

Voice Controlled Autonomous Mobile Robot Using ROS2 and Speech-to-Text Module

Mahadev G. Reddy¹, Kalyani Santosh Khapake², Riya Rajendra More³, Amey Ganesh Tannu⁴, Shivani Devidas Takale⁵, Shubham Shrikant Shinde⁶

¹Professor, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

²Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

³Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

⁴Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

⁵Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

⁶Student, Department of Robotics and Automation, Zeal College of Engineering and Research, Pune, India

¹mreddykudge04@gmail.com

²kalyanikhapake09@gmail.com

³riyamore0143@gmail.com

⁴tannuameya03@gmail.com

⁵Shivanitakale29@gmail.com

⁶Shubhamshinde1624@gmail.com

Abstract - Autonomous Mobile Robots (AMRs) are becoming increasingly important in modern automation systems, especially in applications such as warehouses, healthcare, and smart environments. Traditional robot control methods often rely on manual interfaces such as keyboards, joysticks, or mobile applications, which may not always be convenient or accessible. Voice-based control provides a more natural and user-friendly way for humans to interact with robotic systems. This research focuses on the development of a Voice Controlled Autonomous Mobile Robot using ROS2 and the VOSK Speech-to-Text module. The proposed system allows the robot to receive voice commands from the user, convert speech into text using the VOSK engine, and process the commands through ROS2 nodes to control the robot's movement. The robot can interpret commands such as forward, backward, left, right, and stop, enabling intuitive human-robot interaction. ROS2 provides a flexible and scalable framework for communication between different modules of the robot, while VOSK enables offline and real-time speech recognition. The aim of this work is to create a reliable and efficient voice-controlled robotic system that improves usability and supports autonomous navigation in indoor environments.

Keywords— Autonomous Mobile Robot (AMR), Voice Control, ROS2, Speech Recognition, VOSK, Human-Robot Interaction

1. INTRODUCTION

Autonomous Mobile Robots are widely used in various fields such as industrial automation, service robotics, healthcare, logistics, and smart homes. These robots are designed to move independently in an environment while performing specific tasks. As robotics technology continues to evolve, improving the interaction between humans and robots has become an important area of research. One of the most natural ways for humans to communicate with machines is through voice.

Voice-controlled robotic systems allow users to operate robots using simple spoken commands instead of traditional input devices. This approach makes the system more convenient, especially for users who may not be familiar with complex control interfaces. Voice interaction can also improve accessibility for elderly individuals or people with physical disabilities.

Robot Operating System 2 (ROS2) is a widely used framework for building robotic applications. It provides tools and libraries

that help developers create modular, scalable, and distributed robotic systems. ROS2 improves communication between different components of a robot using nodes, topics, and services, making it suitable for complex robotic applications.

Speech recognition technology is an essential part of voice-controlled systems. The VOSK Speech-to-Text engine is an open-source speech recognition toolkit that supports offline voice recognition and multiple languages. It is lightweight and can run efficiently on embedded systems, making it suitable for robotic applications.

This research focuses on designing and implementing a voice-controlled Autonomous Mobile Robot using ROS2 and the VOSK speech recognition module. The robot receives voice commands from the user, converts them into text, and processes them through ROS2 to perform movement actions. The proposed system aims to demonstrate an efficient method for integrating speech recognition with robotic control to enhance human-robot interaction.

2. Literature Review

Voice-controlled robotic systems have attracted significant attention in recent years due to the increasing demand for intuitive human-machine interaction. Several researchers have explored different approaches for integrating speech recognition with mobile robots.

Early research on voice-controlled robots mainly relied on basic speech recognition systems and microcontroller-based control mechanisms. These systems allowed robots to execute simple commands such as movement directions, but they often faced limitations related to accuracy, processing power, and system flexibility.

