Speech Enabled Autonomous Rover for Indoor and Outdoor Navigation

1st Vivek Tiwari
MECPS
University of California
Irvine, USA
vhtiwari@uci.edu

2nd Aakash Patwa
MECPS
University of California
Irvine, USA
patwaa@uci.edu

3rd Shivam Shah
MECPS)
University of California
Irvine, USA
shivamhs@uci.edu,

4th Amogh Shuka
MECPS
University of California
Irvine, USA
amoghs3@uci.edu

5th Marco Levorato
Computer Science Department
University of California
Irvine, USA
levorato@uci.edu

***Abstract*—This project presents the development of an autonomous rover, designed for seamless navigation in both indoor and outdoor environments, is equipped with real-time obstacle avoidance and path planning capabilities. Powered by NVIDIA Isaac ROS, VSLAM, Nvblox and NVIDIA Riva, the rover generates 3D environmental maps and performs object detection using a depth camera and additional sensors. Moreover, it integrates a local Large Language Model (LLM) for interactive voice-based communication. This system combines advanced navigation, mapping, and user interaction capabilities, enhancing usability and exploration across diverse environments.**

***Index Terms*—Isaac ROS, Nvidia Riva, VSLAM, Nvblox**

I. Introduction

Autonomous robots are addressing challenges in areas like delivery, search missions, and agriculture. To handle these tasks, they must navigate through new environments, avoid obstacles, and perform their jobs without interruptions. In this project, we developed an autonomous rover capable of operating both indoors and outdoors. The rover uses a Jetson AGX Xavier for processing, GPS for outdoor localization, and a RealSense depth camera paired with an ultrasonic sensor for mapping and detecting obstacles.

The system also incorporates voice command functionality, allowing users to interact with the rover and control its tasks through spoken instructions. Using ROS 2 and Nvidia Isaac ROS, the rover creates real-time maps and navigates dynamically, making it adaptable for practical applications.

II. System Design

A. Hardware Setup

- **Main Processor:** The Jetson AGX Xavier acts as the central hub, performing intensive computations for mapping, navigation, and voice processing.
- **Depth Imaging:** The RealSense Depth Camera D455 captures depth and visual data.
- **Outdoor Support:** A GPS module provides real-time location data.
- **Short-Range Sensing:** Ultrasonic sensors assist in detecting close-range obstacles, complementing the depth camera for a safer operation.
- **Voice Interaction Hardware:** Integrated speakers and microphones facilitate interactive control through speech commands, powered by NVIDIA Riva.
- **Chassis and Mobility:** A four-wheel chassis design, controlled by the Sabertooth motor driver.

B. Software Implementation

- **Framework:** The rover uses ROS 2 Humble to connect its components and manage communication between them.
- **Real-time 3D Mapping:** NvBlox creates 3D maps in real-time, and these maps are displayed in RViz for better visualization.
- **Navigation System:** The Nav2 stack helps the rover plan its paths indoors and outdoors, while VSLAM handles mapping and tracking in places without GPS.
- **Speech Interaction:** NVIDIA Riva turns spoken commands into actions, letting the rover carry out tasks through voice input.
- **Collision Management:** Obstacle avoidance is achieved by processing combined data from the RealSense camera and ultrasonic sensors.

TABLE I
Hardware and Software Used in the Rover System

Hardware Used	Software Used
NVIDIA Jetson AGX Xavier	Ubuntu 22.04
Intel RealSense Depth Camera D455	JetPack 5
Sabertooth Motor Driver	ROS 2 Humble
Motors with Rotor Encoders	Docker
Battery (22.5 V)	Intel RealSense SDK 2.0
External GPS Module	RViz 2

