Advances in IT and Electrical Engineering
Volume 31, 2025, Pages 35–41
https://doi.org/10.7862/re.2025.3
THE FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING
RZESZOW UNIVERSITY OF TECHNOLOGY

Original Research/Review

Overview of the software used in developing the Mars rover analogue autonomy system

Krystian Sołtys [1,*], Stefan Sudyk [1], Sebastian Zieliński [1], Dawid Żegleń [1]

[1] Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, 35-959 Rzeszow, Poland;

[*] Correspondence: krystian.soltys7@gmail.com

Received: 23 April 2025 / Accepted: 29 May 2025 / Published online: 6 June 2025

The project is financed by funds from the Polish state budget, granted by the Minister of Science within the framework of the "Student Scientific Circles Create Innovations" Program AGREEMENT No. SKN/SP/601717/2024.

Abstract
This paper presents an overview of the software components used to develop the autonomy system for a student-constructed Mars rover designed by the Legendary Rover Team from Rzeszow University of Technology. The system is built on ROS2, a modular and scalable robotics framework that enables seamless communication between various hardware components. Key technologies include a modular software framework ROS2, simulation tools for virtual testing such as Gazebo, and mapping and navigation systems (NAV2 and SLAM Toolbox) that enables the rover to understand and respond to its surroundings. By integrating various sensors and leveraging open-source solutions, the system supports autonomous movement, obstacle avoidance, and real-time decision-making. The use of simulation allows for safe and efficient development, while the modular approach facilitates easy testing and expansion. This work highlights the potential of modern robotics software to support building the autonomous systems.

Keywords: Legendary, Rover, Autonomy, ROS2

1 Introduction

Mars exploration has been one of the most important challenges in space exploration for many years. The Red Planet arouses great interest offering the promise of discovering traces of life beyond Earth [1]. To explore the mysteries of the Martian surface, scientists use high-tech rovers – the mobile robots that perform exploration missions in extremely harsh environmental conditions [2, 3].

Manual control of a Mars rover from Earth is very difficult due to significant communication delays that can last several minutes. This latency prevents operators from responding instantly to obstacles and hazards. It has become necessary to equip these vehicles with autonomous systems, allowing them to move independently and make decisions in real time. The autonomy of Mars rovers is based on advanced artificial intelligence algorithms [4] and vision systems [5, 6]. These algorithms [7] are of widespread interest to researchers, who are developing them to handle a variety of tasks [8] such as analyzing the surroundings [9], avoiding obstacles [10], and determining a route [11].

In order to develop and test technologies that can be used on real missions to Mars, the Rover Challenge competition was created. Students from around the world face tasks based on genuine exploration challenges using rovers they have constructed. The Legendary Rover Team from Rzeszow University of Technology is developing its own vehicle, along with an autonomy system for controlling it without human involvement. The project uses a set of sensors that together enable the robot to analyze its surroundings and determine its position and movement. These sensors work in combination to provide the rover with spatial awareness and environmental understanding. Among them there is a laser sensor measures distances to objects in space, allowing it to create maps and detect obstacles. A stereovision camera is used to map the surroundings in three dimensions, providing information about the depth and structure of the terrain. An inertial unit records acceleration and angular velocity to determine orientation and movement dynamics. A satellite positioning module provides global positioning data if a signal

This is an Open Access article distributed under the terms of the CC-BY-NC-ND 3.0 PL license, which permits others to distribute the work, provided that the article is not altered or used commercially. You are not required to obtain permission to distribute this article, provided that the original work is properly cited.

is available. Odometry, which relies on measuring wheel rotation, enables the estimation of the distance traveled. Information from these various sources is combined and used by the system to make navigation decisions and avoid obstacles in real time.

The objective of this work is not solely to organize existing solutions, but also to establish a foundation for their more effective selection and integration in future research and development initiatives. Through this approach, the study addresses the current demands of both the engineering and academic communities. Challenge in developing autonomous systems lies not in the scarcity of available tools, but rather in the lack of systematic knowledge about their practical applicability. While numerous software frameworks exist for robotics (e.g., ROS2, Gazebo, SLAM implementations), their effectiveness for specific use cases, particularly in resource-constrained, student-led projects, remains poorly documented. The primary contributions of this study are a systematic case-study and a practical framework for integrating ROS 2-based autonomy tool into resource-constrained educational rovers, as well as the provision of openly accessible guidelines intended to address the lack of practical implementation knowledge in the selection and application of these tools.

