controlled and user-based trials to assess recognition accuracy, execution time, and usability. Each stage is described in detail in the following subsections.

A. Requirements Analysis and System Design

The first stage focused on analyzing the system requirements and defining the overall architecture of the embedded voice-controlled assistant. Two primary requirements were identified: the hardware platform and the software framework. Raspberry Pi 4 Model B (4GB RAM) was selected as the main controller due to its affordability, portability and compatibility with widely used AI and robotics libraries. On the hardware side, the system required reliable audio input, servo control capability and sufficient processing power to handle natural language processing tasks. On the software side, the framework needed to support speech-to-text conversion, integration with Google's Gemini NLP model and efficient robotic actuation through PWM.

Based on these requirements, an architecture blueprint was developed to ensure seamless communication between components and alignment with Industry 4.0 principles of intelligent human–machine interaction. As shown in Figure 1, the blueprint illustrates the data flow from audio input, speech-to-text transcription and Gemini NLP contextual processing, through intent parsing, to either conversational feedback or robotic actuation via PWM control on the Raspberry Pi. Latency checkpoints (T_1–T_3) and angle validation using a digital protractor (±2°) were also embedded in the system design. This blueprint guided subsequent development stages, ensuring that voice commands could be captured, processed and executed in real time for industrial automation tasks.

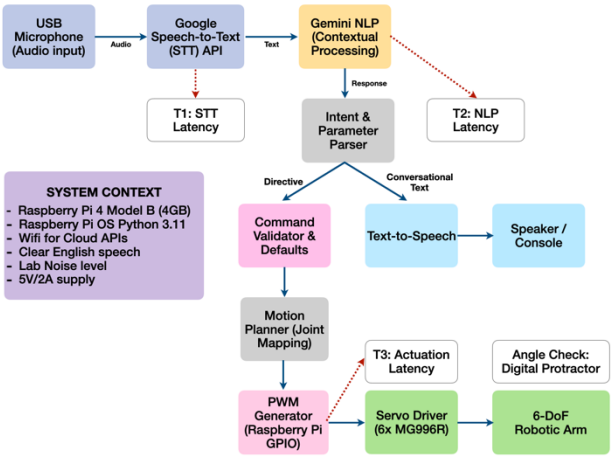


Figure 1. Architecture blueprint of the embedded voice-controlled AI assistant

B. Gemini Integration

The second stage focused on integrating the Raspberry Pi with Google's Gemini model to enable natural language understanding. A two-step process was adopted. First, voice input from a USB microphone was transcribed into text using the Google Cloud STT API, which provided high accuracy and reliability for speech transcription. This approach was necessary because Gemini primarily operates on text-based input and does not directly process audio streams while maintaining conversational history. By separating speech transcription from semantic interpretation, the system achieved more accurate and consistent results.

The transcribed text was then forwarded to Gemini for contextual parsing and intent recognition. Integration was implemented through the *google-generativeai* Python library, with the Gemini-2.0 Flash model selected for its ability to generate coherent and context-aware responses. This configuration ensured that the assistant could not only react to explicit instructions but also interpret implied or indirect commands, thereby enhancing natural and flexible human–machine interaction.

C. Robotic Arm Construction

The third stage involved building and integrating a 6-DoF robotic arm to demonstrate hardware control via the embedded assistant. The arm was constructed using an Acrylonitrile Butadiene Styrene (ABS) frame and six MG996R servomotors, each corresponding to a joint: base, shoulder, elbow, wrist, pitch, wrist roll and gripper. The servos were connected to the Raspberry Pi's GPIO pins, with actuation achieved through PWM signals. Figure 2 shows the assembled robotic arm prototype.

To simplify servo control and ensure reliable actuation, the *gpiozero* Python library was employed. This library provides a high-level interface for GPIO-based devices, abstracting low-level pin control into programmable functions. Commanded angles were verified against actual movements using a digital protractor, with an estimated accuracy of ±2°, ensuring that PWM signals accurately reflected physical movements.

By integrating *gpiozero* with the assistant's command interpretation layer, the robotic arm was controlled directly in response to user voice instructions. This approach ensured that both simple and compound commands were translated

into precise joint movements, thereby validating the assistant's capability for real-time industrial task simulation.

