AUTONOMOUS NAVIGATION APPLIED TO THE IGLUNA LUNAR ANALOGUE MISSION ON COLLABORATIVE ROBOTIC SYSTEMS

Mr. Maximilien Dreier[1, *], **Ms. Jasmine Rimani**[2], **Ms. Maanasa Sachidanand**[3], and **Mr. Gowtham Reddy Govind Reddy**[4]

[1] *ISAE-Supaero University of Toulouse, France,* cecenost@gmail.com

[2] *Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy,* Email: jasmine.rimani@polito.it, *Department of Aerospace Vehicles Designs and Control, ISAE-SUPAERO, Toulouse, France,* Email: jasmine.rimani@isae-supaero.fr

[3] *ISAE-Supaero University of Toulouse, France,* maanasa.sachidanand@student.isae-supaero.fr

[4] *ISAE-Supaero University of Toulouse, France,* gowthamg3g@gmail.com

[*] Corresponding author
Email: maximilienlp@gmail.com

Abstract

The desire to return humans to the Moon is driving research into autonomous systems to help establish a permanent lunar base. Space agencies, companies, and universities accept the challenge of studying solutions to enable human exploration of the cosmos. This paper presents the work conducted during the IGLUNA ESALab@CH Field Campaign, an activity coordinated by Space Innovation (formerly the Swiss Space Center). The IGLUNA analog mission is a testbed to demonstrate technologies for future lunar exploration. The CoRoDro team was responsible for developing an exploration system for lunar lava tubes, a potential human shelter in the Moon's equatorial zone. In the lava tubes, the main challenge is the inability of humans to track and communicate with robotic systems visually. Two earth analogs were used to demonstrate a lunar rover's and a thruster-propelled hopper's autonomous navigation operation: a small rover and a drone performing short flights to simulate the hopper. In the lava tube, the two systems will collaborate to analyze and provide a map of its internal structure. The hopper will give the rover a map via its propelled flights, avoiding obstacles in the lava tube to reach high-interest areas for analysis. This concept has been proven during the IGLUNA field campaign conducted on Mount Pilatus (Switzerland). This paper presents the path planning and SLAM (Simultaneous Localization And Mapping) algorithms implemented on both robots. The drone relies on path-planning algorithms to explore the unknown environment, while SLAM algorithms generate the map. Path planning on the drone acquires the boundaries and establishes the covering path for the exploration area. The SLAM algorithms include several modules that consider the rover's mobility capabilities to create a map that identifies reachable obstacles and regions. The map is shared with the rover and used to reach targets identified by the drone. Taking into account obstacles, targets, risks, and rewards, an optimum global path is generated. SLAM and path-planning algorithms enable the rover to update the map and detect obstacles that the drone missed. This work has been reviewed by experts from ESA, Space Innovation, and Airbus;; an analysis of its performance is presented in this paper. It has shown the interest of fully autonomous robotic missions in extreme and remote conditions, encountered both in space and on Earth.

Keywords: Autonomous Navigation, Collaborative Robots, Lunar Analog Mission, Space Exploration, Moon Exploration, SLAM, Path-Planning, Task-Planning, Computer Vision, Algorithms.

1. Introduction

NASA and ESA are partnering to land humans on the Moon by 2024 with the Artemis program[1] and to achieve sustainable missions by 2028[2]. The other central institutional space agencies from Russia, China, Japan, and India are also planning to establish a regular communication and navigation system between Earth and the Moon [3]. Space agencies such as NASA and ESA support and rely on private companies that are increasingly seeing economic opportunities on the Moon [4]. The knowledge gained from space research is unlocking numerous Earth-based services and products that are revolutionizing nearly all sectors of human activity, making the presence of humankind in space an economic booster on Earth as well as a driver of prosperity. But while research represents only an investment with delayed returns, the Moon proposes an in-situ economy that enables resource exploitation. Thus, the presence of humanity on the Moon would drastically reduce the cost of space research. At the same time, that in-situ economy would allow the industrialization of the Moon infrastructure, drastically reducing the cost of sustaining the connection between the Earth and the Moon, which has two significant consequences: enabling emerging entities such as nations or businesses to enter the space field[3], and unlocking a sustainable space exploration program of the Solar system and beyond. Today's technology level is high enough for universities and students to participate in space development, and CoRoDro has developed robotic systems for Lunar exploration and operations. For all those reasons, the Moon crystallizes all the stakes of the space field, and will undoubtedly see a permanent human colony on its surface within our lifetime. In this context, the CoRoDro student team at ISAE-SUPAERO in Toulouse, France, decided to tackle the challenge of human survival on the Moon, which is constantly bombarded by asteroids and radiation harmful to living beings. Geologists have identified lava tubes on the Moon's surface [5]. Formed by volcanic activity, they consist of underground caves that can be several hundred meters long and several dozen meters high. Protected from the said asteroids and radiation, and accessible through surface holes up to 500 meters wide, they represent a massive opportunity for humans to establish a permanent base there. But before constructing the base, an exploration mission must be conducted. For the apparent reason of safety and feasibility, that task won't be performed by humans but by robots. The robots will have to withstand the harsh conditions on the Moon. They will also have to operate with an unprecedented level of autonomy, as they might be disconnected from any communication and monitoring systems, as well as from any possible external navigation system (GPS), once they enter the lava tube.

CoRoDro developed, for that purpose, an autonomous navigation system and a task planner that autonomously decides how to reach the mission's objectives. Moreover, to increase the mission's success, CoRoDro chose to use two robots with different capabilities: a flying drone and a classical rover. We developed the algorithms needed to enable the collaboration of several robotic systems. The experiments were conducted in a lunar analog environment here on Earth, during the Igluna 2021[1] Field Campaign.

On the Moon, the drone would be propelled by thrusters, while here on Earth, we used a quadrotor for our experiments. Both the rover and the drone possess localization sensors (Tracking camera, wheel encoder, IMU) to help them localize in the environment, and depth sensors (Depth camera, LiDAR) to help them perceive their environment. The drone and the rover also have an embedded computer (Odroid XU4) that handles 100% of the required computations. Thus, the Control Center acted more as a monitoring entity and did not remotely control the robotic system. Its commands were limited to initiate the mission and to start certain phases of the mission, such as giving the green light to begin certain phases of the mission (mapping phase of the drone, exploration phase of the rover, visiting phase of the drone).

2. Case Study

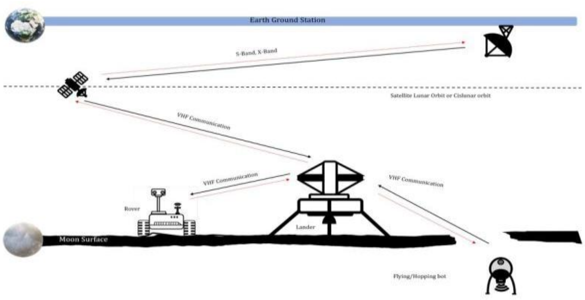

Figure 1: Design Reference Mission

The mission aims to demonstrate the potential of robotic systems and advance their operational autonomy to explore our universe while helping humans with critical tasks. To envision this, a region on Mount Pilatus serves as a testbed for implementing and testing autonomous algorithms and software. In this case, the systems include one drone and one rover. They act as robotic agents that work together to achieve common objectives. First and foremost, the two systems must be able to navigate the outdoor terrain collaboratively. Without external navigation aids such as GPS, the interdependence of the two systems must enable them to explore the environment. Secondly, the systems are

[1] IGLUNA 2021 Edition : https://space-innovation.ch/ igluna/

designed to define and explore the sites of interest. The drone's hovering capability is utilized to scan the environment with the attached sensors. The drone follows a grid-like path to scan and map the terrain. The map serves as a base for detecting sites of interest that can be accessed by the systems individually. The drone processes the map and sends it to the rover. The rover uses the postprocessed map to generate an optimized path to targets using the path-planning algorithm and to localize in unknown terrain. The drone and rover communicate and schedule operational tasks. The design reference mission is shown in Figure 1, and our analog testing site set-up is shown in Figure 2.