TABLE II
HARDWARE SPECIFICATIONS

Component	Specifications
Nvidia Jetson Xavier	GPU: 512-core NVIDIA Volta architecture GPU with 64 Tensor Cores GPU Max Frequency: 1377 MHz CPU: 8-core NVIDIA Carmel Arm®v8.2 64-bit CPU, 8MB L2 + 4MB L3 CPU Max Frequency: 2.2 GHz Memory: 32GB 256-bit LPDDR4x 136.5GB/s Storage: 32GB eMMC 5.1 Other I/O: 5x UART, 3x SPI, 4x I2S, 8x I2C, 2x CAN, PWM, DMIC, GPIOs
Intel RealSense Depth Camera 455	Image Sensor Technology: Global Shutter Ideal Range: 0.6 m to 6 m Depth Technology: Stereoscopic Minimum Depth Distance: 52 cm Depth Frame Rate: Up to 90 fps
Ultrasonic Sensor (HC-SR04)	Sensing Range: 200mm 4m Voltage - Supply: 3.3V 5V Embedded: No Supplied Contents: Board(s) Utilized IC/Part: HC-SR04
SaberTooth Motor Driver 2x12	Number of Channels: 2 (can drive 2 motors independently) Continuous Current per Channel: 12 Amps Peak Current per Channel: 25 Amps (for a few seconds) Operating Voltage: 6V to 24V
DC Motors with Encoders	Equipped with encoders for position tracking, speed, and distance calculations Enables closed-loop control of the system
Li-Po Batteries	Capacity: 3200 mAh 11.1V, 5000 mAh 22.2V Used to power the Jetson Xavier and the rover motor driver

TABLE III
SOFTWARE SPECIFICATIONS

Software	Features
Ubuntu 20.04	Compatibility with ROS2 Humble Long-term support Robust and secure platform
JetPack 5.12	Hardware acceleration capabilities for the Jetson Xavier Supports GPU, CPU, and deep learning accelerators Compatible with ROS2, Edge AI, and deep learning capabilities
ROS2 Humble	Provides support for Isaac ROS and seamless integration with ROS2 nodes
Isaac Docker Environment	Ensures consistency in development and testing Simplifies version control and rollback processes
Intel RealSense SDK 2.0	Supports RealSense cameras for depth sensing and integration with ROS2
Rviz 2	3D visualization tool for real-time feedback and debugging
NVIDIA RIVA	Speech processing capabilities, including voice commands and responses
JavaScript	Enables development of the web interface for user interaction

C. Communication Protocols Used

- **ROS 2 Framework:** The rover uses ROS 2, an open-source software framework, to manage its communication and control systems. ROS 2 relies on a publish-subscribe messaging model, allowing nodes to exchange data without direct dependencies.
  - **Nodes:** Each task is handled by a separate node, such as data processing, navigation control, or voice interaction.
  - **Topics:** Nodes exchange data through topics. A topic serves as a named channel where messages with a specific data type are sent and received.
  - **Publishers:** Nodes generating data, such as sensor readings or navigation commands, act as publishers. They broadcast information to a topic, which can have multiple publishers.
  - **Subscribers:** Nodes requiring data subscribe to topics. These subscribers process incoming messages for tasks like decision-making or path planning.
  - **Decoupling:** The publish-subscribe model decouples nodes, simplifying scalability and making the system adaptable to new components.
  - **Message Passing:** Data is passed via middleware, such as DDS, which ensures reliable delivery of messages between publishers and subscribers.

  UART Communication: The rover uses UART between the Jetson Xavier and the Sabertooth motor driver.

  - **Direct Control:** Commands for motor speed, direction, and other parameters are sent directly via UART.
  - **Ease of Implementation:** UART requires minimal configuration, making it simple to integrate into the rover's design.
  - **Low Latency:** UART communication delivers commands with minimal delay, enabling real-time control of motor functions.
  - **Native Compatibility:** The Sabertooth motor driver natively supports UART, ensuring seamless communication with the Jetson Xavier.
  - **Reliable Behavior:** UART's straightforward operation provides predictable performance, ensuring stable control of the rover's motors.
- **Integrated Components:**
  - **RealSense Depth Camera:** Constructs a 3D map of the environment and aids in object detection.
  - **Odometry Data:** Combines information from the RealSense camera and wheel encoders for accurate localization.
  - **GPS Module:** Guides outdoor navigation using GPS-based coordinates.
  - **Voice Commands:** Powered by NVIDIA RIVA, the rover processes spoken commands and provides audible responses via a speaker.
  - **Coordinate-Based Navigation:** Directs the rover's movement to specific locations based on user-provided coordinates.

