DETAILED DESCRIPTION OF THE HIGH-LEVEL AUTONOMY FUNCTIONALITIES DEVELOPED FOR THE EXOMARS ROVER

Matthias Winter [1], Sergio Rubio [1], Richard Lancaster [1], Chris Barclay [1], Nuno Silva [1], Ben Nye [1], Leonardo Bora [1]

[1] Airbus Defence & Space Ltd., Gunnels Wood Road – Stevenage – SG1 2AS – United Kingdom,

Email: matthias.m.winter@airbus.com, sergio.s.rubio@airbus.com, richard.lancaster@airbus.com, chris.barclay@airbus.com, nuno.silva@airbus.com, ben.nye@airbus.com, leonardo.bora@airbus.com

ABSTRACT

The 2020 part of the ESA ExoMars mission will land a rover on the Martian surface with the aim of establishing if life ever existed on Mars. Additionally the mission will be validating a number of technologies for planetary exploration. To maximise the number and variety of sites visited by the mission, the rover will be fitted with a Mobility subsystem designed to allow it to safely traverse terrain without real-time assistance from Earth. ExoMars is an international cooperation between ESA and Roscosmos with contribution from NASA. Airbus is responsible for the development of the ExoMars Rover Vehicle. Thales Alenia Space (Italy) is the industrial prime.

This paper describes the design and capabilities of the high-level autonomy functionalities that have been developed in the scope of the ExoMars program to allow its rover to autonomously plan its way across previously unknown hazardous Martian terrain. It also presents preliminary test results from both the Airbus Mars Yard and a numerical simulation environment. Note that these functionalities are currently not part of the ExoMars Rover baseline, though a re-establishment of these functionalities into the baseline is currently being evaluated.

1. GNC ARCHITECTURE OVERVIEW

To meet its science objectives, the ExoMars Rover has to traverse long distances through previously unknown hazardous Martian terrain with very limited communication opportunities. To achieve this, the rover needs a highly autonomous mobility system. This mobility system has been designed to have different levels of autonomy. In brief, the ground operator can:

1. Command the rover to reach a target: the mobility system would then autonomously analyse terrain and decide which path to follow to reach the target. This is referred to as 'full autonomy'.
2. Command the rover to follow a path specified by ground: the mobility system would drive the rover following the commanded path while compensating for external disturbances that could push the rover away from the path. If the autonomy algorithms described in this paper are implemented, the rover will also be able to check this path for hazards.
3. Directly drive the rover, for example commanding it to drive straight, to turn on the spot or to stop.

To achieve these functionalities the Mobility subsystem architecture in Figure 1 has been designed (see [1] and [2] for more details). The blocks implementing the functionalities for approach 2 and 3 are presented in detail in [3]; the additional blocks needed to implement approach 1 are presented here.

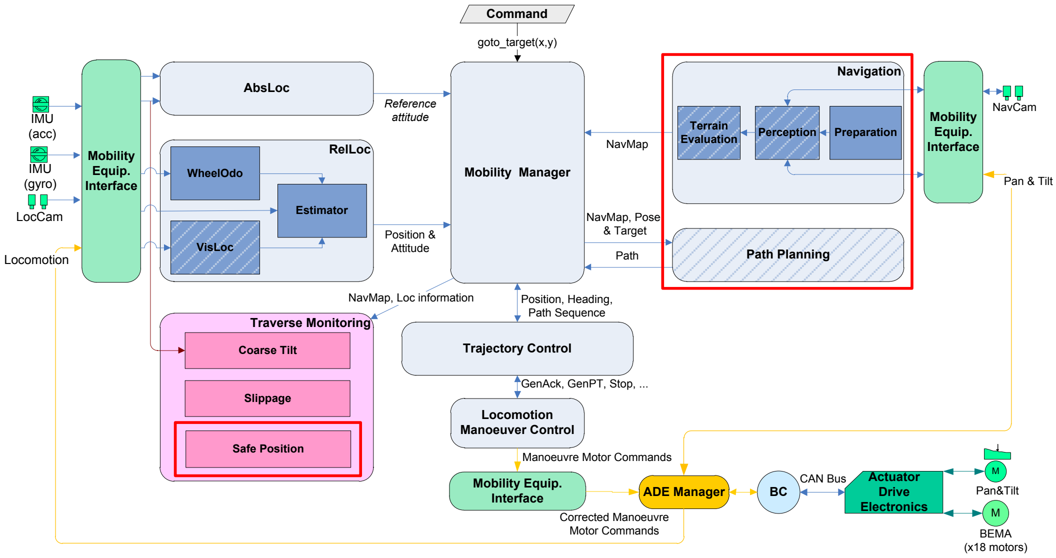


Figure 1. Mobility Subsystem Functional Architecture (full-autonomy functionalities marked in red)

2. HARDWARE

The baseline design of the rover is shown in Figure 2.