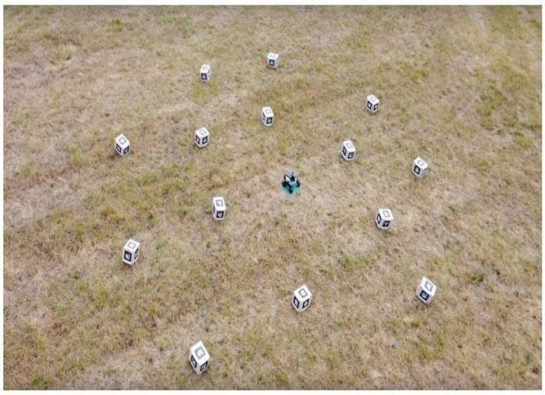

Figure 2: Outdoor testing site set-up

The drone is equipped with two different range cameras (T265, D435i) that help in scanning the environment/terrain and provide key driving inputs to the mission. The rover is composed of a lower part, the ”Leo rover”[2], the mobility platform. It is also composed of an upper part that embeds the rest of the hardware, such as a Lidar, depth-sensing camera, and T265 camera, which provide information on which the path-planning algorithm is driven, as well as a communicating WiFi system and a computer. The drone was fully assembled at ISAESUPAERO by our partners from the DCAS department. The rover's lower part was purchased, and our partners in the DCAS department of ISAE-SUPAERO fully assembled the upper part. Figure 3 shows photos of our two robotic systems, and Table 1 presents their different components.

The mission's criticality is further emphasized by the environmental constraints under which they operate. To simulate operational conditions on a remote planet, the systems do not use GPS data for any computations. The power consumption of each system had to be considered for optimizing operations. Intracommunications depend on the strength of signal reception, which can degrade at any time. The systems are designed to withstand cold and hot conditions while operating in harsh terrain. All these are necessary constraints for simulating missions on the Moon and other planets that present similar operational constraints. Verification and validation at each step lay the foundation for extending future mission capabilities.

Figure 3: Drone and Rover

[2] Leo ROVER :https://www.leorover.tech/

Drone Components	**Rover Components**
Depth Camera (D435i)	Computer (Raspberry Pi)
Tracking Camera (T265)	Controller (Husarion CORE2-ROS)
Communication	Leo Camera
WiFi System	Mobility System(Wheels and actuators)
Computer (Odroid XU4)	Communication Module
Flight Controller	Power System
Speed Controllers	Structural Frame
Motors	WiFi Module (payload)
Power System (Battery)	Sensors (Lidar, D435i, T265)
	Inertial Measurement Unit (payload)
	Power Booster(payload)
	Computer (Odroid XU4)

Table 1: Equipment List.

3. Autonomous Navigation Framework

The developed software is divided into three main modules. There is the Simultaneous Localization and Mapping (SLAM) module, which allows robotic systems to localize themselves in the environment and to scan it to generate a usable map. There is a TaskPlanning module. Based on AI, the module uses information retrieved from the field, along with the status of several robotic systems (battery, position, mobility capabilities), to generate a plan of action for each robotic system. It is the module that is the decision-maker and that provides the robots with orders to achieve the mission. Finally, the Path-Planning module uses the generated map and received navigation orders to move the robots across the environment optimally without injury.

3.1 Simultaneous Localization and Mapping algorithms

The main goal of the drone's SLAM algorithm is to perform a reconnaissance mission of the field to provide a complete map of the testing environment. The main goal of the rover's SLAM algorithm is to improve the provided map, if necessary, to create a robust obstacle avoidance system. Choosing the T265 camera for position retrieval was critical. Indeed, the V_SLAM algorithm embedded in that camera simplifies the mapping objective. Tests have shown that the V_SLAM algorithm provides a reliable position but not a reliable DEM map.

Our idea is to use the T265's position, but not its map. We also use an adapted depth sensor (D435i) to generate the DEM map. Figure 4 shows the process flow diagram of the SLAM algorithm.

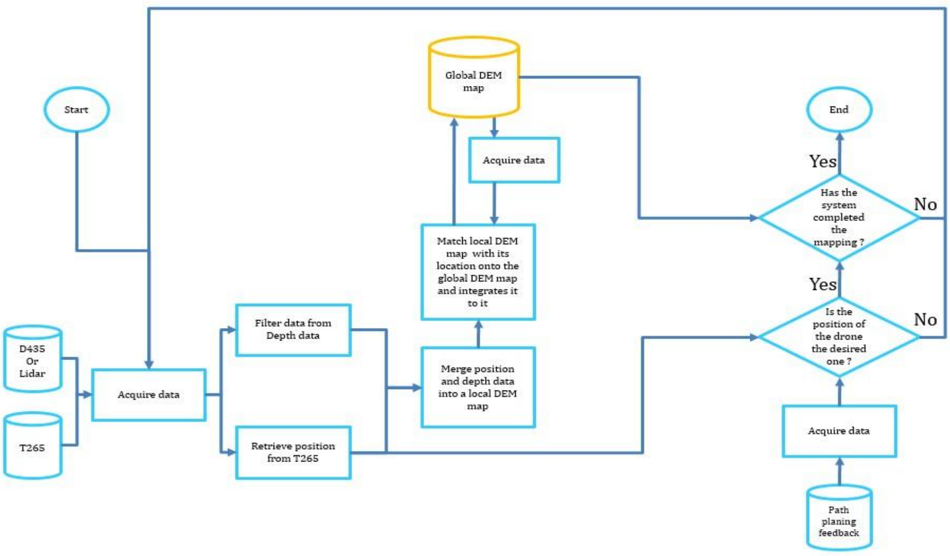

Figure 4: Drone's SLAM process flow diagram

The 3D point clouds generated by the D435i and localized by the T265 have been merged to create a 3D map of the environment using the octomap_server[6] algorithm. To improve the SLAM algorithm's performance, filters and optimization have been implemented. Reducing the amount of data generated with a voxel grid filter[7][8] and an extraction filter reduced the computational cost, enabling the drone's computer to process all generated data (Figure 5). Statistical denoising and outlier-removal filters [9][10] (Figure 6) were also implemented to remove aberrant data and improve the quality of the generated map.

The overall architecture of the mapping algorithms is shown in Figure 8 and presents all the steps from the drone's initial state to the generation of the final expected 2D map [11].

3.2 Real-time object detection

Goal-based autonomous navigation of robots on planets, the Moon, or asteroids enables them to detect points of interest, such as rocks. Rocks have been among the main substances on planets, providing a wealth of knowledge for planetary geology and helping detect whether the conditions on a celestial body are conducive to the existence of microorganisms. This collected information can also help decide which samples to collect and which to send back to Earth. The main objective of the CORODRO team is to send the rover to visit the Points of Interest (POIs) in its environment. These POIs or targets are cuboidal boxes measuring 36 cm × 28 cm × 28 cm and are used to simulate rocks on the Moon.

