IGLUNA 2021

A test bed to demonstrate space technologies in extreme environment

Student Documentation (SD)

Team Name: CORODRO

Revision: Rev. 05

Date of Issue: 31/07/2021

Document Name: IGL_P07_CoRoDro_SD_Rev_05.pdf

Change history

In this section, we listed our revisions and adds to the Student Documentation. All the applicable RIDs' answers are highlighted in yellow.

Revisi	Date	Section	Modifications / Remarks
01	01/12/2020	Detailed Documentation - Section 2.1	Added constraints requirements of drone and rover in table 11.
02	05/01/2021	Detailed Documentation - Section 2.15, SLAM Algorithm	Added paragraphs 2.15.1.3 and 2.15.1.4; added news on conversion module in paragraph 2.15.1.5.
03	30/01/2021	Detailed Documentation - Section 2.15	Added figures 23 about the Block diagram of the Conversion Module; 24 and 22 about examples of post-processing results.
04	01/02/2021	Detailed Documentation - Section 2.11.5	Added the new version of the data budget and last information about the data management.
05	01/02/2021	Detailed Documentation - Section 2.16	Subsection restructuring. Added new subsubsections 2.16.1 and 2.16.2. Added new paragraphs 2.16.1.1 and 2.16.1.2. Section 2.15, Added subsubsection 2.15.4 and 2.15.3.
06	01/02/2021	Core Documentation, Risk Analysis - Section 1.9	Added clarification on the mitigation actions for the IGLUNA 2021 campaign, Table 6.
07	03/02/2021	Detailed Documentation - Section 2.11.5	Added missing information in the data budget Tables 40, 41.
08	05/02/2021	Detailed Documentation - Section 9.3	Updated gantt Figure 6, work-breakdown structure Figure 7, allocated time per member Figure 14 and Table 17. Explained changes in the team.
09	05/02/2021	Detailed Documentation - Section 9.4	Updated the overall section with the new work-breakdown structure.
10	05/02/2021	Detailed Documentation - Section 9.5	Added information about flying a drone in the Lucerne Canton.
11	05/02/2021	Core Documentation - Section 1.10	Added two more points on the localization performances that we would like to check for the SLAM algorithm.

12	05/02/2021	Detailed Documentation - Section 2.6	Added information about terrain setup.
13	06/02/2021	Detailed Documentation - Section 2.15.5	Added information on AR-tags and figures 30 and 31.
14	08/02/2021	Detailed Documentation - Section 2.5.2	Added paragraph 2.5.2.2 Expense and 2.5.2.1 Income. Added Table Expense.
15	08/02/2021	Detailed Documentation - Section 2.21	Updated information and Table 62 Equipment needed.
16	08/02/2021	Core Documentation - Section 1.5	Added last paragraph explaining the link with the project and the Moon Mission.
17	08/02/2021	Core Documentation - Section 1.9	Added COVID-2 risk to Table 6.
18	08/02/2021	Detailed Documentation - Section 2.1	Added functional requirements RF-9 to RF-15 (Table 7, added performance requirements RP-4 and RP-5 (Table 8, added RS-4 to Table 10, added constraints requirements RC-7 and RC-8 to answer the RIDs (in yellow) to Table 11, added constraints requirements RC-9 and RC-10 to Table 11, added communication requirements 12.
19	08/02/2021	Detailed Documentation - Section 2.6	Added Figure 15 and explanation on the terrain set-up during the FC.
20	08/02/2021	Detailed Documentation - Section 2.7	Added Tables with the full ConOPS for the FC, from Table 23 to Table 29
21	08/02/2021	Detailed Documentation - Section 2.9	Introduces overall interfaces requirements during FC as bullet points.
22	08/02/2021	Detailed Documentation - Section 2.11	Reshape of the overall section and added tables 37, 38, 39, 40, 41, 42, 43.
23	08/02/2021	Detailed Documentation - Section 2.12	Added explaination of the data interface choice for the FC.
24	08/02/2021	Detailed Documentation - Section 2.17	Added detailed on the overall algorithm, added Figure 47.
25	08/02/2021	Detailed Documentation - Table 52	Added link between requirements and software/performed test.

26	08/02/2021	Detailed Documentation - Section 2.20.2	Added AR-tags related tests to Table 20.
27	08/02/2021	Detailed Documentation - Section 2.22	Added new internet site link for the CoRoDro team, summed-up outreach work performed up till now on our social media.
28	09/02/2021	Detailed Documentation - Section 2.20.3	Added SLAM Tests results.
29	01/03/2021	Detailed Documentation - Section 2.11	Verification and change of the values of the Tables 37, 38, 39, 40, 41, 42, 43.
30	08/03/2021	Detailed Documentation - Section 5.2	Answer of the CDR's RIDs for mission analysis.
31	15/03/2021	Detailed Documentation - Section 2.12	The map is now created and downsized on the drone. Therefore, the overall section was re-written.
32	15/03/2021	Detailed Documentation - Section 2.16.2	Added description and references for the path planning algorithms A*, D* and DWA.
33	18/03/2021	Detailed Documentation - Section 2.7.2	Added two tables to show the power budget and data budget needed per ConOps (testing).
34	24/05/2021	Detailed Documentation - Section 2.11.5	Added a paragraph explaining the procedure in case of bandwidth issue between the test bed and the control center.
35	25/05/2021	Detailed Documentation - Section 2.20.3	Added occupancy grid map results as final output of the drone mapping task.
36	25/05/2021	Detailed Documentation - Section 2.20.3	Added a test related to the AR_tag detection and their use to link the drone and the rover's coordinate frames.
37	25/05/2021	Detailed Documentation - Section 2.20.2	Updated the Tests Plan on the 25/05/2021.
38	25/05/2021	Detailed Documentation - Section 2.15.4	Added algorithm description for linking the coordinate frames of the rover and the drone.

39	25/05/2021	Detailed Documentation - Section 2.15.1.6	Added a new paragraph about the drone mapping phase algorithms and its automation.
40	25/05/2021	Detailed Documentation - Section 2.15.1.1	Added a new paragraph about the conclusions of the SLAM tests results and their implications on the final configuration of the drone.
40	25/05/2021	Detailed Documentation - Section 2.15.5	Added latest information on the Points of interest detection and progress on AR tag detection algorithm. Added a new figure 31 of the point of interest.
41	27/05/2021	Core Documentation - Section 1.9	Update table 6 and added section 1.9.1.
42	27/05/2021	Detailed Documentation - Section 2.3	Updated the work packages overview with the development plan for the month of June in Figure 9.
43	27/05/2021	Detailed Documentation - Section 2.5.1	Updated Figure14 and Table17 with the workload of the team up to the RR.
44	27/05/2021	Detailed Documentation - Section 2.5.2	Updated budget for the VFC, Table 19.
45	27/05/2021	Detailed Documentation - Section 2.6	Added picture of new testing terrain Figure 11.
46	27/05/2021	Detailed Documentation - Section 2.6	Updated operations and testing procedures in Section 2.7.2 (from Table 20 to Table 29.
47	27/05/2021	Detailed Documentation - Section 2.6	Added subsection 2.8 and timeline for the VFC 17.
48	27/05/2021	Detailed Documentation - Section 2.4.	Added subsection on VFC.
50	27/05/2021	Detailed Documentation - Section 2.16.	Update information on the localization of the rover in Paragraph 2.16.2.3.
51	27/05/2021	Detailed Documentation - Section 2.17.	Update information on the interfaces between task planning and Mission Analysis, Path planning and SLAM. Added subsections 2.18 and 2.19.
52	27/05/2021	Detailed Documentation - Section 2.20.	Updated requirements verification matrix, testing plan and testing results and testing checklist.
53	27/05/2021	Team - Section 6.3.	Updated team members list.
54	27/05/2021	Annex - Section 10.	Added Annex 10 for problem and domain file of the IGLUNA Task Planner.
55	31/05/2021	Annex - Section 4.	Added Section 4 with main test results and lesson learned.

Contents

1 CORE DOCUMENTATION **10**
1.1 Abstract . . . 10
1.1.1 Goal of the Project . . . 10
1.1.2 Our Solution . . . 10
1.2 Graphical Abstract (Project Concept) . . . 11
1.3 Project Statement . . . 11
1.4 Need Statement . . . 11
1.5 Project Relevance . . . 11
1.6 Stakeholder Identification . . . 12
1.7 Project Goals and Objectives . . . 15
1.8 Project Top-Level Requirements . . . 18
1.9 Risk Analysis . . . 18
1.9.1 Risk Analysis matrix . . . 19
1.10 IGLUNA Field Campaign Results . . . 20
1.11 Business Model . . . 21
1.11.1 Executive Summary . . . 21
1.11.2 Canvas Model . . . 22

2 DETAILED PROJECT DESCRIPTION **24**
2.1 Complete Project Requirements . . . 24
2.2 Trade-offs . . . 28
2.2.1 Benchmarks . . . 28
2.2.2 Trade-off 1 . . . 28
2.2.3 Trade-off 2 . . . 29
2.3 Project Management . . . 30
2.3.1 Work Packages Overview . . . 30
2.4 Virtual Field Campaign . . . 32
2.4.1 Testing for the VFC . . . 32
2.4.2 VFC videos . . . 33
2.5 Resources . . . 33
2.5.1 Task-force . . . 34
2.5.2 Budget . . . 35
2.5.2.1 Income . . . 35
2.5.2.2 Expense . . . 35
2.6 Project Set-Up . . . 37
2.7 Concept of Operations . . . 38
2.7.1 Safety Measures . . . 39
2.7.2 Operations and Test Procedures . . . 39
2.8 Virtual Field Campaign Timeline . . . 49
2.9 Project Interfaces . . . 50
2.9.1 Mechanical, Power and Data Interfaces . . . 50
2.10 Project Overview Summary . . . 50
2.10.1 List of Components . . . 50
2.10.2 Project Overview Summary . . . 51
2.11 Project Design . . . 52
2.11.1 Mechanical Design . . . 52

2.11.2 Mass Budget 53
2.11.3 Electronics Design 54
2.11.4 Power Budget 54
2.11.5 Data Budget 55
2.12 Data Management 57
2.12.1 Data Management Concept 57
2.13 Data to monitor 57
2.13.1 Path Planning 58
2.13.2 Localization and Real-Time Mapping 59
2.13.3 Task Planning 59
2.14 Maximum transmission possible 59
2.15 Localization and Real-Time Mapping 61
2.15.1 Localization and Real-Time Mapping Algorithm for the Drone 62
2.15.1.1 Drone different evaluated configurations and Final chosen Configuration 62
2.15.1.2 Algorithm Architecture 62
2.15.1.3 Data Merging Module and Matching Module 65
2.15.1.4 Depth Filtering Module 65
2.15.1.5 Conversion Module: Transforming DEM Map into Cost-Map 66
2.15.1.6 Drone Mapping phase automation 67
2.15.2 Localization and Real-Time Mapping algorithm for the rover 70
2.15.3 Collaborative Localization and Real-Time Mapping between the Rover and the Drone 73
2.15.4 Merging Rover's and Drone's Frames and Localizing the Rover with the Drone 73
2.15.5 Points of Interest 74
2.16 Path Planning 76
2.16.1 Path Planning for the Drone Mapping Phase 76
2.16.1.1 Delimitation of the Operating Area 77
2.16.1.2 Mapping of the Operating Area 78
2.16.2 Path Planning for the Drone and Rover Mission Phase 79
2.16.2.1 Training with Morse 84
2.16.2.2 Mapping 84
2.16.2.3 Localization Issues - Solved 85
2.16.2.4 Local and Global Path Planner 86
2.17 Task Planning 87
2.18 Plan Definition 89
2.18.0.1 Rover and Drone Mission 89
2.18.0.2 Drone Mapping 94
2.19 State Machine 96
2.19.0.1 Interface of the ROS packages 97
2.20 Project Verification and Test Planning 98
2.20.1 Verification Matrix of the Requirements 98
2.20.2 Test Plans 100
2.20.3 Test Results 113
2.20.3.1 Results Test UT-2 113
2.20.3.2 Results Test SM-1.D and Test SM-1.L 113
2.20.3.3 Results Test SM-2.D and Test SM-2.L 115

2.20.3.4 Results Test SM-3.D . . . 120
2.20.3.5 Results Test SM-3.L . . . 122
2.20.3.6 Tests conducted for AR Tag recognition . . . 123
2.20.3.7 Test Merging Frames Rover-Drone . . . 124
2.20.3.8 Tests New rover localization, LOC-1 . . . 125
2.20.3.9 Rover State Machine, DM-1 . . . 125
2.20.3.10 Drone Mapping State Machine, TPD-1 . . . 125
2.20.4 Testing and Operations Checklists . . . 126
2.21 Project Equipment Needed . . . 132
2.22 Outreach . . . 132
2.23 Further Business Model Strategies . . . 132
2.23.1 SWOT Analysis . . . 133
2.23.2 Sponsoring and Strategic Partnership . . . 133

3 VFC Results and Lesson Learned **135**

4 Virtual Field Campaign Testing Results **135**
4.1 SLAM . . . 135
4.1.1 Drone Mapping Phase . . . 135
4.1.2 Points of Interest detection . . . 136
4.2 Path Planner . . . 137
4.2.1 Path Planner Generalities . . . 137
4.2.2 Virtual Campaign Testing . . . 137
4.3 Task Planner . . . 137
4.4 Lesson Learned . . . 146

5 SD REVISIONS **148**
5.1 Preliminary Design Review [PDR] . . . 148
5.1.1 Actions . . . 148
5.1.2 Advice . . . 154
5.2 Critical Design Review [CDR] . . . 163
5.2.1 Actions . . . 163
5.3 Readiness Review [RR] . . . 171

6 ABOUT THE TEAM **187**
6.1 Point of contact . . . 187
6.2 Team Structure Overview . . . 187
6.3 Team Members . . . 188

7 LIST OF ABBREVIATIONS **193**

8 REFERENCES **194**

9 Annexes **197**
9.1 Lessons Learned . . . 197
9.2 Lunar Mission Analysis . . . 197
9.3 Project Management . . . 198
9.3.1 Business Canvas . . . 198
9.3.1.1 Key partners . . . 198

9.3.1.2 Key Activities 198
9.3.1.3 Key Resources 198
9.3.1.4 Cost Structure 200
9.3.1.5 Revenue Streams 200
9.4 Work Packages Breakdown 201
9.5 FOCA Rules 207
9.5.1 Emails from FOCA 207
9.5.2 For the Lucerne Canton and Flying Drones 208
9.5.3 Drones Rules up to 11/02/2021 209

10 Task Planning Domain and Problem Files 210

1 CORE DOCUMENTATION

1.1 Abstract

The CORODRO (COllaborative ROver and DROne) team from ISAE-SUPAERO is a student team dedicated to the study of autonomous navigation and operations for space robotics systems. As space missions turn bolder and push toward deep space, the concept of autonomy becomes an essential asset to space exploration. Our study aims to mature the algorithms and the software for autonomous navigation and operations of robotics systems. More in detail, we want to study and advance technologies linked to the exploration of the lunar lava tubes. Lava tubes are volcanic architectures that unravel under the Moon surface. They may become a radiation and micro-meteoroids shielded human outpost. To assess the safety of these caves, a series of robotic missions are envisioned. They should autonomously map, navigate and operate inside the tubes as well as relay sensitive data toward Earth.

1.1.1 Goal of the Project

To accomplish these tasks, there is the need to test the algorithms and the strategies that will permit to reach the level of autonomy required to explore the lava tubes. Our primary aims are, therefore, to (i) demonstrate the feasibility of autonomous navigation in an unknown field, (ii) demonstrate that effectively collaborative robotic systems can increase the mission return, (iii) optimize the planning of tasks during remote operations. During the lava tube exploration, the rover will deliver a series of hopping and thruster-propelled bots toward the lava tube skylight, Annex 9.2. To simulate a similar set-up and effectively study the operational layer of our collaborative systems, we envision the use of a drone and a rover during the IGLUNA Virtual Field Campaign (VFC).

1.1.2 Our Solution

We will create a simulation environment in ROS (Robot Operating System) to visualize and control our rover and drone. We will upload to our systems' on-board computers our algorithms for autonomous navigation and task planning. The rover and the drone are provided by our partners, however, the bundle of sensors they embark will be tailored for the IGLUNA field campaign. Our systems can sustain the temperatures and the environmental constraints of the hot summers of south of France, where the IGLUNA's field campaign will be held. They would have to plan and act in a harsh environment where snow, wind and temperatures can be a hazard. Those conditions will be an excellent testbed for autonomous decision-making capabilities for both rover and drone. We will analyze their interactions to understand how their joined effort will be able to increase the mission return. We will preliminary test our systems in the ISAE-SUPAERO laboratories and with our partners, to be ready for the field campaign. At the end of this academic year, we look forward to sharing our data with the scientific community to actively contribute to the human exploration of the Moon in the near future. Our vision ahead of the IGLUNA campaign is to push forward the operational autonomy of robotic systems to be able to explore our universe while helping humans during critical tasks or in dangerous situations. All the team is looking forward to contributing to the success of the IGLUNA analogue mission, following the motto of ISAE-SUPAERO: *"Excellence with passion"*.

1.2 Graphical Abstract (Project Concept)

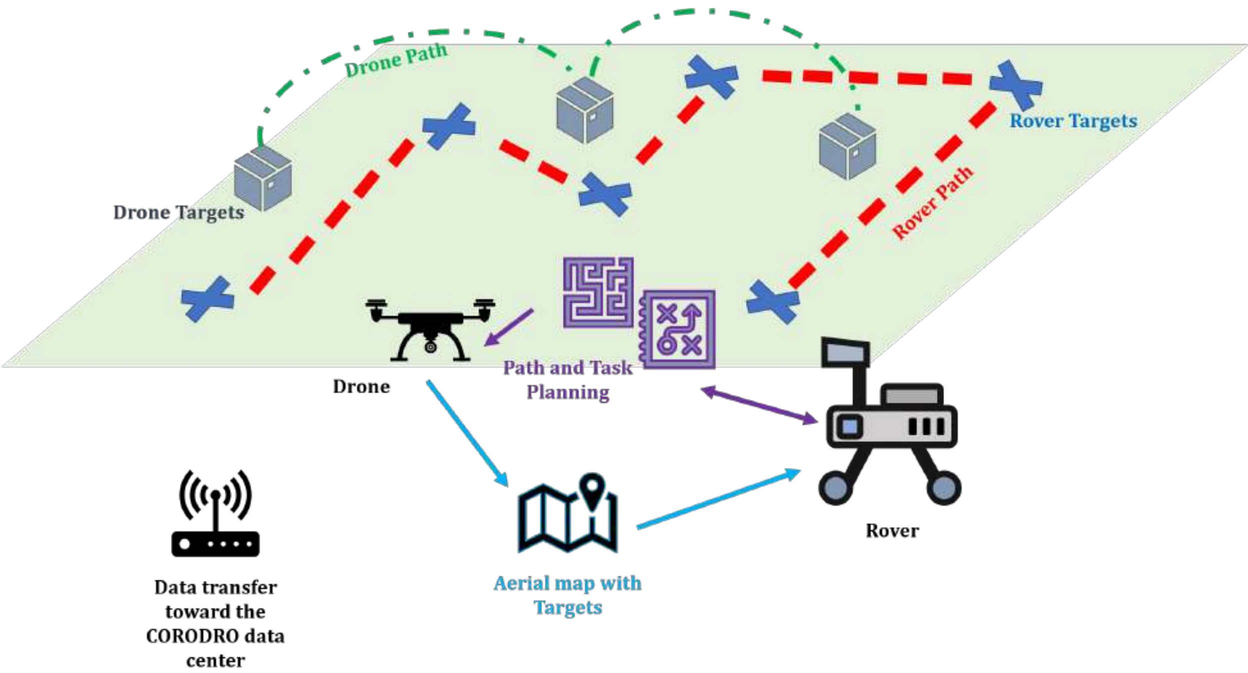


Figure 1: CORODRO project concept. A detailed description is given in Section 2.6

1.3 Project Statement

The CORODRO team will provide an autonomous navigation framework suitable for any mission that envisions collaborative robotic systems exploring an unknown space environment.

1.4 Need Statement

During Earth-orbiting missions, periodic high-quality communications with ground stations are guaranteed for both scientific and house-keeping purposes. This fails to be true for deep-space or underground planetary missions, where the quality of the communications links can be extremely poor. Nevertheless, missions are growing more complex and the capability of planning, controlling and predicting all potential mission events decreases. Autonomous decision-making capabilities onboard the future exploration system are, therefore, considered as critical technologies.

1.5 Project Relevance

Many space agencies, including NASA and ESA foresee that mankind will be back in the Moon in the 2020s [33]. However, strong engineering actions are required to sustain the realization of a permanent human outpost. In particular, one of the most critical aspects of both robotic and human space exploration is related to operations. As space exploration missions grow in complexity, there is a need to balance the mission return, the autonomy level and the workload of the control center operators. That aim can be reached thanks to the technological advancement of the last decade related to artificial intelligence [28]. In this context, space agencies, companies, and universities are engaged in the definition of a broad spectrum of technological maturation studies toward autonomous operations and navigation [31] [28] [27] [24]. As any innovation of the space sector, there is a need to engage in extensive testing to prove the suitability of the algorithm for collaborative robotics, autonomous navigation and autonomous operations. Our project starts in this context of technological maturation of autonomous navigation and operations for robotics systems. Our test-bed will be Earth: we are going to apply our autonomy algorithm to raise a capital that will sustain the deployment of our ideas on the Moon. Our topic of interest can easily double in successful terrestrial applications. It is easier and safer for robotics systems, such as rovers and drones to access the heart of a volcano or a radioactive area than for a human [38] [37]. We can deploy our collaboration algorithms in the agricultural sectors [39] [46]. We can use multiple collaborative and autonomous drones for delivery purposes [22]. Our tested algorithms of autonomous navigation and operation will permit our systems to explore challenging targets on the Moon, like the lava tubes. Lava tubes are volcanic structures that unravel under the lunar surface [34]. During their time inside the tunnels, the robotic exploration systems, may they be rovers or hopping bots, should rely on themselves both to navigate in the unknown environment and to operate. The DRM of the envisioned lunar mission can be found in Annex 9.2. During the field campaign we will test a rover collaborating with an ”Earth” drone. The first objective of this collaborative drone is to simulate the operations of the lunar bot. During the lunar mission the small propelled bot will fly to map and explore the environment before the rover. The same bot will use propelled flight to slow down its fall in the lava tubes while mapping the skylights. In the lava tube, the bot may decide to perform short flights to better map the environment. The second focus of our study is to test the collaboration between diverse robotic platforms and how this difference can really help shaping the success of an exploration mission. We firmly believe that to advance in the exploration spectrum of the Moon and Mars, we should start designing new complementary robotic systems.

1.6 Stakeholder Identification

Our main stakeholders for the IGLUNA Field campaign have been divided into three main categories, Figure 2 :

- University partners - Green
- Industrial partners - Blue
- Supervisors and Advisors - Purple

Our university partners are mainly helping us with expertise, access to hardware and access to facilities. Our industrial partners are providing us with free use of software and access to testing facilities, as well as expertise. ESA and Space Innovation are providing us with rules to comply to for the FC, as well as advice and guidance throughout the project. The FOCA was giving constrains relative to the regulation for flying drones in Switzerland. Therefore all the drone designed followed those guidelines, even if the final testing was conducted in France. To estimate the impact of each stakeholder on the project, we took three criteria into account following the guidelines in [7]:

- Power: the ability of a stakeholder to influence the outcome of the project (H: High and L: Low)
- Interest: interest of the stakeholder in the project, need (H: High and L: Low)

- Impact: impact of the stakeholder on the project, estimated through power and interest (H: High, M: Medium and L: Low)

The results of this analysis are presented in Table 2.

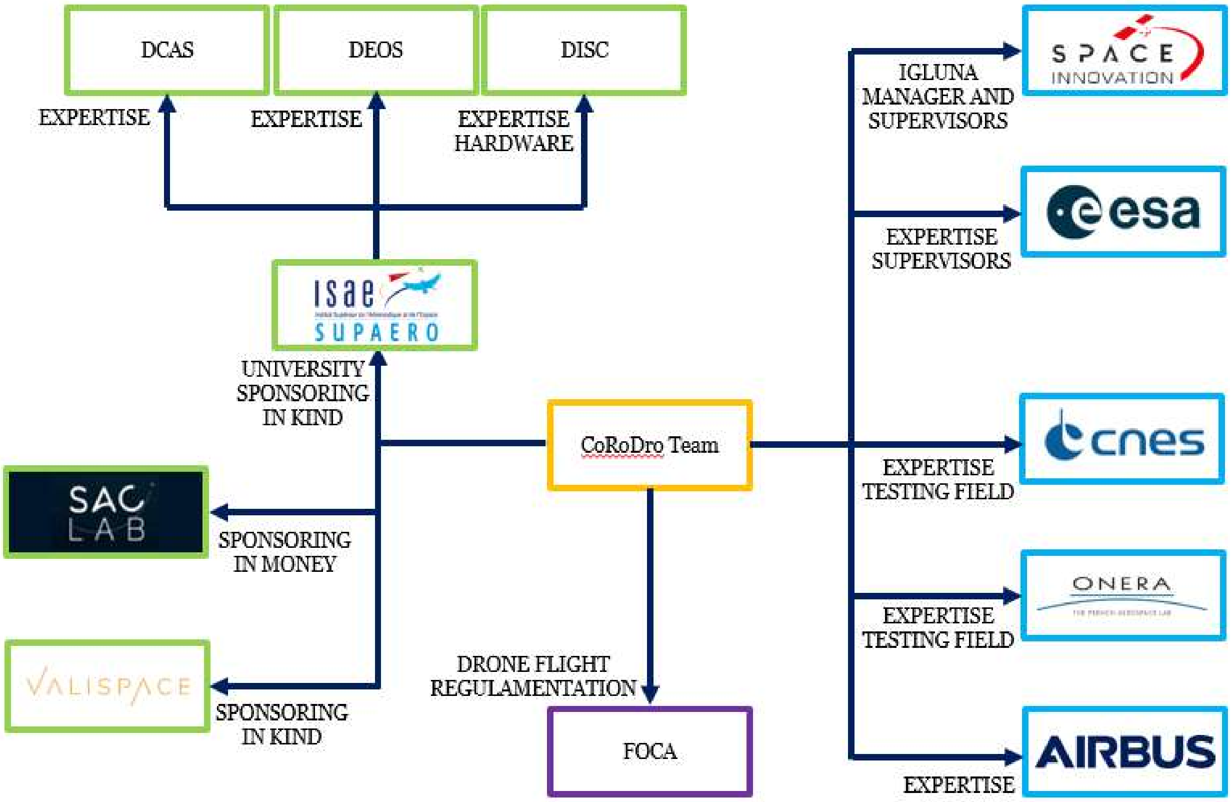


Figure 2: Stakeholders Analysis Overview

Type	Identifier	Role	Power	Interest	Impact	Stakeholder needs	Stakeholder contribution
Active	ISAE-SUPAERO	University	H	H	H	DEOS: Data on the performances of SLAM algorithms DISC: Hardware implementation and performances of robotic systems DCAS: Task and path planning performances for autonomous systems	Expertise Access to testing facilities Access to testing facilities Software Expertise: advise on the best and fitter algorithms to use
Active	SaCLab	Sponsor	H	H	H	Testing results and publications on the project	Expertise Sponsoring
Active	AIRBUS	Sponsor	H	H	H	Keep them up to date	Expertise Primary interest: collaborative systems and mission analysis They are interested in the expertise acquired by the students at the end of the project
Active	CNES	Sponsor	H	H	H	Communication: keep them up to date Test autonomous navigation Test autonomous navigation	Expertise Access to our rover and testing facilities at CNES (Mars terrain) Promote/advertise: project visibility through CNES website
Active	ONERA	Sponsor	H	H	H	Communication: keep them up to date Test Hierarchical Task Planning on the systems	Expertise Sponsoring in kind (software related to autonomous operations) Promote/advertise: project visibility through CNES website Access to open field testing (to be defined)
Active	Valispace	Sponsor	H	H	H	Use of Valispace Software Give feedbacks on the project	Sponsoring in kind (Software)
Active	Verkehrshaus	Exhibition Venue	L	H	L	Exhibition set-up: prototypes not in use on the test bed and video from the test at ISAE-SUPAERO and with partners	Exhibition Space
Active	Pilatus Bahnen	Testing Venue	H	H	H	Carrying out the tests	Testing field
Passive	ESA	Project Supervisor	H	L	H	Well defined documentation and project A successful project	Opportunity to be members of IGLUNA (IGLUNA 2021 is an $ESA_Lab@CH$ project)
Passive	Space Innovation	Project Supervisor	H	H	H	IGLUNA rules and deadlines They can decide if we can finish or not the project	Possibility to attend IGLUNA 2021 Events/reviews/advice Workshops
Passive	FOCA	Regulators Flight Rules	H	L	H	Drone flight rules	Rules to comply to
Passive	Previous IGLUNA Members	Possible Advisors	L	H	M		Advice on IGLUNA
Customers	Teams P03, P08, P10	IGLUNA Teams	L	L	L	Collaboration during the project	Bilateral feedback
Customers	Exhibition Audience	Visitors of the Verkehrshaus	L	H	L	Raise public awareness about space and Switzerland	Project visibility
Customers	Media	Press	L	L	L		Project visibility

Table 2: Stakeholders Analysis

1.7 Project Goals and Objectives

During the VFC, we envision to test autonomous operations and autonomous navigation on collaborative robotic systems, specifically a rover and a drone. The main project objectives can be summarized in three bullet points:

- Main Objective:
  - Perform collaborative outdoor navigation.
- Secondary Objectives:
  - Identify interesting targets.
  - Plan and move toward them thanks to an autonomously computed path.

Tables 3 and 4 present respectively the goals and objectives of the project.

Objective title	Area	Objective description	Relevance	Milestone
Perform collaborative outdoor navigation	Test Bed	The systems should perform collaborative outdoor navigation	The objective is to reach a high level of autonomy while exploring new environments	RR
Find targets of interest	Test Bed	The systems should be able to choose sites of interest to explore autonomously	That is extremely important for search and rescue scenarios on Earth	FC
Autonomous path planning	Test Bed	The systems should be able to plan and move toward the targets of interest autonomously	Relevant in both Earth and Moon scenarios where the systems should be able to move autonomously and decide the best trajectory to follow	RR
Exhibition Show	Exhibition	During the exhibition we may show our simulations, videos of the testing that we have done and find interactive ways to present the robotic systems in space	Present the project to the general public, Find new sponsors for the start-up	CDR, RR
Control Center Setup and data transmission	Control centre	The team should define the data to be transferred to the control centre, when it will be transferred. The team should define the hardware set-up	During the FC we will relay data toward the control centre, it is of the foremost importance to decide which type of data to deliver and how	CDR

Table 3: Mission Objectives

Goal Number	Goal Title	Goal Description	Relevance	Area	Milestone
1	Map environment with drone	The drone should be able to provide a global map to the rover	High	Field Campaign	VFC
2	Move the drone	The drone should be able to move in an unknown environment following a grid like path planning algorithm to create the map for the rover	High	Field Campaign	VFC
3	Move the rover toward the targets	The rover should autonomously move on the trajectory that it generates toward the targets	High	Field Campaign	VFC
4	Move the drone toward the targets	The drone should autonomously move on the trajectory that it generates toward the targets	High	Field Campaign	VFC
5	Task Planning for both rover and drone	Both the drone and the rover should schedule their activities and execute them	High	Field Campaign	VFC
6	Recognize interesting targets	While mapping, the drone should autonomously recognize interesting targets	Medium	Field Campaign	RR

Table 4: Project Goals

1.8 Project Top-Level Requirements

In Table 5 the general requirements (RG) for the system are presented. More detail on the requirements can be found in the ”DETAILED PROJECT DESCRIPTION” in Section 2.1. The main requirements are linked to the software - our project focus is to develop codes for autonomous navigation and operations, not the hardware.

Index	Title	Description	Range/Value
RG-1	Decision Making	The implemented algorithm shall be able to provide autonomous decision-making capabilities	E3 [1]
RG-2	Path Planning	The implemented algorithm shall enable autonomous path planning given a map	[-]
RG-3	Task Planning Architecture	The systems should use a Hierarchical Task planner to schedule operations	[-]
RG-4	SLAM	The systems should be able to map and localize in an unknown environment	[-]
RG-5	Deterministic Behavior	The system shall react in the same, predictable, manner given identical stimuli	[-]

Table 5: Project Top-Level Requirements

1.9 Risk Analysis

Our preliminary assessments of possible risks during the IGLUNA campaign is reported in Table 6.

ID	Domain	Description	i.Likelihood	i.Severity	i.CQuan	i.CQual	Mitigation Action	MA. Status	f.Likelihood	f.Severity	f. CQuan	f.CQual	Team Awareness
W-1	Test Bed	Wheel slippage Localization Problems	Medium	Medium	9	Medium Risk	Use sensors to re-localize (We will implement a Visual-SLAM Algorithm)	3	Low	Low	4	Low Risk	YES
W-2	Test Bed	Wheel stuck in terrain	Low	High	8	Medium Risk	Operator on the field can free the rover	3	Very Low	Medium	3	Low Risk	YES
W-3	Test Bed	Stuck Wheel Motor	Low	High	8	Medium Risk	Even with one wheel stuck the system can move We have spare parts in case of hard failure	3	Low	Medium	6	Low Risk	YES
S-1	Test Bed	Snow present during FC Poor depth camera reading	Low	Very High	10	Medium Risk	Test on July, usually no snow on Mount Pilatus during July The rover is equipped with a LiDAR if we cannot generate a good map with the drone	3	Low	Very High	10	Medium Risk	YES
C-1	Test Bed	Sensors covered by dust (e.g. The drone lands near the rover)	Medium	High	12	Medium Risk	In case there is an operator on the field to clean the sensors before starting the test We can make the drone land far away from the rover to generate less dust We can test the drone when the rover is not on the field	3	Very Low	Low	2	Low Risk	YES
C-2	Test Bed	The sensors are not working properly (false readings)	Low	High	8	Medium Risk	On board both systems we have analogue sensors (drone: Tracking+Depth cameras; rover: LiDAR+Depth camera) that can be used to verify the reading (redundant sensors)	3	Low	Medium	6	Low Risk	YES
C-3	Test Bed	The temperature is very low and may damage the sensors	Low	high	8	Medium Risk	We are going to test in July so we should not have this problem We can pre-heat our systems if the temperatures are low	3	Low	Medium	6	Low Risk	YES
B-1	Test Bed	Low temperature for the systems' battery	Low	Medium	6	Low Risk	We are going to test in July so we should not have this problem We can pre-heat our batteries if the temperatures are low	3	Low	Low	4	Low Risk	YES
Comm-1	Test Bed	Loss of data due to unreliable network	High	Low	8	Medium Risk	Send data to CC Wait for validation code If not received, re-send data. The operator on the filed will download the data of both systems on a USB drive (to keep a back up of all our data)	3	Low	Low	4	Low Risk	YES
Comm-2	Test Bed	The rover or the drone or both networks are down	Low	Very High	10	Medium Risk	Both systems can communicate with the control center If not, they are both equipped with memory We can store the data and recover them for post-processing Operator on the field will download the data of both systems on a USB drive (to keep a back up of all our data)	3	Low	Very Low	2	Low Risk	YES
Dron-1	Test Bed	The drone damages something valuable during its flight	Medium	High	12	Medium Risk	Our testing field will be secluded from the other teams and the hikers We require to have a buffer zone of 2 m per side as additional safety measure In case of contingency we can stop the propellers directly by the operator on the field , or if possible try to safely land We will get near the drone only when it has reached a complete stop	3	Very Low	Very Low	1	Low Risk	YES
TW-1	Design - Development	Too much work for the team	High	Very High	20	High Risk	Recruit people Correctly manage tasks	3	Medium	Medium	9	Medium Risk	YES
COVID-1	ALL	The project may register scheduling delays, sponsoring problems and decrease overall time for the project of team members due to COVID-19 or the amount of university related work	Very High	Very High	25	High Risk	Better defined work packages and objectives (each member knows what to do), try to meet in person when possible, Zoom calls when not Keep the team motivated with the objective of the FC Think about a redistribution of tasks based on the advancement of the different teams	3	Medium	Medium	9	Medium Risk	YES
COVID-2	ALL	We may not be able to join the FC in July, because of COVID-19 related restrictions	Medium	Medium	25	High Risk	Keep on with the testing at ISAE Test with our partners at CNES to have a set-up similar to the FC	3	Medium	Medium	9	Medium Risk	YES
F-1	Test Bed	There is wind in the field	Medium	High	12	Medium Risk	If the wind is more than 5 m/s, we will not fly the drone We will test degraded condition with only the rover Teleoperate the drone to have better control of it Manually move the drone on field to create the map	3	Very Low	Very Low	1	Low Risk	YES
F-2	Test Bed	Rainy day	Low	High	8	Medium Risk	Test only the rover in contingency situation keeping into account the slippage of the wheels with the Visual SLAM algorithm Manually move the drone on field to create the map while protecting it with an umbrella	3	Very Low	Very Low	1	Low Risk	YES
Transp-1	Logistics	Damage on the systems because of transportation	Low	High	8	Medium Risk	We have an insurance on the systems, we are bringing with us spare parts We will bring the system with us probably driving The packaging will be performed by our sponsors, who are used to deliver hardware components to test sites	3	Low	Medium	6	Low Risk	YES
Anim-1	Test Bed	A wild animal tries to interact with the systems	Low	Medium	6	Low Risk	The operator will try to scare the animal away	3	Low	Low	4	Low Risk	YES
Spons-1	Test Bed	We cannot find enough money to sponsor to send more people to the FC	High	Medium	12	Medium Risk	In the worse case we will still be able to send 3 people to the FC at least (Sponsoring from Space Innovation and Politecnico di Torino).	3	Very Low	Medium	3	Low Risk	YES
Pres-1	Logistics	Problems of network connection during reviews and problems in sharing the Student Documentation and presentations	Low	Low	4	Low Risk	Have all the documents on Drive for backup For the connection problem have always a second available network (e.g. cell phone or Eduroam) during reviews	3	Very Low	Very Low	1	Low Risk	YES

Table 6: Risk Analysis

1.9.1 Risk Analysis matrix

Based on the risk analysis from the PDR (6) and the new risk found since the PDR, we've updated our risk analysis with the FMECA and HAZOP techniques :

- FMECA technique to detect and mitigate all machine failure modes, effects and criticalities described using two/three risk factors (Severity, Likelihood & (Detection)).
- HAZOP technique specifically fabricated to address all human injuries and potential events with much lower tolerance levels described with two risk factors (Severity & Likelihood).

After updating the FMECA and HAZOP risk analysis we obtain the following matrices. The first one shows the number of risk in each Likelihood versus Severity category before the application of the mitigation Figure 3. Then after mitigation we obtain a second matrix Figure 4, we can see that most of the risks have now a Low or Medium Severity and Likelihood.

INITIAL RISK ANALYSIS

Likelihood \ Severity	Very Low	Low	Medium	High	Very High
Very High					
High		2	1	5	
Medium			1	3	
Low		1	3	5	1
Very Low			1	1	

Figure 3: Risk matrix before mitigation.

FINAL RISK ANALYSIS (USING MITIGATION ACTIONS)

Likelihood \ Severity	Very Low	Low	Medium	High	Very High
Very High					
High					
Medium		1	3		
Low	1	7	6		
Very Low	2	2	1	1	

Figure 4: Risk matrix after mitigation.

1.10 IGLUNA Field Campaign Results

We can hierarchically summarize our expectations on the IGLUNA VFC testing as:

- Test SLAM algorithms:
  - Generate a 3D map with the drone.
  - Translate the drone map in a 2D global grid-map to be used by the rover.
  - Implement SLAM algorithms on the rover as a back-up if the drone generates a flawed map.
  - Check the performance for SLAM algorithms in terms of:
    - Computational cost.
    - Computational time.
    - How many real obstacles are considered in the grid map.
    - How well the system located itself while creating the outdoor map (during the "mapping" phase).
    - How well the system located itself in the final map (during the "mission" phase).
- Test path planning algorithms:
  - Define a grid-like path for the drone to map the environment.

- Define the path planning algorithms for the drone to reach the different targets (at least 3).
- Define the path planning algorithm to move the rover around.
- Check the performance for SLAM algorithms in terms of:
  * Computational cost.
  * Length of the path.
  * Clearance from the obstacles.
  * Path smoothness.

• Test task planning algorithms:
  - Check the integrity of the overall plan.
  - Check the feasibility of the plan.
  - Check its connection with path planning and SLAM, demonstrate that the core task planning algorithm can effectively orchestrate and interface with the other two algorithms.
  - Check the performance for task planning algorithms in terms of:
    * Computational cost.
    * Success of the task executed.
    * Number of re-evaluation of the plan due to unforeseen events.
    * Overall behavior of the systems.
    * Overall remaining resources (e.g. system battery level) at the end of the tasks.

All the algorithms will be interfaced with ROS as middleware. The data that will be stored on the systems will be as well interfaced with ROS and saved as ”ROSbag”. That saving format can help us replay in simulation the behaviour of the systems under study and better estimate the performances during the post-processing phase.

1.11 Business Model

1.11.1 Executive Summary

CORODRO’s autonomous and collaborative systems can be summed up in 3 words: safe, reliable and intelligent. Our aerospace company will specialize in aerospace logistics, operations and space technology maturation (R&D) for collaborative robotic systems. The company will specialize in the management of autonomous operations and navigation for a range of solutions: from Earth drone to interplanetary probes. Our mission is to conquer space leveraging and shaping the autonomous operation domain with the cutting-edge advancement of artificial intelligence. We point toward a world where the autonomy of robotics systems is the paradigm both on Earth and in space. We are going to exploit the algorithms and the knowledge of cutting-edge R&D space solutions on Earth. Exploiting an AI’s augmented operational layer, our systems will be able to decide their goals, to plan the best sequence of action to reach them, and to execute the plan autonomously moving and acting in their environment. The head office will be located near our primary customers and partners in Toulouse: Airbus, ONERA, CNES and ISAE-SUPAERO. The founding team is a cluster of young and motivated engineers ready to push forward and innovate in the aerospace industry. We welcome trainees and interns always to bring new and challenging ideas, ready to shape the future of space engineering. We are going to provide valuable expertise in the domain of autonomous operations and navigation, while assisting the client from idea conception to prototyping, to real-hand application of robotic aerospace solutions. Our envisioned action radius encompasses:

- Satellite autonomous operations
- Collaborative robotics solution for Moon and deep space surface exploration.
- Management of difficult and potentially dangerous situations for humans through collaborative robotics: (i) exploration of earthquakes struck zones, (ii) marine or underwater exploration, (iii) archaeological research, (iv) wild animal biological research, (v) aerial monitoring of infrastructures, (vi) 3D mapping, (vii) traffic monitoring, (viii) delivery activities.
- Sustainable and reliable robotic help for the 4.0 agriculture sector.

Our objective is to acquire expertise while growing and providing the best cutting-edge technology services to our customers. We guide our customer from the concept idea to the realization of their vision of the future. Our company has the expertise that spans from mission design to the operational layer and the autonomous navigation domain. Our company will double as an operational control centre for the customers that will require that service. Our primary income sources are divided into two main streams: (i) Earth's applications, (ii) space applications. Our envisioned mean revenue will be 1.3 M$ during the first three years of activity, and they are based on the Earth's applications. More information on our revenues' prospect can be found in Annex 9.3. In general, we point to the outdoor drone market related to farming, goods delivery and building/infrastructure inspection. The knowledge and expertise gained on the "Earth's field" will be a valuable asset for the space mission scenario.

1.11.2 Canvas Model

The entries of the business model canvas are presented in Figure 5 and they are analysed more in detail in annex 9.3.1.

BUSINESS MODEL CANVAS

Designed for: CoRoDro - IGLUNA 2021

Designed by: Jasmine Riman

Date: 06/11/2020

Key Partners

- Research centres working on robotic autonomy and autonomous navigation.
- Technical Universities focused on R&D.
- Companies focused on the deployment of robotic solutions for Earth application (agriculture, movies).

Key Activities

- Software Development
- Mission analysis
- R&D to increase algorithm reliability and re-usability

Key Resources

- Workstations to develop the algorithms
- Software f
- Testing hardware
- A core team of at least 10 people
- Initial capital

Value Propositions

Our objective is to push the autonomy of any kind of robotic system.
We will deploy robotic platforms with:

- Autonomous operations capabilities
- Autonomous navigation in an unknown environment.
- Goal-oriented task manager.

Customer Relationships

- Website - Contact Section:
  - Email
  - Real-time chat
  - Call centre (5/7 day; 8/24 h)
- Social Networks (LinkedIn, Facebook, Twitter):
  - Opinion section
  - Direct Messages
  - Comments to post or twits

Channels

- Distribution channels with our key partners.
- Industrial exhibitions.
- Web site

Customer Segments

Businesses:

- Delivery Companies
- Space Industries
- Agricultural-Based Companies
- Space Agencies
- Warehouses management companies.
- Building and Infrastructure companies.

Others:

- Military segment
- Firefighter segment

Cost Structure

- Testing Hardware
- Facilities rent and office equipment
- Software
- Website, social media
- Legal support
- IT support
- Mortgage start-up funding
- Personnel Salary
- Taxes on company and personnel salary

Revenue Streams

- Primary revenue Earth sector applications: (i) mapping, (ii) inspection and monitoring, (iii) data analytics, (iv)emergency, (v) goods delivery.
- Secondary revenue Space sector applications.
- Customers can pay up-front or through financing
- Affiliate marketing

Figure 5: BUSINESS MODEL CANVAS (More information can be found in Annex 9.3.1)

2 DETAILED PROJECT DESCRIPTION

2.1 Complete Project Requirements

The high-level project requirements, introduced in Section 1.8 (code RG), are detailed in this sections in six different categories:

- Functional Requirements (RF), Table 7.
- Performance Requirements, (RP), Table 8.
- Interface Requirements (RI), Table 9.
- Safety Requirements (RS), Table 10.
- Constraints Requirements (RC), Table 11.
- Communication Requirements (COM), Table 12.

We used the documentation in the ERGO project [31] as a guideline to write our requirements.

No.	Title	Description	Value/Range
RF-1	Systems Control	The decision layer shall provide ”go-to command”	-
RF-2	Decision Layer Activity Decomposition	The decision layer shall be able to decompose high level commands into low-level substasks	-
RF-3	System State	The Task Planner should be able to record the ”state of health of the system” and based on that it should re-optimize the plan	-
RF-4	Navigation Map	The drone shall be able to have an accuracy of (5.5 cm $\pm$ 0.5 cm) on the obstacles it detects and that it registers on its generated map	$5.5cm \pm 0.5cm$
RF-5	Navigation Map Obstacles	The drone shall be able to detect 90% of obstacles that have a height superior or equal to 6 cm	$> 0.06m$
RF-6	Operating System	The selected operating system for the drone and the rover shall be based on Linux	-
RF-7	Language	The implemented code shall be in C++ or Python language	-
RF-8	Resource Estimation	The rover software shall be able to estimate the required resources to execute the path	-
RF-9	Navigation Map	The map areas of the navigation map shall include cost data usable to inform of the most desirable area to be traversed	-
RF-10	Decision	The decision layer shall provide validated plans	-
RF-11	Systems	The systems layer shall gather the information produced	-
RF-12	Algorithm	The algorithm should provide an executor of the decision making plan	-
RF-13	Memory Storage	The rover shall have a memory storage to store data before sending it to the control center	-
RF-14	Memory Storage	The drone shall include on-board memory storage	-

Table 7: Functional Requirements

No.	Title	Description	Value/Range
RP-1	Upgradability	The design of the implemented software shall allow for future software upgrades	-
RP-2	Localization Drone	The drone localization shall have an accuracy of (0.5 m $\pm$ 0.2 m)	(0.5$\pm$0.2)m
RP-3	Localization Rover	The rover localization shall have an accuracy of (0.5 m $\pm$ 0.2 m)	(0.5$\pm$0.2)m
RP-4	Framework	The modeling framework shall allow timing constraints (e.g. battery discharge time)	-
RP-05	Obstacle detection	The map generated shall register at least 80% of obstacles that have a height superior or equal to 6cm	-

Table 8: Performance Requirements

No.	Title	Description	Value/Range
RI-1	ROS Interface	The implemented algorithms (Localization and Real-Time Mapping, path planning, task planning) shall be able to be interfaced with ROS	-
RI-2	Communication	The systems should be able to communicate with IP addresses with the control center and between each other	-

Table 9: Interface Requirements

No.	Title	Description	Value/Range
RS-1	Logging	The design of the implemented algorithm shall include items supporting the diagnosis of software anomalies such as log files	-
RS-2	Emergency Stop (E-stop)	The drone and the rover shall be able to be switched off from the control center	-
RS-3	Operator on Field	There should be always be an operator on the field that can act fast and stop the rover or the drone without waiting for the delayed answer from the control center	-
RS-4	Field Set-Up	A buffer zone of 2 meters from the other teams should be considered while planning the Field Campaign Set-Up	2 m

Table 10: Safety Requirements

No.	Title	Description	Value/Range
RC-1	Budget Limit	The maximum project budget in cash is 3400 euros	3400 euros
RC-2	Transport Regulation	All the systems should be covered by an insurance up to 1 million CHF	1 million CHF
RC-3	Outside Temperature	The rover and the drone shall be able to withstand temperatures between -5℃± 2℃ and +40℃ ± 2℃	(-5℃ ± 2℃) and (+40℃ ± 2℃)
RC-4	Windy Environment	The drone should be able to withstand a wind of (5 m/s ± 0.5m/s)	(5 m/s ± 0.5 m/s)
RC-5	Uneven Terrain	The rover should be able to navigate on a terrain with a slope of (-10°±2°)	(+10° ± 2°)
RC-6	Drone Hardware	The drone base (structure, OBC, rotors, mechanical and power interfaces) is given by our sponsors	-
RC-7	Rover Hardware	The overall rover (structure+sensors) is given to us by our sponsors at DISC	-
RC-8	AR Codes	The drone should recognize specific targets: AR codes	-
RC-9	Rope	The border of the field should be bounded by a rope in order to restrict the drone motion	-
RC-10	Obstacle Capabilities of the Rover	The rover cannot cross anything that is higher than 6cm with a slope greater than 90° or something that is higher than 15cm with a slope greater than 60°.	-

Table 11: Constraints Requirements

No.	Title	Description	Value/Range
COM-01	Bandwidth	Maximum bandwidth with control room of 2.5 Mb/s	2.5 Mb/s
COM-02	Communication	The drone and the rover shall always remain in communication range	-

Table 12: Communication Requirements

2.2 Trade-offs

Two trade-offs have been conducted to decide the best set of sensors to equip the drone with for the mapping task. Table 13 identifies the characteristics that we qualified as relevant for the choice of sensor.

Entry	Intel RealSense T265	Intel RealSense D435i	RPLidar A1
Range	infinite (object must be bright enough)	10 m	12 m
Accuracy	10% (at 2 m)	2% (at 2 m)	1° (0.2 cm)
Field of View	163±5°	69.4°×42.5° (±3°)	360°x 0°
Weight	55g	72g	170g
Integrated Software Stereo	no	yes	no
Integrated Software SLAM	yes	no	no
Power Consumption	1.5W	3.5W	0.5W

Table 13: Main characteristics of the sensors

2.2.1 Benchmarks

According to the requirements of the project, we have defined some benchmarks that are summarized in Table 14.

Score	Range	Accuracy	Field of View	Weight	Integrated Software	Power
0	<3m	>7% height	<=47°	<50g	No algorithm	>5W
1	<=5m	<7% height	>47°	<=100g	On Github	<=5W
2	>5m	<1% height	>51°	>100g	Native algorithm	<2W

Table 14: Benchmarks

2.2.2 Trade-off 1

The first trade-off aims at deciding which camera to use for the drone. From Table 15, we can see that the D435i scores better on our benchmarks. However, we have decided to mount both cameras on the drone. Indeed, the scores are pretty close, and since there is the possibility to put both cameras on the drone, we preferred this solution. This will provide native stereo algorithm and localization algorithm at the same time. Thus, we will greatly simplify the software development.

Features	Weight	Intel RealSense T265	Intel RealSense D435i
Range	1	2	2
Accuracy	2	0	1
Field of View	1	2	2
Weight	2	2	2
Integrated Stereo	3	1	2
Integrated SLAM	2	2	1
Power Consumption	1	2	2
Weighted Average		2.43	2.86

Table 15: Trade-off to decide which camera was the most suitable payload to enable localization with the drone

2.2.3 Trade-off 2

The second trade-off aims at deciding whether to use a LiDAR instead of the D435i. We can clearly see on Table 16 that the camera scores much better than the LiDAR, according to our benchmarks and weights. Also, we believe that embedding the LiDAR on the drone may generate many problems related to the power consumption and overall drone autonomy. However, even in this case, we did exclude the use of LiDAR until the RR. In fact, the map generated by the LiDAR is quite precise. In the documentation, we analysed both drone's configuration and commented on the final choice of depth camera in Section 2.15.

Features	Weight	RPLidar A1	Intel RealSense D435i
Range	1	2	2
Accuracy	2	2	1
Field of View	1	2	2
Weight	2	0	2
Integrated Stereo	3	0	2
Integrated SLAM	2	0	1
Power Consumption	1	2	2
Weighted Average		1.43	2.86

Table 16: Trade-off to decide whether to use a LiDAR instead of the D435i

2.3 Project Management

2.3.1 Work Packages Overview

The overall project management tasks are presented on our Trello board. We used the "power-ups" of Trello to design our gantt, Figure 6, and our work-package breakdown structure (briefly shown in Figure 7 and detailed in Annex 9.4). In Figure 7 we detailed the work breakdown structure in respect to the different milestones, while in the Annex 9.4 they are detailed by "topic". As a mean to track the main project needs, we prepared a concise development plan presented in Figure 8. The plan outlines the critical steps we must satisfy to realise a successful FC. It is our internal checklist to verify the advancement of the project.

We were envisioning some test outdoor before the RR. Unfortunately, due to most of the Path Planning team leaving soon after the CDR, we had to postpone the drone testing outdoor after the RR in June [1]. Moreover, to accomplish our objectives regardless of the decreased number of members, we started a more agile code development plan. The code is prototyped, tested, improved and finally re-tested in its final configuration. We started testing almost every week, multiple times per week. That effectively allowed the team to advance quicker with the software interfaces and acquire more knowledge on testing. The buffer time of one month, the month of June has been enough to guarantee the end of all the testing. The development plan for June is shown in Figure 9.

