REALMS 2 - RESILIENT EXPLORATION AND LUNAR MAPPING SYSTEM 2

Dave van der Meer[1], Loïck P. Chovet[1], Gabriel M. Garcia[1], Abhishek Bera, and Miguel A. Olivares-Mendez

Space Robotics (SpaceR) Research Group, Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg {`dave.vandermeer`, `loick.chovet`, `gabriel.garcia`, `abhishek.bera`, `miguel.olivaresmendez`}`@uni.lu`

ABSTRACT

The European Space Agency (ESA) and the European Space Resources Innovation Centre (ESRIC) created the Space Resources Challenge to invite researchers to propose innovative solutions for robotic space prospection with focus on autonomous Multi-Robot System (MRS). This paper proposes Resilient Exploration And Lunar Mapping System 2 (REALMS2), a MRS framework for planetary prospection and mapping. It is based on Robot Operating System version 2 (ROS 2) and uses Visual Simultaneous Localisation And Mapping (vSLAM) for map generation. The REALMS2 uses a mesh network for a robust ad-hoc network. A single graphical user interface (GUI)) controls all the rovers, providing a simple overview of the robotic mission. REALMS2 was used during the second field test of the ESA-ESRIC Challenge and allowed to map around 60% of the area, using three homogeneous rovers while handling communication delays and blackouts.

Key words: lunar surface, visual simultaneous localisation and mapping, vSLAM, rtabmap, illumination.

1. INTRODUCTION

Recently, the Moon attracts the attention of space agencies and private companies for its potential for In-Situ Resources Utilisation (ISRU). For this reason, the European Space Agency (ESA) and the European Space Resources Innovation Centre (ESRIC) strive to bring forward the potential of autonomous robotic systems for space resource exploration. ESA and ESRIC created the Space Resources Challenge [3], where only 13 teams of researchers were accepted to do a first field test to show the potential of their autonomous system, with a focus on Multi-Robot System (MRS). The five best teams were selected to participate in a second field test [4] to find different resources inside a large lunar analogue environment, represented in Figure1.

Figure 1. Second field test arena for the ESA-ESRIC Space Resources Challenge

In the first field test of the Challenge [4], the authors present the Resilient Exploration And Lunar Mapping System (REALMS) [14], a MRS with two rovers performing Visual Simultaneous Localisation And Mapping (vSLAM). REALMS is composed of the Robot Operating System (ROS) for communication between nodes. The field test highlighted several limitations in REALMS [14]. The primary challenges to overcome are that the software framework was designed for centralised systems, the communication stability on the network layer could be improved, there is a lack of scalability of the user experience and the high network load limits the downlink capacity.

This paper proposes Resilient Exploration And Lunar Mapping System 2 (REALMS2), an improved version of REALMS that addresses its weaknesses. The upgraded architecture has extended coverage and is more resilient due to the multi-node interconnection made possible using the Robot Operating System version 2 (ROS 2) framework.

The contributions of this work are the use of a decentralised MRS based on ROS 2 and a mesh network in a real-world application for robotic space activities. The section 2 explores the state-of-the-art of planetary robotics. The section 3 provides an overview of the system setup. The section 4 foregrounds the experiments to validate the system. The section 5 shows how the system is used during the ESA-ESRIC Space Resources Chal-

---

[1] Authors share equal contribution.

lenge. The section 6 shows the results of the experiments and the participation in the challenge. The section 7 and section 8 highlight the lessons learned from this project.

2. RELATED WORK

The key elements for autonomous robotic missions on the lunar surface are MRS, robust ad-hoc network infrastructure and robotic agents that achieve a certain level of autonomy, even when disconnected from the ground station.

Multi-robot systems Within MRS, [13] considered two main distinctions: Centralised and decentralised systems. According to the authors, a centralised system depended highly on a main computing unit, limiting the autonomy of individual agents. In [12], a decentralised system distributes the intelligence and the communication between the rovers, increasing the system's robustness. However, the decentralised system led to a higher computational and network load.

Mesh Network Mesh networks were introduced by the norm IEEE 802.11s [6] as a new topological approach to wireless networks. A mesh network is composed of a set of nodes connected together, where one node can be connected to any number of nodes. They allow to distribute the network load and avoid bottlenecks by reducing the number of central points, like the typical access point of a regular WLAN network.

Hybrid Wireless Mesh Protocol (HWMP) introduced in 2007 as seen in [8] as the hybrid wireless mesh protocol. It defines the airlink metrics that allow it to handle less stable networks. The improved HWMP+ was proposed by [15], focusing on the estimation of the link quality, the author proves a more reliable network providing less network overhead than the original HWMP.

