# A NOVEL APPROACH TO PLANETARY ROVER GUIDANCE, NAVIGATION AND CONTROL BASED ON THE ESTIMATION OF THE REMAINING USEFUL LIFE

**Jasmine Rimani**[1, *], **Stéphanie Lizy-Destrez**[2], and **Nicole Viola**[3]

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

[2] *Department of Aerospace Vehicles Designs and Control, ISAE-SUPAERO, Toulouse, France,* Email:
stephanie.lizy-destrez@isae-supaero.fr

[3] *Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy,* Email:nicole.viola@polito.it

[*] Corresponding author
Email: jasmine.rimani@polito.it

This decade will be remembered as the era when humankind returns to the lunar surface. Research centers, industries, and universities are showing great interest in future missions to the Moon. The main drivers of this new era of exploration are the establishment of a permanent human outpost and the exploitation of in-situ resources. From an engineering perspective, the Moon is an ideal testbed for autonomous operations and technologies that enable deep space exploration. In this context, the mission operations of a lunar rover are closely linked to the performance of the guidance, navigation, and control (GNC) subsystem. These performances, in turn, depend on the system's health, as measured by parameters such as battery level.

This paper analyzes an adaptive and autonomous GNC system for a lunar rover. The GNC relies on the failure identification, isolation, and recovery (FDIR) subsystem to estimate available resources and autonomously plan a path. Specifically, the guidance node selects the optimal path to visit a series of waypoints. Each waypoint offers different scientific rewards. This new approach addresses the needs of deep-space exploration systems with limited communication links. Autonomy and adaptability to unforeseen events are essential. There is a paradigm shift: Earth's mission control delegates some decision-making tasks to the exploration system. This is done primarily to preserve mission success. The Moon, with its relatively short communication delay, is the perfect site to test and refine these new approaches.

Overall, the GNC comprises a navigation node, a guidance node, a resource-estimation node, and a control node. The navigation node simultaneously maps and localizes the rover in the lunar environment. The resource estimation node continuously evaluates the system's remaining useful life (RUL) and monitors health parameters. The proposed algorithm autonomously generates the optimal plan to maximize mission return while preserving system health and avoiding obstacles. The algorithm may choose to skip low-reward waypoints to conserve resources for more valuable sites. The key features of the proposed algorithm are its adaptability to unforeseen events and its onboard decision-making autonomy, which optimizes the rover's path.

**Keywords: Operations, Planetary Rovers, Planning Algorithms, Lava Tubes, Routing Problems**.

**Acronyms**

**DC:** Direct Current

**DOF:** Degrees of Freedom

**ECSS:** European Committee for Space Standardization

**EKF:** Extended Kalman Filter

**FMECA:** Failure mode, effects, and criticality analysis

**FTA:** Failure Tree Analysis

**GNC:** Guidance, Navigation, and Control

**HDDL:** hierarchical domain definition language

**IMU:** Inertial Measurement Unit

**ISAE-SUPAERO:** Institut Supérieur de l'Aéronautique et de l'Espace

**LRO:** Lunar Reconnaissance Orbiter

**PID:** Proportional Integrative Derivative

**PoliTO:** Politecnico di Torino

**ROS:** Robot Operating System

**RUL:** Remaining Useful Life

**SaCLaB:** Space Advanced Concepts Laboratory

## 1. Introduction

This decade has seen a rise in interest among space agencies and industries in human colonization of the Moon [1], [2]. The Earth's natural satellite can serve as a gateway to the exploration of Mars and beyond. It can be exploited as a testing facility to advance the fundamentals of disruptive technologies for both robotic and human space exploration. The exciting aim is to create a permanent human colony on the Moon. Among sites of interest for future human settlement, the lunar lava tubes have been identified as potential targets [3], [4]. Various studies have shown that the temperature and radiation environment within this volcanic architecture may be more human-friendly than in any other region of the equatorial Moon [5], [6]. The focus of the scientific community lies mainly in three lava tubes, Mare Ingenii, Mare Tranquillitatis, and Marius Hills (fig. 1. They have been detected and studied thanks to satellite missions such as Kaguya/Selene, LRO, and Clementine[7]– [10].. However, from satellite images, it is difficult to determine the effective morphology of those underground tunnels, their dimensions, and their geological characteristics. Therefore, precursor robotic missions are envisioned to map lava tubes, assess their safety for humans, and study their geological characteristics [3], [4]. Due to the challenging communication setup during lava tube exploration [3], [4], the foreseen exploration systems should be able to operate autonomously across 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 based on their resources while planning the trajectory. They should be able to respect an *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 generally, the operational domain presents many challenges in both robotic and human space exploration. As space exploration missions grow in complexity, there is a need to balance mission return, autonomy level, and the workload of control centre operators. Various space agencies, companies, and universities are engaged in defining a broad spectrum of technological maturation studies toward autonomous operations and navigation [11]–[14]. In this context, the SaCLaB team at ISAE-SUPAERO and PoliTO is collaborating on new algorithms to enhance the autonomous operations of space robotic systems. The reference mission is the exploration of the lava tubes.