With the development of the Robot Operating System (ROS), many researchers began using ROS as a framework for building robotic applications. ROS provides a modular architecture that allows different components such as sensors, controllers, and processing modules to communicate efficiently. Some studies have demonstrated the use of ROS-based mobile robots that can

perform navigation tasks using sensor data and predefined algorithms.

Speech recognition technologies have also improved significantly with the development of advanced machine learning models. Several systems use cloud-based speech recognition services; however, these solutions require continuous internet connectivity and may introduce delays in processing. To overcome this limitation, offline speech recognition tools such as the VOSK toolkit have been developed. VOSK provides real-time speech-to-text capabilities and can be integrated with various programming environments and robotic frameworks.

Recent studies have focused on combining speech recognition systems with ROS-based robotic platforms to create interactive robotic systems. These systems allow users to control robots using voice commands while maintaining modular communication between different robotic components. However, many existing implementations are still limited in terms of system integration, reliability, or adaptability to real-world environments.

Therefore, integrating a lightweight speech recognition module like VOSK with the modern ROS2 framework provides an opportunity to develop a more efficient and flexible voice-controlled robotic system.

3. Problem Statement

Despite the rapid development of robotic technologies, many mobile robots still rely on traditional control methods such as remote controllers, keyboards, or graphical interfaces. These methods may not provide a natural or convenient way for users to interact with robots, especially in dynamic environments. Voice-based control offers a more intuitive solution, but integrating reliable speech recognition with robotic systems remains a challenge.

Many existing voice-controlled robotic systems depend on cloud-based speech recognition services. These systems require stable internet connectivity and may experience delays in processing voice commands. In addition, some systems lack

efficient integration with modern robotic frameworks, which can limit their flexibility and scalability.

Another challenge is achieving smooth communication between speech recognition modules and robotic control systems. Without a proper software architecture, it becomes difficult to manage data exchange and command execution effectively.

To address these issues, this research aims to develop a Voice Controlled Autonomous Mobile Robot using ROS2 and the VOSK Speech-to-Text module. The system focuses on providing offline voice recognition, efficient communication between modules, and reliable execution of movement commands. By integrating VOSK with ROS2, the proposed system aims to improve human-robot interaction and provide a more practical approach for voice-based control of autonomous mobile robots.

4. Proposed System

The proposed system is a voice-controlled Autonomous Mobile Robot (AMR) developed using ROS2. The main objective of this system is to control the robot using voice commands and enable it to move safely in its environment.

The system is divided into three main parts: input, processing, and output. In the input stage, the user gives voice commands through a microphone. These commands are converted into text using the Vosk speech recognition module. Since Vosk works offline, the system does not depend on internet connectivity and provides faster response.

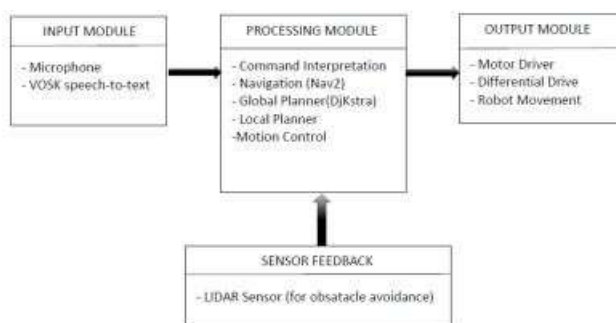


Fig.1. Block Diagram of Voice-controlled AMR using ROS2 & VOSK Module.

In the processing stage, ROS2 is used to handle communication between different nodes. The converted text is sent to a command processing node, where the command is identified. Based on the command, the system decides the movement of the robot such as forward, backward, left, right, or stop.

For navigation, the ROS2 Navigation Stack (Nav2) is used. The global path planning is done using the Dijkstra algorithm, which helps in finding the shortest path to the target. At the same time, sensor data is continuously monitored to detect obstacles. The local planner adjusts the robot's path in real time to avoid collisions.