The remainder of this paper is organized as follows: Section 2 provides an overview of the key software components utilized in the development of the rover's autonomy system, including ROS2, Gazebo, SLAM, and NAV2. Each subsection details the role and integration of these tools within the project. Section 3 summarizes the findings and discusses the implications for future work in the field of autonomous robotics.

2 Software Overview

2.1 ROS2

One of the main tools for implementing autonomy is Robot Operating System 2 (ROS2). It is a set of open source libraries and tools for robot programming. The main benefits of using the ROS2 framework include:

- **Lightweight software** – works on low-power computers and embedded systems, where computing power is limited.
- **Modularity** – ROS2 is based on a modular architecture that is, components are separate modules. This approach reduces the complexity of projects and makes the code easier to navigate. It also simplifies adding new components to the robot (there is no need to interfere with other modules)
- **Scalability** - the framework in question is suitable for both small and large projects. Due to the modularity, adding new components is not a problem. Module can be added or removed without affecting the rest of the system.
- **Simulators compatibility** – The system integrates with simulators, such as Gazebo, which allows you to test your application in a virtual environment.

2.1.1 ROS2 architecture

The architecture of ROS2 is built on the Data Distribution Service (DDS) middleware, which enables decentralized, real-time communication between nodes using a publish-subscribe model. It uses nodes [12], which are separate programs that perform their tasks. For example: a node responsible for lidar, a node responsible for wayfinding, a node responsible for engine control, etc. Equally important are topics, which are actually names that identify the messages being sent. When a node wants to send a message it publishes it on a specific topic, while a subscribing node interested in the data is a subscriber to that topic (Fig. 1). In this way, the publisher and the subscriber are not aware of each other, and are therefore independent modules. The topic can thus be compared to a data bus. The big advantage of this approach is that each node can successfully publish (and listen to) messages on multiple topics, and each topic can transmit information from multiple nodes. Another method of communication between nodes is a so-called service [12]. This is a mechanism that sends a request to another node and waits for a response. Unlike topics, it works in a one-to-one system (one node sends a request and one responds to it).

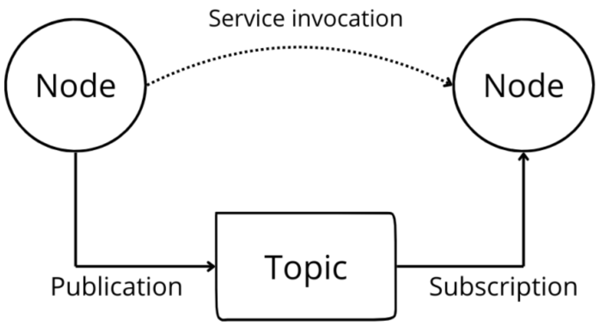

Fig. 1: Scheme of communication between nodes [12]

2.1.2 ROS2 usage in the project

The ROS2 framework, specifically the Humble distribution, is the basis of the rover's software structure. It enables communication between all components of the device. The kernel of the system is NVIDIA Jetson, which acts as the on-board computer. Using tools provided by ROS2, it communicates with peripheral devices such as the GPS module, cameras, lidar, odometry module, etc. Each of these nodes can operate independently of each other and transmit on separate topics. A major advantage of the framework in question is that it eliminates the need to develop custom communication protocols and data frames. This significantly simplifies and accelerates the integration of new modules, thereby enhancing the scalability of the system. Another advantage of the modularity of ROS2 is the ability to efficiently test the implemented solutions, for example, if there is a need to test only the lidar, ROS2 enables this to be done with minimal effort.