ROS 2 As highlighted in [16], ROS 2 was designed considering MRS constraints, such as real-time systems, small embedded platforms, and non-ideal networks. ROS 2 was developed as open source framework for robotic systems [10]. Its communication system was developed on top of a Data Distribution Service (DDS) [11], a middleware for establishing the data exchange between software nodes. ROS 2 offered various DDS implementation, however the standard was FastDDS. It focused on high latency with relatively small packets [2]. DDS allowed to set up multiple robots in the same network in a decentralised manner. This meant that each robot was entirely independent from the rest of the system. ROS relied on a centralised architecture, which was limiting the MRS capabilities. One of the other advantages of ROS 2 over ROS was the real-time management.

3. METHODS AND MATERIALS

REALMS2 uses three rovers. They are modified to provide more computational power, more sensing capabilities, and extended battery life. The software is upgraded from ROS to ROS 2 to allow controlling multiple robots as a decentralised system. Also, a small lunar lander provides extended computing capabilities and serves as a relay for the ad-hoc network to the operators.

3.1. Rover setup

The REALMS2 rovers use multiple hardware and software components.

Hardware The base of the REALMS2 robots is a Leo Rover [7], a commercial robot with ROS 2 support, represented in figure 2. Its main computation boards are a Raspberry Pi 4 and an added NVIDIA Jetson Xavier NX to increase the processing capabilities. The mapping capabilities have improved by adding an RPLidar A2 M8 Light Detecting And Ranging (LiDAR) sensor to two robots. The system is extended by a third robot to increase redundancy and mapping coverage. An 18 V Li-Ion battery with 4.0 Ah capacity is added to each rover for extended battery life.

The communication between the agents is based on a mesh network [6] using Mikrotik Goove 52ac antennas. This allows REALMS2 to distribute the load on the network and allocate the bandwidth more efficiently. Also, the mesh network allows to use the individual rovers as communication relays, which enables the rovers to extend the range of the communication network.

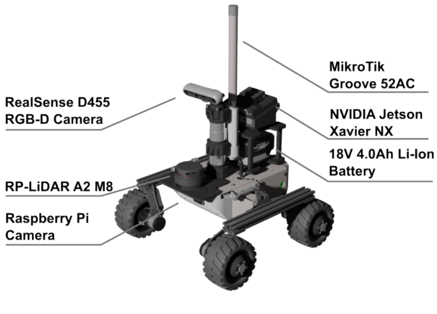

Figure 2. Leo Rover setup used for REALMS2

Software The NVIDIA Jetson Xavier receives all sensor data and runs the vSLAM software Real-Time Appearance Based Mapping (RTAB-Map) [9].

The REALMS2 rovers use Docker to run the individual ROS 2 packages. Using Docker increases modularity and simplifies the deployment of a new robotic agent, increasing the scalability of the system. Docker simplifies starting ROS 2 nodes at the boot sequence of the robots. In REALMS, the ROS nodes were launched manually through a remote connection.

The different maps are displayed to the operator through a monitoring graphical user interface (GUI). A ROS 2 map merging package [1] has been added to simplify the interface. It compares features of the individual maps and estimates the relative transformation to overlay the maps. The map merging is configured to use at least 20% overlap between the maps.

3.2. Lunar lander

The lunar lander serves as a stationary edge-computing infrastructure with an Intel NUC, with a Core-i7 CPU and 32 GB RAM. The lander acts as a network gateway to the ground station. It uses the same communication antenna as the rovers to connect to them and exchange data to be processed. Also, the lander contains a camera and a LiDAR to gain quick insights regarding the field test environment of the challenge.

3.3. User Interface

The user interface consists of a frontend, used by the operator, and a backend, that handles the communication to the different rovers.

Backend REALMS requires at least one operator per robot. The number of operators in the control room during the ESA-ESRIC Challenge is limited to five people. To offer a more scalable system, REALMS2 implements a backend to handle multiple robots through a single interface.

The backend runs on the ground station computer and provides six features: Rover selection, teleoperation, light on/off switching, odometry reset, rover reboot and network monitoring.

The rover selection is crucial for the operator to send commands to a specific rover. The backend takes as input the namespace of the rover and the commands, and distributes them to the corresponding rover. The backend sends no commands to the rovers if no namespace is provided.

The primary commands in ROS 2 are teleoperation instructions for the rovers, signals to turn on or off the LED ring on the rover, to reset the odometry in case the odometry is not trustworthy, and to reboot the rover.

The last feature is the network monitoring. By default, Linux logs incoming and outgoing data traffic. The network monitoring is reading these log files. The backend receives this data from the rovers and sends the bandwidth to the frontend. This method of measuring the bandwidth does not create any overhead to the network.

Frontend The backend only handles the information flow towards the frontend. As frontend, we use Foxglove [5] with a custom panel. In the REALMS framework, RVIZ is employed as the default visualization tool due to its widespread adoption within the ROS community. However, our analysis suggests that Foxglove Studio offers a more intuitive interface for integrating custom panels, although accomplishing this task in RVIZ is not straightforward. The GUI uses panels and tabs to group related information together for a better overview.