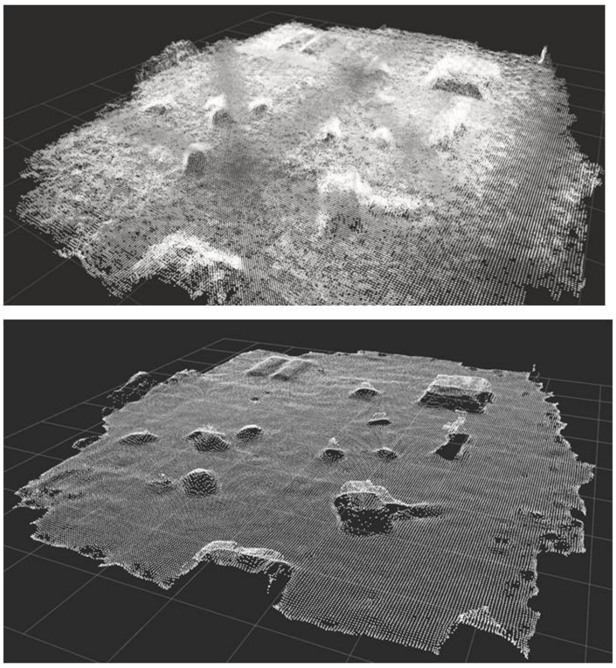

Figure 5: Example of voxel grid filter, statistical denoising, and extraction filter on a 3D point cloud

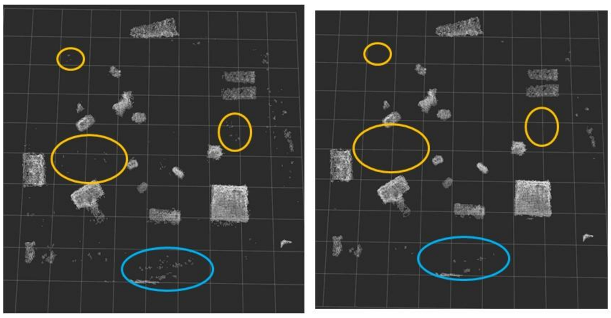

Figure 6: Example of outlier removal filters on a 3D point cloud

Due to their unique features, which make them highly recognizable, our SLAM algorithm is likely to detect them as landmarks. However, the software will not know whether a landmark refers to a POI. That is why it is important not to draw a parallel between POI and the landmarks used internally by the SLAM software.

We used the ARTag Detection software using the Robotic Operative System (ROS) package called "ar_track_alvar"[12] that detects AR-tags on the surface of the POI. The software has been integrated into both the rover and the drone. AR tags are fiducial markers that help us detect POIs. We placed AR tags on the surface of the POI and detected them as multi-tag bundles (a combination of AR tags as a single unit). The input for the software algorithm includes the AR tags detected in the camera (D435i or T265) image and the position of the drone and rover on the DEM Map. The software output is the positions and orientations of the multi-tag bundles with respect to the DEM Map reference frame.

This work can be further advanced by using object detection algorithms to precisely locate and classify POIs in an image. Defining multiple features for rock detection and classification can be computationally intensive. Thus, convolutional neural network-based object detection algorithms can provide a solution for detection in the context of space exploration. With more satellite images of the celestial object and increased computational capabilities, we can train an artificial neural network to classify the rocks of interest from the background in the photos.

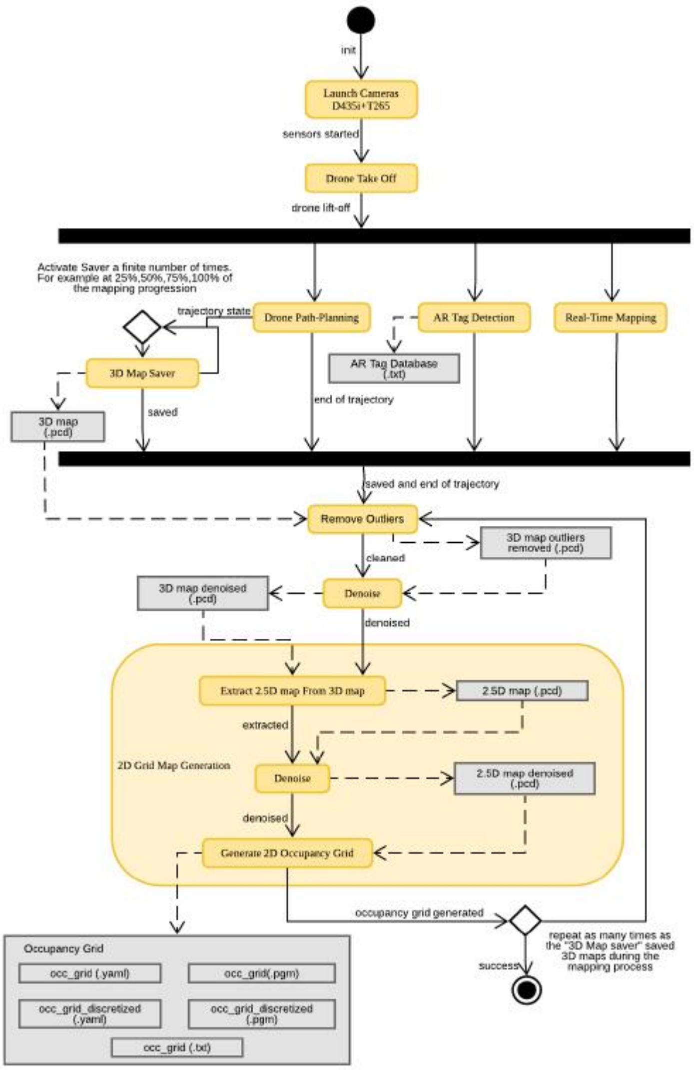

Figure 7: Software architecture for the mapping algorithm

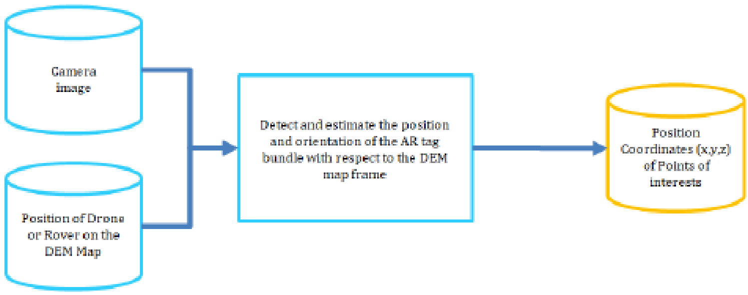

Figure 8: Point of Interest detection flow diagram

3.3 Task-Planning Algorithm

The task planning software will define the overall behavior of the two collaborative systems. It will determine: (i) when to execute the different tasks, (ii) the order of the different waypoints to visit during the mission, and (iii) the overall behaviors of the rover and the drone in the different situations. Planning problems in A.I. have been studied for a long time [13] [14][15] [16] [17]. In the specific case of our study, we will apply HDDL [18], an extension of the PDDL [19]. The HDDL allows defining clusters of actions to be executed by a robotic system, thereby specifying its expected behavior given a set of triggering situations and estimating its resources. The software has six main modules (Figure 9).

