

# PATH PLANNING FOR RECONFIGURABLE ROVERS IN PLANETARY EXPLORATION

J. R. Sánchez<sup>1</sup>, C. J. Pérez-del-Pulgar<sup>2</sup>, and M. Azkarate<sup>3</sup>

<sup>1</sup>Universidad de Málaga, Andalucía Tech, 29070 Málaga, Spain, Email: ricardosan@uma.es

<sup>2</sup>Universidad de Málaga, Andalucía Tech, 29070 Málaga, Spain, Email: carlosperez@uma.es

<sup>3</sup>European Space Agency, ESTEC, Noordwijk, Netherlands, Email: Martin.Azkarate@esa.int

## ABSTRACT

This paper introduces a path planning algorithm that takes into consideration different locomotion modes in a wheeled reconfigurable rover. Such algorithm, based on Fast Marching, calculates the optimal path in terms of power consumption between two positions, providing the most appropriate locomotion mode to be used at each position. Finally, the path planning algorithm is validated on a virtual Martian scene created within the V-REP simulation platform, where a virtual model of a planetary rover prototype is controlled by the same software that is used on the real one. Results of this contribution also demonstrate how the use of two locomotion modes, wheel-walking and normal-driving, can reduce the power consumption for a particular area.

Key words: path planning; wheel-walking; fast marching; modeling.

## 1. INTRODUCTION

Exploring the surface of other planets is one of the main interests of space agencies [1]. The use of rovers in several missions has contributed to solve the problem of carrying scientific tools to extract data without being necessary to send humans into remote space or have an operator online. Nowadays, the European Space Agency (ESA), through the ExoMars mission, intends to place a rover on the Martian surface by 2020, with the aim of discovering any signs of life in the red planet. Previous missions experimented difficulties while traversing certain types of terrain, even some of them ended being trapped in loose soil [2]. To solve this issue, the ExoMars rover presents a kinematic configuration capable of executing an alternative mode of locomotion: *wheel-walking*. Such mode has been experimentally demonstrated [3] to improve traction on certain types of terrain where otherwise the rover would be stuck, as well as recent research [4] provides an analysis of its performance according to soil parameters to clarify in which situations is more efficient than other locomotion modes.

The use of more than one locomotion mode expands the

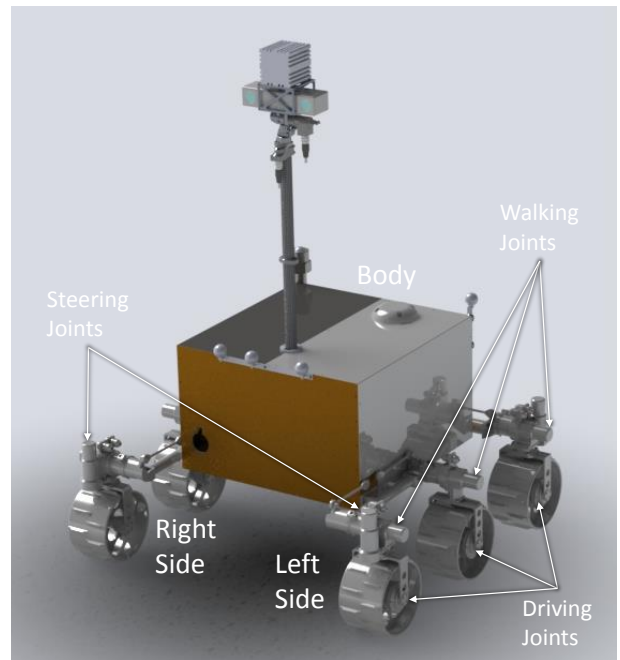
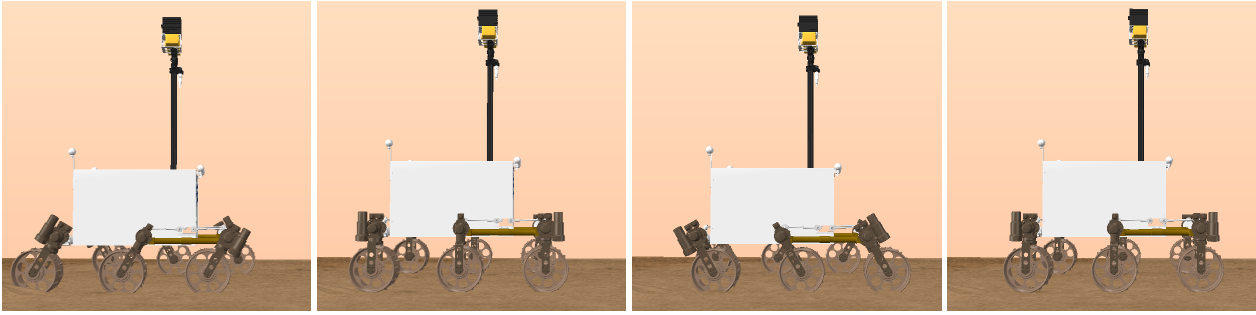


