

Status of the CNES Autonomous Navigation solution accommodated on the ExoMars rover

Michel Delpech¹, Irene Valenzuela¹

¹Centre National d'Etudes Spatiales (CNES), Toulouse, France
michel.delpech@cnes.fr

Keywords: planetary exploration, rover autonomy, validation

1. Introduction

The objective of ExoMars mission is to land and operate on Mars the Rosalind rover to search for signs of past and recent life. Before the Ukrainian crisis broke out, the rover was to be launched in September 2022 by a Proton rocket and land in February 2023 in the Oxia Planum region. The rover control architecture was designed to offer Autonomous Navigation capabilities that were developed within two distinct solutions covering roughly the same functional scope but with different algorithmic approaches. These two solutions have been delivered by ADS-UK and CNES to be qualified before final integration in the rover flight computer. This mission has been now postponed to 2028 within a new cooperation framework and the additional time is providing opportunities to improve the overall system performance. The paper objective is to describe the current status of the CNES Autonomous Navigation Software and present the foreseen program of activities prior to flight that includes some enhancement of functionalities and a consolidation of the parameter tuning approach.

The first section presents the functional and performance requirements applicable to the autonomous behavior and continues with a description of the specific CNES Autonomous Navigation module (*AutoNav* software). This module implements a set of functionalities that are used in two different operational modes: in *AutoTrav* mode, the ground defines the goal and the *AutoNav* software computes periodically a safe path increment to the goal; in *CheckPath mode*, the path is defined by the ground and the *AutoNav* module must analyze the terrain traversability in the path vicinity and authorize or not the path execution. The main input of the traversability and safety analysis for both modes is a model of the local environment built through stereovision techniques.

The next section provides a detailed description of the tests that have been performed on the Ground Test Model in the ALTEC facilities and provides an analysis of the achieved results with numerous illustrations and figures of the *AutoNav* timeline. This part is followed by a presentation of the lessons learned including some recommendations for possible improvement.

The last section presents the different activities that will take place during the time lapse before flight. Besides the nominal maintenance activities that may include some potential software updates required by the customer, CNES is planning to deliver a new version that includes some improvement of the *CheckPath* mode implementation. The section will describe the principles of the current algorithm design that already offers two different behaviors already tested on GTM: a conservative assessment used also for the *AutoTrav* mode and some optimized assessment technique that allows the rover to move safely in more challenging environments. The foreseen additional work concerns the optimized assessment technique which current limitations will be removed.

2. Description of the Autonomous Navigation

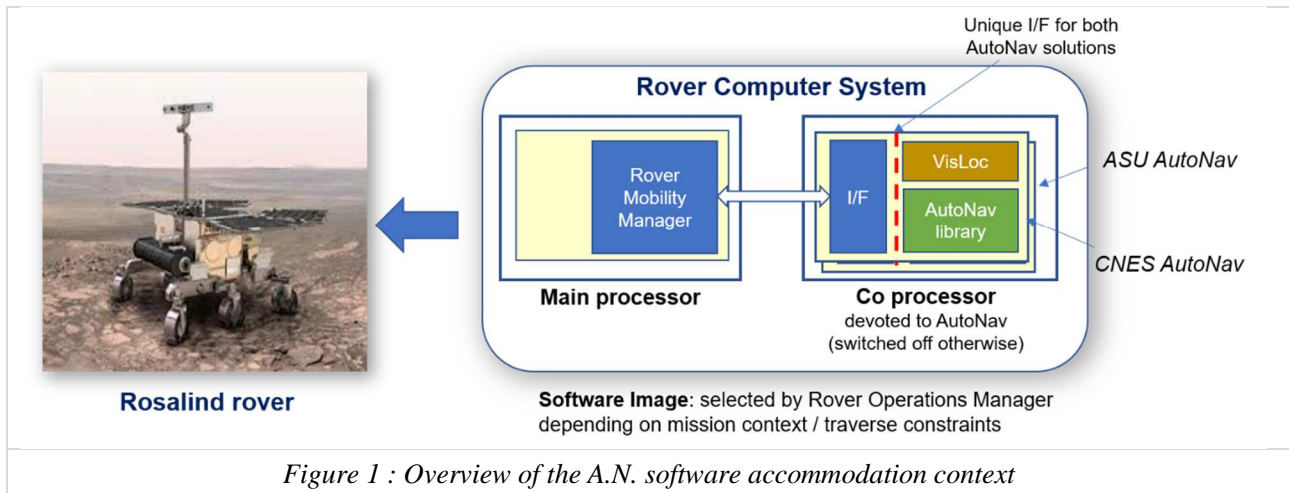
2.1. Main principles

The *AutoNav* Software will allow the Rover to perform autonomously traverses up to 70m per sol using a model of the local environment built through stereovision techniques. In order to maintain the highest possible accuracy in the reconstruction of the local environment, each traverse will require multiple cycles of terrain analysis and path execution. At each navigation stop the rover reacquires new information about its environment that will be used to ensure the rover safety over the next locomotion step. For ExoMars, the maximum distance between two consecutive navigation stops is 2.4m. The main inputs used for the *AutoNav* computations are the images acquired from the rover Stereoscopic Imaging System or Stereo bench, this Stereo bench is placed on the top part of the rover mast which can be rotated using a Pan and Tilt Unit (PTU). At each navigation stop, the rover will perform several stereo-image acquisitions rotating the Pan & Tilt Unit in order to cover a sufficient zone.

The *AutoNav* Software is designed to offer two modes of operation:

- **AutoTrav mode:** the ground defines the goal or a list of waypoints and the *AutoNav* module computes periodically a safe trajectory increment to the goal.

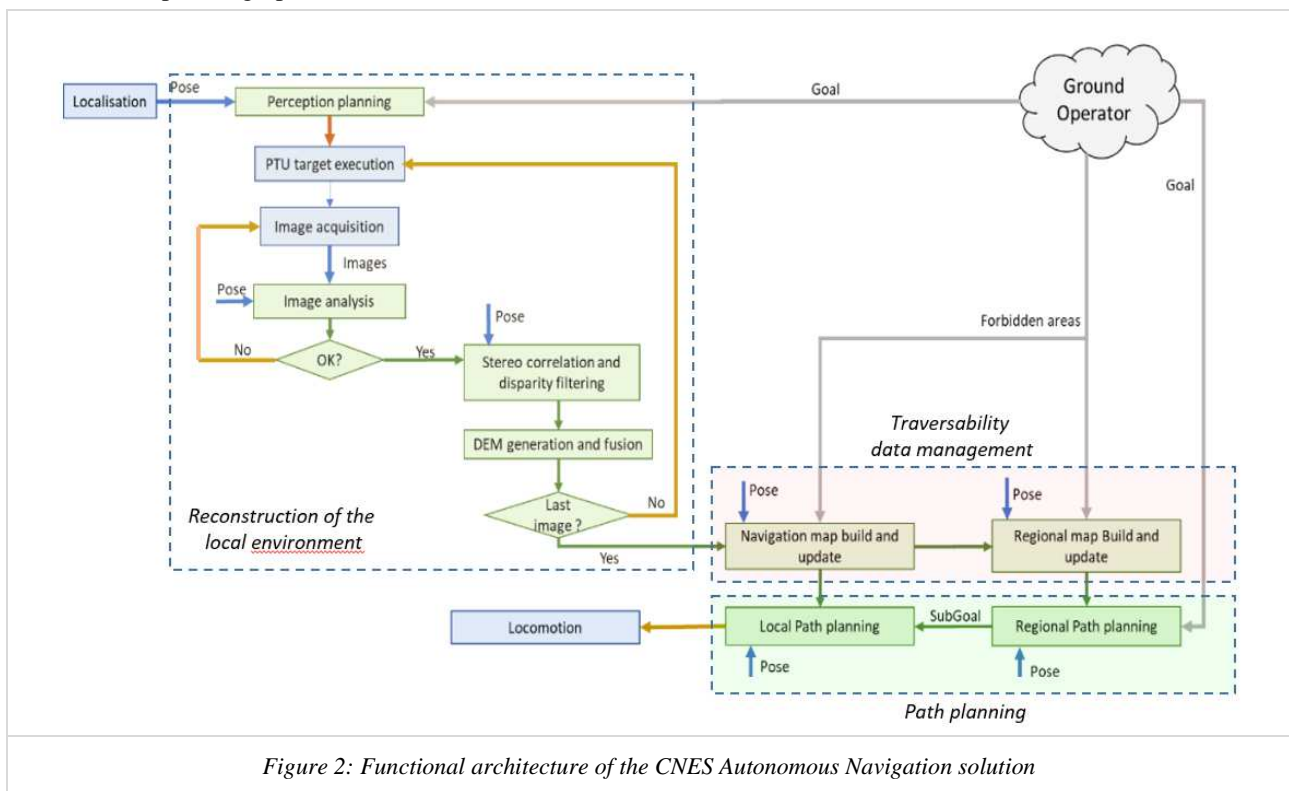
- **CheckPath mode:** the ground defines a trajectory that is to be executed accurately by the Rover. The AutoNav module is in charge of analysing periodically the terrain traversability along the trajectory and signalling the presence of obstacles.