2.2 Gazebo

Gazebo is a an open-source robotics simulator designed to facilitate the development of computationally intensive solutions in the broad field/domain of simulation and robotics [13]. Each library implements a specific functionality, every package offered is self-contained as much as possible, they are compatible with many of the most popular operating systems and there are many possibilities for modification. All this adds up to a very helpful and versatile set of tools for performing simulations of automated and robotic systems. Due to the nature of the rover's target environment, and the problems associated with it, for example the need to avoid obstacles and associated with it, for example the need to avoid obstacles and traverse uneven terrain, there is a need to equip it with new systems.

The systems should allow it to deal autonomously with these problems when manual control is not possible. One of them is the ability to avoid and deal with obstacles it may encounter in yet unexplored terrain. Various distance or image recording devices such as LIDAR and cameras are used for this purpose.

Gazebo facilitates the simulation of LiDAR within an environment designed to approximate the rover's potential operating conditions, incorporating physical interactions through the integration of a physics engine. The components ros_gz_bridge and Gazebo Transport enable communication with Gazebo using topics from the system ROS2. Models of the rover and environment components are created using files in the Unified Robot Description Format URDF, an XML language specification in which the shape of the elements is described by their positioning relative to each other, capabilities and ranges of motion, mass, center of gravity, physical collision properties with other objects, and visual aspects.

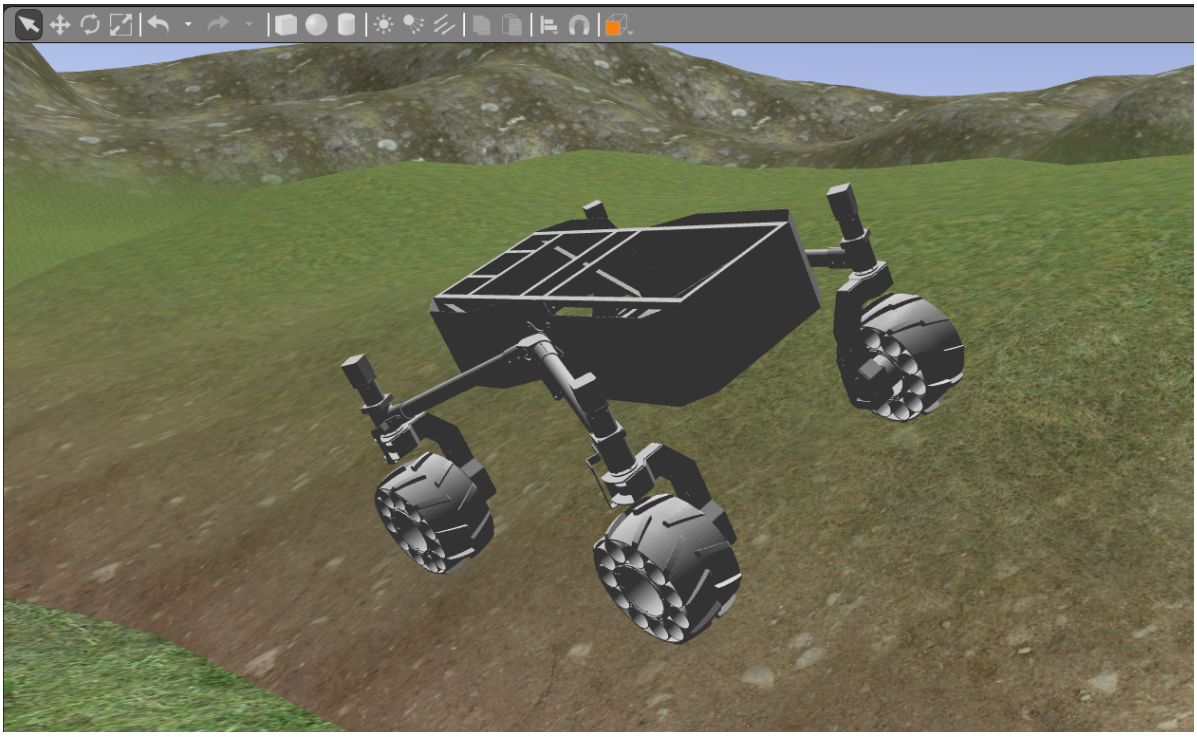

Fig. 2: Rover simulation in Gazebo

Testing models of the rover, the LIDAR device and the rest of its functionality in computer simulation using, the Gazebo program (Fig. 2), eliminates some obstacles preventing the development of computer and measurement systems. This is very useful when the condition of the physical rover does not allow such tests due to time-consuming work carried out on the electronic or structural systems to improve or repair them. In addition, at a small cost, it is possible to create conditions corresponding to those on, for example, the Moon, Mars or the desert where competitions are being held.