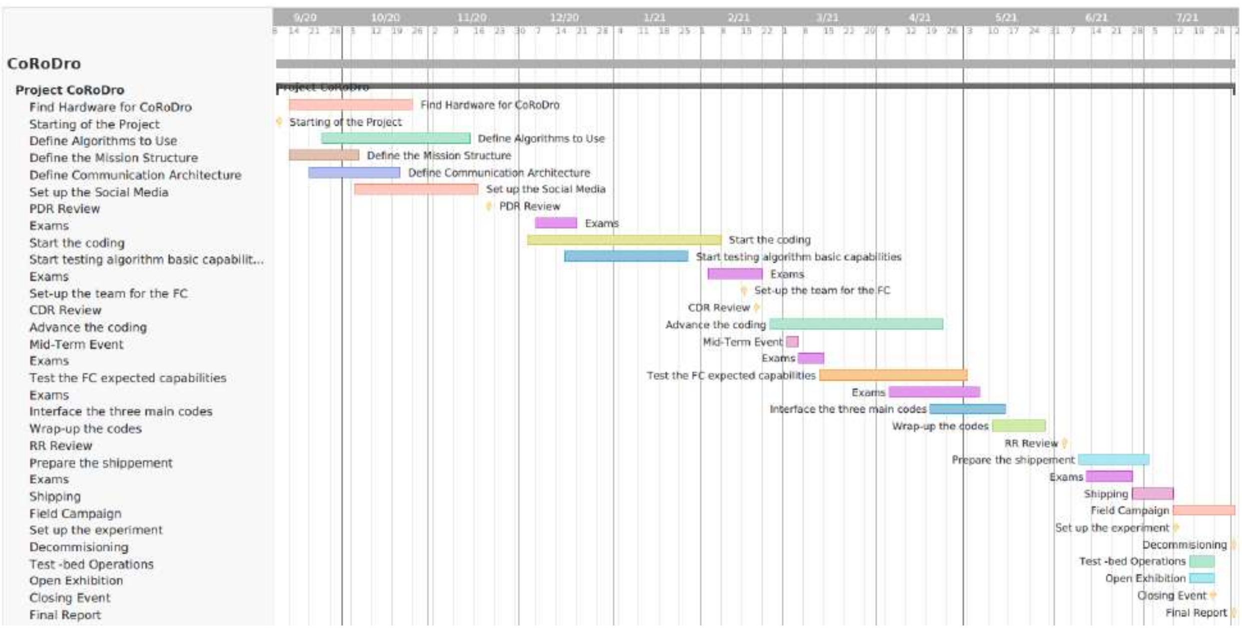


Figure 6: Gantt chart for the CoRoDro Team

[1]-https://drive.google.com/file/d/13F8l3EXAGFRPU9ezoTm6AV2lXdQspX5O/view?usp=sharing

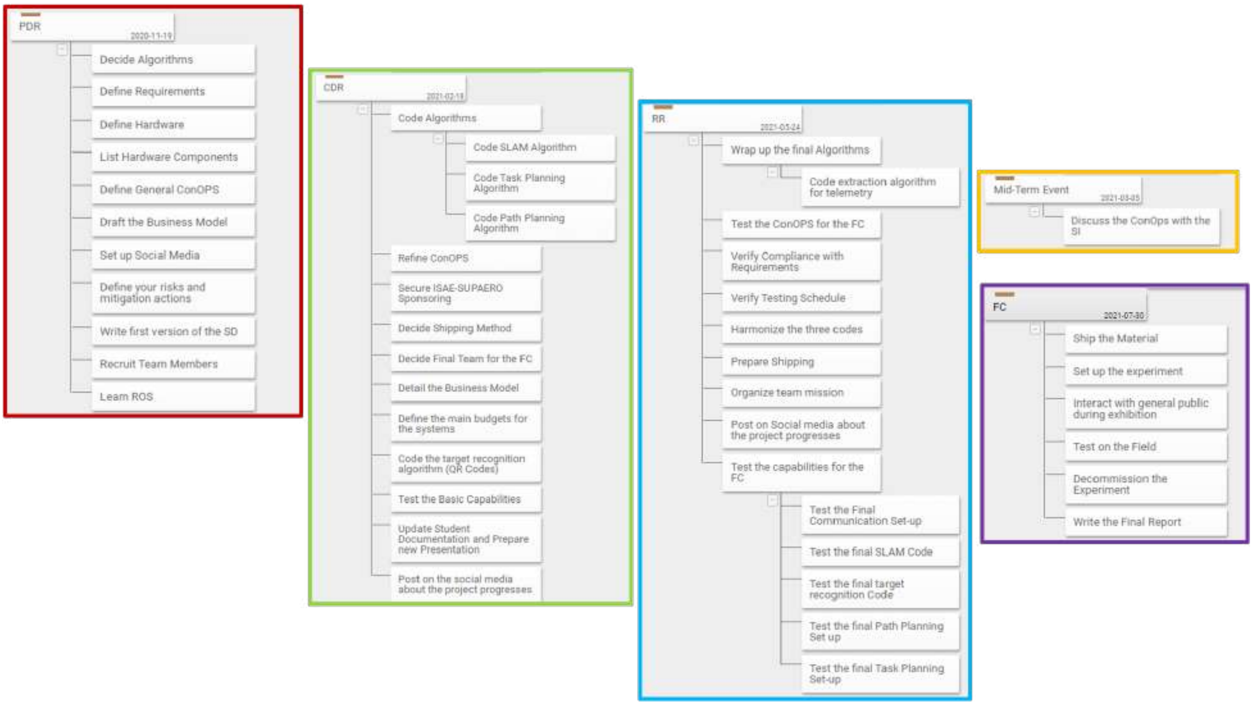


Figure 7: Work-breakdown structure for review of the CoRoDro Team

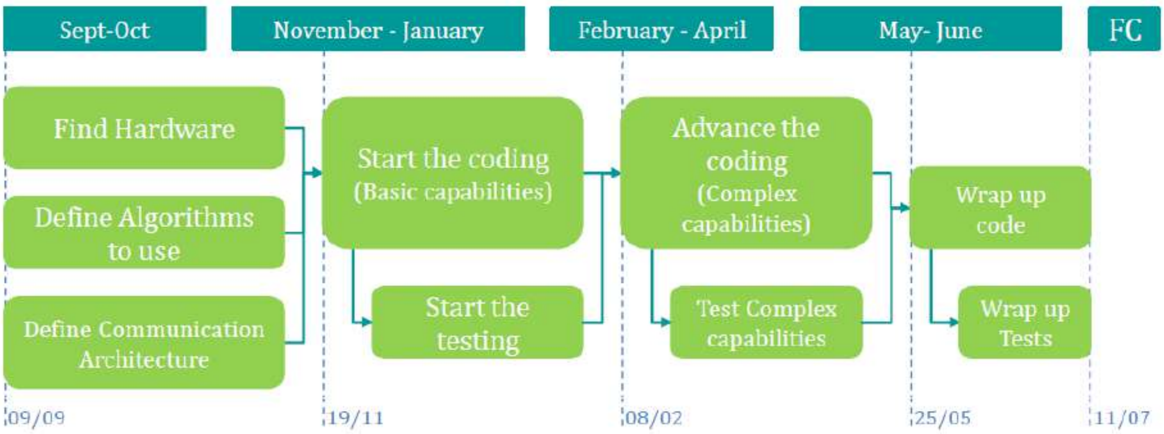


Figure 8: Concise development plan for the CoRoDro Team spanning the academic year.

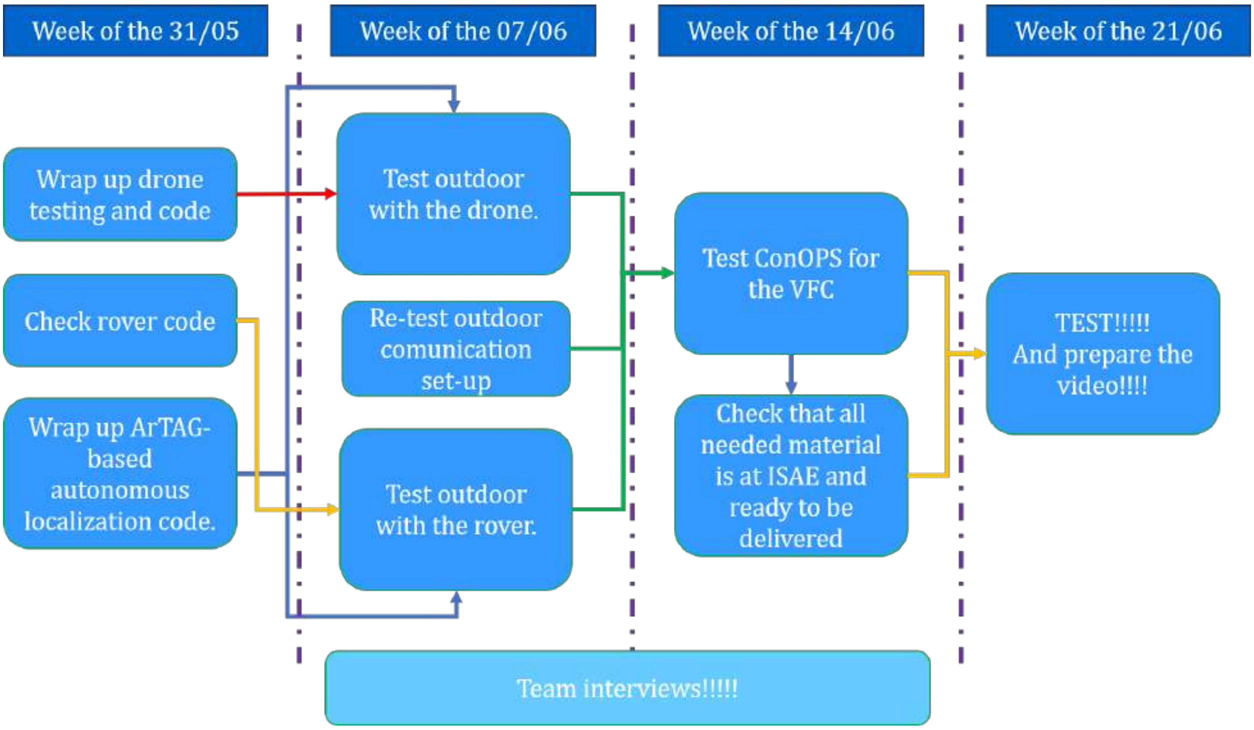


Figure 9: Development Plan for the month of June

2.4 Virtual Field Campaign

Because of the COVID-19 related complexity to come to Switzerland for the Field Campaign, FC, we have decided to organise our Virtual Field Campaign, VFC, in Toulouse. We organized a four days' testing at ISAE in the gym, at the Martian terrain at CNES and at the drone testing terrain of ONERA (Figures 10, 11, 12). Unfortunately the test at CNES had to be cancelled for bad weather.

Figure 10: ISAE-SUPAERO gym.

2.4.1 Testing for the VFC

Testing in Toulouse requires a thorough preparation. We have to ask clearance for non-European members of the team to be able to access the CNES site as well as for filming at CNES. We need to set up our own control center in the three locations. To set up our control center we will need chairs, tables, and a WIFI router to establish our network and extension cables.

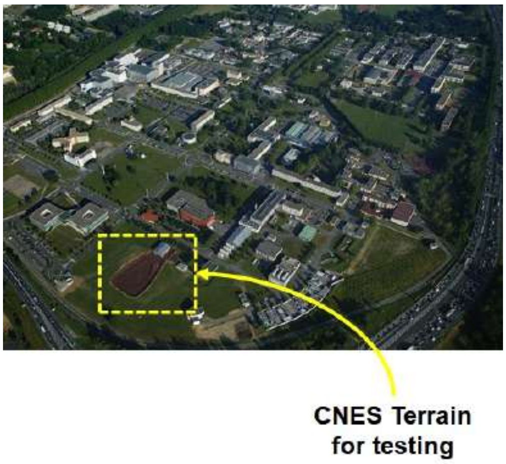


Figure 11: Martian terrain at CNES, Toulouse

Figure 12: Drone testing Terrain of ONERA, Esperce

2.4.2 VFC videos

We will film the testings with the help of the ISAE-Supaero's video club, JTS, and include the best of the testing in our 20 minutes video that will be showed live on Youtube in mid-July as well as in the loop video that will be displayed at our stand at the exhibition hall in the Swiss Museum of Transport. The VFC project show will also contain interviews from members of the team explaining the project and their work and a presentation of our system.

JTS

Figure 13: JTS, ISAE-Supaero's video club

2.5 Resources

2.5.1 Task-force

The seventeen members are divided in sub-teams or squads: (i) project management, (ii) task planning, (iii) path planning, (iv) Localization and Real-Time Mapping, (v) telecommunication, (vi) mission analysis, (vii) outreach and communication. The members are mostly on their first or second year of Masters or in the last year of Bachelors. We lost all of our funding members but one, the project manager, during the project. However, we recruited around twenty people at the end of October (starting from the 26/10/2020). Thus, we had resources to allocate to the various teams. Unfortunately, just before the CDR other six people left the project. After the CDR the Path Planning team lost three members and the Localization and Real-Time Mapping team one member. At the same time, Mission Analysis lost two members.

In Section 6.3, we presented the list of all members up to 27/05/2021 and the amount of time they should be able to dedicate to the project. Figure 14 summarizes this last information, here in the Detailed Documentation.

Table 17 shows the estimated time to accomplish the task versus the real-time spent in each of it.

Figure 14: Time dedicated to the project by each team member

Review	Team	Time to dedicate [h/week]	Total time to perform the work [weeks]	Estimated time to complete tasks[h]	Real time to complete tasks[h]
PRD	Mission analysis	25	7	150	179
PRD	Localization and Real-Time Mapping	45	7	225	321
PRD	Path planning	26	7	130	186
PRD	ROS interfaces	7	7	14	35
PRD	Outreach and Communication	15	7	105	110
PRD	Project management	10	7	70	210
PRD	Telecommunication	7	7	49	21
PRD	Task planning	7	7	49	40
CRD	Mission analysis	19	8	114	114
CRD	Localization and Real-Time Mapping	31	8	186	186
CRD	Path planning	29	8	174	174
CRD	Outreach and Communication	9	8	54	54
CRD	Project management	17	8	102	102
CRD	Telecommunication	6	8	36	36
CRD	Task planning	15	8	90	70
RR	Mission analysis	19	10	175	53
RR	Localization and Real-Time Mapping	31	10	360	320
RR	Path planning	19	10	208	96
RR	Outreach and Communication	9	10	150	48
RR	Project management	17	10	150	105
RR	Telecommunication	6	10	50	30
RR	Task planning	15	10	200	120

Table 17: Planned and effective time allocation for the CoRoDro project. The values will be updated at each revision

2.5.2 Budget

2.5.2.1 Income Our income, presented in Table 18, is composed of donations in money and sponsorship in kind. We have been recently granted with a new donation in cash by the Foundation ISAE on January the 25th 2021. 50% of this donation is anticipated, the rest of it will be donated as a reimbursement of the actual used money up to 1250 €. Most of our donations are of the ”reimbursement” type: the students will be reimbursed with the payed amount of money up to a maximum. The other sponsors in kind of the CoRoDRo team, which are ONERA and the DISC and DCAS departments of ISAE-SUPAERO, allowed us to secure the feasibility of the project without any expense by providing us all the necessary hardware and software. That includes the computers on which we develop our software, the drone, the rover, and all necessary material to maintain and repair the robotic systems. That is why the expenses of CoRoDRo are mainly focusing on financing students for the final event of IGLUNA, the FC. They are presented in detail in paragraph 2.5.2.2.

Income	Type	Value (€)	Date	Note
Fondation ISAE - 1	Donation	1250	25 Jan 2021	-
Fondation ISAE - 1	Donation	1250	31 Jul 2021	Reimbursement
Space Innovation	Donation	554	31 Jul 2021	Reimbursement
SacLab	Donation	346	31 Jul 2021	Reimbursement
DISC ISAE-SUPAERO	Sponsoring in Kind	12500	9 Sep 2020	-
DCAS ISAE-SUPAERO	Sponsoring in Kind	6000	9 Sep 2020	-
ONERA	Sponsoring in Kind	3000	15 Dec 2020	-

Table 18: Financing in money and kind of the CoRoDRo Team

2.5.2.2 Expense Currently, the foreseen expenses are mainly related to the FC in its virtual format event and are presented in Table 19. We are going to perform all our testing in Toulouse at CNES. However, we foresee possible participation in person to the events of the FC if the quarantine requirements are lifted. In that specific case, only two students will be sent to the FC to answer questions about the project, stream shows our testing and be there for the Q&A section of the YouTube stream. We do not foresee bringing our systems with us, so no shipping expenses are expected.

Expense	Type	Value (€)	Note
Wifi Router	Equipment	100	[-]
Card Boxes	Equipment	50	[-]
Car Renting	Equipmen	20	To move the equipment to CNES. The car is provided by ISAE.
Meals at CNES	Travel	600	20€/person, 10 people, 3 days at CNES during testings
Meals in Switzerland	Expenses	700	Meals for 1 person 50€/person for 14 days
Goodies	Expenses	100	General expenses for the exhibition
Traveling to Switzerland	Expenses	200	Addendum if the 300 CHF of the SI do not cover all the travelling expenses

Table 19: Estimated expenses for students for the FC (apart from the goodies, which is a general expense of fixed value for the team)

2.6 Project Set-Up

During the Field Campaign, we would like to test the capabilities of our systems to navigate and operate autonomously in an unknown environment effectively. The required field dimension was defined, starting from the dimension of the testing lab at ISAE-SUPAERO. This first estimation has been then refined during the bilateral meetings with our Space Innovation coach, which decided on an average measure of 10x10 meters with 2 meters of buffer in length and width. From our tests here at ISAE-SUPAERO, we evaluate that the ”Basic Capabilities” take up to three hours between setting up the experiment, connect the systems, connect with the computer on the field and start the experiment. After this first test, the team decided to keep always a buffer time to retake the tests, incrementing the expected time by 50%. During the FC, we will set-up our terrain as shown in Figure 15.The dots on the borders represent AR-tags which are used in the first phase of the drone mission to calibrate its position on the field. Grey dots represent AR tags that simulate obstacles or point of interests that have to be visited by the systems. They will be placed randomly on the field, representing the targets for the drone and the rover during the exploration phase. Once the ar tags are detected, the task-planner decides which ar tags are to be visited by the drone, or by the rover.

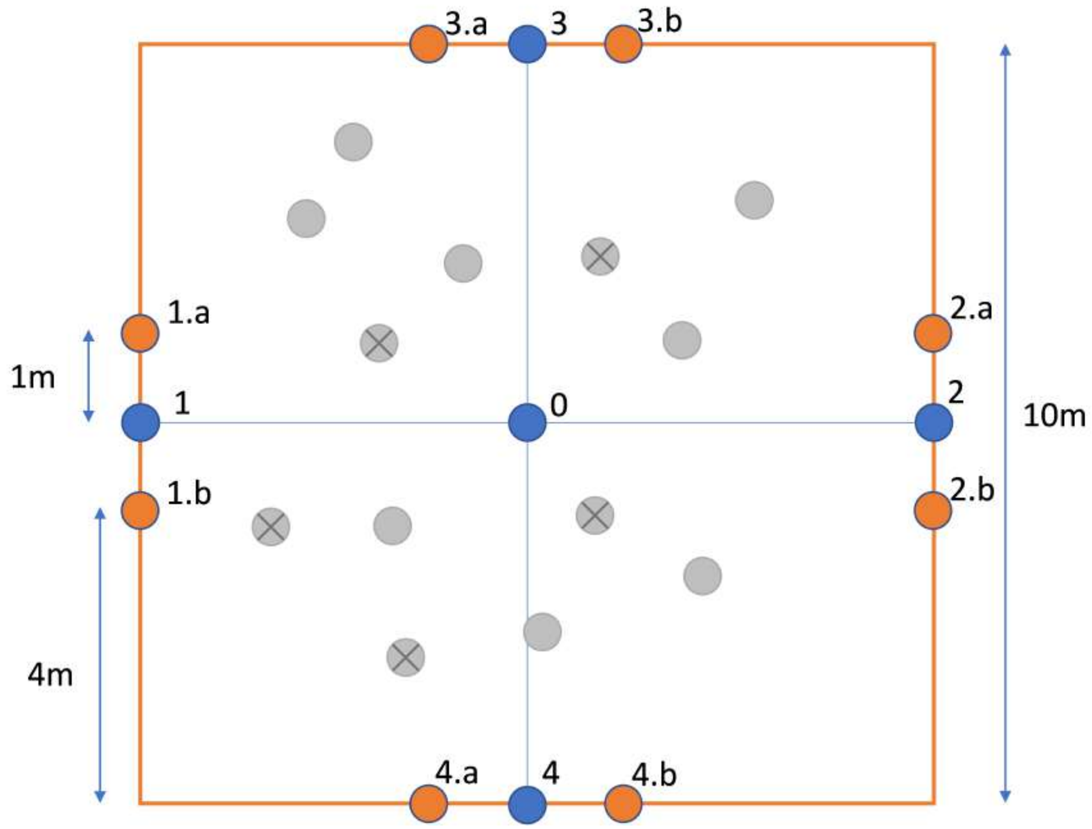


Figure 15: Terrain setup

Our requirements for the testing campaign can be summarized as:

- The envisioned testing sites should be:
    - Preferably flat;

- With dimensions of 10x10 $\pm$ 2 meters;
- Configurable by the team: we would like "to set-up" sites of interest and targets that the drone and the rover should reach, as well as creating more obstacles for the rover.

Following our three terrain requirements the testing have been conducted in the ISAE Gym, Figure10, and at ONERA, Figure12.

- We would need at least one person at the control station, one drone pilot and one operator on the field during the test:
  - One drone and rover operator for safety reasons (changing batteries or move the systems).
  - One drone pilot to intervene in case of any contingency situation with the drone.
  - One control center operator to monitor the test.
- Our tests will span all day long:
  - The morning will be reserved to main testing;
  - The afternoons are reserved to re-test or repeat parts of the tests if needed.

During the exhibition, we wish to show videos of our testing at ISAE-SUPAERO and with our partners.

2.7 Concept of Operations

A graphical representation of our ConOps for the IGLUNA Field Campaign are presented in Figure 16, which are further detailed in Section 2.7.2. The drone and the rover are systems provided by our partners, Section 2.10.2. The drone has autonomy for around ten minutes, while the rover can operate up to five hours. They both move thanks to a battery. During the FC, the drone will be in charge of:

- Create a map for the rover.
- Reach points that the rover cannot reach because there are too many obstacles on its way. The use of the drone will permit to increase the overall mission return.

On the other hand, the rover has more computational power on board and longer battery life. Throughout the FC, the rover will:

- Compute the operational plan for itself and the drone.
- Use the post-processed drone map to optimize its path.

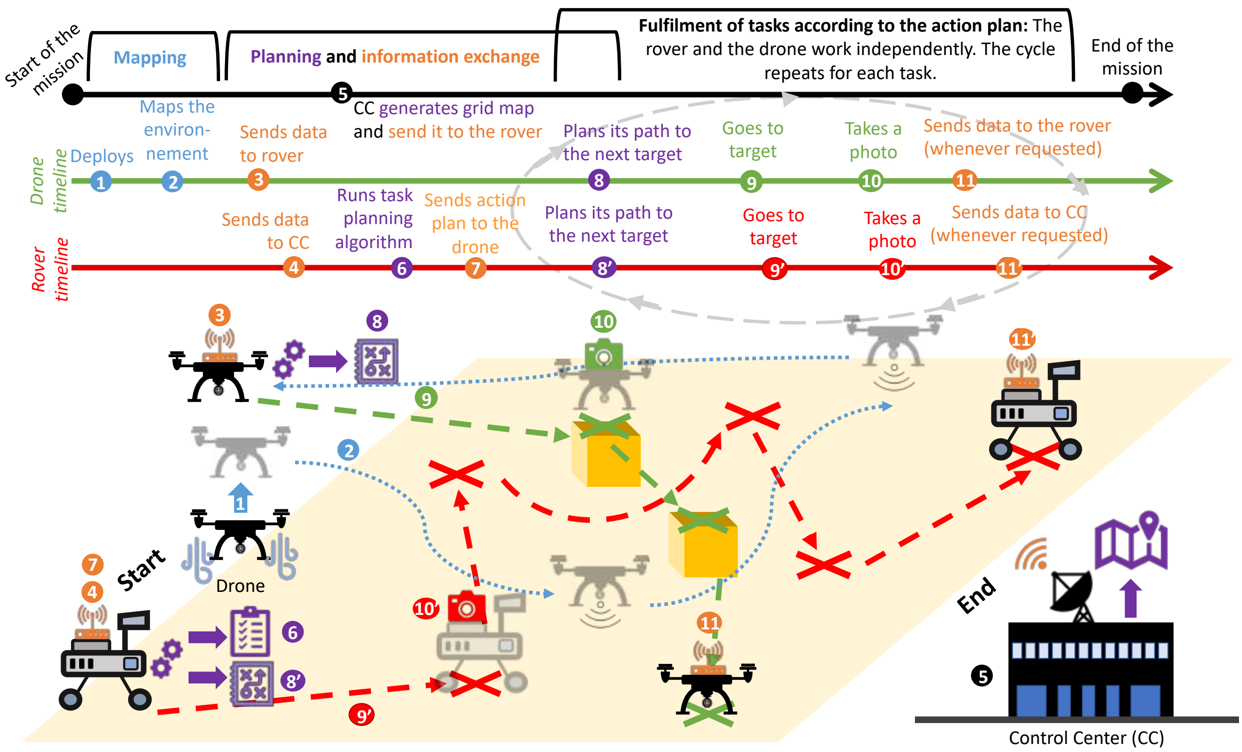


Figure 16: Concept of Operations of CORODRO

2.7.1 Safety Measures

- Drone regulation: the drone regulation requires the presence of, at least, one person for coordination of special aerial activities. The applicant must declare to be familiar with the rules and regulations in the ordinance of the OPEN A3 category. They must assure that these aerial activities do not conflict with current national laws and regulations regarding the protection of data privacy, as well as regarding the protection of military installations. They also need to take necessary actions so that no third parties on ground are endangered by aerial activities under their control. At the same time, before each drone flight, we must inform the operators of the nearby airport.
- E-stop: the drone and the rover are equipped with a E-stop function in order to stop them in case of emergency.
- Hardware spare components: we will bring with us some spare parts in case of failure or malfunctioning of sensors or batteries.

2.7.2 Operations and Test Procedures

During the VFC, we would like to perform three main tests in nominal conditions:

- Drone mapping: the drone will create a map of the environment.
- Rover mission: the rover will acquire the map, run the task planning code, send the plan to the drone and start its mission. The mission consists in moving in the mapped environment, while reaching the points of interest (POIs), and reading their

AR-tags. At the same time it should be able to avoid obstacles and, if necessary, update the map.

- Drone mission: the drone will move to the different waypoints not reached by the rover reading their AR-tags.

At the same time, we would like to test one contingency situation:

- Rover mission without known map: in this case the rover will know the most likely positions of POIs, with a certain level of confidence. The position is given to the rover by the control center operators. They will see what the rover is seeing thanks to the rover camera and they can move the system toward a ArTag. In that case the rover, has to move, map and reach as many POIs as possible autonomously.

It is possible that we would have to deal with wind during our testing. If that is the case, we would need to consider as contingency situation the "flying the drone with wind". This specific testing is not relevant for the lunar scenario, but it is a contingency scenario that we may have on Earth while testing. The envisioned testing during the Field Campaign are detailed in Table 20.

Testing Objective	Number	Hardware	Algorithm	Expected results	Field[m]	Time
Nominal Operations						
Drone mapping	1.1 1.2	Drone Drone	Localization and Real-Time Mapping Path Planning	Obtain a DEM Map Map the environment	10*10(± 2)	3h
Rover mission	2.1 2.2 2.3	Rover Rover Rover	Path Planning Task Planning Localization	Move and accomplish tasks in a known environment	10*10(± 2)	3h
Drone mission	3.1 3.2 3.3	Drone Drone Drone	Path Planning Task Planning Localization	Move and accomplish tasks in a known environment	10*10(± 2)	3h
Contingency Operations						
Move the rover in unknown map	4.1 4.2 4.3	Rover Rover Rover	Path Planning Task Planning Localization and Real-Time Mapping	Move the rover toward its objectives The relative position of the targets is known (e.g. North, North-East)	10*10(± 2)	3h

Table 20: Envisioned test for the IGLUNA Campaign

To be able to accomplish these tests, we linked them with the development plan in Section 2.3 (Figure 8). The basic capabilities are linked to the nominal operations, while the contingency operations are linked to the complex capabilities or Earth atmospheric conditions. During the Field Campaign, we will work on the field setting up our own control station. Therefore, we detailed our ConOps as:

- Mapping of the drone on Table 22,
- Rover mission on Table 26,
- Drone mission on Table 25,
- Beginning and end of the day tasks on Table 23,
- Set-up of the field on Table 27,
- Control Room ConOps on Table 24.

We also detailed what to do in case of systems with low battery, no global map, and slightly and heavily damaged system issues in Table 28.

During the Virtual Field Campaign we will have at least:

- One drone pilot on the field

- One rover and drone operator that will place the systems or be in charge of changing the batteries.
- 1 operator at the control station.

All operations start from these assumptions:

- The tested ConOps are linked to the testing capabilities on Table 20.
- The points of interest will be brought by us (card-boxes with ArTAG bundles on the sides).
- Drone have a E-stop that can be activated remotely by the pilot.
- The rover can be stopped remotely by the operators on the field and the control center.
- The ”zero position” is at the bottom-left of terrain setting. In simulate a recharging station.

Shipping ConOps	Description	Time	Number of operators needed
Packing	Safely pack all the material we want to take to the FC	1 hour	3-4
Loading	Load the material in the car of one of the participants	30 minutes	3-4
Travel	Travel to ONERA	1 hour	1-5
Arriving	Arrive to CNES and put the packages at the designated storage place.	30 minutes	1-5
Total	Shipping mission	3 hours	1 to 5

Table 21: Shipping mission

Mapping Steps	Description	Time	Number of operators needed
Settings	Set the drone in the zero position and turn it on Set the computer and the operator on the field	10 min	2-3
Communication check	”Ping” the drone with the field computer to verify communication	10 min	2
Position of the points of interest	Verify the positions of the points of interest for later evaluation	20 min	2
Start	Start of the drone propellers	1 min	1
Lift-off	Perform lift-off	5 sec	-
Target altitude	Reach target altitude of 3m	2 sec	-

Mapping phase	Run Task Planning, Path Planning, and Localization and Real-Time Mapping algorithms	5 min	-
”Zero position”	Reach end position ”zero pos”	10 sec	-
Landing	Perform landing	5 sec	-
Stop propellers	Stop propellers	10 sec	-
Data transfer	Drone sends DEM map and grid map to the CC	2 min	1
Stop	Complete stop	1 min	-
Recharge	Operator 2 removes the drone from the field to recharge or replace its batteries	60 min	1
Control Center Tasks	Description	Time	Number of operators needed
Quality/Storage	Operator 1 checks the quality of the map and stores the telemetry data	30 min	1
Go to next test	If the map is good enough we can head to the next test (green light to the operators on the field)	2 min	1
Restart test	If the map is bad, the test should restart: send new mapping parameters to the operators on the field and red flag	2 min	1
Total	Mapping mission	214 min (3 h 34 min)	1

Table 22: Mapping

End of the day tasks	What to do at the end of each testing day	Time	Number of operators needed
Take photos	Take photos of the relative position of the objects (drone, rover, points of interest) to be able to reconstruct the field.	30 min	2
Clean the field	Remove all objects (drone, rover, points of interest) from the field	20 min	2
Beginning of the day tasks	What to do at the beginning of each testing day	Time	Number of operators needed
Recreate the field	Recreate the field by taking out all the objects from the storage place.	30 min	2

Table 23: End/Beginning of the day tasks

Field Control Center ConOps	Description	Time	Number of operators needed
Arrival	Set-up the control center	20-25 min	2
Testing 1	Record the testing while waiting for the data to check (depends on the test)	Testing time	2
Testing 2	While the testing is taking place, the battery telemetry will be received	Testing time	2
Receiving of the data	When the data is received, an end check will be performed to see if the set performances are being reached	90-120 min	1
Eventual re-testing	Communicate to the field if the test needs to be repeated	1-2 min	1
Total	Control Center ConOps	147 min (2 h 45 min)	2

Table 24: Control Room ConOps

Drone mission steps	Description	Time	Number of operators needed
Settings	Set the drone on the zero position and turn it on	10 min	2
Communication check	”Ping” the drone with the field computer to verify communication	3 min	2

Receive data	Receive data from the rover (grid map)	3 min	-
Path Planning algorithm	Run Path Planning algorithm Assumptions during FC: No obstacles at 3 m height Keep a constant flight altitude	3 min	-
Start of the mission	Drone moves to target Takes a picture Moves to next target	3 min x 5 targets = 15 min	-
Landing	Perform landing	2 min	-
Stop	Complete stop	1 min	-
Data processing	Drone computes the DEM map and the grid map	3 min	1
Recharge	Operator 2 removes the drone from the field to recharge its batteries	60 min	1
Total	Drone mission	130 min (2 h 10 min)	2

Table 25: Drone mission ConOps

Rover mission steps	Description	Time	Number of operators needed
Settings	Set the rover on the "zero position" and turn it on	20 min	2
Communication check	"Ping" the drone to verify communication Repeat from the CC	10 min	2
Send map to rover	The rover receives the map from the CC	10 min	1
Best task division	The rover evaluates the best task division	15 min	-
Send plan to drone	The rover sends its plan to the drone	15 min	-
Start of the mission	The rover starts its mission: Compute the global map Move to target while running obstacle avoidance algorithm (local path planer) Reach the target and take a photo	6 min x 10 targets = 60 min	-
Stop	The rover stops at the "zero position"	10 min	-
Data transfer	Operator 1 transfers the data to the computer	1 min	1

Recharge	Operator 2 charges the rover batteries and put the old ones to recharge	60 min of recharge + 10 min to change = 70 min	1
Control Center operator 1	Checks the overall performances of the system: Environment well localized Taking the right pictures Obstacle avoidance success Best global path possible Targets reached Map accurate enough and updated by the rover, which recognized new potential sites of interest	60-90 min	1
Control Center Communication of the test fail/pass	Operator 1 tells if the mission is successful or not and asks to re-test if needed	10 min	1
Total	Rover mission	319 min (5 h 18 min)	2

Table 26: Rover mission ConOps

	Total energy budget [KJ]
Mapping of the drone	-Mapping D435i config: 252.186 KJ -Mapping LiDAR config: 251.946 KJ Total:252.186 KJ
Drone mission	-Start + Lift off + Target altitude: 56.321KJ -Stand by: 1.567 KJ -Landing: 4.18 KJ Total used: 62.068 KJ
Rover mission	-Send map to rover + Best task division + Send to drone: 76.920 K J -Compute the global map (receive data): 0.0641 KJ -Move to target while running obstacle avoidance algorithm: 7.764 KJ -Reach the target and take a photo: 4.404 KJ Total used: 89.1X10(targets): 891.152KJ

Table 30: Total Energy budget per ConOps

Before testing set-up	Description	Time	Number of operators needed
Checklist	Check the battery levels of the drone and the rover before going to the field	30 min	1
Packing	Pack all the material (drone, rover, reparation kit, computer, points of interest)	15 min	2
Transportation	Transport the material to the field (from storage place to field)	15 min	2
Set-up	Set the points of interest on the field Map their position to be able to repeat the experiment Set up the CC Set the zero point where the systems start Set the recharge point	90 min	2
Checklist	Check communication between computer in the field and computer in the Control Center, communication between computer in the field and drone and rover Rover and drone respond to a basic ”go to” command	90 min	2
Total	Set-up complete	240 min (4 hours)	1 to 2

Table 27: Set-up

Contingency Situation 1 - Systems with low batteries	Description	Time	Number of operators needed
Stop the system	Basic capabilities: Signal from Control Center Ideal and complex: Internal software shutdowns itself	5 min	1
Inspection	Operator 2 on the field brings the system for inspection	10 min	1
Possible problem 1 - Battery level	Replace the rover batteries (the mission continues) Recharge the used ones (2 h for the rover, 1 h for the drone)	10 min	1
Possible problem 2 - Software issue	Identify the problem and try to solve it If not, switch to basic capabilities (signal from CC)	90 min	1
Contingency Situation 2 - No global map	Description	Time	Number of operators needed
Limit objectives	Run only the local path planner We can use the (known) relative position of the targets	10 min (to reset)	1
Contingency Situation 3 - Slightly Damaged System	Description	Time	Number of operators needed
Stop	Stop the testing	1 min	2
Removal	Remove the system from the field and bring it to the workbench for inspection	15 min	2
Reparation	Try to find the problem and use the reparation kit	120 min	2
Contingency Situation 4 - Heavily Damaged System	Description	Time	Number of operators needed
Stop	Stop the testing	1 min	2
Removal	Remove the system from the field	5 min	2
Continue or abandon	Continue testing with the other system if possible If both systems are heavily damaged (no repair possible) go back to ISAE and use the back-up systems.	-	2

Table 28: Contingency situations

Virtual Field Campaign	Description	Time	Number of operators needed
Set-up	Set-up the computer	10 min	1
Press Conference	Present the project and answer questions from the journalists	1 min + 10 min	2
Project Shows	Show the pre-recorded video, explain the project, broadcast tests done at ISAE and answer questions from the audience (YouTube chat)	40 min	2
Total	Virtual Field Campaign ConOps	1 h	2

Table 29: Virtual Field Campaign ConOps

	Total data budget [Mb/s]
Mapping of the drone	-Tracking Camera: 24.4 Mb/s -Camera D435i + IMU + Others: 8.64 Mb/s Total: 33.04 Mb/s
Drone mission	-Drone mission : 24.48 Mb/s -Drone standby + Drone landing + Drone recharging: 25.572 Mb/s Total: 50.052 Mb/s
Rover mission	-Rover tools: 11.292 Mb/s -Rover modes: 47.64 Mb/s Total: 58.932 Mb/s

Table 31: Total data budget per ConOps

2.8 Virtual Field Campaign Timeline

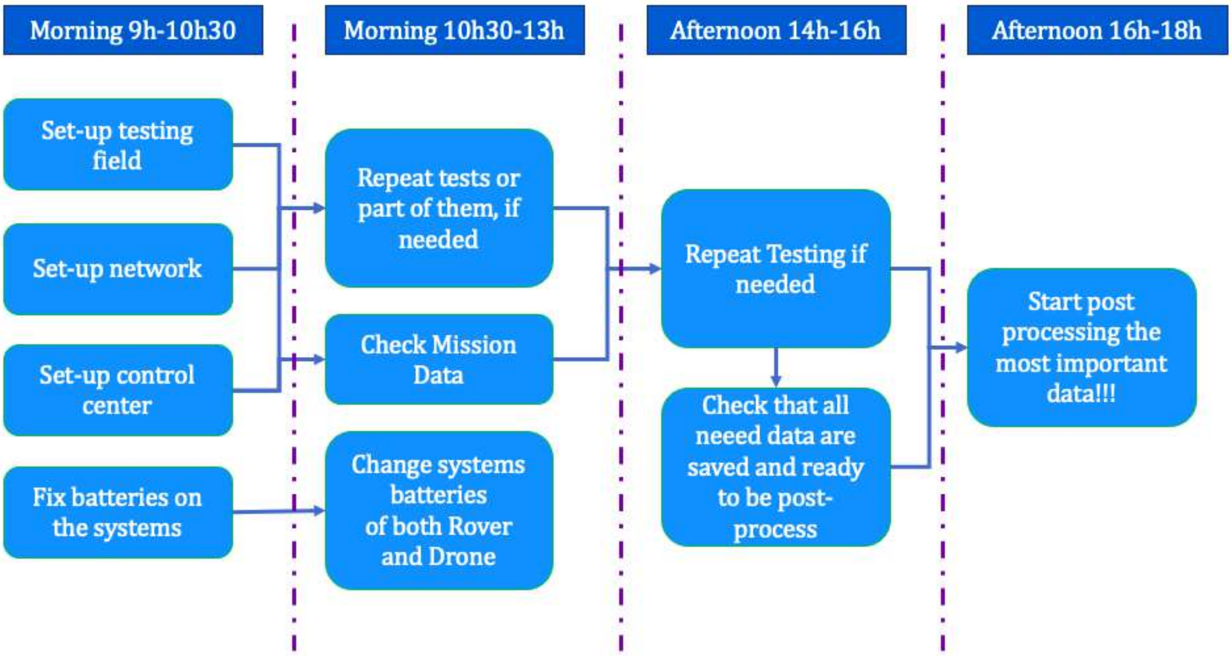


Figure 17: VFC Timeline for one day of testing

2.9 Project Interfaces

2.9.1 Mechanical, Power and Data Interfaces

No mechanical interface with other teams is envisioned for this project. Our systems are provided by our partners, Table 11. Therefore, they are already assembled and defined in hardware and computational power. Our interfaces during the VFC can be summarize as:

- WiFi as communication network provided by the DISC department for the field testing.
- Power sockets on the Field to recharge our batteries.
  - Drone: 1h to charge the batteries at 12 V and 5 Ah.
  - Computer: 12 V should suffice.
- Power socket to connect the computer running ROS in the control center.

Number of batteries brought and needed by day if we perform 3 tests per day:

- Drone: 2-3 batteries per day. 3 batteries brought. (on recharging while the drone is testing).
- Rover: Lower part : 1 battery needed per day. 2 batteries brought.
- Rover: Upper part : 2 Batteries needed per day. 3-4 Batteries brought.

2.10 Project Overview Summary

2.10.1 List of Components

The list of the main components of this project is summarized in Table 32.

Leo Rover	- Cost: USD 2399 - We got directly the part as sponsoring in kind by the DISC team - Sponsored: Yes - Supplier: Lent by ISAE-supaero - Expected lead time: Overall project - up to end of FC - Number: 1 unit - Weight: Around 7.8 kg - Reason: Already available testing platform, open source wireless all-terrain rover - Status: Delivered - Payload : Depth camera, LiDAR, IMU, wheel encoders (between EUR 600 - 2500)
Drone	- Cost: EUR 2900 - We got directly the part as sponsoring in kind by the DISC team - Sponsored: Yes - Supplier: Custom made by ISAE-SUPAERO - Expected lead time: Overall project - up to end of FC - Number: 1 unit - Weight: Around 2 kg - Reason: Experimental outdoor drone, custom made to exactly what is needed. It should resist well the wind - Status: Delivered - Payload: Tracking Camera, Depth Camera, LiDAR (between EUR 600 - 2500)
RPLIDAR-A1	- Cost: Around EUR 500 - We got directly the part as sponsoring in kind by the DISC team - Sponsored: Yes - Seller: Unknown - Expected lead time: Overall project - up to end of FC - Number: 1 for the rover and 1 for the drone - Weight: 170 g - Reason: Best price/performance available LiDAR - Status: Already at ISAE-SUPAERO
D435i	- Cost: EUR 259 - We got directly the part as sponsoring in kind by the DISC team - Sponsored: Yes - Seller: Intel Depth Camera - Expected lead time: Overall project - up to end of FC - Number: 1 for the rover and 1 for the drone - Weight: 72 g - Reason: Set of cameras already available at ISAE-SUPAERO - Status: Already at ISAE-SUPAERO
T265	- Cost: EUR 229 - We got directly the part as sponsoring in kind by the DISC team - Sponsored: Yes - Seller: Intel Track Camera - Expected lead time: Overall project - up to end of FC - Number: 1 for the drone - Weight: 55 g - Status: Already at ISAE-SUPAERO

Table 32: List of main project components

2.10.2 Project Overview Summary

The two main hardware subsystems that will be used during our project are a drone and a rover, Figure 18. The dimensions of the systems are respectively:

- 0.067 m^3 (0.42 x0.40 x0.40) m^3 for the rover,
- and 0.088 m^3 (0.57x0.57x0.27) m^3 for the drone.

Those systems are provided to us by our sponsors and they are considered as constraints, Table 11. The following sections will analyze more in detail our hardware and its characteristics and performances.

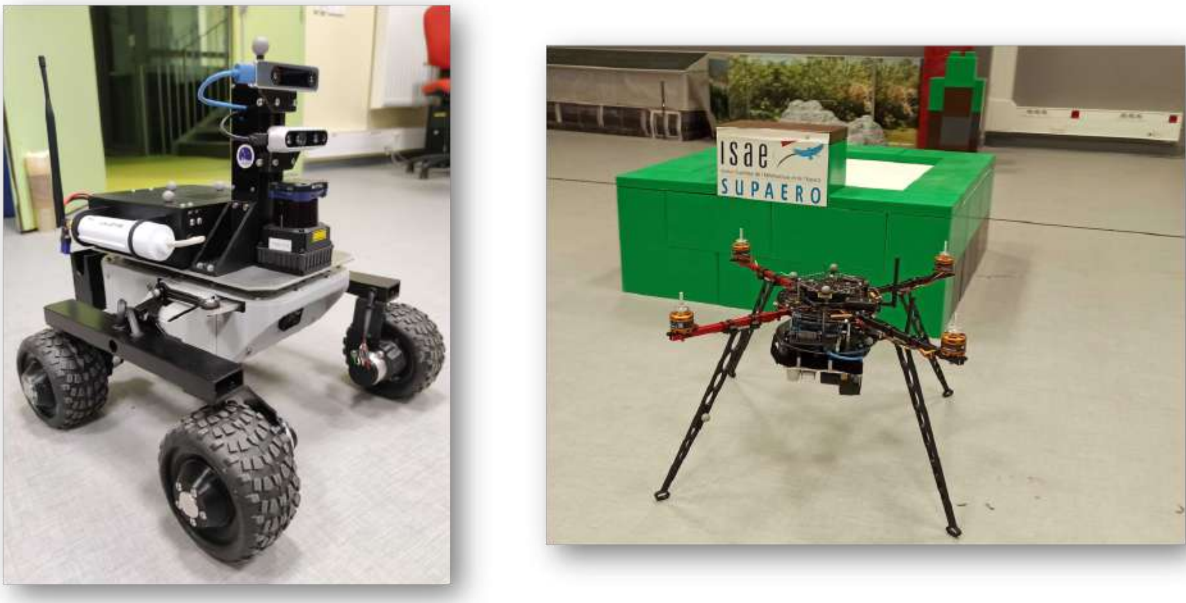


Figure 18: Rover and drone that will be used during the FC

2.11 Project Design

2.11.1 Mechanical Design

As explained in the previous section, we work with two main subsystems: a rover and a drone. The rover is a commercial platform [10], while the DISC department of ISAE-SUPAERO has assembled the drone. The choice of the sensors equipped on the drone is explained in Section 2.2. The sensors for the rover were already mounted on the platform to perform experiments similar to ours, so we kept the same winning set-up.

Drone components	Subsystem	Reference	Specific characteristics
Cameras	Depth Camera	Intel D435i	Maximum frame rate: 90 fps
	Tracking Camera	Intel T265	Maximum frame rate: 120 fps
Communication Transfer Speed : $\approx$ 200 Mbps	Radio Receptor	Futaba R6208 SB	Frequency: 2.4 GHz
	WiFi System	Odroid WiFi Module	Frequency: 2.4 GHz
Computer	Computer	ODROID XU4	
Electronic Avionics	Flight Controller	3DR Pixhawk 3	
	Electronic Speed Controllers	BL-C trl V 3.0	
Motors	4 Motors		
Power System	Battery	Lipo 4s	Capacity: 3700 mAh

Table 33: List of equipment: Drone

Rover components	Subsystem	Reference	Specific characteristics
Sensors	Leo Rover Camera	HOKUYO UST-20LX	Maximum frame rate: 40 fps Mpx
	Depth Sense Camera	Intel D435i	Maximum frame rate: 90 fps
	Tracking Camera	Intel T265	Maximum frame rate: 120 fps
	Lidar	HOKUYO UTM-30 LX-EW	Maximum frame rate: 50 fps
Communication	WiFi Internal RPi Antenna		WiFi 2.4 GHz +5 GHz on internal RPi antennas
	Leo Rover WiFi Access Point Antenna		WiFi 2.4 GHz
	Leo Rover WiFi System		
	Upper Part Rover		
Computer	Computer Leo Rover	Raspberry Pi 3B+	Clock Frequency: 0.4 GHz
	Computer Upper Part Rover	ODROID XU4	
Controller	Controller	Husarion CORE2-ROS	
Mobility System	4 Wheels 4 Tires		Diameter: 130 mm Material: rubber with foam insert (non pneumatic)
	4 Wheel Actuator	Buehler DC Motors	
	4 Wheel Encoder	Pololu Romi 12 CPR Magnetic Encoders	
Inertial Measurement Unit		SBG IG 500A-G5A2P1-P	
Power System	Battery Leo Rover	Battery Li-Ion with internal PCM	Capacity: 5000 mAh
	Battery Upper Part Rover	KUNZER multi pocket booster MPB 150	Capacity: 15000 mAh

Table 34: List of components: Rover

2.11.2 Mass Budget

In Table 35 and 36, we summarize the mass budget of our main systems.

Component	Mass
Cameras	0.3 kg
Communication	0.03 kg
Computer	0.1 kg
Electronics Avionics	0.1 kg
Motors	0.3 kg
Power System	0.4 kg
Structure Frame	0.6 kg
Total	1.8 kg

Table 35: Mass budget of the drone

Rover components and payload	Mass
Computer (Raspberry Pi)	0.05 kg
Controller (Husarion CORE2-ROS)	0.07 kg
Leo Camera	0.02 kg
Mobility System(Wheels and actuators)	1.3 kg
Communication Module	0.02 kg
Power System	0.5 kg
Structural Frame	4.5 kg
WiFi Module (payload)	0.2 kg
Sensors (Lidar, D435i, T265)	0.3 kg
Inertial Measurement Unit (payload)	0.04 kg
Power Booster(payload)	0.5 kg
Computer (Odroid XU4)	0.04 kg
Total	7.72 kg

Table 36: Mass budget of the rover

2.11.3 Electronics Design

The provided systems are already completely wired and the communication network is configured as WiFi-like network. The OBC of the drone is a Raspberry Pi, while the rover is equipped with a Raspberry Pi and an Ondroid, as listed in Section 2.11.

2.11.4 Power Budget

The power budgets per mode of the rover and the drone are presented in Tables 37, 38 and 39.

	Mapping_D435i Config	**Mission**	**Standby**	**Landing**	**Recharging**
Component	**Power Consumption**	**Power Consumption**	**Power Consumption**	**Power Consumption**	**Power Consumption**
T265	1.575 W	1.575 W	1.575 W	1.575 W	0
D435i	3.5 W	3.5 W	3.5 W	3..5 W	0 W
Communication	1.05 W	1.05 W	1.05 W	1.05 W	0 W
Computer	20 W	20 W	20 W	20 W	0 W
Electronics Avionics	4.5 W	4.5 W	0 W	0 W	0 W
Motors (cases shown below)					
Case 1: Indoor (measured)	540 W	540 W	0 W	540 W	0 W
Case 2: Outdoor (+50 % estimation)	810 W	810 W	0 W	810 W	0 W
Total Indoor	**570.62 W**	**570.62 W**	**26.12 W**	**566.12 W**	**0 W**
Total Outdoor	**840.62 W**	**840.62 W**	**26.12 W**	**836.12 W**	**0 W**

Table 37: Power budget of the drone

	SendData	Task Processing	Generate Path	Off
Component	**Power Consumption**	**Power Consumption**	**Power Consumption**	**Power Consumption**
Computer	1.15 W	1.15 W	1.15 W	0 W
Controller	2 W	2 W	2 W	0 W
Leo Camera	0 W	0 W	0 W	0 W
Mobility System	0 W	0 W	0 W	0 W
Network	1.9 W	1.9 W	1.9 W	0 W
DepthSense Cameras	0 W	0 W	0 W	0 W
T265	0 W	0 W	0 W	0 W
LiDAR	0 W	0 W	0 W	0 W
Inertial Measurement Unit IMU	0 W	0 W	0 W	0 W
WiFi Module	7 W	7 W	7 W	0 W
Rover Computer	20 W	20 W	20 W	0 W
Total	**32.05 W**	**32.05 W**	**32.05 W**	**0 W**

Table 38: Power budget of the rover (Part A)

	Receive Data	Take Picture	Safe Mode	Driving
Components	**Power Consumption**	**Power Consumption**	**Power Consumption**	**Power Consumption**
Computer	1.15 W	1.15 W	1.15 W	1.15 W
Controller	2 W	2 W	0 W	2 W
Leo Camera	0 W	0.75 W	0 W	0.75 W
Mobility System	0 W	0 W	0 W	28 W
Network	1.9 W	1.9 W	1.9 W	1.9 W
DepthSense Cameras	0 W	3.5 W	3.5 W	3.5 W
T265	0 W	1.575 W	0 W	1.575 W
LiDAR	0 W	0 W	0 W	2.7 W
Inertial Measurement Unit IMU	0 W	0.4 W	0 W	0.4 W
WiFi Module	7 W	7 W	7 W	7 W
Rover Computer	20 W	20 W	0 W	20 W
Total	**32.0 W**	**38.3 W**	**13.55 W**	**69 W**

Table 39: Power budget of the rover (Part B)

2.11.5 Data Budget

We have created a data budget for both the drone and the rover by monitoring the data output of the captors during tests, shown in Table 40 and Table 41. The LiDAR on the rover will be used only to detect obstacles that are the rover height. The tracking camera on the rover will perform visual odometry, while the depth camera will be used to take picture of the obstacles and POIs. The ”other category” in the data budget regroups topics that send low amount of data, but that are monitored throughout the overall mission:

- Rover: Battery level, odometry, localization information (during mission), GPS readings (to compare with our localization stack), computation time tag, transformation between the reference systems, camera info topic (information on the camera state and readings), LiDAR info topic, map information, commanded linear velocity and commanded angular velocity.
- Drone: battery level, tracking camera odometry, localization information (during mission), GPS readings (to compare with our localization stack), computation time tag, transformation between the reference systems, camera info topic (information on the camera state and readings), LiDAR info topic, map information, commanded linear velocity and commanded angular velocity.

Item	Data [Mb/s]	Margin [%]	Data with Margin [Mbps]
Tracking Camera (T265)	0.15	20	0.2
Camera D435i	6.0	20	7.2
IMU	0.1	20	0.12
Other	0.1	20	0.12

Table 40: Data budget of the drone

Item	Data [Mb/s]	Margin [%]	Data with Margin [Mbps]
Camera D435i	30.0	20	36.0
Tracking Camera (T265)	0.15	20	0.2
LiDAR	0.08	20	0.96
Rover camera	2.5	20	3.0
IMU	0.01	20	0.012
Other	0.1	20	0.12

Table 41: Data budget of the rover

To help us see how much data will be produced throughout all the parts of the project, we also made a data budget for each mode (table 42 and 43), to be able to know more precisely how much data is produced, as all sensors will not need to be always active. All the modes were created by the mission analysis team, to fit with the objectives of the VFC.

Mode	Data [Mb/s]	Margin [%]	Data + Margin [Mbps]	Transm rate [Mbps]
Drone mapping d435i	30.3	20	36.4	0.005
Drone mission	30.4	20	36.5	0.04
Drone standby	0.01	20	0.012	≈ 0
Drone landing	30.3	20	36.4	–
Drone recharging	0.1	20	0.12	≈ 0
Drone off	0	20	0	0

Table 42: Data budget of the drone in different modes

Mode	Data [Mb/s]	Margin [%]	Data + Margin [Mbps]	Transm rate [Mbps]
Rover send data	6.6	20	6.92	0.04
Rover generate path	6.6	20	6.92	0.03
Rover off	0	20	0	0
Rover receive data	6.6	20	6.92	0.08
Rover take picture	9.1	20	9.92	0.06
Rover safe mode	0.1	20	0.12	≈ 0
Rover driving	9.1	20	9.92	0.172

Table 43: Data budget of the rover in different modes

2.12 Data Management

2.12.1 Data Management Concept

The amount of data produced by the sensors of the drone for the Localization and Real-Time Mapping algorithms is quite important. At first, we were not sure to be able to send the overall data set through the SI provided network, so we came up with the idea of a USB drive to transfer data to a computation station on site and downsize them. However, after a sampling downsize we were finally able to create the map directly on the drone, transform it in the required occupancy 2D map and send it through the WiFi network to the computers at the control center and to the rover. All the telemetry will be monitored by the control center through Linux terminal and rviz, ROS visualization tool [2].

During the preparation of the RR, we come out with an idea to downsize the important data in case of anomalies in the network ,as suggest by Airbus. In case of bandwidth issue, if we can't manage to send all the data we want, we can chose to reduce manually the bandwidth and message rate of less vital topics, such as cameras. In case of emergency, we will execute these actions :

- Stop sending data from all camera, except for the tracking camera, with one picture every 10 seconds. This will be done by executing a simple python script that can be used at anytime, in case of issue.
- Sending vital information such as odometry, pose of the system (localization in map) and battery charge at a rate of 1 message per second.

The contingency situations is entered in the execution layer of our task planner, Section 2.17. It activates if there is a communication anomaly or if triggered by the control center with a specific topic the systems listen to.

2.13 Data to monitor

This section shows, for Path Planning (Table 44), Localization and Real-Time Mapping (Table 45) and Task Planning (Table 46), what are the data collected during the tests at the Field Campaign, and why these data are collected.

[2] http://wiki.ros.org/rviz

2.13.1 Path Planning

What Data is collected	Why that Data is collected
AR Tags identification	It is required to assist the drone with localization and orientation during the first phase of the mission (mapping), and then during the exploration phase to recognize the point of interest.
Movement in 3 dimensions from tracking camera	To reconstruct the trajectory of the object (and, during the testing in the lab, to check if the position given by the tracking camera is the absolute one)
Global Path re-planning	How many time did we have to re-plan the global path because of a discrepancy of more than 30% between the obstacles from the drone map and the obstacles seen by the rover. (the 30% is evaluated comparing rover and drone generated map)
Local path plan and global path plan (informations in x,y directions for the rover)	To check the discrepancy between the global and local path, it is needed to evaluate how good are the used motion planner. Thank to this information, we can evaluate path smoothness, clearance from the obstacles and length of the path.
Time to compute path	To evaluate the computational cost of the path calculation.

Table 44: Data collection for Path Planning.

2.13.2 Localization and Real-Time Mapping

What Data is collected	Why that Data is collected
Drone and Rover Position data	Frequency : 12time/min Knowing if the system is behaving correctly
Drone and Rover Battery Level	Frequency : 5time/min Knowing if the system is operational
Drone and Rover D435i Camera Images	Frequency : Each time the system recognize s a point of interest. It is the mission goal : Getting pictures of points of interests on the field.
Drone 3D and 2D map generated	Frequency : 1 time at the end of the mapping process. Human analysis of the map to assess the success or the failure of the mapping phase.
Drone and Rover Commanded linear and angular velocities	Frequency : as max as possible. Monitoring the behavior of the system
Rover 2D updated map (The rover updates the map generated by the drone.)	Frequency : 3time/min. It's the goal mission : generating a map of the environment as accurately as possible.

Table 45: Data collection for Localization and Real-Time Mapping.

2.13.3 Task Planning

What Data is collected	Why that Data is collected
Plan Integrity	Can the planner find a plan to touch all the waypoints
Plan Feasibility	Can the systems perform the plan (check the behavior of the system and if it is coded in the right way)
Task Planning interfaces	Did the planner launch the right algorithms of Localization and Real-Time Mapping and Path planning at the right moment?
Computational Cost	How long does it take to compute a plan?
Plan Re-evaluation	How many times did the system re-evaluate the path?
Available Resources	How much battery did we consume to perform the plan? The objective is to correlate the amount of battery with the driving speed and the distance covered

Table 46: Data collection for Path Planning.

2.14 Maximum transmission possible

The maximum transmission possible for the drone and rover depends on what the WiFi network can handle. As we are simulating a lunar environment, the WiFi needs to operate on is 0.5 Mbps. The data registered in the table of data budget are not all sent thought the WiFi, some of them are processed internally by the robots.
The Tables 44 to 46 show every data that are actually going to be sent to the WiFi toward

either the Control Center or the rover.
Here is an example of the computation that has been made for to make sure that the effective transmission of a mode is below 0.5 Mbps for each mode of the rover and the drone.
The drone D435i camera images data budget: is of 36.5 Mbps.
We send only one image each time we recognize an Artag during the drone mission. The rate of the camera is 30 Hz which means that the camera generates 30 images per second. So, an image represents $36.5Mbps/30 = 1.2Mb$. As we have about 6 Artags to detect at maximum as objectives for the drone so that represents 6 images that are going to be sent throughout the drone mission phase. The drone mission last around 3 minutes, so the D435i image data budget to be transmitted is 6 images of 1.2 Mb in 3 min. That represents 0.04 Mbps.

In conclusion, by computing the transmission rate for each mode, a check has been be made to show that the effective transmission of a mode is below 0.5 Mbps. These results are presented in Tables 43 and 42.

2.15 Localization and Real-Time Mapping

One of the main challenges the CORODRO team has to tackle is to move and localize in an unknown environment. Not only the topographical aspect of the terrain is unknown, but also the field location is set to be far beyond the reach of humankind civilization. Moreover, even if technological achievements such as the GPS coverage should be available on Mount Pilatus, we act as if it is nonexistent. In that extreme environment, the CORODRO team must provide a reliable autonomous navigation system. The navigation system includes a topographic map of the ground coupled with a positioning system. Intelligible for both humans and their machines, that map represents a piece of sophisticated information. Indeed, it enables the capability for an entity to simultaneously localize itself and be precisely aware of its environment.