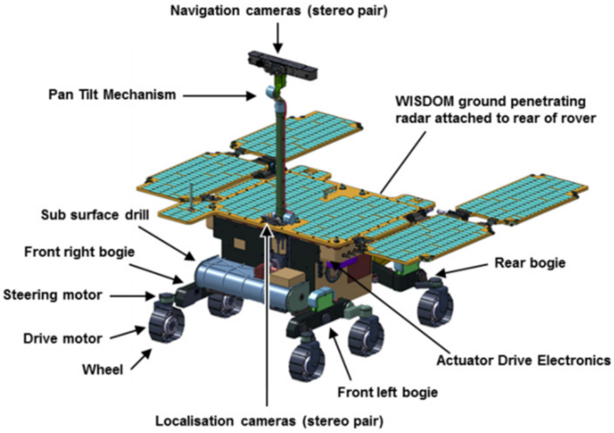


Figure 2. Rover Baseline Design

The hardware needed for the execution of planned paths is presented in [3]. The additional hardware utilised by the autonomy algorithms are:

- Navigation Camera (stereo pair); designed and implemented by Neptec Design Group. Optical characteristics summarised in Table 1. Note that the camera outputs distortion corrected images.
- Deployable Mast Assembly (DMA), including the Pan and Tilt mechanism which reorients the navigation camera; designed and implemented by RUAG Space.

Parameter	Value
Field of View	65 deg
Stereo Baseline	150 mm
Image Resolution	1024 x 1024
Exposure Time	1 ms to 1000 ms
F-Number	f/8
Focal Length	4.0 mm

Table 1. Characteristics of the Navigation Camera

3. AUTONOMY FUNCTIONALITY OVERVIEW

A summary of the key Mobility subsystem performance targets is provided in Table 2.

Parameter	Value	
Distance driven per day	70 m	(full autonomy)
Range accuracy (from target)	7 m	(after 70 m traverse)
Heading accuracy	15°	(after 70 m traverse)
Heading knowledge	5°	(after 70 m traverse)

Table 2. Key Mobility Performance Targets

The ExoMars Autonomy architecture works with a stop & go approach: Every 2.3 metres the rover stops (from here on referred to as navigation stop) to perform the following steps (shown in more detail in Figure 3):

1. Take three stereo images of the surrounding terrain.
2. Model the surrounding terrain.
3. Create a local map of the surrounding terrain specifying safe and unsafe areas as well as the terrain difficulty.
4. Merge the data of the local map into a persistent map which contains the data gathered at previous navigation stops.
5. Plan a 2.3 metres long safe path towards the target.
6. Output a traverse monitoring map to the Traverse Monitoring module.

Then the rover executes the 2.3 metres long path with a closed-loop control system as described in [3].

Note that the stop & go approach is not a fundamental limitation of the ExoMars architecture (see Section 11) but is a conscious design decision taking into account power, thermal and OBC constraints.

4. KEY DESIGN DRIVERS FOR THE ALGORITHMS

The following key design drivers are important to understand the design choices made during the development of the autonomy algorithms:

- **Safety:** The high-level autonomy algorithms have to keep the rover safe at all times. Particularly for collision avoidance (e.g. bottoming out of the rover body on a rock or hitting the solar panels against a rock), the ExoMars Rover FDIR is not able to predict the problem before it has occurred. These can be mission ending events. Therefore, the autonomy algorithms have to be designed to always work conservatively so that they prevent these events from ever happening.
- **Maximise traversable terrain:** Martian terrain can be very challenging. To allow the rover to traverse smoothly towards its target, a sufficiently large proportion of the terrain has to be classified as safe. Therefore, the system is not allowed to be too conservative in its approach. This (together with the need for safety) leads to a need for accurate uncertainty estimation and propagation instead of applying global margins.
- **Repeatability:** If the same Martian terrain is seen from different distances or perspectives, its evaluation has to be sufficiently similar. Particularly the binary decision if something is an obstacle or not has to be repeatable. Otherwise terrain might act as a one-way valve. This is particularly challenging for borderline traversable terrain.
- **Execution time:** The algorithms have to execute on the ExoMars Rover 96 MHz LEON2 co-processor. The execution time goal is two minutes (Note that this requires further optimisation of the terrain analysis algorithms). This is particularly challenging taking into account the amount of data processed (> 1.5 million pixels) and the need for accurate uncertainty estimation and propagation.

