COUPLED PATH AND MOTION PLANNING FOR A ROVER-MANIPULATOR SYSTEM

J. R. Sánchez-Ibáñez[1], **G. J. Paz-Delgado**[1], **P. Romeo-Manrique**[1], **C. J. Pérez-del-Pulgar**[1], **and M. Azkarate**[2]
[1]*Universidad de Málaga, Andalucía Tech, 29070 Málaga, Spain; {ricardosan,carlosperez}@uma.es*
[2]*European Space Agency, ESTEC, Noordwijk, Netherlands; martin.azkarate@esa.int*

ABSTRACT

This paper introduces a motion planning strategy aimed at the coordination of a rover and manipulator. The main purpose is to fetch samples of scientific interest that could be placed on difficult locations, requiring to maximize the workspace of the combined system. In order to validate this strategy, a simulation environment has been built, based on the VORTEX Studio platform. A virtual model of the ExoTer rover prototype, owned by the European Space Agency, has been used together with the same robot control software. Finally, we show in this paper the benefits of validating the proposed strategy on simulation, prior to its future use on the real experimental rover.

Key words: sampling; fetching; fast marching; planetary exploration.

1. INTRODUCTION

Returning samples from other planets is on the spotlight of space agencies, with the objective of analyzing them on Earth [1]. However, it entails a huge series of difficulties to be faced, such as the proper manipulation of the sample. This operation is non trivial mainly due to the extreme conditions of extraterrestrial surfaces such as those found on Mars [2, 3]. For this reason, it is mandatory the use of unmanned vehicles to carry out this task, particularly rovers. This kind of systems has successfully performed exploration tasks in previous missions, and hence they are considered as well to be used for sampling tasks together with the use of manipulators. Moreover, since future missions are demanding the increase of autonomy on rovers, the use of combined path and motion planning algorithms is also relevant to ensure safety. Previously developed algorithms for terrestrial mobile manipulators can be extended taking into consideration the requirements for planetary exploration. Several approaches have been considered in the past, like the use of sequential algorithms [4], or making the base and the manipulator simultaneously follow certain smooth paths [5]. Obstacle avoidance has been also considered using two approaches: to directly plan the path of the end effector, using methods such as fuzzy logic [6] or RRT (*Rapidly exploring Random Trees*) [7], or to compute a path for the base and then make the one corresponding to the end effector dependant on the previous one [8].

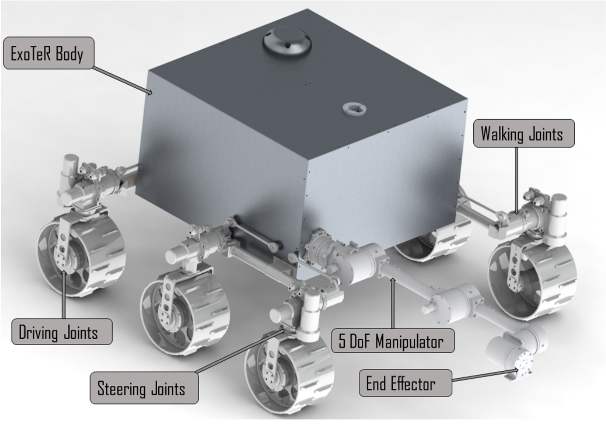


Figure 1: 3D model of ExoTeR carrying a 5 DoF manipulator

This paper presents the functioning of an algorithm that combines path and motion planning within a simulation environment. This algorithm is aimed at mobile manipulators that must retrieve samples from difficult terrains. To build up the simulation environment, virtual models of the rover and an uneven terrain are created and detailed in section 2. The control software used and its connection to the simulation platform are introduced in section 3. Later on, section 4 contains a description about the functioning of the motion planning. The simulation cases where this algorithm is tested are described in section 5.

2. SIMULATION SCENE

The proposed planning algorithm is meant to be implemented on a rover platform named ExoTeR (*Exomars Testing Rover*) [9]. It is a prototype whose kinematic configuration is similar to the one used by the rover that will be sent to Mars as part of the ESA-ExoMars campaign [10]. In this particular case, it is equipped with a manipulator as depicted in Figure 1. Nevertheless, prior to making tests using the real platform, it is desirable to first

Figure 2: Overview of the experimental terrain 3d model

refine the code via simulation. By using the same control software in both situations, simulated and real, the workflow is more agile and straightforward. Besides, the real platform is not compromised and there is not need for preparing a real setup, which results on saving money and time. Hence two virtual models have been prepared to run the simulations: the first one is based on ExoTeR, including its manipulator on board, while the second one takes the form of a terrain that resembles the surface of another planet such as Mars.