Most of the modern SLAM algorithms share the same structure that is explained step by step just below. Our algorithms will fit that description and will be inspired from popular SLAM algorithms.

- Step 1: Estimation and scan:
  - 1.a) The robot estimates its position thanks to an odometry sensor (e.g. IMU or VIO).
  - 1.b) In the meantime, it scans its environment with a depth sensor (e.g. Lidar or Depth Camera).
- Step 2: Optimization: To avoid heavy computational software routines, it is essential to filter the depth scan provided by the sensor.
- Step 3: Landmark extraction: As humans use unique elements that will be easily remembered in order to locate themselves (e.g. the monuments of Paris as intersection points of straight Haussmannian boulevards), our system will emulate a similar behaviour. Landmarks will not be monuments, but something that still can be catchy for the eyes of the robot. That could mean points with a high-gradient variation relative to a physical quantity (e.g. corner points). Those landmarks should also be frequently observable, as there is no utility in a landmark that will be seen only once by the robot.
- Step 4: Association and Localization: The algorithm will try to recognize current landmarks among the past stored landmarks. Thus, given the position of the recognized landmark, and its current distance and angle between the robot and the landmark, the algorithm will be able to retrieve the position of the robot. Knowing its position and its distance with all the new landmarks, it can calculate their position and store them in the landmarks database. Various methods are used. Among them, one can mention extended Kalman filter methods, particle filters.

Once the localization is done, we are able to merge it with a depth scan of the environment retrieved by the D435i camera. It will act as the 5th step of our algorithm: Step 5: DEM map generation: Depth scan will be reprojected thanks to the position to form a local DEM map. That local DEM map will be added to the global DEM map containing the previous local DEM map generated. The robot goes to step 1 until the global DEM map is complete. Every cell of the environment will be identified by its position and contains depth information if the robot has already scanned this area, or a default value if the robot has not come yet. The map will use the Cartesian (x,y) system, where (0,0) is the initial position of the robot, and the y axis the straight forward line when the robot is initializing.

The main goal of the drone's Localization and Real-Time Mapping algorithm is to perform a reconnaissance mission of the field in order to provide a full map of the testing environment. The main goal of the rover's Localization and Real-Time Mapping algorithm is to improve the provided map, if necessary, to create a robust obstacle avoidance system. Also, because the rover is able to update the map, it is a posteriori able to generate a map from zero, meaning that if the drone has encountered a problem, the rover is still able to operate alone on the field (but being necessarily less efficient).

2.15.1 Localization and Real-Time Mapping Algorithm for the Drone

2.15.1.1 Drone different evaluated configurations and Final chosen Configuration

Choosing the T265 camera to retrieve a position was determining. Indeed, the V_SLAM algorithm embedded in that said camera simplifies the process for the mapping objective. Tests have shown that the V_SLAM algorithm is providing a reliable position, but not a reliable DEM map. Our idea is to use the position provided by the T265, but not its map. We use in addition an adapted depth sensor to generate the DEM map (D435i or LiDAR). Thus, we worked on two configurations : 1) T265 + D435i 2) T265 + LiDAR Software development was made considering those two configurations possible: one with the LiDAR as the depth sensor, and one with the depth camera D435i as the depth sensor. The two configurations were evaluated (see test section 2.20.3) and the final configuration chosen was the D435i + T265 according to the comparison of the tests results.

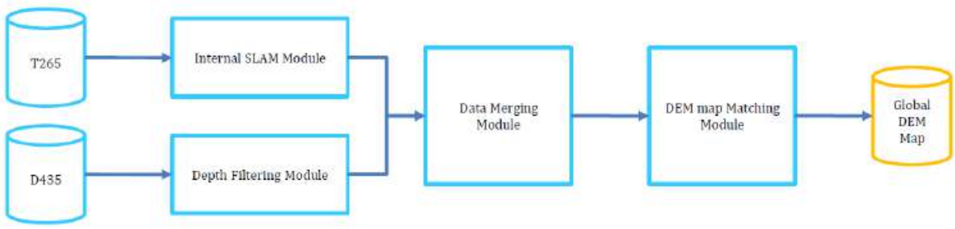


Figure 19: Map generation algorithm for the drone

2.15.1.2 Algorithm Architecture Figure 19 shows the block diagram of the envisioned map generation algorithm for the drone. The software has four main modules, Figure 20.

- Internal SLAM Module: It provides the position of the drone thanks to the V_SLAM algorithm embedded in the T265. It executes the steps 1, 3 and 4.
- Depth Filtering Module: It filters the depth readings in order to ease the computational cost of the DEM map generation. It executes the step 2.
- Data Merging Module: It uses the position and the depth data to generate a DEM map of the drone's local environment. It executes the first part of the step 5.
- DEM Map Matching Module: It integrates the generated local DEM map to the global DEM map. It executes the second part of the step 5.

A general description of the modules in terms of programming languages, internal and external interfaces can be find below in Table 47. The Localization and Real-Time Mapping algorithms use C++ as the main coding language. The software will be then interfaced with the hardware through ROS topics, services and action libraries.

Module	Description	Coding Language	Internal Interfaces	External Interfaces
Internal SLAM Module	It provides the position of the drone thanks to the V_SLAM algorithm embedded in the T265. It executes the steps 1,3 and 4.	Unknown (Intel Property)	INPUT: -T265 sensor data OUTPUT: -Position of the drone	None
Depth Filtering Module	It filters the depth camera readings to ease the computational cost of the DEM map generation. It executes the step 2.	C++/ROS/Python	INPUT: -D435 Depth Camera OUTPUT: -Depth map	None
Data Merging Module	It uses the position and the depth data to generate a DEM map of the drone's local environment. It executes the first part of the step 5.	C++/ROS	INPUTS: -Position -Depth map OUTPUTS: -Local DEM map	None
DEM Map Matching Module	It integrates the generated local DEM map to the global DEM map. It executes the second part of the step 5.	C++/ROS	INPUT: -Local DEM map OUTPUT: -Global DEM map	None
Converting Module	It converts the global DEM map into a 2D Cost-Map.	C++/ROS	INPUTS: -Global DEM map OUTPUTS: -2D cost-map	None

Table 47: Summary of the map generation software modules, description, coding language, interfaces for the drone

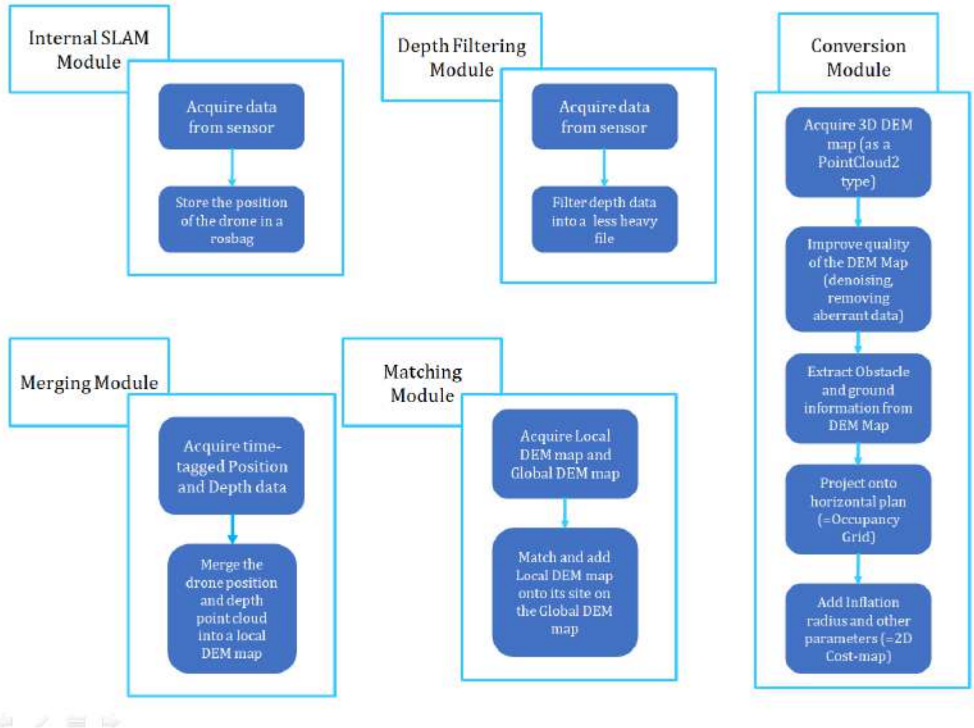


Figure 20: Modules of the map generation algorithm and high-level functions for the drone

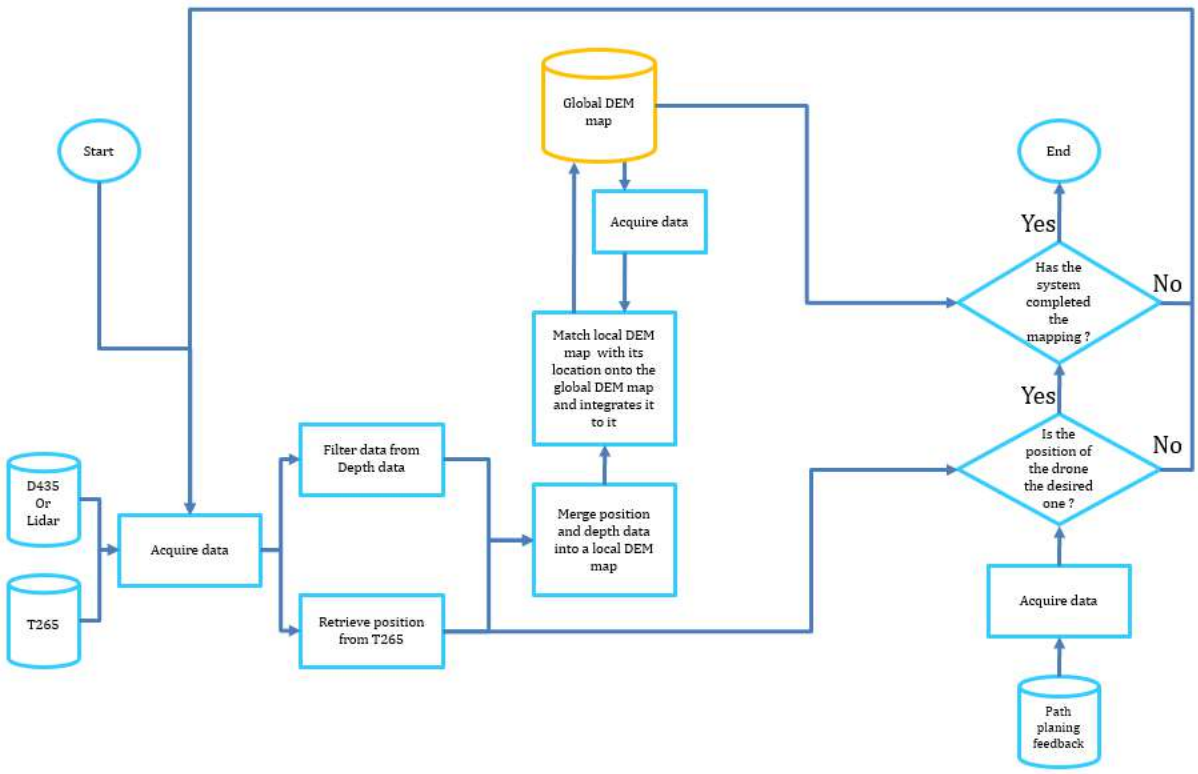


Figure 21: Localization and Real-Time Mapping process flow for the drone

2.15.1.3 Data Merging Module and Matching Module The Localization and Real-Time Mapping team was able to retrieve a full DEM map of the environment with a resolution of 3 cm, Figure 54.

2.15.1.4 Depth Filtering Module The Depth Filtering Module (2.15.1.2) aims to reduce the size and improve the quality of the depth data generated by the sensors. Its goal is to ensure that the Data Merging and Data Matching Module (2.15.1.2) are provided with easy to process (low computational cost) and high-quality data. The Depth Filtering Module has been implemented and contains a list of filters that optimize the overall execution of the software.

In the D435i configuration, at the very beginning of our experiments, we first reduced the frame rate of the camera. The tests have empirically shown that it is required to have at least a 70% overlap between two consecutive depth point clouds for the Localization and Real-Time algorithm to be efficient. Considering a velocity of 0.5 m/s, a flying height of 3 m, and the FOV of the D435i, we can have great quality of the depth images, even if we reduce the overall frame rate of the camera without losing information. Theoretically with the listed parameters, the rate can be dropped to about 1 Hz. In practice, we even lowered it to 0.5 Hz during indoor tests with very stable conditions. However, for the outside tests we did not reduce the rate lower than 4 or 5 Hz/s. This frequency can be manually set as a parameter of the camera itself before the beginning of the mission, or it can be implemented as a post-processing throttle filter. Then, on both configurations (Depth Camera and LiDAR):

- A Voxel Grid filter [21][20] is applied to reduce the size of the input depth point cloud. It downsamples the unorganized input point cloud into an organized point cloud structured as a voxel grid. The size of the voxel is the size of the aimed resolution for the final map, currently 0.03 m. It aims to reduce the computational power needed to process the data.
- Then, a Radius Outlier Removal [13] and a Statistical Outlier Removal [19] filters are executed. Those filters aim to remove aberrant data, points that are too ”scattered” to be considered as part of real obstacles (spurious readings).
- On the LiDAR configuration, a shadow filter is performed: we identified a problem where the LiDAR creates aberrant points at the edges of objects. That effect is commonly called ”shadow”.
- Finally, Statistical Denoising is applied. It improves a lot the quality of the data from the D435i, much more than the data from the LiDAR. It is because the LiDAR is already very accurate and generates less noisy point clouds than the Depth Camera.

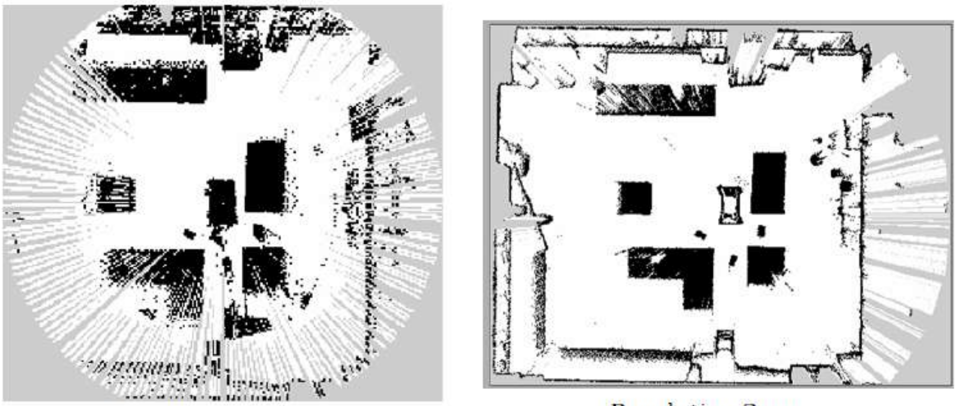


Figure 22: An example of 2D Occupancy Grid (LiDAR configuration) that has been optimized thanks to the Depth Filtering Module

2.15.1.5 Conversion Module: Transforming DEM Map into Cost-Map Task Planning and Path Planning systems require a 2D occupancy grid map, therefore we need to create a conversion module to translate our 3D DEM map to that format.

The conversion algorithm consists of five steps. Firstly, it improves the quality of the 3D DEM map with several tools similar to the Data Filtering Module. One can mention the detection and removal of aberrant data with some filters (radius outlier removal, statistical removal for example), and denoising (by a statistical approach on the neighbors of a chosen point). (Such processes are also used in the Data Filtering Module, to pre-process the data before it enters into the Data Merging Module).

The second step consists of extracting the smallest part of the DEM map that contains all the necessary information for identifying reachable and unreachable areas, as well as obstacles. To be more precise, in that step we remove all the points that are not strictly needed to identify obstacles, reachable or unreachable areas. The output of this step is still a 3D DEM map, but with one-layer thickness, which means that a column (x,y) contains a maximum of one point. It is done by a program called ”Grid map PCL”.

The third step consists of the detection of unreachable areas. An iterative process on each point analyses its closest neighbors in order to deduce if that point is reachable or not by the rover. It removes every point that is considered reachable. To us, an unreachable point means everything that represents a danger for the rover: an obstacle, a steep slope, a location where the rover might fall. Numerically, it consists of computing locally the height gradient in each cell and if that gradient is too high to be reachable by the rover the cell will be considered as an obstacle. Empirical tests showed that the rover cannot cross anything that is higher than 6cm with a slope greater than 90° or something that is higher than 15cm with a slope greater than 60°.

The last step consists of the projection of the remaining points onto a horizontal plane which is the 2D occupancy grid. The occupancy grid contains only three values: 0 (white) for unoccupied, 100 (black) for occupied, -1 (grey) for unknown.

The final output of the mapping task is that 2D occupancy grid written as an occupancy grid file format [11]. It contains information such as the coordinates (x,y) of the origin, the resolution, the height and width of the map, and a list that contains the values (0 or 100) of all the cells.

The first step's result is visible on Figure 24. The second step's result, along with a large selection of visual outputs, is visible in paragraph 2.20.3.3, in Figure 59.

Examples of final outputs of Occupancy Grid generation with Indoor and outdoor environment can be seen in section 2.20.3 on Figures 60 68 64 71.

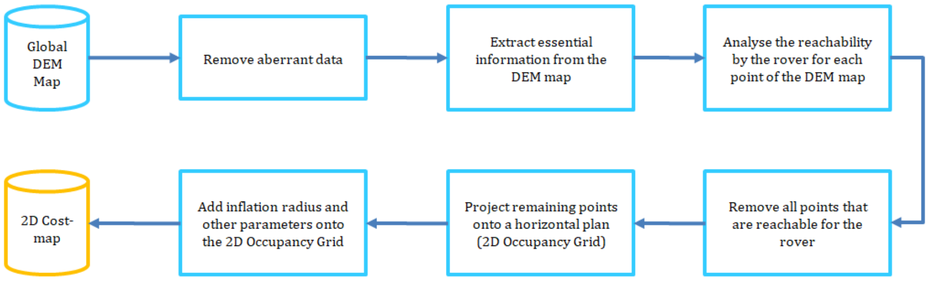


Figure 23: Conversion Module block diagram

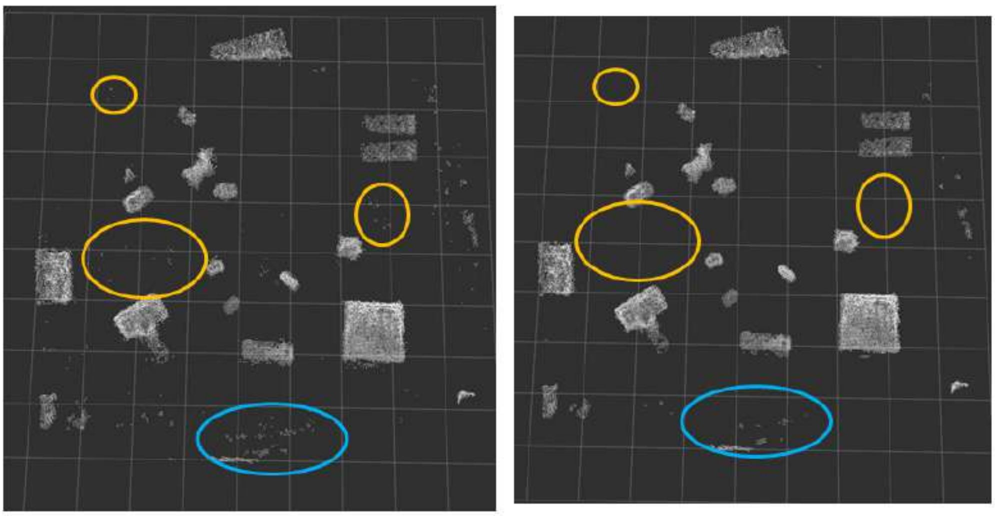


Figure 24: An example of Point Cloud obtained through Localization and Real-Time algorithm (D435i configuration) after the first step of the Conversion Module

2.15.1.6 Drone Mapping phase automation In May, the Task-Planning and Localization and Real-Time Mapping members collaborated to interface the algorithms related to the mapping phase of the drone with the state-machine that autonomously controls the drone. The state machine related to the drone mapping phase starts with the information of the lift-off of the drone coming from the state-machine of the drone path-planning, and ends with the generation of the 2D Occupancy grid of the environment. The activity diagram of the drone mapping state-machine is presented in Figure 25, along with the Class Diagram in Figure 26. Test of the drone mapping state machine has been concluded and its result is presented in Figure 74.

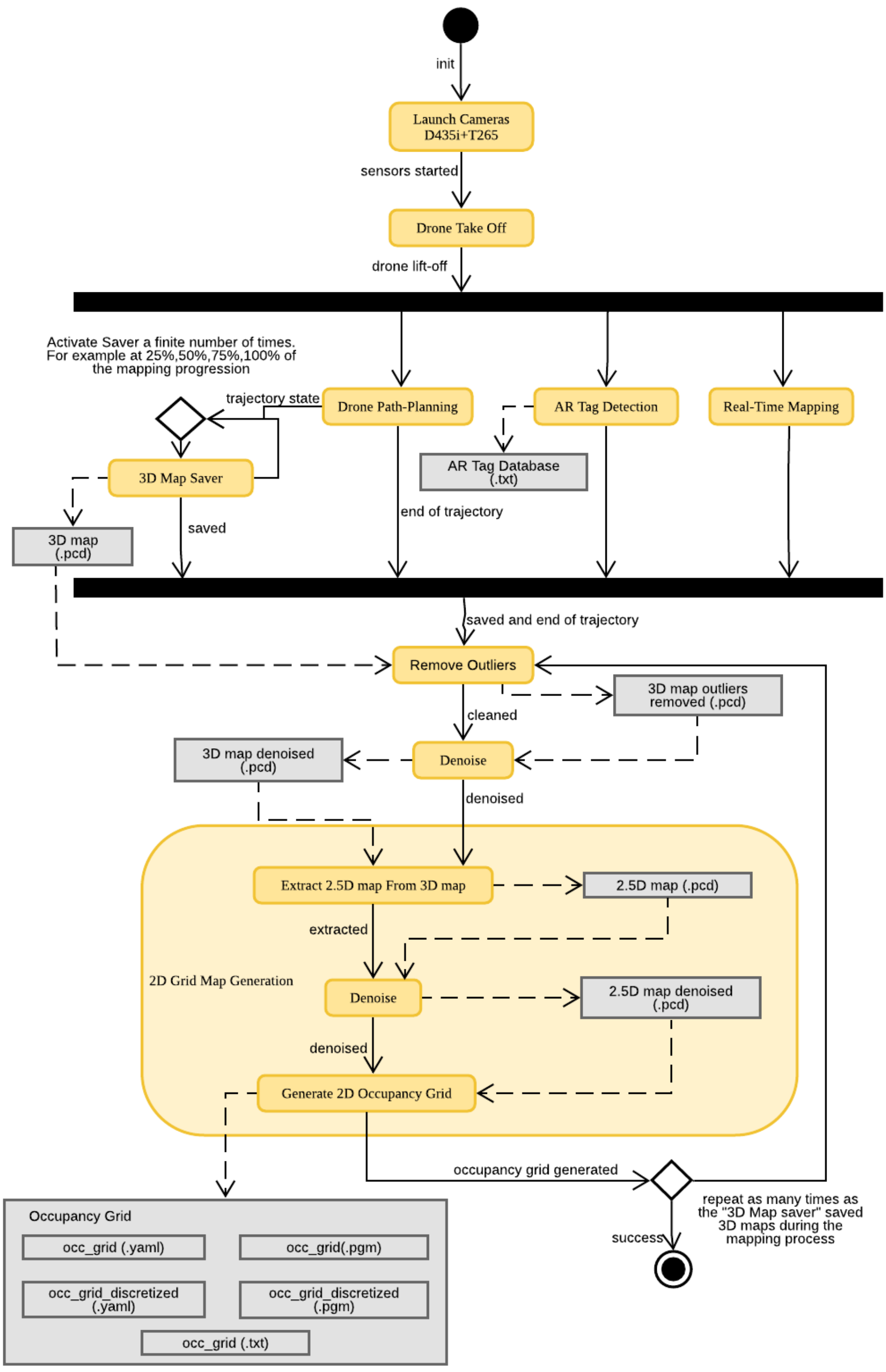


Figure 25: Activity Diagram of the Drone Mapping State-machine

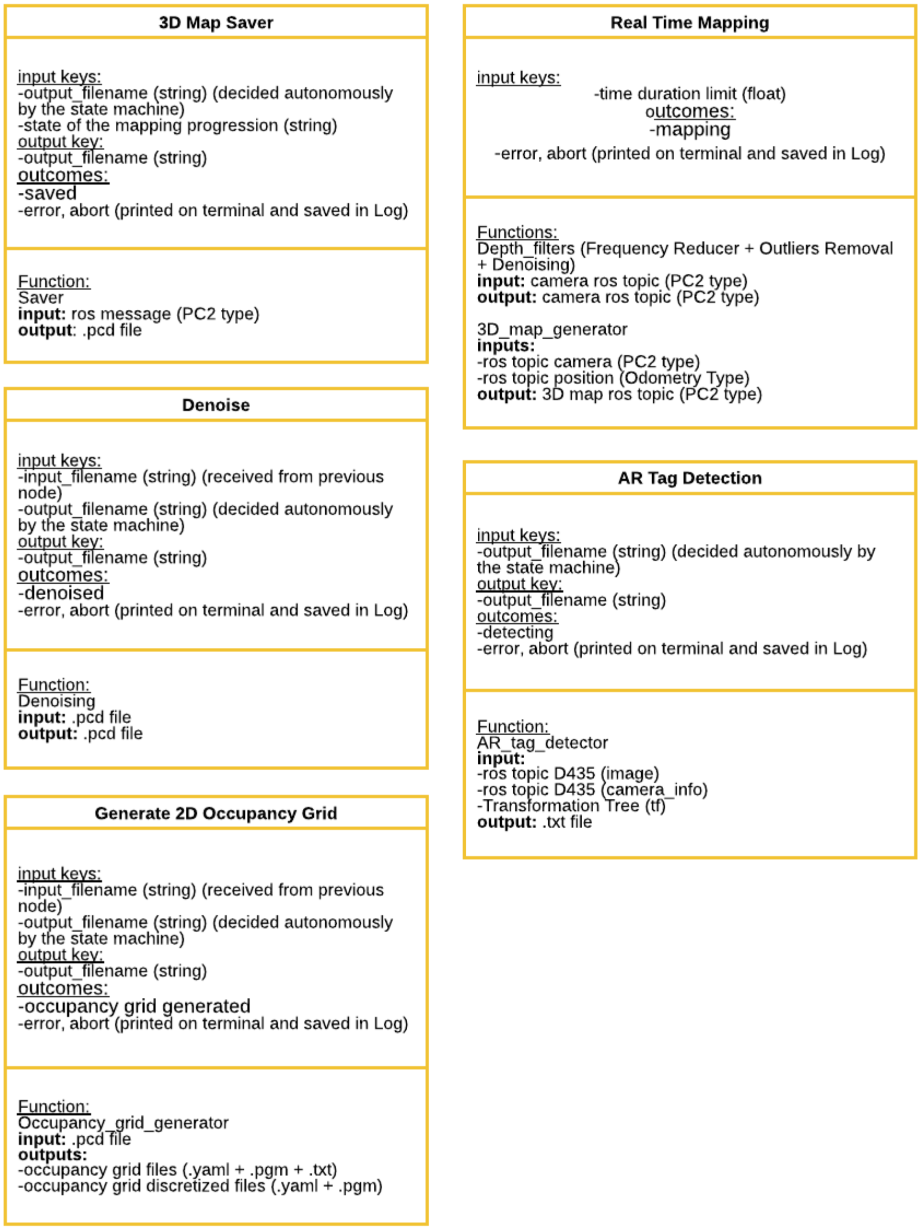


Figure 26: Class Diagram of the Drone Mapping algorithms

The Modules Removal Outliers and Extraction have the same structure as the Denoise Module so they are not shown in the Figure but can be deduced from the Denoise one.

2.15.2 Localization and Real-Time Mapping algorithm for the rover

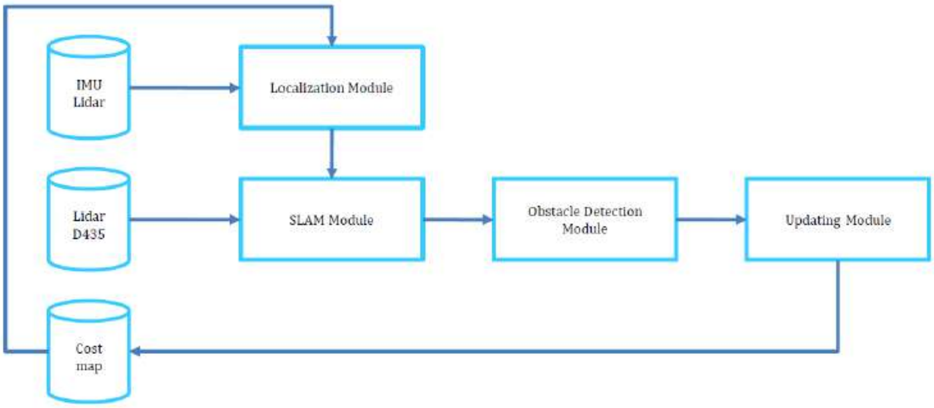


Figure 27: Localization and Real-Time Mapping block diagram for the rover

Figure 27 shows the block diagram of the envisioned Localization and Real-Time Mapping algorithm for the rover. The software has four main modules, Figure 28.

- Localization Module: It estimates the position of the rover on the 2D occupancy grid It executes step 1.a) of the general SLAM algorithm structure described at the beginning of this section.
- Localization and Real-Time Mapping Module: It performs a Localization and Real-Time Mapping in order to retrieve a more accurate position and to retrieve a local DEM map of the rover's local environment. It executes steps 1.b) to 5 of the general Localization and Real-Time Mapping algorithm structure described at the beginning of this section.
- Obstacle Detection Module: It compares the local environment of the rover with the 2D Occupancy grid in order to detect previously unseen obstacles.
- Updating Module: It updates the cost-map with the new obstacles.

A general description of the modules in terms of programming languages, internal and external interfaces can be found in Figure 48. The rover's Obstacle Detection software will use C++ as main coding language. The software will be then interfaced with the hardware through ROS topics, services and action libraries.

The rover's Localization and Real-Time Mapping algorithm comes to remedy the eventual deficiencies of the 2D Occupancy grid provided by the drone. It could be a reliable back-up plan in order to achieve the mission if something went wrong during the drone mapping (contingency situation). If we are in a nominal situation, the Localization and Real-Time Mapping algorithm of the rover acts as a Collaborative Localization and Real-Time Mapping between the rover and the drone in order to acquire a precise map of the environment. The rover Localization Module consists of the Tracking T265 Camera embedded software to retrieve a position. The Localization part of the Localization and Real-Time Mapping Module is done via AMCL(See Path-Planning Section) which enables the relocalization of the rover in the map generated by the drone. The Real-Time Mapping part of the Localization and Real-Time Mapping Module is done by Octomap and can be configured to work either with the T265 + D435i (same as the drone), or with T265 + 2D Horizontal LiDAR (to do a 2D Localization and Real-Time Mapping software). The Obstacle Detection Module is done in the Path-Planning Team with the LiDAR and thanks to the software Move-base. Due to resources limitation, the Updating Module was not implemented.

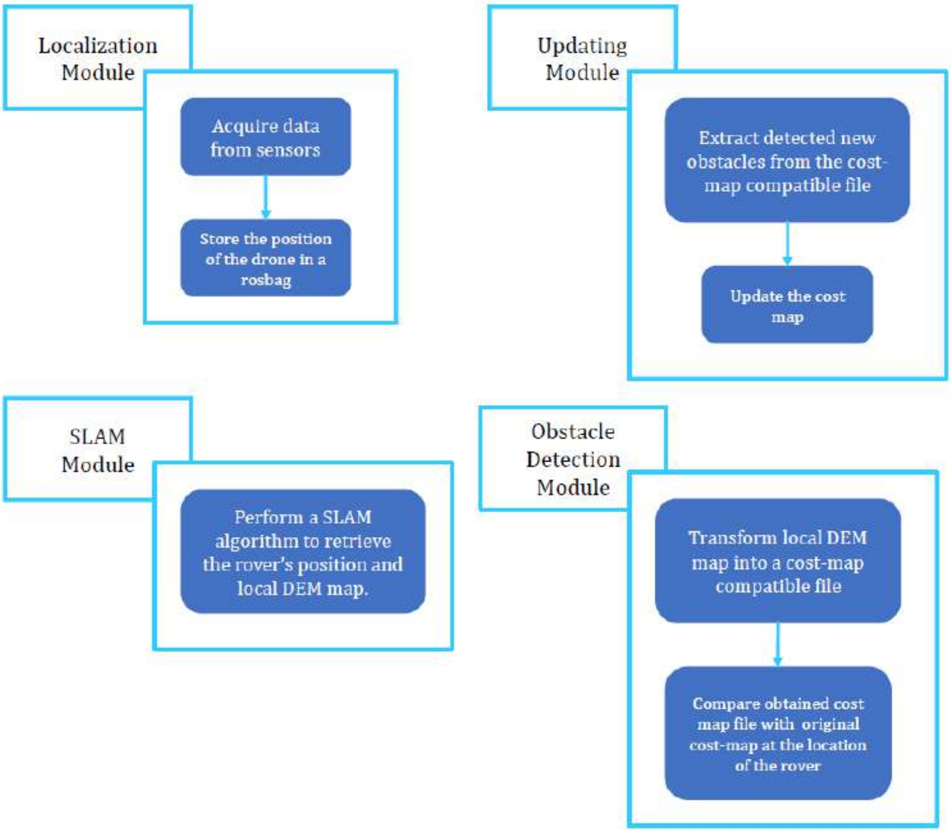


Figure 28: Modules of the obstacle detection software and high-level functions for the rover

Module	Description	Coding Language	Internal Interfaces	External Interfaces
Localization Module	At the initialization, this module synchronizes the position of the rover with the positioning system of the cost-map. This module will estimate the position of the rover on the cost map. It executes 1.a).	C++	INPUT: -IMU -Cost-map OUTPUT: -Estimated position of the rover	None
SLAM Module	It performs a Localization and Real-Time Mapping in order to retrieve a more accurate position and to retrieve a local DEM map of the rover's local environment. It executes steps 1.b) to 5.	C++	INPUTS: -D435 Depth Camera and Lidar -Estimated position OUTPUT: -Local DEM map	None
Obstacle Detection Module	It compares the local environment of the rover with the cost-map in order to detect previously unseen obstacles.	C++	INPUT: Local DEM map Cost map OUTPUT: -Cost-map compatible file of detected obstacles	None
Updating Module	It updates the cost-map with the new obstacles.	C++	INPUTS: -Cost-map compatible file of detected obstacles OUTPUT: -Updated cost-map	None

Table 48: Summary of the obstacle detection software modules, description, coding language, and interfaces for the rover

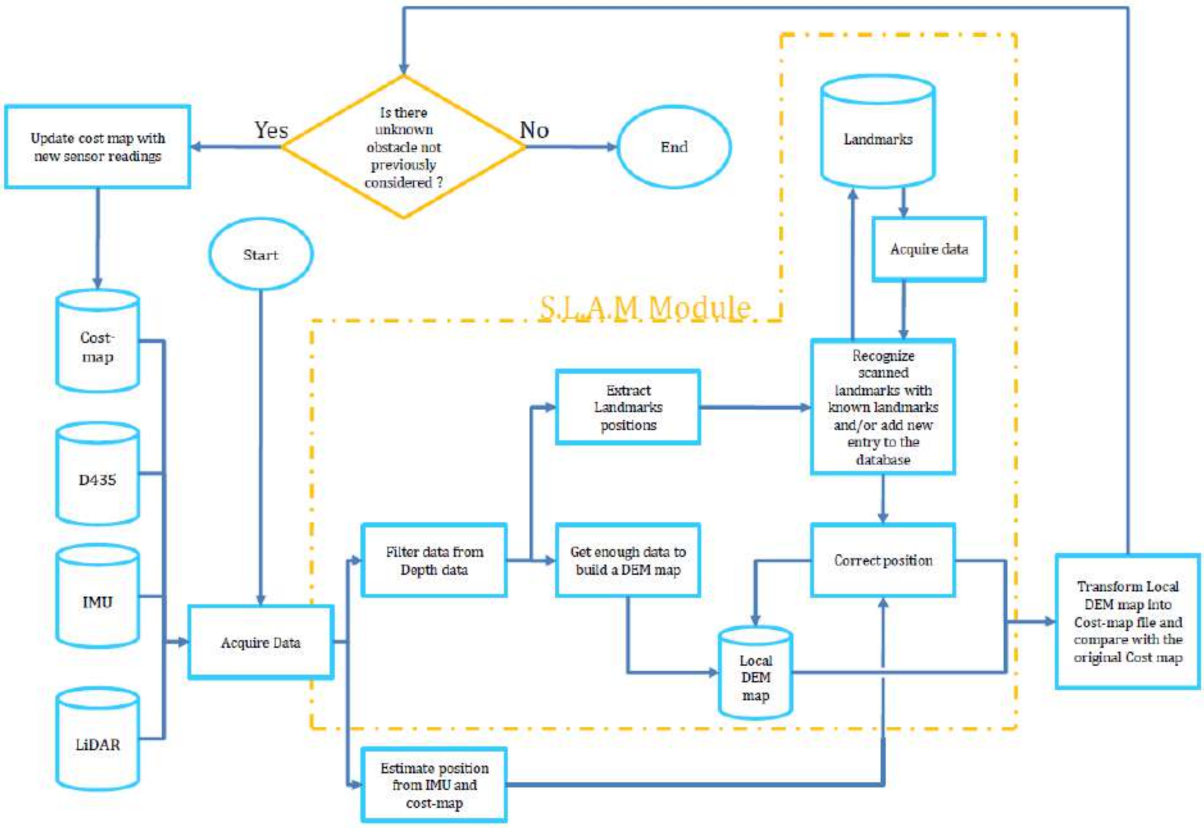


Figure 29: Obstacle detection process flow for the rover

2.15.3 Collaborative Localization and Real-Time Mapping between the Rover and the Drone

Our results show that we can detect 80% of the obstacles that have a height superior or equal to 6 cm with the drone. The limitation is due to the fact that small obstacles (about 6 cm height) are mixed with noise and also with non-obstacle features (such as grass). That is why the *Obstacle Avoidance Module* of the rover comes to remedy an eventual deficiency to detect the 20% estimated remaining undetected obstacles. Moreover, in case the drone was unable to generate a map or to reach the expected level of accuracy, the Collaborative Localization and Real-Time Mapping reveals its usefulness. Indeed, in that case, the rover Localization and Real-Time Mapping algorithms enable the rover to operate on the field even without a global map.

As a collaboration, the goal of the drone's Localization and Real-Time Mapping algorithm is to detect at least the big features of the environment that influence global trajectories for the rover (e.g. big rocks, rift, steep slope). A map is generated taking into account the mobility capabilities of the rover and is sent to the rover. The Path Planning algorithms of the rover compute an optimal path relative to the big features. As the rover starts moving, the goal of the rover's Localization and Real-Time Mapping algorithm is to confirm and/or detect the presence of small features in the local environment of the rover. If some unregistered small obstacles are detected, the path planning software would locally adapt the trajectory of the rover.

2.15.4 Merging Rover's and Drone's Frames and Localizing the Rover with the Drone

Once the drone has finished mapping the environment, the Mission Phase begins. The rover starts moving and performing its Localization and Real-Time Mapping algorithm. The rover uses the 2D occupancy map generated by the drone. Therefore, the rover must know its position in the map frame (which is the frame of the drone during its mapping phase). When the rover is powered on, it generates its own coordinate frame. Without any further information, the rover is unable to compute its position in the map of the drone. Therefore, a link must be found between the coordinate frame of the map and the coordinate frame of the rover. That way, the position of the rover can be computed in the coordinate frame of the 2D Occupancy map. We have found a method to compute that link. We compute the link between the two coordinate frames by using the points of interest on the field. Indeed, they are detected by the drone during the mapping phase and for each detected point of interest we possess their Position (X,Y,Z) and Orientation with respect to the drone coordinate frames. That means that if the rover detects at least one point of interest that has already been detected by the drone, we will possess the position and orientation of that point of interest in both coordinate frame. The analysis of the positions gives us the translation between the two coordinate frames whereas the analysis of the orientations give us the rotation between the two coordinate frames. If one point of interest is commonly detected by the drone and rover we take the transformation computed from this point of interest. If more than 2 points of interest are commonly detected by the drone and rover, we use the Least Square Method between all the transformations computed in order to find the optimal one. That links enables to make the two robotic systems connected, and make them able to collaborate. Indeed, with this established link, any information retrieved by a robotic system is now understandable by the other robotic system. We called the algorithm ”TF-Linker”. Test of this feature is presented in 2.20.3.7.

2.15.5 Points of Interest

The main objective of the CORODRO team is to send the rover to visit what we call Points of Interest (POI) of its environment. Due to their unique features that make them very recognizable, there is a high likelihood that they would be detected by our SLAM algorithm as landmarks (Step 3). However, during the execution of that step in the SLAM algorithm, the software will not know if a landmark is referring to a POI or not. That is why it is important not to draw a parallel between POI and the landmarks used internally by the SLAM software.

We used ARTag Detection program using ROS package ”ar_track_alvar” [15] that will detect AR-tags on the surface of the POI. Figure 30 shows the block diagram of the AR-tag detection algorithm. The software has been integrated into both the rover and the drone. AR-tags are *fiducial markers* that help us detect the POI. In Figure 31, we have considered a cuboid as Point of Interest with AR-tags placed on five sides of the cuboid with dimension as 36 cm*28cm*28cm. We are using 5 cuboids for drone mission and 6 cuboids for the rover mission to be considered as points of interest. We placed multiple AR-tags over the surface of the POI and detected them as multi-tag bundles (a combination of AR-tags as a single unit). Using the ROS package ”ar_track_alvar”, we have taken it as an input for the algorithm the camera image (D435 or T265) and the position of the drone and rover on the DEM Map. The output obtained using the software is the position and orientation of the multi-tag bundles with respect to the DEM Map reference frame.That output will be provided to the Task Planning and Path Planning teams.

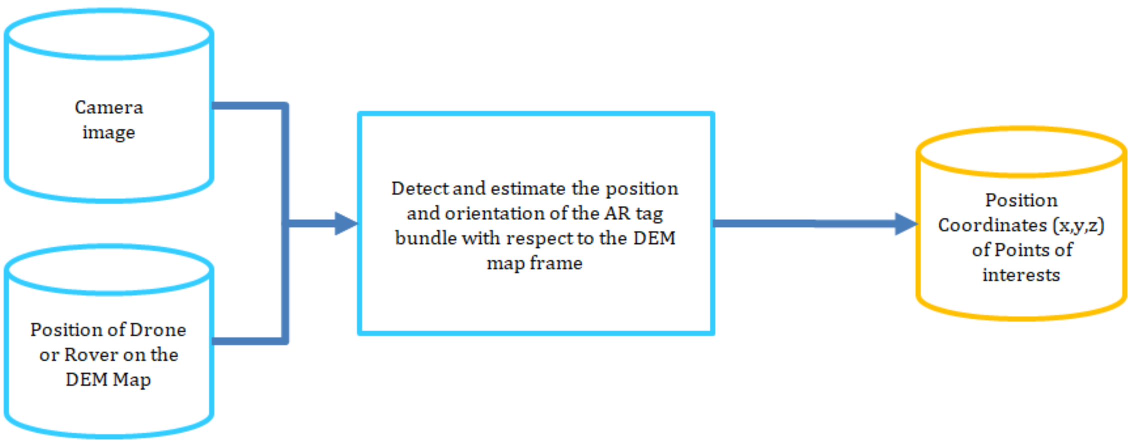


Figure 30: Block diagram for Points of Interest detection

Figure 31: Cuboid with AR-tags as Point of Interest

2.16 Path Planning

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, Section 2.17, 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 effectively move the system to its destination.
- It listens as well to the odometry readings of the encoders, the LiDAR reading of the obstacles, and the visual odometry of depth sensors to localize the system in a global or partial map [16].
- It uses Python as main 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 [16].
- Both systems can be stopped in case of contingency, as shown in Figures 37 and 39, by the control center. In case of a hardware failure, the operators on the field can directly intervene as detailed in Table 28.

This section will organized as follows:

- Analysis of the path planning strategies during the drone mapping phase, Section 2.16.1.
- Analysis of the path planning strategies during the rover and drone mission phase, Section 2.16.2.

2.16.1 Path Planning for the Drone Mapping Phase

The first challenge of the drone is to provide 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 given a maximum flight time (battery-dependent). The envisioned algorithm has four main modules, Figure 32:

- 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.
- Position Feedback Module: With the help of the position given by the drone's Localization and Real-Time Mapping software, the Position Feedback Module is able to estimate the error between the theoretical trajectory and the real one. It rectifies the movement needed for the rover in order to match as much as possible the planned trajectory.

- 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 directed to the PixHawk controller through the topics and services of MavROS.

A general description of the modules in terms of programming languages and internal and external interfaces can be find in Table 49, while Figure 33 shows the logical flow of the path planning algorithm applied to the drone for the mapping phase.

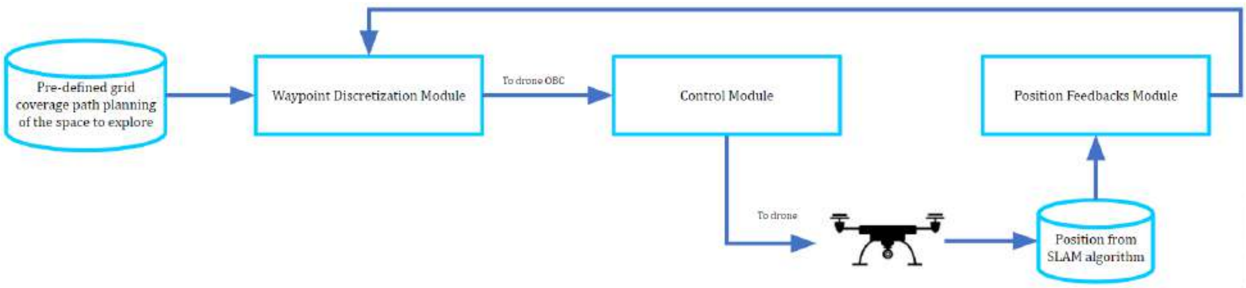


Figure 32: Drone mapping phase block diagram

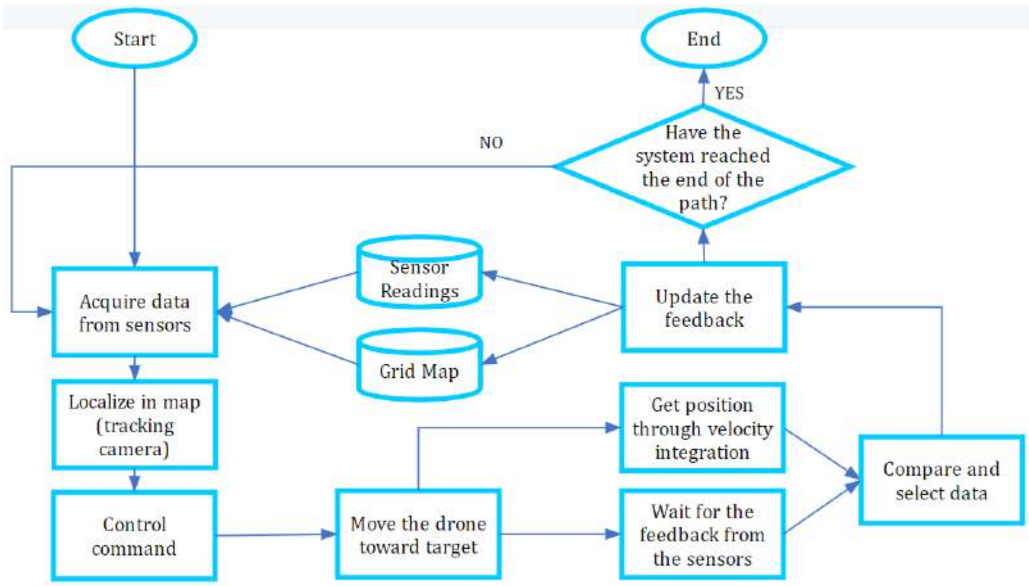


Figure 33: Drone mapping flowchart

2.16.1.1 Delimitation of the Operating Area The first step for the drone during its mapping phase is understanding the limits of its operational area. This statement is valid for both the Earth analog and the lunar missions.

On the Moon, the mission of the flying hopper (simulated 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, hopper 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 such high-contrast border with a rope of vivid color, Section 2.6 and Figure 34.

The first tasks of the mapping phase for the drone will be to identify its operating area. Since the dimensions of the terrain to map are known, Section 2.6, the drone does not need to scan the whole border to understand its operating area. Instead, it is sufficient to obtain data of four reference points lying on the border. Those will be the first landmarks that will provide the mapping limits, Figure 34.

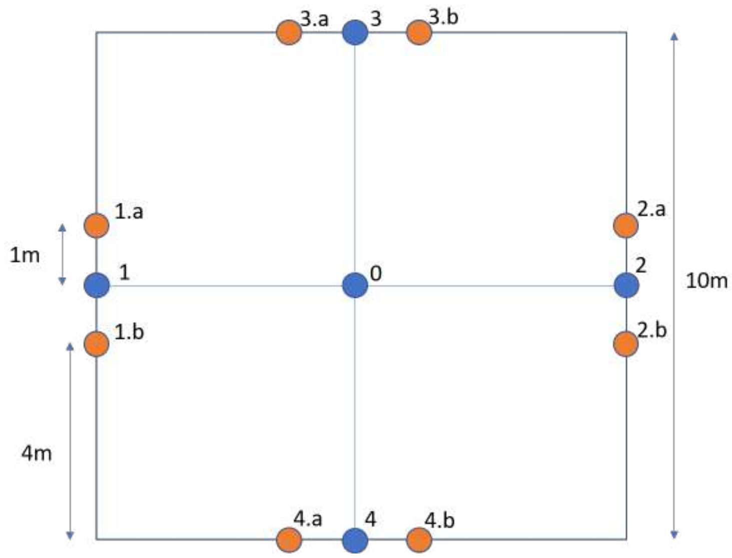


Figure 34: Drone referencing points

The referencing points will be represented by four AR-tags placed at the middle of each line of the square that defines the operating terrain. As the drone will be initially placed at the center of the terrain, it will follow a straight path to each point and then go back to the initial point, tracing a couple of orthogonal axes along both directions, Figure 35. To guarantee the correct referencing of the drone, two extra AR-tags will be placed one meter away from each referencing point anticipating small drifts that may occur during motion, mainly due to wind or localization errors. While the drone acquires data of the borders, the Localization and Real-Time Mapping subsystem will take advantage of this motion by simultaneously initiating the mapping and localization processes.

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

2.16.1.2 Mapping of the Operating Area As the drone now knows the size of its operating area, it can fully map it. As previously explained, Section 2.16.1.1, the drone has already mapped some part of the area during the process of accessing the reference locations. To map the rest of the terrain, a grid-like pattern with several waypoints has been chosen, Figure 35.

Due to the square shape of the operating area, the chosen pattern of straight lines assures the coverage of the remaining terrain without flying over the same areas covered in the drone referencing phase, thus allowing the optimisation of the batteries and consequently the duration of the mission. In order to determine the distance between the path loops, the Field of View (FOV) of the camera had to be taken into account.

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

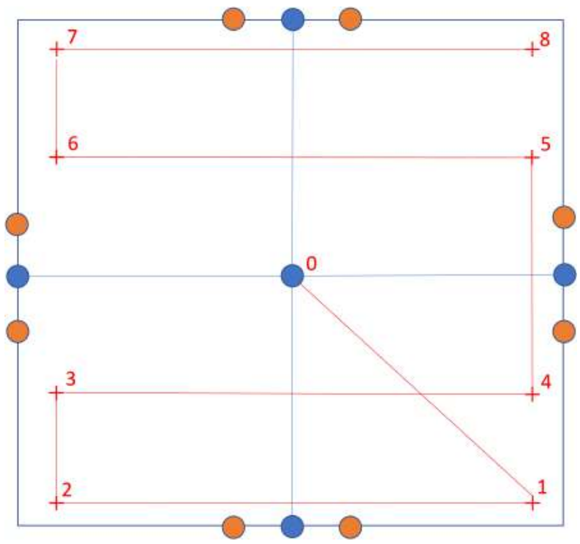


Figure 35: Drone pattern for mapping

Module	Description	Coding Language	Internal Interfaces	External Interfaces
Path Planner Module	The module computes the optimal trajectory for the drone relatively to the Localization and Real-Time Mapping requirements	ROS Yaml file ROS intefaced with Python	INPUTS -Reference points expected positions OUTPUTS -Optimal path for recognition	None
Controller Module	The module interfaces the high-level goals of the path planners with the system actuators	ROS Yaml file ROS intefaced with Python	INPUTS -Commands from the path planners OUTPUTS: -Actions for the actuators of the systems	None
Trajectory Discretization Module	The module discretizes a given trajectory into a sequence of waypoints that have to be followed by the drone	ROS intefaced with Python	INPUTS -Optimal path for recognition OUTPUTS: -Sequence of waypoints to follow	None
Position Feedback Module	With the help of the drone's Localization and Real-Time Mapping algorithm, the drone will be able to estimate its position and therefore its displacement	ROS intefaced with Python	INPUTS -Position output from the drone's Localization and Real-Time Mapping	None

Table 49: Summary of the software modules, description, coding language and interface for the drone mapping phase

2.16.2 Path Planning for the Drone and Rover Mission Phase

The rover and drone mission phase come after the mapping phase of the drone. Thus, in nominal conditions, the rover and the drone are provided with a global 2D Cost-map of the environment, Table 20. Therefore, the main goal of the Path Planning software for the mission phase is to compute the optimal trajectory and to physically move the robots in a known environment.

The software has seven main modules as shown in Figures 36 and 38:

- Translation Module: The module will transform the 3D map generated by the drone to a 2D cost-map. We will use the control center powerful computers to perform this ”translation”.
- Localization Module: The module will estimate the position of the robots in a known map (amcl localization and EKF localization [16]).
- Global Path Planner Module: The module computes the optimal path for the rover in the known 2D cost-map. It uses a D* algorithm from Navigation Stack [16].
- Local Path Planner Module: This module uses the local map provided by the Rover’s Localization and Real-Time Mapping Algorithm in order to adapt locally the trajectory of the rover if needed. It takes into account the new obstacles (if there are) registered on the local map to adapt the trajectory. The local path planner will smooth the overall trajectory followed by the systems.
- Controller Module : The module interfaces the high-level goals of the path planners with the system actuators.
- Trajectory Discretization Module: The module discretizes a given trajectory into a sequence of waypoints that have to be followed by the drone.
- Position Feedback Module: With the Localization Module, the Position Feedback Module is able to estimate the error between the theoretical trajectory and the real one. It rectifies the movement needed for the rover in order to match as much as possible the planned trajectory.

A general description of the modules in terms of programming languages and internal and external interfaces can be found in Table 50, while Figures 36, 37, 38 and 39 show the logical flow of the Path Planning algorithm applied to the drone and rover, respectively, during the mission phase.