Figure 1: ExoTeR with indications of some of its active joints

possibilities of the rover regarding to finding an optimal path. A method proposed by E. Rohmer et al. [5] employs two algorithms: one to find the optimal trajectory using Dijkstra method and other to decide which locomotion mode use at each stage of the obtained path. However, since the performance of each locomotion mode is not considered during the execution of the first algorithm, the resulting trajectory is not optimized in terms of power consumption.

Prior to carrying out field tests to check the performance of this kind of algorithms, simulation software with virtual scenes can be used. An example is *ARTEMIS*, a tool designed by Trease *et al* [6] to analyze the dynamic contact between rover wheels and terrain in *Mars Exploration Rover* having terramechanics into account. This paper presents all the details related to a platform created with the purpose of simulating the execution of path planning algorithms using reconfigurable rovers to vali-



(a) Starting wheel-walking mode (b) Left legs forward, right legs backwards (c) Left legs backwards, right legs forward (d) Finishing wheel-walking mode

Figure 2: Rover starts wheel-walking movement (a). First the legs from one side (left or right) move forwards while the rest move backwards(b). Now the other side moves backwards and the first one forward (c). These steps are continuously repeated until rover stops or changes mode(d)

date them. First, the ExoTeR, an ExoMars-like lab rover prototype owned by ESA, is modeled in the V-REP simulation platform. Furthermore, a virtual scene resembling a Martian surface is implemented. Later on, the same framework implemented in the real ExoTeR, i.e. Rock<sup>1</sup>, is used to communicate with V-REP. Thus, a complete simulation environment is created and serves as a tool that lets us check the performance of different algorithms before making tests in a real hardware. A demonstration is presented in this paper using a path planner based on the Fast Marching method, modified to take into account the available locomotion modes in ExoTeR: *normal driving* and *wheel-walking*.

## 2. SIMULATION SCENE

To properly build a simulation environment, the virtual elements contained within it must resemble as much as possible the real ones in terms of appearance and physical behavior. In particular, to validate the path planning algorithm proposed in this paper two elements are modeled: a reconfigurable rover, ExoTeR, and a real sandbox with different types of rocks and soils resembling the Martian surface.

### 2.1. ExoTeR Rover

A virtual model of ExoTeR is used to simulate its behavior in custom made environments. Such model contains all the physical properties related to inertia and masses of each of its components. In this way, the stress due to weight on both passive and active joints is properly distributed, which is important when calculating the friction forces acting on wheels as will be detailed later. The ExoTeR rover presents a kinematic configuration of 6x4x6 degrees of freedom as depicted in Fig. 1. It is made up of:

- 6 Driving Joints: they are in charge of making the wheels roll. They are commanded only in speed.
- 4 Steering Joints: they are placed in the front and rear legs and are in charge of setting the direction the wheels are facing. They are commanded only in position.
- 6 Walking Joints: they are placed on top of each leg with the purpose of deploy them when necessary. They can be commanded in both speed and position.