2.1. Coupled manipulator-rover system

ExoTeR is a vehicle capable of performing several driving maneuvers thanks to its kinematic configuration. It can even reconfigure its mode of locomotion thanks to its so called *walking joints* [9], although this feature is not under the scope of this paper. Only *driving* and *steering joints* are here relevant. They are used to make the rover advance and turn, making curved trajectories with variable radii. In case the radius is relatively small, the rover executes a *Spot Turn* maneuver, which consists on changing the orientation of the vehicle using zero linear speed, i.e. just rotating without translating. Otherwise, it drives via an *Ackermann steering* maneuver, which makes use of different values of speed on the wheels, as well as different steering angles according to the location of the center of rotation.

With regard to the manipulator, it is attached to ExoTeR at its front and has five motors, each of them corresponding to a Degree of Freedom. The first three of them are used to determine the position of the end effector, while the other two have more influence on its orientation. Although the number of DoFs introduces a restriction in the orientations the manipulator is capable to reach, it can be compensated by the mobility introduced by the rover locomotion subsystem. Besides, the manipulator is robust, in the sense each of its motors contains a huge gearbox to make it lift heavier weights. However, this introduces a drawback in the form of slowness: the maximum speed of each manipulator joint is 0.01 rad/s. This fact is acknowledged by the motion planner so as to prepare the manipulator earlier whenever a sample must be collected.

2.2. Planetary surface

The scene used for the simulation of planetary operations is based on a portion of the experimental terrain located at the University of Málaga[1] (Spain), which is depicted in Figure 2. It shows as an uneven terrain suitable for carrying out planetary navigation tests, presenting diverse soils, hills, rocks and slopes. By means of photogravimetry using the Pix4d software[2], a virtual model of this terrain is built based on aerial images that were taken thanks to a camera drone. Besides, a Digital Elevation Map was obtained as output of this software, having a resolution of 3 cm. This is considered to be high enough to determine the location of even small obstacles as well as other morphological properties such as slopes.

3. IMPLEMENTATION

An schematic describing the elements that make up the simulation environment and the connections between them is shown in Figure 3. The platform in charge of simulating the behaviour of the vehicle and visualizing the scene elements is VORTEX Studio[3]. This software has integrated a physics engine that lets model terrains making use of terramechanic parameters like static and dynamic friction, stiffness, damping or slippage [11]. Besides, VORTEX allows the communication with external software to control any virtual robotic systems present in a scene. The simulation environment we show in this paper makes use of this advantage in two ways. On one hand, information is obtained from the virtual scene describing the pose of the robot as well as the state of its joints, i.e. the real speed and torque they are experiencing. This is useful to later estimate the power consumption by the vehicle while driving. On the other hand, commands to each of the motors can be sent in the same way as with the real system, producing motion in each of the active joints.

The control software used to implement all navigation layers of the rover is built within the Rock[4] (*Robot Construction Kit*) framework. The same layers used for ExoTeR, from path planning to motor commanding, are here connected to VORTEX by means of a bridge programmed using UDP protocol. This bridge is formed by the Rock component that interfaces with the rest of the control software and the UDP module inside VORTEX studio. Once a simulation starts, VORTEX receives from Rock the position and velocity commands corresponding to each active joint, and simulates the motion of the rover on the virtual terrain. It also sends back to Rock information about the position and orientation of the rover body