2.3 SLAM and NAV2

A required element for the operation of the rover's autonomous driving system is the determination of its position in space. In such a demanding task, it is not possible to rely exclusively on data from the GPS module, as access to it may be challenging or entirely impossible to access in a given area. Also during autonomous tasks at competitions, use of the GPS data may be prohibited.

To achieve this, Simultaneous Localization And Mapping (SLAM) technology was used for the operation of the autonomous driving system . SLAM is a process in which mobile robots can map a previously unknown terrain and determine their location based on the constructed map at the same time [14] without using GPS. The created map of the environment in which the robot works can be used to support other tasks, for example, planning a route or providing visualization that is understandable to the operator [15]. Currently, this is an important element applicable to unmanned aerial vehicles (UAVs), autonomous vehicles or augmented reality systems.

2.3.1 SLAM usage in the project

As a solution for integrating SLAM processes with ROS2 technology, Slam Toolbox [16] was chosen for the rover project, and is currently one of the more popular solutions in this field. The use of this toolbox provided us with the necessary elements to build a terrain map and locate the position as it moves, using sensory data published on the corresponding ROS2 theme from the 2D LIDAR scanner, IMU and odometry. The received information is processed and transferred to Slam Toolbox, which in turn creates a 2D map of the surroundings (Fig. 3) based on this information and estimates the robot's location.