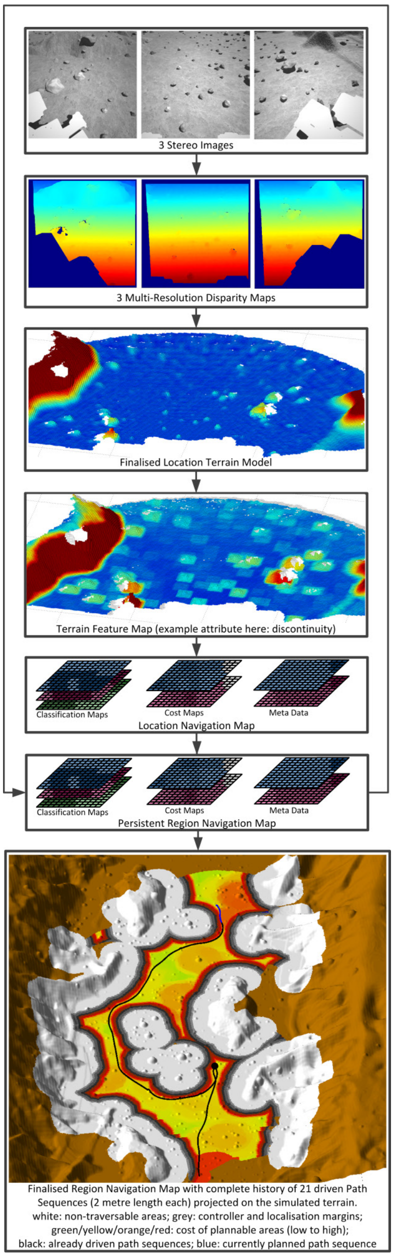


Figure 3. Autonomy Processing Overview

5. PERCEPTION

The perception algorithms have previously been presented in a dedicated paper [4]. Therefore, only a high-level summary is given here.

The perception algorithms create a disparity map from each of the three stereo image pairs acquired at one navigation stop (see Figure 3). A disparity map describes the apparent shift in corresponding pixels between the left and right picture of a stereo image. Pixels corresponding to objects close to the cameras will exhibit a larger disparity than pixels corresponding to objects farther away. For each disparity map pixel, the magnitude of the disparity may be used to transform, through triangulation, the 2D pixel location into a 3D point in space.

Key challenges while designing these algorithms were:

- Accuracy: The disparity map has to be accurate enough to adequately represent Martian terrain such that the terrain analysis algorithms can determine its safety and traversability.
- Execution speed: Each call of the perception system has to take less than 20 seconds to run on the ExoMars Rover 96 MHz LEON2 co-processor.

Figure 4 shows the architecture of the Perception System. Notably a multi-resolution approach is used to maximise the amount of terrain covered by the disparity maps whilst mitigating the adverse processing time implications of using high resolution images.

The perception algorithms have already been extensively tested in the ExoMars Rover GNC simulator, in the Airbus Mars Yard, with images from NASA rovers and on a LEON2 processor [4].

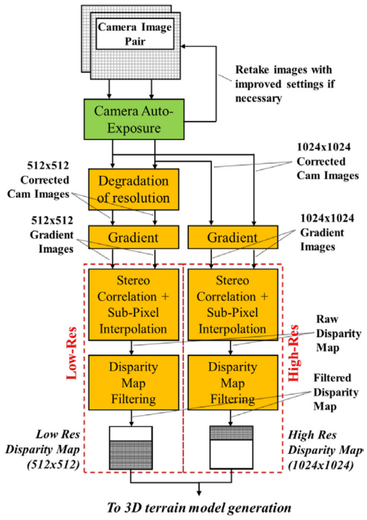


Figure 4. Perception System Architecture Overview

6. TERRAIN MODELLING

At each navigation stop, the terrain modelling algorithms convert the three disparity maps created by the perception algorithms into a single terrain model (called finalised location terrain model, see Figure 3). Additionally, the algorithms create the extended terrain model, which is later on used to check the clearance of the solar panels.

Key challenges while designing these algorithms were:

- Accuracy: The resulting terrain models have to be always conservative but at the same time they are not allowed to be overly conservative because that would reduce the amount of terrain which is evaluated to be traversable.
- Execution speed: The terrain modelling algorithms should take less than 35 seconds per navigation stop to execute on the ExoMars Rover 96 MHz LEON2 co-processor. This is particularly challenging because of the large amount of data (point cloud with more than 1.5 million points).

Utilising triangulation, the geometric properties of the navigation camera are used to convert each pixel of a multi-resolution disparity map (low and high resolution part as shown in Figure 4) into a 3D point saved in a single point cloud. Then these points are binned into a 2D Cartesian grid (see Figure 5). Some filtering is applied to remove unreliable data (e.g. outlier filtering). Afterwards an uncertainty analysis is performed using statistical methods taking into account the expected accuracy of the stereo correlation process, the amount of data available in each grid cell and the geometric properties of the navigation camera.