---

[1] DMS Coordinates: 36°42'57.2"N 4°29'20.5"W

[2] `https://www.pix4d.com/`

[3] `https://www.cm-labs.com/vortex-studio/`

[4] `https://www.rock-robotics.org/`

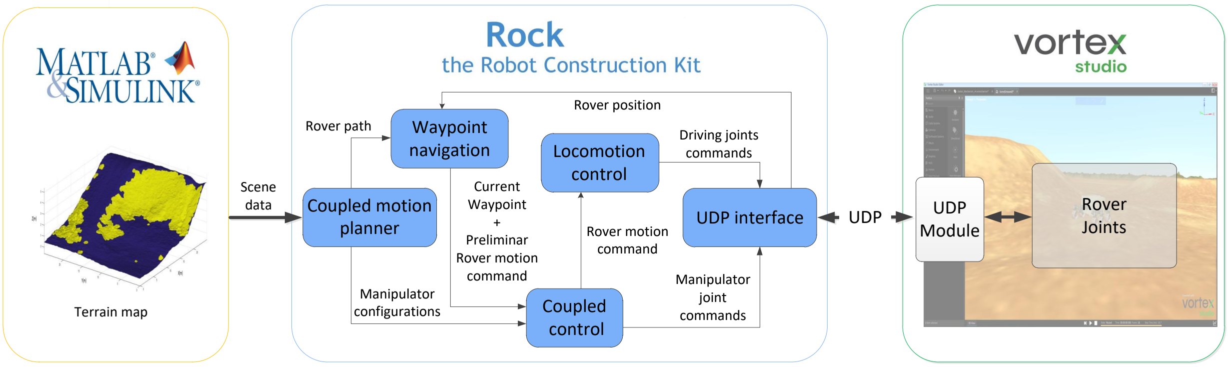


Figure 3: Schematic of the proposed simulation environment

and joints. In this way, Rock is exclusively in charge of the motion control in real time. Each of the Rock components where the control layers are implemented, indicated as well in the schematic of Figure 3, are introduced as follows:

- *Coupled motion planner*. The algorithm proposed, and detailed in Section 4, is implemented in this component. It makes use of information describing the area surrounding the rover, as well as its initial position, the location of the sample to fetch, and indications of the existing obstacles, their placement and size. All this information is provided by MATLAB scripts that process the DEM information of the experimental terrain. As result, this component provides the path and the manipulator configuration that must be acknowledged by the navigation system to reach the sample.

- *Waypoint navigation*. This component computes translational and rotational speeds of the rover required to follow the path computed. This parameters are computed according to the position of the vehicle, provided by the VORTEX platform, and using the algorithms developed in [12]. For this particular case, the computed values are preliminar and still must be processed by the *Coupled Control* component.

- *Coupled Control*. The resulting configurations of the manipulator computed by the *Coupled motion planner* serve as reference for the manipulator joints to reach them. The maximum velocities and position limits are taken into account during this process. Besides, the speed commands from *Waypoint Navigation* for the driving joints may be altered in this part to coordinate the rover and manipulator motion.

- *Locomotion control*. According to the input rover motion commands, speed and position commands are created for each of the driving and steering joints respectively.

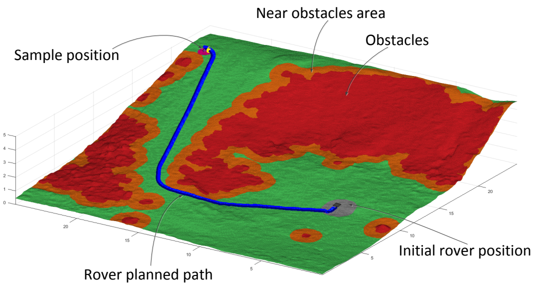
- *UDP interface*. With the commands for the driving joints and the manipulator joints received from *Locomotion control* and *Coupled control* respectively, this component packs the information in a suitable way for UDP communication protocol and sends it to VORTEX studio. This task also receives the current position and orientation of the rover from VORTEX, and sends it back to the *Waypoint navigation* component.

4. COUPLED MOTION PLANNING

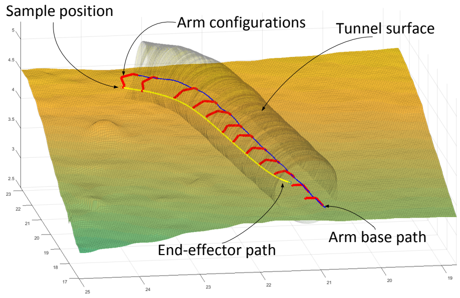
The proposed algorithm is a solution for the problem of a mobile manipulator that has to reach a sample. This must be done in a safe way, meaning the vehicle has to avoid any encountered obstacles in the form of rocks, dunes or craters. For this purpose, both manipulator and rover base should move in a coordinated way, ensuring none of them are harmed during the process. Assuming the rover is located a few meters from the sample, the objective is to deploy the manipulator and place its end effector in contact with the sample, while at the same time the base is approaching to it. Both base and manipulator coordination is a challenging task when there are redundant degrees of freedom (DoF). Mainly, there are three DoF for the motion of the base (two for the position and one for the orientation) and five more for the motion of the manipulator, so there would be infinite configurations to reach a sample.