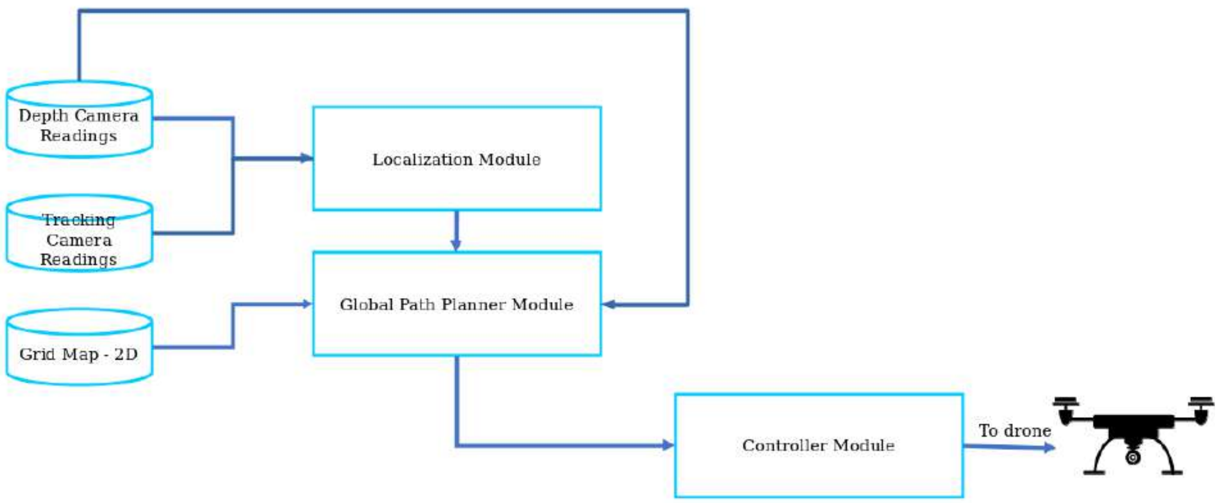


Figure 36: Drone mission phase block diagram

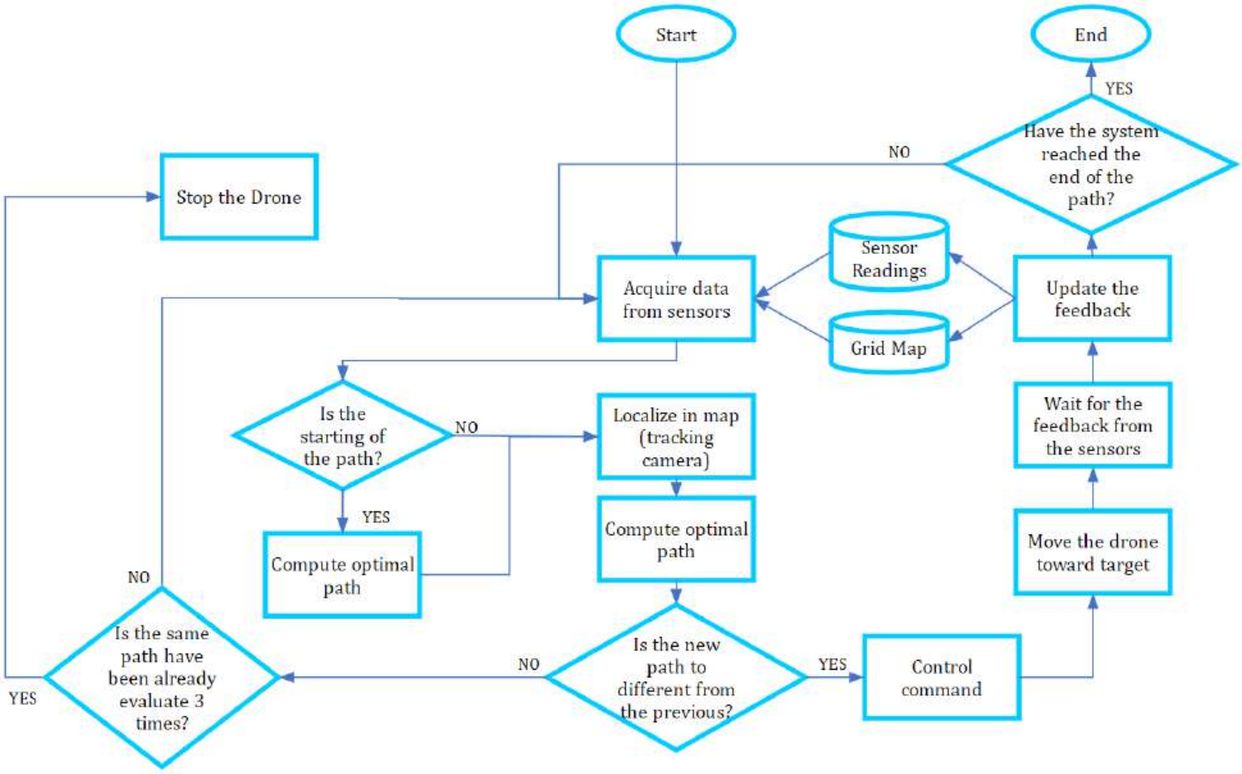


Figure 37: Drone mission flowchart

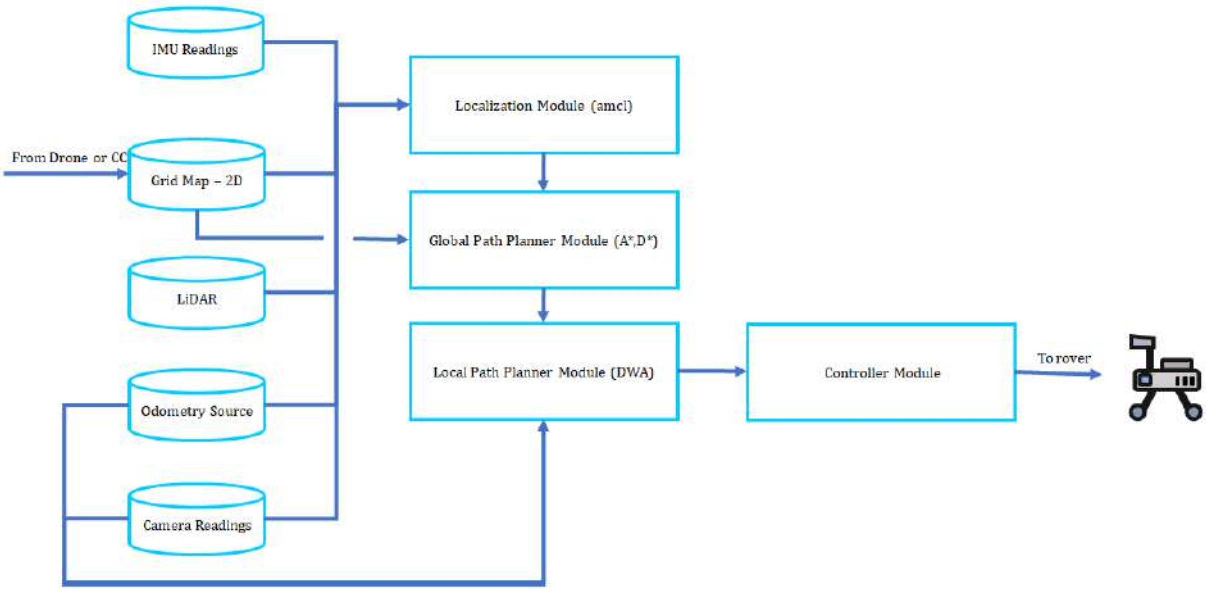


Figure 38: Rover mission phase block diagram

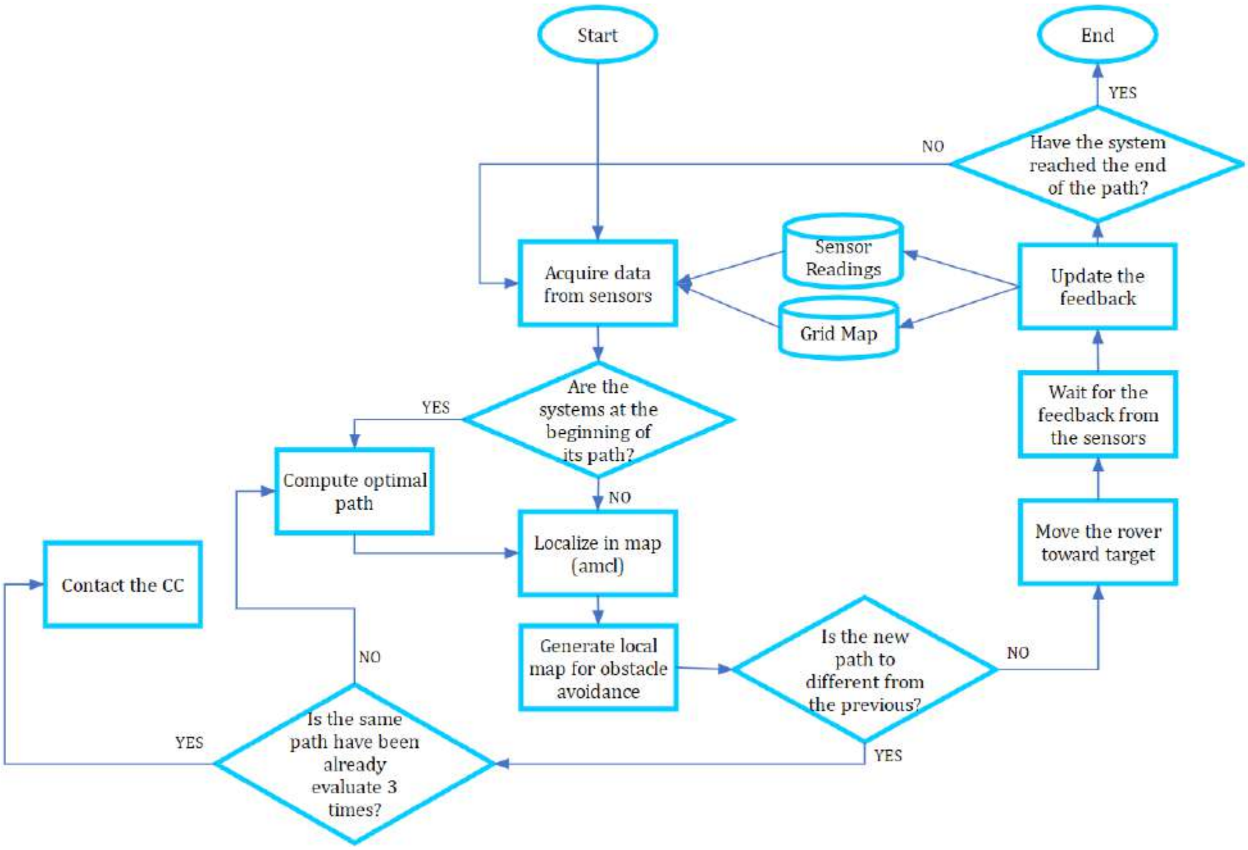


Figure 39: Rover mission flowchart

Module	Description	Coding Language	Internal Interfaces	External Interfaces
Translation Module	The module will transform the 3D map generated by the drone in a grid map and a cost map for the rover or the drone. See detailed description in Localization and Real-Time Mapping Section 2.15.1.5	C++ Python	INPUTS -DEM map OUTPUTS -Grid map -Cost map	None
Localization Module	The module will estimate the position of the rover in a known map (amcl localization - ROS navigation stack)	ROS Yaml file ROS intefaced with Python	INPUTS -Rover -IMU readings, encoders reading, depth camera -Drone -Tracking camera OUTPUTS -System position	None
Global Path Planner Module	The module computes the optimal path for the rover known the grid map and the cost map. It will use algorithms like A* and D* (already implemented in the ROS navigation stack)	ROS Yaml file ROS intefaced with Python	INPUTS -Cost-map OUTPUTS -Global optimal path	None
Local Path Planner Module	The module generates a local map updated with unknown obstacles (if present). The local path planner will smooth the overall trajectory followed by the systems	ROS Yaml file ROS intefaced with Python	INPUTS -Camera readings -LiDAR readings OUTPUTS -Local optimal path -Update in the overall map with new obstacles	None
Controller Module	The module interfaces the high-level goals of the path planners with the system actuators	ROS Yaml file ROS intefaced with Python	INPUTS -Commands from the path planners OUTPUTS: -Actions for the actuators of the systems	None
Waypoint Discretization Module	The module discretizes a given trajectory with a series of waypoints that have to be followed by the drone	ROS intefaced with Python	INPUTS -Grid-like path for recognition OUTPUTS: -Series of waypoints to follow	None
Position Feedback Module	With the help of the tracking camera the drone will be able to estimate its position and therefore its displacement	ROS intefaced with Python	INPUTS -Tracking camera position	None

Table 50: Summary of the software modules, description, coding language and interface

The rover path planning is divided in two main algorithms:

- Local Path Planner
- Global Path Planner

The global path planner uses the algorithms D* [8] and A* [8] to find the shortest path from two points in a given map. Both are graph-based algorithms, but the benefit of A* is the use of a heuristic to prune some of the paths explored and save computational costs. Therefore, no matter how good this heuristic is, there will be a non-zero chance of pruning the actual optimal path and many near-optimal paths. The D* search space is wider, and therefore, the pruning of the optimal path is less likely to happen. In general, D* can be a better re-planner that A* [14] since (i) local changes in the world do not impact much on the path, and (ii) it avoids high computational costs of backtracking. [14].

The metric used by the two algorithms to estimate the path is quite similar, excluding the A* heuristic, as they both search in a given grid the best path. Therefore, the given grid has to implement some constraints for the algorithm to find a real good path given the size and limitations of the system. This is performed not by feeding the algorithm with the ”brute” occupancy grid map from the Localization and Real-Time Mapping algorithms, but with a cost-map. In the global cost map the obstacles are inflated, usually to 0.5%, and the footprint of the system is taken into account to exclude paths that would not fit the size of the robot. In this case, the considered metrics for the cost-map are:

- Size of the obstacles
- Size of the robot footprint

The occupancy grid map of the Localization and Real-Time Mapping code considers as obstacles all the slopes that cannot be traversed by the rover.

The local path planner uses the Dynamic Window Approach (DWA) for obstacle avoidance [6]. The algorithm takes into account the overall path from A*/D*, the ”inflated” radius of the obstacles and the robot footprint to avoid unforeseen obstacles on its path. The idea of the local path planner is to avoid the unforeseen obstacles and then return on the optimal path. If the local and the optimal path start to have an overall euclidean distance superior to a threshold, the overall global path is recalculated. The threshold for us corresponds to the robot footprint, in which case we re-evaluate the path. The basic steps of the DWA are:

- To define the robot control space as (dx,dy, dtheta).
- To analyse different local paths as different coupling of angular and linear velocities for a given short period of time in the future.
- To score its trajectory propagated in the future weighing the parameters:
  - Distance from the obstacles (already inflated by the cost-map);
  - Distance from global optimal path,
  - Avoided unforeseen obstacles,
  - Distance from final target.
- To choose the most suitable one.

Changing the weight of the parameters, it is possible to obtain different performances. In this moment, those parameters are tuned with our testings.

After the PDR, the Path Planning team focused on the drone mapping phase (extensively explained in Section 2.16.1 and on the rover mission phase).

The drone mission phase will recycle the navigation stack of its mapping phase, applied to the case of a known 2D map. We will use the MavROS topic ”set-point” to move the drone toward the targets. For now, we keep the assumption of no obstacles in LOS, line of sight, for the drone during the FC. Therefore, we rely on a 2D cost-map. That will permit us to obtain a functioning algorithm by July, ready to fly for the FC. However, if we have time, we would like to perform a 3D navigation with the drone with a 3D cost map.

On the rover side, we decided to test and work directly on our hardware. The following paragraphs describe the various tests and related encountered problems and solutions. During this testing we created a map directly using the rover because no map was still provided by the drone. However, this first test helped us to better understand the use of ROS Navigation Stack. We already tested the ”cmd_vel” and ”move_base” topics and their actuation with the rover. Those first trials where quite successful. However, now the rover Path Planning team is focusing all its resources on improving the localization algorithm for outdoor applications.

2.16.2.1 Training with Morse Before starting implementing the algorithms on the robotic platform, we used a ROS middleware wrapper called Morse to learn how to localize and navigate using ROS tools. We have implemented the overall ROS navigation stack and we have reached an acceptable level of success. However, the differential driver controller implemented in ROS for our platform, LEO Rover, performs poorly during turning maneuvers. If we compare the response to a 180° turning command in the simulation and the real rover we observed a mismatch in angular position of 30°. For this reason we decided to work directly on the real platform, which was already available in our laboratories.

2.16.2.2 Mapping We have created a map with the rover using two different algorithms: (i) the Gmapping algorithm provided by ROS Navigation Stack, (ii) the Hector Mapping algorithm provided by another ROS library.

- Gmapping uses the scan readings and the transformations between reference systems, usually called ”TF”, to provide a map.
- Hector mapping subscribes only to the scan readings.

At the beginning of the test, the Gmapping algorithm was launched from the computers of our laboratories. Those computers use an internal communication network based on the WiFi protocol. We had various issues with the TF during this first experiment. After various test we understood that this was due to an overload of the WiFi network between the rover, the other robots in the lab and the computers. To solve this problem, we launched the algorithm directly on the rover OBC, that is running ROS Indigo. After the test with Gmapping was successful, we kept on analyzing the output of the Hector Mapping algorithm. Both algorithms provided us with a reliable map. However, in case of contingency situation during the FC, Table 28, we will probably use the Gmapping algorithm to maintain the notion of the transformation between the different reference systems.

2.16.2.3 Localization Issues - Solved The first localization module that we implemented uses a Montecarlo probabilistic localization system for a robot moving in 2D called ROS amcl. The module uses a particle filter to track the pose of a robot in a known map. ROS amcl subscribes to the scan readings, the reference system transformation TF, the initial pose of the robotic system and the map topic provided by the so-called ROS map server.

- Problem: We were trying to fix the errors in the TF that we had with the ROS amcl module related to the wheel odometry. We identified the error in the format of the ROS Topic message for the wheel odometry (Nav msgs/odometry). We are know coding a node that will try to solve this issue. We have envisioned a test the 10/02/2021 to try to fix this problem in order to proceed with more complex algorithms and interface the rover path planning with the task planning algorithm.
- Solution: We fixed the ROS amcl getting a better odomentry reading with two different solutions:
  - Using the visual odometry of the tracking camera.
  - Publishing a "corrected odometry" from the ROS Hector Mapping algorithm.

During the FC we will use the tracking camera visual odometry to refine our position, Figure 40 (The complete video can be found at: https://drive.google.com/file/d/1v9a7znalboiBoNZ-ouLtE6hnlodztDsv/view?usp=sharing. However, the implementation with from the ROS Hector Mapping algorithm will be left as back-up in case we have some issues with the camera.

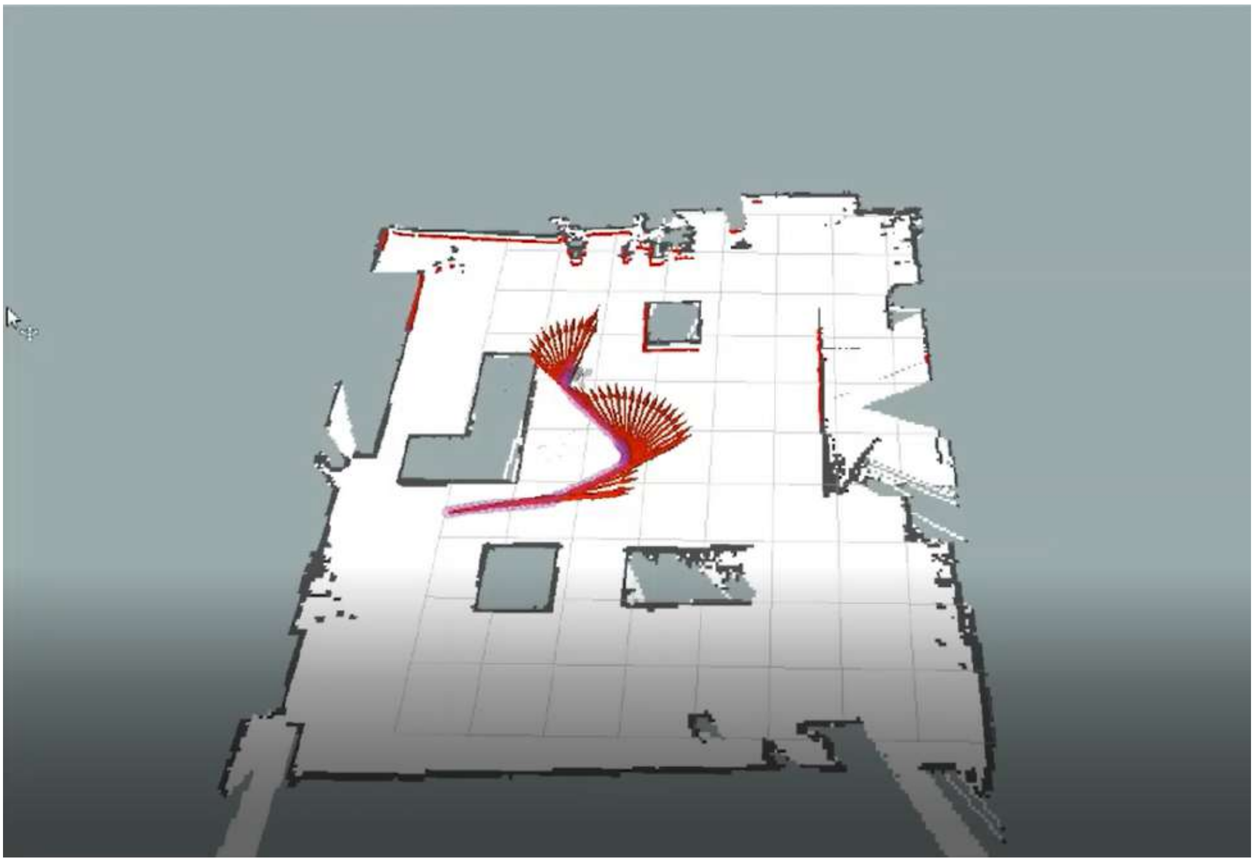


Figure 40: Rover localization in a known map.

2.16.2.4 Local and Global Path Planner To implement the global and local path planner, we have used "ROS Move_base" package. The parameters used by this ROS package are the same ones that have been explained in Subsection 2.16.2. Figure 41 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).

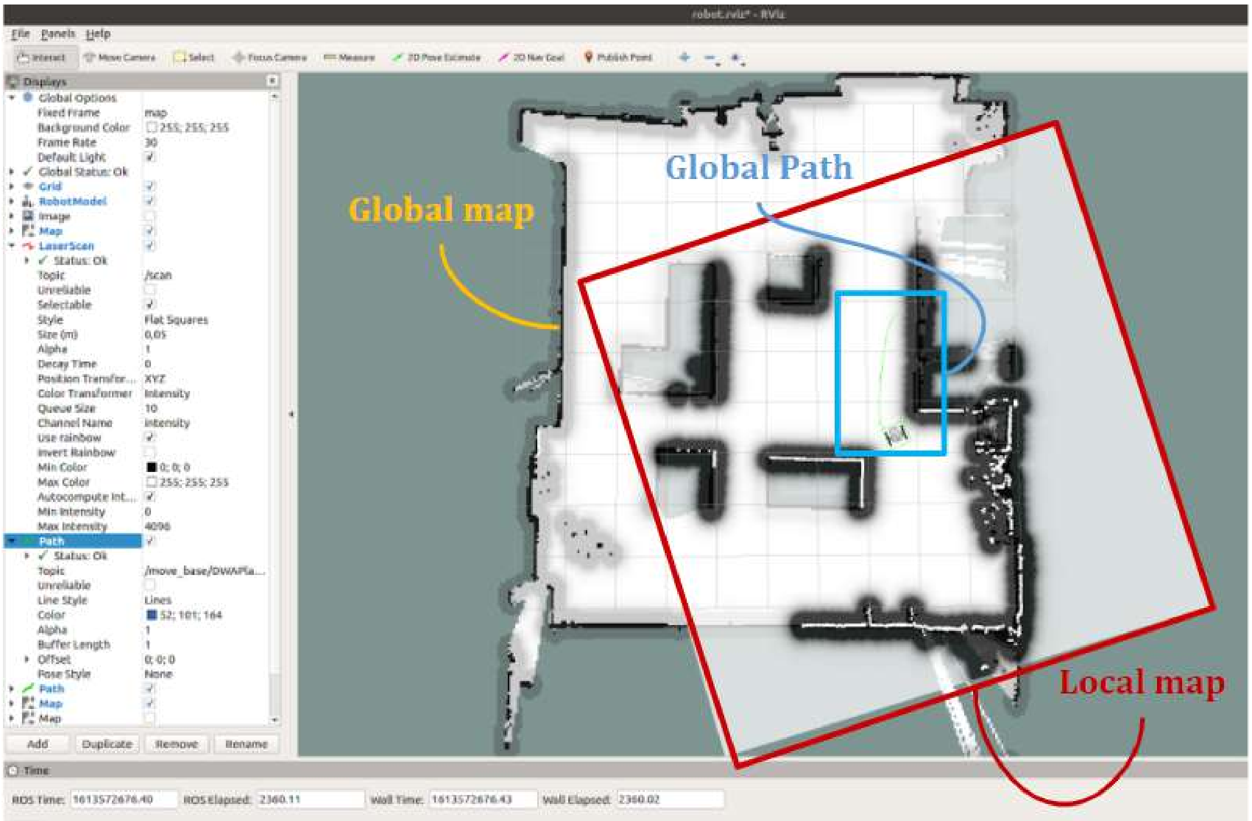


Figure 41: Path Planning set-up for the rover

2.17 Task Planning

The task planning software will define the overall behavior of the two collaborative systems. It will define: (i) when to execute the different tasks, (ii) the order of the different waypoints to visit during the mission, (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. In the specific case of our study, we will apply HDDL, an extension of the PDDL. The HDDL allows to define clusters of actions to be executed by a robotic system, thus coding its expected behaviour given a set of triggering situations and estimating its resources. The software has six main modules, Figure 42. A general description of the modules in terms of programming languages, and internal and external interfaces can be found in Table 51.

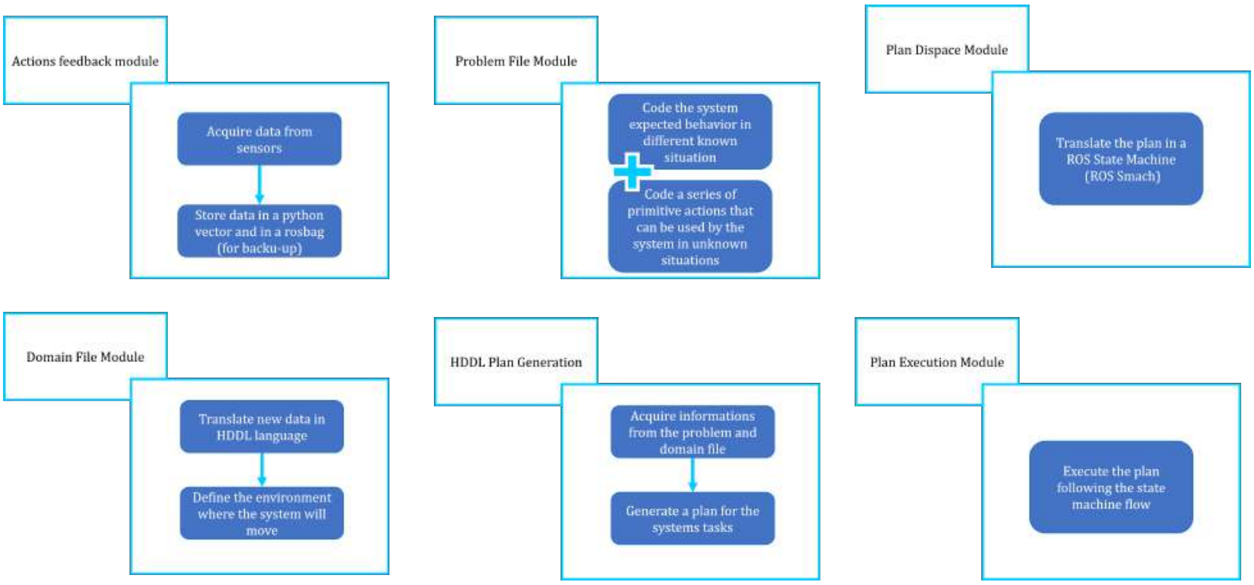


Figure 42: Modules of the task planner and high-level functions

- 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 can be executed by the system. It considers even a cluster of possible actions that can be triggered by particular situations.
- 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.

Module	Description	Coding Language	Internal Interfaces	External Interfaces
Problem File Module	This module groups all the available information on the environment where the system is moving	HDDL	INPUTS: - Sensor readings supplied the Action Feedback Module - Initial generated grid map and position of the systems in the map OUTPUTS: - HDDL Plan Generation Module	None
Domain File Module	This module outlines all the possible actions that can be executed by the system It considers even a cluster of possible actions that can be triggered by particular situations	HDDL	OUTPUTS: - HDDL Plan Generation Module	INPUTS: - Expected behavior of the system - Behavior of the system from testing
HDDK 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.	".txt" File	INPUTS: - Domain file - Problem file OUTPUTS: - ".txt" file to be parsed	None
Plan Dispatch Module	The module will parse the ".txt" file and interface it with a State Machine	Python	INPUTS: - ".txt" file from the HDDL Plan Generation Module	INPUTS: - None for the rover - WiFi-like network for the drone to receive the plan generated by the rover
Plan Execution Module	The module will execute the plan generated by the Plan Dispatch Module using the Action Library of ROS	Python	INPUTS: - State machine generated by the Plan Dispatch Module	None
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 or if a way-point cannot be reached	Python	INPUTS: - Sensor readings from ROS topics and services	INPUTS: - None for the rover - WiFi-like network for the drone to send its sensors readings to the rover

Table 51: Summary of the software modules, description, coding language and interface

Figure 43 shows the block diagram of the envisioned task planner. The software is interfaced with the other teams as follows:

- The ConOps and design of the mission from Mission Analysis is coded in the domain file, Annex 10. The starting point of the code is the Enhanced Functional Flow block Diagram (EFFBD) for the drone and rover mission, Fig. 44.
- The occupancy grid from Localization and Real-Time Mapping is translated into the "initial conditions" of the "problem.hddl" file. We coded an interface that from the grid-map can output the needed predicates of the HDDL problem file, Listing 1 (full problem file in Annex 10).
- 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 as well as the global and local planner, Section 2.15 and 2.16.2.
- The SMACH Monitor State of ROS [42] 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 can be easily followed by the control center operator.

Listing 1: HDDL problem file interface with Localization and Real-Time Mapping

```
(can_traverse rover0 waypoint0 waypoint5)
(can_traverse rover0 waypoint5 waypoint0)
...
(can_traverse rover0 waypoint5 waypoint4)
(can_traverse rover0 waypoint5 waypoint6)
(can_traverse rover0 waypoint6 waypoint5)
```

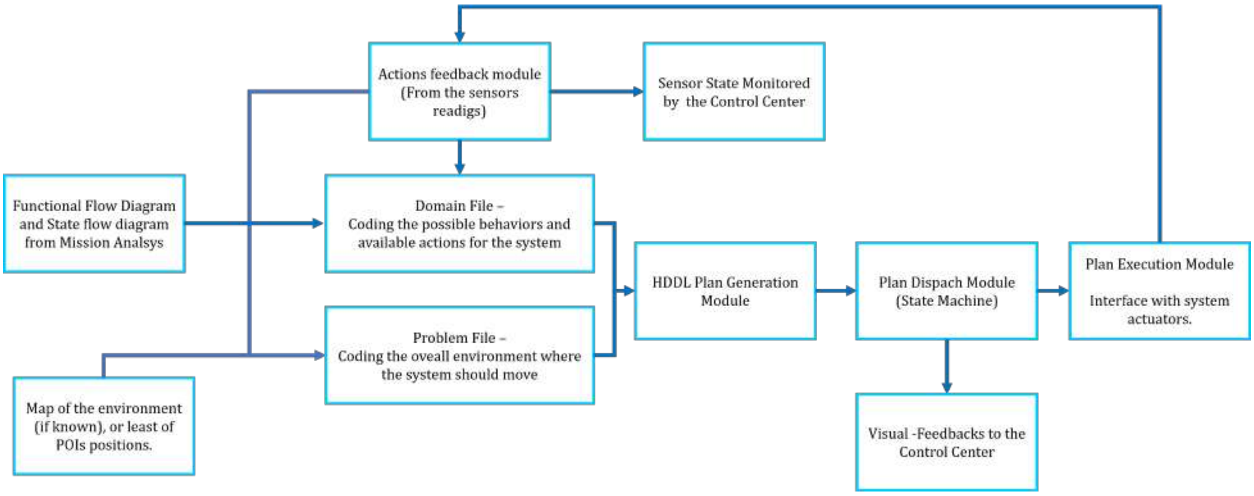


Figure 43: Task planner block diagram

```
...
(can_traverse drone1 waypoint0 waypoint5)
(can_traverse drone1 waypoint5 waypoint0)
(can_traverse drone1 waypoint0 waypoint4)
(can_traverse drone1 waypoint4 waypoint0)
...
(can_traverse drone1 waypoint6 waypoint7)
(can_traverse drone1 waypoint7 waypoint8)
(can_traverse drone1 waypoint8 waypoint7)
...
```

Figures 45 and 46 show the logical flow of the task planning algorithm applied to the rover and the drone, respectively.

The HDDL *plan solver* is coded in Python and it is provided by our sponsor ONERA. It is called HyPOP [36] [23]. This code outputs a plan that is saved in a ".txt" file from a HDDL domain and problem files (Annex 10), First Step of Figure 47. This plan is then parsed by the "HDDL ros wrapper". This ROS node is coded in Python and interfaced with ROS Smach [42], the ROS library to implement and visualize state machines, Second Step of Figure 47. The last step is the effective action of the robotic platform: the actions of the robotic system are followed with the logging information that appears in the terminal, Third Step of Figure 47. The logging information will be effectively seen by the operators of the control center during the field campaign. However, the visualization through state machine is preferred because it is easier to understand and follow.

2.18 Plan Definition

2.18.0.1 Rover and Drone Mission As explained in the previous paragraph, the AI planner exploits the HDDL modelling language. The HDDL language is heavily based on PDDL, only introducing the hierarchical component. The possibility of defining a hierarchy of functions accomplished by the system allows the description of complex scenarios. A clear and straightforward explanation of both HDDL and PDDL can be found in [41]. Both HDDL and PDDL are in the domain of *symbolic AI* and they are based on ontologies, known as knowledge systems in the domain of AI [40]. The important concepts to read

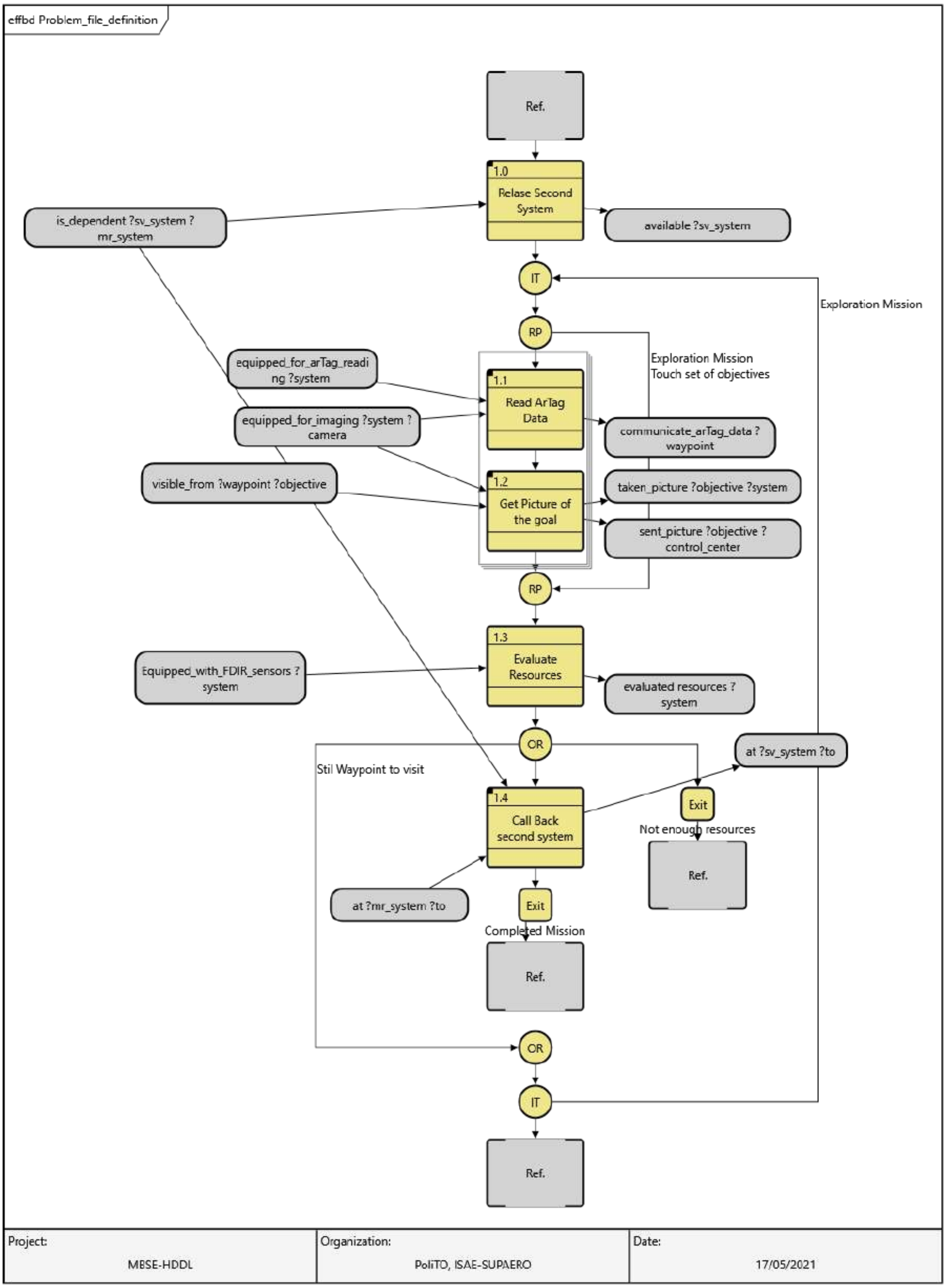


Figure 44: Mission Analysis EFFBD for the IGLUNA mission

and understand HDDL files are:

- *:task()* - those entries define what we expected the system under study to be able to perform.

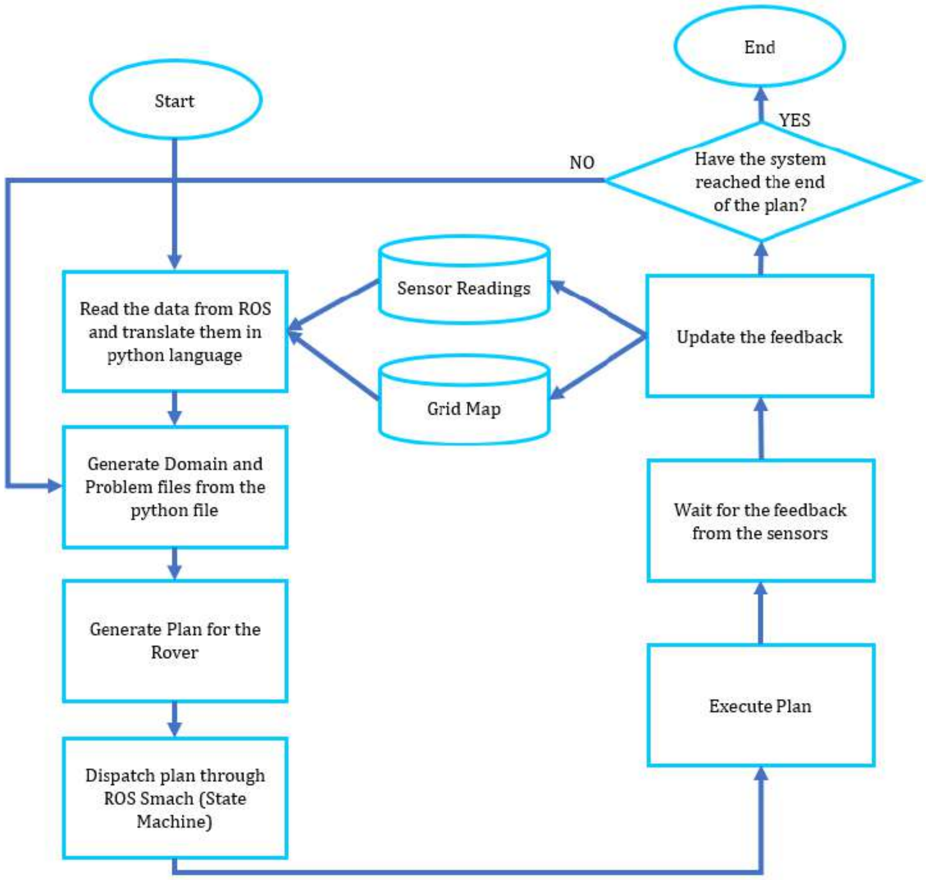


Figure 45: Task planner process flow for the rover

- *:method()* - those entries define how we can accomplish a *:task()* through ordered sub-tasks. These subtasks can then be *compound tasks* or *primitive tasks*. The first are subtasks that can be further decomposed by other methods. The latter are substasks with direct effect on the system. They are usually called *:action()*.
- *:action()* - those entries are the ones that change the state of the system using predicates, true or false statements.
- *:predicates()* - those entries are a true or false statement that describe the state of the system. Those are effectively the sentences that make the plan advance.

The collaborative mission of the drone and rover is a quite complex problem, therefore perfect to model with HDDL. The objective is to optimize the tasks that both systems need to perform simultaneously. In the ideal scenario, the two system would move simultaneously collaborating in finding the targets, read the arTag and take the pictures of the objectives. However, for safety reasons it is possible that the rover and drone will perform their mission separately, for example if there is wind and we cannot maintain a stable drone flight. The rover will visit the waypoints as given by the task planner, and then the drone will complete the exploration on a second run. However, the task planning optimization consider the full on collaborative scenario. The rover and the drone have different capabilities and different view points in regard to obstacles and targets. These capabilities should be taken into account while evaluating their plan. That has been effectively added to the *problem.hddl* file (Annex 10) indicating their visibility from the neighbouring position with

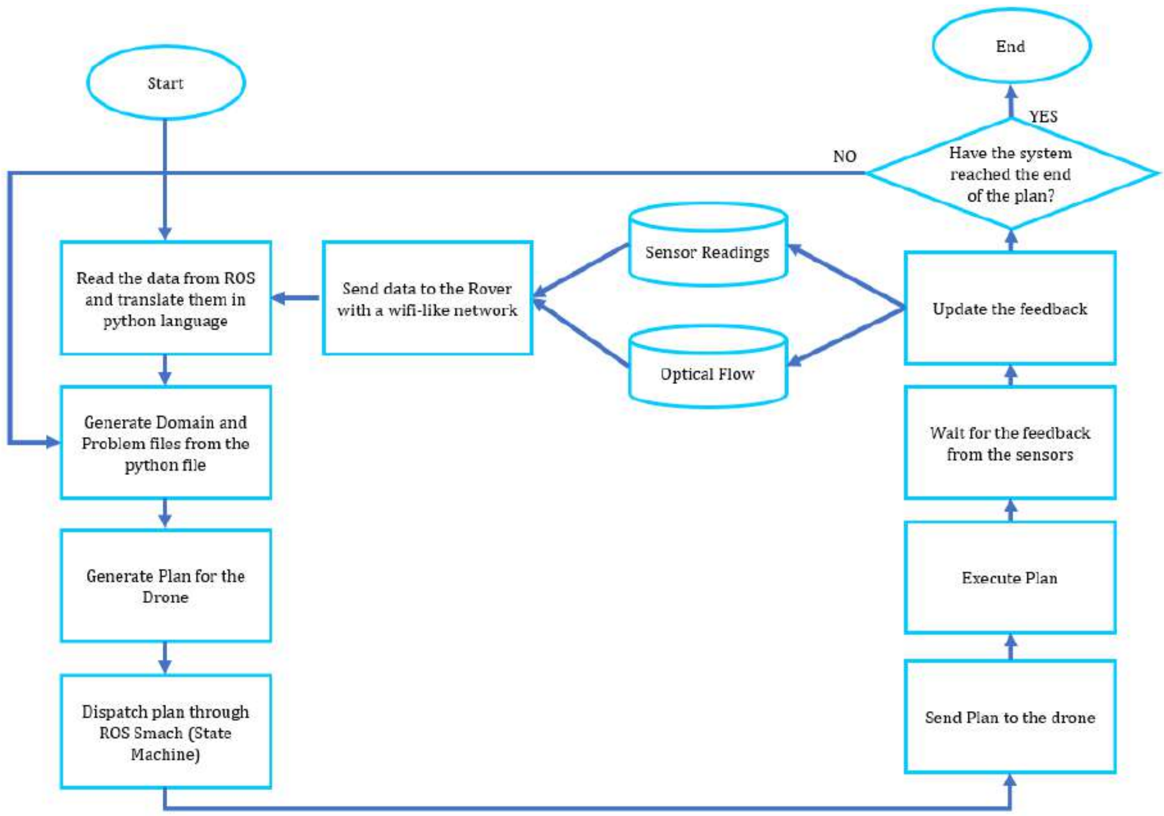


Figure 46: Task planner process flow for the drone

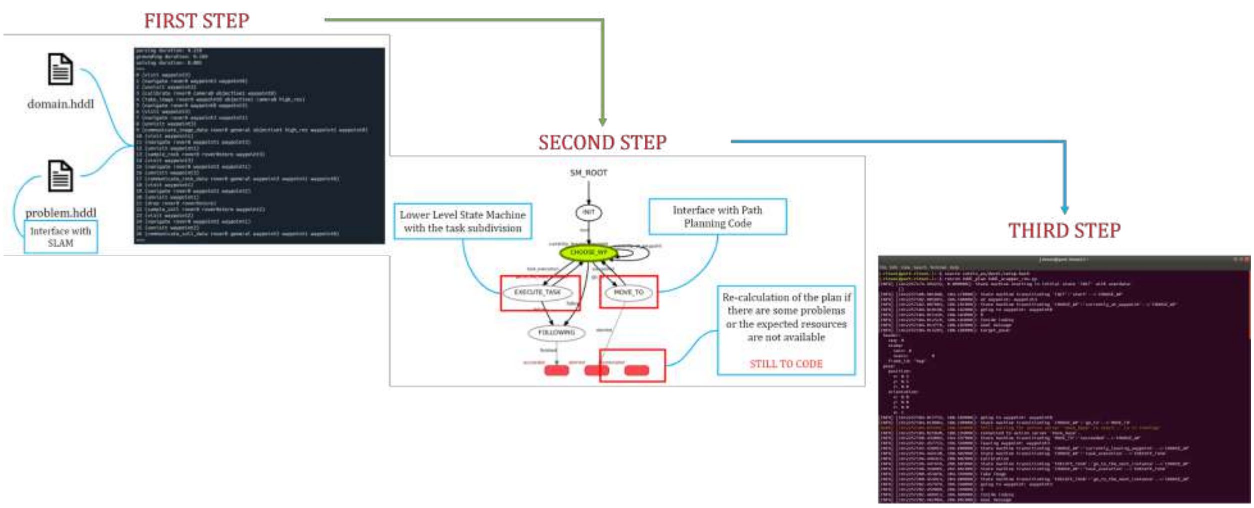


Figure 47: Overall workflow of the task planning code

respect to the point of view of the observing system, Listing 2.

Listing 2: HDDL problem file goal position entries

```
...
(visible_from objective0 waypoint5 rover0)
(visible_from objective0 waypoint6 drone1)
(visible_from objective1 waypoint8 drone1)
(visible_from objective2 waypoint0 rover0)
(visible_from objective2 waypoint2 rover0)
```

```
(visible_from objective2 waypoint4 rover0)
(visible_from objective2 waypoint1 drone1)
...
```

The problem file in Annex 10 refers to a simple benchmark scenario shown in Figure 48. The scenario is a 3mx3m (with a discretization of 1 m) terrain with three points of interest. Both systems start at *waypoint0*. The drone has to be activated by the rover, but it can freely move in the scenario without path constraints. On the other hand the rover cannot reach all the points because of the obstacles (colored square). Therefore, the two systems should collaborate together to accomplish the mission: visit all the points of interest optimizing the overall path length of the two systems. So, the solver will try to find the overall shortest path to optimize the actions of both systems. The model used to move from one cell to the other as well as to define when an objective can be photographed by the system is shown in Figure 49. The predicates that describe movement and "visibility" of a point of interest are directly derived from the Localization and Real-Time Mapping grid map and arTags list with a Python script.

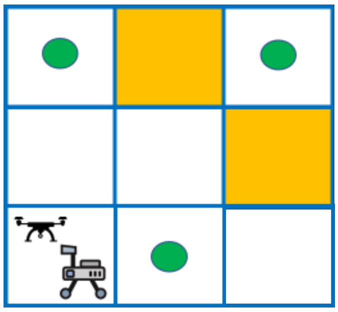


Figure 48: Simple HDDL problem benchmark (code in Annex 10)

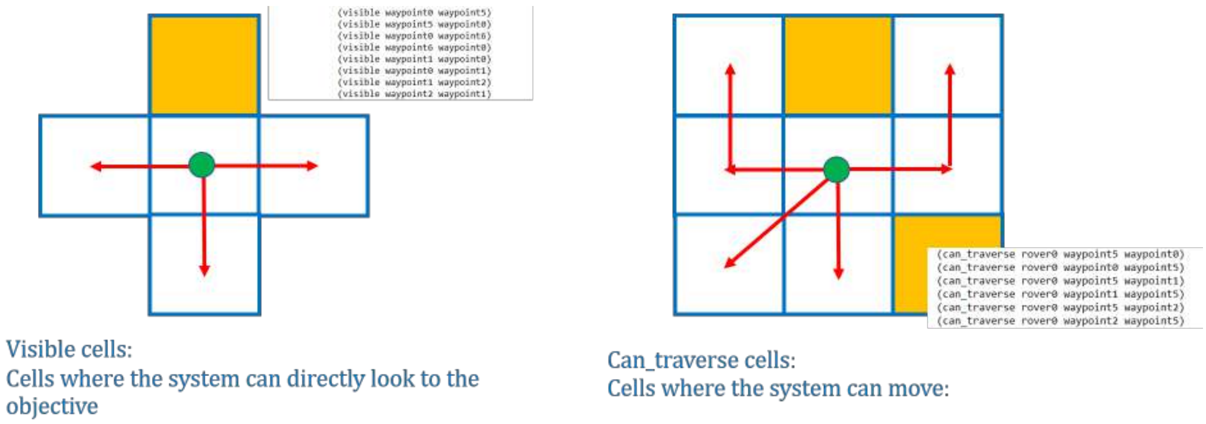


Figure 49: Movement and targeting model in the HDDL problem file (code in Annex 10)

Nevertheless, writing and testing HDDL files is usual quite a challenging task. Therefore, the analysis started from the mission analysis functional layer and ended in coding the *domain.hddl* file that can be found in Annex 10.

The final resulting plan for the benchmark in Annex 10 is shown in Listing 3.

Listing 3: Complete plan based on the HDDL problem file of Annex 10

```
0 (make_available drone1)
1 (visit waypoint0 drone1)
2 (navigate drone1 waypoint0 waypoint1)
3 (unvisit waypoint0 drone1)
4 (read_arTag rover0 waypoint0 objective2 camera0)
5 (communicate_arTag_data rover0 general objective2)
6 (take_image drone1 waypoint1 objective2 camera3 fisheye)
7 (communicate_image_data drone1 general objective2 fisheye)
8 (navigate drone1 waypoint1 waypoint4)
9 (visit waypoint4 drone1)
10 (navigate drone1 waypoint4 waypoint6)
11 (unvisit waypoint4 drone1)
12 (read_arTag drone1 waypoint6 objective0 camera2)
13 (communicate_arTag_data drone1 general objective0)
14 (take_image drone1 waypoint6 objective0 camera3 fisheye)
15 (communicate_image_data drone1 general objective0 fisheye)
16 (navigate drone1 waypoint6 waypoint4)
17 (visit waypoint4 drone1)
18 (navigate drone1 waypoint4 waypoint8)
19 (unvisit waypoint4 drone1)
20 (read_arTag drone1 waypoint8 objective1 camera2)
21 (communicate_arTag_data drone1 general objective1)
22 (take_image drone1 waypoint8 objective1 camera2 depth)
23 (communicate_image_data drone1 general objective1 depth)
24 (visit waypoint0 rover0)
25 (navigate rover0 waypoint0 waypoint5)
26 (unvisit waypoint0 rover0)
27 (read_arTag rover0 waypoint5 objective0 camera0)
28 (communicate_arTag_data rover0 general objective0)
29 (take_image rover0 waypoint5 objective0 camera0 depth)
30 (communicate_image_data rover0 general objective0 depth)
31 (navigate drone1 waypoint8 waypoint4)
32 (visit waypoint4 drone1)
33 (navigate drone1 waypoint4 waypoint5)
34 (unvisit waypoint4 drone1)
35 (get_data_from_sensors rover0)
36 (send_system_state rover0 general)
37 (get_data_from_sensors drone1)
38 (send_system_state drone1 general)
```

2.18.0.2 Drone Mapping The drone mapping plan is a well-defined problem. Therefore, after a small workflow analysis based on Figure 50, it was straightforward to code the drone state machine. However, only the part of the autonomous mapping has been implemented. We still need to interface with the drone path planning code. As soon as the test for the drone path planning will end, we will add to the state machine the notion of where to go and how through set-points. We will launch the path planning code and then the mapping state machine as two different entities in the worst situation of not having the drone path planning. The state machine steps will be triggered with the change of altitude of the drone. The state mapping state machine for the drone has been presented in Paragraph 2.15.1.6.

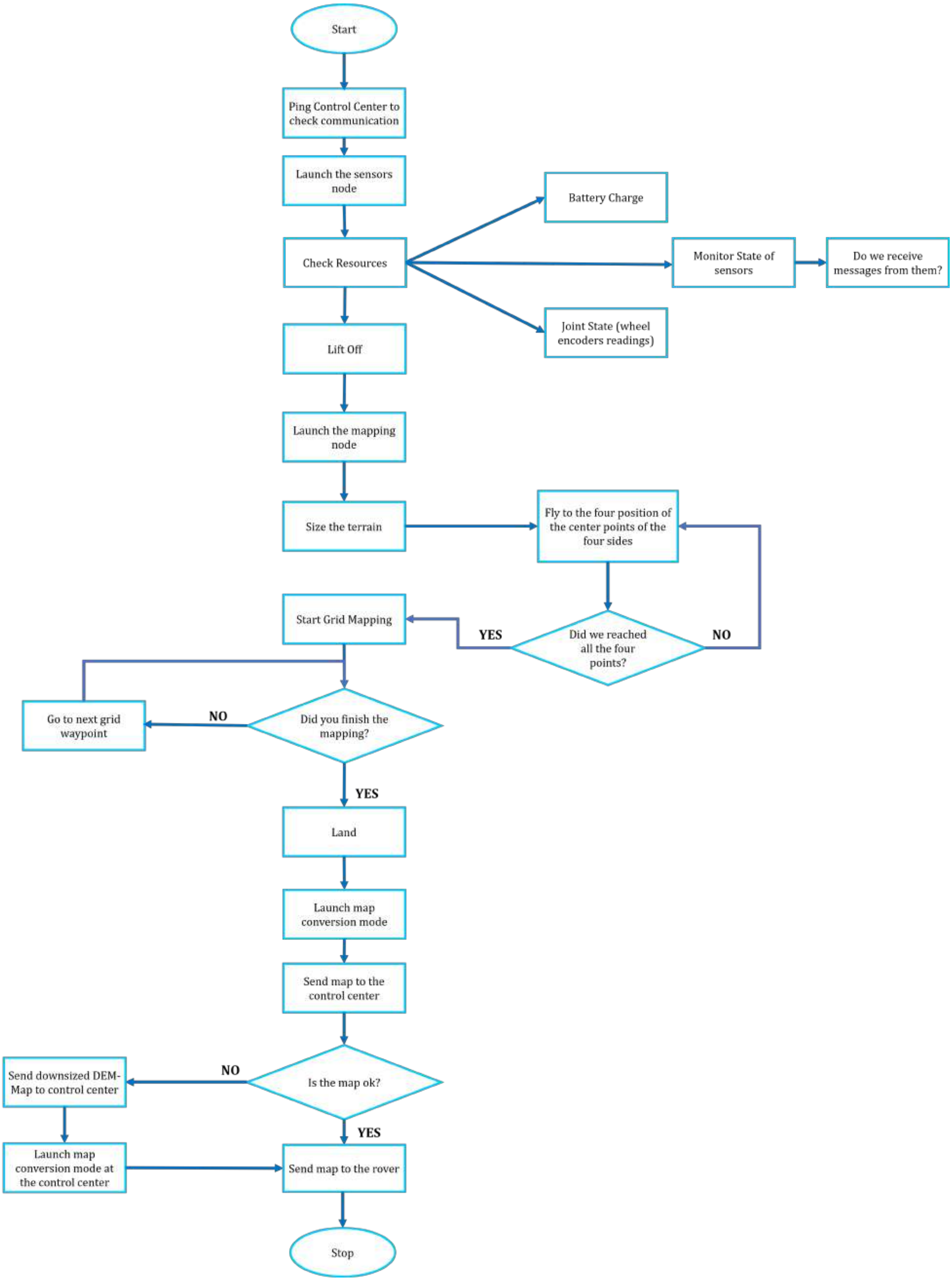


Figure 50: Workflow of action for the drone mapping phase

2.19 State Machine

Starting from the plan defined in Subsection 2.18, it is possible to rapidly code the state machine for the systems. This state machine is the heart of the *Plan Execution Module*. Figure 51 shows the state machine for the rover mission. The state machine for the drone mission will be coded as soon as the drone path planning team has finished its code development. The drone pilot will follow the plan in output from the planner. In this scenario, it does not make much sense to code a state machine.

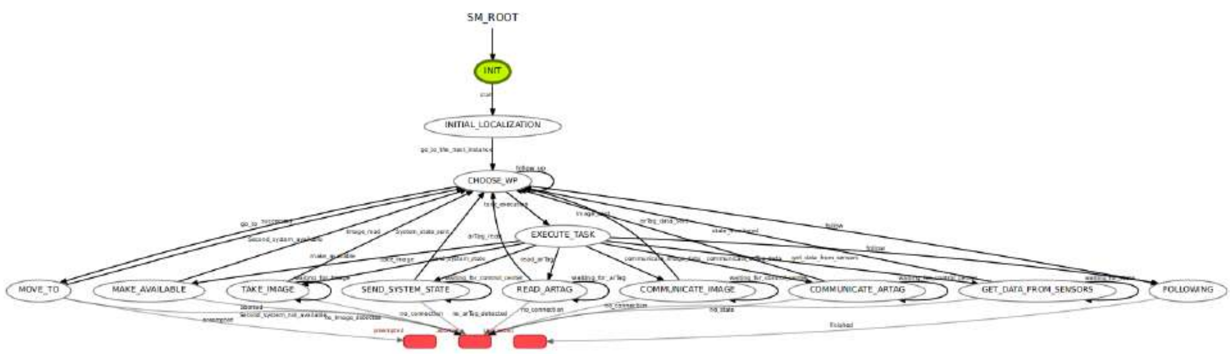


Figure 51: Resulting state machine for the rover task planning

Each entry of the state machine is defined as a Python class and interfaced with ROS. Each entry can have a successful outcome or a failed one. The failed outcome may depend on different preconditions:

- MAKE AVAILABLE: it is the entry automatically turns on the drone and establishes a communication link between rover and drone. The failed outcome is triggered if the drone cannot be ”activated”.
- MOVE TO: it is the entry interfaced with the path planning code. The failed outcome can be generated if:
  - No path to the objective is found.
  - More than two wheels’ motors stop working.
  - The consumed energy is anomalous. The system is consuming too much battery. That usually indicates a not working wheel motor.
- TAKE IMAGE: it is the entry takes a ”screenshot” of the video stream of the depth camera. The fail outcome is triggered if:
  - The system cannot access the depth camera data.
  - If the image is evaluated as ”not compliant” by the control centre.

  The anomaly will be registered with some delay in the latter case: the image should arrive at the control centre, be evaluated, and a ”non-compliant message” is sent to the system.
- GET DATA FROM SENSORS: it is the entry that generates a report on the health of the system with battery level, encoders reading from the last point to point trajectory,

localization data, recorded data frequency from the cameras and LiDAR. The failed outcome is triggered when the system fails to get a stream of data from one sensor.

