The Lunar Rover Mini: towards a Versatile, Open-Source Mobile Robotic Platform for Educational and Experimental Purposes

1st Sam Bekkers
Institute for Robotics and Mechatronics
German Aerospace Center
Weßling, Germany
sambekkers_98@outlook.com

2nd Michelle Bettendorf
Institute for Robotics and Mechatronics
German Aerospace Center
Weßling, Germany
mibettendorf@yahoo.com

3rd Josef Reill
Institute for Robotics and Mechatronics
German Aerospace Center
Weßling, Germany
Josef.Reill@dlr.de

4th Riccardo Giubilato
Institute for Robotics and Mechatronics
German Aerospace Center
Weßling, Germany
riccardo.giubilato@dlr.de

5th Armin Wedler
Institute for Robotics and Mechatronics
German Aerospace Center
Weßling, Germany
Armin.Wedler@dlr.de

***Abstract*—In this paper we introduce the Lunar Rover Mini (LRM), a ground-based mini robot based on the ExoMars rover developed by ESA. The goal of the LRM project is to serve as a low-cost educational and experimentation platform, not only for EXOMARS-like locomotion and mechanics, but also for intelligent robot programming and operation. With a software framework based a variety of open-source components developed at the DLR Institute of Robotics and Mechatronics, the LRM is an easily transportable mobile device that hosts an infrastructure to substitute and deploy modular software packages. As current focus in the research domain is on highly autonomous functionalities, a testing platform that provides the necessary hardware is much sought after. In combination with its modular software framework, the LRM serves as a platform for students, researchers and engineers to develop and test new software packages, with the aim of continuously improving its capabilities.**

Among the ongoing projects is the implementation of a visual-inertial SLAM algorithm, which simultaneously estimates a map of the rover's environment and its past trajectory by fusing information from captured stereo image pairs with measurements from an on-board IMU. A recently finished contribution is the development of a Human-Machine Interface for teleoperating the rover over an external controller, consisting of an extended smartphone with a 6 DoF control device over the phone's USB port. This successfully developed controller might potentially be deployed on other robots, which demonstrates the ultimate purpose of the LRM project.

***Index Terms*—robotics, autonomy, exploration, SLAM, shared autonomy, teleoperation**

I. Introduction

The field of robotic space exploration is constantly growing. There are currently several rover missions on the way, and one of the largest European exploration missions, EXOMARS [1], is struggling with the current political situation for its launch.

Fig. 1: The Lunar Rover Mini 2

In the meantime, the EXOMARS rover has already reached a very high level of development. Many other examples of new exploration robots are currently being developed, such as unmanned aerial vehicles, lava tube exploration vehicles and mobile underwater robots. Autonomous navigation methods and advanced intelligence are also being developed for these future missions. The German Aerospace Center (DLR) is currently preparing a 30kg rover for JAXA's MMX (see [2],

[3]) mission to be launched in 2024. Later this decade, DLR, ESA and many others are focusing on developments for lunar exploration [4]. All these concepts and mobility systems need to be developed, evaluated and verified. Building and launching these space systems is extremely expensive, so a way is being sought to demonstrate these concepts and methods through low-cost prototypes at the DLR for scientific, technical and educational purposes. This paper presents the development and integration of such system based on the ExoMars rover: the Lunar Rover Mini (LRM). The LRM platform provides the same software and high level development stack, as the software known from the DLR robotic hardware. The system includes the Realtime Links-and-Nodes (LN) Middleware with its LN Manager[1], provides Robot Operating System (ROS) modules and several interfaces to it, uses Simulink and the RealTime Workspace for the control tasks and provides an adaption to the mission state machine tool RAFCON [5]. Furthermore, in this publication also a number of recent contributions are presented, among which are the extension to a visual-inertial SLAM system and the development of a human machine interface for teleoperating the LRM.

A. Motivation

As early as the end of the 1990s, DLR launched the robot ASURO [6], where ASURO stands for "Another Small and Unique Robot from Oberpfaffenhofen". ASURO was one of the first Educational robots on the market and has already been sold more than 30,000 times, making it one of the most widely used learning robots on the market. The ASURO robot is more of an electrical/mechatronic development kit, as its low price of under €50 per kit has been well-used in schools for training and in private homes for early adopters as a development toy. The ASURO is now part of the DLR student laboratory [7] for STEM promotion of young talent and is implemented in the Lab training with the ASUROnaut experiment [8], in which a teleoperated ASURO must be controlled via an artificial space environment. The market for educational robots is rapidly growing, of which a good overview is provided by the collection of Fachhochschule Nordwestschweiz FHNW [9]. A comparative project is the EXOMY [10], which allows users to download, experiment and add onto the existing open-source material, including the 3D printed design as the ROS infrastructure. Another product is the VIAM[2], which has a similar mission of bringing a robotic platform on the market for anyone to grow and expand their software writing and robot programming skills.