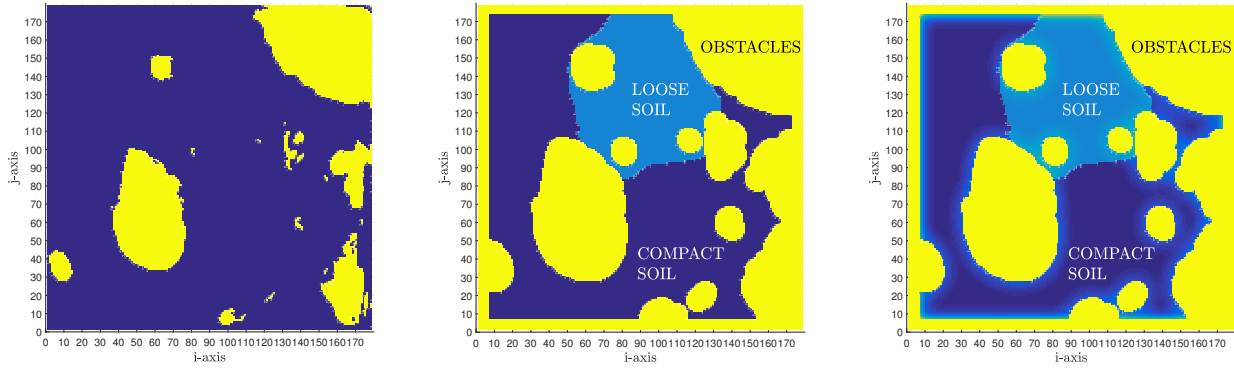
On one hand, *Normal driving* mode uses of all driving and steering joints while keeps the walking ones locked in a constant angle. Depending on the motion command the rover receives, translational or rotational speed, there can be distinguished two basic maneuvers: *Ackerman Turn* and *Spot Turn*. The *Ackerman Turn* is the one that lets the rover advance with a nonzero translational speed while rotating along a radius determined by the rotational speed. In case the rotational speed command is zero, the rover advances straight, which is equivalent to say it rotates with an infinite radius. ExoTeR makes a *Spot Turn* when only the rotational speed is nonzero, meaning it will rotate around a central point without advancing, which is useful when only the change of direction is desired.

On the other hand, *Wheel-walking* mode maintains the steering joints locked to zero, while using the driving and walking joints to perform strides with the legs that improve wheels traction. There are several gaits, i.e. patterns, of *wheel-walking* depending on how the legs are used. In this case, the gait *Side-By-Side* is the one chosen. This gait is introduced in [3] and the way it works is explained step by step in Fig. 2.

### 2.2. Martian Surface

To recreate a Martian environment, 3d point cloud data has been extracted from the Martian terrain in the Planetary Robotics Laboratory at ESA. It is used to determine

<sup>1</sup><http://rock-robotics.org/>



(a) High slopes and the presence of rocks are considered as obstacles, marked in yellow, while the blue area is at first traversable by the rover  
 (b) The area close to the obstacles under a certain distance is also considered as obstacles, marked in yellow, while the blue area is at first traversable by the rover  
 (c) A value of risk indicating proximity to obstacles is added to the cost of the traversable zones to make the rover pass as far as possible from them

Figure 3: Process to obtain a *Cost Map* grid based on the existence of obstacles in the form of high slopes and rocks, different types of terrain and the involved risk of being close to an obstacle area

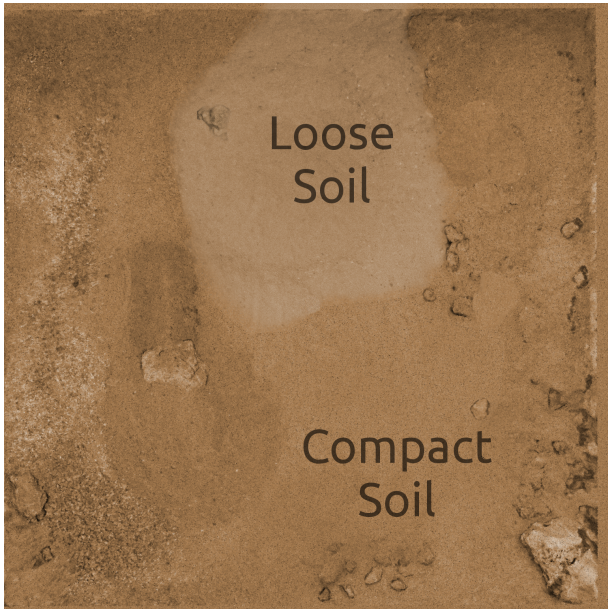


Figure 4: Texture applied in the virtual scene showing two types of soil: loose and compact

on a square grid called *Cost Map* which nodes (intersections between grid edges) are obstacles. In this case, obstacles can be both dangerous elements, like rocks, and high slopes. Each node contains as well information about the type of soil and the risk to encounter with any obstacle. Fig.3 shows the process to obtain the final *Cost Map* for the Martian terrain. It is worth mentioning that the phase of making the obstacle zones grow, as seen in Fig.3b, is to avoid collisions due to the occupancy zone of the rover respect to the grid size. Such growth is equal to the sum of half the width of the rover and the maximum distance this vehicle can deviate from the trajectory.