Two DEM based terrain models are then derived from the point cloud, one containing the data from the low-resolution part of the disparity map and one containing the data from the high-resolution part of the disparity map. This approach is needed to avoid anomalies in the calculation of the mean for grid cells whose data is partially contained in the high-resolution part and partially in the low-resolution part of the disparity map.

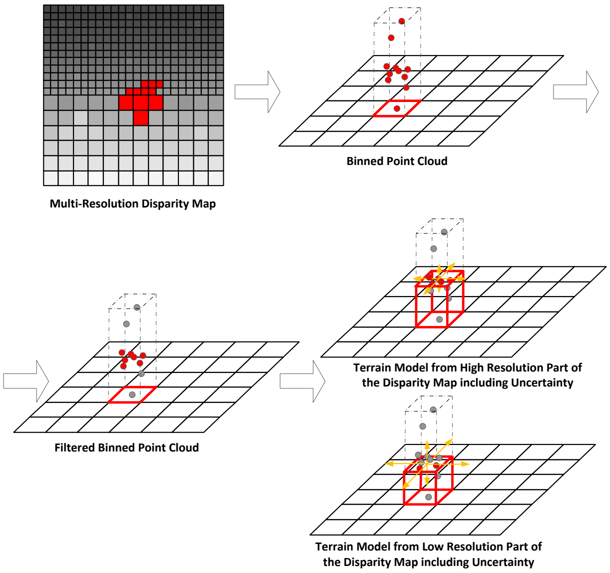


Figure 5. Terrain Model Generation

Afterwards the resulting two terrain models are merged into a single perception terrain model and the three perception terrain models from the three disparity maps are merged into a single location terrain model.

As a final step the uncertainty analysis in the location terrain model is finalised, which includes additional filtering and the creation of three final DEMs:

- Estimated mean elevation of each cell
- Estimated minimum elevation of each cell
- Estimated maximum elevation of each cell

The terrain modelling process also creates the extended terrain model. This terrain model is used during the terrain analysis (see Section 7) to check the clearance of the solar panels. For this check a larger terrain model is needed than can be created through the process used for the finalised location terrain model because the solar panels reach out a considerable amount from the centre of the rover (up to 2.1 metres, see Figure 2).

Note that the finalised location terrain model has to be very accurate and does not have a defined worst-case (i.e. the highest and lowest possible elevation can be the worst-case for different safety analyses performed later), but the extended terrain model can be less accurate as long as it is conservative with regard to its worst-case: the highest possible elevation. Therefore the following steps are taken to increase the area contained in the extended terrain model:

- Less strong filtering.
- Holes in the terrain model are closed with worst-case data (considering the fact that the area behind the hole must have been visible to the camera).
- Some data of the extended terrain model created at the previous navigation stop can be merged into the current extended terrain model. Note that normally this is not an option because of the considerable uncertainty in the rover pose after a 2.3 metre path has been driven. However, in this case a worst-case envelope can be used.

The algorithms have been tested in the ExoMars Rover GNC simulator, in the Airbus Mars Yard, on data from NASA rovers and on a LEON2 processor.

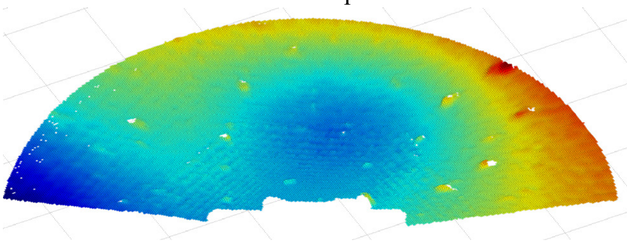


Figure 6. Mean DEM of a Simulation on the Autonomy Test Terrain

7. TERRAIN ANALYSIS

The terrain model is then analysed with respect to the capability of the rover to safely traverse it. In this process, the values of several attributes are estimated. These attributes allow a conclusive evaluation of the known terrain with respect to rover safety and the level of difficulty experienced when traversing it. Some of these attributes are fully defined by the terrain (e.g. terrain discontinuity and slope); others take into account the rover's locomotion system (e.g. rover clearance, solar panel clearance, rover tilt angle and bogie angles). The results are summarised in an ensemble of 2D maps (one per attribute) with a Cartesian grid (see Figure 7).