In the robotics domain, both in planetary context and in mobile robotics in general, many developments in highly automated mission planning, autonomous navigation, and innovative human-machine interface are taking place. The focus of development, training and research is currently shifting from mechatronic solutions to information processing challenges. ASURO however was never intended for learning high level autonomy tasks. The LRM provides the infrastructure for this high-level task development, and will thus serve as a successor to the ASURO robot.

In addition, LRM developments have been part of the ROBEX [11], ARCHES [12] and the current iFOODis [13] project, which focus on the dissemination, training and promotion of young talent, and therefore experiments have been conducted in the summer schools of these projects.

The current development of DLR's planetary exploration strategies focuses on establishing a cooperative, heterogeneous team of robots. Especially in the experimental development phase, the simultaneous operation on multiple complex robots is a time-consuming and demanding, and also dangerous, undertaking. In addition, the professional DLR hardware is expensive and often limited in availability, restricting the accessibility to these platforms for students in particular. The LRM intends to close this gap and serves as an early evaluation platform for initial evaluation purposes, using the same software environment as in the large systems. Furthermore, the project aims to open up all components in such a way that such a system can be purchased by those interested in research and education for no more than €1000.

Moreover, the goal of the system is to create an environment in which many already developed and open source software components can be used (LN [14], RAFCON [5], ROS [15]) and the research-relevant activities related to cooperative robots with their challenges in mapping, localisation, task and work division can be further explored. The very powerful open-source DLR tools such as RAFCON and LN thrive in a community to be deployed and used. This system could help to build one.

The DLR LRM project was launched in 2015 at DLR Oberpfaffenhofen and has since been supported and accompanied by more than 20 students. It is a future goal to bring a fully open-source system to the market, for any robotics engineer to work with.

II. System Overview

A. Mechanics

Contrary to the larger, robustly designed LRU hardware, it is desired to create a small, lightweight platform that is easy to transport, set up and run experiments on. This is achieved by 3D printing as much of the body parts as possible. Due to the small dimensions of 36 × 26 × 39 cm ($L \times W \times H$) and the material choices, the LRM2, as rendered in figure 1, is perfectly suited for experiments and test missions mentioned in VI. The design of LRM is inspired by ESA's Rosalind Franklin[3], consisting of a main body that is supported by a triple bogie suspension system. This mechanical suspension system consists of three passive bogies that freely rotate around their central connection to the main board, to adapt to the uneven structure of the surface below. This passive suspension smoothens the lateral displacement of the centre

[1]https://gitlab.com/links_and_nodes/links_and_nodes
[2]https://www.viam.com/product/platform-overview
[3]https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Exploration/ExoMars/ExoMars_rover

TABLE I: LRM Components and availability

Category	Name	Available	Notes	Link
Hardware	Intel NUC (V7)	✓	On Board Computer	https://www.intel.com/
	Intel RealSense D435i	✓	Stereo Camera & IMU	https://www.intelrealsense.com/depth-camera-d435i/
	Body PCB	✗*	Communication & Power management	-
	Bogie PCB	✗*	Steer and Drive Control	-
	Faulhaber 2619 006 SR	✓	Wheel actuators	https://www.faulhaber.com/en/products/series/2619sr/
	Hitec HS-422 Deluxe	✓	Pan Tilt actuators	https://hitecrcd.com/
Navigation	Realsense tools	✓	Camera driver, depth sensing	https://www.intelrealsense.com/sdk-2/
	RTABMAP	✓	SLAM Framework	https://introlab.github.io/rtabmap/
	RM-Autonav	✗**	Internal tool for autonomous navigation	-
Infrastructure	Links and Nodes	✓	Platform-agnostic process manager	https://gitlab.com/links_and_nodes/links_and_nodes
	ROS Melodic	✓	Open-source robotic middleware	https://www.ros.org/
	Cissy	✗	Internal CI/CD Manager	-

* Not yet available, but foreseen to be released in the near future
** Can be substituted by the ROS navigation stack http://wiki.ros.org/navigation

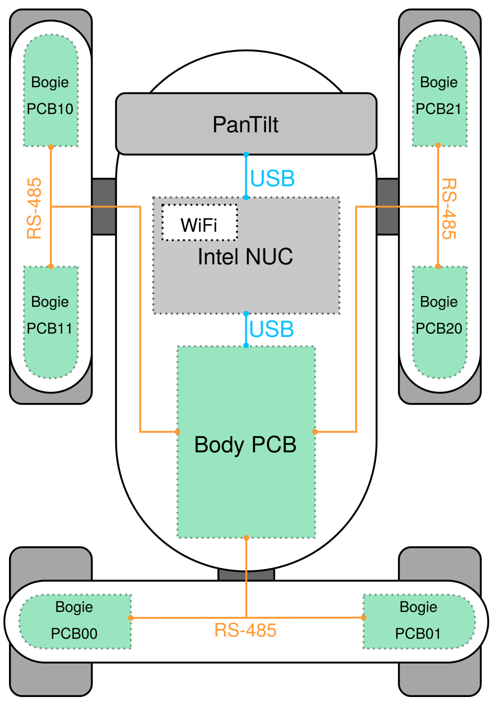


Fig. 2: Rover System Overview. Dashed components are installed underneath a protective cover.

