ISCTE IUL
REPOSITÓRIO
INSTITUTO UNIVERSITÁRIO DE LISBOA

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2025-05-14

Deposited version:
Accepted Version

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Luz, R., Coelho, G., Campos, M., Abrantes, R., Querido, R., Pereira, M....Ventura, R. (2024). Remote operations and streamlining communication in mars analog missions: Robot-agnostic augmented interface for live annotations during teleoperated exploration tasks. In 2024 International Conference on Space Robotics, iSpaRo 2024. (pp. 342-348). Luxembourg: IEEE.

Further information on publisher's website:
10.1109/iSpaRo60631.2024.10688246

Publisher's copyright statement:
This is the peer reviewed version of the following article: Luz, R., Coelho, G., Campos, M., Abrantes, R., Querido, R., Pereira, M....Ventura, R. (2024). Remote operations and streamlining communication in mars analog missions: Robot-agnostic augmented interface for live annotations during teleoperated exploration tasks. In 2024 International Conference on Space Robotics, iSpaRo 2024. (pp. 342-348). Luxembourg: IEEE., which has been published in final form at https://dx.doi.org/10.1109/iSpaRo60631.2024.10688246. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

Remote Operations and Streamlining Communication in Mars Analog Missions: Robot-Agnostic Augmented Interface for Live Annotations During Teleoperated Exploration Tasks

Rute Luz[1,2], Gonçalo Coelho[1], Mariana Campos[1,3], Rui Abrantes[1], Ricardo Querido[1], Margarida Pereira[1], Inês Sousa[1], Jéssica Corujeira[1,3], José Luís Silva[3,4], Rodrigo Ventura[1]

***Abstract*— Human space exploration typically requires considerable teamwork involving multiple disciplines in science and engineering. Effective and efficient field systems and information flow with the support teams are paramount for success, particularly in crewed missions such as the upcoming Artemis mission (Moon) and planned Mars missions. Effective human-robotic teams will be crucial in such scenarios to allow humans to extend their capabilities to previously unreachable or unsafe regions through teleoperation. In this paper, we make a technological description of a new teleoperation interface for such planetary exploration, addressing the challenges of field rover teleoperation and ensuring a streamlined information flow between the field and support teams within the AMADEE-24 Mars analog mission (March 2024). In particular, our interface (MEROP) is used by the astronauts inside the habitat and features (1) visually augmented mission goals (regions of interest and safe paths) provided by the MSC before each task, (2) live geo-referenced annotation and automated output for the MSC (3) situational awareness displays and widgets, and (4) direct and semi-autonomous robot control methods. Furthermore, we describe the system architecture that allowed the integration of our interface with a mission rover (Mercator), whose development is entirely independent of our team. Finally, we present the user feedback from the analog astronauts acquired during preparations for the upcoming AMADEE mission.**

I. INTRODUCTION

A. Context

In the realm of tasks performed via remote robot teleoperation, effective communication and collaboration between (non-scientific) operators in the field and the mission support centre (MSC) are critical to ensure the maximum scientific outcome. However, current conventional communication methods, reliant on post-task analysis or chat-based exchanges between the field and the mission support centre, pose significant challenges regarding complexity [1], [2]. Insights provided by astronauts or operators during mission runs are often lost (e.g., they usually only remember the last moments of the run), exacerbating the difficulty of integrating field knowledge into the flight plan within the exploration context.

This paper introduces a streamlined teleoperation system architecture to improve communication and annotation in exploration missions. The system enables the integration of mission goals from the MSC and live annotations on the image and map from operators (i.e., analog astronauts during AMADEE-24). These goals (i.e., augmented regions/points of interest for exploration and safe roads for navigation) improve task performance by aiding exploration. In addition, the system automatically extracts information from experimental runs, providing valuable data such as rover trajectory and signal strength information along the trajectory. By outputting this information in a standardized Geographic Information System (GIS) format supported by the MSC, the system ensures compatibility with widely used Earth-based tools, facilitating integration into existing workflows.



Fig. 1: MEROP teleoperation hardware used by the analog astronauts during the AMADEE-20 mission and the upcoming AMADEE-24 mission.

Moreover, there is a need to ensure a reliable and effective remote operation of the rover, as communication latency and losses are common challenges [3], [5]. Also, operators have poor situation awareness (SA) due to loss of perceptual information, which can lead to mission-ending situations [4], [5]. Thus, the MEROP system aims to address the challenges of communication latency by providing two modes of control (direct and semi-autonomous) and a reliable communication protocol. It also provides operators with augmented information and haptic feedback to improve operator SA.