2.2. Overall functional architecture

The *AutoNav* architecture relies on three main functional blocks that are presented on Figure 2 and described with further details in the sequel. These blocks are the following:

- Reconstruction of the environment
- Traversability knowledge management
- Path planning / path verification



2.2.1. Reconstruction of the environment

The Digital Elevation Model (DEM) constitutes the representation of the terrain in the rover vicinity. The DEM generated for each navigation stop is actually obtained by the fusion of smaller DEMs derived from perceptions acquired in different directions. Each perception is a pair of stereoscopic images that are processed to build a disparity map through an optimized stereo correlation algorithm. Before activating stereo correlation, the images are analyzed to make sure the luminance properties are adequate and a new acquisition is possibly requested with a new set of parameters. Three perceptions constitute the baseline in the *AutoTrav* mode and this number is usually reduced when *CheckPath* mode is selected. In the ExoMars context, the DEM is a square bitmap centered on the current rover position - its size and cell resolution are configurable and typical values are respectively 14 m x 14 m and 4 x 4 cm.

2.2.2. Traversability data management

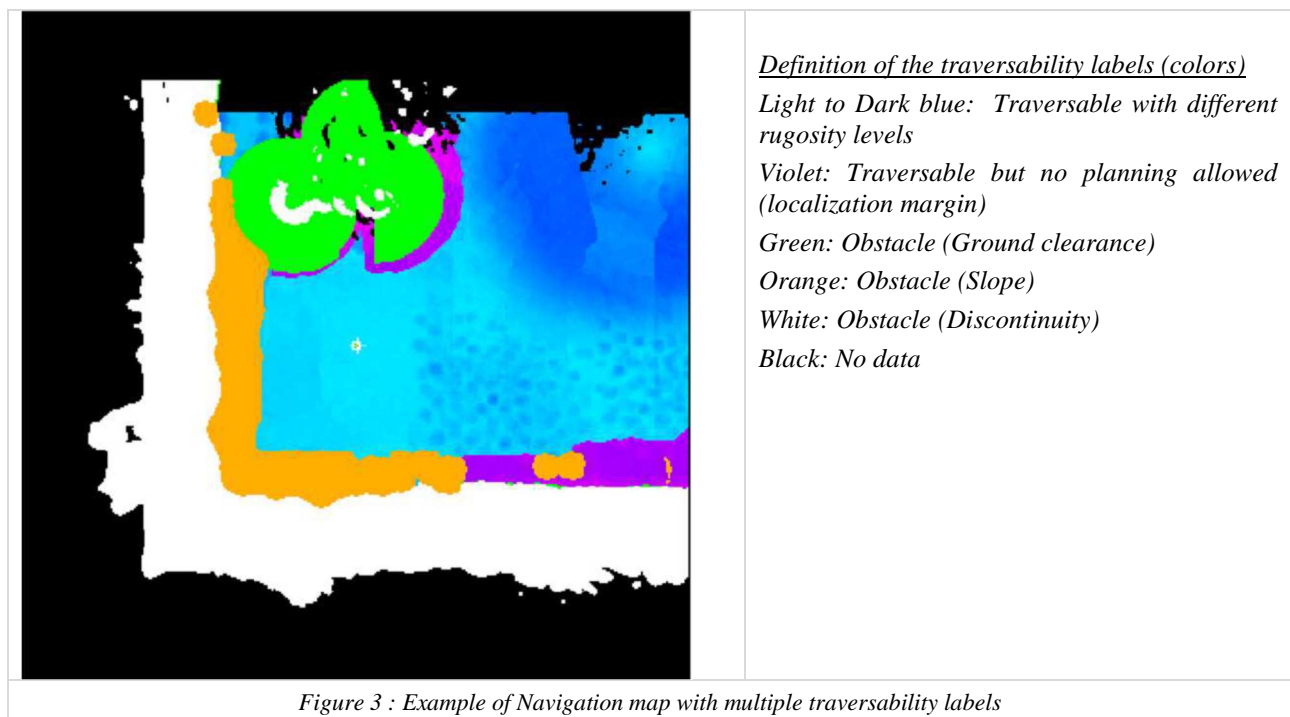
Processing the Digital Elevation Model to manage the overall knowledge of the terrain traversability involves two successive steps associated to the following representations:

- 1) **Local Navigation Map**: the map size and resolution match the DEM ones and the generation of new data consists in assessing the traversability of the different "active" DEM cells using a model of the rover. The most recent cell information is then fused with the traversability data obtained at the previous navigation stops.
- 2) **Regional Map**: this map is introduced to store any relevant traversability information that falls outside the Local Navigation Map and would be lost otherwise. It is designed to keep track of the existence of past obstacles with a minimal memory overhead and uses for that purpose a vector representation. The map update implies the extraction of the obstacle contours in the most recent Navigation Map and its insertion or merging with the already existing list of obstacles.

The traversability assessment is performed in two steps: (1) a first analysis of the terrain roughness or local discontinuities at the wheel scale is applied to the full DEM, (2) the second analysis involves the placement of the rover model over the DEM to determine the rover configuration and the possible intersection between the ground and some rover parts. Multiple criteria are considered to determine if a rover configuration is actually safe:

- rover roll and pitch angles (compulsory)
- boogie angles (compulsory)
- ground clearance for the rover bottom face (compulsory)
- ground clearance for the solar panels (optional)

This step cannot be applied to the cells located near the borders due to the rover placement constraints.



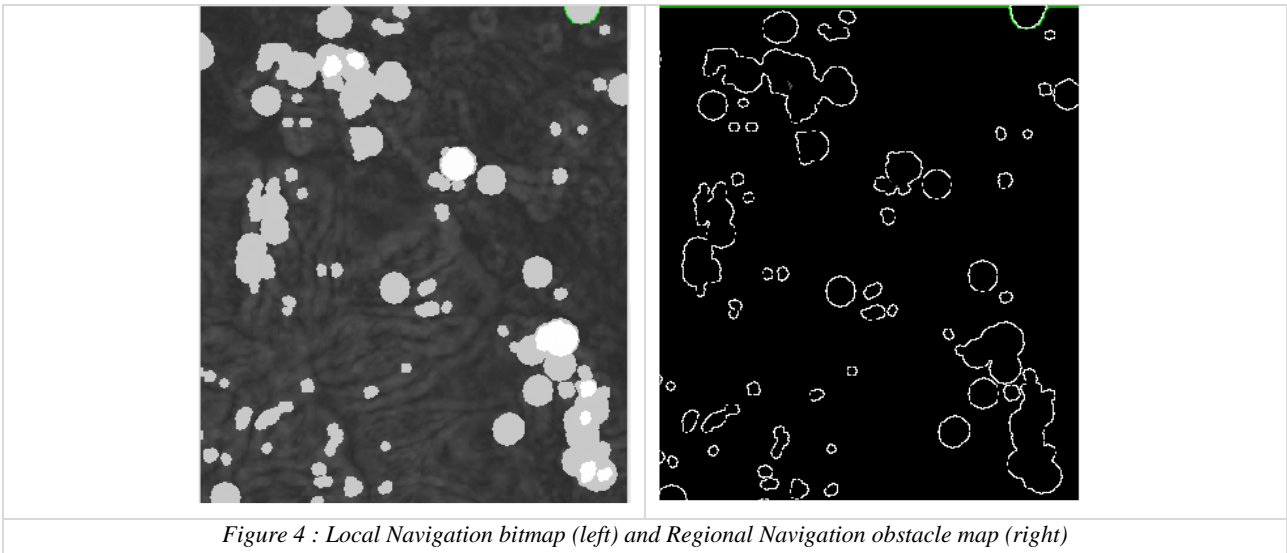


Figure 4 : Local Navigation bitmap (left) and Regional Navigation obstacle map (right)

2.2.3. Path planning

Path planning is performed in two successive steps:

- 1) *Regional Path Planning*: this step exploits the Regional Navigation Map and computes the optimal long range path from the current rover position to the given Goal. Regional path planning is based on the tangent graph theory in which the nodes are constituted of Start Point, Goal Points and Tangent points between the start point, the goal point and the obstacles. The interface between the Regional Path Planning and the Local Path Planning is a Local Sub-Goal, which is defined as the last navigable point of the Regional Path included in the navigable zone around the Rover.
- 2) *Local Path planning*: the process consists in searching for the best valid path that allows to reach the Local Sub-Goal while satisfying the rover curvature and safety constraints. To reduce the dimension of the search space, the algorithm examines a finite number of path candidates that are represented as a series of curved arcs with a given length. The search is performed as follows:
 - Check the validity of a set of arc candidates starting from the current best node.
 - Compare these valid arc candidates considering the estimated cost to reach the goal
 - Store all valid arc candidates in the order of increasing cost and select the best one for the next step
 - Backtrack in the search tree if an exploration step fails to yield valid candidates. The exploration starts again from the next best candidate.

2.2.4. Path verification

This functionality is less demanding since it is limited to the safety analysis of the predefined activity by the ground and determines whether its execution is safe or not. The design of this functionality assumes the following constraint: the rover must be capable to execute some *AutoTrav* command without any ground provided *KickStart* after the completion of the *CheckPath* activity whether it is successful or not. This means that *AutoNav* has to produce and manage all data needed to perform the traversability assessment within the rover blind zone that will exist at any given stop.