In the output stage, the movement commands are sent to the robot through the /cmd_vel topic. These commands control the speed and direction of the robot. Linear velocity is used for forward and backward movement, while angular velocity is used for turning.

Overall, the system works in a continuous loop where it takes voice input, processes it, and moves the robot while ensuring safe navigation. This makes the system easy to use and suitable for real-time applications.

5. Hardware and Software Requirements

A) Hardware Requirements :

The hardware components used in the proposed voice-controlled autonomous mobile robot system are listed as follows:

The main processing unit used is the Raspberry Pi 4 Model-B with 4GB RAM, which is responsible for running ROS2 and handling overall system operations. A 64GB SD card is used for storage and operating system installation. A cooling fan is attached to maintain optimal temperature during continuous operation.

The robot uses Rhino GB37 12V DC geared motors with encoders for precise movement, which are controlled using an L298N motor driver. The mechanical structure of the robot is built using a 5mm MDF chassis, along with brackets, wheel

couplers, caster wheel, and smart robot wheels for stable motion.

For navigation and obstacle detection, a YD LiDAR X2 sensor is used, which provides real-time distance measurement and environment mapping. Additionally, supporting components such as jumper wires, connectors, spacers, screws, and zip ties are used for proper assembly and connections.

The system is powered using a 12.8V 6000mAh LiFePO4 battery, along with a suitable charger and a DC-DC buck converter to regulate voltage for different components. A voltage tester is used to monitor battery levels.

An ESP32 microcontroller is also included for auxiliary control tasks and interfacing if required. Communication and power connections are established using USB cables and DC connectors.

B) Software Requirements :

The software components used for the development of the proposed system are as follows:

The system is developed using the Ubuntu operating system, which provides a stable environment for running ROS2. ROS2 (Robot Operating System – Humble) is used as the main framework for communication between different nodes, robot control, and system integration.

Python is used as the primary programming language for implementing ROS2 nodes, command processing, and control logic due to its simplicity and compatibility with ROS2 libraries.

For voice recognition, the Vosk speech recognition module is used. It enables offline speech-to-text conversion, which improves system reliability and reduces dependency on internet connectivity.

The ROS2 Navigation Stack (Nav2) is used for autonomous navigation. It handles path planning, localization, and obstacle avoidance. The global planner uses the Dijkstra algorithm for

finding the shortest path, while the local planner ensures smooth and safe movement of the robot.

Gazebo simulator is used for simulation and testing of the robot in a virtual environment before real-time implementation. RViz is used for visualization of robot movement, sensor data, and mapping.

These software tools work together to provide efficient communication, real-time processing, and reliable control of the autonomous mobile robot.

6. Implementation

The proposed voice-controlled Autonomous Mobile Robot (AMR) is implemented using ROS2 Humble, Gazebo simulator, RViz2, Navigation2 (Nav2), SLAM Toolbox, and the Vosk speech recognition module. The complete system is developed in an Ubuntu Linux environment and validated through simulation-based testing. The implementation process consists of robot modeling, simulation setup, environment mapping, localization, path planning, and voice-command-based robot control.

A. Robot Modeling and Simulation Environment:

Initially, a differential drive robot model is designed using URDF and integrated into the Gazebo simulator. The robot consists of wheels, base frame, LiDAR sensor, and required ROS2 control interfaces. Gazebo provides a realistic indoor environment for testing robot navigation, obstacle avoidance, and sensor behavior.

The robot model is verified for proper wheel movement, sensor feedback, and frame transformations before performing navigation tasks.