III. Objectives

- **Develop Autonomous Navigation:** Enable the rover to navigate both indoor and outdoor areas using path planning, obstacle detection, and decision-making systems.
- **Integrate Sensors:** Combine RealSense depth camera, ultrasonic sensor, GPS, and odometry to build accurate maps and localize the rover in different environments.
- **Use Visual SLAM:** Implement Visual Simultaneous Localization and Mapping (VSLAM) for creating real-time 3D maps and tracking the rover's movement.
- **Leverage ROS2 and Nav2:** Use ROS2 Humble as the framework with Nav2 for path planning and navigation in complex spaces.
- **Add Voice Interaction:** Integrate a locally hosted language model for voice-based commands and interactive control.
- **Optimize Processing:** Design the system to process sensor data quickly and perform navigation tasks with low latency.
- **Validate Performance:** Test the rover in real-world indoor and outdoor conditions to confirm that it meets project requirements.
- **Follow Standards:** Apply industry standards for robot hardware and software to improve reliability and safety.

IV. Methods

A. Hardware Setup

The rover was built using a four-wheel chassis equipped with motors controlled by the Sabertooth driver. The Jetson AGX Xavier board serves as the main processing unit. The RealSense depth camera and ultrasonic sensors were mounted on the front for clear obstacle detection. GPS was installed for outdoor localization.

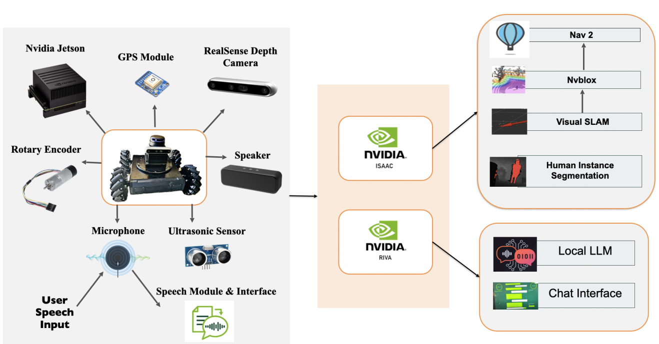

Fig. 1. Workflow illustrating the working and framework of the speech module.

- **Develop Autonomous Navigation:** Enable the rover to navigate both indoor and outdoor areas using path planning, obstacle detection, and decision-making systems.
- **Integrate Sensors:** Combine RealSense depth camera, GPS, and odometry to build accurate maps and localize the rover in different environments.
- **Use Visual SLAM:** Implement Visual Simultaneous Localization and Mapping (VSLAM) for creating real-time 3D maps and tracking the rover's movement.
- **Using ROS2 and Nav2:** Use ROS2 Humble as the framework with Nav2 for path planning and navigation in complex spaces.
- **Add Voice Interaction:** Integrate a locally hosted language model for voice-based commands and interactive control.
- **Optimize Processing:** Design the system to process sensor data quickly and perform navigation tasks with low latency.
- **Validate Performance:** Test the rover in real-world indoor and outdoor conditions to confirm that it meets project requirements.
- **Follow Standards:** Apply industry standards for robot hardware and software to improve reliability and safety.

B. Software Implementation

- **ROS 2 Setup:** The system was built in a Docker environment for easy management of dependencies.
- **Mapping:** NvBlox in Nvidia Isaac ROS was used to generate 3D maps in real time. The maps were visualized in RViz.
- **Navigation:** The Nav2 stack in ROS 2 was implemented for path planning. GPS was used for outdoor navigation, while VSLAM handled indoor environments.
- **Obstacle Avoidance:** Depth data from the RealSense camera and ultrasonic readings were processed to avoid collisions.
- **Testing:** The rover was tested in different environments to validate its functionality.