Of particular concern is the question how the rover would be positioned on the terrain because bottoming out of the rover body on a rock or hitting the solar panels against terrain could potentially be mission ending events. To estimate the attributes related to the rover state (e.g. clearance, solar panel clearance, bogie angles, rover tilt angle), a virtual rover model (RSM: Rover Simplified Model) is placed on each cell of the terrain model and is rotated to cover all possible headings. To any cell, the worst-case attribute values experienced for any heading on that cell are assigned. This is done to make the terrain attributes non-directional, therefore avoiding traversing terrain which is only traversable in one direction and could act as a one-way valve.

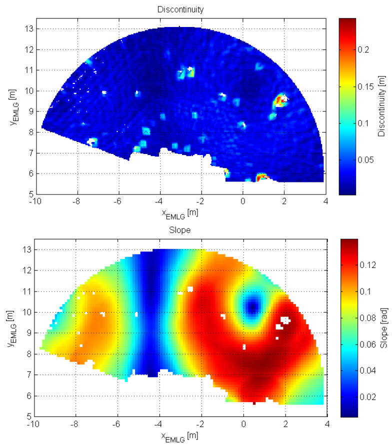


Figure 7. Two Attributes of a Terrain Feature Map

The terrain model the RSM is placed on is not a single DEM, but consists of three DEMs (mean, minimum and maximum elevation) to take into account uncertainties. The placement of the RSM has to take all three of these into account. This is accomplished by not only placing the RSM on the nominal (i.e. mean) DEM but also in all the different combinations of wheels on maximum/minimum elevation that can lead to a worsening of one of the attributes.

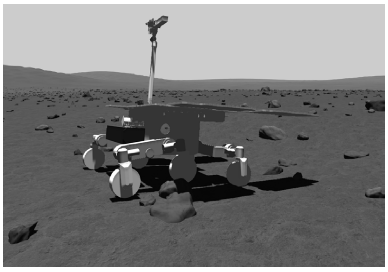


Figure 8. Nominal Placement of RSM

Key challenges while designing these algorithms were:

- Propagation of uncertainties: The terrain model uncertainties had to be propagated efficiently through the estimation process of the terrain attributes.
- Processing resources: The algorithms have to run on limited processing resources (SPARC LEON2 96 MHz processor) and limited memory (< 512 MB). This is particularly challenging because of the large number of rover placements (i.e. on each terrain model cell with each heading).

The algorithms have been tested in the ExoMars Rover GNC simulator, the Airbus Mars Yard and on a LEON2 processor. Figure 8 shows an example of the RSM placement visualised in PANGU.

8. MAPPING

Taking into account the capabilities of the locomotion system, the location navigation map is derived from the terrain feature map (see Figure 3). This map is a 2D map with Cartesian grid and specifies traversable, non-traversable and unknown areas. Moreover cost values are defined for traversable areas based on how difficult and risky traversing them will be. To define the different areas and the cost, traversability tables are used. Traversability tables specify for the value of a single terrain attribute (e.g. clearance) or a combination of terrain attributes (e.g. slope and discontinuity) if the rover can safely traverse it and – if so – a cost value. These tables are provided to the autonomy algorithms and are tuned according to tests performed with the locomotion system of the ExoMars rover.

Next, the local map containing the information gathered at the current navigation stop is merged into the persistent region navigation map, which contains the data gathered at previous navigation stops. From this persistent map the following outputs are generated:

- **Finalised region navigation map:** A map specifying areas the path to the next navigation stop is allowed to be planned through, areas only the long-term path is allowed to be planned through and areas no path is allowed to be planned through. It also contains cost values for the plannable areas.
- **Escape boundary:** A map specifying all possible end points for the long-term path and their cost.
- **Traverse monitoring map:** A map specifying areas the rover position estimate is allowed to enter and areas it is not allowed to enter. This is used by the Traverse Monitoring module to check that the rover stays in safe terrain.

All outputs take into account the expected performance of the Relative Localisation module and the Trajectory Control module [3]. For example (see Figure 9), an area of non-traversable or unknown terrain in the persistent region navigation map will lead to a slightly larger area of terrain the rover position estimate is not allowed to enter in the traverse monitoring map. This is due to the uncertainty in the position estimate when driving the next path. The same area will lead to a yet larger area no path is allowed to be planned through in the finalised region navigation map. This is the case because – due to disturbances – the controller will not be able to permanently keep the rover position estimate perfectly on the planned path.

After driving the path generated by the path planning algorithms, the uncertainty in the estimate of the rover's position and attitude has increased. This leads to an increase of the position uncertainty of previously mapped areas relative to the rover. Therefore, at the next navigation stop, the mapping algorithms update the persistent region navigation map to take the additional uncertainty into account.