- SEND SYSTEM STATE: it is the entry that forwards to the control centre a report generated by ”GET DATA FROM SENSORS”. Its failed outcome is triggered if the system cannot communicate with the control centre.
- READ ARTAG: it is the entry in charge of reading and interpreting the arTag information. Its failed outcomes are the same as ”TAKE IMAGE”.
- COMMUNICATE IMAGE: it is the entry that takes care of communicating the image taken by ”TAKE IMAGE” to the control centre. Its failed outcome is triggered if the system cannot communicate with the control centre.
- COMMUNICATE ARTAG: it is a similar entry to ”COMMUNICATE IMAGE”, but it takes care of communicating the arTAG to the control centre. The failed outcomes are the same as ”COMMUNICATE IMAGE”.
- FOLLOWING: it is an entry that brings the system to a safe configuration (far from obstacles) and stop it.

If the rover fails an outcome of the state machine, it stops. It will re-evaluate its resources, change the predicates in the initial conditions of the HDDL problem file and try to find a new plan to visit as many points of interest as possible. If no plan can be found, the autonomous mission is stopped. The system will contact the control center and wait for their intervention. On the other hand, for the drone autonomous mission, in case of any anomaly we will stop the system. It will land, launch its diagnostics and, if it cannot find a solution, contact the control center.

2.19.0.1 Interface of the ROS packages All the entries of the state machine are interfaced with ROS. However, some of them use specific packages already existing in the available ROS libraries.

- MOVE TO: it is interfaced with the ”Move Base” package of ROS [3], as explained in Subsection 2.16.2.
- TAKE IMAGE: it is interfaced with the ”Image View” ROS package[4] .
- READ ARTAG: it is interfaced with the ”Ar Track Alvar” ROS package [5].

[3] http://wiki.ros.org/move_base
[4] http://wiki.ros.org/image_view
[5] http://wiki.ros.org/ar_track_alvar

2.20 Project Verification and Test Planning

2.20.1 Verification Matrix of the Requirements

The envisioned verification means for the requirements of Sections 2.1 and 1.8 are outlined in Table 52, following the guidelines [7]:

- Testing - T: The requirement will be verified with one or a series of tests at ISAE-SUPAERO or with our partners.
- Inspection - I: The requirement will be verified by only inspecting it.
- Analysis of similarity - A: The requirement will be verified running simulations with ROS and Gazebo.
- Review of design - R: The requirement is verified by design documents.

Req. No.	Description	Verification by (T, I, A, R)	Status
RG-01	The implemented algorithm shall be able to provide autonomous decision-making capabilities (level E3 [1])	T	Done, Section 2.17
RG-02	The implemented algorithm shall enable autonomous path planning	T	Done, Section 2.16
RG-03	The systems should use a hierarchical task planner to schedule operations	R	Done, Section 2.17
RG-04	The systems should be able to map and localize in an environment	T	Done, Section2.15
RG-05	The system shall react in the same, predictable manner given identical stimuli	T	Done, Section 2.17
RF-01	The decision layer shall provide go-to command	T and A	Done, test SC-1 of Table 53
RF-02	The decision layer shall be able to decompose high level commands into low-level substasks	R and I	Done, Section2.17
RF-03	The Task Planner should be able to record the "state of health of the system" and based on that it should re-optimize the plan	R and I	Done, Section 2.17
RF-04	The drone shall be able to have an accuracy of 5.5 $\pm$ 0.5 cm on the obstacles it detects and that it registers on its generated map	T and A	Done, test NM-1 of Table 53
RF-05	The drone shall be able to detect 80% of the obstacles with a height superior or equal to 6 cm	R	Done
RF-06	The selected operating system shall be based on Linux	R	Done
RF-07	The implemented code shall be in C++ or Python language	R	Done

RF-08	The rover software shall be able to estimate the required resources to execute the path	R and A	Done, Section 2.17
RF-09	The map areas of the navigation map shall include cost data usable to inform of the most desirable area to be traversed	R	Done, Section 2.15
RF-10	The decision layer shall provide validated plans	R	Done, Section 2.17
RF-11	The systems layer shall gather the information produced	R	Done, Section 2.17
RF-12	The algorithm should provide an executor of the decision-making plan	R	Done, Section 2.17
RF-13	The rover shall have a memory storage to store data before sending it to the control center	R	Done
RF-14	The drone shall include an on-board memory storage	R	Done
RP-01	The design of the implemented software shall allow for future software upgrades	R and A	Done
RP-02	The drone localization shall have an accuracy of 0.5 ± 0.2 m	T and A	Work in progress, Section 2.15
RP-03	The rover localization shall have an accuracy of 0.5 ± 0.2 m	T and A	Done, Section 2.15
RP-04	The modeling framework shall allow timing constraints	R and A	Done, Section 2.17
RP-05	The map generated shall register 85% of obstacles that have a height superior or equal to 6cm	R and T	Done, Section 2.15
RI-01	The implemented algorithms (Localization and Real-Time Mapping, path planning, task planning) shall be able to be interfaced with ROS	R and A and I	Done
RI-02	The systems should be able to communicate with IP addresses with the control center	R and A and I	Done
RS-01	The design of the implemented algorithm shall include items supporting the diagnosis of software anomalies such as log files	R and A	Done, Section 2.17
RS-02	The drone and the rover shall be able to be switched off from the control center	R	Done
RS-03	There should be always be an operator on the field that can act fast and stop the rover or the drone without waiting for the delayed answer from the control center	R	Done

RS-04	A buffer zone of 2 m from the other teams should be considered while planning the Field Campaign set-up	R	Done, Section 2.6
RC-01	The maximum project budget in cash is 3400 euros	R	Defined
RC-02	All the systems should be covered by an insurance up to 1 million CHF	R	Done
RC-03	The rover and the drone shall be able to withstand temperatures between -5℃± 2℃ and +40℃ ± 2℃	R	Done
RC-04	The drone should be able to withstand a wind of 5 m/s (± 0.5m/s)	T	Work in progress
RC-05	The rover should be able to navigate on a terrain with a slope of 10° (± 2°)	T	Done, test UT-2 of Table 53
RC-06	The drone base (structure, OBC, rotors) is given by our sponsors. We have a choice on the type of sensors that we would like to embark on the platform	R and A	Done, Section 2.11
RC-07	The overall rover (structure+sensors) is given to us by our sponsors at DISC - we can suggest minor modifications (e.g. change OBC)	R and A	Done, Section 2.11
RC-08	The drone should recognize specific targets: AR codes	T	Done, Section 2.15
RC-9	The border of the field should be bounded by a rope in order to restrict the drone motion	R	Done, Section 2.6
COM-01	Maximum bandwidth with control room of 2.5 Mb/s	T	Done, test CM-1 of Table 53
COM-02	The drone and the rover shall always remain in communication range (time $\leqslant$ range/c)	T	Done

Table 52: Verification matrix of the requirements

2.20.2 Test Plans

The preliminary set-up of our testing plan is summarized in Table 53.

Test number	Test name	Completed (Yes / No)	Ref. Requirement
DM-1	Decision making capabilities test	YES	RG-05, RF-03,RF-12, RF-10
R-1	Test the system behavior in unknown situations	YES	RG-01
PP-1	Test path planning rover and drone	YES	RG-02
SM-1.T	Test Localization and Real-Time Mapping accuracy of the position (indoor)	YES	RP-02
SM-1.D	Test Localization and Real-Time Mapping capabilities of the drone's sensors (indoor)	YES	RG-04
SM-1.L	Test Localization and Real-Time Mapping capabilities of the drone's sensors (indoor)	YES	RG-04
SM-2.T	Test Localization and Real-Time Mapping accuracy of the drone's position (flying indoor)	YES	RP-02
SM-2.D	Test Localization and Real-Time Mapping capabilities of the drone (flying indoor)	YES	RG-04
SM-2.L	Test Localization and Real-Time Mapping capabilities of the drone (flying indoor)	YES	RG-04
SM-3.T	Test Localization and Real-Time Mapping accuracy of the drone's position (outdoor)	YES	RP-02
SM-3.D	Test Localization and Real-Time Mapping capabilities of the drone (outdoor)	YES	RG-04
SM-3.L	Test Localization and Real-Time Mapping capabilities of the drone (outdoor)	YES	RG-04
SM-4.T	Test Localization and Real-Time Mapping accuracy of the drone's position (flying outdoor)	YES	RP-02
SM-4.D	Test Localization and Real-Time Mapping capabilities of the drone (flying outdoor)	YES	RG-04
SM-4.L	Test Localization and Real-Time Mapping capabilities of the drone (flying outdoor)	YES	RG-04
SM-5	Test Localization and Real-Time Mapping map inaccuracy	YES	RG-04
SC-1	Test move-base command	YES	RF-01
NM-1	Test map creation capabilities	YES	RF-04
CM-1	Test communication between different platforms	YES	COM-01
CM-2	Test communication between the systems	YES	COM-01
WE-1	Test drone in a windy environment	YES	RC-05
UT-2	Test rover performances on a slope	YES	RC-06
DA-1	Drone sensor assessment test	YES	None
AR-1	Point of Interest Recognition	YES	RC-08
AR-2	Drone sensor assessment for Point of Interest recognition	YES	RC-08
AR-3	Drone flight assessment for Point of Interest recognition	YES	RC-08
AR-4	Distance between Points of Interests	YES	RC-08
LOC-1	Improved Localization Rover	YES	RP-03
TPD-1	Mapping State Machine Drone	YES	RG-05
COD-1	Connecting Rover and Drone through Ar tag	YES	RC-08

Table 53: Test Plan to verify the requirements and be successful during the FC

Test number	DM-1
Test name	Decision making capabilities test
Description	The task planning capabilities of the rover should be tested: capability to create a plan and follow it
Parts to be tested	Rover Task Planner
Test facility	ISAE-SUPAERO
Material / equipment	Rover
Persons needed	2
Date of test	May 2021
Test duration	7h
Test outcome	Successful. The state machine of the rover have been coded and tested. We creating a visual gui to help the control center operators to follow the plan.

Test number	R-1
Test name	Test the system behavior in unknown situations
Description	The systems behavior should be tested in a previously unknown environment
Parts to be tested	Rover navigation stack (path planning and Localization and Real-Time Mapping) and rover task planning, drone Localization and Real-Time Mapping and path planning
Test facility	ISAE-SUPAERO, CNES and ONERA
Material / equipment	Rover and drone
Persons needed	2
Date of test	May 2021
Test duration	7h
Test outcome	Successful. The system stops after 3 re-calculation of paths or if it evaluates that it cannot recalculate a feasible plan.

Test number	PP-1
Test name	Test path planning rover and drone
Description	The path planning capabilities of the systems should be tested - capability of estimating a path from point A to point B and follow it for both rover and drone
Parts to be tested	Rover path planning, drone path planning - in known environment and in unknown environment for recognition phase. (*Already tested for the rover*)
Test facility	ISAE-SUPAERO, CNES and ONERA
Material / equipment	Rover and drone
Persons needed	3
Date of test	February at ISAE - March 2021/April 2021 with partners
Test duration	3h
Test outcome	Successful for the rover and drone.

Test number	SM-1.T
Test name	Test Localization and Real-Time Mapping accuracy of the position (indoor)
Description	In stable conditions, the localization capability of the Tracking Camera in an unknown environment has been tested
Parts to be tested	Tracking Camera
Test facility	ISAE-SUPAERO Lab
Material / equipment	Tracking Camera + Optitrack System
Persons needed	4
Date of test	20/11/2020
Test duration	2h
Test outcome	Successful. A maximum of 5cm has been measured between the tracking camera's position output and the position of the sub-millimetric precision optitrack system.

Test number	SM-1.D
Test name	Test Localization and Real-Time Mapping capabilities of the drone's sensors (indoors)
Description	The simultaneous mapping and localization capabilities of the D435i configuration (D435i + T265) have been tested
Parts to be tested	D435 + T265
Test facility	ISAE-SUPAERO Lab
Material / equipment	D435i + T265 + Optitrack System
Persons needed	4
Date of test	20/11/2020
Test duration	2h
Test outcome	Successful. We were able to retrieve a map that satisfies map requirement RF-04. Concerning RF-05, a simplistic version of the conversion module only allowed the detection of obstacles higher than 14 cm.

Test number	SM-1.L
Test name	Test Localization and Real-Time Mapping capabilities of the drone's sensors (indoors)
Description	The simultaneous mapping and localization capabilities of the LiDAR configuration (LiDAR + T265) have been tested
Parts to be tested	LiDAR + T265
Test facility	ISAE-SUPAERO Lab
Material / equipment	LiDAR + T265 + Optitrack System
Persons needed	4
Date of test	20/11/2020
Test duration	2h
Test outcome	Successful. We were able to retrieve a map that satisfies map requirement RF-04. Concerning RF-05, a simplistic version of the conversion module only allowed the detection of obstacles higher than 11 cm.

Test number	SM-2.T
Test name	Test Localization and Real-Time Mapping accuracy of the position (flying indoors)
Description	When flying in an indoor environment, the localization capability of the drone in an unknown environment has been tested
Parts to be tested	Drone's T265
Test facility	ISAE-SUPAERO Lab
Material / equipment	Drone + Optitrack System
Persons needed	4
Date of test	10/12/2020
Test duration	2h
Test outcome	Successful. A maximum of 8 cm was measured between the tracking camera's position output and the position of the sub-millimetric precision optitrack system.

Test number	SM-2.D
Test name	Test Localization and Real-Time Mapping capabilities of the drone (flying indoors)
Description	When flying in an indoor environment, the simultaneous mapping and localization capabilities of the D435i configuration (D435i + T265) have been tested
Parts to be tested	D435 + T265
Test facility	ISAE-SUPAERO Lab
Material / equipment	D435i + T265 + Optitrack System
Persons needed	4
Date of test	10/12/2020
Test duration	2h
Test outcome	Successful. We were able to retrieve a map that satisfies map requirement RF-04. Concerning RF-05, a simplistic version of the conversion module only allowed the detection of obstacles higher than 14 cm.

Test number	SM-2.L
Test name	Test Localization and Real-Time Mapping capabilities of the drone (flying indoors)
Description	When flying in an indoor environment, the simultaneous mapping and localization capabilities of the LiDAR configuration (LiDAR + T265) have been tested
Parts to be tested	LiDAR + T265
Test facility	ISAE-SUPAERO Lab
Material / equipment	LiDAR + T265 + Optitrack System
Persons needed	4
Date of test	20/11/2020
Test duration	2h
Test outcome	Successful. We were able to retrieve a map that satisfies map requirement RF-04. Concerning RF-05, a simplistic version of the conversion module only allowed the detection of obstacles higher than 13 cm.

Test number	SM-3.T
Test name	Test Localization and Real-Time Mapping accuracy of the position (outdoors)
Description	The autonomous localization capability of the drone's tracking camera in an outdoor environment should be tested - a comparison with a differential GPS should be performed to understand the overall reached accuracy
Parts to be tested	Drone's T265
Test facility	ISAE-SUPAERO's (outdoors)
Material / equipment	Drone + GPS
Persons needed	4
Date of test	Mid-June
Test duration	2h
Test outcome	Successful.

Test number	SM-3.D
Test name	Test Localization and Real-Time Mapping capabilities of the drone (outdoors)
Description	Simulating (with the drone in our hands) a flight of the drone in an outdoor environment, the simultaneous mapping and localization capabilities of the D435i configuration (D435i + T265) have been tested
Parts to be tested	D435 + T265
Test facility	ISAE-SUPAERO
Material / equipment	Drone
Persons needed	4
Date of test	18/01/2021
Test duration	2h
Test outcome	In progress. On flat and simple terrains, a dem map that satisfies RF-04 - due to the fact that there was not a lot of features to detect - has been generated. On grass or gravel, a map that satisfies RF-04 on the real obstacles has been generated. However, some virtual obstacles have been generated (about 1 virtual obstacle for 2 m^2) and the post-processing of the grass during the current conversion module created some "holes" (about 10 cm in diameter) in the map where we lose the information. The test requires the completion of the Conversion Module in order to use the full potential of the generated DEM map. On some tests, the tracking camera shows some weaknesses and there are moments where the position is bumping or becomes inaccurate (and thus creates aberration on the generated map). That is why in parallel we are also working to improve the loop closure detection of the Localization and Real-Time Mapping algorithm.

Test number	SM-3.L
Test name	Test Localization and Real-Time Mapping capabilities of the drone (outdoors)
Description	Simulating (with the drone in our hands) a flight of the drone in an outdoor environment, the simultaneous mapping and localization capabilities of the LiDAR configuration (LiDAR + T265) have been tested
Parts to be tested	LiDAR+T265
Test facility	ISAE-SUPAERO
Material / equipment	Drone
Persons needed	4
Date of test	18/01/2021
Test duration	2h
Test outcome	Successful.

Test number	SM-4.T
Test name	Test Localization and Real-Time Mapping accuracy of the position (flying outdoors)
Description	When flying, the autonomous localization capabilities of the drone's tracking camera in an outdoor environment should be tested - a comparison with a differential GPS should be performed to understand the overall reached accuracy
Parts to be tested	Drone's T265
Test facility	ISAE-SUPAERO
Material / equipment	Drone + GPS
Persons needed	4
Date of test	Mid-June 2021
Test duration	2h
Test outcome	Successful.

Test number	SM-4.D
Test name	Test Localization and Real-Time Mapping capabilities of the drone (flying outdoors)
Description	Simulating (with the drone in our hands) a flight of the drone in an outdoor environment, the simultaneous mapping and localization capabilities of the D435i configuration (D435i + T265) have been tested
Parts to be tested	D435 + T265
Test facility	ISAE-SUPAERO and ONERA
Material / equipment	Drone
Persons needed	4
Date of test	Mid-June 2021
Test duration	2h
Test outcome	Successful.

Test number	SM-4.L
Test name	Test Localization and Real-Time Mapping capabilities of the drone (flying outdoors)
Description	When flying in an outdoor environment, the simultaneous mapping and localization capabilities of the LiDAR configuration (LiDAR + T265) have been tested
Parts to be tested	LiDAR + T265
Test facility	ISAE-SUPAERO and ONERA
Material / equipment	Drone
Persons needed	4
Date of test	Mid-June 2021
Test duration	2h
Test outcome	Successful.

Test number	SM-5
Test name	Test Localization and Real-Time Mapping map inaccuracy
Description	A method shall be designed in order to test the proportion of virtual (nonexistent) obstacles on the generated 2D Occupancy Grid
Parts to be tested	2D Occupancy Grid
Test facility	ISAE-SUPAERO
Material / equipment	2D Occupancy grids from previous tests
Persons needed	1
Date of test	Mid-June 2021
Test outcome	Successful.

Test number	SC-1
Test name	Test move-base command
Description	Both systems should be tested in order to see if they can execute simple displacement commands (e.g move 1 m and rotate 20°)
Parts to be tested	Rover and drone mobility systems
Test facility	ISAE-SUPAERO
Material / equipment	Rover and drone
Persons needed	5
Date of test	Mid-December 2020/Beginning of January 2021
Test duration	3h
Test outcome	Partially Successful. We gave the rover go-to command and the system was able to perform those commands.

Test number	NM-1
Test name	Test map creation capabilities
Description	The ability to provide a good outdoor map should be verified for the drone
Parts to be tested	Drone sensors
Test facility	ISAE-SUPAERO
Material / equipment	Drone
Persons needed	3
Date of test	Mid-January 2021
Test duration	1h
Test outcome	Successful. We were able to create the outdoor map.

Test number	CM-1
Test name	Test communication between different platforms
Description	A series of tests should be conducted to assure that the systems will be able to communicate with the control center - the first test was conducted to connect three computers running UBUNTU together
Parts to be tested	Telecommunication links
Test facility	ISAE-SUPAERO
Material / equipment	3 PCs running UBUNTU 16.04 and 18.04 and ROS
Persons needed	Telecommunication team and DISC team
Date of test	20/10
Test duration	3h
Test outcome	Successful communication between the three systems running the FC-like communication set-up (20 Mbit/s and 2.5 s of delay).

Test number	CM-2
Test name	Test Communication between different platforms
Description	A series of tests should be conducted to assure that the systems will be able to communicate with the control center - a second test should be conducted directly on the rover and drone OBC and on the computer that we will bring to the FC
Parts to be tested	Telecommunication links
Test facility	ISAE-SUPAERO
Material / equipment	3 PCs running UBUNTU 16.04 and 18.04 and ROS
Persons needed	telecommunication Team and DISC team
Date of test	Mid-February 2021
Test duration	3h
Test outcome	Is possible to communicate between the systems Linux interface.

Test number	WE-1
Test name	Test drone in a windy environment
Description	The capacity of the drone to ”resist” to a constant wind and still be able to localize in an environment and create a map should be tested
Parts to be tested	Drone control system
Test facility	ISAE-SUPAERO
Material / equipment	Drone and fans to simulate wind
Persons needed	3
Date of test	April 2021
Test duration	2h
Test outcome	Successful.

Test number	UT-2
Test name	Test rover performances on a slope
Description	The capacity of the rover to climb small slope should be tested. From the information gathered from Space Innovation, the testing field should be quite flat. However, in case of a small slope, the system should be still be able to perform its mission
Parts to be tested	Rover mobility system
Test facility	ISAE-SUPAERO
Material / equipment	Rover and ramps in the lab
Persons needed	3
Date of test	December 2020
Test duration	1h
Test outcome	Successful. We estimated the maximum slope traversable by the rover. Further explanations can be found in 2.20.3.

Test number	DA-1
Test name	Drone sensor assessment test
Description	The test will define which sensor we need to successful obtain a map of the system
Parts to be tested	Drone Sensors
Test facility	ISAE-SUPAERO
Material / equipment	Tracking camera, depth camera and LiDAR
Persons needed	Localization and Real-Time Mapping team and DISC testing team
Date of test	Scheduled 9/11 - Moved to 18/11 due to national lockdown
Test duration	3h
Test outcome	Successful. We compared the performance of the depth camera and the LiDAR to generate an indoor map.

Test number	AR-1
Test name	Point of Interest recognition
Description	The test will define the best shape for objects that can be recognised with the AR-tag detection program
Parts to be tested	Drone Sensors
Test facility	ISAE-SUPAERO
Material / equipment	Spherical and cubical objects, AR-tags, Tracking camera, Depth camera
Persons needed	1
Date of test	6/01/2021
Test duration	1h
Test outcome	Successful. We recorded the needed data to start performing POIs recognition.

Test number	AR-2
Test name	AR-tag detection Algorithm Assessment for Point of Interest recognition
Description	Tested the working of ar-track-alvar package with usb-cam node and also with D435 camera of drone.
Parts to be tested	AR-tag detection algorithm
Test facility	ISAE-SUPAERO
Material / equipment	Cubical objects, AR-tags, Tracking camera, Depth camera
Persons needed	Localization and Real-Time Mapping team and DISC testing team
Date of test	24/02/2021
Test duration	8h
Test outcome	Successful. We were able to detect the AR-tags.

Test number	AR-3
Test name	Drone flight assessment for Point of Interest recognition
Description	The test will define the minimum altitude of flight for the drone for AR-tag detection and it will involve testing different dimensions of AR-tags
Parts to be tested	Drone sensors
Test facility	ISAE-SUPAERO
Material / equipment	Cubical objects, AR-tags, Tracking camera, Depth camera
Persons needed	Localization and Real-Time Mapping team and DISC testing team
Date of test	15/03/2021
Test duration	4h
Test outcome	Successful. Determined the maximum distance for detection of 20 cm AR-tags as 5m

Test number	AR-4
Test name	Distance between drone and Points of Interests
Description	Conducted tests using opti track system to find difference between the actual distance between AR tags and Rover and the distance obtained using ROS ar-track-alvar package"
Parts to be tested	AR-tag detection algorithm
Test facility	ISAE-SUPAERO
Material / equipment	Cubical objects, AR-tags, Tracking camera, Depth camera
Persons needed	Localization and Real-Time Mapping team and DISC testing team
Date of test	14/04/2021
Test duration	4h
Test outcome	Successful.

Test number	LOC-1
Test name	Improved Localization Rover
Description	The rover was having localization problems when it was turning too fast. To avoid this inconvenience, we mounted a tracking camera on the rover and start performing visual odometry.
Parts to be tested	Rover Localization Planner
Test facility	ISAE-SUPAERO
Material / equipment	Rover
Persons needed	2
Date of test	May 2021
Test duration	7h
Test outcome	Successful (Paragraph 2.16.2.3).

Test number	TPD-1
Test name	State Machine Mapping Drone
Description	The state machine of the drone have been coded and tested. The different capabilities and processes of Localization and Real-Time Mapping are launched and closed automatically by the drone. We created a visual guide to help the control center operators to follow the plan.
Parts to be tested	Drone task planner
Test facility	ISAE-SUPAERO
Material / equipment	Drone
Persons needed	2
Date of test	May 2021
Test duration	7h
Test outcome	Successful (Subsection 2.20.3).

Test number	COD-1
Test name	Connecting Drone and Rover through Ar tag
Description	The coordinate frame of the map generated by the drone is unknown when the map is sent to the rover. That is why we use ar tag comparison between their detection in the frame of the drone and in the frame of the rover. The goal is to make the rover be able to localize in the map generated by the drone
Parts to be tested	Drone , Rover
Test facility	Isae-supaero
Material	drone rover ar tag communication network
Persons needed	2
Date of test	May 2021
Test duration	10h
Test outcome	Successful (Subsection 2.20.3).

2.20.3 Test Results

Any testing conducted that requires more description than the one in Section 2.20.2 will be shortly addressed in this section. *This section will be mostly used by the Mission Analysis team*, the other teams are most likely to explain their obtained testing results under the algorithms sections 2.15, 2.16, and 2.17.

2.20.3.1 Results Test UT-2 The rover can cross obstacles of maximum height 5.5 ± 0.5 cm, corresponding to the radius of the wheels.

Although tests with rectangular shaped obstacles were performed, there is the need to try a more round shape to mimic the natural shape of rocks. It is worth noting that the rover will not move nor be damaged if it encounters an obstacle higher than 10.5 cm and smaller than the front to back distance between the two wheels, 16 cm.

The wheels adherence is an important factor on the crossing capabilities of the rover.

Concluding, the length of the obstacle is not really an issue for the crossing capabilities of the rover, the most important is the height of the obstacle (at the very beginning of it) which must be under or equal to the radius of the wheel.

This conclusion provides us a lower limit value for map accuracy, so that it can be detected and avoid any critical obstacles for the rover.

2.20.3.2 Results Test SM-1.D and Test SM-1.L Tests SM-1.D refers to the depth camera configuration of test SM-1 and SM-1.L refers to the LiDAR configuration of test SM-1. We placed the sensors on a rotary bench to get a very stable movement. The goal was to test the sensor's capabilities in the easiest situation.

The test results SM-1.D are shown in Figure 52 and Figure 53.

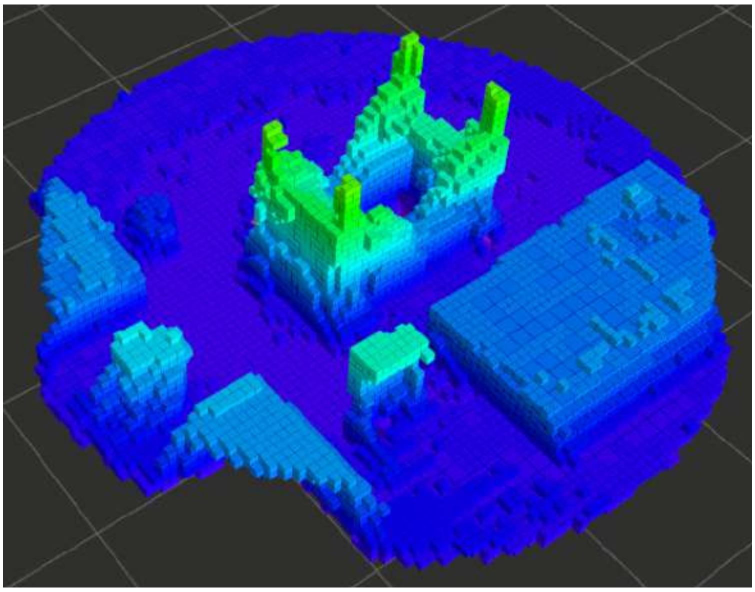


Figure 52: Localization and Real-Time Mapping DEM map test SM-1.D. Resolution 0.03 m

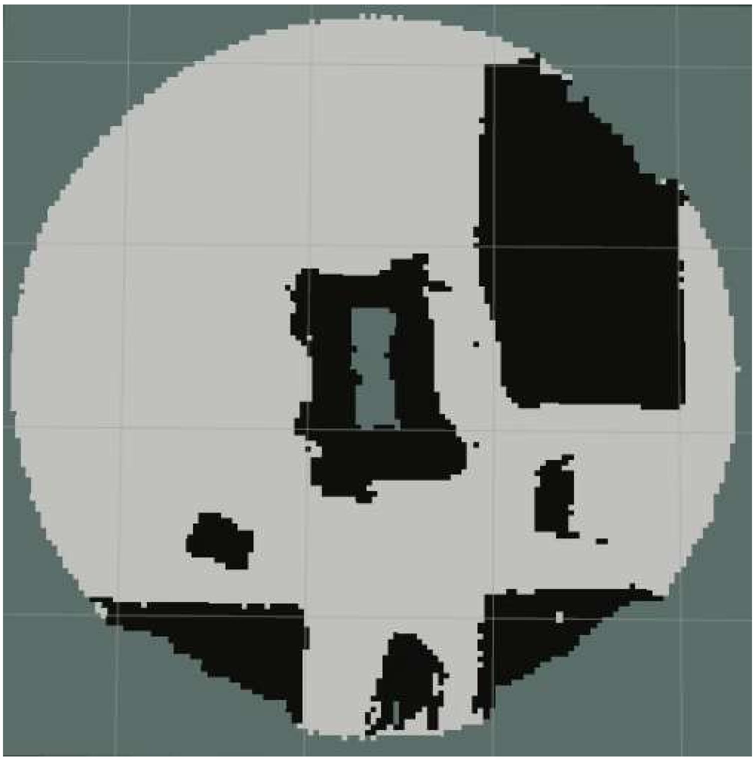


Figure 53: Localization and Real-Time Mapping occupancy grid test SM-1.D. Resolution 0.03 m

The test results SM-1.L are shown in Figure 54 and Figure 55.

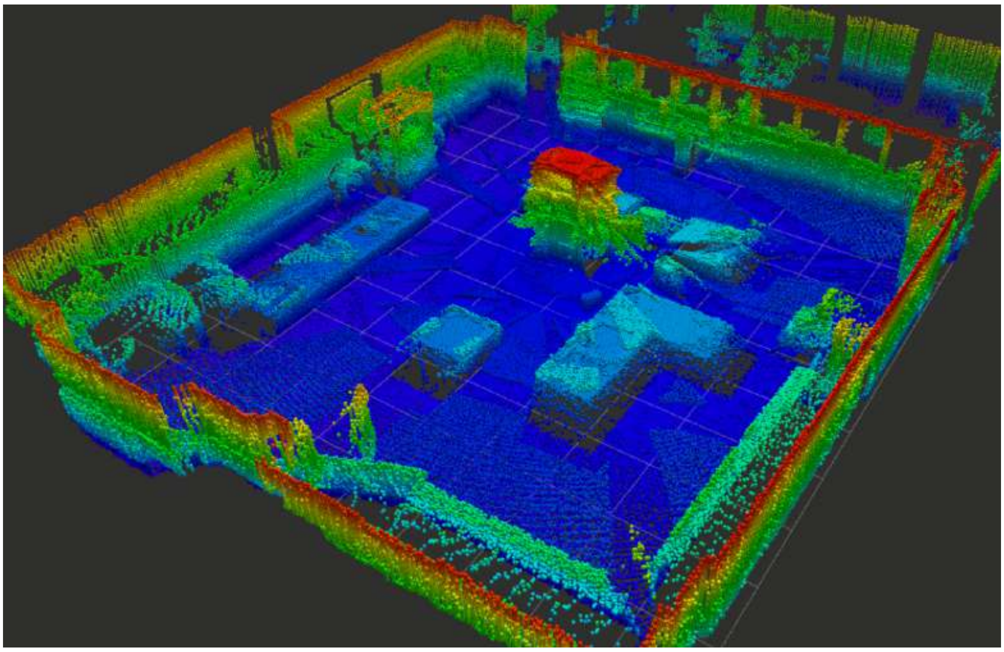


Figure 54: Localization and Real-Time Mapping DEM map test SM-1.L. Resolution 0.03 m

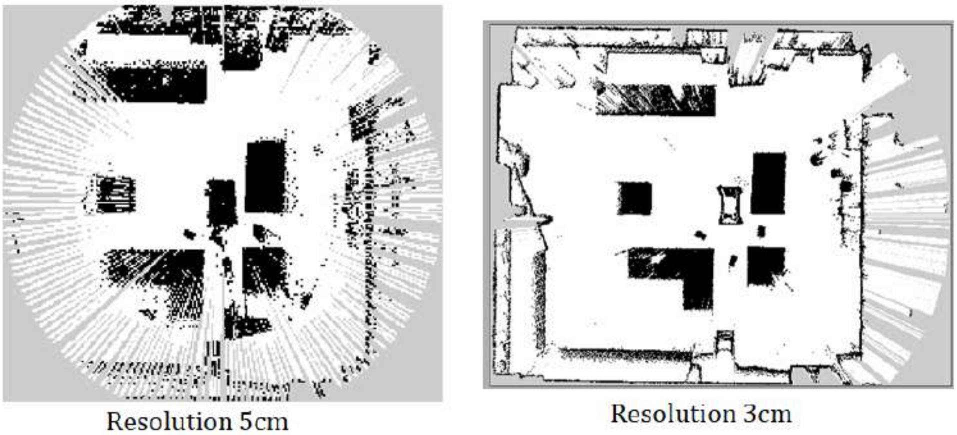


Figure 55: Localization and Real-Time Mapping occupancy grid test SM-1.L before and after depth filtering module implementation. Resolution 0.03 m

2.20.3.3 Results Test SM-2.D and Test SM-2.L Tests SM-1.D refers to the depth camera configuration of test SM-1 and SM-2.L refers to the LiDAR configuration of test SM-2. The drone was flying at a height of 2 m $\pm$ 20 cm. The flight time was 3 minutes and 39 seconds. It was manually piloted and performed a zig zag trajectory over an area of 8x8 m. The trajectory was not the optimal one as the drone repeated some parts of it. The simplistic version of the Conversion Module started showing its limitations, so the Conversion Module work started after that test. The conversion algorithm aims to remove the aberrations that are seen on the Test SM2 results and do the conversion between 3D DEM to 2D occupancy grid. The results of this test are shown from Figure 56 to Figure 58.

Figure 56: Test SM-2 set-up

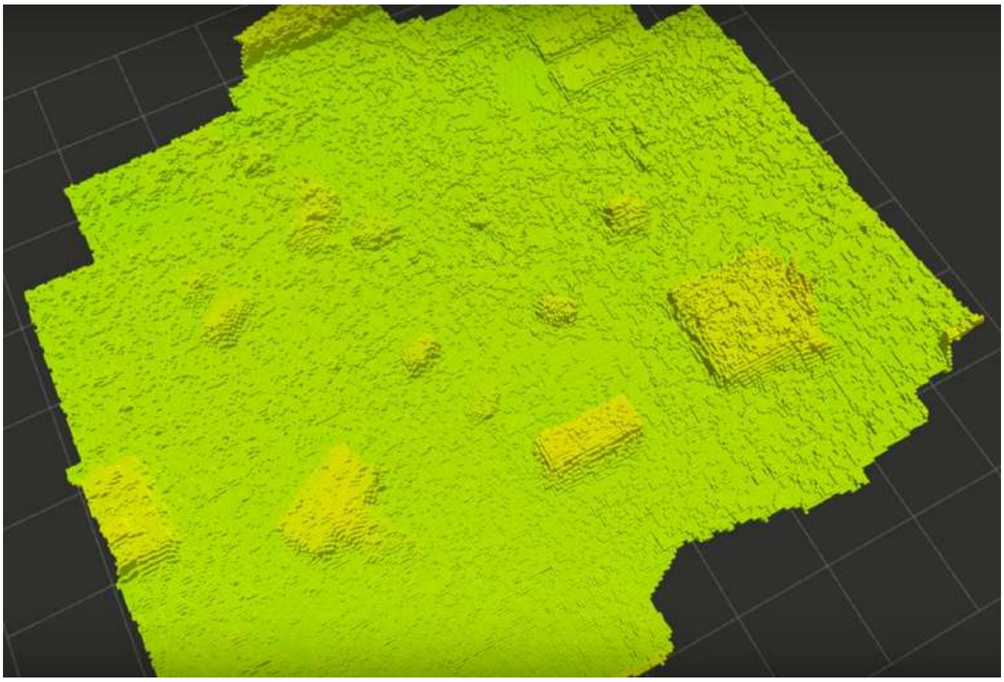


Figure 57: Localization and Real-Time Mapping DEM Map test SM-2.D Resolution 0.03 cm

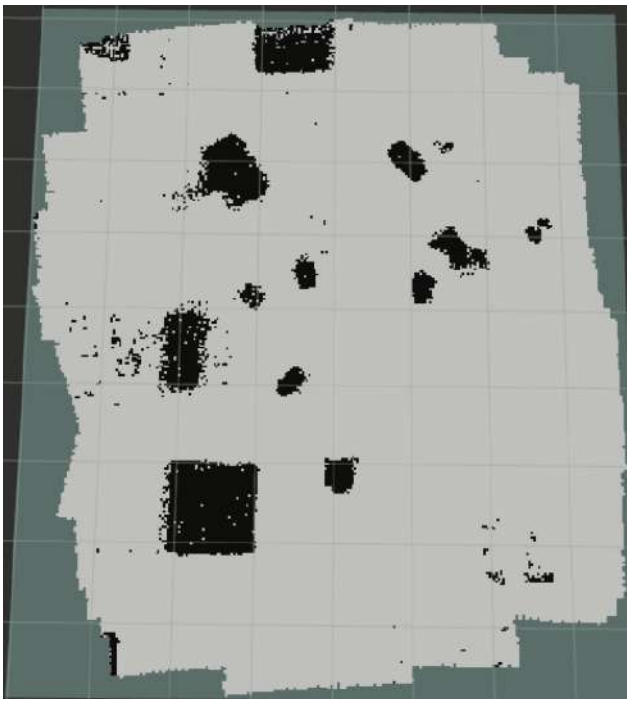


Figure 58: Localization and Real-Time Mapping occupancy grid test SM-2.D. Simplistic version of Conversion Module

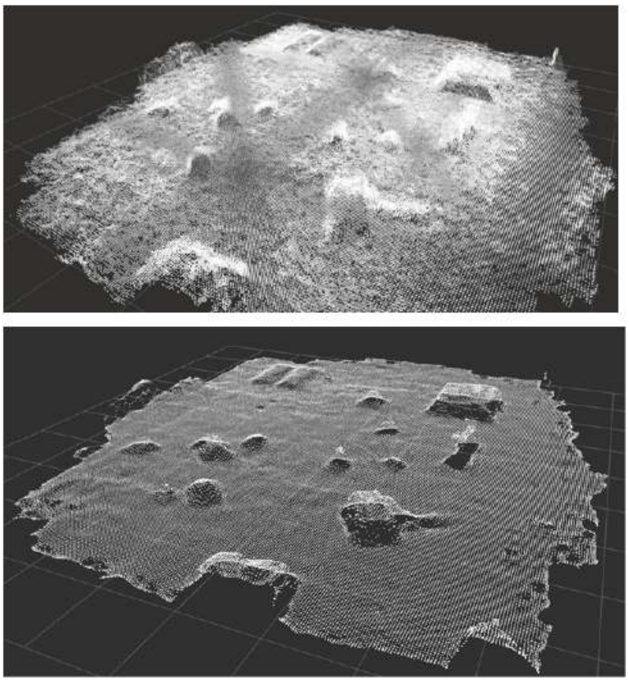


Figure 59: Localization and Real-Time Mapping DEM map test SM-2.D. Before and after the second step of the Conversion Module

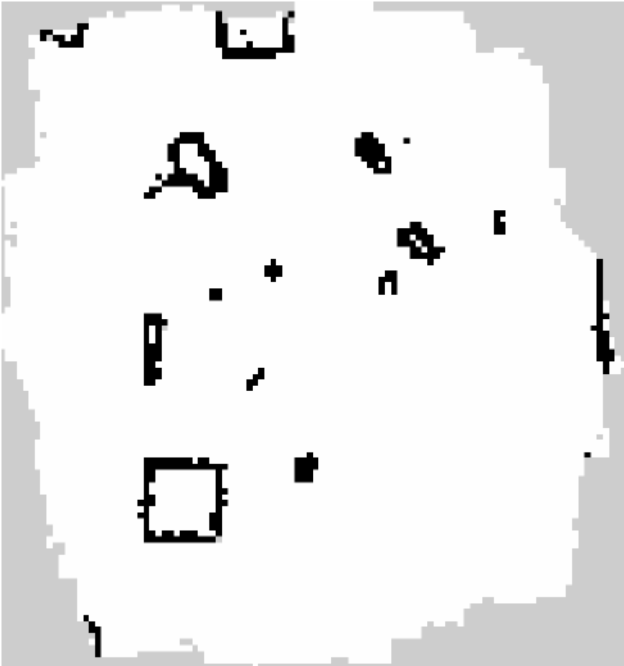


Figure 60: Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, D435i configuration resolution 8cm

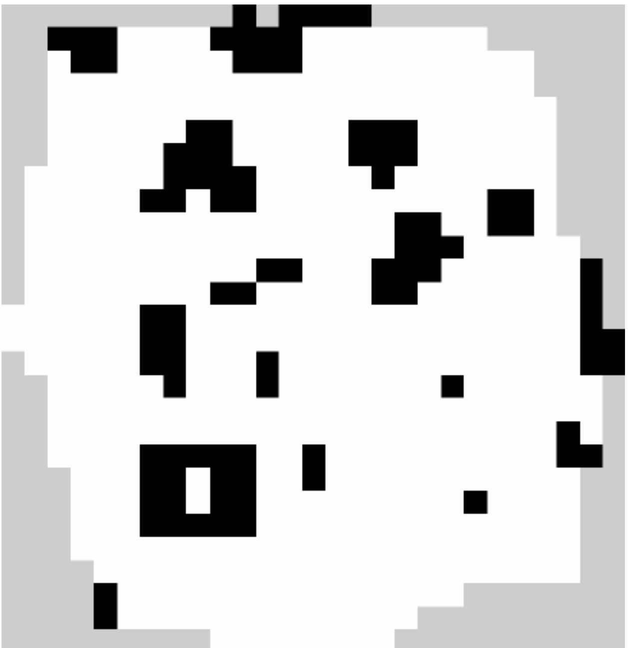


Figure 61: Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, D435i configuration resolution 32cm

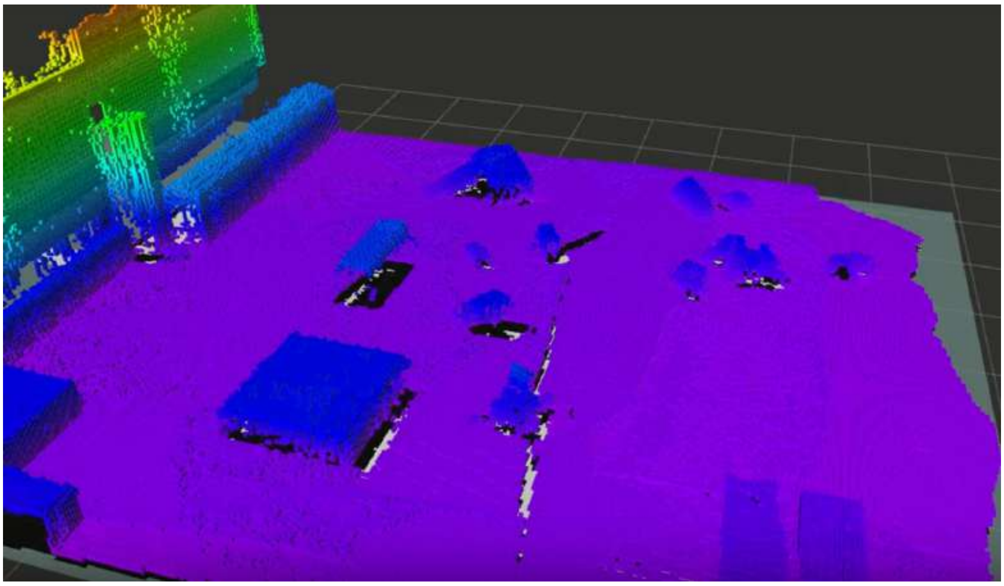


Figure 62: Localization and Real-Time Mapping DEM map test SM-2.L. Resolution 0.03 cm

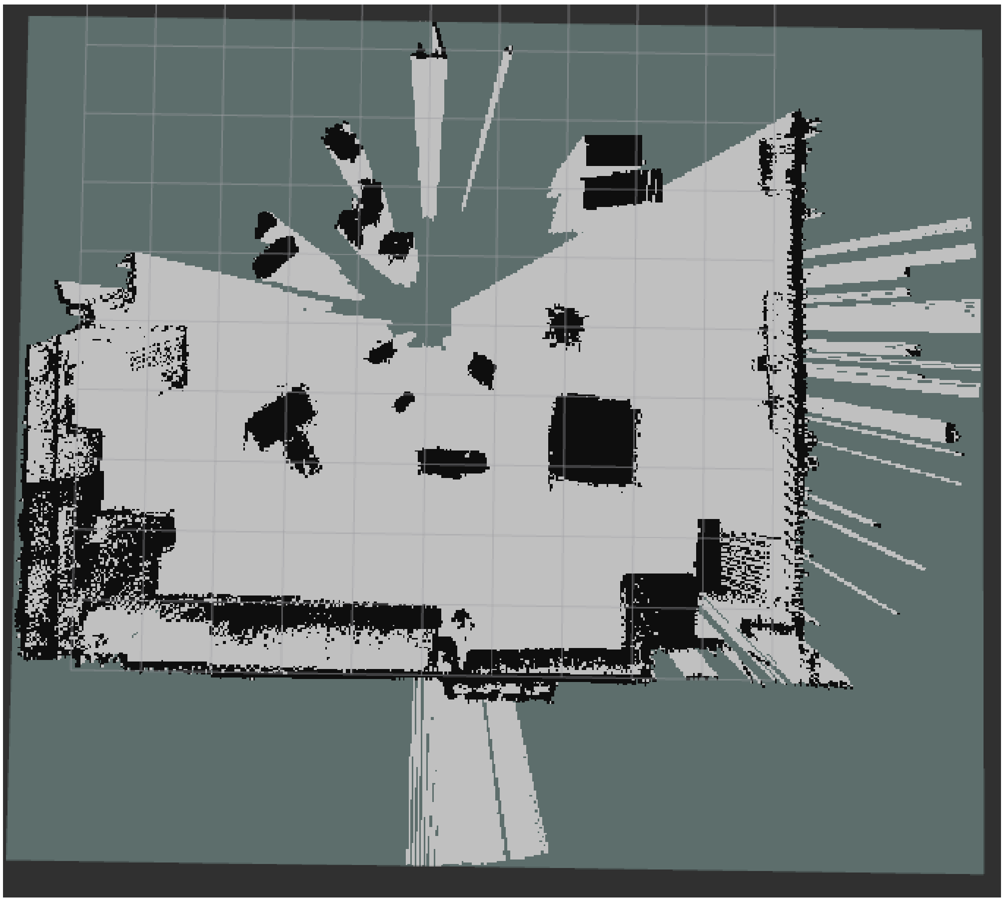


Figure 63: Localization and Real-Time Mapping occupancy grid test SM-2.L. Simplistic version of Conversion Module

Figure 64: Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, LiDAR configuration resolution 8cm

2.20.3.4 Results Test SM-3.D In the outdoor environment, the main obstacle, a bench, was a tough one as it was composed of an alternation of metallic strips and void (allowing the light to get through). The area near the bench was covered with no or little grass about 5 cm max without hole effect. The area of the "loop" was all covered with grass, from 5 cm to more than 10 cm (with dry leaves). The results of the test are shown from Figure 65 to Figure 69.

Figure 65: Localization and Real-Time Mapping test SM-3.D. Bench picture from D435

Figure 66: Localization and Real-Time Mapping test SM-3.D. Grass picture from D435

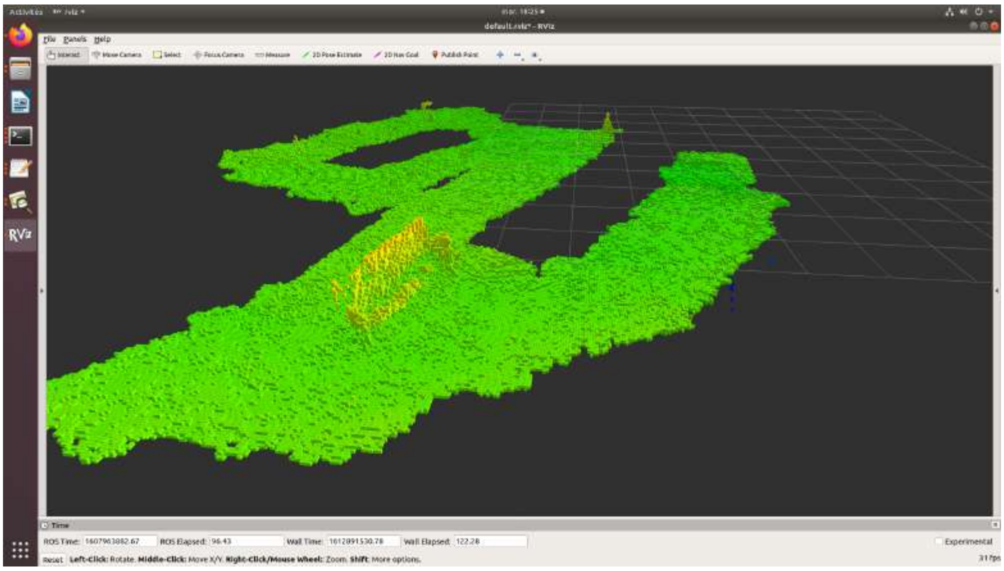


Figure 67: Localization and Real-Time Mapping DEM map test SM-3.D. Resolution 0.03 cm

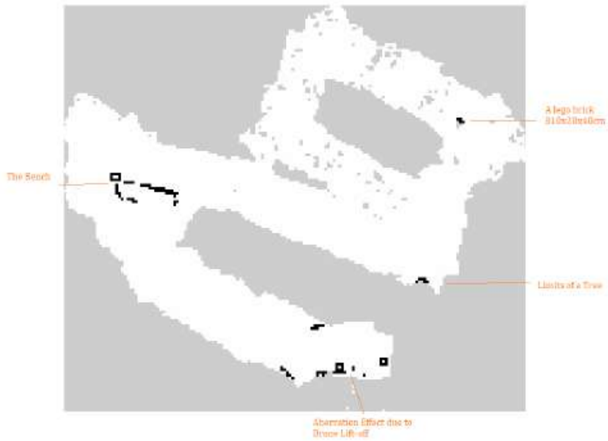


Figure 68: Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, D435i Configuration Outside, resolution 8cm

Figure 69: Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, D435i Configuration Outside, resolution 32cm

2.20.3.5 Results Test SM-3.L Even if the LiDAR was better than the D435 at detecting the bench on the 3D DEM map level, the LiDAR did not react well to the Conversion Module. We suspect it is because the density of points on the 3D DEM map of the LiDAR is less than the density obtained with the D435. The depth camera creates a 3D point cloud that is thick (the ground has several layers) when the LiDAR's point cloud is thinner. The Conversion Module downsizes the 3D point cloud and tries to identify which point to keep for each column. On top of that, the LiDAR is an addition of parallel rays (when the drone is not rotating) that tends to insert some void between elements (1 ray of LiDAR scan - 1 ray of void - 1 ray of LiDAR scan). The downsizing of the 3D points cloud increases the amount of void area. Then, the conversion module smooths the obtained layer with statistical denoising. That statistical denoising decreases the scattering. Therefore, it amplifies the void areas by pushing back points from less dense areas to denser areas. That is how the hole effect might be generated. The origin of that behaviour is the noisy character of grass. LiDAR is more sensible to noise than the depth camera. It is because the depth camera has overlapping successive scans when the LiDAR scans do not overlap a lot. The results of the test are shown from Figure 70 to Figure 71.

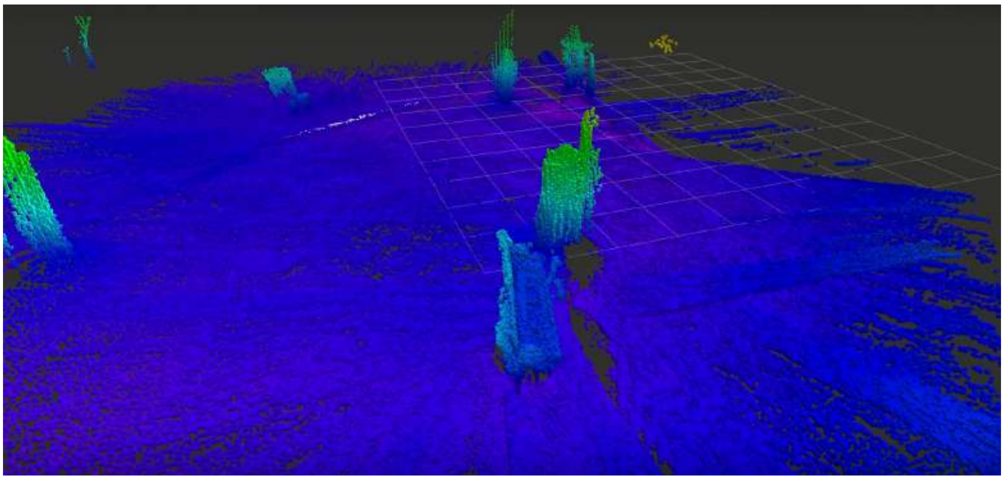


Figure 70: Localization and Real-Time Mapping DEM map test SM-3.L. Resolution 0.03 cm

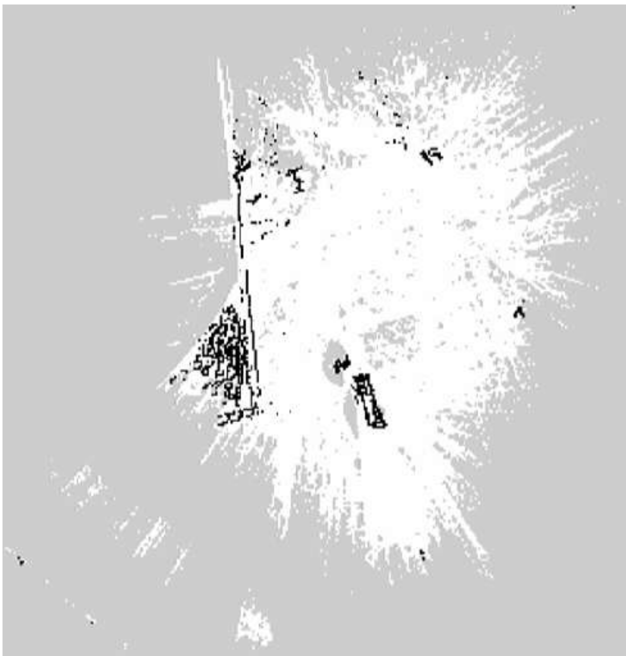


Figure 71: Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, LiDAR Configuration Outside, resolution 8cm

2.20.3.6 Tests conducted for AR Tag recognition

- Test of ar_track_alvar package with usb_cam_node and also with D435 camera of drone.
- Printed AR tags and chose the appropriate size for detection using ROS package ar_track_alvar.

- Test to measure the maximum distance of recognition of AR tags. Use of carton box of size 38x25x22 for the test.
- Tests using opti track system to find difference between “actual distance between AR tags and Rover” and “distance obtained using ROS ar_track_alvar package”.
- Implementation of the link between Opti track system [World] reference frame and the Drone camera D435 [Reference frame] in ROS using TF package.
- Test of the Localization and Real-Time Mapping code integrated with AR Tag detection code and only detection. The final code is not complete as subscription to Ar_pose_marker is pending.
- Detection of the ARtags using the drone frame of reference.
- Detection of the ARtags using the rover frame of reference.

2.20.3.7 Test Merging Frames Rover-Drone The AR tags detected both by the drone and the rover are used to compute the transformation between the coordinate frame of the rover and the coordinate frame of the drone. It is described in detailed in section 2.15.4. The test set-up is shown in Figure 72. The goal is to map the environment and detect as many ar tags as possible with the drone. Then, the database of the detected ar tags and the generated occupancy map are sent to the rover. The rover then is used to detect at least one ar tag in its local environment (by only rotating on itself), the more the better. With the comparison of its own database of detected ar tags and the database of the drone, it computes the transformation between its own coordinate frame and the coordinate frame that was used to generate the occupancy map. If done successfully, it enables the rover to localize itself in the map generated by the drone. That links enables to make the two robotic systems connected, and make them able to collaborate. Indeed, with this established link, any information retrieved by a robotic system is now understandable by the other robotic system. At the moment, the occupancy map of the environment has been generated by the finished Localization and Real-Time Mapping algorithms, the ar tags have been detected and saved to a database by the drone. Both the occupancy map and the database have been sent to the rover. The algorithm for ar tag detection has been implemented and the rover. The algorithm that compute the link between the two frames thanks to the database of the drone and the rover has been implemented on the rover (and will be implemented also on the drone). What is left to do is to use the rover to detect ar tag in its local environment when it is turned on, and to run the code that computes the link between the two frames.