While initially developed for AMADEE missions, the proposed teleoperation system has broader applicability. It can be adapted to other fields where non-scientific operators contribute valuable insights or expert users annotate information for public consumption post-run. Examples include underwater exploration, where tagging interesting reefs for post-task visualization can benefit from this approach.

[1] Institute for System and Robotics, Instituto Superior Técnico de Lisboa, Portugal
[2] HRI-Lab, ESA-ESTEC, the Netherlands
[3] ITI - Interactive Technologies Institute, Portugal
[4] Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Portugal



Fig. 2: The Mercator rover from TÜ Graz and the mission habitat (background) of AMADEE-20. (Image Credits: Florian Voggeneder from ÖWF)



Fig. 3: Analog Astronaut inside the Mars Habitat teleoperating the Mercator Rover with the MEROP OCU during the AMADEE-20 mission.

B. AMADEE-24 Mars Analog Mission

The AMADEE-24[1] Mars analog mission is organized by the Österreichisches Weltraum Forum[2] (ÖWF). It will be hosted in Armenia in partnership with the Armenia Aerospace Agency[3] from March 7th to April 5th of 2024. It is an analog expedition to a Martian-like terrestrial surface to develop exploration mission architectures and prepare for future Mars missions.

This mission comprises several multidisciplinary teams (e.g. Flight Planning, Remote Science Support, Medical teams, etc.) at the Mission Support Center (MSC) in Austria. These teams need constant information from the field (test site) to understand how the mission and experiments are going and monitor the analog astronauts (AAs). To simulate the communication delay between Earth and Mars, there is a constant communication time delay of 10 minutes between the MSC and the Habitat (field).

During the mission, six highly trained analog astronauts, live isolated inside the Habitat (the analog mars outpost), see Fig. 2, which they can only exit by going on extra-vehicular activities (EVAs) using an analog spacesuit. These EVAs are resource and time intensive since the donning and doffing, take at least two hours and a half and require the six AAs. The EVA is then executed by two AAs, with at least three on the communication systems and health monitor consoles. As such the use of teleoperated rovers (Fig. 3) for initial exploration tasks reduces the high demands on AAs and mission time, compared to an EVA.

[1]https://oewf.org/en/amadee-24/, accessed on 05/02/2024
[2]https://oewf.org/en/, accessed on 05/02/2024
[3]https://armenianasa.net/, accessed on 05/02/2024

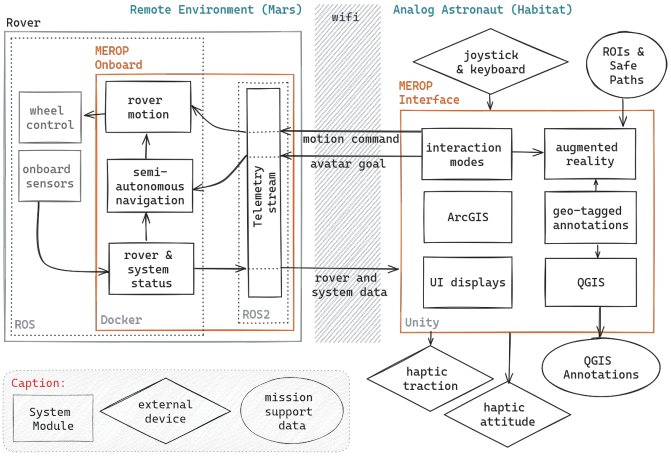


Fig. 4: Modules of the MEROP teleoperation system: (1) onboard components ensure navigation, status feedback and remote communication; (2) interface components integrate MSC data, annotations, rover control, and haptic feedback.

C. Main Contributions

This paper presents the MEROP (More Effective Remote OPerations) teleoperation system comprising an Operator Control Unit (in-house build hardware and graphical user interface) for the AAs and the onboard software deployed in the mission rover. The implemented modules and user experience design were iterated based on lessons learned during AMADEE-20 [6] and valuable feedback from AAs (operators) during the two mission dress rehearsals. The contributions of the presented work are the following:

1) A robot agnostic teleoperation system - the teleoperation system can be used with mobile robots with different characteristics;
2) A system architecture that allows low latency, reliable and resilient communication protocol between operator control unit and robot;
3) System output of geo-referenced annotations and mapping of Wi-Fi signal strength to GIS map layers;
4) Avatar-based teleoperation - a semi-autonomous teleoperation approach to lower operator's workload;
5) Augmented reality (AR) representation of obstacle distance, regions and points of interest, and safe paths that display the mission goals.