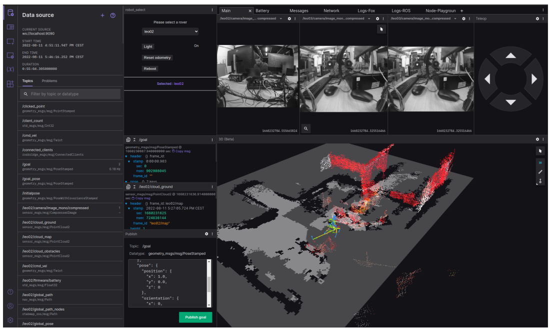

Figure 3. Foxglove Studio - Main interface

As seen in figure 3, the operator has a global overview of the rover map, camera and teleoperation panel. The custom panel on the left side is robot-agnostic and sends the namespaces to the backend. The namespace is automatically detected by scanning all the available topics and extracting the namespaces of the rovers in the system. Also, the frontend can discover new robots in the system during runtime.

The combined solution of the backend and frontend enhances the system's scalability and usability for the operator.

4. EXPERIMENTS

To validate the architecture, we conduct real-world experiments and focus on two specific aspects: range and map-merging capabilities.

4.1. Range

To verify the range of the antennas, the rovers are brought outside to measure the maximum distance at which they still have a stable connection between each other. The experiment stopped when the ping signal cannot reach the other robot, revealing the maximum distance.

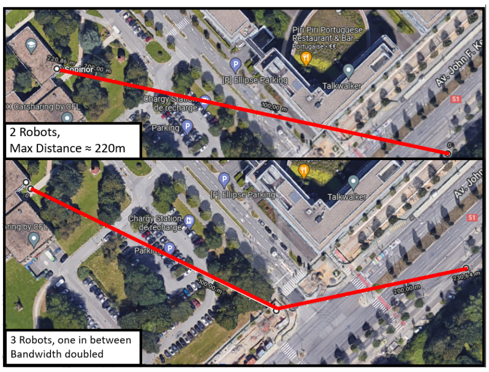

Figure 4. Range experiment

Figure 4 shows a representation on Google Maps with the position of the robots relative to each other. The experiment shows that the robots can ping each other within a radius of approximately 220 meters without any direct obstruction, such as buildings, rocks or dense forest.

To evaluate the effect of a relay rover in the network, we place a third rover in the middle of the other two. With the relay rover, we send ROS 2 camera image messages from the most distant rover through the relay to the operator. The experiment shows that the connection without the relay rover is insufficient to send an image to the operator using the ROS 2 messaging system.

4.2. Map Merging

Two of the REALMS2 rovers are placed in an outdoor environment to test the map-merging capabilities. The rovers are starting at the same position and map different areas. The starting position is a common area with sufficient overlap to merge the map. Figure 5 shows the two maps and how their features are matched. The rovers are controlled through the frontend by a single operator, using the camera streams and generated maps provided by the robots.

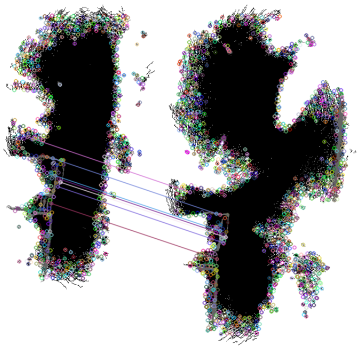

Figure 5. Map merging - matches between two maps covering the same area

5. ESA-ESRIC CHALLENGE

The final round of the ESA-ESRIC Challenge represents the field test of REALMS2. As our solution is suited for mapping, our partner Space Applications Services (SAS) analyses the rocks using spectroscopy. Our rovers are used to scout for potential resources or interesting areas. At the beginning of the mission, the rovers encounter failures in the autonomous navigation stack. Therefore, two operators continue the mission by teleoperating two of the three robots in parallel. As part of the challenge, the organisers requested to shut down one rover as a simulated system failure to verify the resilience to unexpected events. At this point, the operators use the third REALMS2 rover to replace the rover that simulates a failure. The lander acts as a communication gateway between the ground station and the rovers. The rovers send data to the lander during communication blackouts to use the bandwidth while no teleoperation is possible.

6. RESULTS

During the ESA-ESRIC Space Resource Challenge, the system was capable of mapping around 60% of the total surface of $1800m^2$, using three robots, enduring communication delays, blackouts and a planned partial system failure to simulate the conditions of a real lunar mission. Figure 6 shows the mapped area by the contribution of the three rovers represented in blue, green and red. There is a minor scale issue in the blue part of the map in the bottom right corner. This could be some scale estimation drift from the slightly higher movement speed of Leo-01 while traversing the terrain. The map clearly shows the position of two small craters on the left, a large crater on the bottom right and several large rocks. The map merging algorithm failed to calculate the relative transformation due to the sparsity of the map features. The autonomous navigation stack failed to plan paths to the rover's target locations and teleoperation was used to move the rovers.