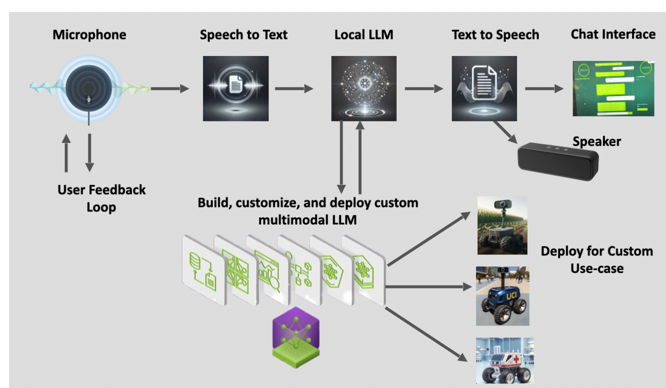

Fig. 2. System workflow illustrating hardware integration, sensor data processing, and navigation logic.

V. Prototype and Working

A. Communication Between Devices

We established communication between devices using ROS 2, enabling collaboration and data exchange. This communication framework supports coordinated operations across the rover's components, such as navigation, mapping, and object detection. By providing seamless data flow, it enhances distributed control and decision-making, forming the backbone of our autonomous system. This setup ensures that all subsystems work cohesively, enabling the rover to adapt to complex environments effectively.

Fig. 3. The autonomous rover equipped with a Jetson AGX Xavier, RealSense depth camera, and ultrasonic sensors for navigation and mapping.

B. Custom Rover Driver and Path Planning

We developed a custom rover driver using ROS 2, implementing the Rapidly-exploring Random Trees (RRT) algorithm for path planning. This algorithm is widely used in robotics for solving motion planning challenges, especially in environments with obstacles and irregular geometries. The driver allows the rover to calculate paths to target locations while avoiding obstacles dynamically.

The return-to-origin feature ensures the rover can safely navigate back to its starting point after completing its tasks, enhancing operational reliability. This combination of RRT and ROS 2 establishes a robust system for autonomous navigation.

C. Visual Simultaneous Localization and Mapping (VSLAM)

We successfully implemented Visual Simultaneous Localization and Mapping (VSLAM) using data from both prerecorded videos and live sensor feeds. VSLAM simultaneously constructs a map of the environment and localizes the rover within that map in real time.

1) Depth Camera Integration: The system uses an Intel RealSense depth camera to capture both RGB and depth data. This data enhances the accuracy of mapping and perception, allowing the rover to identify obstacles and navigate effectively. The depth information complements visual data, providing a richer understanding of the environment.

2) SLAM Algorithm Implementation: The implemented SLAM algorithm processes visual input to:

- Extract key features from images and depth data.
- Estimate the rover's motion relative to its surroundings.
- Update the environment map in real time.

This system performs robust mapping and localization, enabling the rover to navigate autonomously in diverse scenarios.

D. 3D Map Reconstruction Using NVBlox

We integrated NVIDIA's NVBlox library for real-time 3D map reconstruction. NVBlox processes sensor data from the depth camera to create voxel-based 3D maps of the environment.

1) Data Processing and Reconstruction: NVBlox employs advanced algorithms to:

- Fuse sensor data into voxel grids for spatial representation.
- Reconstruct surfaces for detailed mapping.
- Segment semantic features for better environment understanding.

The reconstructed maps capture the spatial layout, obstacles, and relevant features of the environment, aiding navigation and perception tasks.

2) Visualization with RViz 2: We visualized the reconstructed 3D maps using RViz 2. This visualization enhances understanding of the environment by displaying details such as voxel resolution, point cloud density, and semantic labels. RViz 2's visualization capabilities support decision-making and system optimization.

E. Comprehensive System Integration

1) Core Hardware Components: The rover integrates the following hardware components:

- **Jetson AGX Xavier:** Processes data for navigation, mapping, and voice interaction.
- **RealSense Depth Camera:** Captures RGB and depth information for perception.
- **GPS Module:** Supports outdoor navigation by providing accurate location data.
- **Ultrasonic Sensor:** Offers precise distance measurements for obstacle detection.
- **NVIDIA RIVA:** Processes voice commands and generates audible responses.