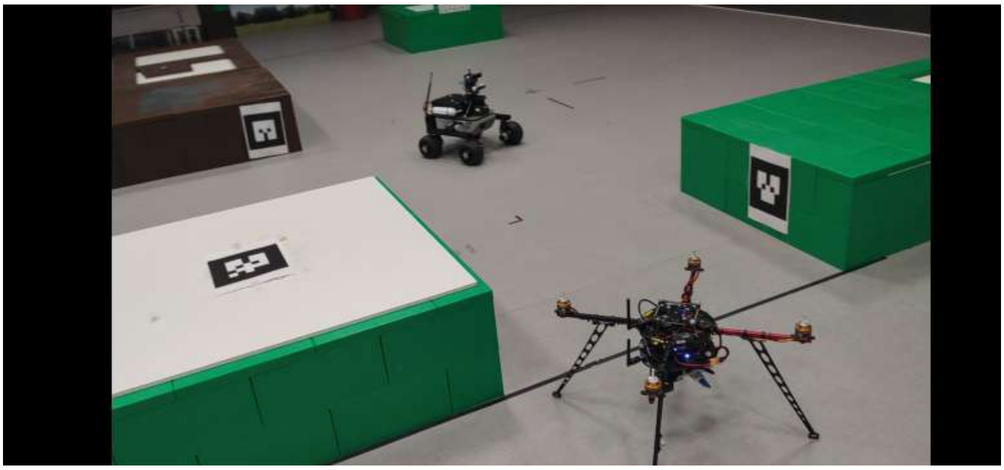


Figure 72: Set-up of the test of Merging Frames of the two robotic systems through features detection

2.20.3.8 Tests New rover localization, LOC-1 Seen the localization problems of the rover when turning to rapidly, we decided to install a tracking camera to correct the data of the EKF. The results are quite promising: we improved our localization of 30% of linear paths, and of more than 50% while turning rapidly in respect to the old EKF (wheel odometry + IMU). The video of the testing results can be found at: `https://drive.google.com/file/d/1v9a7znalboiBoNZ-ouLtE6hnlodztDsv/view?usp=sharing`.

2.20.3.9 Rover State Machine, DM-1 The rover state machine have been coded and interfaced with the path planning and Localization and Real-Time Mapping code. The first test was successful and the output state machine can be seen in Figure 73. The system is able to follow the plan and move autonomously to the different goals given in output.

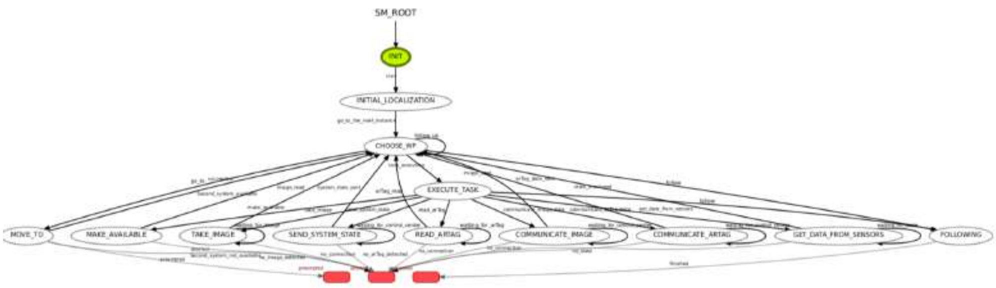


Figure 73: State machine of the Rover Path-planning

2.20.3.10 Drone Mapping State Machine, TPD-1 Regardless the delays we had on the drone path planning - the team decided to already start coding the mapping state machine. Up till now, the state machine is a sequence of ROS launch files, ROS services and python codes that are launched automatically at different triggering situations. A visual graph of the state machine is shown in Figure 74.

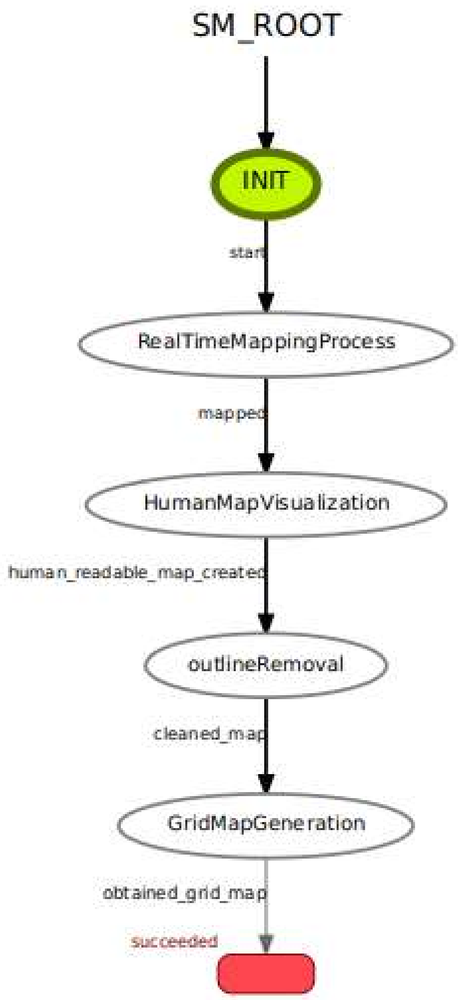


Figure 74: State machine of the Drone mapping phase

2.20.4 Testing and Operations Checklists

In Figure 75 we show the overall timeline of the ConOps during the Field Campaign.

The testing checklists for each ConOps are presented in the following tables:

- Table 58 for set-up of the field,
- Table 59 for the operations before testing,
- Table 55 for drone mission operation,
- Table 56 for rover mission operation,
- Table 57 for the drone mapping,
- Table 60 for control room operation,
- Table 61 for end/beginning of the day operations.

The currently empty ”Check” cells of these tables will be filled in by the team as the Field Campaign progresses.

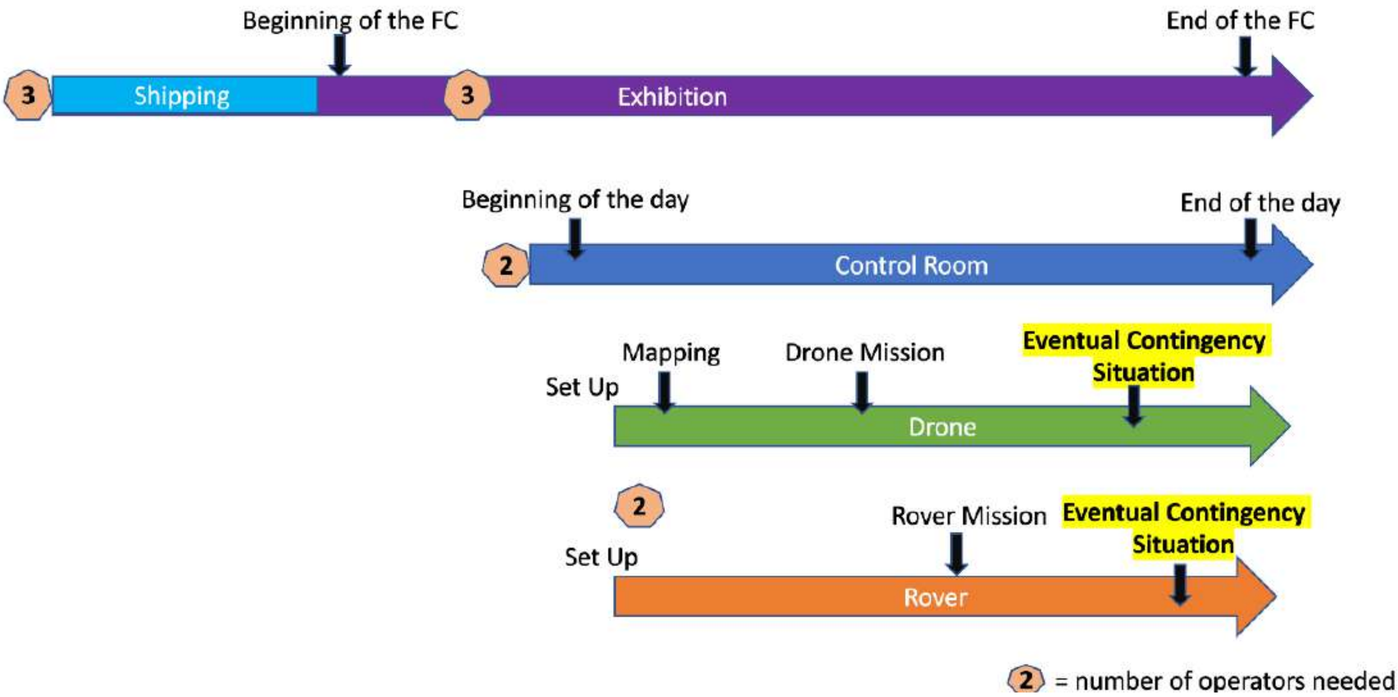


Figure 75: ConOps timeline

The following Safety checklist/Environment Checklist 54 should be use before the following test : Mapping checklist 57, Rover mission checklist 56, Drone mission checklist 55.

Criteria	√	X	Corrective Action	Comment
Safety Check				
Safety guideline of CNES respected ?				
Fire extinguisher nearby ?				
First aid kit ready ?				
Environment Check				
The general area is tidy and free of obstruction and mess				
System is covered from bad weather				
Suspicious materials nearby ?				
Ping the system to check if the wheels or the props are functional.				
Do a telecomanded small up and down with drone or a telecomanded tour with the rover to see that everything is working.				
Optical Check				
Loose wires or other conspicious damage ?				
Check all Components				

Table 54: Safety and Environment Checklist.

Drone Mission Steps	Description	Check
Settings	Set the drone on the zero position and turn it on Set the computer and the operator on the field	
Communication check	"Ping" the drone with the field computer to verify communication	
Receive data	Receive data from the rover (grid map)	
Path Planning algorithm	Run Path Planning algorithm Assumptions during the FC: No obstacle in height at 3m Keep the same flight altitude	
Start of the mission	Drone moves to target Takes a picture Moves to next target	
Landing	Perform landing	
Stop	Complete stop	
Recharge	Operator 2 removes the drone from the field to recharge its batteries	

Table 55: Drone mission checklist

Rover Mission Steps	Description	Check
Settings	Set the rover on the zero position and turn it on Set the computer and the operator on the field	
Communication check	"Ping" the rover with the field computer to verify communication Repeat from the control center	
Send map to the rover	Operator 1 from control center sends map to the rover	
Best task division	The rover will evaluate the best task division	
Send to drone	The rover will send to the drone its plan	
Start of the mission	The rover starts its mission: Compute the global map Move to the target while running obstacle avoidance algorithm (local path planer) Reach the target and take a photo	
Stop	The rover stops at the "zero position"	

Data download	Operator 1 on field goes to the rover and downloads the data on a USB drive	
Recharge	Operator 2 charges the rover batteries and puts the old ones to recharge	
Control Center operator 1	Checks the overall performances of the system: Location: did we localize well the env Pictures: did we capture the right ones Obstacle avoidance success Global path: did we reach the targets, is the path the best one Map: is it accurate enough, did the rover update the map, did the rover recognize new potential sites of interest	
Control Center communication of the fail/pass of the test	Operator 1 will tell if the mission is successful or not, and if needed ask to re-test	

Table 56: Rover mission checklist

Mapping Steps	Description	Check
Settings	Set the drone in the zero portion and turn it on Set the computer and the operator on field	
Communication check	”Ping” the drone with the field computer to verify communication	
Start	Start of the drone	
Lift-off	Perform lift-off	
Target altitude	Reach target altitude of 3 m	
Mapping phase	Run Task Planning, Path Planning, Localization and Real-Time Mapping programs	
”Zero position”	Reach end position ”zero pos”	
Landing	Perform landing	
Stop	Complete stop	
Recharge	Operator 2 removes the drone from the field to recharge its batteries	
Control Center Tasks	Description	Check
Quality/Storage	Operator 1 checks the quality of the map and stores the telemetry data	

Go to next test	If the map is good enough we can head to the next test (green light to the operators on the field)	
Restart the test	If the map is bad, the test should restart: send new mapping parameters to the operators on on the field and red flag	

Table 57: Mapping checklist

Project Set-Up ConOps	Description	Check
Unpacking	Unpacking the material (tools, computers, robots, obstacles)	
Land preparation	Removing larger rocks	
Stations set-up	Setting up the recharging station and the operator station	
Terrain set-up	Set-up of border rope, obstacles and target points Calibration of border reference points	
Packing	Pack the materials at the end of the day in the storage room.	

Table 58: Terrain set-up checklist

Before testing set-up	Description	Check
Checklist	Check the battery level of the drone and the rover before going up to the field	
Packing	Pack all the material (drone, rover, reparation kit, computer, points of interest)	
Transportation	Transport the material to the field (from SUPAERO to CNES)	
Experiment's set-up	Set the points of interest on the field Map their position to be able to repeat the experiment Set up the computer Set the zero point where the systems start Set the recharge point	
Checklist	Check: communication between computer in the field and computer in the control center, communication between computer on the field and drone/rover, rover and drone's respond to a basic ”go to” command	

Table 59: Before testing checklist

Control Room ConOps	Description	Check
Arrival	Set-up the control center at CNES	
Testing 1	Record the testing while waiting for the data to check (depends on the test)	
Testing 2	While the testing is taking place, the battery telemetry will be received	
Receiving of the data	When the data is received, an end check will be performed to see if the set performances are being reached	
Eventual re-testing	Communicate to the field if the test needs to be repeated	

Table 60: Control room checklist

End of the Day Tasks	Description	Check
Take photos	Take photos of the relative position of the objects (drone, rover, points of interest) to be able to reconstruct the field	
Clean the field	Remove all objects (drone, rover, points of interest) from the field and store them in the storage room.	
Beginning of the Day Tasks	Description	Check
Recreate the field	Recreate the field taking out all the objects from the storage room.	

Table 61: End/beginning of the day task checklist

2.21 Project Equipment Needed

The additional equipment needed at the FC is listed on Table 62. Most of this equipment is provided by us or by our sponsors: (i) rover, (ii) drone, (iii) reparation kit, (iv) POIs. The remaining needed equipment is related to the set-up of the testing field for the VFC.

Material	Quantity	Provider	Status	Expected Lead Time
Computers	2 at Control Center	ISAE-SUPAERO Students	Available	From CDR to end of July (after FC)
Rover	1	Sponsor DISC ISAE-SUPAERO	Possessed	From kick-off to end of July (after FC)
Drone	1	Sponsor DISC ISAE-SUPAERO	Possessed	From kick-off to end of July (after FC)
Repair Kit	1	Sponsor DCAS ISAE-SUPAERO	Possessed	June
Terrain Set-Up (Points of Interest, Rope, Landmarks)	1	Sponsor DISC ISAE-SUPAERO Students	Possessed	June
Table and Chair	1 Camping Table	CNES	To be granted	FC
Electrical Sockets	Possible 3 of them	CNES	To be granted	FC

Table 62: Equipment needed for the Field Campaign

2.22 Outreach

The outreach team is in charge of the LinkedIn and Twitter accounts. As the project progress, we release weekly updates of the project through photos and videos. A new website is online (corodro.ae-isae-supaero.fr) and is often updated. Among our activities, we had an interview with Radio-Canada (05/02/2021). We also look forward to contact french media (magazines, newspapers) after the RR and during the FC.

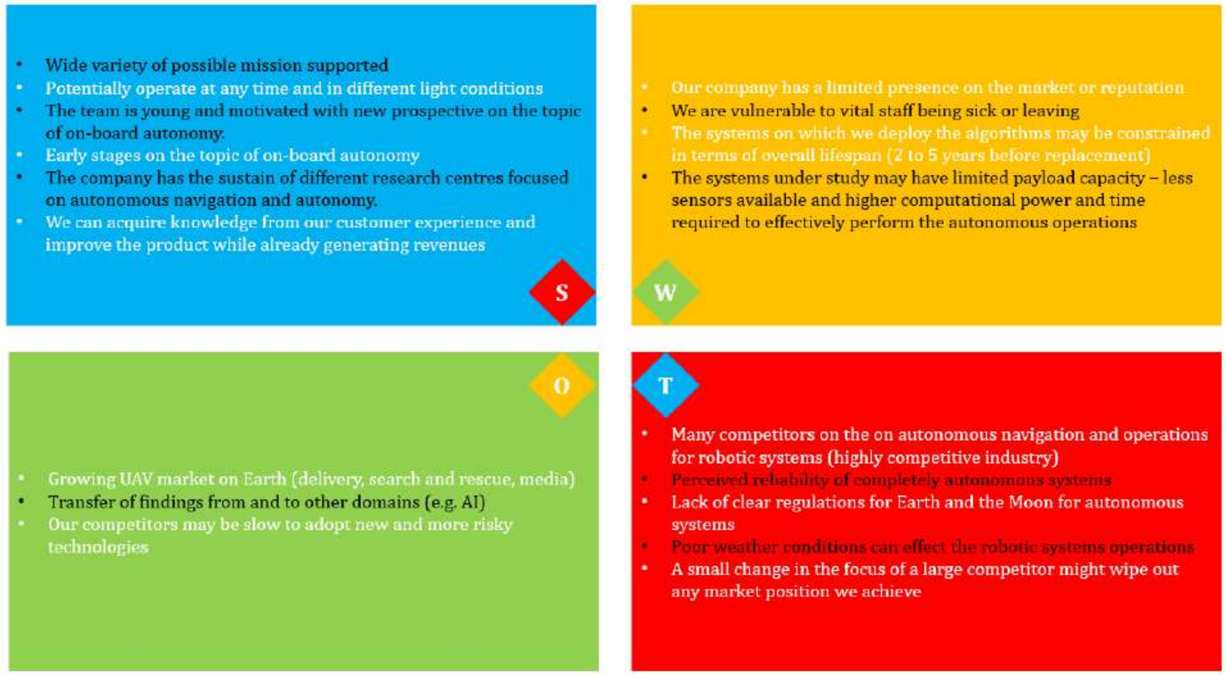
We have six sponsors: most of them are providing us technical knowledge, hardware, testing facilities and exposure. New sponsors are getting contacted for money.

The outreach team is also in charge of the exhibition during the Field Campaign. We are thinking about the different goodies that we will provide during the Exhibition (posters, stickers, pens, etc.). The team contacts are:

- LinkedIn: https://www.linkedin.com/company/igluna-corodro;
- Twitter: @CoRoDro1;
- Website: www.corodro.ae-isae-supaero.fr

2.23 Further Business Model Strategies

2.23.1 SWOT Analysis



2.23.2 Sponsoring and Strategic Partnership

Sponsor	01	Date of signature	22/06/2020
Institution	ISAE-SUPAERO		
Type of support: Academic Space system engineering technical support (DCAS); Autonomous operations (DCAS); Interaction rover/drone (DCAS); Hardware provision (rovers and drones) and technical consulting (DISC); Testing facilities (DISC); Access to meeting rooms and office-like infrastructures (DCAS); Autonomous navigation, Localization and Real-Time Mapping in unknown environment technical support (DEOS); Autonomous navigation and computer vision hardware (DEOS); Promotion of our project in the ISAE-SUPAERO official channels (website and social media).			

Partner	03	Date of signature	06/06/2020
Institution	CNES		
Type of support: Industrial Access to technical expertise (primary interest: autonomous navigation); Access to our rover and testing facilities at CNES; Project visibility through CNES website.			

Partner	02	Date of signature	22/06/2020
Company	Airbus		
Type of support: Industrial Access to expertise (primary interest: collaborative systems).			

Sponsor / Partner	01	Date of signature	24/06/2020
Institution	ONERA		
Type of support: Industrial Access to expertise (primary interest: autonomous operations); Access to keynotes and conferences held by ONERA; Sponsoring in kind (software related to autonomous operations); Promote/advertise on their channels; Access to testing facilities (conditions to be defined after IGLUNA results); Access to open field testing (conditions to be defined after IGLUNA results).			

Partner	04	Date of signature	08/10/2020
Company / Institution	Valispace		
Type of support: Industrial Access to the Valispace mission analysis tool.			

3 VFC Results and Lesson Learned

4 Virtual Field Campaign Testing Results

In this last section, we would briefly explain the main results that we had during the field campaign for the main software subsystems: (i) SLAM, (ii) Path Planning and (iii) Task Planning.

4.1 SLAM

4.1.1 Drone Mapping Phase

During our indoor test in the gymnasium in the beginning of the VFC, tests of the drone mapping phase were unsuccessful at 3m flight height and the drone flight height was lowered to 2m to achieve nice quality of 3D map. The speed of the drone was also set to 0.5m/s. During our final tests outdoor in Esperce, those conditions of 2m height and 0.5m/s did not led to a successful mapping phase. Thus, the height of flight was lowered to 1.5m and the speed to 0.25m/s. The flight duration was about 4minutes. The 3D map is shown in Figure 77, the resolution was 3cm. After the drone landed, it started autonomously the post-process of the 3D map and generated a 2D occupancy grid suited for the rover use. It is shown in Figure 78, the resolution was 8cm. As you can see, the results were satisfying and were meeting our requirements - the drone mapping phase was a success. The video of the mapping phase is available here : `https://drive.google.com/file/d/17fb7JWNFqDch213nDTullePg42_cunmM/view?usp=sharing`. The set-up of the test is shown in Figure **??**. We have already a lot of different ideas how to improve some parts of the mapping algorithms such as : implementing a loop closure module, rewriting some parts that were written quickly due to time limitation, implementing an algorithm that takes into consideration the deformation of the depth scan due to the speed of the drone to be able to make the drone fly faster.

Figure 76: Set up of the Outdoor test in Esperce.

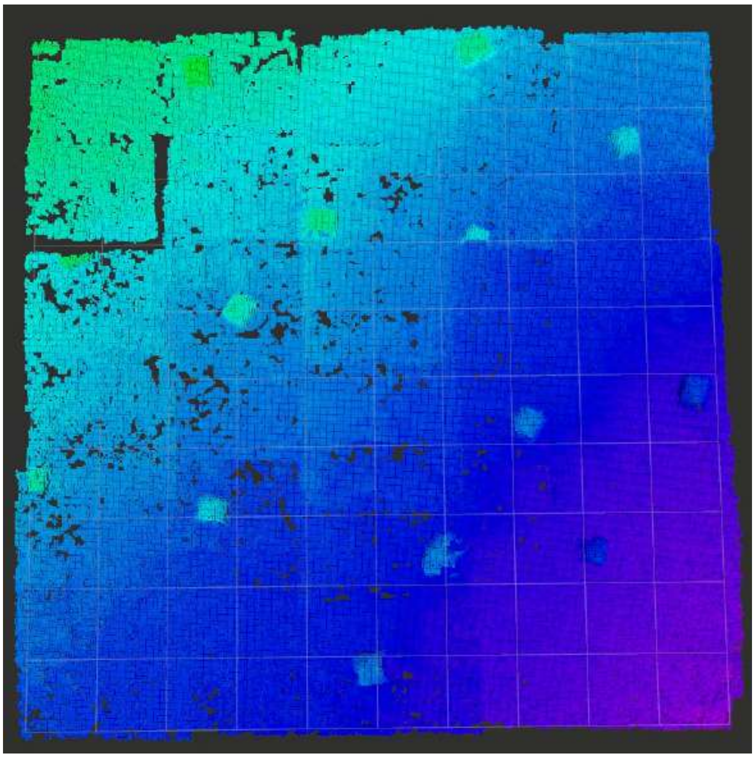


Figure 77: 3D map of Outdoor testing in Esperce. Resolution 3cm

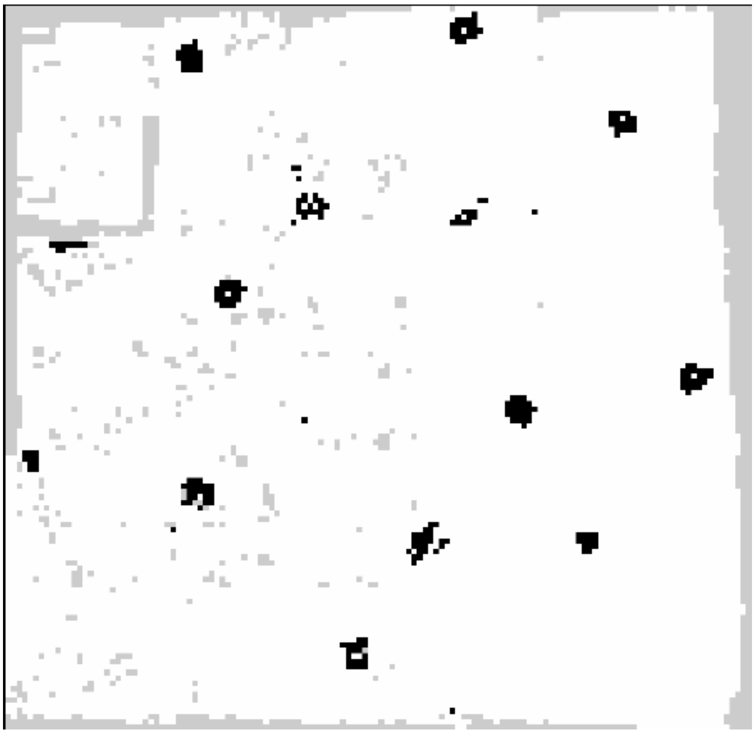


Figure 78: 2D Occupancy grid of Outdoor testing in Esperce. Resolution 8cm

4.1.2 Points of Interest detection

There were 14 AR tags bundles used as points of interest during the field campaign and they were placed randomly on the field. The AR tag detection algorithm embedded on the drone and rover was able to detect all the AR tags and publish the AR Tag bundle identification number, position and orientation information for detecting the unique points of interests on the field. The information of AR tags were further used by the ARTag linker algorithm and task planner algorithm to localise the robotic systems on the map and allow the drone and rover to move towards the points of interest. The plan for the upcoming year is to use AI technology with specifically deep learning methods to detect points of interest such as bright objects and rocks, and find their position and orientation on the field.

4.2 Path Planner

4.2.1 Path Planner Generalities

The drone is controlled in position: the equipped tracking camera gives information of coordinates (x,y) in respect to the ”zero point”. The ”zero point” can be the origin of the coordinate system during the mapping phase or a new position set with the ARTag linker during the mission phase. To be more precise during the mission phase the system can reset its notion of position with respect to the ”point of interest” that it sees. Neither the drone nor the rover uses GPS technology: the only information of position is given by the odometry and SLAM algorithm in the T265 camera. Previous to the field test campaign, we did some tests concerning the precision of the tracking camera, Figure 79. The maximum percentage difference between the data of the optitrack and camera is 2% on height position. That difference is mostly due to the position of the nearest optitrack beacon to the camera. However, we are mostly interested in the coordinates (x,y) of the system 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 very precise. Again for the z-axis there are some differences: the camera is at 30 cm of height with respect to the body of the rover. Therefore, the zero position of the T265 is indeed seen with a 30 cm of displacement by the optitrack.

4.2.2 Virtual Campaign Testing

During the VFC, we had problems with the local path planner of the rover. We didn’t mow the grass, therefore, the LiDAR, the primary sensor used during obstacle avoidance, was ”sensing” obstacles that were not there. Because the terrain was flat and without obstacles, we were able to accomplish the mission using only the global path planner. However, for next year, we will improve the overall design of the local path planner using the cameras inputs as well. Therefore, the systems will be able to navigate more challenging terrains. One of the main focuses of next year IGLUNA will be the coding of the in-house path planner for both the drone and the rover.

4.3 Task Planner

During the VFC, we had the possibility to test the interfaces between the task planner, the SLAM code and the mission analysis code. Even with simple problem files, like the one in Listing 4, we obtained plans of 84 steps, Listing 5. The plan is then interfaced with the hardware with a state machine as shown at the end of Subsection 2.17.

Listing 4: Example problem file during testing.

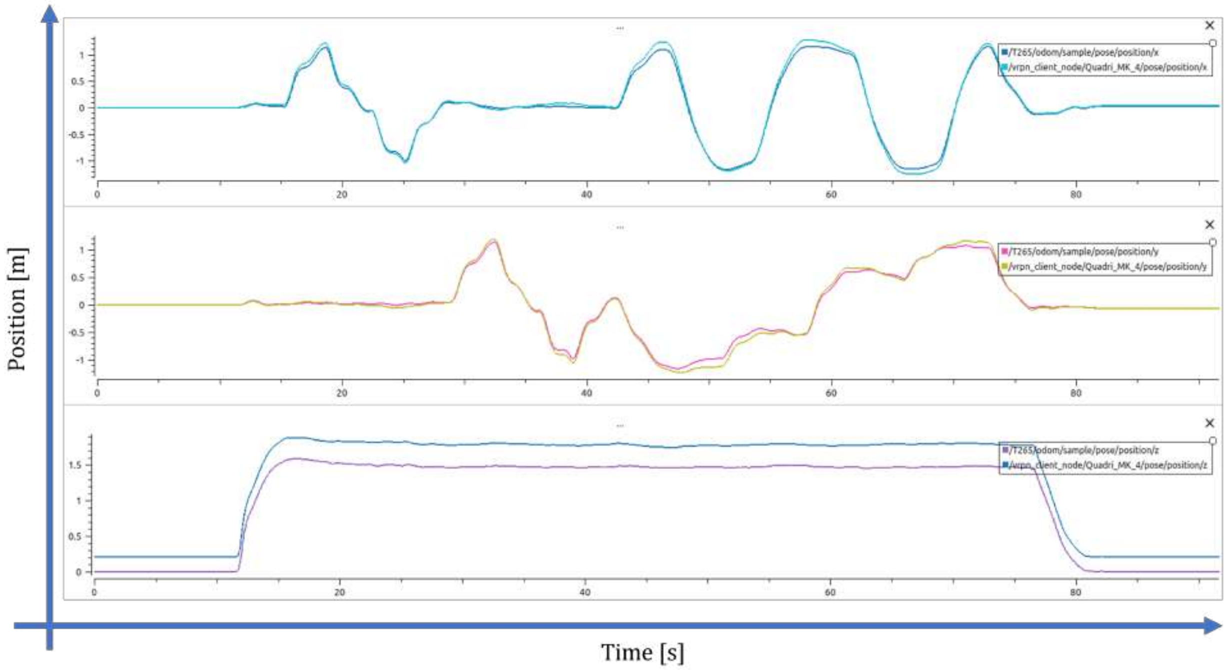


Figure 79: Lab T265 position assessment test for the drone.

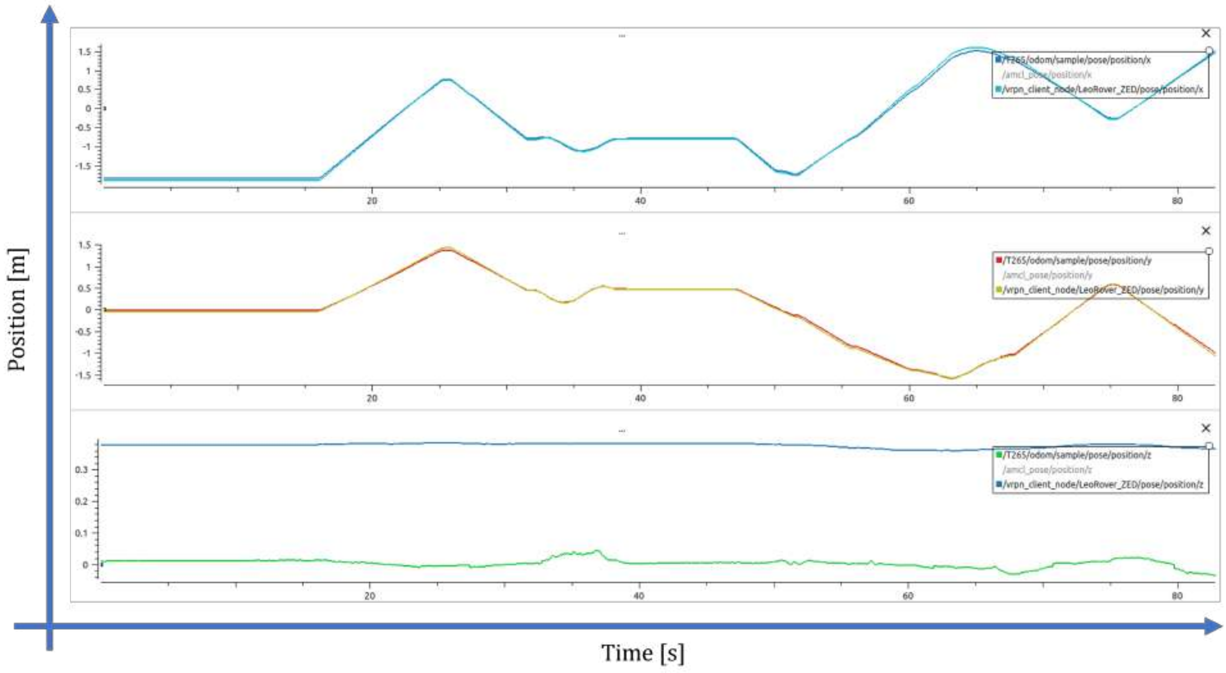


Figure 80: Lab T265 position assessment test for the rover.

```
(define
        (problem pfile1)
        (:domain rover_domain)
        (:objects
```

```
waypoint75 - waypoint
waypoint60 - waypoint
waypoint93 - waypoint
waypoint96 - waypoint
waypoint98 - waypoint
waypoint118 - waypoint
waypoint102 - waypoint
waypoint111 - waypoint
waypoint139 - waypoint
waypoint142 - waypoint
waypoint161 - waypoint
waypoint0_7_1 - waypoint
waypoint0_10_1 - waypoint
waypoint1_10_1 - waypoint
waypoint4_7_1 - waypoint
waypoint7_9_1 - waypoint

objective11 - objective
objective12 - objective
objective10 - objective
objective5 - objective
objective9 - objective
objective8 - objective
objective2 - objective
objective7 - objective
objective6 - objective
objective4 - objective
objective1 - objective


depth - mode
fisheye - mode
rover0 - system
drone1 - system
camera0 - camera
camera1 - camera
camera2 - camera
camera3 - camera
general - control_center

        )
        (:htn
                :parameters ()
                :subtasks (and
                 (task0 (release_second_system drone1 rover0))

                 (task1 (get_image_data objective11))
                 (task2 (get_image_data objective12))
                 (task3 (get_image_data objective10))
                 (task4 (get_image_data objective5))
                 (task5 (get_image_data objective9))
```

```
            (task6 (get_image_data objective8))
            (task7 (get_image_data objective2))
            (task8 (get_image_data objective7))
            (task9 (get_image_data objective6))
            (task10 (get_image_data objective4))
            (task11 (get_image_data objective1))

        )
        :ordering (and
                (< task0 task1)
                (< task0 task2)
                (< task0 task3)
                (< task0 task4)
                (< task0 task5)
                (< task0 task6)
                (< task0 task7)
                (< task0 task8)
                (< task0 task9)
                (< task0 task10)
                (< task0 task11)

        )
)
(:init

        (channel_free general)
        (at rover0 waypoint93)
        (available rover0)

        (equipped_for_imaging rover0)
        (equipped_for_health_monitoring rover0)
        (equipped_for_arTag_reading rover0)

        (at drone1 waypoint93)
        (equipped_for_imaging drone1)
        (equipped_for_health_monitoring drone1)
        (equipped_for_arTag_reading drone1)
        (is_dependent drone1 rover0)
        (available drone1)


        (on_board camera0 rover0)
        (supports camera0 depth)
        (on_board camera1 rover0)
        (supports camera1 depth)

        (on_board camera2 drone1)
        (supports camera2 depth)
        (on_board camera3 drone1)
        (supports camera3 fisheye)
```

```
(can_traverse rover0 waypoint75 waypoint0_6_1)
(can_traverse rover0 waypoint0_6_1 waypoint75)
(can_traverse rover0 waypoint0_6_1 waypoint102)
(can_traverse rover0 waypoint102 waypoint0_6_1)
(can_traverse rover0 waypoint75 waypoint0_7_1)
(can_traverse rover0 waypoint0_7_1 waypoint75)
(can_traverse rover0 waypoint0_7_1 waypoint111)
(can_traverse rover0 waypoint111 waypoint0_7_1)
(can_traverse rover0 waypoint75 waypoint0_8_1)
(can_traverse rover0 waypoint0_8_1 waypoint75)
(can_traverse rover0 waypoint0_8_1 waypoint139)
(can_traverse rover0 waypoint139 waypoint0_8_1)
(can_traverse rover0 waypoint75 waypoint0_9_1)
(can_traverse rover0 waypoint0_9_1 waypoint75)
(can_traverse rover0 waypoint0_9_1 waypoint142)
(can_traverse rover0 waypoint142 waypoint0_9_1)
(can_traverse rover0 waypoint75 waypoint0_10_1)
(can_traverse rover0 waypoint0_10_1 waypoint75)
(can_traverse rover0 waypoint0_10_1 waypoint161)
(can_traverse rover0 waypoint161 waypoint0_10_1)
(can_traverse rover0 waypoint60 waypoint1_8_1)
(can_traverse rover0 waypoint1_8_1 waypoint60)
(can_traverse rover0 waypoint1_8_1 waypoint139)
(can_traverse rover0 waypoint139 waypoint1_8_1)
(can_traverse rover0 waypoint60 waypoint1_9_1)
(can_traverse rover0 waypoint1_9_1 waypoint60)
(can_traverse rover0 waypoint1_9_1 waypoint142)
(can_traverse rover0 waypoint142 waypoint1_9_1)
(can_traverse rover0 waypoint60 waypoint1_10_1)
(can_traverse rover0 waypoint1_10_1 waypoint60)
(can_traverse rover0 waypoint1_10_1 waypoint161)
(can_traverse rover0 waypoint161 waypoint1_10_1)
(can_traverse rover0 waypoint93 waypoint2_10_1)
(can_traverse rover0 waypoint2_10_1 waypoint93)
(can_traverse rover0 waypoint2_10_1 waypoint161)
(can_traverse rover0 waypoint161 waypoint2_10_1)
(can_traverse rover0 waypoint96 waypoint3_7_1)
(can_traverse rover0 waypoint3_7_1 waypoint96)
(can_traverse rover0 waypoint3_7_1 waypoint111)
(can_traverse rover0 waypoint111 waypoint3_7_1)
(can_traverse rover0 waypoint96 waypoint3_10_1)
(can_traverse rover0 waypoint3_10_1 waypoint96)
(can_traverse rover0 waypoint3_10_1 waypoint161)
(can_traverse rover0 waypoint161 waypoint3_10_1)
(can_traverse rover0 waypoint98 waypoint4_6_1)
(can_traverse rover0 waypoint4_6_1 waypoint98)
(can_traverse rover0 waypoint4_6_1 waypoint102)
(can_traverse rover0 waypoint102 waypoint4_6_1)
(can_traverse rover0 waypoint98 waypoint4_7_1)
(can_traverse rover0 waypoint4_7_1 waypoint98)
(can_traverse rover0 waypoint4_7_1 waypoint111)
```

```
(can_traverse rover0 waypoint111 waypoint4_7_1)
(can_traverse rover0 waypoint98 waypoint4_10_1)
(can_traverse rover0 waypoint4_10_1 waypoint98)
(can_traverse rover0 waypoint4_10_1 waypoint161)
(can_traverse rover0 waypoint161 waypoint4_10_1)
(can_traverse rover0 waypoint118 waypoint5_7_1)
(can_traverse rover0 waypoint5_7_1 waypoint118)
(can_traverse rover0 waypoint5_7_1 waypoint111)
(can_traverse rover0 waypoint111 waypoint5_7_1)
(can_traverse rover0 waypoint102 waypoint6_9_1)
(can_traverse rover0 waypoint6_9_1 waypoint102)
(can_traverse rover0 waypoint6_9_1 waypoint142)
(can_traverse rover0 waypoint142 waypoint6_9_1)
(can_traverse rover0 waypoint111 waypoint7_8_1)
(can_traverse rover0 waypoint7_8_1 waypoint111)
(can_traverse rover0 waypoint7_8_1 waypoint139)
(can_traverse rover0 waypoint139 waypoint7_8_1)
(can_traverse rover0 waypoint111 waypoint7_9_1)
(can_traverse rover0 waypoint7_9_1 waypoint111)
(can_traverse rover0 waypoint7_9_1 waypoint142)
(can_traverse rover0 waypoint142 waypoint7_9_1)
(can_traverse rover0 waypoint111 waypoint7_10_1)
(can_traverse rover0 waypoint7_10_1 waypoint111)
(can_traverse rover0 waypoint7_10_1 waypoint161)
(can_traverse rover0 waypoint161 waypoint7_10_1)

(can_traverse drone1 waypoint75 waypoint0_7_1)
(can_traverse drone1 waypoint0_7_1 waypoint75)
(can_traverse drone1 waypoint0_7_1 waypoint111)
(can_traverse drone1 waypoint111 waypoint0_7_1)
(can_traverse drone1 waypoint75 waypoint0_10_1)
(can_traverse drone1 waypoint0_10_1 waypoint75)
(can_traverse drone1 waypoint0_10_1 waypoint161)
(can_traverse drone1 waypoint161 waypoint0_10_1)
(can_traverse drone1 waypoint60 waypoint1_10_1)
(can_traverse drone1 waypoint1_10_1 waypoint60)
(can_traverse drone1 waypoint1_10_1 waypoint161)
(can_traverse drone1 waypoint161 waypoint1_10_1)
(can_traverse drone1 waypoint98 waypoint4_7_1)
(can_traverse drone1 waypoint4_7_1 waypoint98)
(can_traverse drone1 waypoint4_7_1 waypoint111)
(can_traverse drone1 waypoint111 waypoint4_7_1)
(can_traverse drone1 waypoint111 waypoint7_9_1)
(can_traverse drone1 waypoint7_9_1 waypoint111)
(can_traverse drone1 waypoint7_9_1 waypoint142)
(can_traverse drone1 waypoint142 waypoint7_9_1)

(visible_from objective11 waypoint75 rover0)

(visible_from objective10 waypoint93 rover0)
(visible_from objective5 waypoint96 rover0)
```

```
(visible_from objective8 waypoint118 rover0)

(visible_from objective7 waypoint111 rover0)
(visible_from objective6 waypoint139 rover0)

(visible_from objective1 waypoint161 rover0)
(visible_from objective11 waypoint75 drone1)
(visible_from objective12 waypoint60 drone1)
(visible_from objective10 waypoint93 drone1)
(visible_from objective5 waypoint96 drone1)
(visible_from objective9 waypoint98 drone1)
(visible_from objective8 waypoint118 drone1)
(visible_from objective2 waypoint102 drone1)
(visible_from objective7 waypoint111 drone1)
(visible_from objective6 waypoint139 drone1)
(visible_from objective4 waypoint142 drone1)
(visible_from objective1 waypoint161 drone1)


        )
)
```

Listing 5: Example of obtained final plan

```
2021-06-25 10:29:36 - WARNING  - __main__ - parsing duration: 0.175
2021-06-25 10:29:36 - INFO     - __main__ - Building HiPOP problem
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - Domain uses
    typing
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - Domain uses
    '=' predicate
2021-06-25 10:29:36 - INFO     - hipop.grounding.objects - Types: 8
2021-06-25 10:29:36 - INFO     - hipop.grounding.objects - Objects: 36
2021-06-25 10:29:36 - INFO     - hipop.grounding.literals - Predicates:
     20
2021-06-25 10:29:36 - INFO     - hipop.grounding.literals - Atoms: 1287
2021-06-25 10:29:36 - INFO     - hipop.grounding.literals - Fluents: 20
2021-06-25 10:29:36 - INFO     - hipop.grounding.literals - Rigid
    relations: 1
2021-06-25 10:29:36 - INFO     - hipop.grounding.literals - Rigid atoms
    : 0
2021-06-25 10:29:36 - INFO     - hipop.grounding.literals - Rigid
    literals: 1296
2021-06-25 10:29:36 - INFO     - hipop.grounding.literals - Init state
    literals: 1343
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - Goal state
    literals: 0
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - PDDL actions
    : 10
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - Possible
    action groundings: 3442
```

```
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - action
    grounding duration: 0.126s
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - Grounded
    actions: 3442
2021-06-25 10:29:36 - INFO     - hipop.grounding.hadd - h_add computed
    for 4785 elements
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - hadd
    duration: 0.021s
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - Reachable
    actions: 690
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - PDDL methods
    : 8
2021-06-25 10:29:36 - INFO     - hipop.grounding.problem - Possible
    method groundings: 11175
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - method
    grounding duration: 1.761s
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - Grounded
    methods: 11175
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - PDDL tasks:
    6
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - Possible
    task groundings: 54
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - task
    grounding duration: 0.000s
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - Grounded
    tasks: 54
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - lifted TDG
    duration: 0.000s
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - initial TDG
    duration: 0.101s
2021-06-25 10:29:38 - INFO     - hipop.grounding.problem - TDG initial:
     14671
2021-06-25 10:29:39 - INFO     - hipop.grounding.problem - TDG
    filtering duration: 0.855s
2021-06-25 10:29:39 - INFO     - hipop.grounding.problem - TDG minimal:
     10547
2021-06-25 10:29:39 - INFO     - hipop.grounding.problem - TDG HTN
    filtering duration: 0.022s
2021-06-25 10:29:39 - INFO     - hipop.grounding.problem - TDG HTN:
    5668
2021-06-25 10:29:39 - INFO     - hipop.grounding.problem - TDG
    heuristics duration: 0.357s
2021-06-25 10:29:39 - INFO     - hipop.grounding.problem - Motion
    predicate: channel_free
2021-06-25 10:29:39 - INFO     - hipop.grounding.problem - Motion
    predicate: is_dependent
2021-06-25 10:29:39 - INFO     - hipop.grounding.problem - Mutex
    computation duration: 0.013s
2021-06-25 10:29:39 - WARNING  - __main__ - grounding duration: 3.413
2021-06-25 10:29:39 - INFO     - __main__ - Solving problem
2021-06-25 10:29:39 - WARNING  - __main__ - solving duration: 0.005
```

```
==>
0 (make_available drone1)
1 (navigate rover0 waypoint0_7_1 waypoint111)
2 (visit waypoint111 rover0)
3 (navigate rover0 waypoint111 waypoint161)
4 (unvisit waypoint111 rover0)
5 (read_arTag rover0 waypoint161 objective1 camera0)
6 (communicate_arTag_data rover0 objective1)
7 (take_image rover0 waypoint161 objective1 camera0)
8 (communicate_image_data rover0 objective1)
9 (navigate drone1 waypoint0_10_1 waypoint161)
10 (visit waypoint161 drone1)
11 (navigate drone1 waypoint161 waypoint142)
12 (unvisit waypoint161 drone1)
13 (read_arTag drone1 waypoint142 objective4 camera2)
14 (communicate_arTag_data drone1 objective4)
15 (take_image drone1 waypoint142 objective4 camera2)
16 (communicate_image_data drone1 objective4)
17 (navigate rover0 waypoint0_10_1 waypoint161)
18 (visit waypoint161 rover0)
19 (navigate rover0 waypoint161 waypoint139)
20 (unvisit waypoint161 rover0)
21 (read_arTag rover0 waypoint139 objective6 camera0)
22 (communicate_arTag_data rover0 objective6)
23 (take_image rover0 waypoint139 objective6 camera0)
24 (communicate_image_data rover0 objective6)
25 (navigate rover0 waypoint0_10_1 waypoint161)
26 (visit waypoint161 rover0)
27 (navigate rover0 waypoint161 waypoint111)
28 (unvisit waypoint161 rover0)
29 (read_arTag rover0 waypoint111 objective7 camera0)
30 (communicate_arTag_data rover0 objective7)
31 (take_image rover0 waypoint111 objective7 camera0)
32 (communicate_image_data rover0 objective7)
33 (navigate drone1 waypoint0_10_1 waypoint161)
34 (visit waypoint161 drone1)
35 (navigate drone1 waypoint161 waypoint102)
36 (unvisit waypoint161 drone1)
37 (read_arTag drone1 waypoint102 objective2 camera2)
38 (communicate_arTag_data drone1 objective2)
39 (take_image drone1 waypoint102 objective2 camera2)
40 (communicate_image_data drone1 objective2)
41 (navigate rover0 waypoint0_10_1 waypoint161)
42 (visit waypoint161 rover0)
43 (navigate rover0 waypoint161 waypoint118)
44 (unvisit waypoint161 rover0)
45 (read_arTag rover0 waypoint118 objective8 camera0)
46 (communicate_arTag_data rover0 objective8)
47 (take_image rover0 waypoint118 objective8 camera0)
48 (communicate_image_data rover0 objective8)
49 (navigate drone1 waypoint0_10_1 waypoint161)
```

```
(visit waypoint161 drone1)
(navigate drone1 waypoint161 waypoint98)
(unvisit waypoint161 drone1)
(read_arTag drone1 waypoint98 objective9 camera2)
(communicate_arTag_data drone1 objective9)
(take_image drone1 waypoint98 objective9 camera2)
(communicate_image_data drone1 objective9)
(navigate rover0 waypoint0_10_1 waypoint161)
(visit waypoint161 rover0)
(navigate rover0 waypoint161 waypoint96)
(unvisit waypoint161 rover0)
(read_arTag rover0 waypoint96 objective5 camera0)
(communicate_arTag_data rover0 objective5)
(take_image rover0 waypoint96 objective5 camera0)
(communicate_image_data rover0 objective5)
(read_arTag rover0 waypoint93 objective10 camera0)
(communicate_arTag_data rover0 objective10)
(take_image rover0 waypoint93 objective10 camera0)
(communicate_image_data rover0 objective10)
(navigate drone1 waypoint0_10_1 waypoint161)
(visit waypoint161 drone1)
(navigate drone1 waypoint161 waypoint60)
(unvisit waypoint161 drone1)
(read_arTag drone1 waypoint60 objective12 camera2)
(communicate_arTag_data drone1 objective12)
(take_image drone1 waypoint60 objective12 camera2)
(communicate_image_data drone1 objective12)
(navigate rover0 waypoint111 waypoint0_7_1)
(visit waypoint0_7_1 rover0)
(navigate rover0 waypoint0_7_1 waypoint75)
(unvisit waypoint0_7_1 rover0)
(read_arTag rover0 waypoint75 objective11 camera0)
(communicate_arTag_data rover0 objective11)
(take_image rover0 waypoint75 objective11 camera0)
(communicate_image_data rover0 objective11)
<==
```

4.4 Lesson Learned

During the intense months of IGLUNA, we learned a lot about ourselves, our skills and how to work in a multidisciplinary team. To resume the main learned points:

- Define a clear initial management line;
- Do not switch too often between management tools;
- Trello can be a good management tool;
- If you want a strong business case, put two or three people to work and focus only on that task;
- Try to insert the teamwork as curricular activities to motivate the students;

- Present and showcase a project in a professional format (SD + interactions with experts)
- Push for a team identity, regardless of the difficulties of the periods;
- Enjoy the time spent in the lab;
- Interact and get to know your teammates, they may have wonderful stories to tell you;
- Overall, remember to have fun!

The period spent setting up the IGLUNA analogue mission was hard and challenging. It pushed the whole team to its limits. However, it was fun spending the days and the nights in the lab coding and fixing problems. In the end, we feel we wanted to do more and that is a good thing, because we already know how to improve our systems. The experience gathered at the Field Campaign and in Luzern is unique and could not have been gained elsewhere; we learned how to deploy our heavily regulated systems outside as a professional team, and we were able to compare our ideas and systems with the other robotic teams present in Luzern and learn from them. It was an amazing experience to get to know so many different people, with different backgrounds during the Field Campaign. Overall, it has been a very rewarding experience.