Figure 4a represents the path planning step of the proposed algorithm, showing a planned path from an initial rover position to a sample placed near obstacles. This step is almost identical to the path planning algorithm developed in [13]: it makes use of the Fast Marching Method to compute a smooth, optimal and continuous path. For this particular case, it has been modified to ensure the rover is able to reach samples that may be located inside hardly accessible areas. Moreover, in order to improve the safety of this operation, repulsive potential fields are created around obstacles. This has effect on the resulting path in two ways: first, the computed path gets further from obstacles. Second, in case the sample is at the border of an obstacle, the path will enter perpendicularly to this edge. This is translated into avoiding high turning maneuvers when the rover is close to the sample, inside an area called *Sampling Area*. Within it, the rover stops at a certain point of the path. This is decided upon all feasible sampling positions in the section of the path entering the *Sampling Area*. The final sampling position is deliberated based on manipulability and reachability criteria evaluation. Moreover, sampling positions that are non-secure are cast aside. In the case all candidate sampling positions are unsafe, the sample is declared as non-reachable.

The second step is the motion planning of the manipulator as depicted in Figure 4b. This process is subdivided into two stages: the early deployment of the manipulator and the planning of the end effector path. The first stage consists on moving the joints of the manipulator from its resting position to another in which the manipulator is deployed and ready to later take the sample. During this stage it is ensured the rover will continue advancing without colliding with any surrounding obstacle. The second stage is focused on the space reachable by the manipulator while the rover is following its respective path. Another path is created by means of the 3d version of the Fast Marching Method, executed in this case on the 3d space delimited by a tunnel around the rover path, as shown in Figure 4b. This 3d path is the one that must be followed by the end effector of the manipulator while the rover is driving. Finally, in order to synchronize both systems, rover and manipulator, while following both paths, it is computed the manipulator configuration to be reached at each moment. This is done by computing the inverse kinematic model. In case the manipulator cannot reach some of the planned configurations due to constraints such as joints limitations or collisions with the rover itself due to its design, the 3d planned path is slightly modified as consequence, resulting on new viable configurations.



(a) Step 1: Rover path planning



(b) Step 2: Manipulator motion planning

Figure 4: Depiction of the two steps that make up the proposed coupled motion planning algorithm

5. EXPERIMENTS

In order to validate the proposed coupled motion planning algorithm before testing it on the real platform, a series of simulations were carried out. Their main purpose was to analyze the performance of the algorithm under different circumstances, i.e. to check whether the coupled system could safely reach a series of samples located on diverse extreme cases. For this reason, five cases were set up as seen in Figure 5. During each simulation execution, the rover always started from the same initial position, having as well the same orientation and being placed on a horizontal plain surface. Then, prior to starting any movement the motion planning algorithm was executed. It was conditioned by the obstacles defined on the map, which were determined under the basis of slope and roughness thresholds. After producing the paths required to guide the rover to the sample, the rover immediately started driving. At the same time, the manipulator joints actuated according to the planning algorithm guidance to make the end effector keep track of the path assigned to it. In this way, the rover deployed the manipulator while it kept advancing towards the sample location, finishing all motion when the end effector was placed on top of the sample. Footage showing the first and fourth cases can be found on a YouTube video[5].

With regard to the simulation cases, in the first two of them (Case 1 and 2) the samples were located on the opposite sides of a mound, as can be checked in Figure 5. There were other elements near them in the form of rocks and slopes that, since they could harm either the rover or the manipulator, were considered as obstacles. In case 1, the rover traversed a narrow corridor placed on *Risky Area* (the area close to obstacles) before having to reorient itself and finally head to the sample. Its final manipulator configuration is depicted in Figure 6a. Unlike in that case, in the second one the rover was able to follow a more open path prior to reaching the sample, since it went through more *Safe Area*. Later on, the last three cases, from 3 to 5, consisted on taking samples located close to the edge of a cliff, where the rover had to be extremely careful in order not to fall down and collapse. Thus, the approach to the sample was done perpendicularly to this edge. This is observed in case 3: in the last stretch of the path, close to the slope, the rover took a sharp curve and headed to the sample, taking it with a final configuration shown in Figure 6c. The fourth case is very similar to