of mass and ensures that all six wheels remain in contact with the ground to maintain full control while driving over rough or irregular surfaces. Inside each bogie are two servo motors, controlling the steering angle of the two wheels mounted underneath that bogie. The wheels are driven by a servo motor placed inside each wheel.

The main body houses the On-Board-Computer, a body PCB, a WiFi antenna and a 14.9V LiPo battery. This battery provides power all on-board energy consuming components. Mounted on top of the main body is the PanTilt unit which houses an Intel RealSense D435i. The PanTilt unit is controlled by two servo motors controlling the horizontal (pan) orientation and vertical (tilt) orientation.

B. *Electronics*

As the main focus of the system is on hosting collaborative robotic exploration algorithms rather than testing innovative hardware solutions, the chosen hardware components all mass-produced electronics. Figure 2 shows a system and hierarchical overview of the rover. The On-Board-Computer (OBC) of the system is a standard Intel NUC with an i7 processor running an Open-Suse Leap 15.4 operating system. The OBC runs the high level perception, navigation and autonomy functions. The Ethernet input port on the NUC can either be used to directly plug in an Ethernet cable to the RMC institute network, or to connect the on-board WiFi antenna. Via the institute net operators can establish an SSH connection to the NUC in order to start the desired processes. During missions outside of the institute, a mobile network is set up to which the rover's WiFi antenna connects, as well as the operator's laptop. Within this mobile setup the operator can also connect over SSH to the rover. The NUC computer runs the high level control algorithm and has a direct connection via USB-C to the perception unit (Intel RealSense) which is mounted on the pan-tilt. Thinking in command hierarchy levels, the NUC decides on the next desired values like steering angle, wheel speeds, and field of view. Via USB connection a serial communication port to is established to the Body PCB.

The Body PCB is the node that collects up the sensor data and manages the communication to all actuators, being the 12 servo motors for wheel control and the 2 servo motors for commanding the Pan-Tilt unit by PWM signals. Additionally the Body PCB hosts a low cost inertial measurement unit and provides the power conditioning and distribution unit that provides all necessary voltages from a single battery. Each Bogie PCB is connected to this central Body PCB via RS485 communication and operates one steering actuator and one drive actuator each. Both the body and bogie PCB designs are currently not yet available, but this is expected to change in the future.

The motors used in these actuator units are SR-series DC motors from Faulhaber with integrated gearing (ratio 1:207) and position encoder (16 increments per motor revolution). On each Bogie PCB there is a H-bridge driver chip from Texas Instruments (DRV8833) to drive one steering and one traction

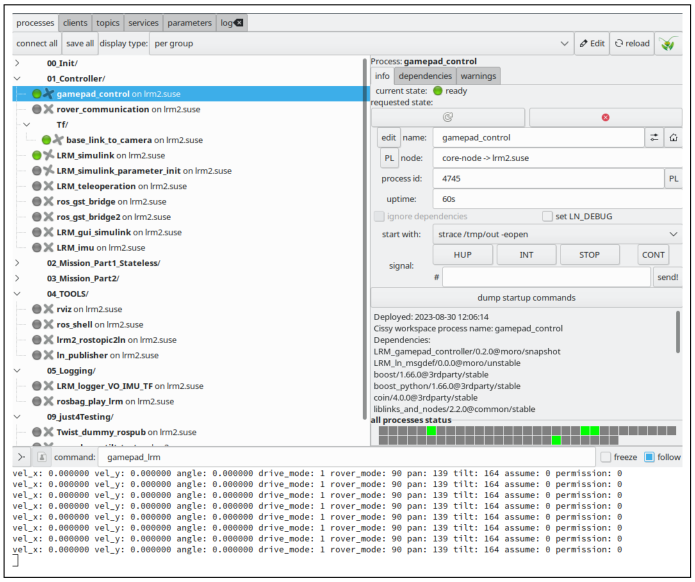


Fig. 3: The Links and Nodes Manager: on the left, a hierarchical set of processes, with a coloured status indicator. On the bottom, a console output for the process under investigation. On the right, the dependencies of the investigated process in terms of Conan packages.

motor each and measure the currents and positions. As the absolute position of the steering angle is a crucial information to drive the rover, this angle is measured by a 12-bit rotary position sensor. This measurement principle is also used to detect each of the three relative angles between bogie and main body of the rover.