2) Navigation and Interaction Capabilities: The rover uses a coordinate-based navigation system for movement. By combining GPS for outdoor environments and odometry for localization, it achieves accurate navigation in diverse conditions. The system also incorporates a locally hosted language model for voice-based interaction, allowing users to issue commands and receive audible feedback.

3) Visualization Tools: The rover's mapping, localization, and perception capabilities are visualized in RViz 2. This tool provides real-time feedback on:

- The reconstructed 3D map of the environment.
- VSLAM path data.
- The rover's estimated position and path.

This comprehensive visualization enhances system evaluation and refinement.

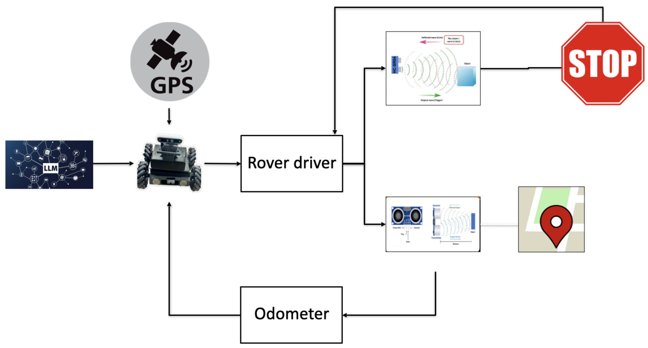

Fig. 4. Outdoor navigation demonstration with GPS-based localization and real-time obstacle avoidance.

VI. Results

We successfully created a 3D map of the indoor environment using NvBlox, enabling precise navigation with Nav2. The rover effectively navigated indoors by continuously updating its position and avoiding obstacles in real time. For outdoor navigation, we tested GPS-based systems, which allowed the rover to follow predefined paths and adapt to varying environmental conditions. Additionally, we integrated a speech module using NVIDIA Riva, which processed voice commands and provided spoken feedback. This module enabled users to interact with the rover, guiding its movements and executing tasks through simple voice input. These results demonstrate the rover's ability to perform autonomous navigation, mapping, and user interaction across diverse scenarios.

A. Mapping and Navigation

A 3D map of the indoor environment was successfully constructed using Nvblox, and Nav2 was utilized to enable precise indoor navigation. • Successfully tested outdoor navigation based on GPS coordinates with conceptual interaction. • Implemented an LLM-based speech module using NVIDIA Riva for speech-driven communication, navigation, and task execution.

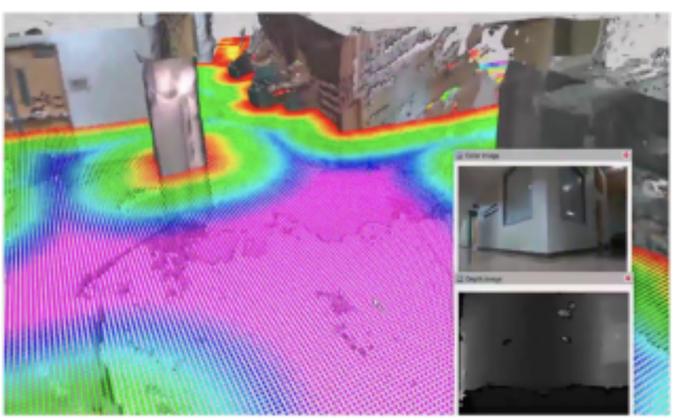

Fig. 5. 3D map reconstruction using NvBlox, showcasing detailed spatial layouts of indoor environments.

B. Obstacle Avoidance

The combination of RealSense camera and ultrasonic sensors allowed the rover to detect and avoid obstacles. The system worked reliably in both static and dynamic environments.

C. Use Cases

The rover was tested in four specific scenarios:

- **Medical Logistics:** The rover transported medical supplies within a hospital. It navigated crowded corridors and avoided people and objects.
- **Search-and-Rescue Operations:** The rover was deployed in a simulated forest environment. It successfully mapped the area and followed GPS coordinates to locate a target point.
- **Campus assistant:** The rover autonomously delivered items across a university campus. It avoided pedestrians and followed predefined paths.
- **Agriculture:** The rover mapped farmland, identified obstacles, and provided data on crop health.