Finally, the MEROP system has been tested and deployed in two mobile robots with different characteristics. A Pioneer-3AT-based robot is used in-house for system validation, and the Mercator rover is taking part in AMADEE-24. However, the proposed system is not tailored to any specific robot and provides an agnostic approach that can be deployed in various robots.

II. SYSTEM DESCRIPTION

A. System Architecture

The MEROP system architecture, represented in Fig. 4, comprises two main components: First, the MEROP interface (Figs. 5, 7,8 and 11), used by the AAs inside the habitat,

allows to remotely control the rover while providing rover-related information (e.g., localization, signal data, operation mode) and augmenting mission-related goals in the video stream (e.g., paths and regions of interest). Second, the MEROP onboard software system is deployed on the rover, creating a bridge between a generic rover and the MEROP teleoperation interface. This one subscribes to the rover sensor data, does the onboard processing of the rover and system status (e.g. traction and signal strength), and computes the motion commands depending on the selected operation mode (direct or semi-autonomous)

This system architecture was designed to comply with a series of assumptions regarding the rovers: (1) we are operating a rover in which we should not change the system or make only minimal changes, i.e. install new hardware or software dependencies; (2) deployment and integration of the system with a new robot are typically limited to a couple of days; (3) most common robotics platforms in the community still resort to ROS 1; and (4) the rover can receive low-level motion commands and provide standard outdoor sensor data, i.e., camera image, geo-localization and heading, IMU data, wheel odometry, and lidar or point cloud data.

The resulting system design follows a modular approach that isolates the MEROP onboard system within a Docker container, interacts with the ROS 1 system of the rover (section II-F) and resorts to ROS 2 for robust communication protocols (section II-G). On the habitat, the MEROP system resorts to Unity, ROS2 and ArcGIS third-party packages for the UI displays (section II-B) and augmented reality features (sections II-D and II-C).

B. MEROP Teleoperation Interface

The teleoperation User Interface (UI) was developed in Unity and its functionalities and layout iterated based on lessons learned from AMADEE-20 and AMADEE-24 dress rehearsals.

The teleoperation User Interface (UI) was developed in Unity, and its functionalities and layout were iterated based on lessons learned from AMADEE-20 and AMADEE-24 dress rehearsals. The final UI layout, shown in Fig. 5, emphasizes the video feed and uses the peripheral areas for other displays. We iterate the position] and type of the displays to facilitate the interpretation of the system status and reduce adaption, training time, and mental effort to interact with different UI features. As such, we created the bottom area of system status, inspired by the commonly recognized Windows bar and widgets, and mapped the joystick buttons (input method) directly to the position of the interactive features of the UI (see Fig. 6). For example, to switch the operation mode, the operator clicks on the top-right button of the joystick and sees the result of that action on the top-right of the screen. Finally, the UI displays and functionalities are listed and described below:

1) **Main Camera View**: Video stream from the rover's main camera, by default a front facing camera.
2) **Secondary Camera View**: Video stream from a secondary camera, usually a back facing camera, with a smaller size compared to the main camera. Pressing the top-left button swaps the main and secondary cameras views.

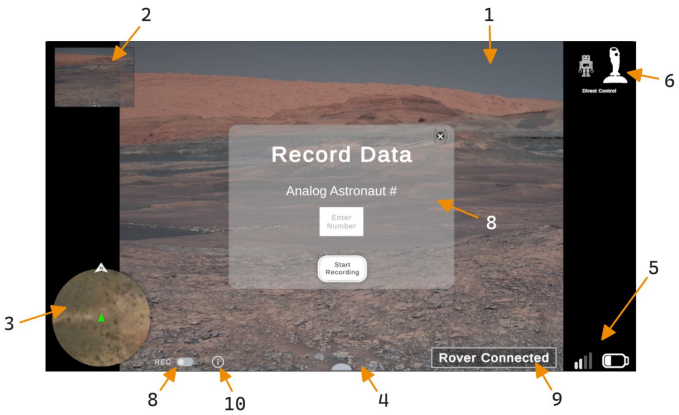


Fig. 5: Teleoperation interface layout: 1-Main Camera; 2-Secondary Camera; 3-Map; 4-Proximity Display; 5-Rover Status Indicator; 6-Operation Mode; 8-Recording Menu and Indicator; 9-Notification messages; 10-Information tab.