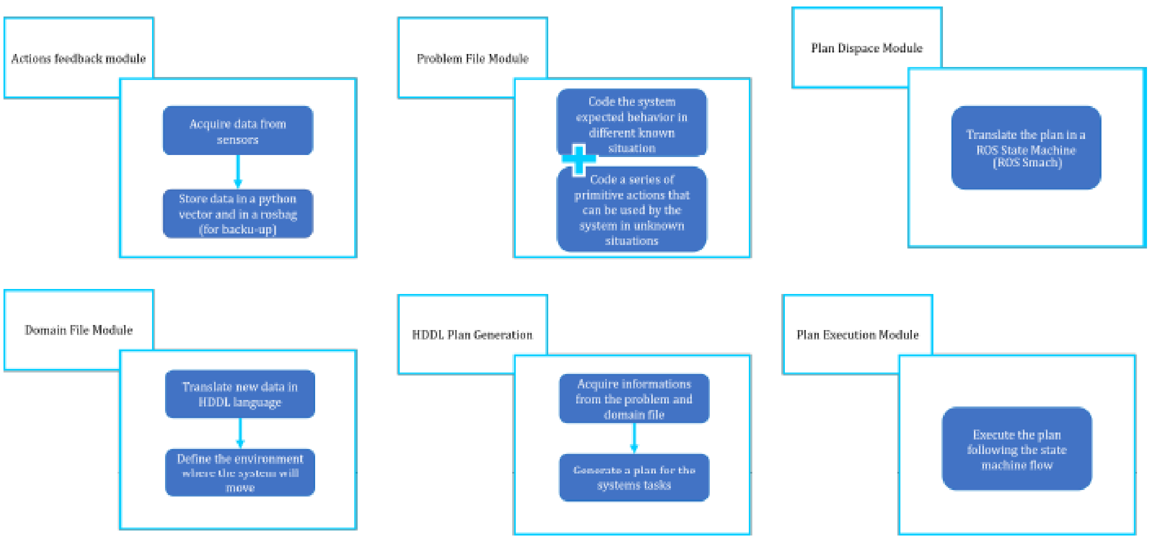

Figure 9: Modules of the task planner and high-level functions.

- Problem File Module, "problem.hddl" - This module groups all the available information on the environment where the system is moving and its goals.
- Domain File Module, "domain.hddl" - This module outlines all the possible actions that the system can execute. It considers even a cluster of potential actions that particular situations can trigger.
- HDDL Plan Generation Module - The module will generate the plan to be executed by the systems: (i) when to move, (ii) when to communicate, (iii) with whom to communicate, (iv) in which order to reach each of the waypoints, etc.

- Plan Dispatch Module - The module will parse the plan just generated and interface it with a State Machine.
- Plan Execution Module - The module will execute the plan generated by the Plan Dispatch Module using the Action Library of ROS.
- Actions Feedback Module - The module will store and manage the inputs from the sensors. It will supply the readings to the Problem File if new obstacles are detected, if a waypoint cannot be reached, etc, to recompute the overall plan.

The software is interfaced with the other teams as follows:

- The ConOps and design of the mission from Mission Analysis are coded in the domain file. The starting point for the code is the Enhanced Functional Flow block Diagram (EFFBD) for the drone and rover mission, shown in Fig. 10.
- The occupancy grid from Localization and RealTime Mapping is translated into the ”initial conditions” of the ”problem.hddl” file. We coded an interface that, given a grid map, outputs the required predicates for the HDDL problem file.
- The Action Library of ROS with the Move Base package configured for both rover and drone by the Path Planning team. The move base package provides go-to commands and can be easily interfaced with the localization module and the global and local planners.
- The SMACH Monitor State of ROS [20] to monitor the sensors topic and stop, change, or reconfigure the plan of the rover. If there is an anomaly on the drone, we are just going to stop it — hard failure — or make it go to the recharging station — manageable failure.
- The interface for the control center is coded as a ”visual” state machine that the control center operator can easily follow.

Figures 11 and 12 show the logical flow of the task-planning algorithm for the rover and the drone, respectively.

3.4 Path-Planning Algorithm

The Path Planning software plans and executes the trajectory for both the drone and the rover. Its main features and interfaces can be summarized as:

- It collaborates with the Task Planning software to determine the goal/destination of the robot.
- It elaborates the Localization and Real-Time Mapping software output data, grid map, and position, to compute the optimal trajectory for the system under study.
- It controls the actuators to move the system to its destination effectively.
- It listens as well to the odometry readings of the encoders, the LiDAR readings of the obstacles, and the visual odometry of depth sensors to localize the system in a global or partial map [21].
- It uses Python as the primary coding language. However, the localization modules will be computed in C++ for better computational velocity.
- The software is interfaced with the hardware through the ROS topics, services, and action libraries. In our case, we extensively use ROS Navigation Stack [21].
- Both systems can be stopped in case of contingency, as shown in Figure 18, by the control center. In the event of a hardware failure, field operators can intervene directly.

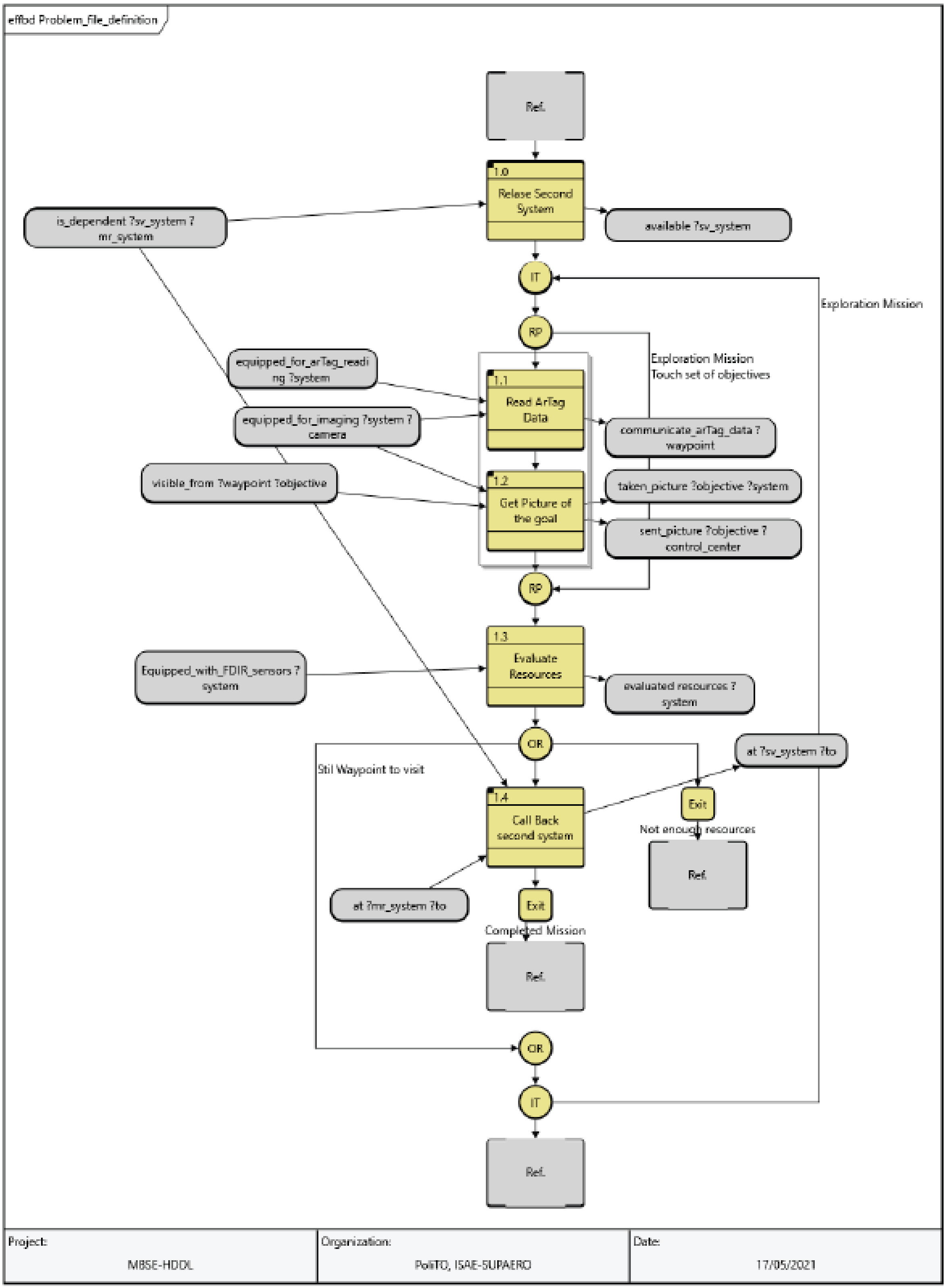

Figure 10: Mission Analysis EFFBD for the IGLUNA mission.