The processing to be performed depends on the type of trajectory commanded by the ground:

- *Point Turn*: the rover is requested to perform a rotation on the spot to reach a new heading – the function analyses the safety of this move
- *Path*: the rover is requested to follow a trajectory defined by a list of Cartesian positions – the function analyses of the safety of this displacement taking into account possible localization and locomotion errors. Two versions have been implemented:
 - *Conservative*: The traversability analysis is applied to the full DEM generated at the current cycle and it includes all the possible rover yaw configurations (full revolution). This is identical to the analysis performed in *AutoTrav* mode.
 - *Optimized*: The traversability analysis is focused on a limited set of cells and rover yaw configurations. The associated products are generated during the map generation step that intervenes at the end of the perception cycle. This optimized version allows to increase the traverse capabilities as it focuses the analysis around the

actual heading of the commanded paths. However, a full analysis is still performed in parallel to satisfy the above mentioned constraint. Note that it is not currently applicable to the Point Turn activity.

2.3. Design limitations

2.3.1. Size of the navigation map

Data fusion between consecutive navigation stops is currently performed at the navigation map level to avoid the complexity of co-registering DEMs computed at different rover locations. Since most of the traversability assessment implies the rover placement over the DEM and sometimes also the check of the interaction between the solar panels and the ground, the analysis is only fully achievable on a subset of the cells: the ones located far enough from the borders. This means that the band of the terrain located in front of the rover can be only analysed using the discontinuity criterion. The assessment is therefore partial and if no discontinuity is detected the cells within this band are flagged as unknown even though it is observable by the cameras (Figure 5.1). This restriction has two impacts:

- 1) the length of the *KickStart* trajectory needs to be properly sized when starting some activity in *AutoNav* mode so as to reach the full traversability analysis zone.
- 2) the length of the commanded path has to be adjusted when the DEM size gets reduced (the rover is for instance approaching the top of a ridge) and the *NavMap* horizon appears too close from the current rover position. This situation is illustrated on Figure 5.2 where the displacement between cycles $k-1$ and k is not consistent with the size of the known area. A knowledge gap appears in the navigation map and will affect the path planning process.

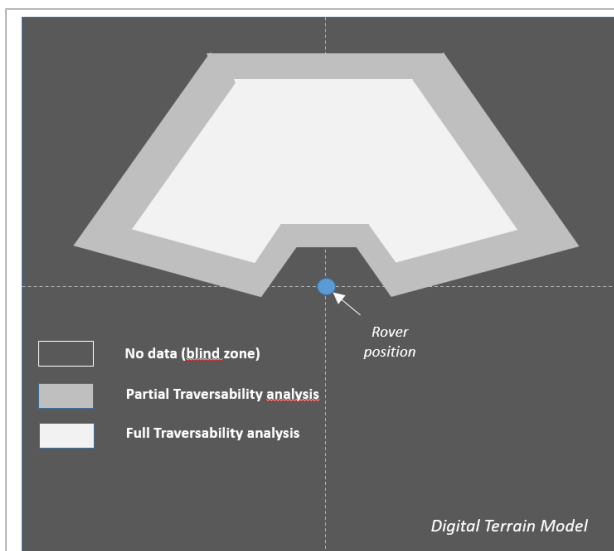


Figure 5.1: Part of the Digital Terrain Model usable for traversability assessment

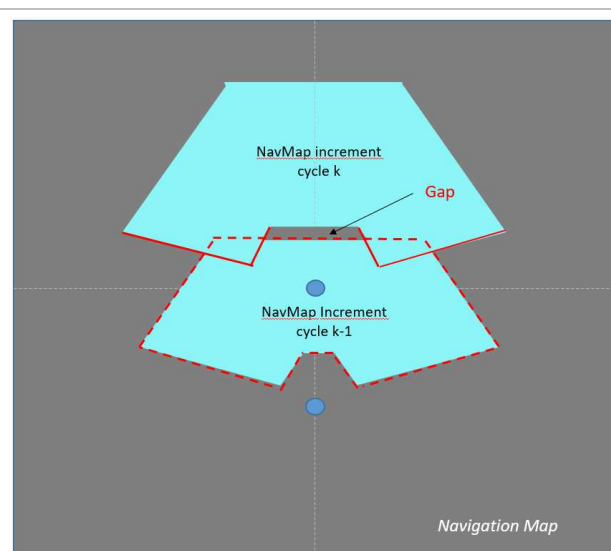
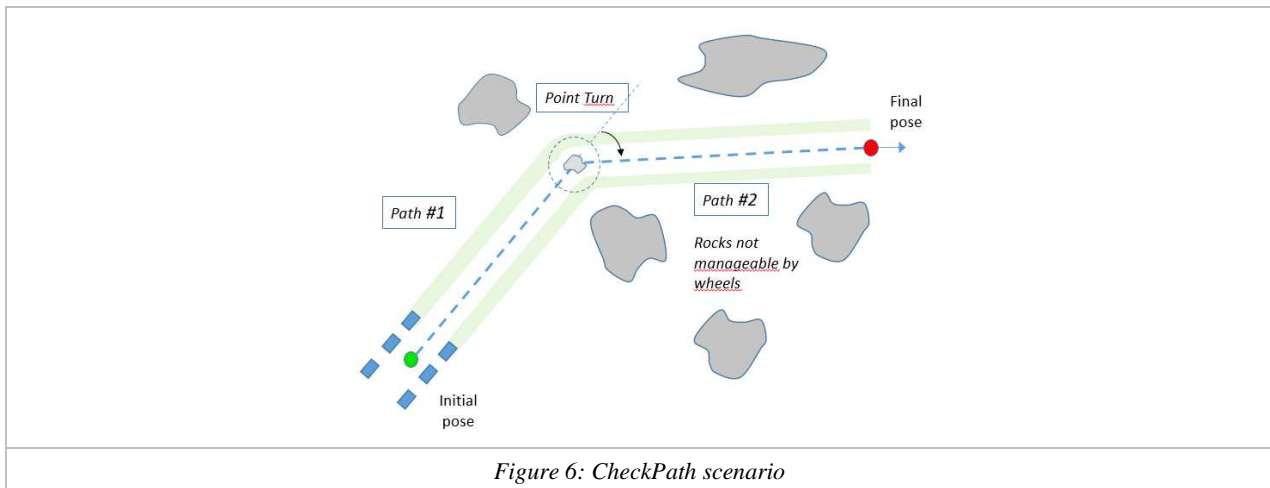


Figure 5.2: Merging of Navigation Maps from two consecutive cycles - The path commanded at cycle $k-1$ is too large for the current size of the navigation map

2.3.2. CheckPath capability and performance

The current implementation suffers from the following limitations:

- the Point Turn command does not benefit from the optimized analysis (the analysis restricted to a smaller set of rover yaw configurations). This leads to a limitation of the capability to cross difficult terrains using a sequence that is for instance composed of two Paths separated by a Point Turn. Such a limiting scenario is presented on Figure 6 where a rock with specific characteristics is standing in the area where a Point Turn could be commanded (the rock is compatible with ground clearance requirements but not crossable by the rover wheels). In this case, the first path is considered safe with the optimized analysis but the Point Turn will be declared unsafe even though the rover can rotate over the obstacle.
- there is no noticeable gain in terms of navigation data processing since a full Navigation Map is built and updated at each cycle – this situation could be improved by relaxing the on-board situation awareness constraints when *CheckPath* mode is activated



3. Test activities and results

3.1. Context

Before exercising the *AutoNav* software in a real environment, it has been extensively validated using the Numeric Software Validation Facility (NSVF) that constitutes the official test bench to validate and verify the rover on-board software. A replica of this tool developed by ADS-UK has been delivered to CNES to allow the validation of the specific CNES solution. The NSVF is composed by the high-fidelity simulation of the on-board computer (OBC) and Rover Vehicle Simulator with simulation models of the Rover equipment, environment as well as the vehicle's dynamics and kinematics. This facility allows closed loop simulation with a full image of the on-board software hosted by the OBC simulator.

This section presents an overview of the results obtained during the Ground Test Model (GTM) test campaigns that were the preliminary steps to the Qualification Review Process for the CNES *AutoNav* Software. The Rover Module GTM provides a complete functioning hardware build of the Rover Vehicle and represents the model most similar to the Proto Flight Model (PFM). These activities are conducted in the Mars Yard of the Rover Operations Control Centre (ROCC) that is illustrated on Figure 7.1. The yard dimensions are 16 m by 20 m and features a mostly flat sandy terrain with a mound in the middle. During these tests the rover is operated by the ROCC as in real operations.