5 SD REVISIONS

5.1 Preliminary Design Review [PDR]

5.1.1 Actions

Gabriela LIGEZA, Space Innovation		
RID	Solution	Reference
You mentioned wind stronger than 5 km/h and rainy day as a medium risk. Taking into consideration that you will not be able to fly the drone in these conditions I would consider this as a high risk. It is likely that it could rain on the top of Pilatus, and then you will not be able to test. It would be good to come up with a clear mitigation action in this case. Can you protect the battery from the rain? In case of rainy day, is there anything you would be able to test?	In reality, we had a bad communication issue while writing all the PDR documentation. The drone can sustain up to 5 m/s, or 18 km/h. Usually on Mount Pilatus the strongest wind is around 18 km/h when is at its worst [3]. However, in the worst case scenario we came up with three possible different solutions: (i) test in contingency situation with just the rover, (ii) teleoperate the drone to have better control of it, (iii) manually move the drone on field to create the map (in this scenario the operator will be the ”GNC” of the drone). We used the last method for some test outdoor here at ISAE, as unfortunately we could not fly the drone for some restrictions of the school. The overall result was not too bad, and in the worst case scenario it can be a feasible solution for creating the map.	RID: SD p.16 Solution: Section 1.9

Jing QJAO, ETH Zurich		
RID	Solution	Reference
The project is documented with good structure and instructive and thoughtful contents. It would be better that the figures and tables come sequentially according to their number and below the paragraph where it is firstly mentioned.	We totally agree with the comment. However, it is hard to impose to LaTeX to fix the figures and table in specific places. We will try to fix as many ”floating content” as possible to effectively answer to this RID.	RID: all SD Solution: all SD

You mentioned that given the position of the recognized landmark and its current distance between the robot and the landmark, the algorithm will be able to retrieve the position of the robot. Only one distance measurement cannot help you calculate the position. You are also going to use the angles, right? I hope it is just a expression problem.	Yes, it was indeed a typo. Distance alone is not enough to retrieve a position. Our algorithm uses distance and angle in order to retrieve the position of the robot. The typo has been rectified in the concerned paragraph 2.15.	RID: SD p.53 Solution: Paragraph 2.15

Gaetan PETIT, Space Innovation		
RID	Solution	Reference
Who are you customers? You need to list who will pay for your service at the end and for what product.	For the RR, we will conduct a market study on our potential costumers in the fields of agriculture, goods delivery, warehouse management and building and infrastructures for Earth applications. We will present in detail the possible applications of our service in these fields.	RID: SD p.12 Solution: no ref in the main text

Maximilian EHRHARDT, ESA		
RID	Solution	Reference
I would like to see more use of the provided facilities like the control center. Even though you focus on autonomy, a human will most likely be in the loop for space missions.	We implemented more interaction between humans and robots in the mission analysis. Humans will be the operators on field, they will collect some of the data, they will analyse the overall accuracy the map and the telemetry and they will be monitoring the system in the control center. We added all of this in the ConOps of Section 2.7. Nevertheless, our end goal is to build systems as much autonomous as we can. However, we do agree that humans are an asset to any exploration mission.	RID: SD p.34 Solution: Section 2.7

Maria ORESHENKO, Space Innovation		
RID	Solution	Reference

”Revision history” is supposed to reflect the changes in the document at different reviews (i.e. Rev 01 will be PDR, Rev 02 will be CDR, etc.). You can add a separate one with detailed changes if that is useful for the team, but please put in at the end of the document.	We fixed the revision history.tex file.	RID: SD p.2 Solution: no ref in the main text
Project timeline and work packages are not easy to understand from the graphics presented, and seem to be too general/top level. Please present a detailed and clear timeline with all internal/external milestones depicted (slide 14 but with more details).	We started redefining more in detail our gantt and workpackage breakdown. The updated version can be found in Section 2.3 and it will be presented on the new slides for the CDR.	RID: Slides 12-14 Solution: Section 2.3

Diogo BRANDÃO, ESA		
RID	Solution	Reference
What do you plan to have as operation time? Using the rover and the drone 4 hours tops and then needing to manually recharge does not fall into the idea of autonomous exploration and it is a critical matter. An interesting approach would be developing self-charging algorithms and designing stations to that effect to have an autonomous system that can run indeterminably.	The idea is not to run the two systems until battery extinction, but to have a ”resources” management algorithm that in case of low battery will take the system to a ”recharge position”. In this position an operator will change their batteries. Our overall idea of ConOps for the FC is presented in section 2.7. Your suggestion to create an autonomous recharging station would be nice to analyse. However, we lack the time and the monetary resources to bring that design idea to an end by the FC. During this first year of activity of the CoRoDro team we prefer to focus on the software autonomy and its interaction with the outside world. We will definitely consider this suggestion for our next year project.	RID: / Solution: Partially in Section 2.7

It is important to better justify the choice of the rover, having trade-offs between other rovers as you have done with the camera set-up.	In this case the rover was a ”ready to be handed” product of our sponsors. We did not have a choice, nor the resources to choose another model. Therefore, we now added the rover and the drone as ”Constraints Requirements” in Table 11	RID: / Solution: Table 11

Manuel GEROLD, Space Innovation		
RID	Solution	Reference
Specify risk Dron-1: in which ways can people/infrastructure be harmed/damaged?	If the drone has a malfunction, it can impact the hardware or people nearby. Therefore, during our drone testing we require: (i) to be far from the other teams, (ii) to have a buffer of 2 m per side around our testing field, (iii) to get near the drone only when it has reached a completely stop.	RID: SD p.16 Solution: Table 6
Clarify how the drone relates to a potential lunar mission (or what it represents).	The drone’s first role is to simulate the operations of a propelled lunar bot. During the lunar mission, the small propelled bot will flight to map and explore the environment before the rover. The same bot will use the propelled flight to slow down its fall in the lava tubes while mapping the skylights. In the lava tube the bot may decide to perform short flights to better map the environment. The drone’s second role is to simulate a collaboration between robots with different characteristics. The focal point is the collaboration between diverse robotic platforms and how this difference can really help shaping the success of an exploration mission. In our configuration, the drone embodies a high mobility detection platform while the rover embodies a slow but highly-qualified analysing/resource-extracting platform.	RID: SD p.8 Solution: Section 1.5

Table 6: add values/ranges and units to your requirements as soon as possible.	We added all the values for the requirements, where a value can be specified. Most of the software requirements do not depend on a numerical value but on a coding structure. Therefore, we will code them so that they can respect the software requirements. We added the link between the software and the requirements in the CDR presentation and in Section 2.20.2.	RID: SD p.18 Solution: Overall Sections 2.1 and 2.20.2
Section 2.5: What do you do in case of bad weather (wind, rain $\rightarrow$ mitigation planning)?	During rainy days, we can test only the rover. Therefore, we will use the contingency ConOps of Section 2.6. During rainy days we cannot operate the drone by flying it. However, in the worst case scenario we can move it manually. We used this method for some test outdoor here at ISAE, as unfortunately we could not fly the drone for some restrictions of the school. The overall result was not too bad, and in the worst case scenario it can be a feasible solution for creating the map. In case of wind, we most likely will not have a problem with it - the drone can sustain 5 m/s of wind, around 18 km/h, which is the maximum wind usually experienced on Mount Pilatus [3]. The risk analysis mitigation action has been updated on Table 6.	RID: SD p.29 Solution: Table 6
Go more into detail for a detailed ConOps (or refer to section where it is tackled; start with shipping and end with a successful completion of the FC). Add time stamps to each phase, the number of people necessary, etc.	We went more into detail about the ConOPS in Section 2.7. We tried to cover all different phases of our operations entering as much as possible into detail. We will definitely update our ConOps during the Mid Term event in March. We will have two people on the field, two in the control room and three at the exhibition.	RID: SD p.30 Solution: Overall Section 2.7

Great that you already thought of safe operations for the drone. Did you already get in contact with the Federal Office of Civil Aviation (FOCA) of Switzerland? If not, we could approach them together.	We already contacted the FOCA. They indeed gave us some limitations and rules. We added their email and rules to the Annexes to assure traceability with all the team.	RID: SD p.31 Solution: Overall Section 9.5
Project interfaces: please focus on interfaces with the test bed, control room and exhibition hall. Try to be as specific as possible so that we can provide you with everything you need.	We updated our interfaces on both the presentation of the CDR and the student documentation in Section 2.9.	RID: SD p.34 Solution: Overall Section 2.9
Section 2.9.1. Please clarify: are all necessary sensors already attached and tested or are they open points with this regard?	All necessary sensors are already attached and tested. The only open point is on the LiDAR mounted on the rover. It might be used in addition to the D435 camera during the Localization and Real-Time Mapping algorithm in order to increase the reliability of the rover's position. The LiDAR (+ odometry sensors) would perform a 2D horizontal Localization and Real-Time Mapping. Otherwise, the LiDAR would have no use and would be on OFF mode during all the mission, acting as nonexistent.	RID: SD p.36 Solution: Referred to Section 2.10.2, but no direct reference in the text
More detailed mass budget (in table format); dive deep (e.g. rover structure, battery, sensors, etc.).	We updated our mass budget in Section 2.11. However, we would like to highlight that our team works on the software part and all the systems are lent by our sponsors. Therefore, we added them as constraints requirements, Table 11, as suggested during one of our bilateral meetings.	RID: SD p.35 Solution: Overall Section 2.11
More detailed power budget: which sensor uses which amount of power (you can also think of splitting the power budget into the phases of your ConOps to make sure you will return safely to your home base)	We updated our power budget in Section 2.11. However, we would like to highlight that our team works on the software part and all the systems are lent by our sponsors. Therefore, we added them as constraints requirements, Table 11, as suggested during one of our bilateral meetings.	RID: SD p.37 Solution: Overall Section 2.11

5.1.2 Advice

Gabriela LIGEZA, Space Innovation		
RID	Solution	Reference
Great plan for the exhibition! Would it be possible to maybe install or have a live camera on your drone and show to the public your live operations at Pilatus? Not too crucial but it would be nice if it does not add too much of complexity and additional costs. ;)	This is an idea we are working on. During the exhibition we will have a screen and we would like to live stream the operations on the field. When we would not have active operations, we would stream our tests at ISAE and the simulation we have done during the overall project.	RID: SD p.29 Solution: no ref in the main text

Jing QJAO , ETH Zurich		
RID	Solution	Reference
I understand that you are going to use the camera/lidar/INS data collected from the drone to generate a DEM. While in the path planned for the drone/rover is based on a 2D grid occupancy map, how do you determine the flying height of the drone? What about the situation of the drone flying in a cave (consider obstacle avoidance for the drone)? How do you get a DEM map in the case of a cave (how the camera is mounted on the drone)?	The height is defined considering parameters such as the capacity of the battery, the size of the field to map, the field of view of the camera sensors, and the maximum velocity the drone can reach while performing a Localization and Real-Time Mapping. Considering the value of those parameters, we obtained a threshold for a flying height between 2 m and 3,5 m. Moreover, tests have shown that a flying height between 2.5 m and 3.5 m gives great results. In the case of a cave, an obstacle avoidance module for the drone needs to be implemented (easier implementation than the obstacle avoidance for the rover because on the drone we already have the fully operational Localization and Real-Time Mapping that provides the drone with a precise position). We would add a LiDAR in order to add a sensor (on the top of the drone) dedicated to the scanning of the surroundings of the drone.	RID: / Solution: no ref in the main text

For the Localization and Real-Time Mapping algorithm, I understand that you are going to use VIO/IMU to get the positions of the robot. There would be accumulated errors as the rover goes further away, how do you control this? Any markers with absolute positions available? If not, how to ensure the accuracy of the generated maps?	The configuration of the field and the path planning (zigzag trajectory for the drone, always scanning a past part of its trajectory) ensures enough loop closure to be detected by the Localization and Real-Time Mapping algorithm and reduce the drift of estimated position. Also, concerning the drift of altitude of the drone, we have in our possession a proven and time-tested software that ensures a stable altitude of flight (see videos from an indoor test). Thus in a relatively non-windy environment, the drift on altitude is very low as the conditions are very stable. About absolute positions, we plan to put absolute markers on the boundaries of the field (refer to path planning section). Their goal is to be visited by the drone at the very beginning in order to know the boundaries coordinates of the field in which the drone will operate. In other words, the drone will visit those points to get their position in the frame generated by the drone. As their position is absolute (and correspond to the edges of the field), the drone will know the exact shape of the map in its frame. From there, the trajectory of the drone is implemented in order to map the whole field. We do not plan to voluntarily visit those reference points again. Indeed, we are confident about our loop closure detection in our Localization and Real-Time Mapping algorithm. However those reference points can be used as a back-up plan and can be visited to reduce the drift of the position.	RID: / Solution: no ref in the main text

If the rover/drone are in a unknown environment, real-time sensor data and path plan should be exchanged between the rover/drone and the control center, which should be well tested concerning the computation efficiency and real time communication.	We would like the rover or the drone to construct autonomously the map of the unknown environment, as much as possible and with all the problems concerning that approach. However, the communication constraints imposed by SI are giving a lot of problems on the "real time communication point". Usually, during our mapping here at ISAE, we always monitor the systems real-time, to act fast if there are problems (e.g. the constructed map is not accurate). In this case, that will be difficult. Therefore, we are thinking of monitor the main "topic" anyway, keeping into account the communication delay and at the same time have an operator on the field that in case of contingency situations can stop the system. All the tests on which we are focusing right now are thought to assure the total autonomy of the systems in this unknown environment. Anyway, we do agree that this part is really challenging and we are open to any suggestion in this sense.	RID: / Solution: no ref in the main text
Can you really achieve the claimed map accuracy of 0.10 ± 0.5 cm by drone?	That was a typo and our internal error of miscommunication. We meant 0.10 ± 0.05 m.	RID: SD p.18 Solution: Section 2.1
How far can be the CC, rover, and drone away from each other?	Indeed the system will be in a totally different location than the control center. The communication between all our systems will be conducted through a WiFi-network setup with SI. This network has a limit in the overall bandwidth of 20 Mbit/s, but it is available on our terrain so we do not need to set up a radio-telecommunication with the control center. That is a harsh constraint that we have to keep into account for all our operations with our systems.	RID: / Solution: no ref in the main text

Gaetan PETIT, Space Innovation		
RID	Solution	Reference
Check out the Swiss startup for navigation software business model https://www.embotech.com and let us know if you need an intro.	Yes, that would be really appreciated!	RID: SD p.12 Solution: no ref in the main text
This is a great market: outdoor drone market related to farming, goods delivery, building/infrastructure, inspection. Why not to talk more about this? Especially for farming drone plus rover could be ideal. Check startups like: `https://agricircle.com` `https://www.flyability.com` `https://www.involi.com` `https://droneharmony.com` `https://www.gamaya.com`	*Seen previously: For the RR, we will conduct a market study on our potential costumers in the fields of agriculture, goods delivery, warehouse management and building and infrastructures for Earth applications. We will present in detail the possibles applications of our service in these fields.* For the study we will take in account these start-ups.	RID: SD p.17 Solution: no ref in the main text
Figure 29 is maybe a little optimistic for 2020.	Indeed, we will update all those tables for the RR. Part of the project management team is focusing only on the "Business Case".	RID: SD p.82 Solution: no ref in the main text

Maximilian EHRHARDT, ESA		
RID	Solution	Reference
RF-4. Is this precision really achievable? And on the other hand is it actually needed to fulfill the tasks?	That was a typo and our internal error of miscommunication. We meant 0.10 ± 0.05 m.	RID: SD p.18 Solution: Section 2.1

Think about how the AI algorithm will detect points of interest and on what data this should be based.	We are studying the problem right now. We have researched on the following algorithms and methods: (i) ROS object detection using HSV algorithm [17], (ii) color block tracking using blob detection [4]. (iii) ARTag detection. With the guidance of our sponsor at ISAE, we have decided to work on ARTag detection that will help us estimate the position and orientation of ARTags placed on the surface of points of interest. This detection is built with the help of ROS package "ar track alvar" [15]. The output of this detection is to position coordinates of each object with respect to the DEM map frame of reference. If you have any suggestion about the usage of AR markers for Point of Interest Recognition, please let us know. We also need guidance to estimate the minimum altitude of flight for drone to detect the AR tags for the FC.	RID: SD p.44 Solution: no ref in the main text
The rover could also be localized by using the drone images and identify the rover position relative to landmarks.	Yes, that was one of the ideas in case we have problems with our localization stack. Indeed, the Localization and Real-Time Mapping team is exploring different possibilities on this matter.	RID: / Solution: no ref in the main text

Test whether the Raspberry Pi is feasible enough for your tasks. It is cheap and capable, but also for later hardware iterations a more powerful system could be a better option.	The hardware was handed to us and all the systems we are considering run a Rasberry Pi. This is our first year of activity as a team and we lack financing that can permit us to tailor our hardware. At the same time, we can look at the bright side: we have an hardware that is restricted in terms of computational power and, therefore, simulate more realistically the limited power of the hardware in a real lunar mission. We hope next year, if we are successful during this IGLUNA campaign, to raise some money to start buying and building our own hardware. That will definitely help us running more complex algorithms and keep up with the challenge.	RID: / Solution: no ref in the main text

Maria ORESHENKO, Space Innovation		
RID	Solution	Reference
It is highly unlikely that there will be snow on Pilatus in July. ;)	Noted!	RID: SD p.16 Solution: no ref in the main text

Diogo BRANDÃO, ESA		
RID	Solution	Reference

The mission seems to focus a lot on the interaction between one rover and one drone, with the rover storing the data acquired by the drone and transmitting it to the CC. Did you took into account future expandability? Meaning adding more rovers and drones into a future mission. How would it work? Data from each drone to each rover (pack of 2) or keep the data decentralized?	We do take into account future expandability - we would like to explore this idea, with a more complex mission scenario including sampling activities for the following IGLUNA editions. If we are lucky and more people will be interested in the project, we would like to explore the swarming approach where all the systems are at the same level and create a bigger coordinated exploration entity. However, in order to accomplish these tasks we need to create more knowledge in our group about the different aspects of swarms of robotic systems.	RID: / Solution: no ref in the main text

Does the redundancy of Depth Camera + LiDAR make sense? Will it not be better to include a tracking camera as well in the rover for a more constant and precise location? That way you could not only detect obstacles but also merge the locations obtained from the drone and rover to the global DEM.	The Tracking Camera T265 on our drone consists of a Stereo Camera coupled with an IMU. It performs a Localization and Real-Time Mapping algorithm, directly on a processing unit embedded in the T265. The rover is mounted with several sensors including an IMU + the Depth Camera D435i. We are implementing with those two sensors a much more sophisticated Localization and Real-Time Mapping algorithm than the Localization and Real-Time Mapping algorithm performed on the T265. That is why the position of the rover is set to be even more precise than the position of the drone (even if the position of the drone is already very precise compared to a simple estimation from IMU or Visual Odometry). Moreover, the LiDAR of the rover can be added (and surely will be) as input to the Localization and Real-Time Mapping algorithm. It will perform what is called a 2D Localization and Real-Time Mapping. It will increase even more the precision of the position. A remark: the Depth Camera on the rover is able to see the ground because it is mounted on the rover facing slightly the ground area in front of the rover. However, the LiDAR is mounted on top of the rover's frame and its orientation is a 360° horizontal plan. That is why the LiDAR would be useful for the Localization and Real-Time Mapping algorithm only if the Operating field presents big features that are higher than the Rover. Otherwise, there is no chance the LiDAR would detect anything. In that case, the LiDAR would have absolutely no utility and will be turned off during all the mission time. To answer the idea of merging the locations obtained from the drone and rover to the Global map, it is something that is planned. The obsta-	RID: SD p.42 Reference: Localization and Real-Time Mapping algorithm for the rover - Section 2.15.2

Figure 20 and 21. Drone and rover mission phase flow: what do you intent on using as criteria for comparing the paths and classifying them as too different or close enough to the previous path? Both in rover and drone path planning.	For path planning our criteria are: (i) how long and how many resources the computation of the path took, (ii) how distant from the obstacles was it, (iii) how short was the path, (iv) how many resources (mainly battery level) did it consume, (v) did we reach all the waypoints in the most efficient manner (similar to the travel salesman problem). Based on this criteria, we will classify the performances of path planning. We have as well the problem of localization, but that is one of the parameters of interest for the Localization and Real-Time Mapping algorithm. Do you have any suggestion on the matter?	RID: SD p.48 Solution: no ref in the main text
Are there plans for using ROS2 instead of ROS1? If you have plans to develop as a company, the option of choosing the more future proof interface would make more sense. Even when most of the packages you mentioned are already done or bridged to ROS2.	During this year we will use ROS1 for two main reasons: (i) all our robotic systems are already configured with ROS1 Indigo, (ii) most of the team is in the process to learn ROS, therefore the usage of ROS1 permits to speed up the process thanks to the ROS community support and the tutorials. However, we indeed plan a migration toward ROS2 the next year: ROS 2 is opening the application of ROS to certified robotics - it is the effort of having a real time OS with a deterministic scheduler. That feature can be quite important for the start-up development!	RID: / Solution: no ref in the main text

Manuel GEROLD, Space Innovation		
RID	Solution	Reference
Section 1.9. What about rain and wind? At what conditions would the drone be grounded?	We may have the drone grounded for the ”mission phase” of the drone, Section 2.6, if it rains or the wind is stronger than 18 km/h. However, for the map phase, in the worst case scenario, we will manually move the drone on the terrain manually.	RID: SD p.10 Solution: no ref in the main text

Table 5: on what data did you base your evaluation of F-1 and F-2? Maybe revise as bad weather for a week is not unlikely to happen at Mt. Pilatus	We updated our risk mitigation actions on Table 6.	RID: SD p.16 Solution: Section 1.9
Table 2: try to add measurable goals (i.e. values/ranges and units) to follow a red line throughout the project and for being able to evaluate on whether your mission was a success.	We added numerical values to the requirements where we could. For the others we linked them with the software we are developing for the FC on the slides of the CDR and in Section 2.20.2.	RID: SD p.14 Solution: all SD
Double-check if there are only two trade-offs to be done for your system.	Up to now, yes, we are not conducting other trade-offs. We still can change the cameras that we are using or install more sensors without incurring in additional cost, thanks to our sponsors. Therefore, we will see what sensor bundle is the most suitable for our algorithms and a successful FC.	RID: SD p.22 Solution: no ref in the main text
Think of splitting the data budget into the phases of your mission.	We added the budgets per phase in Section 2.11.	RID: SD p.37 Solution: Section 2.11

5.2 Critical Design Review [CDR]

5.2.1 Actions

Gabriela LIGEZA, Space Innovation		
RID	Solution	Reference
Please specify how much power budget you will need per each day of operations. Same with data budget.	Two tables, Tables 30 and 31, have been set to show the power needed for the following ConOps: drone mission, mapping of the drone and rover mission. Those tables allow us to know how much power and data we will need per day (depending on the number of testing we want to do in that day). We also added in Section 2.9, the number of batteries needed by test for the drone and the rover.	RID: Slides 19-21 Solution: Section 2.9, and Tables 30 and 31

Table 11 - constraints requirements. You mentioned you want to suggest some minor modification to the rover (like OBC) - this has to be already decided at the CDR stage. Same with the drone!	Indeed, everything is already decided and operative on both systems. However, the constraints requirements were defined in "a general view", keeping into account even the modification at the beginning of the project. We changed a bit the formulation of the requirement to comply with your comment.	RID: SD p.25 Solution: Table 11

Jing QIAO, PhD, ETH Zurich		
RID	Solution	Reference
Is there a rule that the drone/rover should follow when planning its path?	Both systems try to find the shortest path to move from one way point to the other while keeping distance from the obstacles. The security distance is achieved with a fake inflation of the footprint of the rover/drone of 5% and of the obstacles of 5%. The shortest path is found using two standard algorithms called A* and D* for global path planning. More details are given in Section 2.16.2. We would like to compare the performance of the two during the FC. At the same time, for local obstacle avoidance we use the "dynamic window approach". It is a local path planner that tries to avoid obstacles unforeseen in the map and then tries to go back to the first optimal path evaluated by the A* or D* as fast as possible. More details are given in Section 2.16.2.	RID: / Solution: Section 2.16.2

At which point the CC thinks the job is completed and determines to call back the rover?	During the overall mission, the system will contact the control center when it reaches the last way point of its mission, notifying it of the end of the tasks. Meanwhile, the control centre will use a state machine to visualize and track the different positions reached by the system during its mission. Therefore, we can evaluate that we effectively achieved the end of mission and we got the right notification from the systems. If the systems reach their end position without notifying the control centre, we will record one of the task planning objectives' failure. However, we would be able to call the systems back because we monitored the overall mission.	RID: / Solution: no ref in the main text
Is there a distance threshold between the rover and drone? Any possibility that the drone will get lost? Emergency plan?	The distance threshold of the WiFi enabling the communication between the rover and the drone is of 30 m. Since the flying perimeter is less than 10 m there is no possibility for the rover to get lost. Nevertheless, if the drone manages to get off the perimeter, the pilot in the field will stop it and will take care of it.	RID: / Solution: no ref in the main text
Considering the rather low temperature on the mountain with snow, the battery consumption may be larger than normal case. Have you taken this into account?	There is only a significant impact on battery consumption below 0 ℃, and since we will be in Mount Pilatus in July, the average temperature for this month is of 8 ℃, with the minimum normally around 4 ℃ and maximum of 12 ℃. These temperatures should not significantly affect the consumption.	RID: / Solution: no ref in the main text

Manuel GEROLD, Space Innovation		
RID	Solution	Reference

In your ConOps table, please add the number of people needed to do a task or supervise a phase. You can also add a column for the control room, the exhibition and the test bed.	Added in the Tables 22, 23, 24, 25, 26, and 27 (mapping, beginning and end of the day tasks, control room ConOps, drone mission, rover mission, and set-up of the field, respectively).	RID: / Solution: Tables 22, 23, 24, 25, 26, and 27.
What do you do if the drone crashes on one of the first days?	If the drone crashes and there are minor damages to the hardware we will fix it with the proper tools brought to the field (repair kit). If there are major damages after the crash and the drone is unusable, we would turn to our emergency plan. This consists in getting the drone to take several shots at the terrain before the field test and saving the maps for the case that the drone is severely damaged during the field test, so we can use only the rover for the mission.	RID: / Solution: no ref in the main text

Diogo BRANDÃO, ESA		
RID	Solution	Reference
You say that the drone should be capable of detecting 100% of obstacles that have an height superior to 6 cm, but then you say that the map generated shall register at least 80% of the obstacles that have an height equal or bigger than 6 cm.	The detection by the drone has been lowered to a more realistic 80%. The generated map, which is updated by the rover, has a detection that has been increased to 85% of obstacles superior or equal to 6 cm height.	RID: SD p.24 Solution: Table 8
LiDAR power consumption of 0.5 W here, 10 W on page 50.	The value has been update in Table 33.	RID: SD p.26 Solution: Table 33

The fact that you have to manually use a USB drive to transfer the data really hurts the autonomy part of the project. You refer that using a Jetson Nano in the rover would not help since the transfer time is the same, but was it thought to include a Jetson Nano on the drone? It would add only like 150 g to the weight if I am not mistaken.	We have been working on the algorithms and we are now proud to say that the drone can execute all the algorithms that generate the final occupancy grid for the rover. Such an occupancy grid weighs less than 200 Ko and can easily be sent through the communication network to the rover and to the control center. No USB key is needed anymore.	RID: / Solution: no ref in the main text
In relation to the conversion module problem, in which you say the low density point cloud from the LiDAR provides bad results on the conversion module. Did you try to find ways to increase the density of the point cloud? Either by increasing the RPM of the LiDAR or by including an algorithm that could populate the point cloud?	The RPM of the LiDAR cannot be increased. However, concerning the advice of working on an algorithm that populates the point cloud, that is a great suggestion we will work on. Thank you.	RID: / Solution: no ref in the main text

Alex TORRES, CNES		
RID	Solution	Reference

The equipment list of drone and rover could be improved in several ways: i) Coherence: the same information should be given for all parts (if you give the capacity of a battery, you should give the capacity of all batteries). ii) Completeness: crucial functional information could be added (baseline of stereo cameras, pose errors on tracking camera, sampling rates rather than data throughput). iii) Correctness: specifications should be double checked for errors (do you really use a 220 mAh on the drone, or a 2200 mAh one? If the computer on the drone is a Raspberry Pi (not indicated), should the consumed power be really different? Data production is exactly the same for the tracking camera and the drone computer? Voltage and capacity are written twice for the upper section of the rover). I would suggest to describe the equipment in a table rather than a list, in order to check the presence of corresponding characteristics and easily compare them.	The equipment list has been put in two tables, Table 33 and 34, one for the drone and one for the rover. Each part of the systems' hardware is named with its precise reference and, when useful, some other characteristics (for example, for all the batteries we mentioned their capacitance). We also gave extra information on the cameras (maximum frame rate, resolution). The power consumption of the systems have been double checked and updated in three other tables, Tables 37, 38, and 39.	RID: SD p.49 Solution: Tables 33 and 34, and Tables 37, 38, and 39
Be careful with spelling: "weel acurator", several French words (alimentation, captors, variator (for ESC)).	Misspelling corrected.	RID: SD p.49 Solution: all SD
Format of data production figures: a human readable format would be preferable.	Tables 37, 38, and 39 updated in order to be readable.	RID: SD p.50 Solution: Tables 37, 38, and 39
Power consumption is very different between the Raspberry Pi on the drone and the rover, and they are different than the figures given in the equipment list. Which is right?	There is no Raspberry Pi on the drone, the computer is an ODROID-XU4. The power consumption of the Raspberry Pi of the rover and the ODROID-XU4 of the drone have been double checked and updated in the following tables, Tables 37, 38, and 39.	RID: SD p.53-54 Solution: Tables 37, 38, and 39

Data budget for the drone: D435 and LiDAR are listed on data production but not on equipment list or data production by mode.	Data budget for the drone D435 and LiDAR is only listed on the data budget, Tables 40 and 41. The equipment list is now put in tables and provides only the precise references of each system's hardware part and some extra information when needed (besides from power consumption and data budget, listed in the Tables 37, 38, 39, and 40, 41).	RID: SD p.55 Solution: Tables 40, 41, 42, and 43
The reduction of telemetry data is a real concern of space missions. Instead of trying to circumvent this issue with a USB download, you could try what it ought to be done in practice: if your on-board computer cannot process and send a lighter occupancy grid, maybe you should change to a coarser DEM that could be processed on board or send by TM? Exchanging high resolution point clouds is not very realistic.	We have been working on the algorithms and we are now proud to say that the drone can execute all the algorithms that generate the final occupancy grid for the rover. Such an occupancy grid weighs less than 200 Ko and can easily be sent through the communication network to the rover and to the control center. No USB key is needed anymore.	RID: SD p.56 Solution: no ref in the main text
In the diagram, the position of the rover and drone should be an input of the position estimation block, not the AR tag detection.	Diagram has been corrected.	RID: Slide 33 Solution: AR tag block diagram 30
In the diagrams, sensor readings should only go into the localization blocks, not the planning modules. And the grid map, is it really needed in the localization module? Detailing data exchanges helps to understand the workings of the system and to translate it into code.	Diagrams have been adjusted.	RID: Slide 37 Solution: Rover mission block diagram 38 and drone mission block diagram 36

Drone mapping with wind is probably not the most interesting or realistic contingency case for a lunar mission.	That is totally right. In fact, it is a contingency scenario of the IGLUNA analogue mission. It is possible that during the testing we would have some wind blowing on our drone and we should anyway be able to control it and generate the map to keep on with the mission. So in this case we go down to Earth to be able to prove a concept of collaborative robotics with different systems. We changed the table 20 and the related Section 2.7.2	RID: SD p.37 Solution: Table 20 and the related Section 2.7.2
I think there is some confusion with the concept of SLAM. Sequential operations of localization and mapping function do not qualify as SLAM unless they do some kind of optimization or filtering of a joint state.	The T265 performs a SLAM algorithm to retrieve the position and corrects it for example in the case the robot loses track of itself for a while. The fact is that the map generated thanks to such an embedded SLAM algorithm is not precise enough for enabling the wanted navigation system for the rover. So we decided to generate another map in real-time thanks to the localization of the T265. It is true that the mapping process does not influence back the localization process. We are working on an Update Module that updates and corrects the map when the localization is adjusted thanks to the detection of a loop closure on that generated map (and we would have a mutual influence of localization and mapping processes on each other).	RID: SD p.65 Solution: no ref in the main text

5.3 Readiness Review [RR]

Alex TORRES, CNES		
RID	Solution	Reference
However, the specific characteristics chosen for some equipment are not the most relevant (max fps instead of FOV or resolution, frequency if wifi instead of range,...). In particular the frequency given for the Raspberry Pi is particularly misleading: one expects to see its clock frequency (1.4 GHz), not the wifi frequency	Some of the specific characteristics have been updated with most relevant ones. The clock frequency of the Raspberry Pi was a typo, it is actually 400 MGz.For the Leo Rover Network, the Leo Rover website `https://www.leorover.tech/the-rover` only gives : • WiFi 2.4 GHz access point with external antenna • WiFi 2.4 GHZ + 5 GHz on internal RPi antennas for connectivity We don't have any other information therefore we chose to let the frequency range as specific characteristic for the WiFi.	RID: SD p.50 Solution: p50
2.11.5 Some apparent inconsistencies when comparing the data budget to that in the previous version. The data throughput of the T265 has been divided by a factor of 100. Why? And how do you explain then that the total data budget (Table 42) has increased from 20 to 30 Mbps? Have you computed the data rates yourselves or do you use the values given by the data-sheets? If the D435 of the rover produces 30Mbps (table 41), how come the maximum total throughtput is only 9.1Mbps (table 43)?	We did some mistakes on past data budgets. The last one should be the most reliable, and all the data was monitored with the real setup. During the testing, we throttled both the visual odometry and the video frame rate of the D435 to stay in the imposed limitation. The overall "publishing" frequency of the T265 toward the control center has been decrease at 6 fps. The D435 is activated only when we have to take a picture, the rest of the video stream is not useful for the mission. .	RID: SD p.54 Solution: No reference in the text

2.14 SLAM. In my opinion, it is not a good idea to put all mapping and localization functions under a chapter SLAM. Mapping and localization are in genreal two distinct functions, and one can talk about SLAM only when the pose of the vehicle and the position of the landmarks are estimated together in a common optimization. Scanning of the environnement by a depth camera is not always step 1b of SLAM, and DEM generation and merging is typically not a function of SLAM. Most common visual SLAM algorithms give *sparse* maps only as a by-product, but its main output is the vehicle pose (that's why the T265 does not give a DEM). And the dense map needs to be generated by a perception function. (The Lidar-based SLAM of the rover in 2.14.2 is a bit different, it gives a sort of dense map, which is in fact the measured point clouds reprojected from the estimated attitudes) ?	Renamed Localization and Real-Time Mapping.	RID: SD p.58 Solution: SD p.60
2.14.1.1 If the LIDAR has finally been de-scoped, this paragraph would probaly be clearer without mentioning it	Paragraph added ”Drone different evaluated Configurations and Final chosen Configuration” that presents a clearer information.	RID: SD p.61 Solution: SD p.61

2.14.1.3 If the conversion module does the same kind of point cloud downsampling than the depth filtering module, wouldn't it be possible to downsample to the required resolution directly on depth filtering, to reduce the required computation in data merging and DEM matching modules?	The post process in the Depth Filtering Module is keeping the 3cm resolution of the 3D dem map because this resolution is needed to detect tiny (about 6cm height) obstacles later in the Conversion Module. With an input of 3cm resolution map in the conversion module, it is able to detect such tiny obstacles. Then, depending on the resolution of 2D grid the operator wants to achieve, the conversion module is discretizing the 2D map to a lower or equal resolution of the input 3D map resolution that was given to it. The discretizing process (which reduces the resolution) looks at the proportion of obstacles/free-space in a given cell to determine if that cell of a lower resolution should be considered as free space or as obstacle. That is why having a resolution of 3cm is needed as an input of the Conversion Module and that if the 3D map is discretized to a lower resolution before, the software loses its ability to detect and take into consideration tiny obstacles.	RID: SD p.63 Solution: SD p.63

2.14.1.3 Could you give some more details about the conversion process? How do you transform the 3D cloud in 2.5D DEM? And if you only take into account the height gradient to decide the traversability, how do you take into account the slope?	The 3D to 2.5D conversion is made by keeping the point of highest altitude on every column (x,y). The cell (x,y) has value the altitude of that highest point. There are of course some tiny processes to determine if a point is relevant or shall be discarded. The package ”Grid-Map-PCL” was found to be performing efficiently that step so it was used to perform that task. Concerning the question about height gradient vs slope : The only difference between the height gradient and the slope is that the gradient is a vector and the slope is a scalar. The value of the height gradient for a cell is the value of the slope in that cell. There is no difference between the slope and the height gradient, it is the same information.	RID: SD p.63 Solution: No ref in the main text
2.14.2 The rover SLAM module is presented as a possibility. Have you implemented this algorithm, and in particular, the obstacle detection module (comparison between cost map and DEM)?	The rover uses AMCL in order to take in account the obstacles that are not registered on the given 2D occupancy map for planning its trajectory and adjusting it. Given the resources we had, the updating module to update the 2D map however was not implemented.	RID: SD p.70 Solution: SD p.70
2.14.3 Your collaborative Localization and Real-Time Mapping seems a good idea, computing a global plan from big obstacles seen from the drone and letting the rover Localization and Real-Time Mapping deal with the smaller ones. But have you thought what would happen if you have big areas covered with small but non traversable obstacles?	That case enters the already considered cases in which the rover cannot compute an initial plan to go to its target or in which the rover will encounter unregistered obstacles. In that case, the rover aims to reach its target, by locally adjusting its trajectory due to the tiny obstacles.	RID: SD p.69 Solution: SD p.72

2.19.3.4 the reason why the bench is a very bad target to test the stereobench is not that ”it lets the light pass through”, but rather that it presents a periodic pattern, which means that the matching between cameras can get ambiguous results. Fortunately, periodic pattern like these are hardly found in natural environments.	We took into account the reviewer comment	RID: SD p.118 Solution: No reference in the text

Tatiana Benavides, SI		
RID	Solution	Reference
Are you planning to send something else (eg mock-up models, goodies or a poster) besides videos to the museum?	We should be able to come with some of our camera sensors to the exhibition to show our algorithms. We would have some goodies as well.	RID: pg. 32 Solution: SD annex p.195
How would the points of interest be recognized in an unknown environment where there are no AR tags (eg Moon lavatube exploration)?	he Computer vision Module that detects AR tags can easily be replaced by another computer vision algorithm using Convolutional neural networks that can be used for detecting any type of objects depending on the mission goal. We chose AR tags to stay generic and to prove our system is working.	RID: pg 71 Solution: No ref in the main text
Define in detail content of video loops for museum exhibition	The video will present the team and the different work of the members of CoRoDro (Path-Planning, Task-Planning, SLAM, Telecommunication, Project Management, Communication and Outreach). It will show the videos of the tests we performed along with a description of what the viewer should understand watching those tests videos.	RID: pg 32 Solution: No ref in the main text

Can you give more details on test SM-5 ”A method shall be designed in order to test the proportion of virtual (nonexistent) obstacles on the generated”? By virtual you mean inserted on the map as virtual reality? Do you need to first map with drone and then insert elements?	I wish we were working on Virtual Reality. But ”virtual” here was meant to be understood as ”non-real”, so the test SM-5 is about detecting all the false obstacles registered on the map that are actually representing a wrong information. The test would then provide a proportion of ”false obstacles” on ”all obstacles” and give an insight on how accurate the map is.	RID: pg 105 Solution: No ref in the main text
Think of everything you might need and spares (chargers, electrical extension cables, etc). Personal items could include snacks, sunscreen, sunglasses, hat, water bottles, etc. Assign roles to the team members doing the tests, have a ”runner/reserve” just in case extra things are needed.	We kept into account the reviewer comment. During the test only the required operators were present.	RID: pg 128 Solution:
Have you conducted the market study mentioned in the answer of the RID from G. Petit?	Many team members left before the RR, therefore we didn’t have the time to complete the full market study.	RID: pg 135 Solution: No ref in the main text

Manuel Gerold, SI		
RID	Solution	Reference
2.11.5: Great overview of the data budget. Could you please state a conclusion for each of the systems (e.g. showing the maximum transmission possible for the system and comparing it with the budget needed).	We show for the drone and the rover the maximum transmission possible and we compare it with the budget needed	RID: pg. 58 2.14 Solution: pg 58 2.14 and Tables 42, 43
2.4 and 2.8: Clear planning for the VFC. Would you also have a storyboard of your project video for the FC?	The video plan is a follows : -Presentation of the project -Presentation of the drone and rover -Mission Analysis talk -SLAM talk -Path Planning Talk -Telecom Talk -Outreach and Communication Talk -Project Management Talk -Testing Videos and comments -Results of the tests -Conclusion and Future prospects	RID: pg. 32-47 Solution: No ref in the main text

Jing Qiao, ETH		
RID	Solution	Reference
2.6 Project setup: How will the system work if there is no physical border, or AR-tags, to define the working field or calibration?	If there are no physical border defined by target or point of interest, then we would implement a path planning algorithm that would enable the drone to autonomously map the area around it given a limit condition such as a radius limit from its starting point or a surface area limit like 50m square.	RID: pg. 35 Solution: No ref in the main text

2.6 Project setup: Targets will be represented by manually placed grey dots? Will you identify your targets by image recognition? Can it be based on position?	Targets are represented by cuboidal box which has AR-Tags on 5 faces of the box. It was a mistake in the text about "grey dots" early in the document. Each AR-tag is recognized by an edge detector algorithm. ar_track_alvar package is used which internally is an ARToolKit. It is a simplified marker detection algorithm and any detected potential marker is then matched against existing tag templates. If the marker is successfully mapped, it is accepted as valid and next the value inside the black frame is decoded. When targets are firstly detected by the drone, their position is unknown so no the recognition of the target cannot be based on position, the position is actually the output of the AR-tag detection algorithm. When the rover or the drone wants to visit previously detected targets during their mission phase, they know an approximate position of the target. Thus, that approximate position is given to the robotic systems to go there. But, once the robotic system has reached the target position, the identification of the target is based again on the ar_tag detection based on image recognition. It is for controlling if the robotic system is looking at the target as expected, and to confirm the position of the target (the position of the ar_tag is again an output of the detection, not an input).	RID: pg. 35 Solution: No ref in the main text
2.7 Concept of Operations: The drone has autonomy for around ten minutes, why? due to battery limit?	The autonomy is due to battery limit, the hardware was not build by the team. Therefore, we had to adapt on what was readily available from our sponsors.	RID: pg. 37 Solution: SD p.49

2.7.2 Operator 1 checks the quality of the map and stores the telemetry data. Can the quality be checked autonomously without operator?	Autonomously assessing the quality and validity of a map is an open topic of research today. It is feasible but it requires a lot of work and it was not our subject of research this year. So currently one still needs a human to check the quality of the generated map.	RID: pg. 40 Solution: No ref in the main text
Table 32: needs to be better displayed to show which description for which instrument	We tried to make the table clearer.	RID: Solution: Table 32
All the figures/flowcharts are nicely drawn, it would be better if they can be in vector image	We would have liked them to be in vector image. However, overleaf can't compile since the document is too big.	RID: Solution: No ref in the main text

Levin Gerdes, ESA		
RID	Solution	Reference
A lot of important tests are not done yet (path planning rover and drone, communication tests, Localization and Real-Time Mapping accuracy, rover localization,..), when are these foreseen? (Update and clarifications followed in discussion; most of these finished between document submission and collocation.	The test have been all conducted and the VFC was successful in terms of obtained results.	RID: slide 13 Solution: No ref in the main text
Just a typo, tests are foreseen for first two weeks of June, not July	Yes, we confirm that it was a typo.	RID: slide 15 Solution: No ref in the main text
Unclear how the drone can map the area. The boundaries and other markers are well understood, but you mention path planning before mapping. (Clarification in colloc.: exact drone pose within well defined test area is known, no just the test area dimensions)	The shape of the area to be mapped is known. So from that and from knowing the starting point of the drone in that map we can set up a pre-designed trajectory for the drone (path plan) to be executed. The drone follows that path and maps the area during that path.	RID: Solution: No ref in the main text

Temperature requirement insufficient for testing in Toulouse in summer. Your hardware certainly survives more (and will have to operate in more demanding conditions) than -5 to +20 degrees C. Please fix RC-3. Additionally, your requirement description reads as -5 to 20 degrees, but you write the values -7 to +22. Make sure no choice is based on these low temperatures only.	The documentation was initially written according to a Field Campaign in Switzerland. The requirement RC-3 is adapted to the conditions on top of Mount Pilatus where the temperatures there match the RC-3 requirement. The requirement has been updated for the VFC.	RID: p 26 Solution: Table 11
Please update the slope requirement, RC-5. You currently only require the ability to traverse slopes of +8 to +12 degrees. Should probably read as -10 to +10. But it's good that you have this requirement in the first place!	Fixed	RID: pg 26 Solution: Table 11
Please explain COM-03. Why do you require a maximum camera range? Do you refer to focus? Why is this in the communication requirement?	We sincerely don't know who in the team added that requirement, as it was not in the first version of the SD. We erase it, because (i) we never considered it, (ii) it doesn't make sense.	RID: pg 26 Solution: No ref in the main text
The description of RS-2 doesn't fit an Emergency Stop (you describe the problem in RS-3). Please make sure your team members in the field have actual e-stops. This can be included in RS-3. I'm not sure what RS-2 means, just being able to shut down equipment from the control center in general?	The drone and the rover have both an E-stop. The RS-2 gives the control center the possibility of switching off one of the systems if needed to stop the testing remotely.	RID: pg 27 Solution: No ref in the main text
How does the drone's e-stop work? Does it perform an emergency landing right there or does it switch off its motors?	Fore the rover the E-stop just stop the system. For the drone, there are two possibilities during emergency: (i) going from mode offboard to pilot mode, where the pilot on the field take control of the drone; (ii) stop the system where it is killing off the motors.	RID: pg 25 Solution: No ref in the main text

2.7.2. How does the rover know the approximate POIs' positions? And what is that level of confidence? Even with a drone map (in the nominal case) you only have a certain level of confidence (no perfect map, no correlation in the beginning).	The AR-tag detection algorithm embedded in the rover helps give the position and orientation of POI with respect to the rover frame of reference. The POI's position can then be computed in the map frame thanks to AMCL that localize the rover in the map of the drone, or with a personal algorithm called "TF-Linker" presented in the "Merging Rover's and Drone's Frames and Localizing the Rover with the Drone" subsection of the "Localization and Real-Time Mapping" section.	RID: pg 38 Solution: "TF-Linker" SD p.72
Setup of the control center at CNES seems very fast at 25 minutes. Have you tried this a few times already in a different place?	Indeed we have done it several times and we can confirm that the set-up of the control center is not very long. It includes putting chairs and table, a wifi antenna (that is already configured) and the computers, as well as securing an electricity source.	RID: pg 42 Solution: No ref in the main text
You note possible re-testing down with 1-2 minutes. Please double check that you take account of the testing and not only of notifying that the test needs to be repeated.	Indeed the "re-testing" is just a notification to the operators on the field. After the notification the process starts from zero as explained in the ConOPS.	RID: pg 42 Solution: No ref in the main text
You mention that 1 rover battery is needed per day. Please mention the number of batteries you bring and keep charged here. (Clarified in discussion: more batteries are available; the exact number of batteries and chargers that will be brought is noted somewhere else in the document.).	Fixed	RID: pg 49 Solution: SD p.49

How and when do you improve on your grid map resolution to go from your current 14 cm obstacle detection down to 6 cm?	The information about the 14cm obstacle detection was dated from the PDR and was not up to date. The improvement was made with the completion of the Conversion Module. Before, at the PDR, the Conversion Module was only able to crop everything below a certain altitude so it worked fine with the flat terrain of our lab. That simplistic version aimed to crop out the ground, the remaining voxels in the 3D map would constitute the 2D grid map by just projecting them on a plane. With the completion of the Conversion Module, it's a more sophisticated process explained in the Localization and Real-Time Mapping section. It is able to determine intelligently if an area is reachable for the rover or not and so it even works on terrain that are not flat at all.	RID: pg 61 Solution: SD p.65
Please mention rover's locomotion capabilities in requirements or assumptions. This is design driving information and should not be hidden in a submodule description so late into the document.	The rover generic capabilities to perform on uneven terrain has always been written in the requirements. Later empirical tests revealed new capabilities of the rover and that is why they are not presented as "requirements". According to your advice, a line in constraints requirements "RC-10" has been added for "Obstacle capabilities of the rover".	RID: pg 63 Solution: SD p.26
Figure 27: Why are you using a custom Localization and Real-Time Mapping solution on the rover and don't just reuse what you are using on the drone? (From the discussion: the drone's solution can be used on both the drone and the rover. Not finally decided which one will be used. ACTION: Please make sure both your document and presentation reflect your final decision)	In the case of a relatively flat terrain, the rover uses AMCL to relocalize in the map created by the drone. In case the drone encountered a problem and the rover was not provided with a map, then the rover executes the same algorithm to create a map as the drone did.	RID: pg 67 Solution: SD p.72

”To improve the accuracy of the rover position, it is possible that we use the LiDAR mounted on the rover for a 2D Localization and Real-Time Mapping.” This decision should normally have been taken for the CDR. What is your decision or when do you make the call?	The LiDAR on the rover was used to perform a 2D Localization and Real-Time Mapping task during the contingency case where the drone failed to provide a map. So in the end we used this but not in order to improve the accuracy of the rover position, but to ensure the success of the mission in contingency cases.	RID: pg 67 Solution: SD p.72
Please feature your assumption on the terrain prominently. I would expect a table of assumptions just before the requirements. This way, you can refer from requirements to assumptions and everything becomes more traceable. It also becomes much clearer which aspects are impacted by a change of assumptions. E.g., what needs to change when the terrain is different, when the rover is more/-less capable, etc. Just as well intended advice for future projects.	We kept into account the comment of the reviewer for the next editions of IGLUNA.	RID: Solution: No ref in the main text

Diogo Brandão, ESA		
RID	Solution	Reference
Repeated objectives 1 and 3	Fixed.	RID: pg. 15 Solution:

Clarify about the paths generated both for the rover and the drone (if they are different paths), and where are these paths generated. Along the document you mentioned that the path for the drone is generated on the rover, and then you say that it is generated on the drone.	They are several paths that are mentioned throughout the document. There is the operational path (Task-Planner) that contains the different orders to be given to the drone and to the rover ("rover visits objective number 3"; "drone visits objective number 5") and which is generated on the rover. There are also path plans (Path-Planning) that give the path to take between the actual position of the robotic system and its goal. Those paths are generated on the robotic system that executes them (the rover computes its own path plan to move across the field, and the drone computes its path plan to move across the field).	RID: pg. 37 Solution: No ref in the main text
After mapping the environment in point 2, will the drone land and wait for until it receives the path from the rover? Or is it all done during flight? This question also relates with the fact that the drone only flies for 10 minutes. This was explained on the presentation but needs to be clear on Concept of Operations	For safety reasons, the drone and the rover never operate at the same time on the field. So when the drone map is done the drone is removed from the field and the rover is placed in. And when the drone needs to operate the rover is removed from the field. So because the plan is generated on the rover, the drone needs to wait outside the field for the rover to compute the drone's plan. Then the drone is placed again on the field and executes its plan in the second flight.	RID: pg. 37 Solution: No ref in the main text
How do you know the most likely positions if you don't have a map at all in the rover? (Rover mission without known map)	In this case the control center will come to rescue. They will see what the rover is seeing thanks to the rover camera and they can move the system toward a ArTag. In that case the rover, has to move, map and reach as many POIs as possible autonomously.	RID: pg. 38 Solution: Subsection 2.7.2

Why are you getting the usb drive for later evaluation if you already had the data sent to the control center? (Data Download table 22: Mapping)	Now that we post process most of the data on-board we only send to the wifi the most important data. To make a full later evaluation we need register some rosbags that are usually pretty heavy collections of data. Therefore we use a usb key just to not spend 1h waiting to receive a rosbag to replay the test.	RID: pg. 41 Solution: No ref in the main text
The rover battery ensures up to 5h of running time. 5 hours might not be enough for the full day testing. A second battery might be needed. This was answered on the presentation and you said you had two batteries, please include it in the SD	The rover does not operate all the time, we already performed several days of testing with one battery for the rover lower part. The number of batteries has been added page.49	RID: pg. 48 Solution: SD p.49
Did you try to use only the D435i to do both localization and mapping? Although the localization may not be as accurate, the IMU from the d435i could be enough and then you could save some weight on the drone as well. You explained this on the presentation with the argument that the Localization and Real-Time Mapping wasnt accurate enought just with the d435i, but it would be good to see some of these results that you have tested on the SD	A research was conducted on SLAM algorithms using only the D435i but due to the time we had and the fact that this research was not giving better results quickly enough we decided to stick to the first version T265 as position + D435i as only depth scan.	RID: pg. 59 Solution: No ref in the text.
Are you using PCL library to reduce the resolution of the point clouds? I assume you are using the Voxel grid from pcl. Mentioning the libraries you used for each software module would be interesting for helping other students that are reading your SD to find answers to their own projects	It has always been mentioned in the Depth-Filtering Module page 64 where those algorithms are used as well in the references	RID: pg. 60 Solution: SD p.69

Isn't it possible to define a higher sensitivity on the generation of the 2D grid map? If the accuracy is present on the 3D map, the 2D grid map should be able to detect up to that accuracy if you wish no? (Data Merging module and Matching module)	That is already the case, the grid map generation considers the information related to the accuracy of the 3D map in entry. The tiniest the resolution of the 3D map will be, the more accurate the 2D map will be.	RID: pg. 61 Solution: No ref in the main text
Here (page 63, last 4 paragraphs) you say that you are converting the 2D occupancy grid into a 2D costmap. This cosmap includes obstacle inflation to avoid collision of the rover into the obstacles. But in the next paragraph you say that the ouput is the 2D occupancy grid and not the 2D costmap. In the images on page 115 and 116 its seems you are using only the occupancy grid. Please clarify that and make it consistent along the document text and diagrams/images	In the end, the Localization and Real-Time Mapping section is only computing a 2D Occupancy grid as final output of the mapping phase. The 2D Cost-Map is dynamically generated by the Path-Planning team on the Rover via the software Move-base.	RID: pg. 63/115/116 Solution: SD P.63
Wouldn't be simpler just to add one smaller AR tag on the top of the rover? This way you would have minimum error on the rover position and would be simpler to program. If you wanted also orientation you could add 2 AR tags in line on the rover. It looks like you thought about this because in the image 50 for indoors you are using vicon balls to track the rover.	First of all, in the indoor test we are equipped with an Opti-track system composed of cameras around the room that track the said vicon balls, it does not need the drone to track the vicon-balls. Then, it is true we thought about puting a tag on top of the rover, we mentioned it in the Localization and Real-Time Mapping section, but we then decided not to perform this due to safety reasons. If the drone encounters a problem and crashes on the rover, it damages all the robotic systems we have and results in a hard failure of the mission. That is why the rover and the drone never operate at the same time on the field and that that localization solution was discarded.	RID: pg. 70 Solution: No ref in the main text

6 ABOUT THE TEAM

6.1 Point of contact

Who?	Email	Mobile Phone Number
Jasmine Rimani	jasmine.rimani@polito.it jasmine.rimani@isae-supaero.fr	+39 389 488 4444
CoRoDro Team	corodro.igluna@gmail.com	

6.2 Team Structure Overview

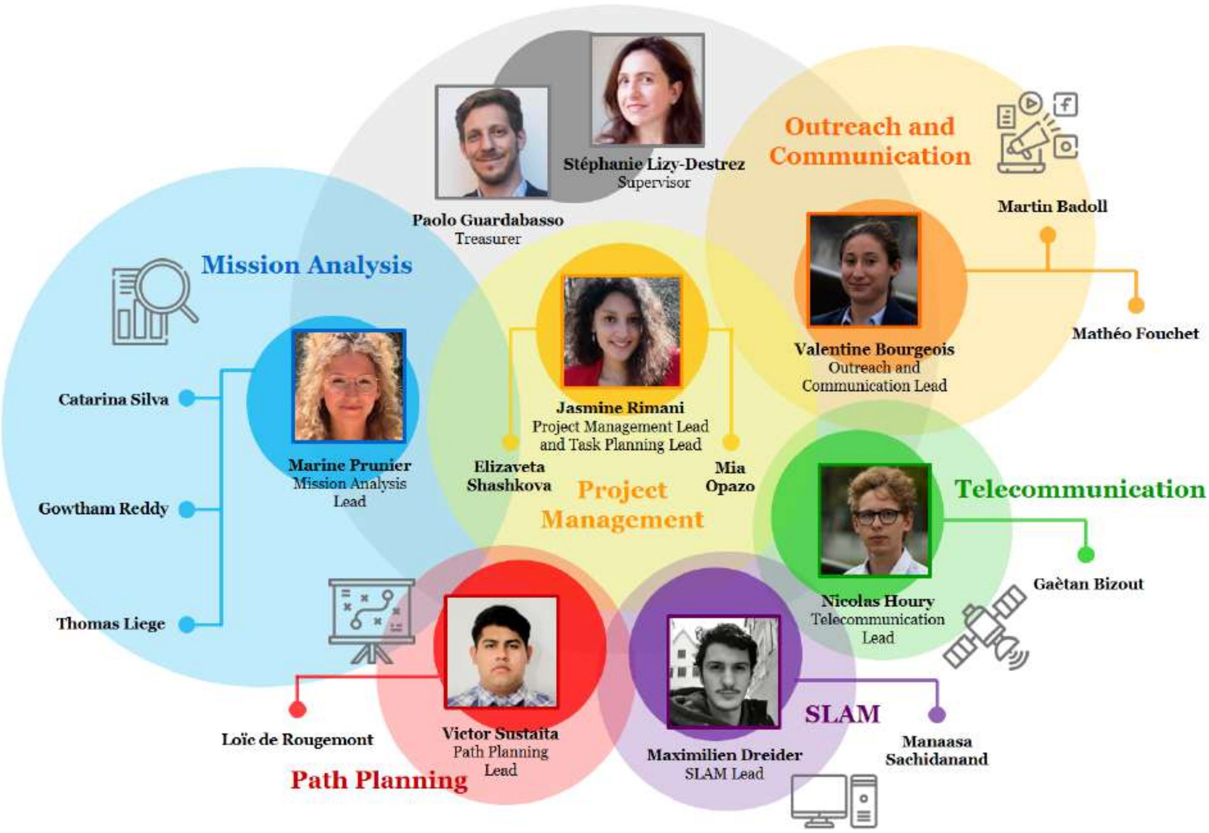


Figure 81: Team Structure

6.3 Team Members

Name: Jasmine Rimani
Responsibility: Project Manager, Operations Lead and Path Planning Team Member (drone and rover)
Education Reached: Master's Degree
Current Level: PhD Student
Thesis: Autonomous Navigation and On-Board Autonomy for Space Exploration Systems via Detection Isolation and Recovery
Workload: 15h
Other Interests: Space Systems Engineering, Failure Management, Autonomous Operations, Programming, Robotics

Name: Marine Prunier
Responsibility: Mission Analysis Lead
Education Reached: BSc in Physics
Current Level: MSc in Aerospace Engineering
Workload: 5h
Email: Marine.PRUNIER@student.isae-supaero.fr
Other Interests: Space Systems Engineering, Cinema, Fencing

Name: Victor Hugo Sustaita Rodríguez
Responsibility: Path Planning Drone Lead
Education Reached: BSc in Mechatronics Engineering
Current Level: MSc in Aerospace Engineering
Workload: 5/6h
Email: Victor.SUSTAITA-RODRIGUEZ@student.isae-supaero.fr
Other Interests: Languages, Automation, Music

Name: Maximilien Dreier
Responsibility: SLAM Lead
Education Reached: BSc in Mathematics
Current Level: MSc in Applied Mathematics
Workload: Full-Time Internship
Email: maximilien.dreier@isae-supaero.fr
Other Interests: The Infinitely Large Scale, Neuroscience, Robotics

Name: Nicolas Houry
Responsibility: Telecommunications Lead
Education Reached: Baccalauréat
Current Level: Engineering Degree in Aerospace
Workload: 5h
Email: nicolas.HOURY@student.isae-supaero.fr
Other Interests: Programming, Astronomy, Music

Name: Valentine Bourgeois
Responsibility: Outreach and Communication Lead
Education Reached: Baccalauréat
Current Level: Engineering Degree in Aerospace
Workload: 5/6h
Email: valentine.BOURGEOIS@student.isae-supaero.fr
Other Interests: Tennis, Space, Programming

Name: Paolo Guardabasso
Responsibility: Treasurer
Education Reached: Master's Degree
Current Level: PhD Student
Thesis: Study of Cislunar Environment for Orbital Debris Mitigation
Workload: 1h
Other Interests: Space Systems Engineering and Mission Design, Analogue Mission Experiment Design, Astronaut Science, Space Exploration History

Name: Elizaveta Shashkova
Responsibility: Project Management
Education Reached: BSc in Physics
Current Level: MSc in Aerospace Engineering
Workload: 5h
Email: elizaveta.shashkova@student.isae-supaero.fr
Other Interests: Hiking, Travelling

Name: Mia Opazo
Responsibility: Project Management
Education Reached: BSc in Physics and BSc in Mathematics
Current Level: MSc in Aerospace Engineering
Workload: 5h
Email: miriam.opazo-mendez@student.isae-supaero.fr
Other Interests: Space Exploration, Languages, Travelling

Name: Catarina Antunes da Silva
Responsibility: Mission Analysis
Education Reached: BSc in Electrical and Computer Engineering
Current Level: MSc in Aerospace Engineering
Workload: 4h
Email: catarina.antunes-da-silva@student.isae-supaero.fr
Other Interests: Astronomy, Solar Energy, Yoga, Travelling

Name: Gowtham Reddy
Responsibility: Mission Analysis
Education Reached: BSc in Aerospace Engineering
Current Level: MSc in Aerospace Engineering
Workload: 3h
Email: Gowthamreddyg3g@gmail.com
Other Interests: Football, Cycling, Control, Systems

Name: Thomas Liege
Responsibility: Mission Analysis
Education Reached: Baccalauréat
Current Level: Engineering Degree in Aerospace
Workload: All the time needed
Email: ThomasLIEGE@student.isae-supaero.fr
Other Interests: Workout, Philosophy, Basketball, Astrophysics

Name: Loïc de Rougemon
Responsibility: Path Planning
Education Reached: Baccalauréat
Current Level: Engineering Degree in Aerospace
Workload: 5h
Email: loic.de-rougemont@student.isae-supaero.fr
Other Interests: Comics, Programming, Space Engineering

Name: Maanasa Sachidanand
Responsibility: SLAM
Education Reached: Btech in Electrical and Electronics Engineering
Current Level: MSc in Aerospace Engineering
Workload: 5h
Email: maanasa.sachi1997@gmail.com
Other Interests: Teaching Volunteering, Singing, Programming, Embedded Systems

Name: Gaétan Bizot
Responsibility: Telecommunication
Education Reached: Baccalauréat
Current Level: Engineering Degree in Aerospace
Workload: 2/3h
Email: gaetan.bizot@student.isae-supaero.fr
Other Interests: Judo, Playing Music Instruments, Physics

Name: Martin Badoil
Responsibility: Outreach and Sponsoring
Education Reached: Baccalauréat
Current Level: Engineering Degree in Aerospace
Workload: 4h
Email: martin.badoil@student.isae-supaero.fr
Other Interests: Soccer, Piano, Cycling

Name: Matheo Fouchet
Responsibility: Outreach and Sponsoring
Education Reached: Baccalauréat
Current Level: Engineering Degree in Aerospace
Workload: 3h
Email: matheofouchet@gmail.com
Other Interests: Mars, Space In General

Name: Mael Mansouri *(Former member)*
Responsibility: Path Planning Rover Lead
Education Reached: Baccalauréat
Current Level: Engineering Degree in Aerospace
Workload: 5/6h
Email: mael.mansouri@student.isae-supaero.fr
Other Interests: Sailing, Space Systems Engineering, Space Mechanic Engineering

Name: Nathan Goy *(Former member)*
Responsibility: Mission Analysis
Education Reached: Baccalauréat Scientifique
Current Level: Engineering Degree in Aerospace
Workload: 5h
Email: nathan.goy@student.isae-supaero.fr
Other Interests: Space Exploration, Space Systems Engineering, Astronomy, Astrophysics

Name: Saumik Islam *(Former member)*
Responsibility: Path Planning
Education Reached: Diplôme universitaire de technologie
Current Level: Engineering Degree in Aerospace
Workload: 5h
Email: saumik.islam@student.isae-supaero.fr
Other Interests: Space Systems Engineering

7 LIST OF ABBREVIATIONS

Abbreviation	Description
A.I.	Artificial Intelligence
CDR	Critical Design Review
DCAS	Department of Aerospace Vehicles Design and Control
DEOS	Department of Electronics, Optronics and Signal Processing
DEM	Digital Elevation Model
DISC	Department Complex Systems Engineering
DoD	Definition of Done
DRM	Design Reference Mission
EKF	Extended Kalman Filter
ESA	European Space Agency
ESA BIC	ESA Business Incubation Center
E-Stop	Emergency Stop
FC	Field Campaign
FOCA	Federal Office of Civil Aviation
FOM	Figures of Merit
HDDL	Hierarchical Domain Definition Language
IP	Intellectual Property
ISAE-SUPAERO	Institut Supérieur De L'aéronautique Et De L'espace
IT	Information Technology
KO	Kick Off meeting
LOS	Line Of Sight
OBC	On-Board Computer
ONERA	Office National d'Études et de Recherches Aérospatiales
OSIP	Open Space Innovation Platform
PDDL	Problem Domain Definition Language
PDR	Preliminary Design Review
PoliTO	Politecnico di Torino
R&D	Research and Development
PoI	Point of Interest
Rev.	Revision
RR	Readiness Review
ROS	Robot Operating System
SD	Student Documentation
SI	Space Innovation
SMACH	State Machine
SME	Small and Medium Enterprises
SLAM	Simultaneous Localization and Mapping
SWOT	Strength, Weakness, Opportunities, Treats
TRL	Technology Readiness Level
VFC	Virtual Field Campaign

8 REFERENCES

References

[1] Autonomy in space systems. https://www.h2020-ergo.eu/project/background-on-autonomy-software-frameworks/autonomy-in-space-systems/. Accessed: 2020-11-10.