3) **Map**: Displays the a top-view of the area using a ego-referenced configuration centered in the rover's position and oriented towards North. ROIs, safe paths and geo-tagged annotations as shown in Fig. 8. Pressing and holding the bottom-left button maximizes the map, providing a wider view of the area and ROIs. The map is created through the ArcGIS package which allows to integrate image and elevation data of the surface.
4) **Proximity display**: AR display regarding nearby obstacles displayed in both main and secondary views covering a total of 360-degree field of view (180-degree in each view). The display is nearly transparent during nominal situations and displays different sizes and colors (yellow to red) depending on the proximity and position of the obstacle (see Fig. 7).
5) **Rover status indicators**: Two icons indicating the rover's battery status and onboard signal strength.
6) **Operation mode**: Display of the currently active operation mode: direct control or the avatar-based operation, shown in Fig. 11 and described in section II-F. Pressing the top-right button switches the mode.
7) **Annotation menu**: Radial menu overlaid in the center of the interface (see Fig.8) which allows the operator to make geo-tagged image and text annotations (see Fig. 8). Pressing the bottom-right button enables and disables the menu and the way hat switch selects the annotation type (see Fig. 6).
8) **Recording menu and indicator**: Menu window appears on start-up and triggers the onboard telemetry recording. When the operator presses "start recording", the REC widget toggles to green. The recording is stopped by clicking the REC toggle widget.
9) **Notification messages**: temporary pop-up messages, warnings, and connectivity confirmations.
10) **Information Tab**: pop-up informational window with

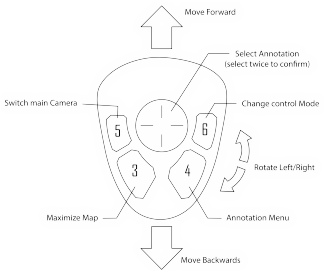


Fig. 6: Joystick mapping.

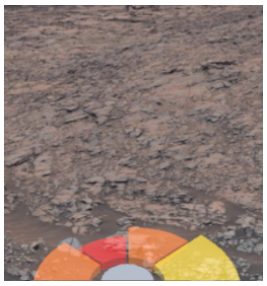


Fig. 7: Proximity display.

the joystick controls and button mapping schematic (Fig. 6). The schematic is overlaid on the image and removed by clicking the "(i)" widget.

C. Augmented Regions of Interest and Safe Paths

As part of the mission goals, the MSC defines a series of Regions of Interest (ROIs) that the AAs should explore during the experimental run. The communication approach during AMADEE-20 was chat-based interfaces and drawings on a map. However, this often leads to miscommunications or difficulty integrating the information during the run. To streamline the communication between the MSC and the field crew regarding these mission goals, the MEROP UI uses augmented reality to display this information. Thus, facilitating the visualization and integration of the mission goals within the remote exploration.

The MSC transfers GIS shape files to the habitat (Mars) before the experimental run. These are input for the AR displays (see Fig. 8) of the MEROP UI and define ROIs and safe paths. The AR computation resorts to the ArcGIS package for Unity to place these geo-referenced GIS data on a virtual surface, which contains both imagery and elevation information. We can generate an AR overlay superimposed on the video feed by combining this process with Unity virtual cameras and the rover telemetry (latitude, longitude, heading and camera orientation). Finally, for better situational awareness, the AR is displayed on both front and back-facing views (see Fig. 8).

D. Augmented Geo-Tagged Annotations

The debriefing process is the second challenge in effective communication between support and field teams. This is the process where the astronauts report the relevant findings and challenges of the experimental run back to MSC. However, since this is only performed at the end of the run, AAs tend only to remember events that occur near the end or forget to report back due to their busy schedule and high workload. Moreover, even when they reported back to MSC, the communication channel was a chat interface, which was insufficient to encompass the knowledge acquired during the run. As a result, during AMADEE-20, the identified Points of Interest were not considered in subsequent flight plans.

To streamline the task debriefing and integration of field knowledge into the flight plans, the MEROP UI allows the operator to make geo-localized photo and text annotations during the run (radial menu in Fig. 8). New annotations

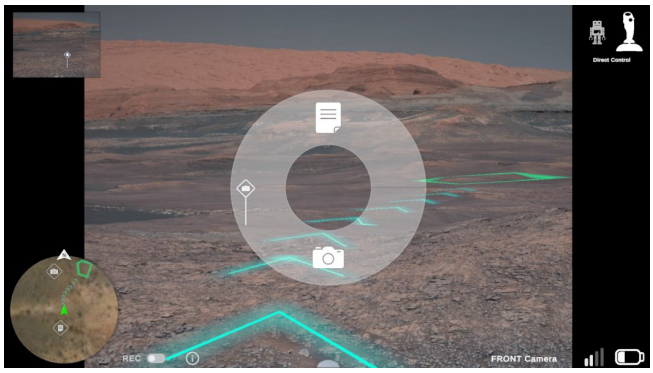


Fig. 8: Interface with augmented ROIs and safe paths, from the MSC, annotation radial menu, an augmented geo-tagged annotations from the operator.

are visually confirmed with a new augmented beacon on the current rover position with the corresponding icon (see Fig. 8).Moreover, in addition to the manual annotations made by the AAs, the UI automatically generates GIS layers containing rover trajectory and signal strength along the run.