Figure 2. Six degrees-of-freedom (6-DoF) robotic arm prototype constructed with ABS frame and MG996R servomotors

D. Gemini Context Training

The fourth stage focused on adapting Gemini to interpret voice commands within the context of robotic arm operation. Since Gemini is a general-purpose large language model, it was adapted with contextual prompts and explicit command-function mappings to ensure reliable interpretation of industrial instructions. These mappings defined equivalences between different phrasings and their associated robotic actions. For instance, both *"move left"* and *"shift to the opposite of right"* were mapped to the same base joint rotation. When parameters such as angle were not specified, the system assigned default values (e.g., a 45° rotation for directional commands).

Manual evaluation was performed to refine these mappings. Commands were issued repeatedly, and each response was logged as *PASS* (correct or acceptable action) or *FAIL* (incorrect or no action). This iterative process improved consistency and reduced errors caused by ambiguous phrasing.

Representative examples of these mappings are shown in Table 4, which illustrates how different voice instructions were normalized into robotic functions. By embedding contextual awareness into Gemini's responses, the assistant handled not only direct commands but also implicit or varied phrasings, improving usability in practical scenarios.

Table 4
Example mappings of user commands to robotic arm functions

Example command	Mapped Context or Function	Response Mode
"*Move left*" or "*Shift opposite of right*"	Base joint rotates 45° left	Direct action
"*Move elbow to -45 degrees*"	Elbow joint rotates to -45°	Direct action
"*Explain about AI*"	Gemini generates explanatory text	Conversational

E. System Overview

The overall workflow of the embedded voice-controlled AI assistant is illustrated in Figure 3. The process begins with audio input captured by a USB microphone, which is transcribed into text using the Google STT API. The transcribed text is then forwarded to Gemini via the *Google-Generative AI* library for contextual interpretation. If the interpreted response contains a control directive, the Raspberry Pi translates it into PWM signals that actuate the servomotors of the 6-DoF robotic arm. If the response is conversational, the assistant provides feedback through text or audio output.

The architecture blueprint in Figure 1 complements this workflow by showing the internal modules and evaluation checkpoints (T_1–T_3), while Figure 3 highlights the high-level system flow. The design was developed and tested under controlled laboratory conditions, although real-world deployment may encounter additional challenges such as background noise, varied speech patterns or hardware constraints. Nevertheless, the design demonstrates the feasibility of deploying advanced NLP on embedded hardware for industrial automation tasks.

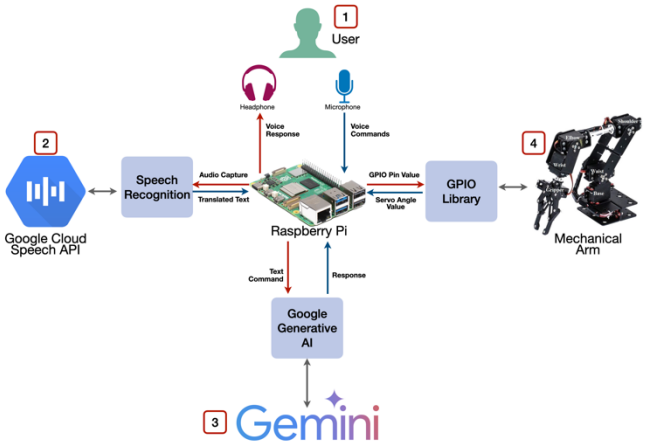


Figure 3. System Overview of the Embedded Voice-Controlled AI Assistant

F. Performance Evaluation and Fine-Tuning

The final stage involved optimizing the system and validating its performance through structured evaluation. Two primary metrics were adopted based on findings from prior studies[11,12,18]: recognition accuracy and execution time. Evaluation was conducted in two phases.

In Phase 1 (Controlled Testing), ten representative voice commands were selected to reflect a mix of direct, compound, and contextual instructions. Each command was issued three times, and the outcomes were logged as either *PASS* (correct or acceptable execution) or *FAIL* (incorrect or no execution). Recognition accuracy was calculated based on the majority outcome across the three trials. For execution time measurement, latency was defined as the end-to-end delay from audio input to robotic arm movement, which incorporates speech-to-text processing (T_1), Gemini interpretation (T_2) and PWM-driven actuation (T_3). Failed attempts were repeated until the correct action occurred, ensuring consistency in latency reporting.