[2] Business planning. https://www.bplans.com/. Accessed: 2020-10-21.

[3] Climate mount pilatus. https://www.meteoblue.com/en/weather/historyclimate/climatemodelled/mount-pilatus_switzerland_2659201. Accessed: 2021-01-02.

[4] Color block tracking. https://by-the-w3i.github.io/2019/10/06/ColorBlockTracking/. Accessed: 2021-01-02.

[5] The drone market: Insights from customers and providers. https://www.comptia.org/content/research/drone-industry-trends-analysis. Accessed: 2020-10-21.

[6] Dynamic window algorithm motion planning. . Accessed: 2021-01-02.

[7] Ecss-e-st-10-02c. http://everyspec.com/ESA/download.php?spec=ECSS-E-10-02C.047796.pdf. Accessed: 2020-10-21.

[8] Finding the shortest path, with a little help from dijkstra. https://medium.com/basecs/finding-the-shortest-path-with-a-little-help-from-dijkstra-613149fbdc8e. Accessed: 2021-01-02.

[9] Foca. https://www.bazl.admin.ch/bazl/en/home/good-to-know/drones-and-aircraft-models/permits-drones.html. Accessed: 2021-01-02.

[10] Leo rover. https://www.leorover.tech/. Accessed: 2020-10-21.

[11] Occupancy grid format. http://docs.ros.org/en/diamondback/api/nav_msgs/html/msg/OccupancyGrid.html. Accessed: 2021-01-30.

[12] Pld space - wikipedia page. https://en.wikipedia.org/wiki/PLD_Space. Accessed: 2020-10-21.

[13] Radius outlier removal filter. https://pointclouds.org/documentation/tutorials/remove_outliers.html#remove-outliers. Accessed: 2021-01-30.

[14] Robotic motion planning: A* and d* search. http://www.cs.cmu.edu/~motionplanning. Accessed: 2021-01-02.

[15] Ros ar-track-alvar package. http://wiki.ros.org/ar_track_alvar. Accessed: 2021-01-02.

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

[17] Ros object recognition. https://github.com/joffman/ros_object_recognition. Accessed: 2021-01-02.

[18] Space robotics market by solution, services (satellite servicing, on-orbit assembly & manufacturing, de-orbiting services, re-supply, surface mobility, & launch support), application, end-user, and region - global forecast to 2023. https://www.marketsandmarkets.com/Market-Reports/space-robotic-solution-market-61497036.html. Accessed: 2020-10-21.

[19] Statistical outlier removal filter. https://pointclouds.org/documentation/tutorials/statistical_outlier.html#statistical-outlier-removal. Accessed: 2021-01-30.

[20] Voxel definition. https://en.wikipedia.org/wiki/Voxel. Accessed: 2021-01-30.

[21] Voxel grid filter. https://pcl.readthedocs.io/projects/tutorials/en/latest/voxel_grid.html. Accessed: 2021-01-30.

[22] Amazon Prime. Amazon prime air.

[23] Patrick Bechon, Charles Lesire, and Magali Barbier. Hybrid planning and distributed iterative repair for multi-robot missions with communication losses. *Auton. Robots*, 44(3-4):505–531, 2020.

[24] S. Chien, R. Knight, R. Stechert, A.and Sherwood, and G. Rabideau. Integrated planning and execution for autonomous spacecraft. *IEEE Aerospace Conference*, 1999.

[25] G. De Angelis and et al. Lunar lava tube radiation safety analysis, 2002.

[26] ESA and European Commission. The european space technology master plan. *ESTMP*, 2018.

[27] ESA Advanced Concepts Team. Ai in space workshop, 2013.

[28] C. Frost. Challenges and opportunities for autonomous systems in space. 2011.

[29] Chad R. Frost. Challenges and opportunities for autonomous systems in space. *Computer Science*, 2011.

[30] R. Greeley. Lava tubes and channels in the lunar marius hills. *The Moon*, 1971.

[31] Horizon-2020. European robotic goal-oriented autonomous controller (ergo).

[32] T Horvath and P.O. Hayne. Thermal environments and illumination in lunar pits and lava tubes, 2018.

[33] ISECG. The global exploration roadmap. *The International Space Exploration Coordination Group*, 2019.

[34] T Kaku and et al. Detection of intact lava tubes at marius hills on the moon by selene (kaguya) lunar radar sounder. *Geophysical Research Letter*, 2017.

[35] Samuel J. Lawrence and et al. Lro observations of morphology and surface roughness of volcanic cones and lobate lava flows in the marius hills. *Journal of Geophysical Research: Planets*, 118, 2013.

[36] Charles Lesire and Alexandre Albore. pyhipop – hierarchical partial-order planner. *International Planning Competition*, 2020.

[37] P. Martin, S. Kwong, N. T. Smith, Y. Yamashiki, O. Payton, F. Russell-Pavier, J. Fardoulis, D. Richards, and T. Scott. 3d unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility. *Int. J. Appl. Earth Obs. Geoinformation*, 52:12–19, 2016.

[38] NASA. Nasa robot plunges into volcano to explore fissure.

[39] Panagiotis I. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios. A compilation of uav applications for precision agriculture. *Comput. Networks*, 172:107148, 2020.

[40] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

[41] Dominik Schreiber. Getting started with automated planning, 2019.

[42] Peter Zane Schulte. A state machine architecture for aerospace vehicle fault protection. *PhD Dissertation, Georgia Institute of Technology*, 2018.

[43] Allen G. Taylor and A. Gibbs. Automated search for lunar lava tubes in the clementine dataset, 1998.

[44] W. Whittaker. Technologies enabling exploration of skylights, lava tubes and caves, 2012.

[45] W.r Whittaker. Exploration of planetary skylights and tunnels, 2014.

[46] C. Yinka-Banjo and Olasupo Ajayi. Sky-farmers: Applications of unmanned aerial vehicles (uav) in agriculture. 2019.

9 Annexes

9.1 Lessons Learned

This section will be completed after the FC.

9.2 Lunar Mission Analysis

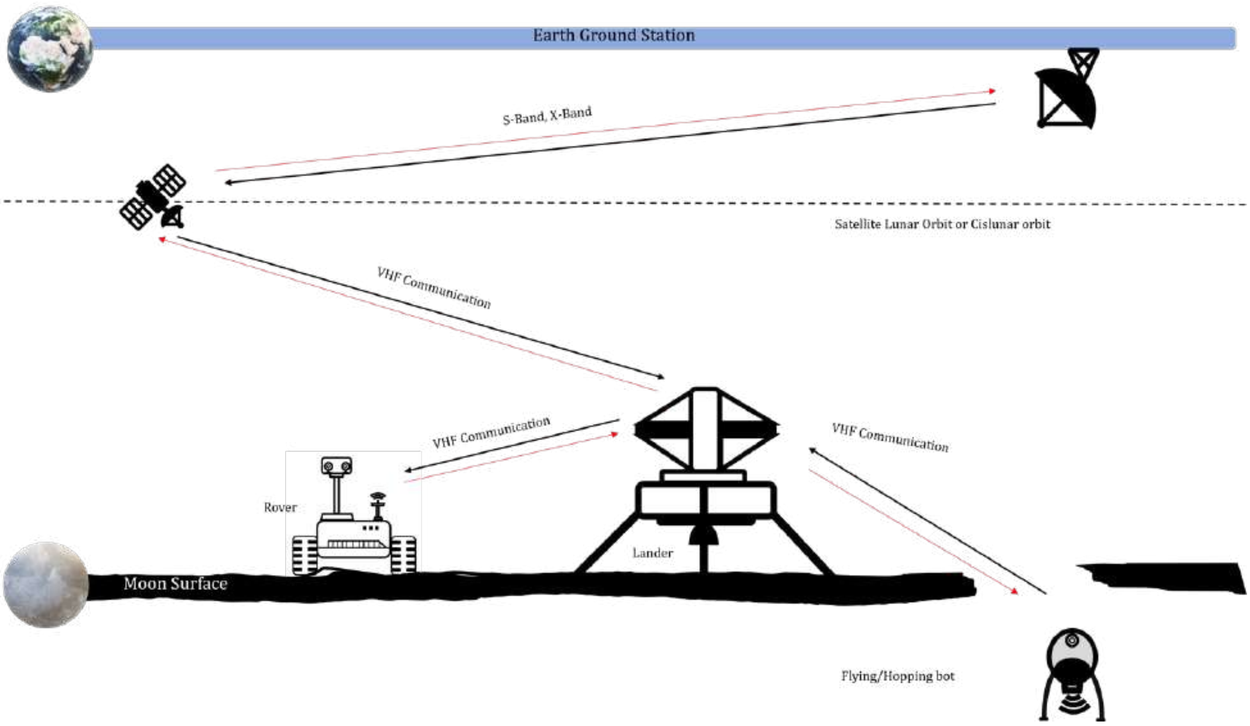


Figure 82: Design reference mission for the exploration of the lunar lava tubes

This decade has seen a rise in the interest on the Moon human colonization of both space agencies and industries [33, 26]. The Earth's natural satellite can become the gate toward the exploration of Mars and beyond. It can be exploited as a testing facility to advance disruptive technologies fundamental for both robotic and human space exploration. The foreseen exciting aim is the creation of a permanent human colony on the Moon. Among the sites of interest as future human settlement, the lunar lava tubes have been selected as possible targets [44, 45]. Various studies have shown that that the temperature and radiation environment inside this volcanic architecture may be more human-friendly than any other place on the equatorial region of the Moon [25, 32]. The focus of the scientific community lays mainly in three lava tubes, Mare Ingenii, Mare Tranquillitatis, and Marius Hills. They have been detected and studied thanks to satellite missions such as Kaguya/Selene, LRO and Clementine [35, 34, 30, 43]. However, from the satellite images is difficult to understand the effective morphology of those underground tunnels, their dimensions and their geological characterization. Therefore, precursor robotic missions are envisioned for mapping the lava tubes, assess their safety for humans and study their geological characteristics [44, 45]. Due to the challenging communication set-up during the lava tube exploration [44, 45], the foreseen exploration systems should be able to operate autonomously between the different communication windows. They should be able to map their surroundings, understand and choose their targets, and decide the best path to follow. At the same time, they should be able to monitor the "health" of their critical subsystems and act accordingly to their resources while planning the trajectory. They should be able to respect a *E3* level of autonomy of the ECSS space segment operability standard. Therefore, one of the most critical aspects of the missions related to the lava tubes is deeply linked to operations. More in general, the operational domain is associated with many challenges with both robotic and human space exploration. As space exploration missions grow in complexity, there is a need to balance the mission return, the autonomy level and the workload of the control centre operators. Different space agencies, companies, and universities are engaged in the definition of a broad spectrum of technological maturation studies toward autonomous operations and navigation [31, 29, 24, 27]. In this context, the SaCLab team at ISAE-SUPAERO and PoliTO are collaborative working on new algorithms to enhance the autonomous operations and autonomous of space robotic systems. The reference mission is the exploration of the lava tubes. While the systems under study are a rover and a drones' swarm, dedicate to mapping and characterizing the lava tubes. With the word ”drone”, it is identified a hopping/flying system equipped with a thruster that can perform small flights when required (for example during the descent on the lava tube from the skylight).

9.3 Project Management

9.3.1 Business Canvas

9.3.1.1 Key partners

- Research centers, like ONERA, are always pushing forward the innovation in the aerospace domain. A partnership with them can help us obtain funds from the European Commission. For the H2020 funding framework consortium of research centers, universities and companies have more chance to win the funds.
- Technical universities like, ISAE-SUPAERO or Politecnico di Torino, can help us during the first years of the start-up as well as providing interns to the start-up.
- In the first years of activities, our primary revenue stream is based on the Earth market - therefore we look forward to closely work with companies interested in ariculture 4.0, storage inspection and scientific instrumentation for autonomous exploring geological sites.

9.3.1.2 Key Activities

- Our primary focus is the ”autonomous navigation and operations” software development - our software shall be able to be interfaced with different platforms.
- The team background is in space systems and mission analysis - therefore, we will import this knowledge in our company accompanying our customers from the conception of the need to the deployment of the software in the platform.
- Through all our activity, we focus on acquiring knowledge and push further the AI planning and autonomous navigation algorithms. It will be a continuous cycle of concept definition, testing, deployment on real application, and continuous support. The innovation is the key value of our business model, hence R&D is our working paradigm.

9.3.1.3 Key Resources We aim for a lean business model. Our principal needs can be summarized as:

- Performing workstation running ROS, CATIA or SolidWorks, Vitech Geneysis 7.0, Valispace, and Microsoft Office Suite, with access to storage cloud.

- Hardware to test our algorithms on real platforms and studying their overall behaviour.
- ”Human resources” to study, analyze and come up to a solution for our customers. The core team will start as a cluster of free-lancers for the first years to cut down the expenses in term of social security. We will recruit interns from our partners’ universities to help on the R&D development section.
- Initial funding capital.

We estimate our startup funding requirements following the guidelines in [2]. The initial required funding is 32 k€ subdivided as it is shown in Table 90. We envisioned a mortgage to fund our start-up of 40 k€, to be repaid in ten years with a fixed interest of 0.38%. In our preliminary cost structure, we considered: (i) legal consultancy, (ii) stationary, (iii) professional liability insurance, (iv) business licenses, (v) website development, (vi) office/lab rent and furniture, (vii) testing hardware budget, (viii) software purchase.

Our ten years development plan can be outlined as:

- We will start with InnovSpace, the ISAE-SUPAERO’s start-up incubator.
- After bulding up the customer service and strengthen the customer segment as well as the partnerships, we will deposit our IP patent for our software.
- We will move toward the ESA BIC installation in Toulouse, to obtain (i) initial funding support, (ii) affordable office space, (iii) marketing solutions, (iv) networking activities, (v) mentoring program, (vi) legal advice. The initial funding supported by the ESA BIC installation tops-up to 50 k€ and relates exclusively to R&D, IP protection and market consultancy.
- Our last step will be applying to the Airbus BizLab to accelerate our start-up toward a full company.

The initial start-up funding should be covered by a mortgage, own capital or other third-party sources. We target different European grants, Table 89.

Entry	Type	Amount	Unit
ESA BIC	Funding (to be used for R&D, IP protection, third-party consultancy and market research)	Up to 50	k€
ESA BIC	Additional loan (to cover office support, marketing solutions, networking activities, mentoring program, technical support and legal advice)	Up to 50	M€
Horizon2020	Funding (data from ERGO [31])	3	M€
SME	European funding for small and medium companies	Around 2.5 (data from PLD Space [12]	M€
OSIP Opportunities	ESA funding opportunities	Variable fundings	[-]

Table 89: Envisioned European funding

9.3.1.4 Cost Structure

A recap of our preliminary cost structure is outlined in Table 90.

Entry	Type	Envisioned Amount per Year	Unit
Office Rent	Fixed	9	k€
Website and Social Media	Fixed	3	k€
Personnel Salary (with taxes, for 10 people working full time)	Fixed	500	M€
Personnel Mobility	Variable	Up to 5	k€
Legal Support	Variable	Up to 5	k€
IT Support	Variable	Up to 5	k€
Mortgage for Office Equipment (to extinguish during the first 3 years)	Fixed	7.8	k€
Mortgage for Start-Up (to extinguish during the first 5 years)	Fixed	7	k€
Marketing Actions	Variable	Up to 3	k€
Testing Equipment	Variable	Up to 3	k€

Table 90: Envisioned cost structure start-up

9.3.1.5 Revenue Streams

In general, our market analysis shows that there is great interest in Earth application envisioning collaborative robotics in terms of autonomous drones operations and navigation. The overall forecast global market for the field robotics sector is estimated to be around 43.1 B$ by 2024 [5]. The global market in space robotics is esteemed to be around 2.9 B$ [18]. The current market forecast shows a plausible rise up to 4.36 B$ by 2023 [18]. Our end target is the Moon and deep space robotics market. However, that sector is not yet mature to provide enough money streams to be sustainable. Therefore, our R&D will focus on the development of the Earth technologies, keeping into account the needed developments space-wise. Starting from this assumption, we outlined our revenues for the first five years as shown in Figure 83 .

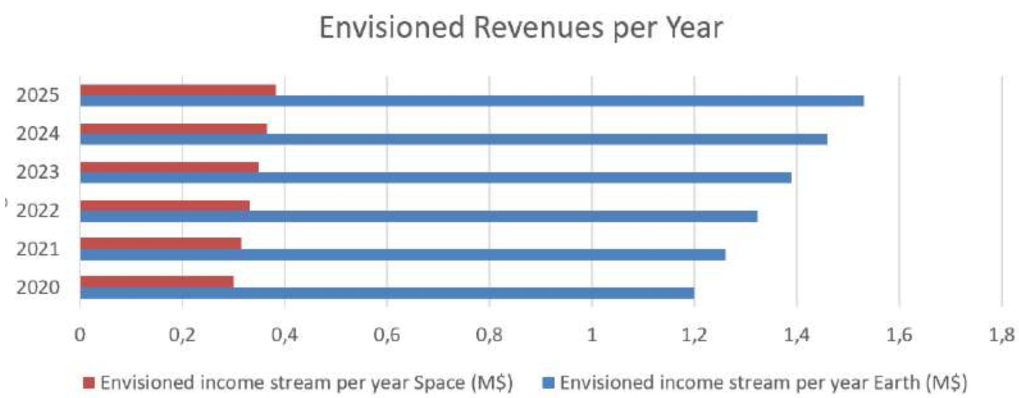


Figure 83: The plot shows the envisioned revenue stream for the first six years of activity. We assumed a cost inflation rate of 0.6% per year in the Eurozone and a steady growth of 5% for both our main stream channels

9.4 Work Packages Breakdown

In this section we will present our main work-package in detail.

Work package number	01	Start Date	KO	Deadline	PDR
Work package title	Decide Algorithm				
Work package manager	Jasmine Rimani				
WP1 Objectives					
Task 1-1 Description of task: Decide SLAM Algorithm Task manager: Maximilien Dreier Deadline: PDR					
Task 1-2 Description of task: Decide Task Planning Algorithm Task manager: Jasmine Rimani Deadline: PDR					
Task 1-3 Description of task: Decide Path Planning Algorithm Task manager: Mael Mansouri Deadline: PDR					
Task 1-4 Description of task: Learn ROS Task manager: Jasmine Rimani Deadline: PDR					
Task 1-5 Description of task: Test Telecommunication Set-up Task manager: Nicolas Houry Deadline: PDR					

Work package number	02	Start Date	PDR	Deadline	CDR
Work package title	Code Algorithm				
Work package manager	Jasmine Rimani				
WP1 Objectives					
Task 1-1 Description of task: Code SLAM Algorithm Task manager: Maximilien Dreier Deadline: CDR					
Task 1-2 Description of task: Code Task Planning Algorithm Task manager: Jasmine Rimani Deadline: CDR					
Task 1-3 Description of task: Code Path Planning Algorithm Task manager: Mael Mansouri Deadline: CDR					
Task 1-4 Description of task: Estimate Data Budget Task manager: Nicolas Houry Deadline: CDR					

Work package number	03	Start Date	CDR	Deadline	RR
Work package title	Wrap-up Algorithm				
Work package manager	Jasmine Rimani				
WP1 Objectives					
Task 1-1 Description of task: Wrap-up SLAM Algorithm Task manager: Maximilien Dreier Deadline: RR					
Task 1-2 Description of task: Wrap-up Task Planning Algorithm Task manager: Jasmine Rimani Deadline: RR					
Task 1-3 Description of task: Wrap-up Path Planning Algorithm Task manager: Mael Mansouri Deadline: RR					
Task 1-4 Description of task: Harmonize the Three Codes Task manager: Jasmine Rimani Deadline: RR					
Task 1-5 Description of task: Test the Codes Task manager: Jasmine Rimani Deadline: RR					
Task 1-6 Description of task: Test Final Telecommunication Architecture Task manager: Nicolas Houry Deadline: RR					

Work package number	04	Start Date	KO	Deadline	PDR
Work package title	Mission Analysis				
Work package manager	Nathan Goy				
WP1 Objectives					
Task 1-1 Description of task: Define Requirements Task manager: Nathan Goy Deadline: PDR					
Task 1-2 Description of task: Define Hardware Task manager: Nathan Goy Deadline: PDR					
Task 1-3 Description of task: Define General ConOps Task manager: Nathan Goy Deadline: PDR					
Task 1-4 Description of task: Define Risk and Mitigation Actions Task manager: Nathan Goy Deadline: PDR					

Work package number	05	Start Date	PDR	Deadline	CDR
Work package title	Mission Analysis				
Work package manager	Nathan Goy, Marine Prunier				
WP1 Objectives					
Task 1-1 Description of task: Check Verification Requirements Task manager: Catarina Silva Deadline: CDR					
Task 1-2 Description of task: Define Final Budgets Task manager: Gowtham Reddy Deadline: CDR					
Task 1-3 Description of task: Refine ConOps Task manager: Marine Prunier Deadline: CDR					
Task 1-4 Description of task: Answer RIDs for Mission Analysis Task manager: Nathan Goy Deadline: CDR					

Work package number	06	Start Date	CDR	Deadline	RR
Work package title	Mission Analysis				
Work package manager	Marine Prunier				
WP1 Objectives					
Task 1-1 Description of task: Check Verification Requirements Task manager: Catarina Silva Deadline: RR					
Task 1-2 Description of task: Final ConOps Task manager: Marine Prunier Deadline: RR					
Task 1-3 Description of task: Answer RIDs for Mission Analysis Task manager: Marine Prunier Deadline: RR					

Work package number	07	Start Date	KO	Deadline	PDR
Work package title	Project Management				
Work package manager	Jasmine Rimani				
WP1 Objectives					
Task 1-1 Description of task: Draft the Business Model Task manager: Jasmine Rimani Deadline: PDR					
Task 1-2 Description of task: Set Up Social Media Task manager: Valentine Bourgeois Deadline: PDR					
Task 1-3 Description of task: Write First Version SD Task manager: Jasmine Rimani Deadline: PDR					
Task 1-4 Description of task: Recruit Members Task manager: Jasmine Rimani Deadline: PDR					

Work package number	08	Start Date	PDR	Deadline	CDR
Work package title	Project Management				
Work package manager	Jasmine Rimani				
WP1 Objectives					
Task 1-1 Description of task: Refine Business Model Task manager: Elizaveta Shashkova Deadline: CDR					
Task 1-2 Description of task: Keep Posting on Social Media Task manager: Valentine Bourgeois Deadline: CDR					
Task 1-3 Description of task: Write Second Version SD Task manager: Miriam Opazo-Mendez Deadline: CDR					
Task 1-4 Description of task: Recruit Members Task manager: Jasmine Rimani Deadline: CDR					
Task 1-5 Description of task: Decide Shipping Method Task manager: Jasmine Rimani Deadline: CDR					
Task 1-6 Description of task: Define Team for the FC Task manager: Jasmine Rimani Deadline: CDR					
Task 1-7 Description of task: Secure the ISAE Foundation as Sponsor Task manager: Valentine Bourgeois Deadline: CDR					

Work package number	09	Start Date	CDR	Deadline	RR
Work package title	Project Management				
Work package manager	Jasmine Rimani				
WP1 Objectives					
Task 1-1 Description of task: Final Version Business Model Task manager: Elizaveta Shashkova Deadline: RR					
Task 1-2 Description of task: Keep Posting on Social Media Task manager: Valentine Bourgeois Deadline: RR					
Task 1-3 Description of task: Write Third Version SD Task manager: Miriam Opazo-Mendez Deadline: RR					
Task 1-4 Description of task: Ship Material Task manager: Jasmine Rimani Deadline: RR					
Task 1-5 Description of task: Organize Team Mission Task manager: Jasmine Rimani Deadline: RR					

Work package number	10	Start Date	RR	Deadline	FC
Work package title	Field Campaign				
Work package manager	Jasmine Rimani				
WP1 Objectives					
Task 1-1 Description of task: Set Up Experiment Task manager: Jasmine Rimani Deadline: FC					
Task 1-2 Description of task: Inform General Public Task manager: Valentine Bourgeois Deadline: FC					
Task 1-3 Description of task: Write Final Report Task manager: Miriam Opazo-Mendez Deadline: FC					
Task 1-4 Description of task: Test on the Field Task manager: Jasmine Rimani Deadline: FC					
Task 1-5 Description of task: Interact with the Public Task manager: Valentine Bourgeois Deadline: FC					
Task 1-6 Description of task: Decommission Experiment Task manager: Jasmine Rimani Deadline: FC					

9.5 FOCA Rules

9.5.1 Emails from FOCA

If your drone weighs more than 0.5 kg but less than 30 kg then :

- The flight must take place more than 100 m from a gathering of people.
- The flight must take place more than 5 km from an aerodrome.
- The flight must take place outside the control zones.
- You must have liability insurance cover of CHF 1 million.
- The flight must take place outside the temporary prohibited or restricted areas.
- Legal provisions on data protection and privacy (Federal Data Protection Act).

I am sending you some useful links that you may wish to consult.

The following link refers to the drone guide - decision-making aid. It will tell you under what circumstances you would need authorisation. If you always stay on the green side, you do not need an authorisation.

The following link contains the RPAS map. On this map you will find the areas where you are not allowed to fly or only with a clearance.

The colours mean:

Yellow and red: no-fly zones for nature protection reasons. No UAV flights are allowed.

Blue: UAV flights are allowed up to 150 m without authorisation. Beyond that, an authorisation from the airport is required if the drone weighs more than 500g.

Purple: UAVs weighing less than 500 g can fly without authorisation. However, pilots must take into account that manned air traffic must not be endangered. The latter always has priority over unmanned air traffic. For drones weighing more than 500 g, a permit from the aerodrome is required.

From 01.01.2021, the new EU regulations will be adopted. You will find all the information on this subject (those published so far) and what this means for you on our website.

In your situation, this means that you are in the open A3 category because you do not have a CE class marking as you have a ”homemade” drone.

This means that:

- do not fly over gatherings of people;
- reasonably assume that no uninvolved persons are flown over;
- horizontal safety distance of at least 150 m from residential, commercial, industrial or recreational areas.

Furthermore, you will have to register and participate in an online course on our website at the beginning of 2021.

Please find attached the new EASA regulations.

Finally, please note that in Switzerland the cantons are authorised to establish additional flight restriction zones. This means that you should contact the cantons in which you intend to operate to find out whether other regulations have to be observed.

The register and the online training is organised by the FOCA. You should not spend too much time on the register and the course, as we try to keep this procedure as simple as possible. For the open category a few minutes, possibly hours if the topic is completely unknown. For the specific category it depends on the course modules, but the training will be more consistent. As soon as the register and the online training are ready, it will be possible to apply for your registration on our website. It is planned to implement these functions in January 2021. However, of course, if there are delays, it will not be mandatory to register and make the training as long as these functions are not available to the public.

9.5.2 For the Lucerne Canton and Flying Drones

SI asked their contact at Pilatus about the Lucerne Canton, and he said there is no more special requirements except to contact the airport Alpnach every time before take-off for clearance and fill in the form at the skyguide website to obtain a permit.

9.5.3 Drones Rules up to 11/02/2021

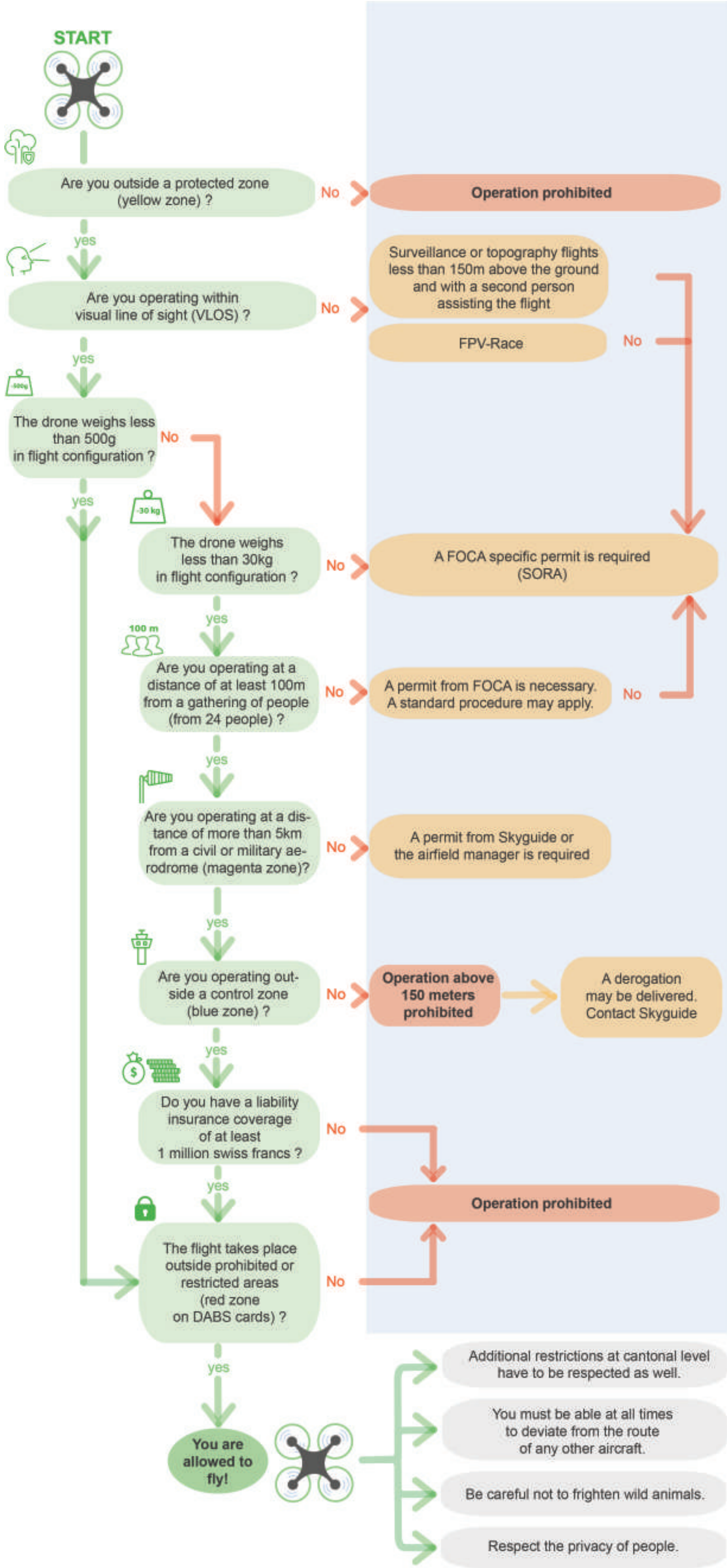


Figure 84: Guidelines for flying a drone in Switzerland from FOCA site [9]

10 Task Planning Domain and Problem Files

Listing 6: HDDL domain file

```
(define (domain rover_domain)
        (:requirements :typing :hierachie)
        (:types
                waypoint - object
                mode - object
                system - object
                camera - object
                control_center - object
                objective - object
                arTag - object
        )
        (:predicates
                (at ?arg0 - system ?arg1 - waypoint)
                (available ?arg0 - system)

                (can_traverse ?arg0 - system ?arg1 - waypoint ?arg2 - waypoint)

                (channel_free ?arg0 - control_center)

                (communicated_image_data ?arg0 - objective ?arg1 - mode)
                (communicated_health_data ?arg0 - system)
                (communicated_arTag_data ?arg0 - objective)
                (get_arTag ?arg0 - system ?arg1 - objective)

                (equipped_for_imaging ?arg0 - system)
                (equipped_for_arTag_reading ?arg0 - system)
                (equipped_for_health_monitoring ?arg0 -system)

                (got_health_data ?arg0 - system)

                (have_image ?arg0 - system ?arg1 - objective ?arg2 - mode)

                (on_board ?arg0 - camera ?arg1 - system)

                (supports ?arg0 - camera ?arg1 - mode)

                (visible ?arg0 - waypoint ?arg1 - waypoint)

                (visible_from ?arg0 - objective ?arg1 - waypoint ?arg2 - system)

                (visited ?arg0 - waypoint ?arg1 - system)

                (is_dependent ?arg0 - system ?arg1 - system)
                (navigate_tog ?arg0 - system ?arg1 - system)
        )


        (:task get_image_data
                :parameters (?objective - objective ?mode - mode)
                :precondition ()
                :effect ()
        )

        (:task read_arTag_data
                :parameters (?objective - objective)
                :precondition ()
                :effect ()
        )

        (:task evaluate_available_resources
                :parameters (?system - system)
                :precondition ()
                :effect ()
        )

        (:task navigate_abs
                :parameters (?system - system ?to - waypoint)
                :precondition ()
                :effect ()
        )

        (:task send_image_data
                :parameters (?system - system ?objective - objective ?mode - mode)
                :precondition ()
                :effect ()
        )

        (:task send_arTag_data
                :parameters (?system - system ?objective - objective)
                :precondition ()
                :effect ()
        )

        (:task call_back
                :parameters (?system2 - system ?system1 - system)
                :precondition ()
                :effect ()
        )

        (:task release_second_system
                :parameters (?system2 - system ?system1 - system)
                :precondition ()
                :effect ()
```

```
)

(:method m_call_back_ordering_0
        :parameters (?sys1 - system ?sys2 - system ?to - waypoint ?from - waypoint )
        :task (call_back ?sys2 ?sys1)
        :precondition (and
                (is_dependent ?sys2 ?sys1)
                (at ?sys2 ?from)
                (at ?sys1 ?to)
        )
        :subtasks (and
        (task0 (navigate_abs ?sys2 ?to))
        )
)

(:method m_release_second_system
        :parameters (?sys1 - system ?sys2 - system)
        :task (release_second_system ?sys2 ?sys1)
        :precondition (and
                (is_dependent ?sys2 ?sys1)
        )
        :subtasks (and
        (task0 (make_available ?sys2))
        )
)

(:method evaluate_available_resources_ordering_0
        :parameters (?system - system ?control_center - control_center)
        :task (evaluate_available_resources ?system)
        :precondition (and
                        (equipped_for_health_monitoring ?system)
        )
        :subtasks (and
           (task0(get_data_from_sensors ?system))
           (task1(send_system_state ?system ?control_center))
        )
        :ordering (and
                (< task0 task1)
        )
 )

(:method m_navigate_abs_1_ordering_0
        :parameters (?from - waypoint ?system - system ?to - waypoint)
        :task (navigate_abs ?system ?to)
        :precondition (and
                (at ?system ?from)
        )
        :subtasks (and
         (task0 (visit ?from ?system))
         (task1 (navigate ?system ?from ?to))
         (task2 (unvisit ?from ?system))
        )
        :ordering (and
                (< task0 task1)
                (< task1 task2)
        )
)

(:method m_navigate_abs_2_ordering_0
        :parameters (?system - system ?to - waypoint)
        :task (navigate_abs ?system ?to)
        :precondition (and
                (at ?system ?to)
        )
        :subtasks (
        )
)

(:method m_navigate_abs_3_ordering_0
        :parameters (?from - waypoint ?mid - waypoint ?system - system ?to - waypoint)
        :task (navigate_abs ?system ?to)
        :precondition (and
                (and
                        (not (at ?system ?to))
                        (not (can_traverse ?system ?from ?to))
                        (can_traverse ?system ?from ?mid)
                        (not (visited ?mid ?system))
                )
        )
        :subtasks (and
         (task0 (navigate ?system ?from ?mid))
         (task1 (visit ?mid ?system))
         (task2 (navigate ?system ?mid ?to))
         (task3 (unvisit ?mid ?system))
        )
        :ordering (and
                (< task0 task1)
                (< task1 task2)
                (< task2 task3)
        )
)


(:method m_navigate_abs_1_ordering_1
        :parameters (?from - waypoint ?sys1 - system ?sys2 - system ?to - waypoint)
        :task (navigate_abs ?sys1 ?to)
        :precondition (and
                (at ?sys1 ?from)
                (is_dependent ?sys2 ?sys1)
```

```
        )
        :subtasks (and
         (task0 (visit ?from ?system))
         (task1 (navigate_with ?sys1 ?from ?to))
         (task2 (unvisit ?from ?system))
        )
        :ordering (and
                (< task0 task1)
                (< task1 task2)
        )
)

(:method m_navigate_abs_3_ordering_1
        :parameters (?from - waypoint ?mid - waypoint ?sys1 - system ?sys2 - system ?to - waypoint)
        :task (navigate_abs ?system ?to)
        :precondition (and
                (and
                        (not (at ?sys1 ?to))
                        (not (can_traverse ?sys1?from ?to))
                        (can_traverse ?sys1 ?from ?mid)
                        (not (visited ?mid ?system))
                        (is_dependent ?sys2 ?sys1)
                )
        )
        :subtasks (and
         (task0 (navigate_with ?sys1 ?from ?mid))
         (task1 (visit ?mid ?system))
         (task2 (navigate_with ?sys1 ?mid ?to))
         (task3 (unvisit ?mid ?system))
        )
        :ordering (and
                (< task0 task1)
                (< task1 task2)
                (< task2 task3)
        )
)

(:method m_send_image_data_ordering_0
        :parameters (?l - control_center ?mode - mode ?objective - objective ?system - system ?x - waypoint ?y - waypoint)
        :task (send_image_data ?system ?objective ?mode)
        :precondition ()
        :subtasks (and
         (task0 (communicate_image_data ?system ?l ?objective ?mode))
        )
)


(:method m_send_arTag_data_ordering_0
        :parameters (?l - control_center ?objective - objective ?system - system ?x - waypoint ?y - waypoint)
        :task (send_arTag_data ?system ?objective)
        :precondition ()
        :subtasks (and
         (task0 (communicate_arTag_data ?system ?l ?objective))
        )
)



(:method m_read_arTag_data_ordering_0
        :parameters (?camera - camera ?objective - objective ?system - system ?waypoint - waypoint)
        :task (read_arTag_data ?objective)
        :precondition (and
                (and
                        (equipped_for_imaging ?system)
                        (on_board ?camera ?system)
                        (equipped_for_arTag_reading ?system)
                        (visible_from ?objective ?waypoint ?system)
                )
        )
        :subtasks (and

         (task0 (read_arTag ?system ?waypoint ?objective ?camera))
         (task1 (send_arTag_data ?system ?objective))
        )
        :ordering (and
                (< task0 task1)
        )
)



(:method m_get_image_data_ordering_0
        :parameters (?camera - camera ?mode - mode ?objective - objective ?system - system ?waypoint - waypoint)
        :task (get_image_data ?objective ?mode)
        :precondition (and
                (and
                        (equipped_for_imaging ?system)
                        (on_board ?camera ?system)
                        (supports ?camera ?mode)
                        (visible_from ?objective ?waypoint ?system)
                )
        )
        :subtasks (and
         (task0 (navigate_abs ?system ?waypoint))
         (task1(read_arTag_data ?objective))
         (task2 (take_image ?system ?waypoint ?objective ?camera ?mode))
         (task3 (send_image_data ?system ?objective ?mode))
        )
        :ordering (and
```

```
                    (< task0 task1)
                    (< task1 task2)
                    (< task2 task3)
          )
)

(:action navigate
          :parameters (?x - system ?y - waypoint ?z - waypoint)
          :precondition
                    (and
                              (can_traverse ?x ?y ?z)
                              (available ?x)
                              (at ?x ?y)
                    )
          :effect
                    (and
                              (not (at ?x ?y))
                              (at ?x ?z)
                    )
)

(:action navigate_with
          :parameters (?x - system ?y - waypoint ?z - waypoint ?k -system)
          :precondition
                    (and
                              (can_traverse ?x ?y ?z)
                              (available ?x)
                              (at ?x ?y)
                    )
          :effect
                    (and
                              (not (at ?x ?y))
                              (at ?x ?z)
                              (at ?k ?z)
                    )
)


(:action take_image
          :parameters (?r - system ?p - waypoint ?o - objective ?i - camera ?m - mode)
          :precondition
                    (and
                              (on_board ?i ?r)
                              (equipped_for_imaging ?r)
                              (supports ?i ?m)
                              (visible_from ?o ?p ?r)
                              (at ?r ?p)
                    )
          :effect
                    (and
                              (have_image ?r ?o ?m)
                    )
)

(:action read_arTag
          :parameters (?r - system ?p - waypoint ?o - objective ?i - camera)
          :precondition
                    (and
                              (on_board ?i ?r)
                              (equipped_for_imaging ?r)
                              (visible_from ?o ?p ?r)
                              (at ?r ?p)
                    )
          :effect
                    (and
                              (get_arTag ?r ?o)
                    )
)



(:action communicate_image_data
          :parameters (?r - system ?l - control_center ?o - objective ?m - mode)
          :precondition
                    (and
                              (have_image ?r ?o ?m)
                              (available ?r)
                              (channel_free ?l)
                    )
          :effect
                    (and
                              (not (available ?r))
                              (not (channel_free ?l))
                              (channel_free ?l)
                              (communicated_image_data ?o ?m)
                              (available ?r)
                    )
)

(:action communicate_arTag_data
          :parameters (?r - system ?l - control_center ?o - objective)
          :precondition
                    (and
                              (get_arTag ?r ?o)
                              (available ?r)
```

```
                        (channel_free ?l)
                )
            :effect
                (and
                        (not (available ?r))
                        (not (channel_free ?l))
                        (channel_free ?l)
                        (communicated_arTag_data ?o)
                        (available ?r)
                )
    )

    (:action visit
            :parameters (?waypoint - waypoint ?system - system)
            :precondition ()
            :effect
                (and
                        (visited ?waypoint ?system)
                )
    )

    (:action unvisit
            :parameters (?waypoint - waypoint ?system - system)
            :precondition ()
            :effect
                (and
                        (not (visited ?waypoint ?system))
                )
    )

    (:action make_available
            :parameters (?system - system)
            :precondition ()
            :effect
                (and
                        (available ?system)
                )
    )


    (:action get_data_from_sensors
            :parameters (?sys - system)
            :precondition ()
             :effect (and
                 (got_health_data ?sys)
             )
    )

    (:action send_system_state
            :parameters (?sys - system ?cc - control_center)
            :precondition
                (and
                        (available ?sys)
                        (channel_free ?cc)
                )
            :effect(and
                        (not (available ?sys))
                        (not (channel_free ?cc))
                        (communicated_health_data ?sys)
                        (channel_free ?cc)
                        (available ?sys)

                )
    )

)
            ...
```

Listing 7: HDDL problem file

```
(define
    (problem pfile1)
    (:domain  rover_domain)
    (:objects
            waypoint0 - waypoint
            waypoint1 - waypoint
            waypoint2 - waypoint
            waypoint3 - waypoint
            waypoint4 - waypoint
            waypoint5 - waypoint
            waypoint6 - waypoint
            waypoint7 - waypoint
            waypoint8 - waypoint

            depth - mode
            fisheye - mode
            rover0 - system
            drone1 - system
            camera0 - camera
            camera1 - camera
            camera2 - camera
            camera3 - camera
            general - control_center
            objective0 - objective
            objective1 - objective
            objective2 - objective
    )
    (:htn
```

```
        :parameters ()
        :subtasks (and
         (task0 (release_second_system drone1 rover0))
         (task1 (get_image_data objective0 depth))
         (task2 (get_image_data objective1 depth))
         (task3 (get_image_data objective0 fisheye))
         (task4 (get_image_data objective2 fisheye))
         (task5 (call_back drone1 rover0))
         (task6 (evaluate_available_resources rover0))
         (task7 (evaluate_available_resources drone1))

        )
        :ordering (and
                (< task0  task1)
                (< task0  task2)
                (< task0  task3)
                (< task0  task4)
                (< task2  task5)
                (< task3  task5)
                (< task1  task5)
                (< task4  task5)
                (< task5  task6)
                (< task6  task7)

        )
)
(:init

        (channel_free general)
        (at rover0 waypoint0)
        (available rover0)
        (equipped_for_imaging rover0)
        (equipped_for_health_monitoring rover0)
        (equipped_for_arTag_reading rover0)

        (can_traverse rover0 waypoint0 waypoint5)
        (can_traverse rover0 waypoint5 waypoint0)
        (can_traverse rover0 waypoint0 waypoint4)
        (can_traverse rover0 waypoint4 waypoint0)
        (can_traverse rover0 waypoint0 waypoint1)
        (can_traverse rover0 waypoint1 waypoint0)
        (can_traverse rover0 waypoint1 waypoint2)
        (can_traverse rover0 waypoint2 waypoint1)
        (can_traverse rover0 waypoint1 waypoint4)
        (can_traverse rover0 waypoint4 waypoint1)
        (can_traverse rover0 waypoint1 waypoint5)
        (can_traverse rover0 waypoint5 waypoint1)
        (can_traverse rover0 waypoint4 waypoint0)
        (can_traverse rover0 waypoint0 waypoint4)
        (can_traverse rover0 waypoint4 waypoint5)
        (can_traverse rover0 waypoint5 waypoint4)
        (can_traverse rover0 waypoint5 waypoint6)
        (can_traverse rover0 waypoint6 waypoint5)

        (at drone1 waypoint0)
        (equipped_for_imaging drone1)
        (equipped_for_health_monitoring drone1)
        (equipped_for_arTag_reading drone1)
        (is_dependent drone1 rover0)

        (can_traverse drone1 waypoint0 waypoint5)
        (can_traverse drone1 waypoint5 waypoint0)
        (can_traverse drone1 waypoint0 waypoint4)
        (can_traverse drone1 waypoint4 waypoint0)
        (can_traverse drone1 waypoint0 waypoint1)
        (can_traverse drone1 waypoint1 waypoint0)
        (can_traverse drone1 waypoint1 waypoint2)
        (can_traverse drone1 waypoint2 waypoint1)
        (can_traverse drone1 waypoint1 waypoint3)
        (can_traverse drone1 waypoint3 waypoint1)
        (can_traverse drone1 waypoint1 waypoint4)
        (can_traverse drone1 waypoint4 waypoint1)
        (can_traverse drone1 waypoint1 waypoint5)
        (can_traverse drone1 waypoint5 waypoint1)
        (can_traverse drone1 waypoint2 waypoint4)
        (can_traverse drone1 waypoint4 waypoint2)
        (can_traverse drone1 waypoint2 waypoint3)
        (can_traverse drone1 waypoint3 waypoint2)
        (can_traverse drone1 waypoint4 waypoint3)
        (can_traverse drone1 waypoint3 waypoint4)
        (can_traverse drone1 waypoint8 waypoint3)
        (can_traverse drone1 waypoint3 waypoint8)
        (can_traverse drone1 waypoint4 waypoint0)
        (can_traverse drone1 waypoint0 waypoint4)
        (can_traverse drone1 waypoint4 waypoint5)
        (can_traverse drone1 waypoint5 waypoint4)
        (can_traverse drone1 waypoint4 waypoint6)
        (can_traverse drone1 waypoint6 waypoint4)
        (can_traverse drone1 waypoint4 waypoint8)
        (can_traverse drone1 waypoint8 waypoint4)
        (can_traverse drone1 waypoint4 waypoint7)
        (can_traverse drone1 waypoint7 waypoint4)
        (can_traverse drone1 waypoint5 waypoint6)
        (can_traverse drone1 waypoint6 waypoint5)
        (can_traverse drone1 waypoint5 waypoint7)
        (can_traverse drone1 waypoint7 waypoint5)
        (can_traverse drone1 waypoint7 waypoint6)
        (can_traverse drone1 waypoint6 waypoint7)
        (can_traverse drone1 waypoint7 waypoint8)
```

```
                (can_traverse drone1 waypoint8 waypoint7)

                (on_board camera0 rover0)
                (supports camera0 depth)
                (on_board camera1 rover0)
                (supports camera1 depth)

                (on_board camera2 drone1)
                (supports camera2 depth)
                (on_board camera3 drone1)
                (supports camera3 fisheye)

                (visible_from objective0 waypoint5 rover0)
                (visible_from objective0 waypoint6 drone1)
                (visible_from objective1 waypoint8 drone1)
                (visible_from objective2 waypoint0 rover0)
                (visible_from objective2 waypoint2 rover0)
                (visible_from objective2 waypoint4 rover0)
                (visible_from objective2 waypoint1 drone1)
        )
)
                ...
```

List of Figures

1 CORODRO project concept. A detailed description is given in Section 2.6 11
2 Stakeholders Analysis Overview 13
3 Risk matrix before mitigation. 20
4 Risk matrix after mitigation. 20
5 BUSINESS MODEL CANVAS (More information can be found in Annex 9.3.1) 23
6 Gantt chart for the CoRoDro Team 30
7 Work-breakdown structure for review of the CoRoDro Team 31
8 Concise development plan for the CoRoDro Team spanning the academic year. . . . 31
9 Development Plan for the month of June 32
10 ISAE-SUPAERO gym. 32
11 Martian terrain at CNES, Toulouse 33
12 Drone testing Terrain of ONERA, Esperce 33
13 JTS, ISAE-Supaero's video club 33
14 Time dedicated to the project by each team member 34
15 Terrain setup 37
16 Concept of Operations of CORODRO 39
17 VFC Timeline for one day of testing 49
18 Rover and drone that will be used during the FC 52
19 Map generation algorithm for the drone 62
20 Modules of the map generation algorithm and high-level functions for the drone . . . 64
21 Localization and Real-Time Mapping process flow for the drone 64
22 An example of 2D Occupancy Grid (LiDAR configuration) that has been optimized thanks to the Depth Filtering Module 66
23 Conversion Module block diagram 67
24 An example of Point Cloud obtained through Localization and Real-Time algorithm (D435i configuration) after the first step of the Conversion Module 67
25 Activity Diagram of the Drone Mapping State-machine 68
26 Class Diagram of the Drone Mapping algorithms 69
27 Localization and Real-Time Mapping block diagram for the rover 70
28 Modules of the obstacle detection software and high-level functions for the rover . . 71
29 Obstacle detection process flow for the rover 72
30 Block diagram for Points of Interest detection 74
31 Cuboid with AR-tags as Point of Interest 75
32 Drone mapping phase block diagram 77
33 Drone mapping flowchart 77
34 Drone referencing points 78
35 Drone pattern for mapping 79
36 Drone mission phase block diagram 80
37 Drone mission flowchart 81
38 Rover mission phase block diagram 81
39 Rover mission flowchart 82
40 Rover localization in a known map. 85
41 Path Planning set-up for the rover 86
42 Modules of the task planner and high-level functions 87
43 Task planner block diagram 89
44 Mission Analysis EFFBD for the IGLUNA mission 90
45 Task planner process flow for the rover 91
46 Task planner process flow for the drone 92
47 Overall workflow of the task planning code 92

48 Simple HDDL problem benchmark (code in Annex 10) . . . 93
49 Movement and targeting model in the HDDL problem file (code in Annex 10) . . . 93
50 Workflow of action for the drone mapping phase . . . 95
51 Resulting state machine for the rover task planning . . . 96
52 Localization and Real-Time Mapping DEM map test SM-1.D. Resolution 0.03 m . . . 113
53 Localization and Real-Time Mapping occupancy grid test SM-1.D. Resolution 0.03 m 114
54 Localization and Real-Time Mapping DEM map test SM-1.L. Resolution 0.03 m . . . 114
55 Localization and Real-Time Mapping occupancy grid test SM-1.L before and after depth filtering module implementation. Resolution 0.03 m . . . 115
56 Test SM-2 set-up . . . 116
57 Localization and Real-Time Mapping DEM Map test SM-2.D Resolution 0.03 cm . . . 116
58 Localization and Real-Time Mapping occupancy grid test SM-2.D. Simplistic version of Conversion Module . . . 117
59 Localization and Real-Time Mapping DEM map test SM-2.D. Before and after the second step of the Conversion Module . . . 117
60 Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, D435i configuration resolution 8cm . . . 118
61 Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, D435i configuration resolution 32cm . . . 118
62 Localization and Real-Time Mapping DEM map test SM-2.L. Resolution 0.03 cm . . . 119
63 Localization and Real-Time Mapping occupancy grid test SM-2.L. Simplistic version of Conversion Module . . . 119
64 Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, LiDAR configuration resolution 8cm . . . 120
65 Localization and Real-Time Mapping test SM-3.D. Bench picture from D435 . . . 120
66 Localization and Real-Time Mapping test SM-3.D. Grass picture from D435 . . . 121
67 Localization and Real-Time Mapping DEM map test SM-3.D. Resolution 0.03 cm . . . 121
68 Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, D435i Configuration Outside, resolution 8cm . . . 122
69 Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, D435i Configuration Outside, resolution 32cm . . . 122
70 Localization and Real-Time Mapping DEM map test SM-3.L. Resolution 0.03 cm . . . 123
71 Localization and Real-Time Mapping 2D Occupancy Grid obtained with Conversion Module, LiDAR Configuration Outside, resolution 8cm . . . 123
72 Set-up of the test of Merging Frames of the two robotic systems through features detection . . . 125
73 State machine of the Rover Path-planning . . . 125
74 State machine of the Drone mapping phase . . . 126
75 ConOps timeline . . . 127
76 Set up of the Outdoor test in Esperce. . . . 135
77 3D map of Outdoor testing in Esperce. Resolution 3cm . . . 136
78 2D Occupancy grid of Outdoor testing in Esperce. Resolution 8cm . . . 136
79 Lab T265 position assessment test for the drone. . . . 138
80 Lab T265 position assessment test for the rover. . . . 138
81 Team Structure . . . 187
82 Design reference mission for the exploration of the lunar lava tubes . . . 197
83 The plot shows the envisioned revenue stream for the first six years of activity. We assumed a cost inflation rate of 0.6% per year in the Eurozone and a steady growth of 5% for both our main stream channels . . . 200
84 Guidelines for flying a drone in Switzerland from FOCA site [9] . . . 209

List of Tables

2 Stakeholders Analysis . . . 14
3 Mission Objectives . . . 16
4 Project Goals . . . 17
5 Project Top-Level Requirements . . . 18
6 Risk Analysis . . . 19
7 Functional Requirements . . . 25
8 Performance Requirements . . . 26
9 Interface Requirements . . . 26
10 Safety Requirements . . . 26
11 Constraints Requirements . . . 27
12 Communication Requirements . . . 27
13 Main characteristics of the sensors . . . 28
14 Benchmarks . . . 28
15 Trade-off to decide which camera was the most suitable payload to enable localization with the drone . . . 28
16 Trade-off to decide whether to use a LiDAR instead of the D435i . . . 29
17 Planned and effective time allocation for the CoRoDro project. The values will be updated at each revision . . . 35
18 Financing in money and kind of the CoRoDRo Team . . . 35
19 Estimated expenses for students for the FC (apart from the goodies, which is a general expense of fixed value for the team) . . . 36
20 Envisioned test for the IGLUNA Campaign . . . 40
21 Shipping mission . . . 41
22 Mapping . . . 42
23 End/Beginning of the day tasks . . . 43
24 Control Room ConOps . . . 43
25 Drone mission ConOps . . . 44
26 Rover mission ConOps . . . 45
30 Total Energy budget per ConOps . . . 45
27 Set-up . . . 46
28 Contingency situations . . . 47
29 Virtual Field Campaign ConOps . . . 48
31 Total data budget per ConOps . . . 48
32 List of main project components . . . 51
33 List of equipment: Drone . . . 52
34 List of components: Rover . . . 53
35 Mass budget of the drone . . . 53
36 Mass budget of the rover . . . 54
37 Power budget of the drone . . . 54
38 Power budget of the rover (Part A) . . . 55
39 Power budget of the rover (Part B) . . . 55
40 Data budget of the drone . . . 56
41 Data budget of the rover . . . 56
42 Data budget of the drone in different modes . . . 56
43 Data budget of the rover in different modes . . . 56
44 Data collection for Path Planning. . . . 58
45 Data collection for Localization and Real-Time Mapping. . . . 59
46 Data collection for Path Planning. . . . 59

47 Summary of the map generation software modules, description, coding language, interfaces for the drone . . . 63
48 Summary of the obstacle detection software modules, description, coding language, and interfaces for the rover . . . 72
49 Summary of the software modules, description, coding language and interface for the drone mapping phase . . . 79
50 Summary of the software modules, description, coding language and interface . . . 82
51 Summary of the software modules, description, coding language and interface . . . 88
52 Verification matrix of the requirements . . . 100
53 Test Plan to verify the requirements and be successful during the FC . . . 101
54 Safety and Environment Checklist. . . . 127
55 Drone mission checklist . . . 128
56 Rover mission checklist . . . 129
57 Mapping checklist . . . 130
58 Terrain set-up checklist . . . 130
59 Before testing checklist . . . 130
60 Control room checklist . . . 131
61 End/beginning of the day task checklist . . . 131
62 Equipment needed for the Field Campaign . . . 132
89 Envisioned European funding . . . 199
90 Envisioned cost structure start-up . . . 200