To integrate geographical annotations in Unity and export a comprehensive GIS project, we developed an integration scheme that connects Unity to QGIS. This required integrating data and functionalities from ROS2, Python, Pipes and XML projects as represented in Fig. 10 and described below:

1) **Unity Interface with ROS Subscriptions**: A C# file embedded in the Unity Interface to interact with ROS 2 topic data, particularly GPS data for the geo referencing.
2) **Annotation Trigger in Unity**: User interaction triggers the initiation of an annotation, which prompts the transmission of the GPS coordinates and the annotation's content via Pipes to a specified Python script. For text annotations, the written text is sent via pipes, and for photo annotations, the integration includes capturing a screenshot from the main view, stored in

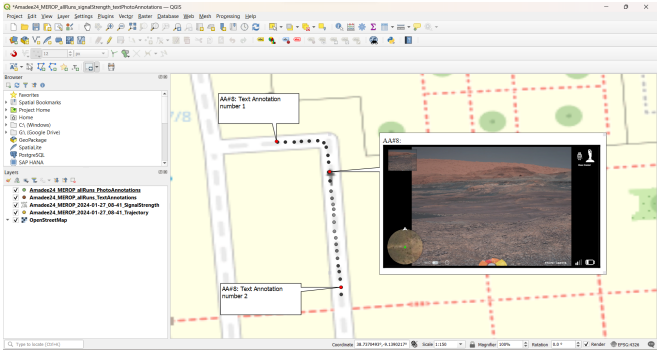


Fig. 9: Example of GIS Project visualisation: manual annotations of text and photo and automatic trajectory (points) and signal strength (gradient) layers.

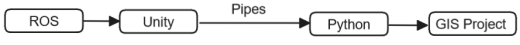


Fig. 10: Scheme of the annotations pipeline: ROS based telemetry and geo-annotations are integrated in a GIS project.

a predefined path and sent via pipes.

3) **Python Processing Module**: The Python script receives the data through Pipes. This data is input to generate the GIS project programmatically with annotated layers. This step involves leveraging the XML script of a base GIS project. The Python script dynamically adds additional XML code to this base, facilitating the creation of the final GIS Project.

The automatic output of the exploration run from the MEROP UI is a GIS Project containing several telemetry and annotation GIS layers (see Fig. 9). Finally, in order to generate a self-contained compressed file that can be sent to the MSC overnight, the AAs resort to QGIS[4] to export the generated project.

E. Operation Modes

To control the rover, our system allows the operator to easily switch (one button click) between two types of operational strategies: (1) direct control, where joystick motions map directly to the rover's motion (see Fig. 6), and (2) an avatar-based approach [9], where the joystick motions map into the motion of an augmented avatar (see Fig. 11). This augmented avatar represents a high-level navigational, i.e., the position and orientation we want the robot to reach.

Our previous research shows that implementing a hybrid operational approach allows the operator to adapt to different situations, tasks and environments [9]. Moreover, previous work [11] showed that operators using autonomous or semi-autonomous approaches often request low-level approaches (direct control) to handle situations where autonomy fails. Thus, we integrated both operational strategies into the MEROP system. Finally, to achieve this hybrid system, our onboard system integrates several modules to allow direct control, avatar-based navigation (autonomous navigation) and switching functionalities (multiplexer-based pipeline).

F. Avatar-Aided Teleoperation

When in avatar mode, the operator controls and sees augmented on the image a visually similar representation of the rover, i.e., avatar. This one represents the navigation goal and enhances awareness of the robot's proportions in the remote environment. By design, unlike typical discrete waypoint navigation, the avatar moves continuously to provide an engaging experience, close to a direct operation, maintaining the input device (joystick) while ensuring a low workload. During this operation, the rover onboard modules constantly compute and execute a way to reach this dynamic goal.

Similar to the other AR elements of the UI, the avatar resorts to ArcGIS integration within Unity to provide its location (latitude and longitude) and orientation (north heading). This one is then sent to the MEROP onboard system and integrated into the semi-autonomous modules, namely:

1) **Rover and Avatar Localization:** We resort to their geo-localization (latitude and longitude) for rover and avatar localization. However, we need to project this data in a Cartesian referential to plan a path and consequent motions to reach the avatar. We adopted the UTM (Universal Transverse Mercator) coordinate system to project our geo-localization data here. Thus, the rover and avatar localization are projected in a UTM-based referential with the origin near the habitat.
2) **Path Planning and Robot Control:** The MEROP onboard system implemented a Model Predictive Control (MPC). This one considers a dynamic model, a cost function for the trajectories, and certain limits on the trajectories (constraints) [10] to compute the path and velocities to reach the avatar goal. We adopted this greedy planning approach because it does not require a known costmap of the remote environment and resorts to a generic model description, requiring minimal adjustments. Thus, aligning with our agnostic approach.