C. Software framework

The LRM shares the high-level software infrastructure with the Lightweight Rover Unit (LRU) [16], a planetary exploration rover developed at the DLR Institute of Robotics and Mechatronics (DLR-RMC). The design principles that affect the software infrastructure are inter-operability and modularity. Except for system-specific components, this paradigm allows to exchange software packages between robotic platforms, developed at the institute, and enabling a fast development cycle. The framework is composed of several elements:

- **Links and Nodes**: a process manager targeted at the deployment, on a host or remote robotic system, of a complex and hierarchical set of processes. Through a graphical user interface, the manager allows to define their execution order, monitor their status, and provide dedicated inter-process, as well as real-time, communication capabilities. Links and nodes allows us, in the context of the LRM, to manage the execution of an heterogeneous set of processes, that can be written within, e.g., ROS or Matlab Simulink indiscriminately. The software is released as an open-source package, see Table I
- **Cissy** (Continuous Integration Software System): to allow a fast development workflow, a CI/CD (Continuous Integration / Continuous Development) pipeline is created, that includes several open source components. *Conan*[4], an open source package manager, to allow sharing within the institute and robots, of executables and libraries compiled for multiple profiles and architectures. *Jenkins*[5], an automation server to manage the execution of building binaries, coordinating a variety of build servers. *Artifactory*[6], an artifact manager, that allows storage, archiving, and distribution of built packages. The Cissy pipeline allows therefore not only an agile solution to fix, deploy and test software packages on the rover, but also define precise status of the functional software portfolio, through enforcing strict version requirements, and dependencies, of all the Conan packages that the robot needs for, e.g., perception, navigation, teleoperation, etc.

The software framework operates on the OBC. We established the data flow between various components via the two middlewares: Links and Nodes, and ROS (Robot Operating System) Melodic [15]. All the perception, navigation and mapping functions are implemented using ROS. Communication of low-level, and low-latency control data happens using Links and Nodes topic, while exchange of high-level, and high-bandwidth, sensor data happens through ROS topics. On board of the LRM, typically around 20 software processes are simultaneously and in parallel executed. In order to manage these processes, the process manager of Links and Nodes (LN-manager) is used to monitor the execution status and process outputs of ROS and LN-processes running on the rover. Additionally, this LN-manager offers the possibility to define run and restart dependencies as well as conditions for expected process behaviour. This provides a clear overview whether all required components are operating normally during operations.

III. Rover Operations and Control

To control the locomotion of the rover, three control nodes are developed. Switching between these controllers is easy as a result of LN's modular framework, as the operator can manage and start the desired control node in the LN manager 3. The first control node is based on a handheld Logitech Wireless Gamepad F710. The operator's control inputs are sent over Bluetooth to a USB receiver that is directly mounted to the NUC. The received signals are converted to a universal velocity command vector (*geometry_msgs/Twist*-type messages). This general velocity command allows for smooth future development of additional control nodes, establishing a flexible framework to test and develop new control packages.

The second control option is the autonomous controller, based on the autonomous navigation stack developed at DLR-RMC. The navigation stack comprises several components:

- **local planner**: a path-planning algorithm, computes the path of lowest cost towards a defined goal, using a fast implementation of the A$*$ planner. The path is computed over cost-maps, generated by a local mapping pipeline

[4]https://conan.io/
[5]https://www.jenkins.io/
[6]https://jfrog.com/artifactory/

which produce rolling elevation maps using depth images and the output of state estimation.

- **motion controller**: given a path to follow, a motion controller publishes desired velocities depending on the drive mode (see Sec. III-A), to maintain the rover on the right path to the goal.
- **global planner**: when loop closure happen (see Sec. V-C), the map is corrected and the current goal position is validated. A global planner corrects the position of navigation goals taking into account the shift of the current robot pose estimate w.r.t. a global navigation frame.

The third control node is based on a smartphone paired with a 6 DoF control device, on which will be further elaborated in section V-B.