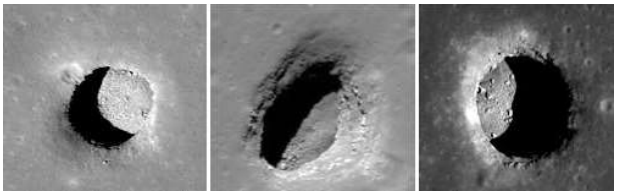

*Figure 1: Lava tubes (Marius Hills, left; Mare Ingenii, center; Mare Tranquillitatis, right). Image from SELENE and LRO missions [3].*

The systems under study are a rover and a drone, dedicated to mapping and characterizing the lava tubes. The word “drone” refers to a hopping/flying system equipped with a thruster that can perform short flights when required (for example, during the descent into the lava tube from the skylight). The focus is on the operational layer: information about the overall health of systems is considered an essential statistic for designing more autonomous systems. This paper focuses on the rover-related operations.

The rover should map the lava tubes on the Moon's surface and deploy drones near them. The study began with the analysis of its “traverse mode operations”. In this paper, all the operations that engage the mobility system of the rover to move on the lunar surface are labeled as ”traverse mode operations”. During the nominal mode of the ”traverse operations” the rover may switch between two main operative modes: (i) ”Local Reactive” when there are many obstacles in the map, (ii) ”Light Local Guidance” when the map has fewer obstacles fig.2. The software used to study and carry on the mission analysis and to define the operations is Vitech Genesys 7.0 [15].

During the traverse, the rover consumes the energy stored in the batteries. In the event of faults such as a stuck motor or parasitic current, the rover will draw total available energy faster. Therefore, it becomes interesting to study an algorithm that can help the rover reconfigure its path based on the available RUL. The rover should be able to choose the shortest path that connects different sites of scientific interest. Each site of interest has a return in terms of scientific operations that the rover can perform. From the available estimated RUL, the robotic system should be able to choose the best set of sites of interest that it can cover. Similar studies have been carried out by [16], [17] on a testbed linking RUL and operations. However, the emphasis was on timely detection and recognition of failures or faults in the subsystems of the testing rover. On the contrary, this study focuses on path-planning algorithms that receive RUL as input to optimize the rover's path. This paper aims to analyze what is usually framed as a ”routing problem” applied to a rover [18], [19]. Hence, the algorithm presented in this paper targets path planning and task planning.

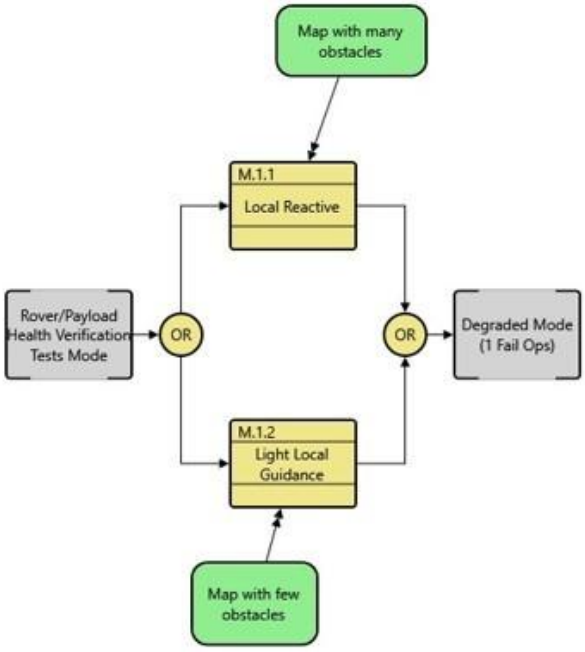


*Figure 2: Nominal traverse mode for the lunar rover (Vitech Genesys 7.0 [15]).*

At a high level, a simulation framework similar to the ”travel salesman problem” [20] is run. The algorithm is, then, coupled with an A* algorithm [21] as the global planner and a dynamic window [21] as a local planner. To create a challenging grid for the global path planner, a code that lays a plausible distribution of rocks on the Moon, with various dimensions and sizes, is run before the actual simulation [22]. At the same time, to make the simulation more realistic, a battery consumption model is implemented, along with state equations for the rover, its motor, and sensors