In Phase 2 (User Testing), two engineering students were invited to operate the system without prior training. They issued a subset of commands and the system's responses were measured against a maximum time threshold of 10 seconds. Commands executed correctly within the time limit were considered *PASS*, while failures or delayed responses beyond the threshold were marked as *FAIL*. This stage provided additional insights into system usability for inexperienced

operators, simulating conditions closer to real-world deployment.

The commands used for Phase 1 testing are summarized in Table 5, along with their expected outcomes. By combining accuracy, latency and user experience measurements, this dual-phase evaluation ensured that the system was benchmarked under controlled conditions and validated for practical deployment. This framework provides a more comprehensive assessment compared to prior studies, which often emphasized a single metric such as recognition accuracy or noise robustness.

Table 5
Voice commands used for Phase 1 evaluation

No	Command	Expected Output
1	Move the arm to left	Base joint rotates left
2	Move the shoulder to 70 degrees	Shoulder joint rotates to 70°
3	Move the elbow up a little bit, then sway to the back	Elbow +20°, Shoulder −45°
4	Move to the opposite of left	Base joint rotates right
5	Rotate the wrist and then move up	Wrist rotates, arm moves upward
6	Come closer	Shoulder joint moves forward 45°
7	Tell me about ducks and reset the arm	Gemini provides information, arm resets to default
8	30 is on your right, 20 is on your left, move to bigger number	Arm moves to the right
9	A fire is happening in front of you, move away	Arm moves backward or away from current position
10	Don't move	Arm remains in current position

IV. RESULTS AND DISCUSSION

This section presents the results of the development and evaluation of the embedded voice-controlled AI assistant. Outcomes are analyzed in terms of system integration, command recognition accuracy, execution time and user usability.

A. Prototype Implementation

The final integrated prototype of the embedded voice-controlled AI assistant is shown in Figure 3, in which the Raspberry Pi, microphone and robotic arm are integrated as a complete system. The prototype demonstrates seamless interaction between the hardware and software components, enabling the robotic arm to execute tasks directly in response to user voice commands. To illustrate functionality, the system was tested with compound instructions multi-joint movement. Figure 4 depicts the arm position before and after the execution of the command *"Move elbow up a little bit and then sway back"*. The results confirm that the assistant is capable of translating complex spoken input into coordinated robotic actions, validating the integration of speech-to-text processing, Gemini-based interpretation and servo actuation through PWM.

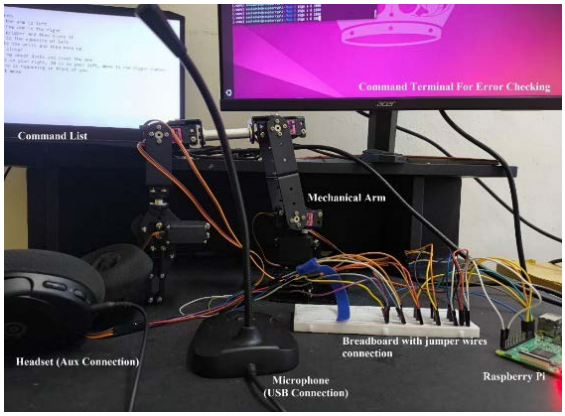
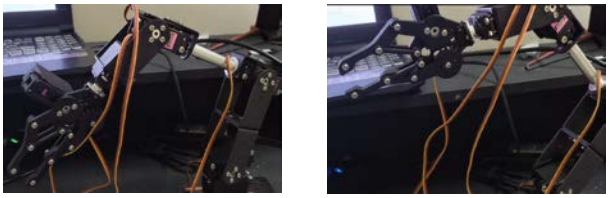


Figure 3. Final Prototype of the embedded AI assistant with robotic arm



(a) (b)

Figure 4. Robotic arm position (a) before and (b) after command execution

B. Confusion Matrix Analysis

Controlled testing was conducted using the ten representative commands listed in Table 5. Accuracy results are presented in Table 6, where each command was attempted three times and the majority outcome recorded.

Out of ten commands, nine were successfully executed, yielding an overall accuracy of 90%. Complex contextual instructions (e.g., Command 9) produced more errors and often required clarification or repetition. These results are consistent with prior works such as Sikorski et al. [18], who reported ~90% interpretation accuracy, but extend beyond them by including compound and contextual commands.