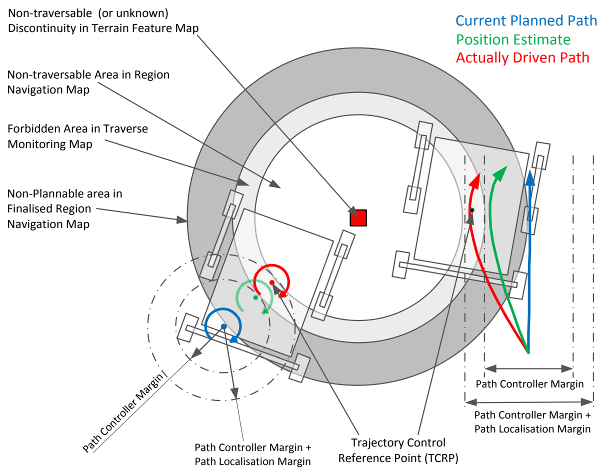
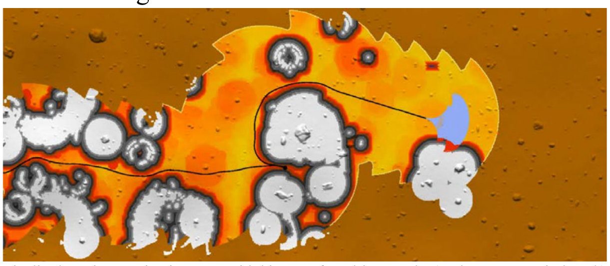


Figure 9. Example of the Margins around an Obstacle (simplified, not up to scale)

Key challenges during the development of these algorithms were:

- Limit dynamic of maps: The persistent region navigation map changes over time because data is added, data perceived at previous navigation stops becomes increasingly uncertain and areas in some cases get re-classified when re-mapped from a different perspective and/or distance. This poses a big challenge when trying to avoid the autonomy algorithms getting trapped after reaching a dead-end (e.g. because the entry into the dead-end has closed due to uncertainty propagation). Many small mechanisms (e.g. buffers and logic about when to consider specific uncertainties) were implemented to achieve a smooth running in challenging Martian terrain.
- Execution speed: These algorithms contain a considerable number of steps which use traditional images processing functionalities (e.g. dilation, erosion, flood fill) on large maps (millions of cells/pixels). To achieve the target execution time of less than 20 seconds to run on the ExoMars Rover 96 MHz LEON2 co-processor, high-performance C functions had to be implemented to rapidly perform image processing functionalities on binary and grey-scale images.



Finalised Region Navigation Map with history of 14 driven Paths (each 2.3 metres in length) projected on the simulated terrain. White: non-traversable areas; grey: controller and localisation margins; green/yellow/orange/red: cost of plannable areas (low to high); black: already driven paths, blue: currently considered paths.

Figure 10. Finalised Region Navigation Map

These algorithms have been extensively tested in the ExoMars Rover GNC simulator, in the Airbus Mars Yard and on a LEON2 processor. Figure 10 shows the finalised region navigation map created by the algorithms during a traverse in the GNC simulator.

9. PATH PLANNING

At each navigation stop, the path planning algorithms plan the next path to be driven by the Trajectory Control module. The planned path is up to 2.3 metres long and consists of smooth curves and point turn manoeuvres (in which the rover will change its heading on the spot without linear displacement).

Unless the rover is close to the target, the majority of the terrain between the rover and the target is unlikely to have been classified. Therefore, the path planning algorithms have to plan a route across the finalised region navigation map towards the escape boundary (outputs of the mapping process, see Section 8). This route is planned such that:

- Areas of the finalised region navigation map that are unknown or deemed as non-plannable are avoided.
- The costs in the finalised region navigation map cells that the route crosses are minimised.
- The anticipated future cost in yet unknown cells which will have to be traversed after reaching the escape boundary is minimised.
- The path generated from the start of the planned route is compatible with the dynamics and manoeuvrability of the rover.

The possible manoeuvres that make up the start of the route are strictly limited. This ensures dynamics compatibility and allows the route to be planned rapidly on the rover's constrained computational resources.

At the start of the route there can either be no point turn, or a point turn of any 45 degree heading increment. After this, the planned route will consist of three smooth curves. The curvature of these is constrained to being either a straight line or one of four possible curvatures to either side, up to a maximum of 0.7 rad/m (which is half of the rover's maximum curvature capability to ensure actuator freedom for the control system to correct disturbances).

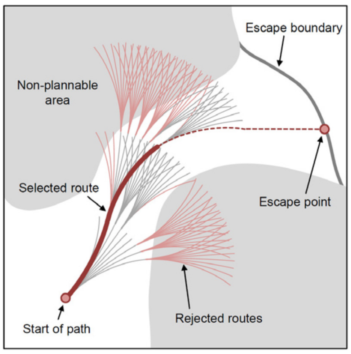


Figure 11. Illustration of the Path Planning Algorithms

The planning is performed by executing an A* search algorithm across a single hybrid search graph. It consists of lattice edges and nodes that represent the point turn and the smooth curves, and rectilinear grid edges and nodes for the section of the search between the end of the smooth curves and the escape boundary. This A* search also takes into account an additional cost for any termination point for the long-term path (selected on the escape boundary) which is not the nearest to the target. This way it considers the additional distance (and therefore cost) the rover will have to drive before reaching the target.