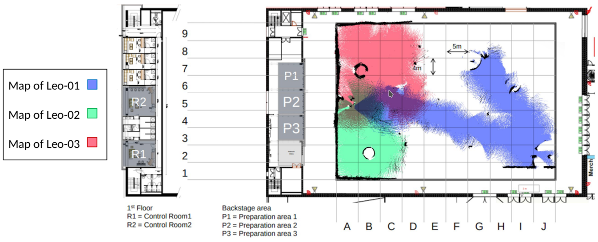

Figure 6. Area Mapped by REALMS2 during the ESA-ESRIC Space Resources Challenge

7. DISCUSSION

Some limits were encountered during the various experiments and the challenge. The map merging must be more robust in sparse environments and provide the transform matrices between the maps. Second, the rovers must automatically detect odometry loss and activate a custom recovery behaviour. Third, the autonomous navigation stack needs to be more robust.

8. CONCLUSION

This project showed that fully distributed MRS systems, such as those implemented with ROS 2, improve the resilience of planetary missions, especially when combined with a distributed communication system such as a Mesh network. The use of Docker allows to increase the modularity and thus improves the scalability of the system.

ACKNOWLEDGMENTS

The authors thank Space Application Services and their team in Belgium with special thanks to Jeremi Gancet, Fabio Polisano and Matteo de Benedetti for making the participation in the ESA-ESRIC Space Resources Challenge possible. The authors also want to thank ESA and ESRIC with special thanks to Massimo Sabatini, Franziska Zaunig, Thomas Krueger and Bob Lamboray for organising this event and hosting all team members in Esch-sur-Alzette, Luxembourg during the challenge. The authors thank Prof. Kazuya Yoshida from the Space Robotics Lab at the Tohoku University. This research was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant references 14783405, 17025341 and 17679211. For the purpose of open access, and in fulfilment of the obligations arising from the grant agreement, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission. This project was partly funded by the ESA-ESRIC Space Resources Challenge, Contract No. 4000137334/22/NL/AT.

REFERENCES

[1] T. Clephas, J. Galvis, and C. A. Alvarez. m-explore ros2 port. `https://github.com/robo-friends/m-explore-ros2`, 2021. Accessed on 2023-09-13.

[2] eProsima. eProsima Fast DDS Performance. `https://www.eprosima.com/index.php/resources-all/performance/eprosima-fast-dds-performance`, 2023. Accessed on 2023-07-03.

[3] ESA - European Space Agency. 1st field test - esa-esric challenge. `https://www.spaceresourceschallenge.esa.int/general-3`, 2023. Accessed on 2023-06-28.

[4] ESA - European Space Agency. 2nd field test - esa-esric challenge. `https://www.spaceresourceschallenge.esa.int/copy-of-1st-field-test`, 2023. Accessed on 2023-06-28.

[5] Foxglove. Foxglove studio. `https://github.com/foxglove/studio`, 2021. Accessed on 2023-09-13.

[6] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. IEEE 802.11s: The WLAN Mesh Standard. *IEEE Wireless Communications*, 17(1):104–111, Feb. 2010. Conference Name: IEEE Wireless Communications.

[7] K. Ideas. Leo Rover - build and program your own robot. `https://www.leorover.tech/`, 2023. Accessed on 2023-07-31.

[8] IEEE - Standards Association. 802.11s-2011 - ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 10: Mesh networking, 2011.

[9] M. Labbe and F. Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2), 2019.

[10] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 05 2022.

[11] Y. Maruyama, S. Kato, and T. Azumi. Exploring the performance of ros2. In *Proceedings of the 13th International Conference on Embedded Software*, page 1–10, Pittsburgh Pennsylvania, 2016. ACM.

[12] L. Parker. Alliance: an architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2), 1998.

[13] L. E. Parker. *Multiple Mobile Robot Systems*, page 921–941. Springer Berlin Heidelberg, 2008.

[14] D. van der Meer, L. Chovet, A. Bera, A. Richard, P. J. Sánchez Cuevas, J. R. Sánchez-Ibáñez, and M. Olivares-Mendez. Realms: Resilient exploration and lunar mapping system. *Frontiers in Robotics and AI*, 10, 2023.

[15] L. Yang and S.-H. Chung. HWMP+: An Improved Traffic Load Sheme for Wireless Mesh Networks. In *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, pages 722–727, June 2012.

[16] T. A. Yuya Maruyama, Shinpei Kato. Exploring the performance of ros2. *Frontiers in Robotics and AI*, 10, 2016.