Two other square grids, containing elevation data, have been created from the Martian surface point cloud. One of them, with a high resolution grid size, is in charge of the graphical visualization of the terrain shape and appearance. A texture is applied on it, based on an orthonormal image of the real terrain. This texture has been modified in order to clearly represent two areas with different terrain types: one with loose soil and other with compact one. In this way, two types of terrain with different friction and slip ratio values are set as can be seen in Fig. 4. Each one of them is optimal for each locomotion mode so as to clearly check the performance of the algorithm. The other square grid is the one used for calculating the dynamic contact with the wheels, having a lower resolution in order to reduce computation load.

### 3. IMPLEMENTATION

The components of the simulation environment and the way they are interconnected are shown in Fig. 5. Matlab is used to create the *Cost Map* as well as the square grids in charge of visualization and dynamic interaction with the rover. Furthermore, since there is no much research about wheel-soil interaction for the *wheel-walking* mode comparing to *normal drive* [7], a function is built from several simulations performed in [4] to decide which locomotion mode is better for a certain terrain depending on wheel-soil interaction parameters. This function is used by the path planner in its cost function and works as follows: it returns the value of the power required to traverse a certain terrain depending on its soil parameters and the locomotion mode employed. The modeling of the interaction between wheels and terrain is simplified by means of two soil parameters [8]: the dynamic friction, which is basically the resistance to the roll of the wheels, and

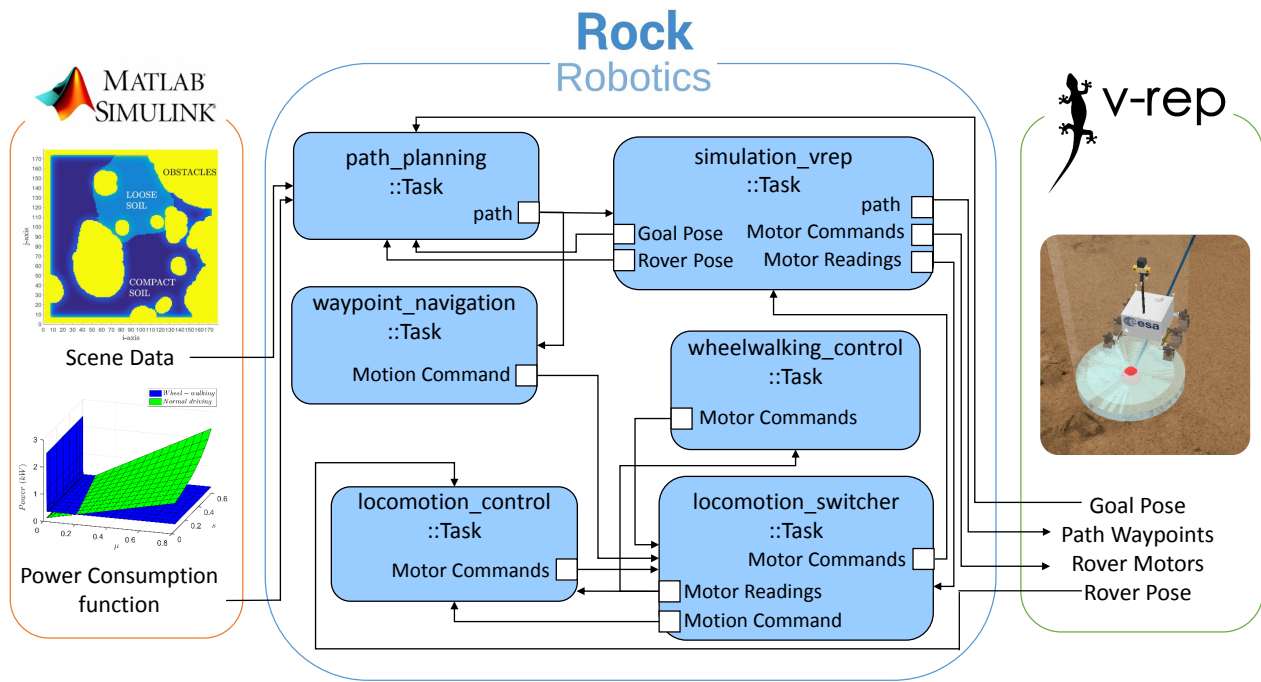


Figure 5: Simulation Environment Architecture

the slip ratio, calculated from the ideal and real speed of the rover. In this work, such parameters are given to the rover beforehand. However, a classifier, as proposed by Brooks *et al.* [9], could be based on a vision system to estimate the soil parameters of a terrain before traversing it.