[23]. In this paper, a solution to this problem is presented. The main objective is to develop a planning algorithm that optimizes the operations of the lunar rover. The remaining part of the paper is organized as follows: (i) 2 introduces the framework of the study; (iii) 3 explains the mathematical and logical framework of the work as well as its results; (iii) 4 wraps up the paper, the main outputs, as well as the expected developments.

## 2. Case Study

The exploration mission to the lava tubes requires a high level of autonomy: a continuous, robust communication link cannot be established beneath the lunar surface. The system that will explore the lava tubes should autonomously decide the best path to follow and the most interesting targets to reach within the timeframe of one battery discharge. Following this logical flow, the main objective of this paper is to develop a task-planning algorithm that can reconfigure the system's path based on available resources, e.g., the overall battery level. The robotic system used for algorithm testing is a small skid-steered rover with an average battery life of 5 hours. The testbed in ISAE-SUPAERO's laboratories has been simulated in ROS-Gazebo and with a six-DOF model in Python. This model has been directly validated against the testing platform following the guidelines in [24]. It is primarily used to assess the feasibility of the studied algorithms during the initial development stages. The foreseen reference mission consists of touching a series of waypoints with different scientific rewards while avoiding obstacles. During the conceptualization of the task planning algorithm, the rover Python model is used. The implemented Python code accounts for the motor's propulsive force and terrain friction [25]. In the Earth-analog mission simulation, aerodynamic drag is treated as a force acting on the rover. Its contribution is small; however, it is relevant on a small testing platform with a limited linear velocity. The 6 DOF equations are explicated from eq. 2 to eq.4.

$$\dot{\mathbf{v}} = \frac{1}{m}\mathbf{F} - \omega x \mathbf{v} \quad (1)$$

$$\dot{\omega} = \mathbf{I}^{-1}(\mathbf{M} - \omega x \mathbf{I}\omega) \quad (2)$$

$$\mathbf{F} = \mathbf{F_p} - \mathbf{F_a} - \mathbf{F_f} \quad (3)$$

$$\mathbf{M} = \mathbf{M_p} - \mathbf{M_f} \quad (4)$$

Where:

- $x$ indicates the external product between two vectors;
- $\mathbf{v}$ is the rover velocity;
- $\dot{\mathbf{v}}$ is the rover linear acceleration;
- $\omega$ is the rover angular velocity;
- $\dot{\omega}$ is the rover angular velocity;
- m is the rover mass;
- $\mathbf{I}$ is the rover inertia matrix;
- **Fp** is the force generated by the rover motors.
- **Fa** is the aerodynamic force acting on the rover (used only for Earth-analogue simulations).
- **Ff** is the friction force generated by the wheels' interactions with the terrain.
- **MP** is the moment generated by the rover motors.
- **Mf** is the friction moment generated by the wheels' interactions with the terrain.

Beyond the rover's dynamics, the simulator includes a model of the wheels' DC motors for current drawn, rotational velocity, and motor temperature [16], [17]. The battery is simulated following the guidelines in [23]. The sensors and the relatively noisy readings are implemented as described in [16], [17], [25], [26]. The testbed is equipped with encoders, an IMU, and a stereo camera. The same sensors are simulated in Gazebo and reconstructed in the simulation code. During the test, in ISAE-SUPAERO, an optitrack is used to estimate the rover's absolute position, velocity, and

response to the commands. ROS gives the framework for the navigation node, using a pre-existing map and an EKF [27] for localization. The simulated map is derived from the [28] elevation model and is explored using the built-in ROS mapping algorithms. The terrain in the lunar equatorial zone is enriched with a ”likely to be” rock distribution following the guidelines from [22]. In fact, the satellite imagery is not fine enough to capture the true distribution of rocks. The same problem is experienced with Earth satellite maps. Therefore, an algorithm is run before the simulation to lay a rock bed with rocks of different sizes. The rover can traverse some obstacles, but others are too difficult to overcome. Usually, for a four-skid-steered rover, the maximum traversable obstacle's height is equal to half of its wheel diameter. The global path planner takes that information into account during its computation to estimate the best path. The control of the rover trajectory is executed with a PID controller that considers the distance from the target and the difference in ”heading” between the rover trajectory and the target position.