Table 6
Phase 1 Evaluation Results (Recognition Accuracy)

Command	Trial 1	Trial 2	Trial 3	Final Result
1	PASS	PASS	PASS	PASS
2	PASS	PASS	PASS	PASS
3	PASS	PASS	PASS	PASS
4	PASS	PASS	PASS	PASS
5	FAIL	PASS	PASS	PASS
6	PASS	FAIL	PASS	PASS
7	PASS	PASS	PASS	PASS
8	PASS	FAIL	PASS	PASS
9	PASS	FAIL	FAIL	FAIL
10	PASS	PASS	PASS	PASS

Execution time measurements are shown in Table 7. As expected, simple commands (e.g., Commands 1 and 2) averaged less than 4 seconds, while multi-step contextual instructions (e.g., Commands 7–9) required longer, with a maximum average of 10.07 seconds. Despite this variation, all commands met the target requirement of under 10 seconds, aligning with performance thresholds suggested in Forlini et al. [11] for practical human–robot collaboration.

Table 7
Phase 1 Evaluation Results (Execution Time in Seconds)

Command	Trial 1	Trial 2	Trial 3	Average Time
1	3.42	3.56	3.75	3.58
2	3.21	3.77	3.34	3.44
3	5.34	5.21	5.33	5.29
4	3.91	3.67	3.55	3.71
5	5.11	4.67	4.98	4.92
6	2.99	3.45	3.68	3.37
7	9.78	10.34	10.11	10.07
8	8.21	8.86	8.41	8.49
9	7.78	8.82	8.56	8.38
10	2.21	2.17	2.45	2.28

These results demonstrate that the assistant performs reliably within industrially acceptable execution times, even for multi-step commands. Compared to Aguilera et al. [12], who reported child-directed usability but did not measure execution latency, this study provides a more comprehensive performance profile.

C. Phase 2 Evaluation

In addition to controlled testing, user-based evaluation was conducted to examine the system's usability when operated by individuals without prior exposure to its design. Two engineering students were invited to interact with the assistant under laboratory conditions. They issued a subset of representative commands (Commands 1, 4, and 7 from Table 5), which included simple actions (base rotation), direct positional changes and a compound instruction combining conversational output with robotic reset. The objective of this phase was not to establish statistical generalizability, but rather to provide preliminary insights into how untrained users engage with the system.

The results of the user trials are summarized in Table 8. Both participants executed simple commands (Commands 1 and 4) successfully within the 10-second time threshold. Performance diverged for the more complex Command 7: one participant exceeded the time limit, while the other completed the task in 9.87 seconds. These findings highlight the variability introduced by user interpretation and command phrasing, which is consistent with studies that note the sensitivity of speech-driven systems to input variations [20].

Table 8
Phase 2 Evaluation Results

Student	Command 1 (s)	Command 4 (s)	Command 7 (s)	Result
1	3.12	5.65	>10 (FAIL)	2/3 PASS
2	3.67	7.44	9.87 (PASS)	3/3 PASS

Although limited in scale, this evaluation demonstrates that the system can be operated by untrained users, but task complexity directly affects performance and consistency. The inclusion of user testing with engineering students is appropriate for this proof-of-concept stage, as it reflects typical end users who may apply such a system in educational or laboratory automation contexts. For broader deployment, larger-scale trials with diverse participants in realistic industrial environments are necessary to validate general usability and robustness.

D. Discussion

The evaluation results demonstrate that the embedded voice-controlled AI assistant achieved 90% recognition accuracy with execution times ranging from 2 to 10 seconds. These outcomes indicate that the system fulfills the design objectives of enabling real-time interaction while maintaining reliability across both simple and compound commands. Compared with prior studies that often focused narrowly on recognition accuracy [18], noise robustness [19] or execution latency [11], this study provides a more comprehensive assessment by jointly considering accuracy, latency and user usability. Furthermore, the integration of Google's Gemini model on a Raspberry Pi platform illustrates that advanced natural language processing can be effectively deployed on low-cost embedded hardware, an aspect that has been underexplored in earlier research [12].

Nevertheless, several limitations were identified. First, noise robustness was not extensively evaluated, as the experiments were conducted in controlled laboratory settings. Real-world industrial environments are typically subject to background noise and overlapping speech, which may degrade recognition performance. Future work should incorporate noise filtering techniques and robustness testing under varied acoustic conditions. Second, the current system is restricted to English-only commands, which constrains its applicability in multilingual industrial contexts. Extending the system to support multiple languages, particularly those relevant to local industrial environments, would broaden its usability. Third, while the prototype successfully operated a six degrees-of-freedom robotic arm, the use of low-cost components such as MG996R servomotors may limit precision and long-term durability. Integrating industrial-grade actuators and sensors will be required for reliable deployment in actual factory environments.