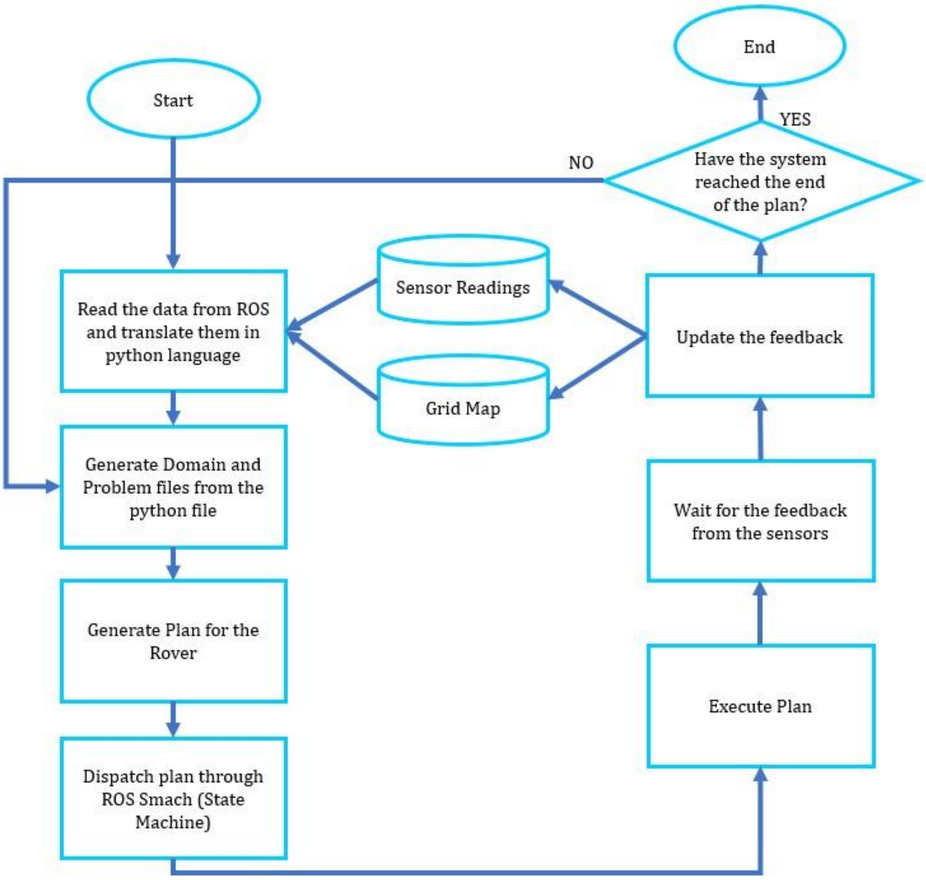

Figure 11: Task planner process flow for the rover.

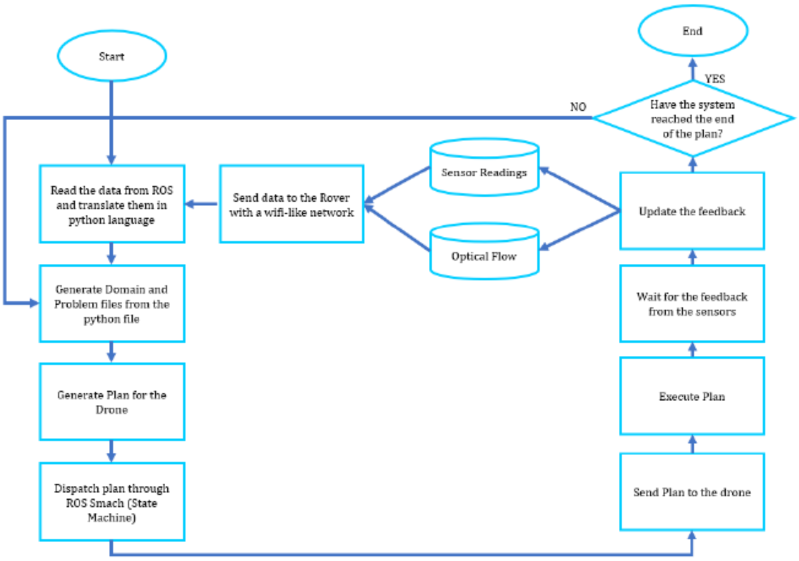

Figure 12: Task planner process flow for the drone.

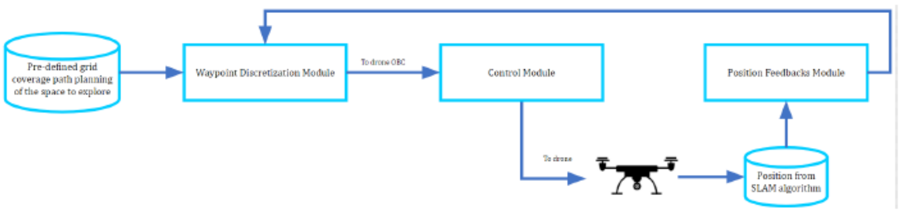

Figure 13: Drone mapping phase block diagram

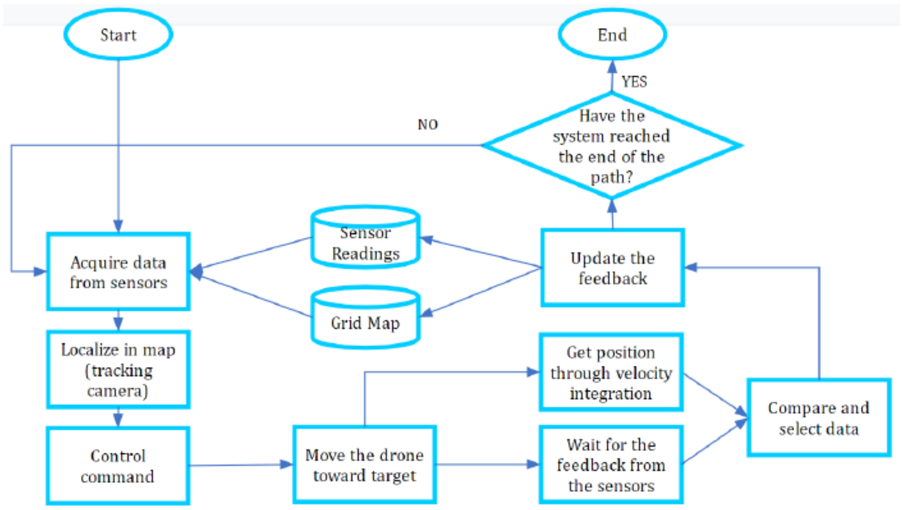

Figure 14: Drone mapping flowchart

Since the mission starts with the recognition phase, the drone's first challenge is to produce a map of an unknown environment. Thus, the goal of the Path Planning software for the drone mapping phase is to compute and control the optimal trajectory for the drone to map the maximum amount of terrain within a given maximum flight time (battery-dependent). The envisioned algorithm has four main modules, as shown in Figure 13:

- Path Planner Module: The module computes the optimal trajectory for the drone relative to the Localization and Real-Time Mapping requirements.
- Trajectory Discretization Module: The module discretizes a given trajectory into a sequence of waypoints that have to be followed by the drone.
- PositionFeedbackModule: With the help of the position given by the drone's Localization and Real-Time Mapping software, the Position Feedback Module can estimate the error between the theoretical trajectory and the real one. It rectifies the rover's movement to match the planned trajectory as closely as possible.
- Controller Module: The module interfaces the high-level goals of the path planners with the system actuators. This module consists of the PixHawk controller on the drone interfaced with ROS (Robot Operating System) through the "MavROS package". Therefore, the motion commands are sent to the PixHawk controller via the MavROS topics and services.