$$v_l = v_{des} - [K_p(\psi_{des} - \psi) + K_i \int (\psi_{des} - \psi)dt + K_d(r_{des} - r)] \quad (5)$$

$$v_r = v_{des} + [K_p(\psi_{des} - \psi) + K_i \int (\psi_{des} - \psi)dt + K_d(r_{des} - r)] \quad (6)$$

Where:

- *Vl* is the linear velocity provided by the motors on the left side of the rover.
- *Vr* is the linear velocity provided by the motors on the right side of the rover.
- *vdes* is the desired velocities for the rover;
- *Kp,Ki, and Kd* are the PID gains.
- $\psi$*des and rdes* are the desired heading and rover angular velocity, respectively.
- $\psi$*,r* are the real heading and rover angular velocity.

The controller's maximum voltage limits the maximum reachable motor velocities. For the testing platform, the maximum allowed linear velocity is around 0.5 m/s, while the rotational velocity is 3°/s.

The paper focuses on the guidance layer and its interaction with the task planner. The guidance for the rover is divided into a global planner that estimates the best path when the global map is known, and a local path planner. The global path planner provides the task planner with information on a probable path length to the task. Two global path planners have been implemented, the A* and the D* [21]. However, the results presented in this paper are relative to the A* algorithm. The local path planner refines the trajectory to make it smoother for the rover to follow. Moreover, the local path planner avoids unforeseen obstacles along the rover's path. The algorithm uses the“dynamic window” path planner as the rover's local planner [21].

## 3. Task Planning Algorithm

Fig. 3 gives a logical overview of the algorithm implemented for path planning. The objective of the task planner applied to the battery can be summarized as follows: *given a series of waypoints with different scientific rewards, can the rover find the best path to touch as many waypoints as possible, maximizing the overall scientific reward while accounting for available resources?* The algorithms begin acquiring a list of interesting targets that the system should visit. The targets may be based on interesting objectives defined by the control centre, on measurements from the rover cameras, or on measurements from other robotic systems in the field collaborating with the rover. The distance can be estimated with different levels of fidelity: (i) if the obstacles presented on the field are known, it is possible to use an A* algorithm to estimate the best path; (ii) or, if the distribution is not known and cannot be estimated, it is

possible to consider the geometric distance between the waypoints as the first guess. In all cases, the planning algorithm is re-run at each site with the new reading for the overall consumed battery and the actual distance covered. In these measurements, the ”guidance duty cycle” [29] is considered as well. The ”guidance duty cycle” indicates how much of the battery is dedicated to movement during the rover mission. When the site of interest is reached, the testing rover will take a picture of the target, whereas in the real planetary mission under design, the rover will have to perform measurements using its scientific payload. The guidance duty cycle can be linked to the battery's overall RUL and the maximum reachable distance for the system. From the battery model, knowing the overall drawn current, it is possible to estimate the RUL, fig. 4. While knowing the current and the voltage of the battery, as well as the ”guidance duty cycle”, it is possible to estimate the rover velocity and the maximum coverable rover distance indicates as *d_asymptotic*, eq.7, fig.s 5 and 6. Where:

- $E$ is the battery energy;
- $\eta$ is the ”driving duty cycle”;
- $Crr$ is the terrain resistance coefficient set at 0.15 [29];
- $m$ is the rover mass;
- g is the gravitational acceleration.