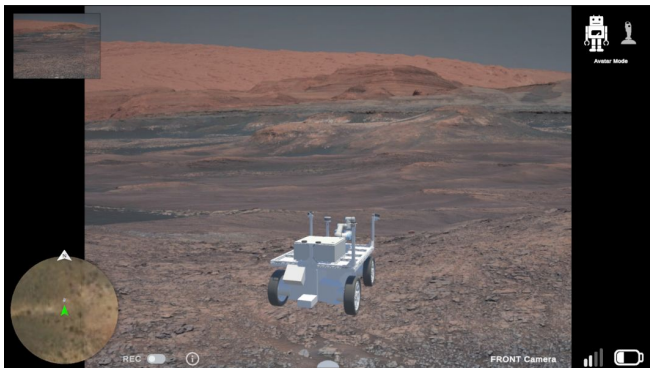


Fig. 11: Interface with augmented avatar, particular case of operation of the Mercator rover.

The implemented MPC approach approximates the optimal control problem by an optimization problem over a sliding window, that minimizes a cost function, $l(x, u)$, subject to several constraints Eq. 1.

$$\begin{aligned} \arg\min_{u} \quad & \sum_{n=0}^{N} l(x_k, u_k) \\ \text{s.t.} \quad & x_k = f(x_{k-1}, u_k) \\ & x(0) = x_0 \\ & u_k \in \mathcal{U},\ k \in [0, N] \\ & x_k \in \mathcal{X},\ k \in [0, N+1] \end{aligned} \tag{1}$$

where f denotes the discrete time system dynamics, $\mathcal{U}$ the actuation feasible set, and $\mathcal{X}$ the feasible state space. For each Δt, the MPC solves the problem N times consecutively, in which N is the dimension of the predictive window, hence the predictive factor. The first control given by the predictive window is the control that will be given to the robot.

The system used in the MPC follows the well known differential drive kinematics:

$$f(x_{k-1}, v, \omega) = x_{k-1} + \Delta t \cdot \begin{bmatrix} cos(\varphi) & 0 \\ sin(\varphi) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

[4]https://qgis.org/en/site/, accessed on 17/02/2024

where x_k is the Cartesian position vector at the present time, x_{k-1} is the last position vector measured. Linear velocity is represented by v and angular velocity is represented by ω, which are the control variables for this model, and φ is the variation angle between the orientation on time k and $k-1$.

The cost function, which encodes the goal of the controller, considered is $l(x,u) = |x-x_g|^2_Q + |u|^2_R$. The variable x corresponds to the position of the robot, x_g is the position of the desired goal and u is the control variables, linear and angular velocity. For the MEROP system, the cost function corresponds to the distance between the actual position and goal position and power of the control variables. Through iterative tuning, the following gains were chosen:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.005 \end{bmatrix} \qquad R = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{3}$$

The MPC problem is solved iteratively in real time, using an interior point method optimization solver: the IPOPT implementation in the casadi library. For the system to be as robot-agnostic as possible, the MPC uses a short window, $N = 100$, considering a $\Delta t = 0.1$, because of the system frequency, $10Hz$. These parameters took into account the CPU usage, which needs to be as low as possible. For the computers tested on, CPU usage was not a problem for MPC.

G. Remote Communication

After assessing the communication challenges from AMADEE-20 and recognizing the limitations of relying solely on TCP for communication (ROS 1), we transitioned to a UDP-based system to enhance real-time responsiveness, accepting occasional message loss. This shift was enabled by using ROS2 for the telemetry stream bridge[5] between the rover and the remote teleoperation interface. ROS2 uses DDS as its middleware, designed to be transport-agnostic and allows users to configure QoS settings. We selected the "Sensor data" QoS profile to optimise for our requirements, prioritizing best-effort reliability and a smaller queue size. As a result, considering these configurations and utilizing the default JPEG compression algorithm of ROS2, the system ensures video streaming of two RGB images at 640x480p resolution and 30 frames per second, with a throughput of approximately 4 MBps.