Figure 14 shows the logical flow of the drone's path-planning algorithm during the mapping phase.

The first step for the drone during its mapping phase is understanding the limits of its operational area. This statement applies to both the Earth analog and the lunar missions.

On the Moon, the mission of the flying hopper (sim-

Figure 14: Drone mapping flowchart

ulated by a drone in the IGLUNA analog mission) will explore portions of craters or lava tubes. Such structures present an easily recognizable border, which is the rim of the crater or the edges of the lava tube. The edges of such structures have high contrast with their local environment. Therefore, they can be detected and tracked by the cameras of rovers, hoppers, and lunar landers.

Going back to Earth, knowing the coordinates of the borders, the drone can estimate its operating area. At the FC, we will simulate a high-contrast border with a vividly colored rope (Figure 15).

The first task in the drone's mapping phase will be to identify its operating area. Since the terrain to be mapped has known dimensions, the drone does not need to scan the entire border to understand its operating area. Instead, it is sufficient to obtain data from four reference points lying on the border. Those will be the first landmarks that will provide the mapping limits, Figure 15.

The referencing points will be represented by four ARtags placed at the middle of each line of the square that defines the operating terrain. As the drone will be initially positioned at the center of the terrain, it will follow a straight path to each point and then return to the initial point, tracing a couple of orthogonal axes in both directions, as shown in Figure 16. To guarantee correct drone referencing, two extra AR tags will be placed 1 meter from each referencing point, anticipating small drifts that may occur during motion, mainly due to wind or localization errors. While the drone acquires data on the borders, the Localization and Real-Time Mapping subsystem will leverage this motion to initiate the mapping and localization processes simultaneously.

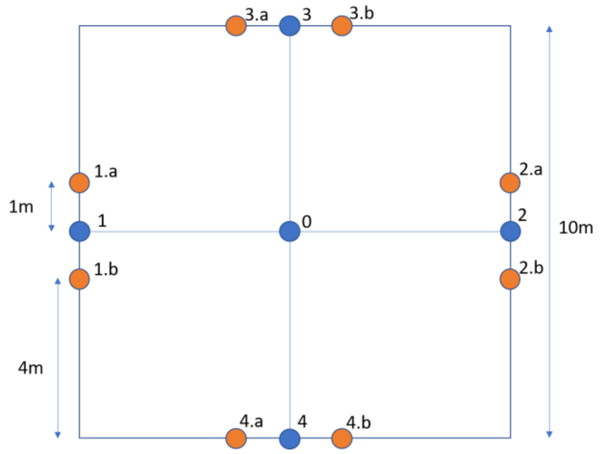

Figure 15: Drone referencing points

The drone's control module will be driven by velocity and time commands (MavROS"velocity setpoint" topic). The system localizes by integrating its velocity over time, using dead reckoning, and comparing it with the tracking camera's readings. Based on tests conducted for the Localization and Real-Time Mapping software, a velocity of 0.5 m/s or less was found to be the ideal flight velocity for the drone. An altitude of 3 meters can be maintained during the flight (as already defined in the mission analysis). The constant altitude allows simplifying the mapping phase. It is reasonable to assume we are performing 2D navigation with the drone; therefore, a 2D cost map will suffice as input to the drone navigation stack.

To map the remaining terrain, a grid-like pattern with several waypoints has been chosen (Figure 16).

Due to the square shape of the operating area, the chosen straight-line pattern ensures coverage of the remaining terrain without flying over areas already covered in the drone referencing phase, thereby optimizing battery use and, consequently, the mission duration. To determine the distance between the path loops, the camera's Field of View (FOV) had to be accounted for.

The T265 tracking camera and the D435i camera were considered, with the D435i's conservative minimum FOV of 42.5 degrees taken as the design condition for path planning. At an altitude of 3 m, the drone's field of view on the ground is 1.16 m, with a diameter of 2.3 m. It is for this reason that the path for mapping the operating area comprises loops at 2 m intervals (a buffer of 0.3 m to be conservative).

The rover and drone mission phases follow the drone's mapping phase. Thus, in nominal conditions, the rover and the drone are provided with a global 2D

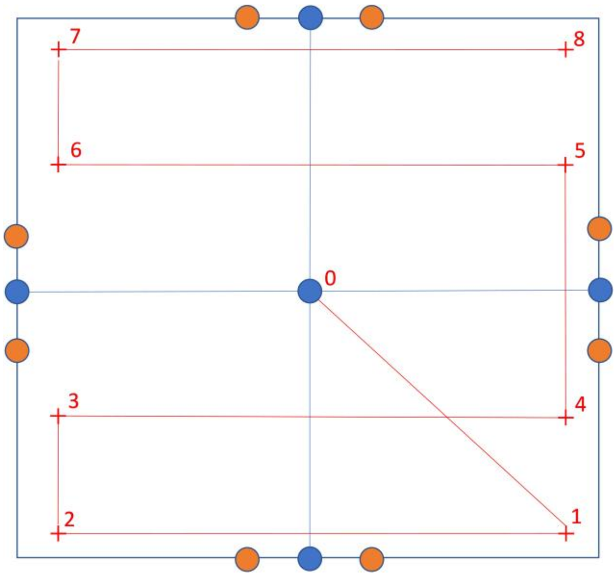

Figure 16: Drone pattern for mapping

Cost-map of the environment. Therefore, the main goal of the Path Planning software for the mission phase is to compute the optimal trajectory and physically move the robots in a known environment.

Figures 17, 18, 19, and 20 show the logical flow of the Path Planning algorithm applied to the drone and rover, respectively, during the mission phase.