In terms of usability, the Phase 2 evaluation revealed variability in user performance, particularly for compound commands. This aligns with observations in prior studies [20] that speech-driven systems are sensitive to variations in phrasing and user experience. The choice of engineering students as test participants is appropriate at this proof-of-concept stage, as they represent technically literate but untrained users who mirror potential adopters in educational or research settings. However, the small sample size of only two participants means the results should be interpreted as pilot-scale findings rather than statistically generalizable outcomes. Larger-scale trials with more diverse participants are required to validate system adaptability.

Resource consumption also warrants consideration. Although the Raspberry Pi 4 successfully supported STT transcription, Gemini interpretation and PWM control, the observed 2–10 second latency reflects the computational constraints of embedded platforms. Future work could explore optimization strategies such as model pruning, edge-optimized NLP frameworks or selective offloading of tasks to more capable edge servers.

Overall, the findings confirm that the proposed assistant is suitable for laboratory-scale industrial automation tasks and highlight clear directions for enhancement. Addressing noise robustness, extending multilingual support and conducting larger-scale usability trials represent the next steps toward achieving a more robust and generalizable

embedded voice-controlled AI assistant for Industry 4.0 applications.

V. CONCLUSION

This paper has presented the design and implementation of an embedded voice-controlled AI assistant for robotic arm operation in industrial automation. The system integrated Google's Gemini natural language model with a Raspberry Pi platform to interpret voice commands and control a 6-DoF robotic arm through PWM. The methodology encompassed requirement analysis, system integration, context training and dual-phase evaluation to assess both technical performance and user usability.

The prototype achieved 90% recognition accuracy with execution times between 2 and 10 seconds, demonstrating reliable real-time interaction under laboratory conditions. Compared with prior works, the proposed system provides a more comprehensive evaluation by jointly addressing recognition accuracy, latency and user experience, while also demonstrating that advanced NLP capabilities can effectively be deployed on low-cost embedded hardware.

Despite these achievements, several limitations remain. The system was tested under controlled environments, making noise robustness an area for future testing. Its reliance on English-only commands restricts broader applicability and the use of low-cost servomotors limits precision and durability for real industrial deployment. In addition, user testing was limited in scale, reflecting preliminary insights rather than generalizable conclusions. The observed latency further highlights the computational constraints of running cloud-based NLP on embedded hardware, highlighting the need for resource optimization in future implementations.

Future work should address enhancing noise resilience, extending multilingual support, and incorporating industrial-grade hardware for improved precision and durability. Broader usability studies with diverse participant groups in realistic industrial settings are also needed to validate scalability. Optimization strategies such as model pruning, edge-friendly LLMs or hybrid cloud–edge architectures should also be investigated to reduce resource overheads and improve responsiveness.

In summary, this study demonstrates the feasibility of deploying LLM-powered voice assistants on embedded hardware for industrial automation, establishing a foundation for more natural, efficient, and accessible human–machine interaction in Industry 4.0 environments.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest regarding the publication of this paper.

AUTHOR CONTRIBUTION

The authors confirm contribution to the paper as follows: study conception and design: Nurulfajar Abd Manap, Teow Chean Yang; data collection: Teow Chean Yang; analysis and interpretation of findings: Nurulfajar Abd Manap, Teow Chean Yang; provision of robotic and mechanical expertise: Azma Putra; draft manuscript preparation: Nurulfajar Abd Manap. All authors reviewed the findings and approved of the final manuscript.

REFERENCES

[1] K. Seaborn, N. P. Miyake, P. Pennefather, and M. Otake-Matsuura, "Voice in human–agent interaction: A survey," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–40, 2021.

[2] M. Norda, C. Engel, J. Rennies, J. E. Appell, S. C. Lange, and A. Hahn, "Evaluating the efficiency of voice control as human–machine interface in production," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 3, pp. 2476–2487, 2024.

[3] L. Beňo, E. Kučera, P. Drahoš, and R. Pribiš, "Transforming industrial automation: Voice recognition control via containerized PLC device," *Scientific Reports*, vol. 14, no. 1, pp. 29387, 2024.