H. MEROP Teleoperation Case

Given the isolation and limited conditions of the habitat where the MEROP teleoperation hardware needs to be deploy, we designed and built a self-contained case, shown in Fig. 1, to transport and provide a practical hardware setup for experimental runs. This case contains: (1) laptop with the MEROP UI, (2) Extreme 3D Pro joystick to interact with the UI and move the rover and avatar (3) in-house built attitude [12] and traction [13] haptic devices, (4) video camera, to record the experimental setup, and (5) several replacement parts. This teleoperation case was previously deployed and validated during the AMADEE-20 mission [6].

[5]https://github.com/ros2/ros1_bridge, accessed on 17/02/2024

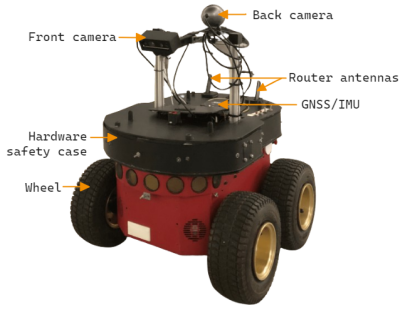


Fig. 12: Kayjay - Rover maintained by the MEROP team.

III. SYSTEM DEPLOYMENT

MEROP system was deployed on two different robots: (1) the KayJay rover (Fig. 12) maintained by us and served as a validation platform for the MEROP system, and (2) the Mercator rover (Fig. 2), taking part in the AMADEE-24 and whose development is entirely independent of our team. One the one hand, the KayJay rover is a low budget robotic platform with minimal functionalities and sensors. Thus, providing us with a scenario of limited hardware and computational resources, which our system should be robust to endure. On the other hand, the Mercator rover [8] is a complex and robust mobile platform design for outdoor navigation, but where the available time for integration time extremely short. This system integration was limited to the three days of the mission's dress rehearsal (January 2024).

We resort to a configuration file that defines the rover and environment-specific characteristics to deploy our system in different robots. Thus making the bridge with our robot-agnostic system. Mainly, it defines the ROS topics of the robotic platform (camera, GNSS, IMU, lidar, etc), the naming of tf frames of different system components (e.g., camera frames for AR elements), information about UTM zone for the Cartesian projections, and network details (network name and frequency for the network scanning). A 3D model of the Mercator rover was also integrated into the UI as the augmented avatar (Fig. 11).

Given the implemented system architecture based on the deployed docker container, we could set up the MEROP onboard system in Mercator within a few hours. The remaining time was used to make adjustments (e.g., setting fixed IP within the mission network), correct minor system bugs (incorrect GNSS coordinate system NED vs ENU) and teach the astronauts to set up and use the MEROP system. Debugging the system was particularly challenging due to the high number of people using the mission network simultaneously and its limited bandwidth.

IV. DISCUSSION

The proposed robot-agnostic teleoperation system has been successfully realized through iterative development. The system now offers significant enhancements to address limitations and challenges identified during AMADEE-20, such as communication issues, SA, and efficient run debriefing. The MEROP system facilitates communication between

the Mission Support Center (MSC) and the Habitat, streamlining the workflow for the Flight Planning team. Moreover, the system effectively reduces the operator's workload while ensuring a low latency, reliable, and resilient communication protocol between the operator control unit and the robot. Notably, the proposed architecture exhibits flexibility, enabling integration with various robots possessing diverse features. The MEROP system holds the potential for accommodating new robots in the future, though it is presently confined to the following robot's requirements: (1) ROS 1 middleware, (2) 2 onboard cameras, (3) GPS (Global Positioning System) receiver, (4) IMU (Inertial Measurement Unit), and (5) Lidar or point cloud data.

Both deployed robots use wheel-based locomotion. However, we hypothesise that the system could also be advantageous for other types of locomotion, e.g., legged robots or even flying robots. In such cases, the ROIs and POIs augmentation continue to be relevant issues, and avatar-based navigation has also been demonstrated for drones.

Tests of the proposed system have been conducted in controlled environments, including laboratory settings and outdoor scenarios, as well as during both AMADEE-24 Dress Rehearsals. Apart from its technical attributes, such as flexibility and extensibility, it necessitates evaluation from a user-centric perspective through systematic user studies (scheduled for execution during the upcoming AMADEE-24 mission). These studies aim to statistically validate the proposed system's effectiveness for operators. However, it is worth noting that the technological description of the developed system holds significant scientific value for analogous projects.

V. CONCLUSIONS

This paper describes, in-depth, a robot-agnostic teleoperation interface for planetary rovers, integrating several features designed to streamline communication among team members. It includes features such as visual augmentation of the robot video stream with mission goals, geo-referenced shared annotations and situation awareness augmentation. It supports both direct teleoperation and semi-autonomous goal-based teleoperation.