On the other hand, the framework ROCK is used to interconnect the different layers of control of the rover, going from high (path planning) to low level (motors commanding). All code related to the developed ROCK components and their respective libraries can be found in the ExoTeR GitHub repository<sup>2</sup>. These layers of control, whose associated tasks can be seen in the scheme of Fig. 5, are detailed as follows:

- **Path\_Planning:** it makes use of a custom path planning library where the previously explained Fast Marching method is implemented, returning as result the optimal path as a vector of waypoints.
- **Waypoint\_Navigation:** it employs a pure-pursuit based path follower to command the rover to smoothly follow the trajectory provided by Path.Planning with a deviation margin under certain limits.
- **Locomotion\_Control:** it creates commands for every active joint to execute driving maneuvers, like *Ackerman* or *Spot Turns*, depending on the values of the desired rotational and translational velocity that receives.

<sup>2</sup><https://github.com/exoter-rover>

- **Wheelwalking\_Control:** it is in charge of making the rover reproduce the Wheel-walking gait based on readings from both active and passive joints.
- **Locomotion\_Switcher:** it acts as an arbiter that makes safer the switching between modes of locomotion and ensures that the walking joints are locked when going with *normal driving* mode and that steering joints are set to zero when changing to *wheel-walking* mode.
- **Simulation\_Vrep:** it is in charge of the communications between the ROCK framework and the virtual elements, like the ExoTeR model, in the V-REP simulation platform.

Finally, a bridge is created between Rock and V-REP. The scene in V-REP makes use of the path information to graphically represent the obtained trajectory and sends the motor commands, received from Rock, to the virtual rover. Also, it returns to the framework the measurements of each active and passive joint in the rover, as well as the position and orientation of the final goal that is previously set by the user. Inside V-REP, the physics engine VORTEX is used to model the dynamics related to the contact between the rover and terrain. Its main advantage is the use of real physic parameter values instead of invented ones, as the rest of V-REP physical engines do. VORTEX lets the user handle the friction forces acting when two bodies collide, depending on the direction they are moving. Having Fig. 6 as reference, when two shapes collide (the cylinder primitive shape of one wheel and the shape of the terrain in this case) they create a contact point and a contact plane. Such plane is formed by



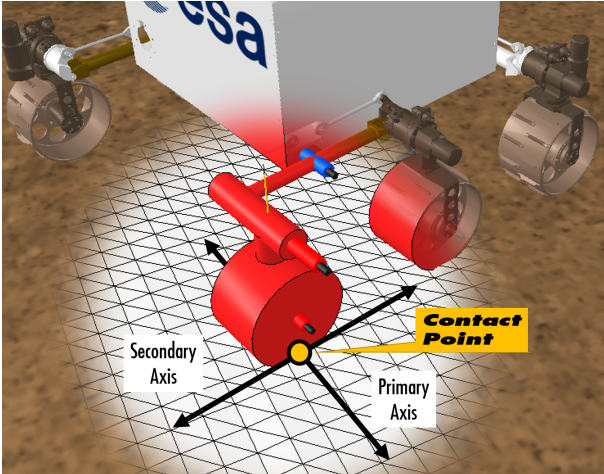


Figure 6: VORTEX Primary and Secondary Friction Axis during contact between wheel and soil

two axis, called primary axis and secondary axis. The user can set the dynamic properties of the friction forces acting on each axis. In our case, the primary axis is parallel to the driving joint axis, as seen in Fig. 6. It is because otherwise, anomalies appears during the simulation, such as undesired forces acting on the wheel laterals. The VORTEX friction model employed is the one called *Scaled Box*. It is based on *Coulomb's friction* and, unlike *Scaled Box Fast*, it doesn't re-use the value of the normal force applied on wheels in certain simulation steps values when doing calculations for optimization, but ensures more stability during simulations. The forces acting on both the primary and secondary axis are affected by the parameter *Friction Coefficient*, which indicates the value of the dynamic friction coefficient that enters into action when the wheel is moving respect to the terrain. Also, the static friction is present when the wheel is going to start a movement, being indirectly set by the *Static Friction Scale*, which is a parameter representing the ratio between the static and the dynamic friction coefficients. Other VORTEX parameters that have influence on the wheel-soil interaction are *Slip*, not as commonly known in mobile robotics (ratio between actual velocity and desired velocity), but an acting viscous force instead, and *Slide*, which introduces a backwards velocity to the rover.