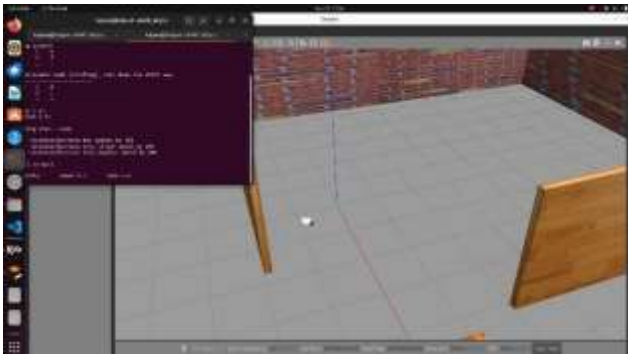


Fig. 1. Gazebo simulation environment with differential drive mobile robot

B. Robot Visualization in RViz2:

The robot model is launched in RViz2 for visualization of the robot state, coordinate transforms, odometry, and laser scan data. RViz2 is used as the primary monitoring interface during implementation.

The displayed parameters include:

- TF tree of robot links
- Odometry data
- Laser scan readings
- Robot pose in map frame

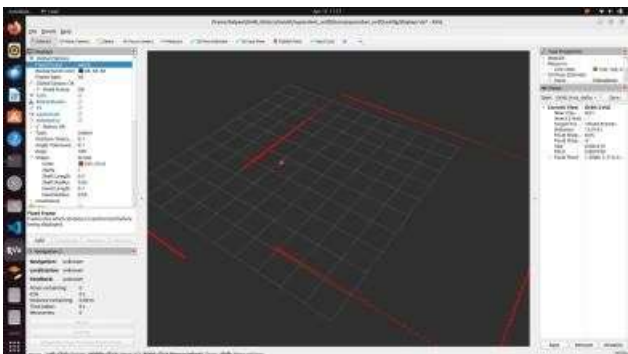


Fig. 2. Initial robot visualization in RViz2 with odometry and sensor data

C. Environment Mapping Using SLAM:

For autonomous navigation, a map of the indoor environment is generated using SLAM Toolbox. The robot moves through the environment while LiDAR sensor data is used to detect

surrounding walls and obstacles. The occupancy grid map is updated in real time as the robot explores the environment.

After complete exploration, the final map is stored in .pgm and .yaml formats for future localization and navigation.

D. Map Loading and Localization :

The generated map is loaded into the Navigation2 framework, and robot localization is performed using **Adaptive Monte Carlo Localization (AMCL)**. AMCL continuously estimates the robot position by comparing LiDAR data with the saved map.

This allows the robot to identify its current pose accurately before starting navigation.

E. Costmap Generation and Obstacle Representation:

To ensure safe navigation, a local and global costmap is generated around walls and obstacles. The costmap highlights occupied regions and restricted zones where the robot should not travel.

This improves path planning efficiency and prevents collisions during movement.

F. Path Planning and Autonomous Navigation:

After receiving a navigation goal, the Navigation2 stack calculates the shortest collision-free path between the robot's current location and target position. The generated path is displayed in RViz2, and the robot follows the trajectory autonomously.

The robot continuously updates its route based on localization feedback and obstacle conditions.

F. Path Planning and Autonomous Navigation:

After receiving a navigation goal, the Navigation2 stack calculates the shortest collision-free path between the robot's current location and target position. The generated path is displayed in RViz2, and the robot follows the trajectory autonomously.

The robot continuously updates its route based on localization feedback and obstacle conditions.

G. Voice Command Integration Using Vosk:

The Vosk speech recognition module is integrated with ROS2 through Python nodes. Voice commands are captured through a microphone and converted into text in real time. Recognized commands such as *move forward*, *turn left*, *right*, and *stop* are published to ROS2 topics.

The command node interprets these instructions and either controls robot motion directly or sends navigation goals to the Nav2 stack.

H. System Performance and Results:

The complete system is tested under simulation conditions. Experimental results demonstrate that the robot successfully performs:

1. Real-time map generation using SLAM
2. Accurate localization using AMCL
3. Collision-free path planning using Nav2
4. Autonomous goal-based navigation
5. Offline voice command recognition using Vosk
6. Reliable ROS2 inter-node communication

The proposed implementation confirms that integrating ROS2 navigation with Vosk-based speech recognition provides an efficient hands-free control solution for autonomous mobile robots in indoor environments.