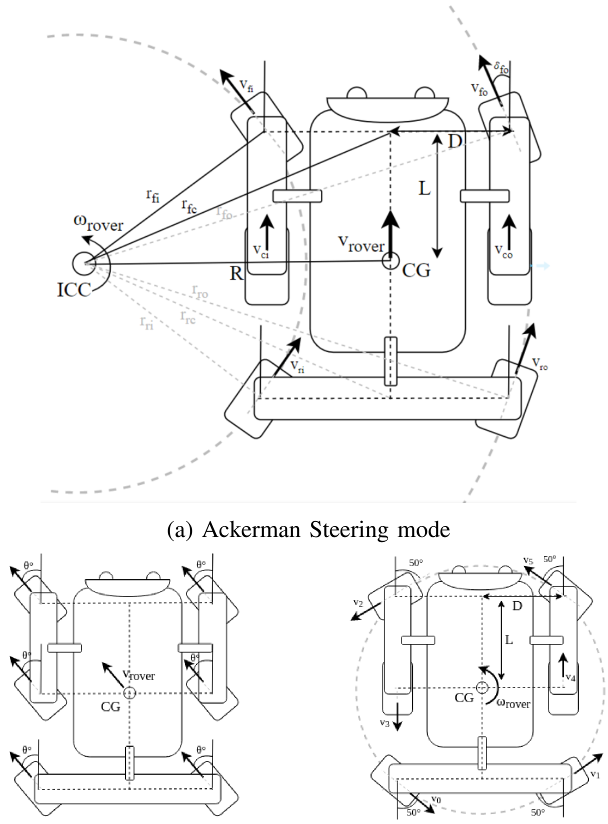
A. High-level controller

The Simulink control model communicates with other LN-nodes using custom built S-functions. These S-functions define publishers and subscribers which enables Simulink to listen to the messages generated by various controllers and published in the LN network. As multiple controllers can publish *Twist* messages in simultaneously, the model acts as a gatekeeper to select which signals from which controller should be passed on to the kinematic model. The handheld controllers always have priority over the autonomous control stack, but they can temporarily grant ”permission” to the model to listen to the autonomous navigation messages. This is done by a manual click of a button on the Gamepad. In case of potential dangerous movements by the autonomous controller, control is easily shifted back to manual control using the same button, which allows operators to quickly take over manual control.

Besides filtering the controller input messages, another core responsibility of the high-level controller is to transform the command controls from any controller into a set of motor commands for the 12 servo motors that manipulate the steering angle and driving speed of each wheel. With the aim to enhance manoeuvrability in different scenario’s, three driving modes are implemented. Depending on the chosen driving mode, the Simulink model generates the requested control commands for every motor. The operator can easily switch between driving modes during operation and can choose between *Crabwalk mode* (straight-line driving in any given direction), *Ackerman mode* (turning in a car-like motion) or *Rotation mode* (pure rotation on the spot). During autonomous navigation, the Simulink model itself is tasked with selecting the desired driving mode, based on the received commands from the autonomous navigation stack.

IV. Localisation and Mapping

For safe autonomous navigation, an accurate map of the environment is needed in which the rover is able to localise itself. The LRM solves this SLAM (Simultaneous Localisation and Mapping) problem by employing a *visual SLAM* pipeline, which analyses image data from a stereo camera setup. The RealSense D435i houses both a stereo camera setup as well as



(a) Ackerman Steering mode

(b) Crabwalk mode

(c) Rotation mode

Fig. 4: The three driving modes for locomotion [17]

an internal IMU, which makes it a suitable device for the LRM and allows for the potential development of a *visual-inertial SLAM* algorithm. An advantage of the stereo camera setup is that it does not suffer from *scale ambiguity* and *scale drift*, two issues that monocular camera setups have to deal with. On our rover, the existing open-source RTAB-Map[7] module is utilised which optimises the map using a GTSAM back-end [18]. The optimisation problem is solved by constructing a so-called *factor graph* based on the motion constraints between two consecutive camera poses, retrieved by a Bundle Adjustment (BA) algorithm. BA estimates the transformation between two camera poses based on the relative motion of observed features in two subsequent images. Employing GTSAM enables us to create and update the map estimate during real-time operations, which is frequently shared with the navigation modules over the ROS infrastructure to create the occupancy grid visualised in figure 5. This occupancy grid is used by the navigation stack to compute a collision-free path to a operator defined destination in Rviz.

V. Contributions

As it is one the primary tasks for the LRM is to serve as an experimentation platform, it has gratefully been used as a development platform for several student theses. During the

[7]https://introlab.github.io/rtabmap/

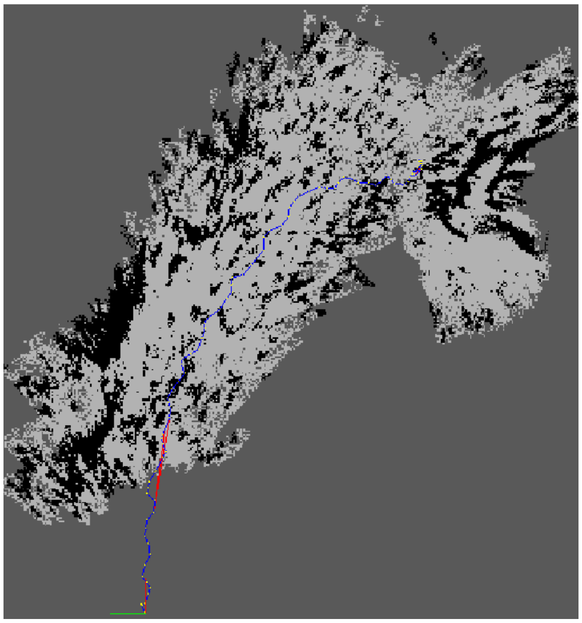


Fig. 5: Occupancy grid of the Dried Moon Lake, Vulcano, Italy. Black areas are unavailable locations, light grey is free space and dark grey is unexplored area. [17]

project lifetime, both the design and hardware aspects, and the software packages of the LRM have seen various iterations, all contributing to the current project state. A selection of recent contributions is discussed in this section.