$$d_{asymptotic} = \frac{E\eta}{C_{rr}mg} \qquad (7)$$

**Algorithm 1** Path Planner for battery related faults

```
Define list of waypoints
for Every waypoint ii in the list to be touched do
    Remove waypoints already touched
    Define "Guidance Duty Cycle" d_gc
    Evaluate drawn current level i
    Evaluate remaining RUL rul
    Evaluate maximum reachable distance by the rover for a given current and "Guidance Duty Cycle" d_coverable
    for Every other waypoint jj do
        Calculate the distance between the waypoints, edge_dist
        if d_coverable < any(edge_dist) then
            Stop Traverse Mode
            Evaluate the shortest path to touch jj waypoints from the ii waypoint, short_path
            Evaluate the overall scientific reward, edge_reward
            if edge_dist < d_coverable then
                if edge_reward < previous edge_reward then
                    Store new best path
                end if
                if edge_reward = previous edge_reward then
                    if edge_dist < previous edge_dist then
                        Store new best path
                    end if
                end if
            end if
        end if
    end for
    Provide new traffic plan to the path planner
    if No change in drawn current level then
        Go to to next waypoint
        if Last waypoint touched then
            Stop Traverse Mode
        end if
    end if
end for
```

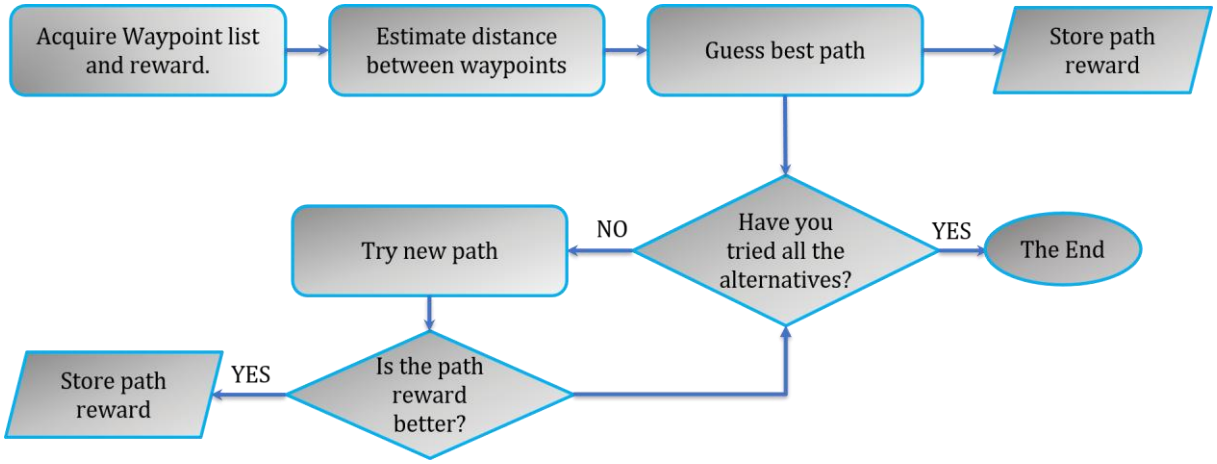


*Figure 3: Flowchart explaining the logical process used for the task planning algorithm.*

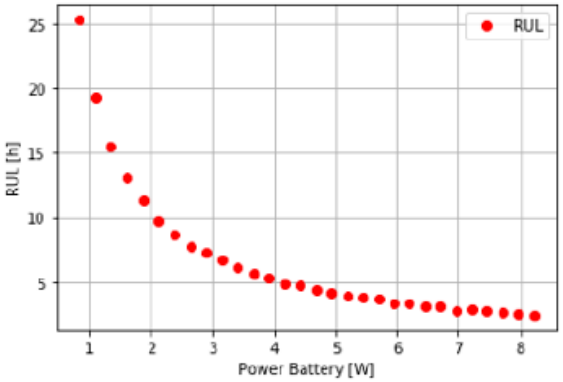


*Figure 4: Battery RUL for a given consumed battery power.*

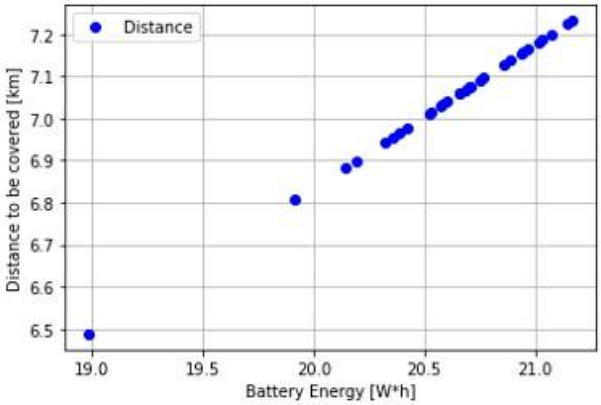


*Figure 5: Maximum distance that can be covered by the rover for a given battery energy and an 80% guidance duty cycle.*

The best path to follow is estimated using a modified traveling salesman problem [20]. One of the outputs from the implemented algorithm is the computation of the shortest path to touch a series of waypoints knowing the rock distribution, the overall free-mean space, and scientific reward for each site as shown in fig. 7. However, the algorithm is able to reconfigure the rover path based on the RUL, the chosen duty cycle, drawn current and the overall maximum allowed distance, as shown in fig. 8. Overall, the task planning algorithm can provide the global path plan to be followed by the rover, given several constraints that we labeled as available resources.