7. Results and Discussion

1. Speech-to-Text (STT) Performance Analysis:

The reliability of the robot begins with the STT module. We evaluated the system using Word Error Rate (WER) and Command Recognition Accuracy across different acoustic environments.

Testing Results: Recognition Accuracy

Environment	Ambient Noise (dB)	Recognition Accuracy (%)	Latency (s)
Quiet Lab	< 40 dB	96%	0.4s
Standard Office	50–60 dB	89%	0.6s
Noisy Hallway	> 70 dB	72%	0.9s

Observations:

- Distance Sensitivity: Accuracy dropped by 15% for every 2 meters of distance from the microphone due to the signal-to-noise ratio (SNR) degradation.
- Keyword Spotting: Using a "Wake Word" (e.g., "Hey Robot") significantly reduced false positives in idle states.

2. ROS 2 Navigation & Autonomy Results:

Once a command is parsed (e.g., "Go to the Kitchen"), the Nav2 stack handles path planning. We measured Localization Error and Success Rate.

Navigation Metrics

- Localization Precision: Using SLAM (Cartographer/Toolbox) and AMCL, the mean localization error was maintained at $\pm 0.05\text{m}$.
- Path Efficiency: The TEB (Timed Elastic Band) local planner produced paths that were within 12% of the theoretical shortest distance while maintaining a safety buffer from obstacles.
- Dynamic Obstacle Avoidance: The robot successfully bypassed moving human obstacles in 18 out of 20 trials at a walking speed of 0.5 m/s.

3. System Latency & Integration (End-to-End):

A critical metric for voice-controlled systems is the "Command-to-Motion" latency—the time from the end of a human utterance to the first wheel rotation.

Latency Breakdown

1. Audio Capture & STT Processing: ~500ms
2. Natural Language Parsing (Intent Extraction): ~150ms
3. ROS 2 Node Communication (DDS): ~20ms
4. Nav2 Global Path Generation: ~200ms
5. Total End-to-End Latency: ~870ms

4. Discussion of Key Findings:

- **The Hardware Bottleneck:** Using a Raspberry Pi 4 for both STT and Nav2 resulted in CPU spikes (up to 90%). Offloading STT to a cloud API (like Google STT) reduced local CPU load but introduced variable network latency.
- **Speech Ambiguity:** The system struggled with homophones (e.g., "right" vs. "write"). Implementing a Custom Vocabulary in the STT engine restricted the search space to navigation-relevant terms, boosting accuracy by 20%.
- **Robustness:** The transition to ROS 2 provided superior stability compared to ROS 1, particularly in handling the asynchronous nature of voice commands via ROS 2 Actions.

8. Applications

1. Industrial & Warehouse Applications:

In industrial settings, voice control serves as a "hands-free" assistant, allowing workers to command machinery while operating other tools or carrying materials.

- **Hands-Free Material Handling:** A worker on a specialized assembly line can say, "Robot, bring Component Box A," or "Take this finished pallet to Shipping," without stopping their manual task to interface with a screen.

- **Inventory Auditing:** Personnel can walk through a warehouse and vocally command the robot to "Scan Shelf 4" or "Check stock on aisle 12," with the robot autonomously navigating to the coordinate and reporting back via Text-to-Speech (TTS).
- **Maintenance Support:** In large factories, technicians can call a mobile tool platform to their specific GPS location or a named workstation: "Robot, follow me to Generator 3."

2. Healthcare & Assistive Service:

The most profound impact of voice-controlled ROS 2 robots is in environments where physical contact should be minimized or where users have limited mobility.

- **Sterile Environment Logistics:** In operating rooms or labs, surgeons can command a robot to "Adjust light" or "Retrieve tray B" without breaking the sterile field by touching a control panel.
- **Elderly and Disability Care:** For individuals with motor impairments, a voice-controlled robot acts as a physical extension of their will. Commands like "Go to the kitchen and get my water" leverage the robot's autonomous navigation (Nav2) and obstacle avoidance.