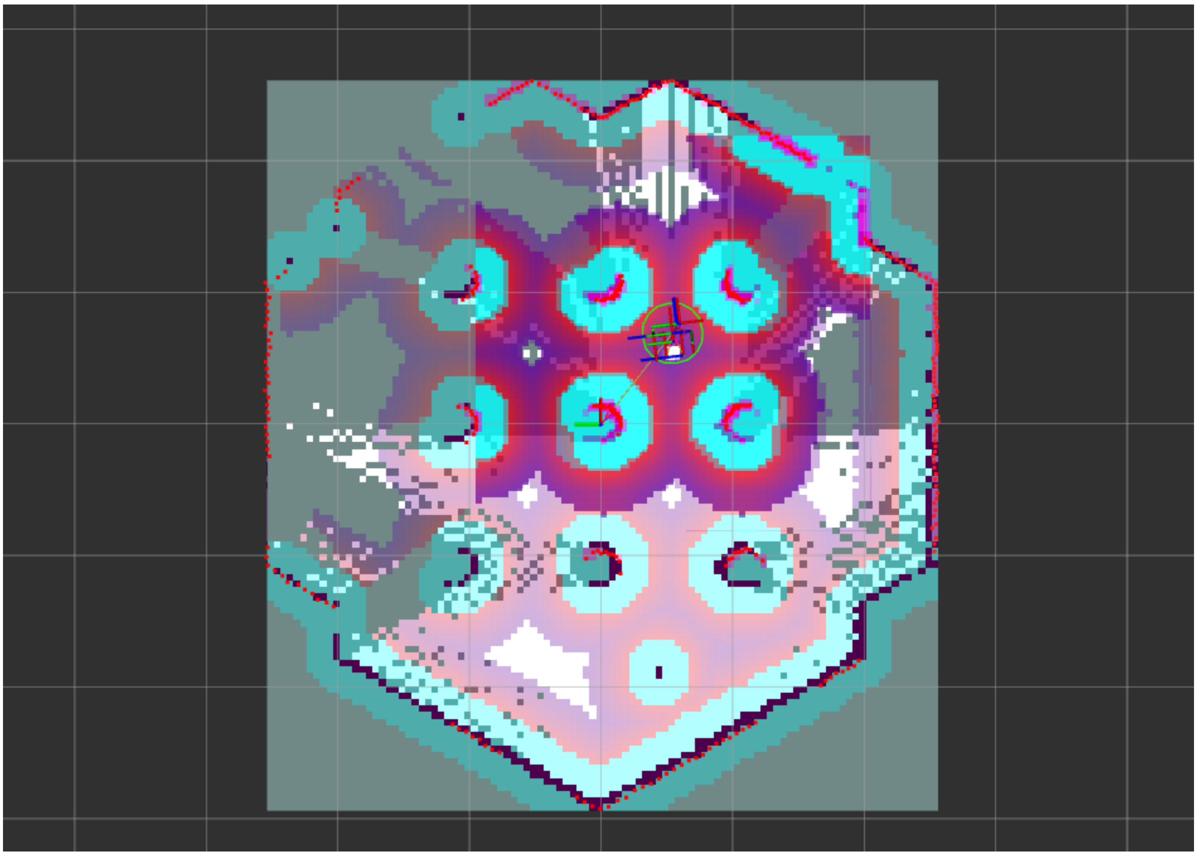

Fig. 3: SLAM toolbox visualization

However, in the course of its operation, the rover may encounter various terrain conditions in which the previously used solution providing only 2D mapping will not work. In this case, the most advantageous technique is to use Visual Simultaneous Localization and Mapping (VSLAM), which provides mapping of the environment and localization of the rover. Due to the presence of the mounted depth camera, it gives a possibility to build a 3D point cloud allowing even better analysis of the rover's operating environment, taking into account the slopes of the terrain.

2.3.2 NAV2

A key element for realizing rover control with minimal operator intervention is autonomous navigation. To provide it, the NAVIGATION2 (NAV2) framework and system [17] was selected, which is fully compatible with ROS2. This tool provides the necessary modules to support autonomous driving. The use of behavioral trees enables the creation customized and intelligent navigation behavior by orchestrating multiple independent components [18]. The architecture of this solution uses a Global Planner responsible for calculating the shortest path to a destination, as well as controllers that use local information to calculate the shortest local path.

In the rover project, NAV2 is used to enable autonomous navigation in both known and unknown environments. The framework integrates seamlessly with the SLAM Toolbox to utilize the generated maps for path planning. The rover's sensors, including LIDAR, cameras, and IMU, provide real-time data to NAV2, which processes this information to make navigation decisions.

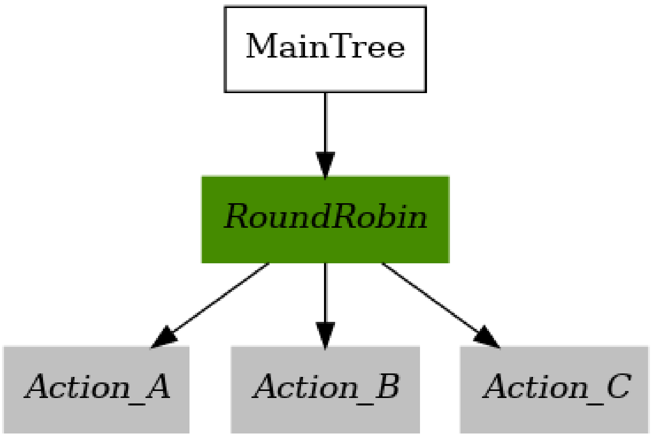

Fig. 4: NAV2 tree structure [18]

The behavior trees (Fig. 4) in NAV2 are customized to handle specific scenarios, such as navigating through narrow passages or avoiding moving obstacles. The global planner calculates the optimal route to the target location, while the local planner ensures that the rover can adapt to changes in the environment. Recovery behaviors are implemented to handle situations where the rover encounters unexpected obstacles or loses localization.

3 Summary

Modern mobile robotics, owing to the adoption of open source libraries, development platforms and simulation environments, enables the creation of advanced systems without the need to build everything from scratch. The article discusses an example of a complex project that integrated many existing tools to realize an autonomous mobile robot – in this case, a Mars rover. It demonstrating practical approaches to rapid development and testing in virtual and real-world environments. The project in question uses ROS2, a modern framework for robotics that provides a universal structure for communication between different components of the system. Its key feature is modularity: each component (e.g. sensor, motor, camera) acts as an independent "node" and communicates with others via so-called topics or services. This makes it easy to expand and test the system, which significantly speeds up software development and its adaptation to different platforms.