#### 4. PATH PLANNING ALGORITHM

The path planning algorithm to be tested in this work finds the optimal path in terms of power consumption, which is a critical factor during an exploration mission. The capacity to reconfigure the locomotion mode expands the possibilities for the rover to find optimal paths. Otherwise, with just one mode the available options for the rover are more limited mainly due to existing terrain conditions. In other words, a reconfigurable rover equipped with a proper path planner could be more flexible to find optimal trajectories than rovers with just one

locomotion mode.

The path planning algorithm proposed in this work is based on the Fast Marching method, which has been used in recent researches for underwater [10] and mobile [11] robotic applications. It has a similar computational effort as Dijkstra alike algorithms and consist on calculating the amount of cost required to arrive at each point of the map by using a type of differential equation named Eikonal. The Eikonal equation is employed to numerically calculate the propagation of a wave that starts from a certain point and expands through the rest of nodes at a rate that depends on each of them. In our case, the parameter that propagates is the amount of energy the rover requires to arrive at each point, having as result after executing the algorithm a potential field indicating such parameter for each node.

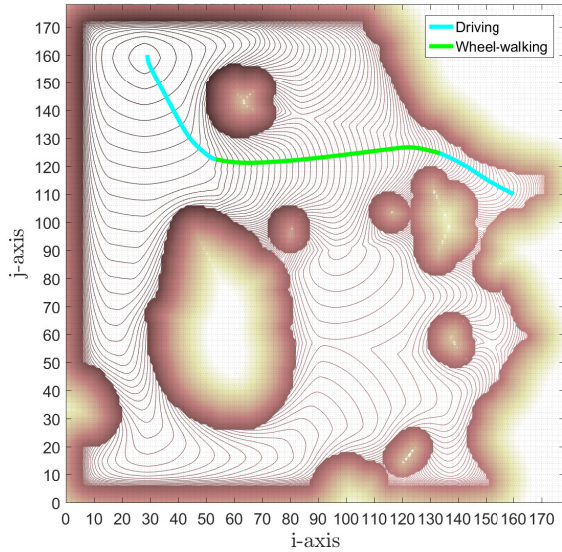
The rate of propagation in a node is proportional to the power required to traverse it. In Fig.7 can be seen the propagation of a wave on the *Cost Map* taken from Fig.3c in two cases that will be described later. The lines are composed by points with the same required amount of energy to arrive at each of them. The distance from one line to the next one generally increases in an inversely proportional way to the power required to traverse the terrain where they are. On the compact soil the wave propagates further than on the loose soil, and since the obstacles are considered a very high cost the rate of propagation of the wave in those areas is quite low. Moreover, as the power required to traverse a terrain is always a positive value the required amount of energy increases as the wave propagates. Therefore, there are never local minimum points in the resulting stationary potential field.

By applying a Gradient Descent Method on the obtained field a smooth and continuous path is obtained. This trajectory is not restricted to grid limitations such as the angle the rover turns when going from one waypoint to another. The way to determine the locomotion mode to employ at each waypoint is by interpolating with the preferred mode of each of its neighboring nodes.