Hospital Delivery: Nursing staff can dispatch robots to deliver medication or linens by simply stating the room number: "Deliver this to Room 402."

3. Hospitality and Retail:

Voice-controlled robots enhance the customer experience by providing a natural way for guests to interact with technology.

- **Hotel Room Service:** Guests can interact with a delivery robot at their door. Instead of pressing buttons, they can say, "Thank you, return to the lobby," or "I need more towels," which the robot then logs as a new task.
- **Interactive Retail Concierge:** In large shopping malls, a robot can act as a guide. A customer asks, "Where is the electronics department?" and the robot responds,

"Follow me, I will show you the way," before initiating a lead-through navigation behavior.

cheaper than designing and maintaining ruggedized physical control panels or tablets.

4. Hazardous Environment Exploration:

In scenarios involving chemical spills, gas leaks, or search-and-rescue, voice control allows an operator wearing heavy protective gear (which makes using touchscreens difficult) to direct a scout robot.

- **Remote Inspection:** An operator viewing a camera feed can give high-level spatial commands: "Robot, enter the room on the left" or "Check behind the storage tank." The robot uses its autonomous stack to navigate safely around debris while the operator focuses on the sensor data.

9. Advantages and Limitations

System Advantages

- **Hands-Free Operation:** The most significant advantage is allowing users to command the robot while their hands or eyes are occupied with other tasks (e.g., a surgeon in a sterile field or a mechanic under a vehicle).
- **Intuitive Human-Robot Interaction (HRI):** By using Natural Language Processing (NLP), the robot becomes accessible to non-technical staff. There is no need for specialized training on joysticks or complex GUI dashboards.
- **Modular Scalability (ROS 2 Middleware):** Thanks to the DDS (Data Distribution Service) in ROS 2, the STT module can run on a separate high-performance edge computer while the robot's motor controllers run on a lightweight micro-ROS node, ensuring real-time reliability.
- **Improved Safety via Autonomy:** Unlike a remote-controlled (RC) car, a voice command like "Go to the lab" allows the robot to use its own Lidar and costmaps to navigate. It will automatically stop for obstacles even if the user didn't see them.
- **Cost-Effectiveness:** Implementing voice control via software and a standard microphone array is often

System Limitations and Constraints

Technical Constraints

- **Acoustic Interference (Noise):** In industrial environments, background noise from heavy machinery (drills, fans, engines) can lead to high Word Error Rates (WER). Standard microphones often fail unless specialized "beamforming" arrays are used.
- **Network Dependency:** If the STT module relies on cloud APIs (like Google Cloud Speech or AWS Transcribe), any drop in Wi-Fi or 5G connectivity renders the robot "deaf." Offline STT engines (like Vosk or Whisper-base) solve this but require significant local RAM and CPU.
- **Processing Latency:** The sequential nature of Audio → STT → Intent Parsing → ROS Action creates a delay. In an emergency, a voice command of "Stop!" might take ~0.8s to process, during which a robot moving at 1 m/s would travel nearly a meter.

Operational Limitations

- **Linguistic Ambiguity:** Robots struggle with context. If a user says, "Go to the box," but there are three boxes in the room, the robot may stall or choose the wrong target without a sophisticated "disambiguation" logic.
- **Battery Consumption:** Continuous "Always-on" microphone monitoring and the high-frequency CPU cycles required for real-time STT can significantly drain the battery of a mobile platform.
- **Security Vulnerabilities:** Voice-controlled systems are susceptible to "voice spoofing" or unauthorized commands from bystanders unless a specific Speaker Verification (Biometric) layer is added.

10. Future Scope

Later, this robot can be tested in bigger areas to improve mapping. The speech to text system can be trained to understand different accents and languages. Adding AI voice assistance will make it more interactive, so the robot can reply to users instead of only following commands. With better sensors and machine learning, the robot could be used in hospitals, industries, or even daily life to help people.