---

[5] `https://youtu.be/aqm--CTUp-I`

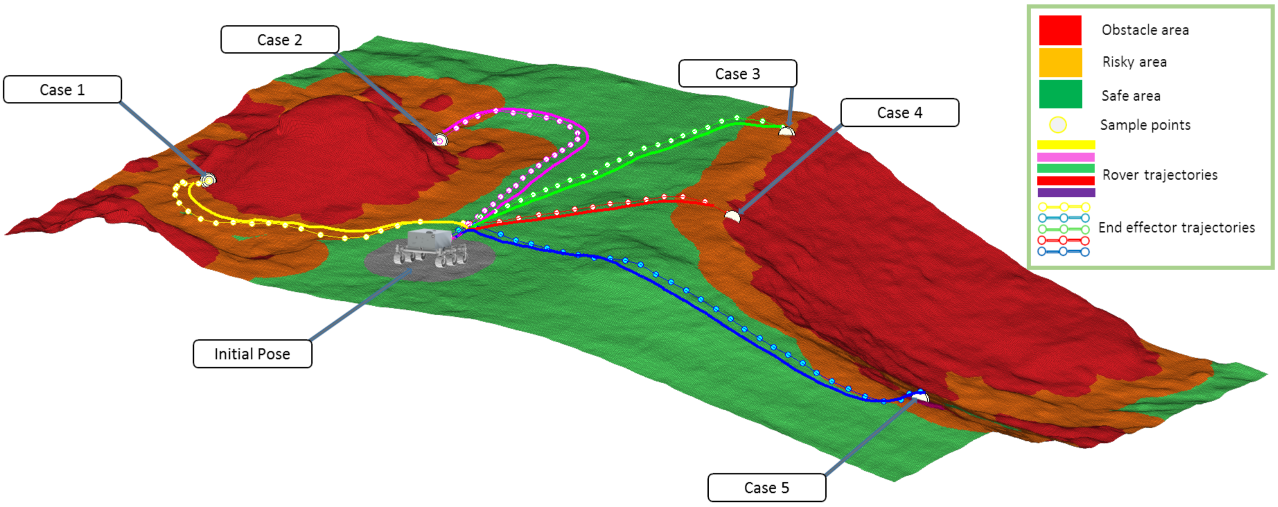
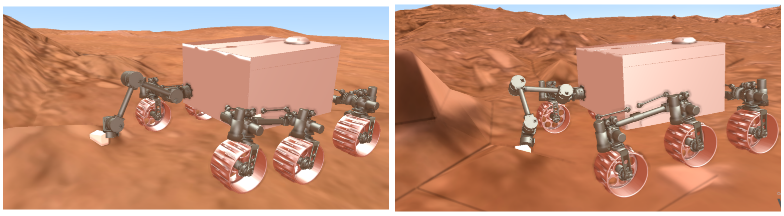
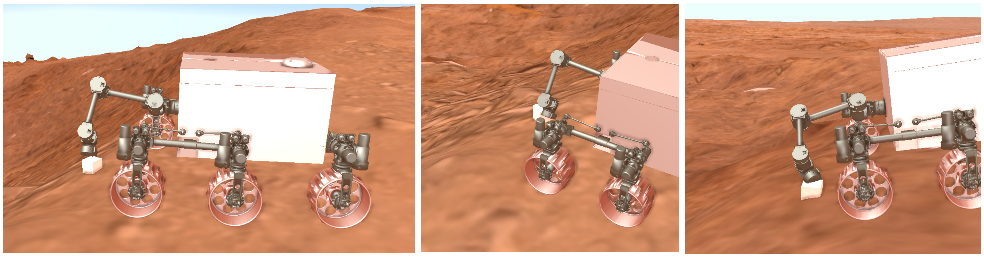


Figure 5: Overview of all sampling cases used for the simulations, including the paths created for the rover base and the end effector.



(a) Case 1. (b) Case 2.



(c) Case 3. (d) Case 4. (e) Case 5.

Figure 6: Close-up screen-shots of the virtual rover taking a sample during the simulation cases.

the previous one, excepting that the sample position was critically inside the slope, resulting in a extreme configuration of the manipulator depicted in Figure 6d. Finally, the last case included a new difficulty as the rover was on a smooth slope while sampling, as shown in Figure 6e.