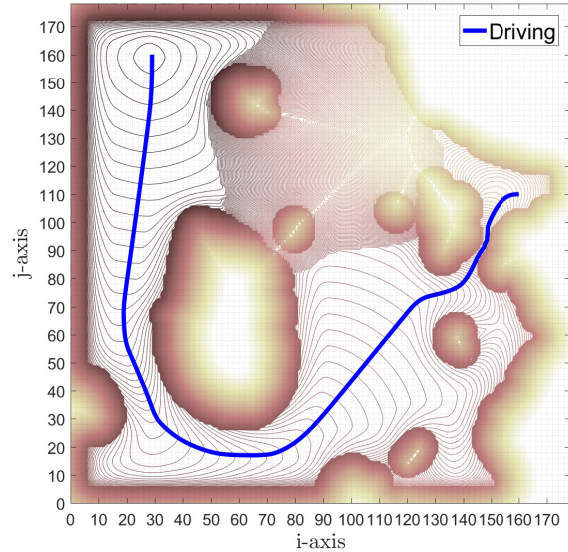
#### 5. EXPERIMENTS

Two experiments were carried out to test the capacities of the path planning algorithm presented in this paper. Each one consisted on making the rover perform the same task: go from the top left corner of the map to a certain zone surrounded by rocks placed on the right. The only difference between both experiments was just that in one of them the rover could use also the wheel-walking mode but not in the other.

After executing the simulation environment for both cases, the resulting trajectories are shown in Fig.8. When the rover has available the *wheel-walking* mode the path planner is able to find a totally different trajectory passing through the loose soil than when it has only *normal driving* mode available. The reason can be deduced by



(a) Case using both ExoTeR locomotion modes



(b) Case using only driving mode

Figure 7: Resulting potential fields and trajectories for two cases

observing the resulting potential fields for each case in Fig.7. When the rover employs both locomotion modes it is able to advance through the loose soil to arrive at the desired destination since the power required to use *wheel-walking* is not much higher compared to the power required to use *normal driving* on the compact soil. In the case where the rover only uses *normal driving* mode, due to being the cost of traversing that same zone too high it needs to perform a much larger and costly travel because of the existence of a large obstacle area in the middle of the scene. As expected, when propagating the Fast Marching wave it becomes much slower in the loose soil when wheel-walking is not available, and due to this the lines with equal cost are much more close to each other. It is worth mentioning that thanks to the use of a parameter apart from cost representing the risk to be close to an obstacle, as explained before, both trajectories tend to pass without approximating to nearby obstacles unless it is strictly necessary. The areas near the obstacles act as repelling potential fields.

## 6. CONCLUSIONS

In this paper all the features of a simulation environment to test path planning algorithms have been described, as well as it has been used to validate a Fast Marching based algorithm that takes into account two locomotion modes.

First, the physics related to the interaction between the rover elements and the surface of the scene have been modeled, taking into account involved friction forces, masses and inertias. VORTEX, an multibody dynamics engine contained in V-REP has been used for this pur-

pose.

Second, the complete software architecture of the simulation environment has been fully detailed. The grids containing elevation and cost data as well as information related to the pose of the initial and destination nodes, needed by the path planner, have been introduced. Also, the ROCK tasks in charge of controlling the rover movements have been presented as well as their interconnections and the way they communicate with the V-REP platform.

Finally, the path planning algorithm proposed to be used with reconfigurable rovers has been tested on the simulation environment. The results indicate that the availability of alternate locomotion modes as well as the use of a proper path planner like the one proposed in this paper lets get even more optimal paths in scenes with different types of soils. In this way, the optimization of the energetic resources of the rover is improved.

As future work it is intended to consider more parameters like slope when extracting power consumption for each locomotion mode. In this way, the rover could find the optimal path considering more than one locomotion mode taking into account both composition and shape of the terrain.

## REFERENCES

- [1] Ajey Lele. Mars missions: Past, present and future. In *Mission Mars*, pages 85–92. Springer, 2014.
- [2] Marcus Y Woo. Roving on mars. *Engineering and Science*, 72(2):12–20, 2009.