The path will be refined by the local path planner and the obstacle avoidance algorithm. Moreover, the actual covered distance will be updated based on the rover's actual performance along its trajectory, including maximum velocity and rotation rate.

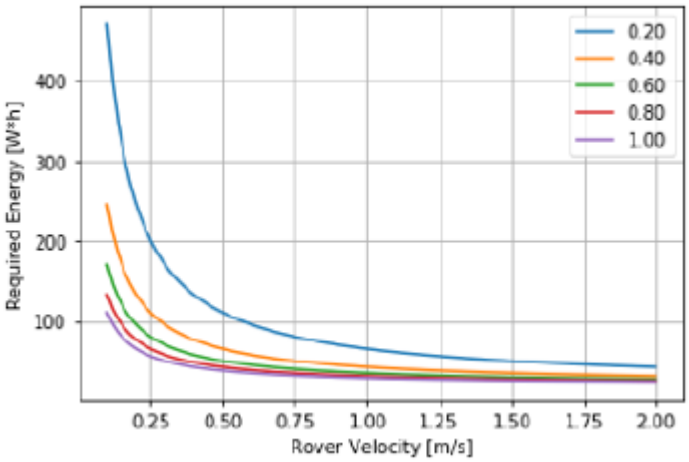


*Figure 6: Required energy needed for driving with a given rover velocity, for given duty cycle (from 20% (0.2) to 100% (1.00))*

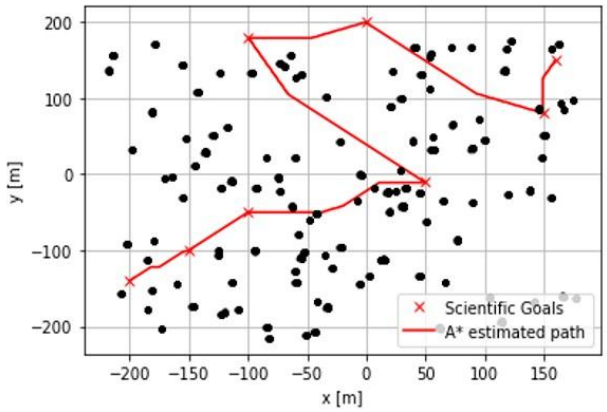


*Figure 7: Shortest path to touch a series of waypoints using A* and the task planning algorithm.*