A. LRM1

The OBC on the earlier version of the LRM was an Intel Atom E3845. During the first attempts to deploy ORB-SLAM2 as mapping agent, the computational power of this OBC proved insufficient as the the OBC lacked the capability to process captures images at the desired framerate, detoriating the robustness of the map. Since then, the Intel NUC has served as OCB. The earlier design of the LRM is also slightly smaller than the current version. It had the dimensions 25 $\times$ 20 $\times$ 24 cm ($L \times W \times H$). As an increased wheel diameter was desired for (1) higher driving speeds and (2) the ability to traverse terrain covered in small pebbles, the rover size was scaled up. The new wheels are covered with a 3D printed soft rubber tire, which provide traction on rocky surfaces. The main body is fully closed in order to protect the internal components from environmental factors. Because the body and bogie PCB's remained the same, the software is easy to transfer. Except for some dimensional parameters, the software stack is identical to LRM1.

B. Teleoperation

The teleoperation node is developed around a smartphone connected to a 6 DoF controller via the USB port. The controller, called spacemouse, was developed by colleagues at DLR. The Human-Machine Interface consists of sending the control inputs to the rover and displaying the camera stream and the SLAM map on the smartphone.

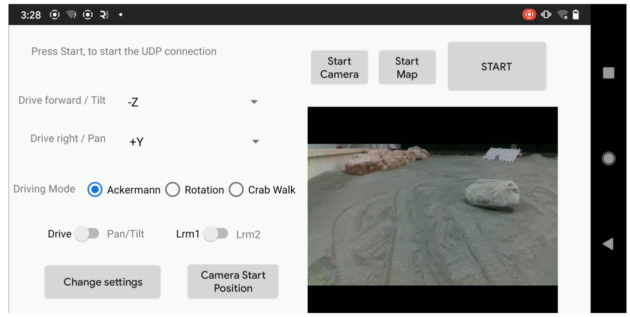


Fig. 6: Final GUI of the Human-Machine Interface

The spacemouse consists of a flexible suspended knob within the housing that connects to the smartphone. The knob can be manipulated in all directions, establishing 6 DoF. In the GUI, the operator then defines how these DoF are translated to velocity commands and how the PanTilt unit is controlled. Also, within the GUI can be specified which rover is wished to control, i.e. LRM1 or LRM2.

For transferring the camera stream and map three different options were implemented and evaluated: VNC, ROS-mobile and GStreamer. VNC is a software tool for remotely controlling a computer, for which is an app is available on the Google Playstore. As the NUC runs an Open-Suse Linux distribution, this option can be considered. Within the LN manager there is a process for visualising the camera and map using RViz, which could show the map on the smartphone over VNC. Another option for map display is the open-source ROS-mobile app which displays ROS topics. To minimise the video stream delay, images are compressed before publishing on these topics. Lastly, the third approach is to compress the media, send it to the smartphone, decompress and then display the received data. This can be done using GStreamer pipelines. For the smartphone app, Java with a C/C++ extension is needed. As the spacemouse control can't be integrated into the VNC and ROS-mobile app and therefore a separate control app would be needed, whereas GStreamer can be implemented in the spacemouse control app.

For the final Human-Machine Interface for teleoperating the rover, the spacemouse control app and the three different media stream options were evaluated by performing two user case studies: one on controlling the PanTilt movement over the spacemouse and one on teleoperating the rover through a parkour using the three developed options. Based on camera stream delay, app robustness, intuitiveness and the confidence afterwards regarding one's ability to teleoperate the rover, the majority of participants in the user case study preferred GStreamer. The final GUI of the spacemouse control integrated with GStreamer is displayed in fig. 6.

C. Visual-Inertial SLAM

An ongoing contribution is the extension of the existing visual SLAM algorithm to a visual-inertial system. Visual-inertial systems show several advantages in terms of map accuracy and robustness. Since there is already an Inertial

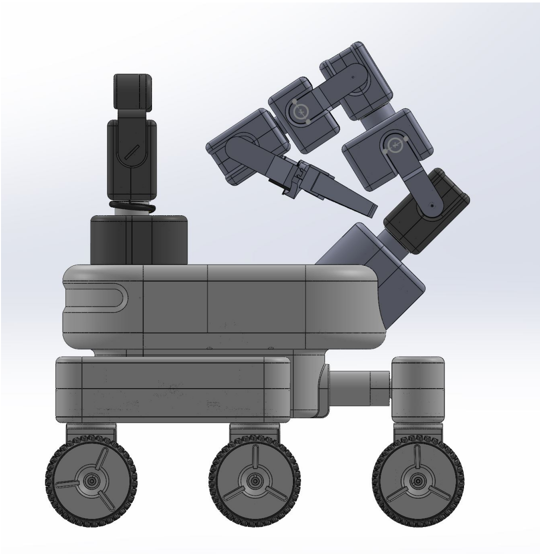


Fig. 7: Design for the Robotic Arm on LRM2 [20]