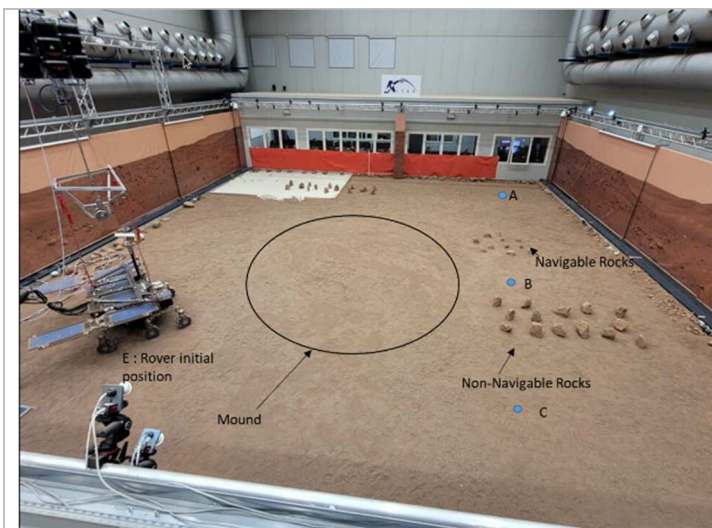


Figure 7.1: Overall view of the terrain
(image credit ESA/TAS-I/ALTEC)

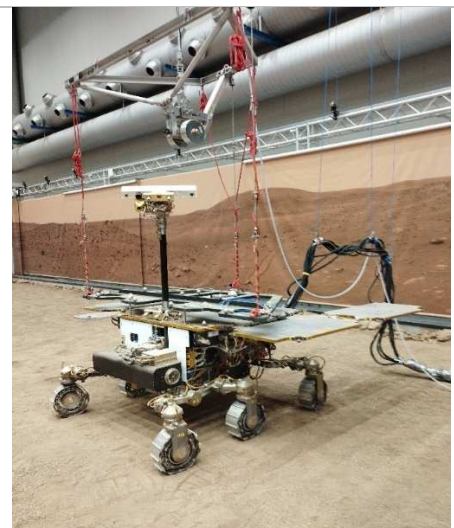


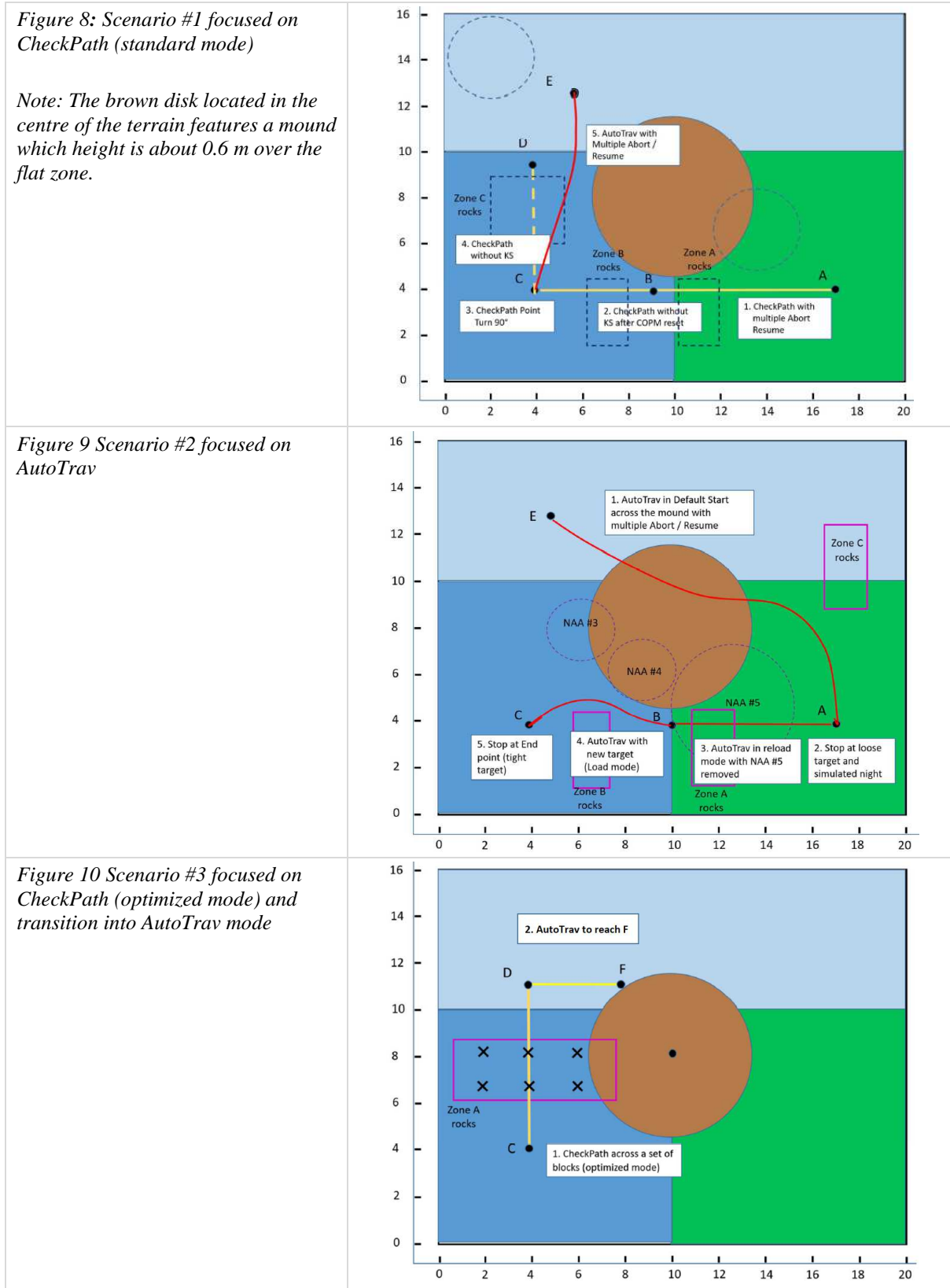
Figure 7.2: GTM rover with its suspension system

Two campaigns have been carried out:

- the Confidence Test Campaign defined by CNES
- the IST/SVT Campaign defined by ESA/TAS-I.

3.2. Confidence Test Campaign

The Confidence Test Campaign was run from 9th to 13th May 2022 with CNES on site participation. CNES proposed three nominal scenarios that were agreed with ESA, TAS-I and ALTEC. Each scenario was composed of several segments allowing the verification of the main *AutoNav* Software functionalities exercising *CheckPath* and *AutoTrav* mode across the terrain with different degrees of difficulty. The scenarios are presented on Figures 8 to 10.



In conclusion, more than 50 m of navigation system driving were achieved. All segments were considered as passed except one that was affected by some shortcoming: the rover stopped 1 m away from the goal at the end of the second

scenario since the goal was considered as non-reachable due to a path-planning constraint that was not properly implemented (this limitation has been corrected in the subsequent version).

In order to give more information about the AutoNav behaviour during the GTM Confident Test Campaign, a detailed analysis is presented for the scenario #2.

3.2.1. Scenario #2: Detailed AutoTrav Results

This scenario consisted on 3 segments, all in *AutoTrav* mode. The results (Navigation Maps) obtained for the three segments of this scenario are presented here below:

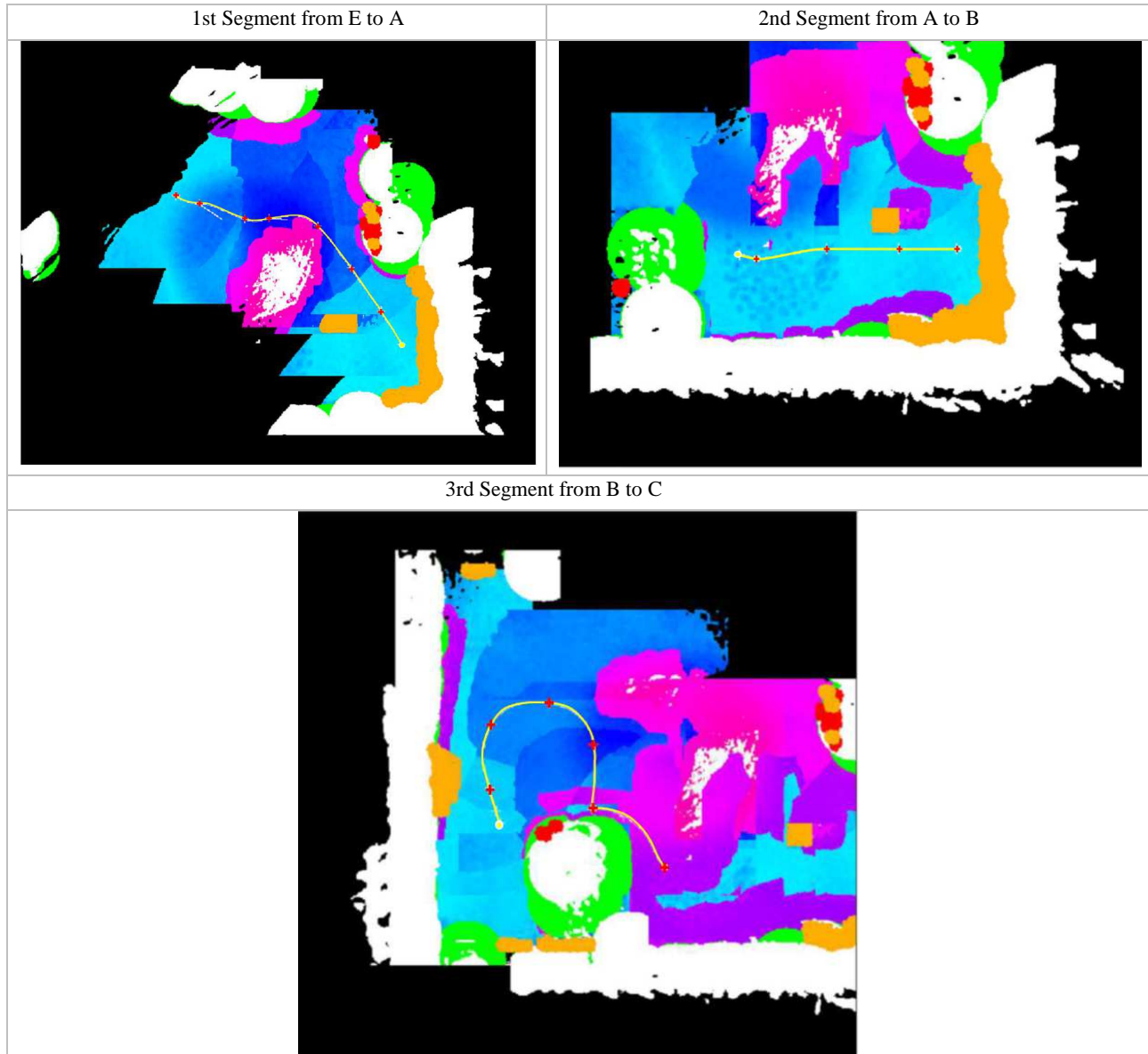


Figure 11: Commanded Trajectories superimposed into the navigation maps for scenario #2