Once the route is planned, the optional point turn and the first two smooth curves are output to the Trajectory Control module as the path to be driven. The third smooth curve and the rectilinear section of the route are not output. They are included in the planning to ensure that the planner makes sensible long term decisions.

10. SYSTEM LEVEL TESTING

The high-level autonomy functionalities have been successfully tested together with the GNC algorithms which execute the path [3] (where possible) in the following test environments:

- **ExoMars Rover GNC Simulator:** Simulator with high-fidelity environment and equipment models developed for the formal functional verification of the ExoMars Rover Vehicle Mobility Software.
- **Airbus Mars Yard in Stevenage** (Figure 12): Indoor test facility with more than 500 square metres of representative Martian terrain. The breadboard rover (MDM) utilises breadboard models of the locomotion subsystem and the navigation camera designed for the ExoMars Rover.
- **Data from existing NASA rovers:** Pictures taken by the MER rovers.
- **LEON2 processor (SPARC):** Pender Electronic Design GR-CPCI-XC4V board with an implementation of a LEON2 processor on it. This environment is used to evaluate the execution speed of the algorithms on a representative processor.

Several hundred 70 metre traverses with randomised starting and target positions have been simulated on the challenging Autonomy Test Terrain (ATT_0001) in the ExoMars Rover GNC simulator with very good results:

Simulation Result	Value
Rover Kept Safe	100 percent
Reached Final Target	90 percent
Cannot find a Way to the Final Target	10 percent

Table 3 Test Results of Test Campaign with randomised Starting and Target Position (~70 m long traverses)

These are very good results, particularly considering that ATT_0001 is an envelope of the worst-case terrain the system is designed to be able to traverse through. Therefore, in easier terrain much higher success rates (near to 100 percent are expected). Moreover it is shown that the conservatism in the system is sufficient, particularly considering that the rover driving is simulated with the maximum amount of wheel sinkage (8 cm) assumed by the algorithms on 100 percent of the sand surface.

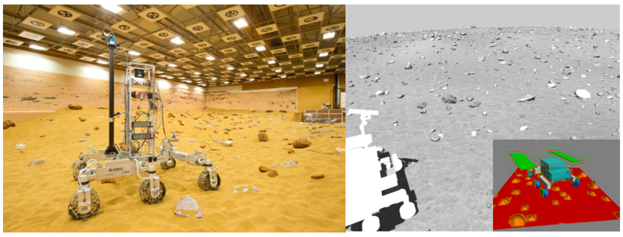


Figure 12. Airbus Mars Yard with MDM Breadboard Rover & Visual and Rover Dynamics Model

11. BEYOND EXOMARS: CONTINUOUS DRIVE

Building on the mature ExoMars design, Airbus has been investing in enhancing it for future missions such as Mars Sample Return (MSR) and science missions which have to travel large distances (on Moon or Mars). The obvious architectural change to allow a rover to cover larger distances is the switch from a stop & go approach to a continuous drive approach. With the constraints on ExoMars (processing resources, thermal and power) and the target distance, the stop & go approach proved to be the most cost efficient solution. However, the ExoMars Rover (with the full autonomy functionality) would not be able to travel sufficient distance for many MSR mission concepts.

Therefore, the architecture update shown in Figure 13 has been designed: Instead of repeating the traditional Sense, Model, Plan & Act (SMPA) phases sequentially, the new architecture switches into a mode in which after the sensing phase, the modelling and planning is performed while the rover is executing the already existing path planned while driving the previous path. Effectively the rover is only stopping to take the three stereo images (therefore avoiding the problem of merging the data from stereo images with high relative uncertainty due to pose uncertainty and mast oscillation). In a streamlined overall GNC architecture this should be achievable in less than 10 seconds, therefore reducing the stopping time to a small percentage of the stopping time of the ExoMars architecture. This could also potentially be more streamlined by using one or several cameras with a wider field of view or LIDAR.

The continuous drive architecture has been implemented and successfully tested in the ExoMars Rover GNC simulator using only existing ExoMars functionalities in a multi-threaded setup and new functionalities in Relative Localisation which predict the additional uncertainties in the rover pose after driving the next path and the uncertainties of previous rover poses relative to the new future pose at the end of the path.