6. CONCLUSIONS

This paper presents the verification and validation of a motion planning algorithm aimed at retrieving samples in planetary exploration missions. This algorithm produces the paths and joint references that must be followed by a rover and a robotic manipulator attached to it. It is intended for future work to carry out tests using this algorithm on the real experimental platform ExoTer. For this reason, a simulation environment has been built to test and debug such implementation. Within it, the algorithm is programmed using the same software that controls the ExoTeR rover. In this way, the same implementation refined during simulations can be later used in the real platform in a straightforward way. The simulation platform chosen to model the rover and the terrain, VORTEX studio, presents great potential not only for this particular application but also for testing future navigation software, thanks to its capability to handle big scenes and the physics engine it integrates.

With regard to the algorithm here tested on simulations, it has demonstrated to effectively guide the combined system to take samples even when these are placed on uneven terrains. This is translated into augmenting the number of samples that can be retrieved, i.e. to increase the scientific return of the mission. As stated before, is foreseen as future work to carry out experiments using this algorithm on the real rover-manipulator system. Furthermore, this algorithm will be also employed within the EU-H2020 project *Autonomous DEcision making in very long traverses*[6] (ADE), serving as a planning module used by a rover to catch any sample of interest that may be found on its way. It is expected this algorithm will be refined and improved, considering more terrain parameters (not just morphology but composition as well) and maximizing the performance of alternate locomotion modes to augment the reachability of the rover manipulator.

ACKNOWLEDGMENTS

This work was partially supported by the European Space Agency under contract number 4000118072/16/NL/LvH/gp - 2.

REFERENCES

[1] Andrea Merlo, Jonan Larranaga, and Peter Falkner. Sample fetching rover (sfr) for msr. In *12th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, pages 1–10. ESA-ESTEC, 2013.

[2] Marcus Y Woo. Roving on mars. *Engineering and Science*, 72(2):12–20, 2009.

[3] Ajey Lele. Mars missions: Past, present and future. In *Mission Mars: India's Quest for the Red Planet*, pages 85–92. Springer, 2014.

[4] Vinay Pilania and Kamal Gupta. Mobile manipulator planning under uncertainty in unknown environments. *The International Journal of Robotics Research*, 37(2-3):316–339, 2018.

[5] MH Korayem, R Abbasi Esfeden, and SR Nekoo. Path planning algorithm in wheeled mobile manipulators based on motion of arms. *Journal of Mechanical Science and Technology*, 29(4):1753–1763, 2015.

[6] Rekha Raja and Ashish Dutta. Motion planning of a mobile manipulator using fuzzy controller to dexterity measures. In *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, pages 1–6. IEEE, 2015.

[7] Giuseppe Oriolo and Christian Mongillo. Motion planning for mobile manipulators along given end-effector paths. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2154–2160. IEEE, 2005.

[8] Grzegorz Pajak and Iwona Pajak. Point-to-point collision-free trajectory planning for mobile manipulators. *Journal of Intelligent & Robotic Systems*, 85(3-4):523–538, 2017.

[9] Martin Azkarate, Martin Zwick, Javier Hidalgo-Carrio, Robin Nelen, Tim Wiese, Pantelis Poulakis, Luc Joudrier, and Gianfranco Visentin. First Experimental investigations on Wheel-Walking for improving Triple-Bogie rover locomotion performances. In *13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, pages 1–6. ESA-ESTEC, 2015.

[10] J Vago, O Witasse, H Svedhem, P Baglioni, A Haldemann, G Gianfiglio, T Blancquaert, D McCoy, and R de Groot. ESA ExoMars program: the next step in exploring Mars. *Solar System Research*, 49(7):518–528, 2015.

[11] Ali Azimi, Daniel Holz, Jozsef Kövecses, Jorge Angeles, and Marek Teichmann. A multibody dynamics framework for simulation of rovers on soft terrain. *Journal of Computational and Nonlinear Dynamics*, 10(3):031004, 2015.

[12] Jan Filip, Martin Azkarate, and Gianfranco Visentin. Trajectory control for autonomous planetary rovers. In *12th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, pages 1–7. ESA-ESTEC, 2013.

[13] J.R. Sanchez, C.J. Perez-del Pulgar, and Martin Azkarate. Path planning for reconfigurable rovers in planetary exploration. In *14th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, pages 1–7. ESA-ESTEC, 2017.

---

[6] `https://h2020-ade.gmv.com/`