For the 3rd segment, it was initially planned in the test description that the distance between the Non-Allowed Area NAA#3 and the rocks would allow the rover to pass through the corridor left between the two obstacles. During the test setup, this distance was reduced and therefore the corridor was closed. As a result, the rover proceeded to go around the NAA to achieve the final goal, as it can be seen on Figure 11.

For this sequence the rover reached the consecutive goals avoiding all the obstacles, the behaviour of the *AutoNav* Software was fully nominal.

Detailed information gathered during the second segment is presented here below. The initial rover position (A) and the goal (B) with respect to the mission terrain frame were:

- Initial position: $x=16.9\text{m}$, $y=4.2\text{m}$, heading= 295°

- Goal position: $x=9.5, y=4.0\text{m}$

In total 6 cycles were performed, the table below presents the view of the rover from the external cameras of the ROCC, the view from the Stereo bench cameras for the central acquisition as well as the commanded Path Sequence for each cycle. The rover correctly considers the rocks as navigable and is able to reach the given goal precisely.







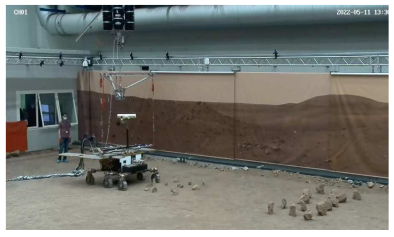

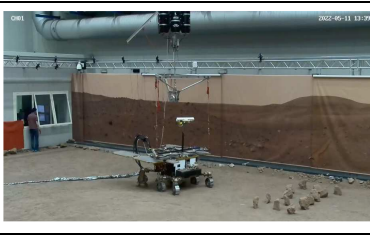

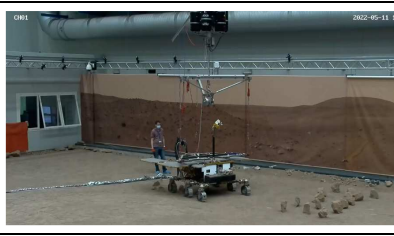
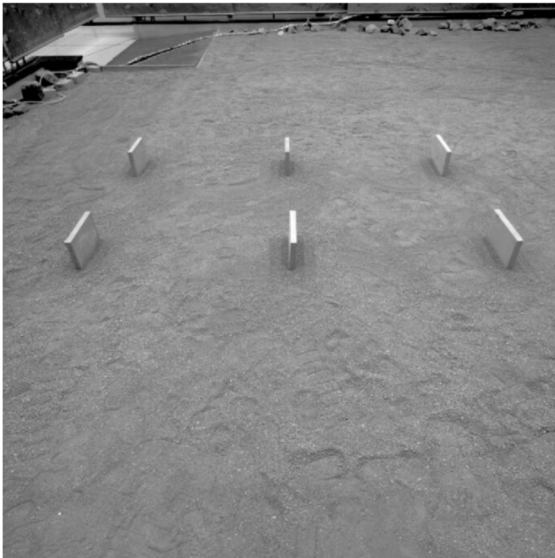
			
Pose #1 – Point Turn = 115°	Pose #1 central view	Pose #2 – Path = 1.9 m	Pose #2 central view
			
Pose #3 – Path = 2.4 m	Pose #3 central view	Pose #4 – Path = 2.4 m	Pose #4 central view
			
Pose #5 – Path = 0.56 m	Pose #5 central view	Pose #6 – No path commanded	

Figure 12: Rover position and view for the different navigation cycles of segment #2 (image credit ESA/TAS-I/ALTEC)

NB: Pose #1 view illustrates the robustness of the *AutoNav* system which behaviour is not affected by the presence of windows and associated reflections in the camera field of view. Stereo correlation is not impacted and the vertical walls are properly reconstructed.

3.2.2. Detailed CheckPath results

The optimized version of the CheckPath mode is exercised during scenario #3 where the commanded path is a rectilinear displacement over an area containing 2 rows of identical obstacles (Figure 13.1). The obstacle height (20 cm) is beyond the terrain discontinuity threshold and is therefore not crossable by the rover wheels. A direct path is nevertheless achievable if the median obstacles remain below the rover bottom face since the obstacle height is compatible with the ground clearance margin set for the test.



Cycle	Rover Position X (m)	Rover Position Y (m)	Rover Heading (dg)
1	4.000	4.000	90.57
2	4.016	5.878	89.74
3	4.007	8.287	90.28
4	4.009	10.68	89.38
5	3.997	11.021	90.06

The length of the linear displacement is 7 m and is executed in 4 mobility steps.

Figure 13.1: Central view of the terrain at the beginning of the CheckPath activity

Figure 13.2: Rover configuration for all cycles of the CheckPath activity (4 mobility steps)

The data products created during the two first cycles are presented on Figure 14 to illustrate the behaviour of the optimized CheckPath mode. The first column displays the DEM where the obstacles are clearly visible and accurately reconstructed. The second column provides the full Navigation Map where the obstacle area is considered non navigable since all cells in that area are analysed with all criteria active and taking into account all possible yaw configurations. The map that is actually used for path verification is shown in the third column: the cell validity is only considered for a set of cells around the commanded path and for a limited range of yaw configurations (-/+20°).

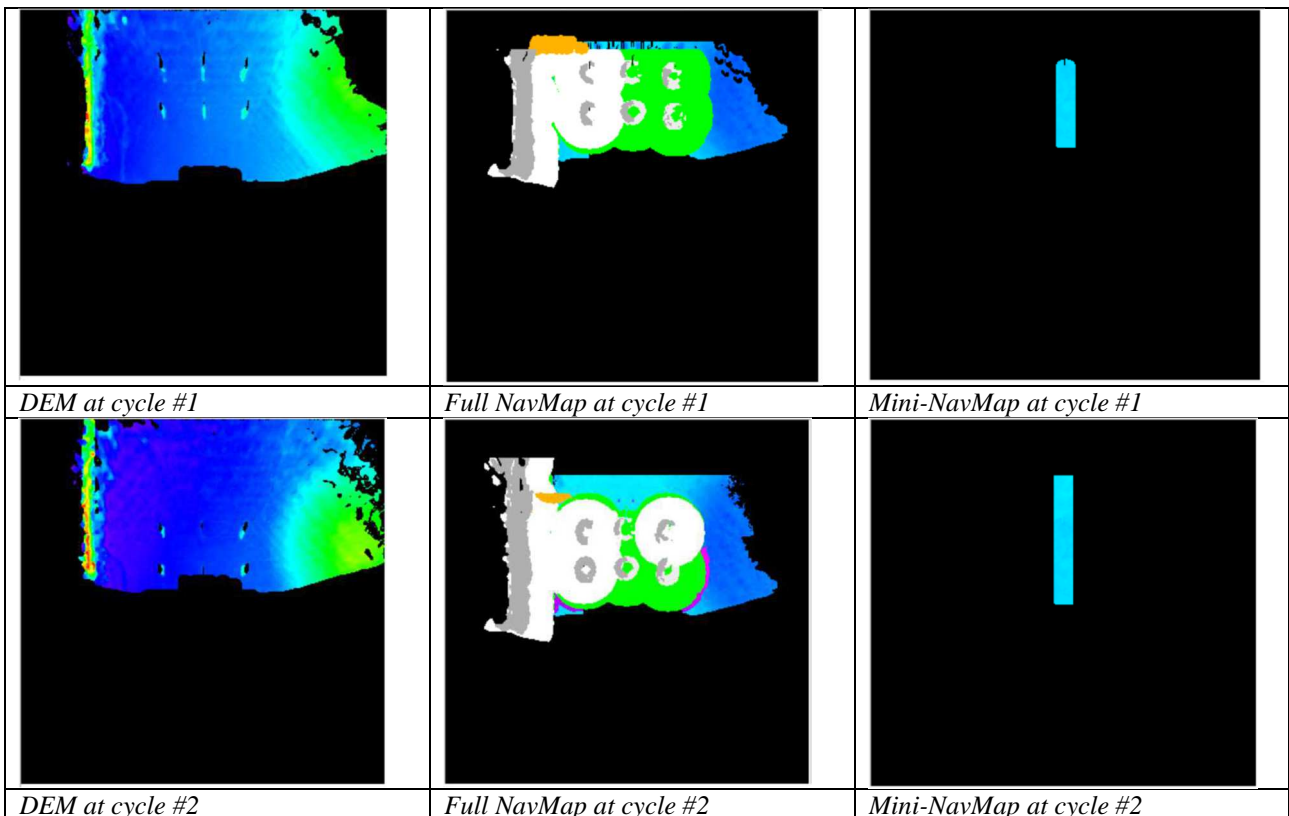


Figure 14: Data products for the first cycles of the optimized CheckPath test (scenario #3)

The crossing of this obstacle area was successful and demonstrated the potential benefit of this technique to get through difficult areas through a more aggressive management of the rover safety.