Measurement Unit (IMU) on board as a component of the Realsense D435i, this is an interesting and logical next step to enhance the mapping capabilities of the LRM. A downside of the use of such semi-professional IMU, despite being efficient for bringing down the hardware costs, is that measurements from this device are contaminated with measurement biases. Simple integration of the linear accelerations and angular velocities would thus results in a large IMU drift, leaving the resulting position estimate useless. A high accuracy can be still achieved by *tightly* coupling the measurements through jointly optimising over the camera and IMU measurements. This is possible because the biases are considered to be slowly varying Gaussian walk. Hence, if an estimate of the bias at a previous timestamp can be obtained, the measurements of the next timestamp can be cleaned up by removing this bias, assuming that the biased has not changed much. Bias estimation is a functionality that is included in the already employed GTSAM optimiser. It conveniently allows for the incorporation of these IMU measurements in the optimisation problem by performing on-manifold measurement pre-integration, following [19].

D. *Robotic Arm for Manipulation*

A planned project for the near future is the development and installation of a robotic arm on the rover body. The aim is to be able touch, grasp or move objects in order to manipulate the environment. The current design is a 7 DoF manipulator, including an end effector gripper. The design as displayed in figure 7 is based on an inverse kinematic calculation model to derive the size of the necessary motors. A future project is the actual installation on the LRM, including the electronics wiring and message protocol alignment to control the arm. Possibly this could be integrated in the DLR RAFCON [5] system.

VI. iFOODIS Vulcano Summer School 2023

In July 2023 a group of students participated in the yearly organised Vulcano Summer School. During the ten-day field trip, the operational functionalities of LRM2 were tested on the vulcanic soil of Vulcano Island, Italy. This is a representative environment for Moon- and Mars-like terrains which makes it suitable for test missions of the rover. A lot of insights regarding the design of the rover and potential hardware failures were gained, which prove to be very useful for next iterations of the LRM design. Also, part of the user case study about the teleoperation module was conducted on this field trip. Moreover, during this trip valuable data was collected by recording *rosbags*, which can be fed into the Visual-Inertial SLAM pipeline after completion to obtain a more accurate map of this environment. This map can be stored and used for future missions on Vulcano.

VII. Conclusion and Summary

In this paper we presented our mini rover project, the LRM. It has already proven to be a useful platform for students, scientists and computer engineers in the past years, accommodating to their common need of having a small, mobile hardware platform in order to develop the high-level autonomous robotic intelligence software the research domain is currently focused on. Using the open-source available DLR-RMC software Links and Nodes, a modular software framework is established on which users can develop, deploy and test new software packages. This results in continuous improvements of the rover capabilities and intelligence, such as the extension of the SLAM pipeline to a visual-inertial SLAM algorithm. This will increase the accuracy of the generated maps, providing more useful maps for safe autonomous navigation. Another successful contribution is the new suitable Human-Machine Interface for a smartphone, including a 6 DoF controller. Based on user case studies, three different media transfer methods were analysed, from which resulted that the GStreamer method outperformed the others.

VIII. Acknowledgements

We would especially like to thank the Helmholtz Association and DLR for supporting this activity. A special thanks goes to the colleagues who all helped out and responded to all the questions and assistance that came their way. This work was supported by the Helmholtz Association, project alliance ROBEX (contract no. HA-304), project ARCHES (contract no. ZT-0033), and project iFOODis (contract no. KA-HSC-06 iFOODis).

References

[1] "Exomars factsheet. european space agency, 2022," https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Exploration/ExoMars/ExoMars_Factsheet, accessed: 14.09.2023.

[2] S. Barthelmes, T. Bahls, R. Bayer, W. Bertleff, M. Bihler, F. Buse, M. Chalon, F. Hacker, R. Holderried, V. Langofer, R. Lichtenheldt, S. O. Moser, K. Sasaki, H.-J. Sedlmayr, J. Skibbe, L. Stubbig, and B. Vodermayer, "Mmx rover locomotion subsystem - development and testing towards the flight model," in *2022 IEEE Aerospace Conference, AERO 2022*. IEEE, März 2022. [Online]. Available: https://elib.dlr.de/186923/