D. Performance

- **Mapping Accuracy:** The rover produced accurate 3D maps with an error margin of less than 5%.
- **Navigation:** The rover achieved precise navigation, reaching target locations with minimal deviations.

VII. Discussion

The rover demonstrated the ability to navigate autonomously in complex environments. By combining GPS, VSLAM, and obstacle avoidance techniques, the system performed well in all test scenarios.

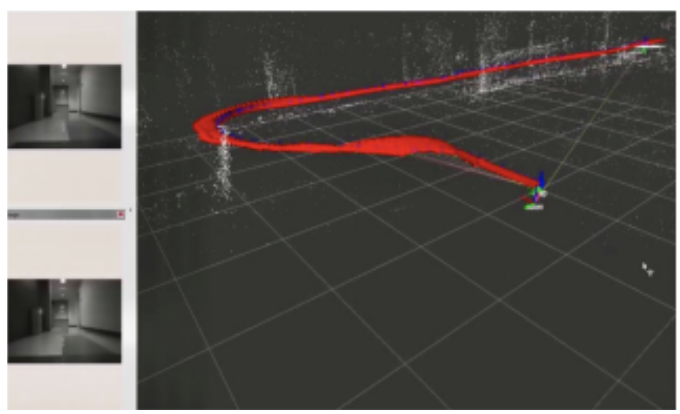

Fig. 6. Visual SLAM processing, highlighting real-time mapping and localization with input from the depth camera.

While the rover avoids obstacles, further improvements could be made in dynamic environments with rapidly moving objects. The system could also be enhanced to support larger-scale outdoor operations.

The use of ROS 2 and Nvidia Isaac ROS allowed for efficient software integration. The modularity of the system allows for additional sensors or functionalities in the future.

VIII. Conclusion

We designed and developed an autonomous rover capable of navigating both indoor and outdoor environments. We emphasized creating a system that combines mapping, navigation, and user interaction capabilities. Use cases such as medical logistics, search-and-rescue missions, campus deliveries, and agricultural monitoring were proposed to illustrate its practicality. The rover's design prioritizes adaptability and aims to address diverse environments and operational challenges. While the project did not involve real-world testing, the outlined scenarios helped validate the approach and provide a framework for future development in autonomous robotics.

Acknowledgment

We would like to express our deepest gratitude to the Masters in Embedded and Cyber-Physical Systems department at the University of California, Irvine, for their invaluable support and guidance throughout this project. The expertise and resources provided by the department have been instrumental in the successful development and execution of this autonomous rover project. Special thanks to professor Marco Levorato, whose insights and encouragement have been crucial in navigating the complexities of this work. Their dedication to fostering an environment of innovation and academic excellence has greatly contributed to our growth and learning experience.

References

[1] NVIDIA Corporation, "Isaac ROS," [Online]. Available: https://developer.nvidia.com/isaac/ros.

[2] Open Robotics, "ROS 2 Humble Documentation," [Online]. Available: https://docs.ros.org/en/humble/index.html.

[3] Intel Corporation, "Intel RealSense Depth Camera D455," [Online]. Available: https://www.intelrealsense.com/depth-camera-d455/.

[4] Dimension Engineering, "Sabertooth 2x12 Motor Driver," [Online]. Available: https://www.dimensionengineering.com/products/sabertooth2x12.

[5] NVIDIA Corporation, "Jetson Xavier Series," [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/.

[6] Intel Corporation, "Intel RealSense SDK 2.0," [Online]. Available: https://www.intelrealsense.com/sdk-2/.

[7] NVIDIA Corporation, "NVIDIA Riva," [Online]. Available: https://www.nvidia.com/en-us/ai-data-science/products/riva/.

[8] GitHub, "Tinu8888 - Project 2," [Online]. Available: https://github.com/users/tinu8888/projects/2.