3.3. IST/SVT4 Test Campaign

After the correct completion of the Confidence Test Campaign and minor corrections, the formal IST/SVT4 Campaign took place on May 19th 2022. The test sequence, that was common to the two *AutoNav* Solutions on ExoMars, was specified by ESA, TAS-I and ALTEC. The test sequence was a succession of segments in *CheckPath* and *AutoTrav* mode activating several functionalities of the *AutoNav* Software.

Six segments were executed and all passed except the last one that triggered a Path Planning failure. This behaviour was partially caused by a wrong configuration of the traverse that was not initialized in the expected configuration.

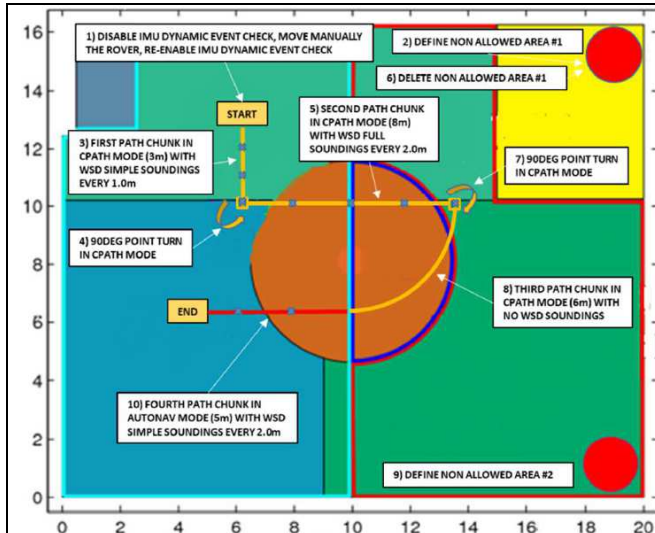


Figure 15: ISTV Test campaign

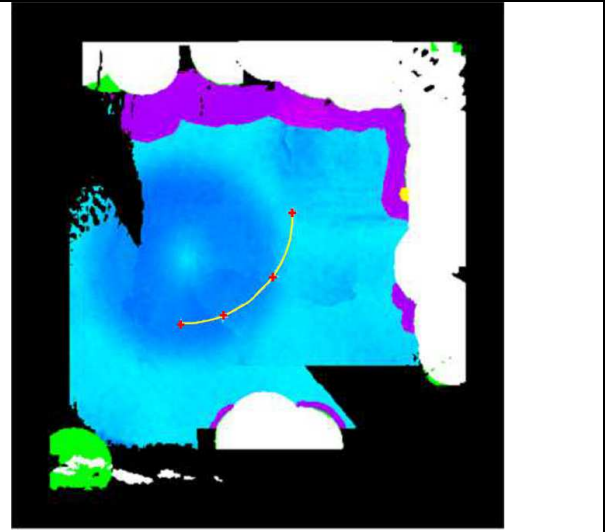


Figure 16: Navigation Map built from the start of segment 3 to the end of Segment 5 – the path commanded for Segment 5 is superimposed

As a conclusion, the GTM tests allowed to validate the CNES *AutoNav* in the most representative rover scenario. The behaviour of the CNES *AutoNav* was considered as nominal and the performances of the solution were compliant with the project requirements.

3.4. Lessons learned

3.4.1. Programming

During the execution of tests, the definition of the *KickStart* path appeared to be a critical step considering the possibility to cause planning failure when its length is not consistent with the rover blind zone size. If the solar panel check is activated, the blind zone gets larger and the *KickStart* shall be increased in order to be sufficient long to reach the area identified as traversable. The ability to perform automatically some geometric checks before command execution has been identified as an important time saver even during the early preparation phases.

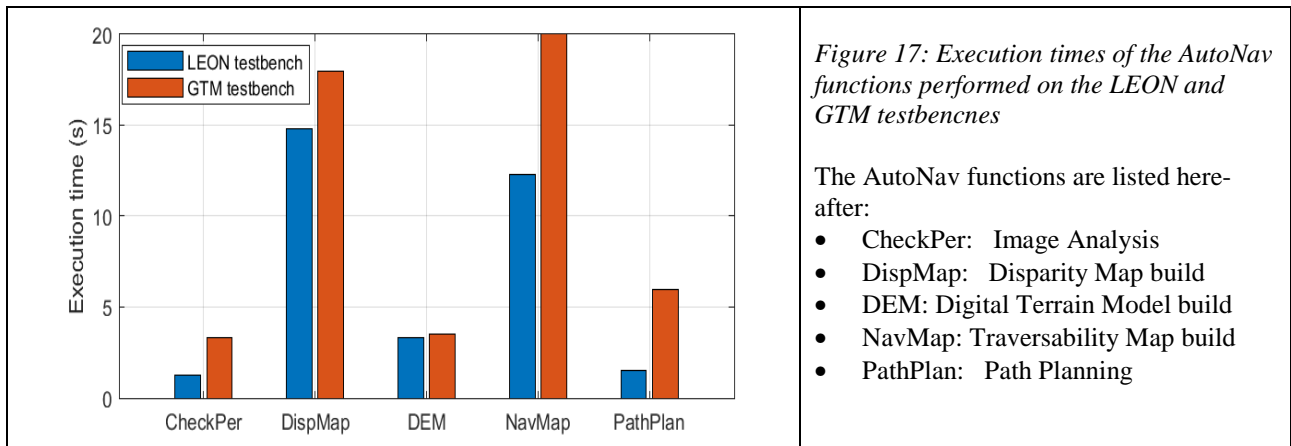
3.4.2. Execution time

The tests allowed to assess the duration of the different steps of the Autonomous Navigation process and estimate the traverse capability over time. One of the objectives was to identify the impact of the *AutoNav* functionalities in the overall duration of each rover navigation cycle. On the GTM test bench, the time measurement of individual activities is based on the time stamping of messages generated by the AN Manager running on the Main Processor that indicate the activation and the completion of any activity. For the *AutoNav* functions running on the Co-Processor, these time markers provide an overestimation since the time difference includes overheads associated at least to the roundtrip data transfer between the two computers. The assessment of these functions has been initially performed within CNES premises using a LEON2 test bench which characteristics/utilization context are given here-after:

Board n°	GR-CPCI-AT697
Frequency	80 Mhz
Stack usage	Activated
Cache utilization	Activated
Code optimization	-O2

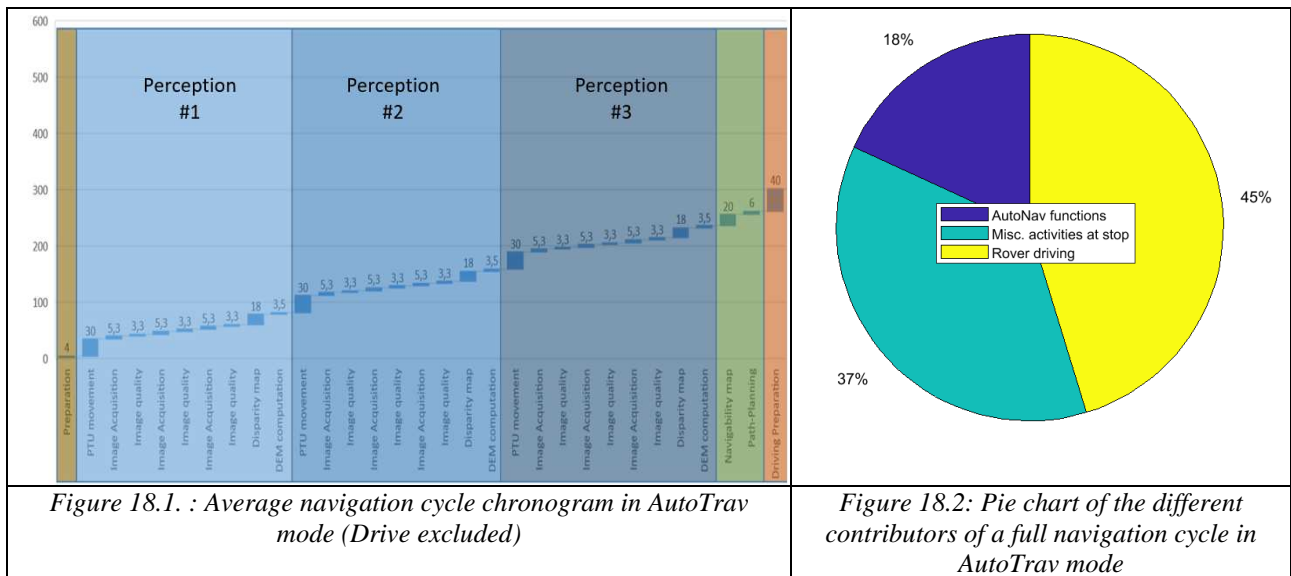
There is some variability on the execution time of some functions depending on the configuration option and this is particularly the case for the traversability map generation where the activation of the solar panel leads to some noticeable increase. This variation is leveraged out by considering a 50% utilization of the Solar panel check option.