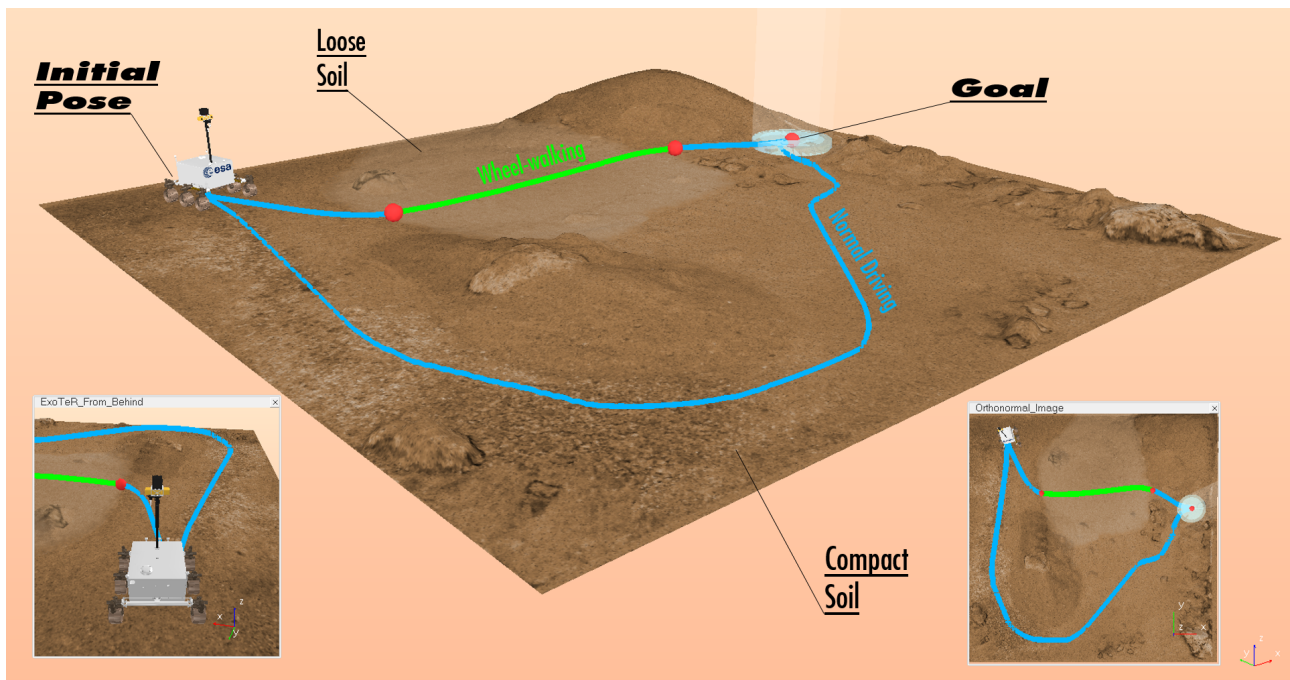


Figure 8: Screenshot of the virtual scene in V-REP showing the trajectories obtained as result from the two cases

- [3] 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*, pages 1–6. ESA, 2015.
- [4] C.J. Pérez-Del-Pulgar, J.R. Sánchez, A.J. Sánchez, M. Azkarate, and G. Visentin. Path planning for reconfigurable rovers in planetary exploration. In *AIM 2017 IEEE International Conference on Advanced Intelligent Mechatronics*, pages 1–6. IEEE Robotics and Automation Society, 2017.
- [5] Eric Rohmer, Giulio Reina, and Kazuya Yoshida. Dynamic simulation-based action planner for a reconfigurable hybrid leg-wheel planetary exploration rover. *Advanced Robotics*, 24(8-9):1219–1238, 2010.
- [6] Brian Trease, Raymond Arvidson, Randel Lindemann, Keith Bennett, Feng Zhou, Karl Iagnemma, Carmine Senatore, and Lauren Van Dyke. Dynamic modeling and soil mechanics for path planning of the mars exploration rovers. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 755–765. American Society of Mechanical Engineers, 2011.
- [7] Masataku Sutoh, Junya Yusa, Tsuyoshi Ito, Keiji Nagatani, and Kazuya Yoshida. Traveling performance evaluation of planetary rovers on loose soil. *Journal of Field Robotics*, 29(4):648–662, 2012.
- [8] Carlos Canudas-de Wit, Panagiotis Tsiotras, Efsthios Velenis, Michel Basset, and Gerard Gissinger. Dynamic friction models for road/tire longitudinal interaction. *Vehicle System Dynamics*, 39(3):189–226, 2003.
- [9] Christopher A Brooks and Karl Iagnemma. Self-supervised terrain classification for planetary surface exploration rovers. *Journal of Field Robotics*, 29(3):445–468, 2012.
- [10] Clement Petres, Yan Pailhas, Yvan Petillot, and Dave Lane. Underwater path planning using fast marching algorithms. In *Oceans 2005-Europe*, volume 2, pages 814–819. IEEE, 2005.
- [11] Santiago Garrido, María Malfaz, and Dolores Blanco. Application of the fast marching method for outdoor motion planning in robotics. *Robotics and Autonomous Systems*, 61(2):106–114, 2013.