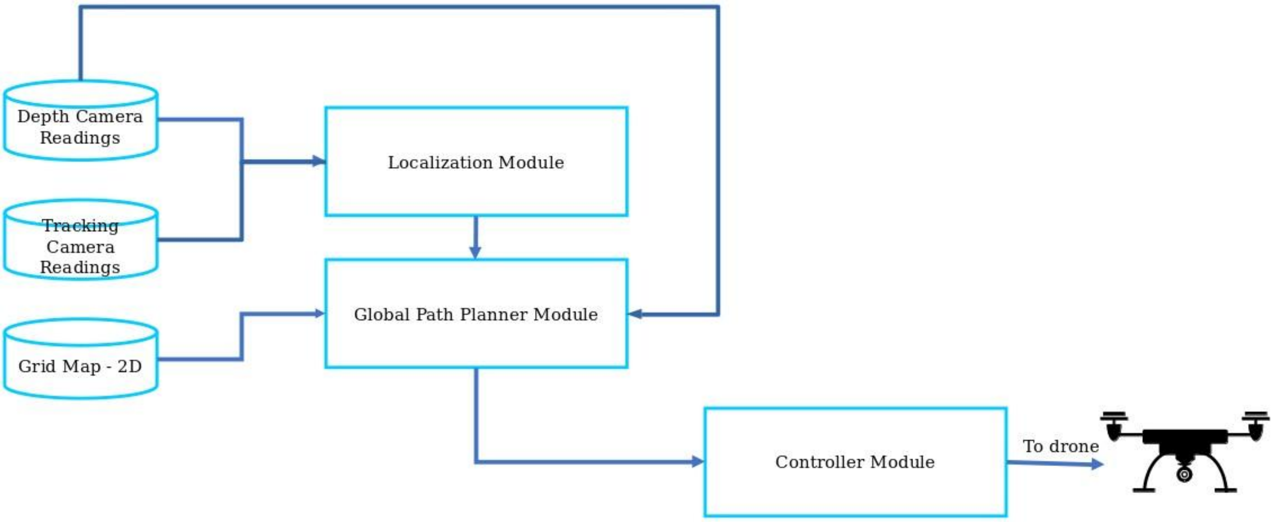

Figure 17: Drone mission phase block diagram

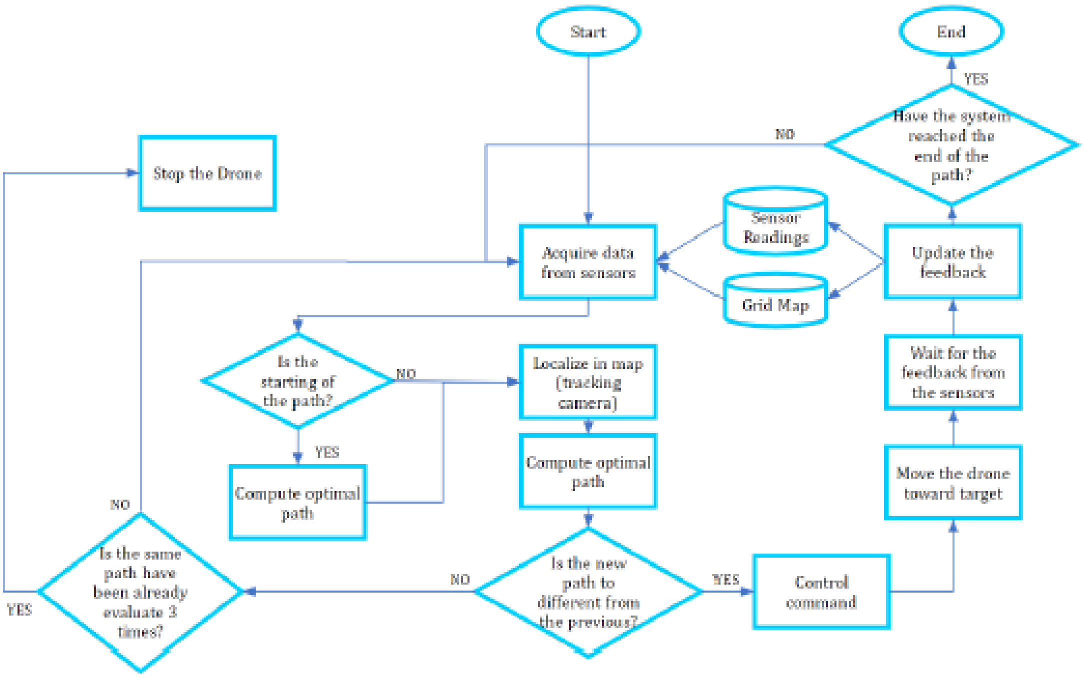

Figure 18: Drone mission flowchart

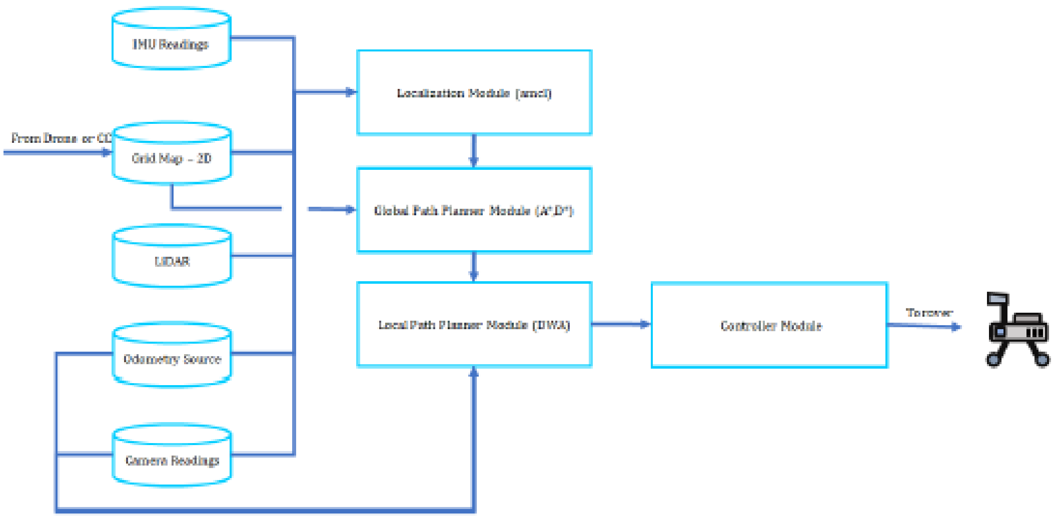

Figure 19: Rover mission phase block diagram

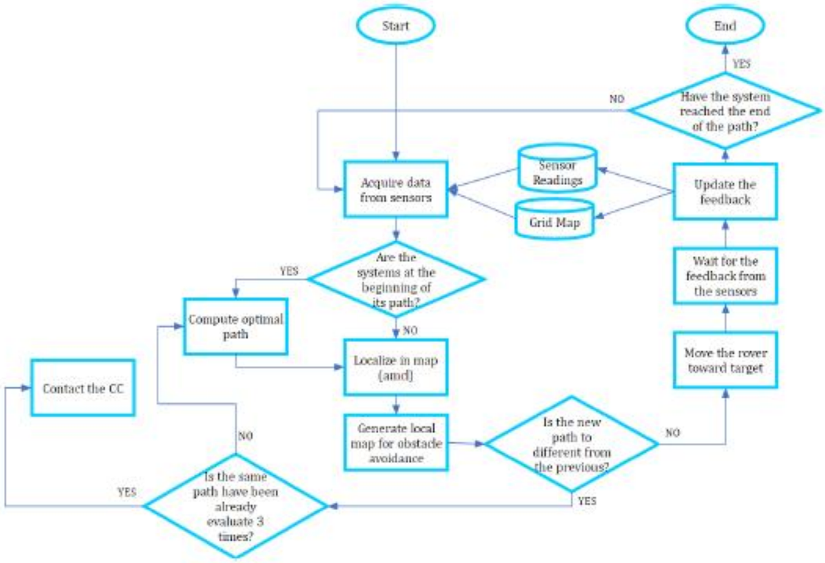

Figure 20: Rover mission flowchart

To implement the global and local path planner, we have used the" ROS Move_base" package. Figure 21 shows the visualization of:

- Global Map (with the first layer of inflation of the obstacles),
- Local Map (with the second layer of inflation of the obstacles),
- Global Path (where the rover footprint is shown in green).

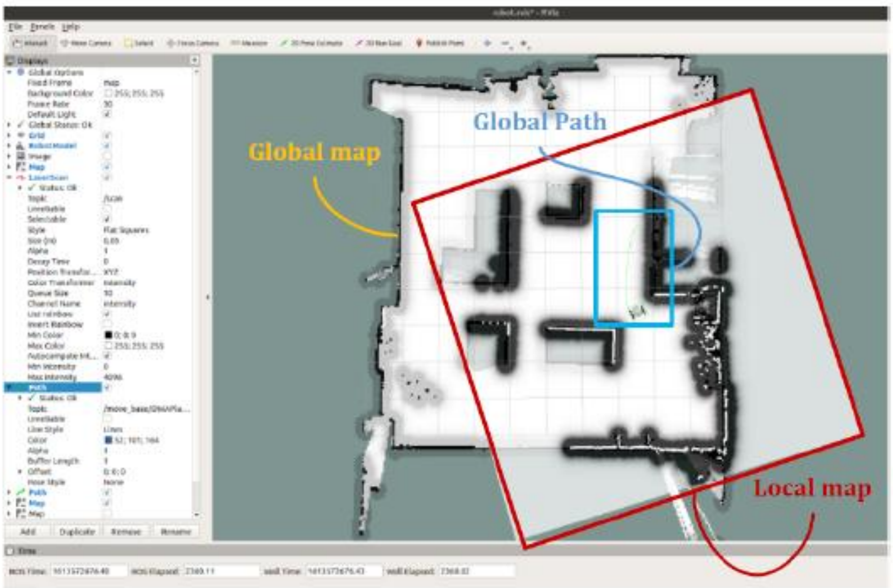

Figure 21: Path Planning set-up for the rover

4. Autonomous Navigation Main Results

The Field Campaign mission involving the drone and the rover collaborating to explore an unknown environment happened according to the mission description described in Figure 18: Drone mission flowchart in the Case Study section 2. The set-up of the testing field is shown in Figure 2 and consisted of an unknown area with randomly placed POIs, simulating objectives and obstacles simultaneously. The 3D and 2D maps generated by the drone during its mapping phase are shown in Figures 23 and 22.