The execution time differences between the LEON2 and GTM test benches are presented on Figure 17 and the results appear consistent considering the above mentioned remarks. The execution times are larger in the GTM context and the difference is mostly explainable by the overhead due to the data storage services utilization (data storage is managed by the Main Processor).



The AutoNav functions are not the only contributors to the duration of the Stop phase. Miscellaneous operations need to take place to ensure the acquisition of all data required for the proper environment analysis. These include the rover tranquilization period prior to IMU readout and the Pan and Tilt Unit motions needed for the different perceptions.

To illustrate the multiple contributions to the duration of a navigation cycle in *AutoTrav* mode, Figure 18.1 presents the chronogram of the different activities that take place during the Stop phase in some average configuration. It must be outlined that the navigation cycle involves three perceptions and three consecutive image acquisitions with different tuning parameters per perception.



Finally, Figure 18.2 shows the different contributors to the duration of a complete navigation cycle in the most favorable case where the rover is driving the longest path increment (2.4 m). In this case, the duration of the Drive phase is 250 s for a total cycle duration of 541 s. An extrapolation of these figures leads to an achievable rover speed of about 16 meters in the best possible conditions. By comparison, the speed actually observed during the GTM tests is presented on Table 2. The mean speed values computed over the 3 segments performed in *AutoTrav* mode are smaller but this difference is explained by the presence of Point Turns during the traverse, shorter path increments at the end of the segment and some activation of the Abort/Resume capability.

	Segment1	Segment2	Segment3
Time segment (s)	4522,7	2682,3	4522,3
Time segment (h)	1,25	0,745	1,25
Distance segment (m)	15,54	7,38	13,909
Mean speed (m/h)	12,37	9,9	11,07

Table 2: Rover mean speed in AutoTrav mode

The fact that *AutoNav* functionalities contribute to less than 20% of the cycle total duration and less than one third of the Stop phase clearly indicates that no significant improvement can be obtained through an optimization in that area.

Conversely, some possibilities of time reduction exist within the others activities of the Stop phase. The first one concern the PTU motions that contribute to 90 seconds for 3 perceptions. Most of this latency could be avoided by starting the next PTU motion once the images are considered valid since the processing of the current perception would actually run in parallel. This scheme scenario was actually proposed for the Sample Fetch Rover project and could be implemented for ExoMars.

Some additional measure with smaller benefit could concern the parameter setting used for image acquisition. For each new perception, a default setting is always applied to the first acquisition and the pair of images is rejected most of the time. Converging to the proper image exposure may require in that case two additional cycles of image acquisition and analysis (each cycle lasts about 9 s). Some noticeable time saving could be obtained if the final setting of each perception is stored to be used for the first acquisition of the next perception.

Speed improvement: Selecting the *CheckPath* mode when the terrain conditions appear favorable constitutes the alternate way to increase the mean speed. However, the expected gain will remain moderate considering the already mentioned constraint taken into account for the design: the rover must be capable to execute some *AutoTrav* command without any ground provided KickStart after the successful completion of a *CheckPath* sequence.

In this context, a full navigation map is still generated at each cycle and the time reduction comes from three areas: the number of perceptions is smaller (2 instead of 3), the size of the DEM and the navigation map is smaller since the solar panels check is not active, path planning is replaced by the simple and quick verification of a small set of cells. The duration of a full navigation cycle is reduced by about 80 s and this corresponds to a mean speed gain of about 15%.

The next section that mentions the future activities will address the possible improvement of the mean speed in *CheckPath* mode.

4. Status and further activities

CNES *AutoNav* passed successfully the tests on the GTM. This achievement was significant considering the intensive validation effort to be squeezed in a particularly short timeframe constrained by the late delivery of some essential test bench software components. CNES *AutoNav* is a complex software that includes a high number of internal verification and protection mechanisms that enable to detect a large spectrum of anomalies, interrupt the processing and report to the main application. The schedule constraints have imposed to focus the validation effort on the expected nominal and degraded cases that have been triggered through specific test scenario. Full coverage has been reached nevertheless but using analysis for a noticeable number of anomaly cases which triggering was either difficult or impossible to achieve at system level. Some improvement of the coverage metrics is therefore desirable for a class B software and is being considered.

The next foreseen launch date is now in 2028 and this postponement brings lot of challenges but also opportunities. On the challenge side, the maintenance of the development and testing environment comes first. The software and hardware tools must be kept operational until the end of the mission and the obsolescence management is a concern for some components. For CNES, the NSVF is the most critical asset since it is developed and provided by the rover manufacturer (ADS-UK). A specific contract is to be setup to ensure the maintenance of this testing facility in case of breakdown or necessary upgrade. The latter situation is likely to occur since the extra time offered for testing on the GTM will allow to keep populating the lessons learned and identify limitations or even anomalies. The rover mobility software is one of the valid candidates for update and this will concern the NSVF as a consequence. Therefore, CNES must remain ready to perform late *AutoNav* software evolutions and associated validation campaigns to guarantee its full compatibility with the rest of the system.

Conversely, the additional time offers valuable opportunities to perform some slight improvement of the CNES *AutoNav* software at the functional and performance level. In this context, it has been agreed with ESA to postpone CNES *AutoNav* qualification to allow some upgrade of the software quality metrics that would be performed in combination with the above-mentioned evolutions.

CNES is actually planning to improve the current *AutoNav* version in two different directions:

- introduce a refinement of the *CheckPath* mode implementation that currently offers two different behaviours that have been already tested on GTM: the conservative assessment similar to what is used also in the *AutoTrav* mode and an optimized assessment technique that allows the rover to move safely in more challenging environments
- increase the traversability analysis scope at short range

4.1. *CheckPath* improvement

To improve the efficiency of the path verification functionality, it is now assumed that the ground will always provide a *KickStart* trajectory when entering the *AutoTrav* mode after some activity performed in any other mode (*CheckPath* or *Direct Drive*). In this context, there is no need any more to ensure the availability of terrain knowledge around the rover at any navigation stop. The assessment of the terrain traversability can be limited to a set of cells surrounding the commanded path with margins that take into account the localization and mobility errors. For each cell, the assessment considers also a limited range of yaw configurations around the rover direction. This allows to reduce dramatically the generation of the Navigation Map to be used for the verification of the Path activity – this map is called “Mini NavMap” in the sequel.

The verification of the *Point Turn* activity imposes some adaptation of the data management process since the range of yaw configurations to be considered depends on the commanded rover heading that is still unknown at this stage. If a *Point Turn* is commanded after the completion of the current *Path*, the terrain assessment will have to be updated for a set of cells located around the rover position. This situation is illustrated on Figure 19.1 that presents a ground commanded trajectory sequence composed of two rectilinear *Path* activities separated by a *Point Turn* activity. The range of yaw configurations is the same for the two *Path* activities (Range #1) whereas the assessment of the *Point Turn* activity requires an extended yaw range (Range #2) taking into account the rover rotation magnitude.

In order to anticipate the possible execution of a *Point Turn* at the end of the first *Path* activity and the necessary update of the Mini NavMap generated at the current cycle, some additional data is to be stored: the terrain information around the *Path* endpoint. This information provided through a specific small size DEM allows to compute at the next cycle a reduced navigation map associated to the rover blind zone that is merged to the *Mini-NavMap*. The data processing scheme that takes place during the *Point Turn* cycle is illustrated on Figure 19.2. The processing is actually identical for all cycles except the first one that may benefit from a *KickStart*. The small size DEM is always stored but is used only if necessary. In addition, no perception is being requested for the *Point Turn* activity since all information about the environment is already available. This allows to speed up significantly the current cycle.