At the time of writing of this paper, this interface is being deployed to the AMADEE-24 analog mission as part of the MEROP experiment, where analog astronauts will use it to control the Mercator rover for exploration missions.

The augmented live annotation interface and the mission support crew (including scientists and non-scientists) will integrate the GIS output into the exploration cascade of the mission. This will provide insights about the advantages of using such a pipeline for the exploration cascade, maximize scientific output and provide valuable information to future system iterations.

Initial feedback from the astronauts showed that iterations on the UI layout, button mapping and functionalities were intuitive and quick to learn. They also reported that they found the avatar approach promising, as they struggled with low-level commands during AMADEE-20 due to high and variable latency, and that they expect annotation features to reduce their post-task workload.

The AMADEE-24 will be crucial to identifying beneficial features and field challenges. Thus, addressing the current limitations of the system and the combination of results from both the user study and field mission will enable further system improvement.

ACKNOWLEDGMENTS

The authors would like to thank Francesca Willcocks, Adrian Belli, Oscar Lafuente, and Seda Özdemir-Fritz from ÖWF for their support in the preparation of our system, feedback and GIS preparation. This work was supported by Fundação para a Ciência e Tecnologia (FCT, Portugal), through PhD Grant PD/BD/150634/2020, FCT Projects UIDB/50009/2020, UIDB/04466/2020 and UIDP/04466/2020, and "Aero.Next Portugal" project.

REFERENCES

[1] Jennifer Esper and Carrie Olsen. "A Study of International Space Station Ground/Crew Communication Methods with Applications to Human Moon and Mars Missions," AIAA 2007-6271. AIAA SPACE 2007 Conference & Exposition. September 2007.

[2] Vera Hagemann, Lara Watermann, Florian Klonek, Christiane Heinicke, Communication quality affects performance of astronauts and support teams through increased workload: Insights from the AMADEE-20 analog Mars mission, Acta Astronautica, Volume 210, 2023, Pages 162-175, ISSN 0094-5765, https://doi.org/10.1016/j.actaastro.2023.05.021.

[3] Dan Lester, Harley Thronson, Human space exploration and human spaceflight: Latency and the cognitive scale of the universe, Space Policy, Volume 27, Issue 2, 2011, Pages 89-93, https://doi.org/10.1016/j.spacepol.2011.02.002.

[4] Sophie Goemaere, Katrijn Brenning, Wim Beyers, Angelo C.J. Vermeulen, Kim Binsted, Maarten Vansteenkiste, Do astronauts benefit from autonomy? Investigating perceived autonomy-supportive communication by Mission Support, crew motivation and collaboration during HI-SEAS 1, Acta Astronautica, Volume 157, 2019, Pages 9-16, ISSN 0094-5765, https://doi.org/10.1016/j.actaastro.2018.11.048.

[5] Robin R. Murphy. 2014. Disaster Robotics. The MIT Press.

[6] Gernot Groemer, AMADEE-20 Mission Report v1.1, 08 Jan 2022, ÖWF, https://oewf.org/en//download/20628/?tmstv=1708185401, accessed on 16-02-2024

[7] AMADEE-24 Mission Information, https://oewf.org/en/amadee-24/, accessed on 17-02-2024

[8] Halatschek, R., Konanur Ramanna, K. D., Url, W., & Steinbauer, G. (2020). Universal Offroad Robot Platform for Disaster Response. Paper presented at 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics , Virtuell, United Arab Emirates.

[9] Luz, Rute, José Luís Silva, and Rodrigo Ventura. "Enhanced teleoperation interfaces for multi-second latency conditions: System design and evaluation." IEEE Access 11 (2023): 10935-10953.

[10] Caireta, I.M. Model Predictive Control for a Mecanum-Wheeled Robot in Dynamical Environments; Universitat Politècnica de Catalunya Facultat de Matemàtiques i Estadística, Universitat Politècnica de Barcelona: Barcelona, Spain, 2019.

[11] P. Schmaus, D. Leidner, R. Bayer, B. Pleintinger, T. Kruger, and N. Y. Lii, "Continued advances in supervised autonomy user interface design for METERON SUPVIS justin," in Proc. IEEE Aerosp. Conf., Mar. 2019, pp. 1–11.

[12] Corujeira, José, José Luís Silva, and Rodrigo Ventura. "Attitude perception of an unmanned ground vehicle using an attitude haptic feedback device." 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). IEEE, 2018.

[13] Luz, Rute, et al. "Traction awareness through haptic feedback for the teleoperation of UGVs." 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). IEEE, 2018.