[3] S. Ulamec, P. Michel, M. Grott, U. Böttger, H.-W. Hübers, N. Murdoch, P. Vernazza, K. Özgür, J. Knollenberg, K. Willner, M. Grebenstein, S. Mary, P. Chazalnoel, J. Biele, C. Krause, T.-M. Ho, C. Lange, J. T. Grundmann, K. Sasaki, M. Maibaum, O. Küchemann, J. Reill, M. Chalon, S. Barthelmes, R. Lichtenheldt, R. Krenn, M. Smisek, J. Bertrand, A. Moussi, C. Delmas, S. Tardivel, D. Arrat, F. IJpelaan, L. Melac, L. Lorda, E. Remetean, M. Lange, O. Mierheim, S. Reershemius, T. Usui, M. Matsuoko, T. Nakamura, K. Wada, H. Miyamoto, K. Kuramoto, J. LeMaitre, G. Mas, M. Delpech, L. Celine, A. Rafflegeau, H. Boirad, R. Schmisser, C. Virmontois, C. Cenac-Morthe, A. Besson, and F. Rull, "A rover for the jaxa mmx mission to phobos," in *70th International Astronautical Congress, IAC 2019*. International Astronautical Federation, Oktober 2019, pp. IAC–19. [Online]. Available: https://elib.dlr.de/130664/

[4] N. Schmitz, M. Anand, F. Altieri, S. Barber, J. Biele, C. H. van der Bogert, J. Flahaut, S.-E. Hamran, H. Hiesinger, K. H. Joy, F. Postberg, M. C. D. Sanctis, S. Schröder, S. Ulamec, A. Wedler, M. Landrø, F. McDonald, and J. Carpenter, "Concepts for a polar ice and volatiles prospecting mission in the framework of esa's el3 program." in *European Lunar Symposium 2022*, Mai 2022. [Online]. Available: https://elib.dlr.de/191626/

[5] "Dlrs rafcon, opensource epl framework to develop your robotic tasks using an intuitive graphical user interface"," https://dlr-rm.github.io/RAFCON/, accessed: 14.09.2023.

[6] "Wikipedia: Asuro ein frei in c programmierbarer roboter..." https://de.wikipedia.org/wiki/ASURO, accessed: 14.09.2023.

[7] "The dlr_school_labs, out of the classroom - into the lab," https://www.dlr.de/schoollab/, accessed: 14.09.2023.

[8] "Das asuronaut experiment im rahmen des dlr schoollabs im dlr," https://www.dlr.de/de/bilder/2017/1/asuronaut-der-kleine-roboter-asuro-erkundet-die-marsoberflaeche_25958, accessed: 14.09.2023.

[9] "Fachhochschule nordwestschweiz fhnw, programmieren lernen - tools für die schule," https://www.code4school.ch/, accessed: 14.09.2023.

[10] "Exomy: A low-cost 3d printed rover," https://www.hou.usra.edu/meetings/isairas2020fullpapers/pdf/5044.pdf, accessed: 15.09.2023.

[11] "Project webpage of the helmholtz alliance Robotic Exploration of Extreme Environments – ROBEX"," https://www.robex-allianz.de/, accessed: 14.09.2023.

[12] "Project webpage of the helmholtz project arches – "Autonomous Robotic Networks to Help Modern Societies"," https://www.arches-projekt.de/, accessed: 14.09.2023.

[13] "Project webpage of the helmholtz project ifoodis – "improve food supply through robotic and sensory monitoring systems"," https://www.dlr.de/rm-neu/en/desktopdefault.aspx/tabid-18808/, accessed: 14.09.2023.

[14] "Link and nodes middleware - process management and realtime capable communication middleware for distributed systems"," https://blog.pal-robotics.com/wp-content/uploads/2014/09/Schmidt_WSAH.pdf, accessed: 14.09.2023.

[15] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[16] M. J. Schuster, S. G. Brunner, K. Bussmann, S. Büttner, A. Dömel, M. Hellerer, H. Lehner, P. Lehner, O. Porges, J. Reill, S. Riedel, M. Vayugundla, B. Vodermayer, T. Bodenmüller, C. Brand, W. Friedl, I. L. Grixa, H. Hirschmüller, M. Kassecker, Z.-C. Marton, C. Nissler, R. Felix, M. Suppa, and A. Wedler, "Towards autonomous planetary exploration: The lightweight rover unit (lru), its success in the spacebotcamp challenge, and beyond," *Journal of Intelligent & Robotic Systems*, November 2017, video available at https://link.springer.com/article/10.1007/s10846-017-0680-9 and at https://www.youtube.com/watch?v=wCTkSxcna8o – Open Access: This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. [Online]. Available: https://elib.dlr.de/116749/

[17] A. Ricardez Ortigosa, "Systems integration with autonomous navigation of the lunar rover mini for a space demo mission," 2023.

[18] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam)

[19] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, *IMU preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation*, 2015.

[20] M. L. Groß, "Design und konstruktion eines roboter armes für den lunar rover mini," 2023.