A variety of available solutions were utilized, enabling comprehensive testing within a virtual environment. Simulation allowed the reproduction of near-real conditions without the need for physical equipment. This made it possible to test the operation of individual components, such as sensors, motors and decision-making algorithms, in a controlled and safe environment. A key aspect of autonomy is the robot's ability to simultaneously build a map of its surroundings. It must also determine its position in space without relying on external location sources. This process is based on analyzing data from various sensors and enables orientation even in previously unknown terrain. In the next stage, using the built map and data from the environment, the robot independently plans its route to the destination and makes decisions related to avoiding obstacles or responding to changing conditions. The entire system has been designed to allow it to develop, test and gradually increase its level of autonomy without having to use a physical platform each time, thereby significantly reducing the time required for development and testing.

References

[1] Yiğit, E.: *Exploring Mars's harsh atmosphere* Physics Today, 2024, 77(7), 42-50.

[2] Muirhead B.K.: *Mars rovers, past and future* IEEE Aerospace Conference Proceedings, 2004, pp. 128–134.

[3] Patil A.: *A Comparative Study of Mars and Earth for Human Exploration and Habitation: Environmental, Technological, and Architectural Considerations* Technological, and Architectural Considerations, 2024.

[4] Aishath Murshida A., Chaithra B. K., Nishmitha B., Pallavi P. B., Raghavendra S., Mahesh Prasanna K. *Survey on artificial intelligence.* Int J Comput Sci Eng, 2019, 7, 1778-1790.

[5] Singh A., Kalaichelvi V., Karthikeyan R. *A survey on vision guided robotic systems with intelligent control strategies for autonomous tasks.* Cogent Engineering, 2022, 9(1), 2050020.

[6] Kalandyk D., Kapuściński T.: *Temporal signed gestures segmentation in an image sequence using deep reinforcement learning.* Engineering Applications of Artificial Intelligence, 2024, 131, 107879.

[7] Tong W., Hussain A., Bo W. X., Maharjan S.. *Artificial intelligence for vehicle-to-everything: A survey.* IEEE Access, 2019, 7, 10823-10843.

[8] Kalandyk D.: *Reinforcement learning in car control: A brief survey*. Selected Issues of Electrical Engineering and Electronics (WZEE), 2021, pp. 1–8.

[9] Kuutti S., Bowden R., Jin Y., Barber P., Fallah S. *A survey of deep learning applications to autonomous vehicle control.* IEEE Transactions on Intelligent Transportation Systems, 2020, 22(2), 712-733.

[10] Olson C.F., Abi-Rached H., Ming Ye, Hendrich J.P.: *Wide-Baseline Stereo Vision for Mars Rovers*. Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems , 2003, pp. 1302–1307.

[11] Meiring, G. A. M., Myburgh, H. C.: *A review of intelligent driving style analysis systems and related artificial intelligence algorithms.* Sensors, 2015, 15(12), 30653-30682.

[12] ROS Documentation, *ROS Concepts*. https://wiki.ros.org/ROS/Concepts, access on 2025-04-15.

[13] Gazebo Documentation, *Gazebo ROS2 integration*. https://gazebosim.org/docs, access on 2025-04-15.

[14] Bailey T.,Durrant-Whyte H.: *Simultaneous localization and mapping (SLAM): part II*. IEEE Robotics & Automation Magazine, 2006, Vol. 13, no. 3, pp. 108–117.

[15] Cadena C. et al: *Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age*. IEEE Transactions on Robotics, 2016, vol. 32, no. 6, pp. 1309–1332

[16] Macenski et al: *SLAM Toolbox: SLAM for the dynamic world.* Journal of Open Source Software, 2021, vol. 6, no. 61, pp. 2783

[17] Macenski S., Martín F., White R., Clavero J. G.: *The marathon 2: A navigation system*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 2718–2725

[18] Nav2 Documentation, *Nav2 Overview*. https://docs.nav2.org/index.html, access on 2025-04-15.