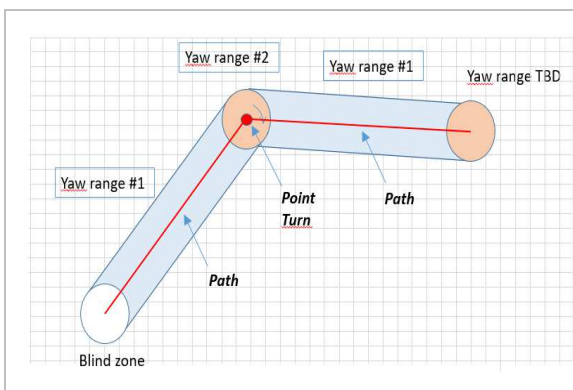


Figure 19.1: Commanded trajectory sequence composed of two Path separated by a Point Turn

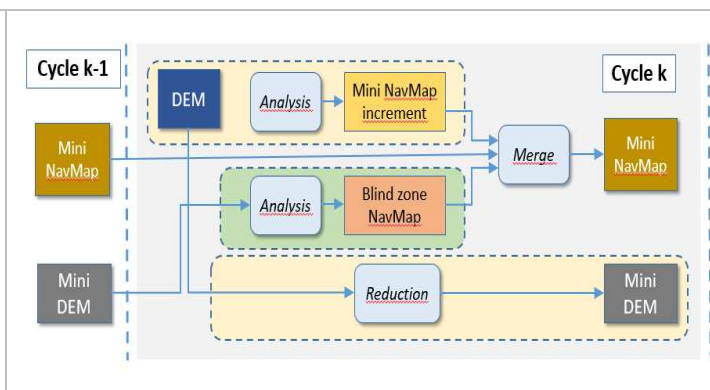


Figure 19.2: Data processing scheme applicable to any cycle in *CheckPath* mode – Green box: performed if a *Point Turn* is commanded – Yellow box: performed in any other cycle

4.1.1. Generation of the Mini-NavMap

For the sake of clarity, the different steps involved in the generation of the *Mini-NavMap* are described here-after:

- 1) *Generation of index bitmaps*: this step consists in projecting in the DEM the section of the path to be executed in the current cycle in order to determine the set of cells concerned by the traversability assessment. These cells are grouped in the specific bitmaps:
 - bitmap #1 is associated to the rover origin reference points and the active cells are to be analyzed according to the slope and ground clearance criteria - some additional information is attached to each active cell: the reference rover heading to be used for the definition of the possible range of heading configurations
 - bitmap #2 is associated to the reference points of the rover wheels and the active cells are to be analyzed according to the discontinuity criterion
- 2) *Generation of the discontinuity map*: the input is the DEM local maxima map and determines the discontinuity value for the active cells of bitmap#2. Next, the cells located in the vicinity of non traversable cells are also flagged as non traversable in order to drive the center of the rover away from the large discontinuities. To do so, the radius of non traversability expansion takes is the sum of the rover half size and the localization error.
- 3) *Generation of the triple criteria map (slope, ground clearance and boogie clearance)*: this step takes as input the map of local maxima as well as the map of active cells (bitmap #1) and the associated table of reference

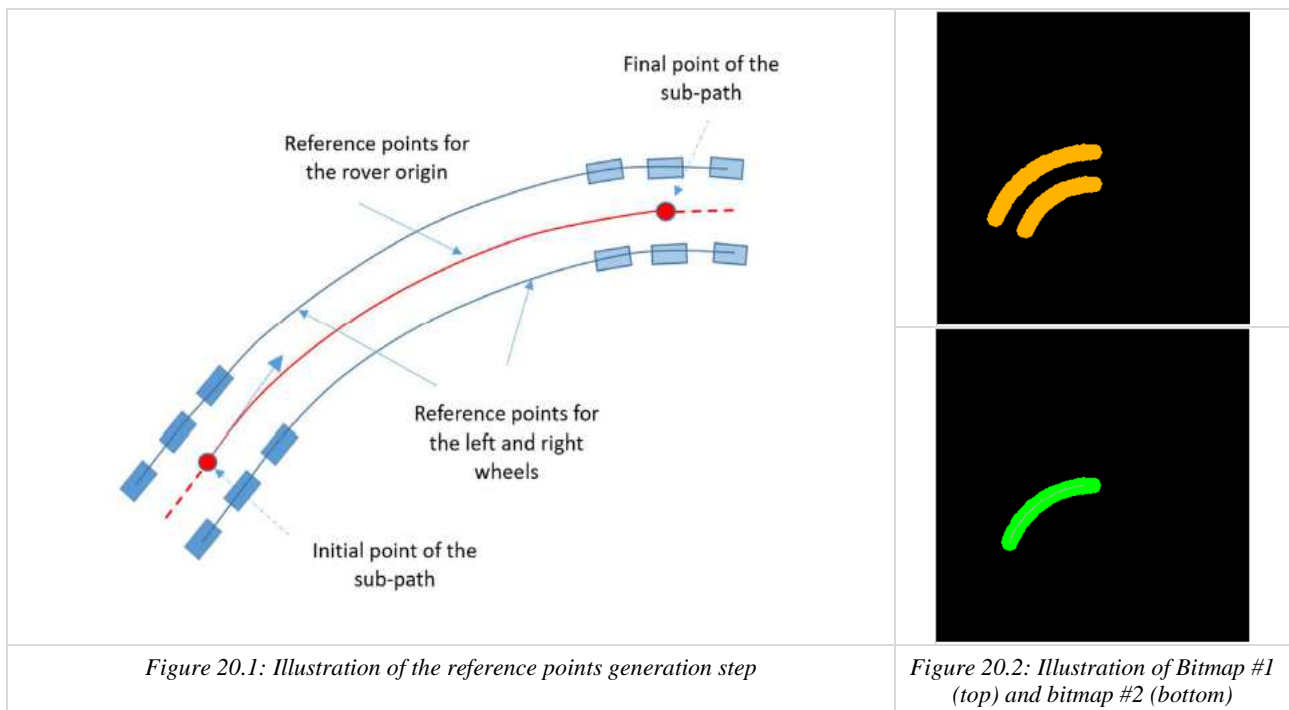
headings. It computes for all active cells of bitmap #1 the set of rover configurations within the possible heading range. Cells are flagged as traversable if they satisfy the 3 criteria of slope, ground clearance and boogie clearance.

- 4) *Generation of the Mini-NavMap*: all data present in the discontinuity and the triple criteria maps are collected in the *Mini-NavMap* increment and then merged with the previous *Mini-NavMap* if it already exists.

4.1.2. Management of the Blind-zone NavMap

As indicated on Figure 19.2, two types of processing are performed depending on the context:

- when a *Path* activity is verified, the DEM reconstructed is stored in a reduced form: only the area surrounding the endpoint of the path to be executed at the next mobility step is actually extracted – this area covers the rover footprint including the solar panels if requested and that assumes a possible 360° revolution on the spot
- when a *Point Turn* activity is processed, only the DEM produced at the previous cycle is considered and it is analysed taking into account the commanded heading span – the generation of the associated *NavMap* involves the same steps described in section 4.1.1



4.2. Improvement of the traversability analysis scope at short range

As indicated in section 2.3.1, the traversability assessment is partial at close range since the rover placement cannot be performed on the associated cells of the current DEM. For this close range region, full traversability data is already available but it is based on the DEM computed at the previous cycle which uncertainty is distance dependent. Some risk of traversability overestimation does exist if the DEM inaccuracy has not been adequately accounted for.

The improvement that is foreseen consists in fusing the DEM data from two consecutive cycle to allow a full assessment of the short range band mentioned previously. So far, this fusion was not included in the baseline considering the accuracy and computational cost of the DEMs co-registration techniques. In the ExoMars context, this task is challenging due to the rover relative pose uncertainties between two consecutive navigation stops where the tilt angle contributes to the most part. The method to be implemented consists in estimating the relative tilt between the relevant overlapping section of the two DEMs and focuses on two narrow strips located in the front and rear of this section. The estimation is performed for multiple position offsets to determine the best fit for both strips. If the residue associated to the best fit is low enough, the fusion is allowed and a new traversability assessment can be achieved for that section. Otherwise, the old assessment prevails.

5. Conclusions

The development, validation and verification approach described before has been carried out for the CNES *AutoNav* Software during the past years. The completion of the GTM Test Campaign was an important milestone for the CNES *AutoNav* Software and the final reward of a huge effort to overcome the very specific technical and schedule challenges

imposed to the whole software project. The results of the GTM Test Campaigns have confirmed the maturity of the software that responded to the needs of the project through the respect of the rover safety requirements and the demonstration of high computational time performances.

Taking into account the current status of the ExoMars project with a launch date not before 2028, the nominal activities concern the execution of the Maintenance Plan that constitute a real challenge considering the CNES dependence on the NSVF testing facility. During this maintenance phase, the possibility to manage functional updates impacting the *AutoNav* Software and requested by the project needs is also being considered and this adds some additional uncertainty in the process. In parallel, the maintenance process offers the opportunity to take into account lessons learned and perform some improvement of the *AutoNav* software at functional and performance level. Some enhancement of the software metrics is expected to be implemented as well to reach the highest quality standard. These evolutions will obviously imply additional validation activities involving CNES, NSVF and GTM test facilities before the final Software Acceptance Review. The scope is potentially ambitious and will be therefore carefully tuned taking into account the real added value of the upgrades and the available resources within CNES and the project partners.

References

- [1] Autonomous navigation: a development roadmap for ExoMars. L. Rastel, M. Maurette, ASTRA 2006.
- [2] Robotic planetary exploration: autonomous navigation in cluttered unknown environments A.Torres AIAA 2012
- [3] Onboard Autonomous Path Planning for a Planetary Exploration Rover A.Rusu, S.Moreno AIAA 2013
- [4] Perception and autonomous navigation for planetary missions M.Delpech ASTRA 2015
- [5] CNES Autonomous Navigation solution for ExoMars Rover: Validation approach and results – SpaceOps 2023 – 6-10 March 2023 – Dubai.