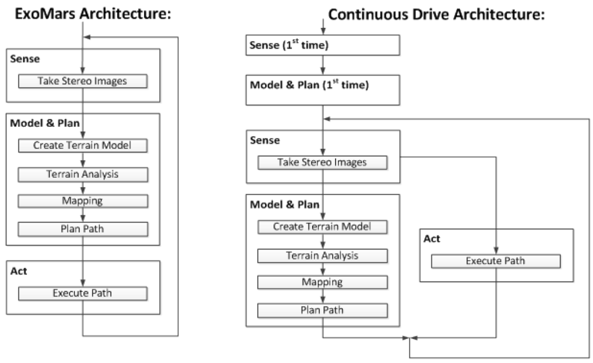


Figure 13. Comparison of ExoMars Architecture with the new Continuous Drive Architecture

Currently this new architecture is being migrated from the simulator to a breadboard rover to be tested in the Airbus Mars Yard.

12. CONCLUSION

This paper presents the design of the high-level autonomy functionalities developed for the ExoMars Rover. The development of these algorithms has now been completed and they have been successfully tested in the high-fidelity ExoMars Rover GNC simulator, the Airbus Mars Yard and with data from NASA rovers. They have also been performance tested on representative hardware (LEON2 CPU). They are now ready to move to the flight software production stage enabling the final end-to-end validation at system level with HITL (hardware in the loop).

Unfortunately these algorithms are currently not part of the ExoMars Rover baseline. The current baseline assumes that the ground operator will define the whole path to be driven during one sol in advance. The following challenges will be faced by the ground team:

- Estimate rock heights and slope angles from up to 30 metres away to evaluate if the rover can safely traverse them without bottoming out or hitting the solar panels (Figure 12 shows rocks up to 20 metres away).
- Estimate the danger of terrain hidden behind rocks and sand dunes.
- Take into account the increasing error in the position and attitude estimate of the rover during a whole day.
- Make sure that the rover safely traverses a sufficient distance per sol whilst meeting the ExoMars science objectives.

These are areas in which the autonomy functionalities presented in this paper would greatly improve the capabilities of the ExoMars Rover. They would:

- **Increase science return:**
  The high-level autonomy functionalities are designed to enable the rover to safely traverse long distances (average speed > 10 m/hr) even in complex terrain (i.e. significant population of rocks, undulating sands etc.). From the experience of driving the MDM in the Airbus Mars Yard, defining paths up to 30 metres in advance is only possible in extremely easy (and therefore often scientifically not very interesting) terrain.
  The full-autonomous system can traverse through more complex terrain because it evaluates its traversability when only a few meters away, enabling it to accurately estimate rock heights and slope angles and look behind all traversable rocks. Naturally also the error in the position and attitude estimate of the rover is considerably smaller if only accumulated for a few metres of driving. Note that scientifically interesting terrain is often particularly challenging terrain.
- **Reduce risk of mission loss:** By the time the autonomy software is loaded onto the rover, it will have been extensively tested in a formal V&V campaign with hundreds of simulations and hardware in the loop testing. The path planned by a ground operator will inherently be much less pre-determined and based on much less data (images only from one single position per day). On top comes the human factor, as recently demonstrated when a breadboard rover operated by an astronaut on the ISS got stuck on a rock in the Airbus Mars Yard [5]. It is important to note that there is no FDIR protection against bottoming out the rover body or hitting the solar panels against a rock. Both are plausible mission ending scenarios, which will have to be addressed one way or another.

13. REFERENCES

1. Winter, M., Barclay, C., Pereira, V., Lancaster, R., Caceres, M., McManamon, K., Nye, B. Silva, N., Lachat, D., Campana, M. (2015). ExoMars Rover Vehicle – Detailed Description of the GNC System, *ASTRA Conference 2015*.
2. Gao, Y. (Editor), Iles, P., Winter, M., Silva, N., Bajpai, A., Muller, J., Kirchner, F. (2016). Contemporary Planetary Robotics: An Approach toward Autonomous Systems – Surface Navigation Chapter, John Wiley & Sons, ISBN: 978-3-527-41325-6.
3. Bora, L., Lancaster, R., Nye, B., Barclay, C., Rubio, S., Winter, M. (2017). ExoMars Rover Control, Localisation and Path Planning in a Hazardous and High Disturbance Environment, *ASTRA Conference 2017*.
4. McManamon, K., Lancaster, R., Silva, N. (2013). ExoMars Rover Vehicle Perception System Architecture and Test Results, *ASTRA Conference 2013*.
5. Taubert, D., Allouis, E. et al. (2017). METERON SUPVIS-M – An Operations Experiment to Prepare for Future Human/Robotic Missions on the Moon and Beyond, *ASTRA Conference 2017*.