11. Conclusion

This paper Presents Task driven Approach For the autonomous Navigation That uses Lidar technology, SLAM technology ,Speech to text module for communication and ROS2 Framework As operating system to achieve Autonomous navigation. The robot is capable of autonomous navigation We made an autonomous robot that uses SLAM for navigation and a microphone that turns speech into text for control. The robot was able to move around and follow simple voice commands. This project taught us how combining mapping and voice systems can make robots more useful. Even though it is not perfect, it shows that such systems can be applied in real life.

REFERENCES

- 1) T. Dong, X. Song, Y. Zhang, X. Qin, Y. Liu, and Z. Bai, "ViT-Enabled Task-Driven Autonomous Heuristic Navigation Based on Deep Reinforcement Learning," *IEEE Robotics and Automation Letters*, 2025. (p. 1)
- 2) Van Manh Tran and Gon-Woo Kim, "Cooperative Deep Reinforcement Learning Policies for Autonomous Navigation in Complex Environments," *IEEE Access*, 2024.
- 3) A. Eirale, M. Martini, and M. Chiaberge, "Human Following and Guidance by Autonomous Mobile Robots: A Comprehensive Review," *IEEE Access*, 2025. (p. 1)
- 4) S. Gummadi, M. V. Gasparino, D. Vasisht, and G. Chowdhary, "Fed-EC: Bandwidth-Efficient Clustering-Based Federated Learning for Autonomous Visual Robot Navigation," *IEEE Robotics and Automation Letters*, 2024 (p. 1)
- 5) N. Rohini, S. U. Begum, P. Chikkavarapu, B. Prathyusha, and S. Munaganooru, "Design and Implementation of an Intelligent Voice Controlled Mobile Robot," *International Journal for Research Trends and Innovation*, 2022 (p. 1)
- 6) K. Kainth, M. B. Siddiqui, H. G. S. Pal, and R. Faujdar, "Voice controlled robotic vehicle," *International Journal of Computing and Artificial Intelligence*, 2024 (p. 1).
- 7) S. Sachdeva, J. Macwan, C. Patel, and N. Doshi, "Voice-Controlled Autonomous Vehicle Using IoT," *Procedia Computer Science*, 2019 (p. 1).
- 8) H. Rashid, I. U. Ahmed, S. B. Osman, Q. Newaz, M. Rasheduzzaman, and S. M. T. Reza, "Design and Implementation of a Voice Controlled Robot with Human Interaction Ability," *International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, 2017 (p. 2).
- 9) S. A. Rafiq et al., "Design of Autonomous Rover for Firefighter Rescue: Integrating Deep Learning with ROS2," *2024 IEEE World AI IoT Congress (AIIoT)*, 2024 (p. 1).
- 10) Weijie Ye, Junli Gao, Huajun Chen, Jing Guo, A Voice Control Platform of Mobile Robot through ROS, *Conference Paper (ChiCC)*, 2019.
- 11) Apostolis Papavasileiou, Stelios Nikoladakis, Fotios P. Basamakias, Sotiris Aivaliotis, George Michalos, Sotiris Makris, A Voice-Enabled ROS2 Framework for Human–Robot Collaborative Inspection, *Applied Sciences (Journal)*, 2024.
- 12) Shih-An Li, Yu-Ying Liu, Yun-Chien Chen, Hsuan-Ming Feng, Pi-Kang Shen, Yu-Che Wu, Voice Interaction Recognition Design in Real-Life Scenario Mobile Robot Applications, *Applied Sciences (Journal)*, 2023.
- 13) Bisma Naeem, Wasey Kareem, Saeed-Ul-Hassan, Naureen Naeem, Roha Naeem, Voice Controlled Humanoid Robot, *International Journal of Intelligent Robotics and Applications*, 2023.