[4] J. P. Ayala Taco, O. A. Ibarra Jácome, J. L. Ayala Pico, and B. A. López Castro, "Development of an industrial safety system based on voice assistant," *Applied Sciences*, vol. 13, no. 21, pp. 11624, 2023.

[5] T. Zheng, E. H. Grosse, S. Morana, and C. H. Glock, "A review of digital assistants in production and logistics: Applications, benefits, and challenges," *International Journal of Production Research*, vol. 62 no. 21, pp. 8022-8048, 2024.

[6] M. McTear, "Rule-Based Dialogue Systems; Architecture, Methods and Tools" in Conversational AI, Springer Cham, 2021, pp. 43-70.

[7] M. Zdravković and H. Panetto, "Artificial intelligence-enabled enterprise information systems," *Enterprise Information Systems*, vol. 16, no. 5, pp. 505–523, 2022.

[8] S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas, and B. Santhosh, "Artificial intelligence-based voice assistant," in *Proc. 4th World Conf. Smart Trends Syst., Security and Sustainability (WorldS4)*, pp. 593–596, 2020.

[9] V. P. S. C. Priyadarshana, et. al., "Voice controlled robot manipulator for industrial applications," in *Proc. IEEE 13th Annu. Inf. Technol., Electron. and Mobile Commun. Conf. (IEMCON)*, pp. 109–115, 2022.

[10] Y. Liu, W. Wang, and Y. Li, "Realization of contactless elevator control panel system based on voice interaction technology," in *Proc. 3rd Int. Conf. Control Syst., Math. Modeling, Automation and Energy Efficiency (SUMMA)*, pp. 312–317, 2021.

[11] M. Forlini, M. Babcinschi, G. Palmieri, and P. Neto, "D-RMGPT: Robot-assisted collaborative tasks driven by large multimodal models," arXiv preprint arXiv:2408, 2024.

[12] C. A. Aguilera, A. Castro, C. Aguilera, and B. Raducanu, "Voice-controlled robotics in early education: Implementing and validating child-directed interactions using a collaborative robot and artificial intelligence," *Applied Sciences*, vol. 14, no. 6, pp. 2408, 2024.

[13] P. Rajakumar, K. Suresh, M. Boobalan, M. Gokul, G. D. Kumar, and R. Archana, "IoT-based voice assistant using Raspberry Pi and natural language processing," in *Proc. 3rd Int. Conf. Power, Energy, Control and Transmission Syst. (ICPECTS)*, pp. 101–107, 2022.

[14] B. S. Babu, V. Priyadharshini, and P. Patel, "Review of voice controlled robotic arm–Raspberry Pi," *Eur. J. Electr. Eng. Comput. Sci.*, vol. 5, no. 2, pp. 1–4, 2021.

[15] A. Chaudhari, K. Rao, K. Rudrawar, P. Randhavan, and P. Raut, "Development of robotic arm prototype," in *Proc. 2nd Int. Conf. Sustainable Computing and Data Communication Syst. (ICSCDS)*, pp. 287–292, 2023.

[16] S. Wu, S. Huang, W. Chen, F. Xiao, and W. Zhang, "Design and implementation of intelligent car controlled by voice," in *Proc. Int. Conf. Comput. Netw., Electron. and Automation (ICCNEA)*, pp. 214–218, 2022.

[17] F. Yu, C. Zhou, X. Yang, Z. Guo, and C. Chen, "Design and experiment of tomato picking robot in solar greenhouse," *Trans. Chinese Soc. Agric. Machinery*, vol. 53, no. 1, pp. 115–124, 2022.

[18] P. Sikorski, K. Yu, L. Billadeau, F. Esposito, H. AliAkbarpour, and M. Babaiasl, "Improving robotic arms through natural language processing, computer vision, and edge computing," arXiv preprint, May 2024.

[19] Y. Koc, A. A. Tarcin, and D. Kose, "Evaluation of voice recognition platforms and methods for edge AI devices," in *Proc. 8th Int. Artif. Intell. and Data Processing Symp. (IDAP)*, pp. 155–160, 2024.

[20] R. Xu, Z. She, J. Chen, B. Y. Lu, R. Huang, and X. Li, "Preliminary study of the voice-controlled electric heat radiator," in *Proc. Int. Conf. Adv. Commun. Technol. (ICACT)*, pp. 1092–1096, 2020.