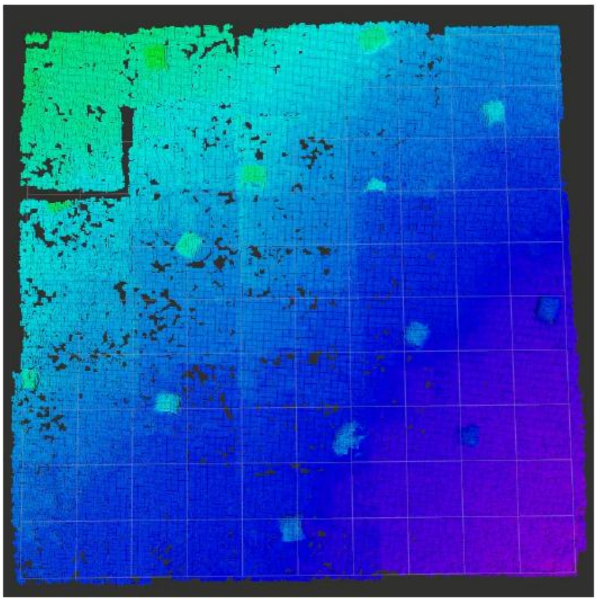

Figure 22: 3D DEM map generated by the drone. Resolution 3cm

Figure 23: 2D Occupancy grid map generated by the drone. Resolution 8cm

Due to the nature of our experiments, the actions of the drone (flying pattern) and the exploration phase of the rover can't be shown in this document. However, you can see how our robotic systems behaved during the Field Campaign in our video.[3]. You will see how our drone flew autonomously over the environment to map it and return to its original position with precision. You will see our rover using the map to visit points of interest while avoiding obstacles. And finally, you will see our drone autonomously visiting the remaining points of interest, as per the orders received from the global task planner embedded in the rover.

5. Conclusions

All our work was performed between October 2020 and June 2021, with our final outside Field Campaign happening in late June 2021. Thus, the CoRoDro project is in its first year of development, and we will discuss

our results and our plans for next year. Our in-house laboratory tests showed a maximum percentage difference of 2% between the optitrack and T265 data for the drone's height position. That difference is primarily due to the position of the nearest optitrack beacon to the camera. However, we are primarily interested in the system's coordinates (x, y) during its flight, and in that case, the difference drops to less than 1%. The rover mounts the same bundle of cameras, a D435i and T265. In this case, the movements are less sudden; therefore, the (x,y) positions are exact.

During the Field Campaign, 14 AR tag bundles were used as points of interest and placed at random locations across the field. The AR tag detection algorithm embedded on the drone and rover detected all AR tags. It published the AR Tag bundle identification number, position, and orientation information to detect unique points of interest in the field. The information of AR tags was further used by the ARTag linker algorithm and task planner algorithm to localize the robotic systems on the map and allow the drone and rover to move towards the points of interest. The plan

[3]CoRoDRO Virtual Field Campaign Show : https://www.youtube.com/watch?v=rSvPARSduhQ

For the upcoming year, the goal is to use AI technology, specifically deep learning methods, to detect points of interest —such as bright objects and rocks —and determine their positions and orientations on the field.

Concerning the rover, we had problems with the local path planner. We didn't mow the grass; therefore, the LiDAR, the primary sensor used for obstacle avoidance, was"sensing" obstacles that weren't there. Because the terrain was approximately flat, we were able to accomplish the mission using only the global path planner. However, next year we will improve the overall design of the local path planner, aiming to utilize camera inputs better as well. Therefore, the systems will be able to navigate more challenging terrains. One of the main focuses of next year's IGLUNA will be the coding of the in-house path planner for both the drone and the rover.

References

[1] *Artemis mission 2024*, https://www.nasa.gov/specials/artemis/.

[2] *NASA news for the moon in 2028*, https://www.nasa.gov/ feature/https://www.nasa.gov/feature/nasa-seeks-us-partners-to-develop-reusable-systems-to-landastronauts-on-moon/.

[3] *Institutions' plans for the moon*, https://www.esa.int/About_Us/Corporate_news/ESA_advances_its_plan_for_satellites_around_the_Moon.

[4] *In situ attractiveness of the moon:* http://interactive. satellitetoday.com/the-lunar-economy-from-vision-to-reality/.

[5] *Lunar lava tube*, https://en.wikipedia.org/wiki/Lunar_lava_tube.

[6] *Octomap server*, http://wiki.ros.org/octomap_server.

[7] *Voxel grid filter*, https://pcl.readthedocs.io/projects/tutorials/en/latest/voxel_grid.html.

[8] *Voxel definition*, https://en.wikipedia.org/wiki/Voxel.

[9] *Radius outlier removal filter*: https://pointclouds. org/documentation/tutorials/remove_outliers.html#remove-outliers.

[10] *Statistical outlier removal filter*, https://pointclouds.org/ documentation/tutorials/statistical_outlier.html#statistical-outlierremoval.

[11] *Occupancy grid format*, http://docs.ros.org/en/diamondback/api/ nav_msgs/html/msg/OccupancyGrid.html.

[12] *Ros ar-track-alvar package*, http://wiki.ros.org/ar_track_alvar.

[13] S. Chien, G. Rabideau, R. Knight, R. Sherwood, *et al.*, "Aspen-automated planning and scheduling for space mission operations," in *Space Ops*, vol. 82, 2000.

[14] N. Muscettola, P. P. Nayak, B. Pell, and B. C. Williams, "Remote agent: To boldly go where no AI system has gone before," Artificial Intelligence, vol. 103, no. 1-2, pp. 5–47, 1998.

[15] A. J. Coles, A. I. Coles, M. M. Munoz, O. E. Savas, *et al.*, "On-board planning for robotic space missions using temporal PDDL," in *11th International Workshop on Planning and Scheduling for Space (IWPSS)*, 2019.

[16] W. Chi, S. Chien, and J. Agrawal, "Scheduling with complex consumptive resources for a planetary rover," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 30, 2020, pp. 348–356.

[17] Y. Gao, G. Burroughes, J. Ocón, S. Fratini, *et al.*, "Mission operations and autonomy," *Contemporary Planetary Robotics: An Approach Toward Autonomous Systems*, pp. 321–401, 2016.

[18] D. Höller, G. Behnke, P. Bercher, S. Biundo, *et al.*, "HDDL: An Extension to PDDL for Expressing Hierarchical Planning Problems," in *AAAI Conference on Artificial Intelligence (AAAI)*, NYC, NY, USA, 2020. DOI: 10.1609/aaai.v34i06.6542.

[19] M. Fox and D. Long, "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains," *Journal of Artificial Intelligence Research (JAIR)*, vol. 20, 2003. DOI: 10.1613/jair.1129.

[20] P. Z. Schulte, "A state machine architecture for aerospace vehicle fault protection," *PhD Dissertation, Georgia Institute of Technology*, 2018.

[21] *Ros navigation*, http://wiki.ros.org/ navigation, Accessed: 2020-10-21.