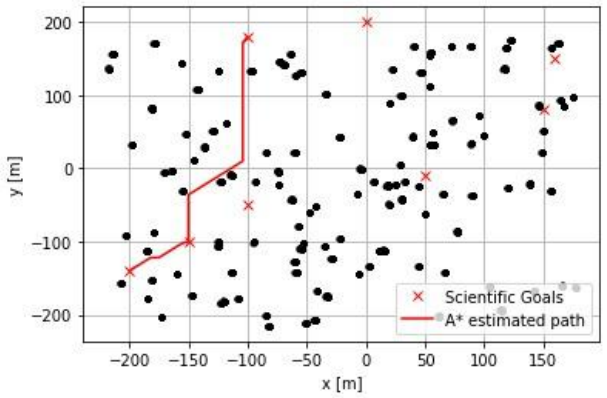


*Figure 8: Reconfigure trajectory based on RUL and "guidance duty cycle".*

## 4. Conclusions

The paper presents an overview of the GNC architecture implemented for the system under study. The guidance layer is merged with a task-planning layer to determine the best trajectory to follow based on the estimated battery level, the“guidance duty cycle,” and the scientific rewards of the various waypoints. The planner considers even contingency scenarios such as the one presented in this paper. Indeed, the battery may draw more quickly if a fault or failure is present in the system, thereby consuming more energy. Therefore, the current is modelled as the sum of the nominal system needs and some parasitic loads derived from faults or failures. The rover's maximum allowed distance and estimates determine the best path to maximize the overall scientific reward, choose the shortest path, and avoid exceeding energy consumption during traverse mode operations. However, beyond faults affecting the battery, the task planner should consider faults such as a stuck motor or improper sensor readings that can change the system's ”available” resources. These contingency situations are now under study for the traverse mode. The likelihood of failure or fault can be estimated from historical data in the literature. Those data are coupled with FTA and FMECA analyses to understand the effects of faults and failures on the overall system operational layer. Therefore, the work presented in this paper will evolve to consider path reconfiguration across different contingency scenarios. Thus, the failure and fault analysis of the rover mobility system and GNC will be among the expected outcomes of the study. Nevertheless, a task planner should consider and optimize the tasks foreseen for the system of interest. Hence, the task planner will be extended and formalized using HDDL. This extension will help create a cluster of envisioned actions for the rover to perform across different degraded scenarios.

**References**


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

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

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

[4] W. Whittaker, *Exploration of planetary skylights and tunnels*, 2014.

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

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

[7] S. J. Lawrence 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*, vol. 118, 2013.

[8] T. Kaku et al, “Detection of intact lava tubes at Marius Hills on the moon by Selene (Kaguya) lunar radar sounder,” Geophysical Research Letters, 2017.

[9] R. Greeley, “Lava tubes and channels in the lunar Marius Hills,” *The Moon*, 1971.

[10] A. G. Taylor and A. Gibbs, *Automated search for lunar lava tubes in the Clementine dataset*, 1998.

[11] Horizon-2020, *European robotic goal-oriented autonomous controller (ergo)*.

[12] C. R. Frost, “Challenges and opportunities for autonomous systems in space,” *Computer Science*, 2011.

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

[14] ESA Advanced Concepts Team, *AI in space workshop*, 2013.

[15] Vitech Corporation, *Genesys: Enhancing systems engineering effectiveness*, 2020.

[16] E. Balaban, S. Narasimhan, M. Daigle, J. Celaya, *et al.*, “A mobile robot testbed for prognostics-enabled autonomous decision making,” in *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2011, PHM 2011*, 2014.

[17] E. Balaban, S. Narasimhan, M. J. Daigle, I. Roychoudhury, *et al.*, “Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms,” *International Journal of Prognostics and Health Management*, vol. 4, 2013.

[18] N. Abhishek, *Reinforcement learning with Open AI, TensorFlow and Keras Using Python*, 9. 2012, vol. 3. DOI: 10.1109/MED.2013.6608833. arXiv: 1603.02199.

[19] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002. DOI: 10.1145/504729.504754.

[20] A. Blum, S. Chawla, D. R. Karger, T. Lane, *et al.*, "Approximation algorithms for orienteering and discounted-reward TSP," *SIAM Journal on Computing*, vol. 37, no. 2, pp. 653–670, 2007. DOI: 10. 1137/050645464.

[21] A. Sakai, D. Ingram, J. Dinius, K. Chawla, *et al.*, *Pythonrobotics: A Python code collection of robotics algorithms*, 2018. arXiv: 1808.10703 [cs.RO].

[22] A. Ellery, *Planetary Rovers*. Springer-Verlag Berlin Heidelberg, 2016.

[23] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson, *et al.*, "Python battery mathematical modelling (pybamm)," *ECSarXiv. February*, vol. 7, 2020.

[24] K. J. Worrall, "Guidance and search algorithms for mobile robots: application and analysis within the context of urban search and rescue," 2010.

[25] L. M. Ireland, "INVERSE SIMULATION AS A TOOL FOR FAULT DETECTION & ISOLATION IN PLANETARY ROVERS," *GNC 2017: 10th International ESA Conference on Guidance, Navigation & Control*, DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011. 1669v3.

[26] J. Marzat, H. Piet-Lahanier, F. Damongeot, and E. Walter, "Model-based fault diagnosis for aerospace systems: A survey," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 226, no. 10, pp. 1329– 1360, 2012. DOI: 10.1177/0954410011421717.

[27] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Springer, Jul. 2014.

[28] R. V. Wagner, E. J. Speyerer, and M. Robinson, "New Mosaicked Data Products from the LROC Team," in *46th Lunar and Planetary Science Conference*, 2015.

[29] X. Xiao and W. L. Whittaker, "Energy Utilization and Energetic Estimation of Achievable Range for Wheeled Mobile Robots Operating on a Single Battery Discharge," no. August 2014.