

The MMX Rover on Phobos: The Preliminary Design of the DLR Autonomous Navigation Experiment

Mallikarjuna Vayugundla¹ Tim Bodenmüller¹ Martin J. Schuster¹ Marcus G. Müller¹

Lukas Meyer¹ Patrick Kenny² Florian Schuler¹ Markus Bihler¹ Wolfgang Stürzl¹

Bernhard-Michael Steinmetz¹ Jörg Langwald¹ Andreas Lund² Riccardo Giubilato¹

Armin Wedler¹ Rudolph Triebel¹ Michal Smíšek¹ Markus Grebenstein¹

Institute of Robotics and Mechatronics¹, Institute for Software Technology²
German Aerospace Center (DLR)
Muenchener Str. 20, 82234 Wessling, Germany
firstname.lastname@dlr.de

Abstract—This paper summarizes the challenges of deep space planetary robotic-exploration missions, using the example of the DLR Autonomous Navigation Experiment within the MMX Rover project, and presents a preliminary design of the proposed solution to safe navigation of the MMX Rover on Phobos. The MMX Rover, a joint contribution of the German Aerospace Center (DLR) and the Centre National d’Etudes Spatiales (CNES) is part of the Martian Moons eXploration (MMX) Mission by the Japan Aerospace Exploration Agency (JAXA), whose mission objective is to understand the origin of the two Martian moons. The mission involves scientific data collection and a sample return from Phobos, the bigger of the two Martian moons. The mission is scheduled to launch in the third quarter of 2024. The MMX Rover will fly as a payload and will be jettisoned onto Phobos from a low altitude of about 40 metres. The rover has multiple objectives: terrain assessment to mitigate the risk for the sample-return approach of the spacecraft, exploration of the surface of Phobos and act as a scientific and technology demonstration platform. In this context, the Robotics and Mechatronics Institute, DLR, is preparing to perform an Autonomous Navigation Experiment as a technology demonstration to showcase the advantages of robot autonomy for exploring distant celestial bodies. If successful, the MMX Rover will be the first man-made object to land on and explore Phobos.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. DLR AUTONOMOUS NAVIGATION EXPERIMENT ON PHOBOS.....	2
3. MMX ROVER DESIGN	3
4. DLR NAVIGATION SYSTEM	5
5. CONCLUSION	14
ACKNOWLEDGMENTS	14
REFERENCES	15
BIOGRAPHY	16

978-1-7281-7436-5/21/\$31.00 ©2021 IEEE

Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

1. INTRODUCTION

The study of a planet’s moon gives precious insights towards the understanding of planetary formation processes. The chemical analysis of lunar samples supported the theory that the Moon originated from the impact between the Earth and an impactor [1], and that lunar material mainly derived from the proto-Earth’s mantle [2]. The origin of the Martian moons Phobos and Deimos is however unclear, as very few missions have been dedicated to their study and analyses never have been performed in-situ. Apart from the Soviet Phobos-1 and Phobos-2 missions, the first failing during transit and the second capturing 37 images but failing before deploying a lander, no missions have been specifically dedicated to the Martian Moons. To this date, knowledge on their orbital properties and composition has been given by missions targeted at Mars or other bodies which, as a secondary objective, also observed Phobos and Deimos [3]. The first of these missions, and that which made the closest approach to the Martian moons, Mariner 9, observed their topography from a range of 1200 km. This mission was followed by the two Viking spacecraft. Within the Viking missions a complete Digital Terrain Model (DTM) and image mosaic of Phobos were built. The density of Phobos and Deimos was estimated to be less than 2 g/cm³, labeling them as anomalous with respect to Mars and asteroids. The analysis of Phobos’s librations also confirmed the hypothesis that the Martian Moon is composed of a thick layer of regolith and a low density homogeneous interior. The spectral analyses of OSIRIS (Optical, Spectroscopic and Infrared Remote Imaging System) onboard the Rosetta spacecraft during its Mars fly-by [4] supported the theory that Phobos is a captured D-Type asteroid [5], a theory that originated after spectral measurements from the Mars Pathfinder spacecraft [6].

To gain more insight on the formation of the Martian moons, the Japan Aerospace Exploration Agency (JAXA) plans a sample-return mission named MMX: Martian Moons eXploration. The MMX spacecraft will orbit both Phobos and Deimos on quasi-satellite orbits, land on Phobos, gather samples of regolith and bring them back to Earth. [7], [8]. The MMX spacecraft is scheduled to be launched in 2024 and return in 2029. The mission has two principal goals. The first is to understand the origin of both moons and specifically to confirm one of the two hypotheses on Phobos’ formation:

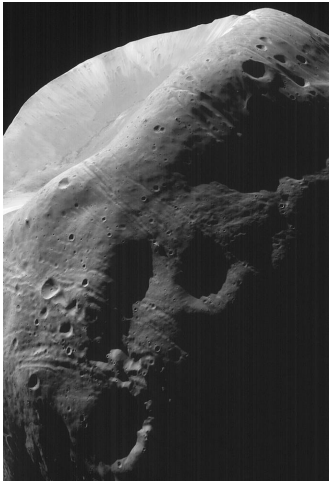


Figure 1. Close-up Image of Phobos captured by the Mars Orbiter Camera (MOC) onboard Mars Global Surveyor spacecraft; Courtesy NASA/JPL-Caltech.

a captured asteroid or the result of a giant impact. The second main goal is to observe surface processes on Mars such as the water-cycle dynamics or the air-ground system from a new vantage point. To further aid the in-situ analysis of Phobos, the MMX spacecraft will also deploy a rover, developed jointly by the Centre National d'Etudes Spatiales (CNES) and the German Aerospace Center (DLR). The main scientific goals of the rover will be the characterization of the mechanical properties, thermal properties, and mineralogical composition of the surface material, as well as the close-up imaging of the surface structure [9]. As part of the scientific payload, the MMX Rover will accommodate a Raman spectrometer (RAX), a radiometer (miniRAD), wheel cameras and navigation cameras. Thanks to close in-situ imaging of the surface characteristics, the rover will also collect data to aid the MMX spacecraft for its subsequent landing on Phobos [10]. The MMX Rover will serve, furthermore, as a technology demonstrator for locomotion in the low-gravity environment of Phobos and for the autonomous navigation. This paper is dedicated to an analysis of the system, environmental, and mission specific challenges to autonomous navigation on Phobos with the MMX Rover. Furthermore, it presents our preliminary design of the planned architecture for autonomous navigation considering the environment and system specific challenges and constraints.

The rest of the paper is structured as follows: Section 2 presents an overview on the DLR Autonomous Navigation Experiment on Phobos with the MMX Rover. Section 3 presents details on the MMX Rover and the sensors it carries. Section 4 presents the preliminary design of the planned architectures for autonomous navigation as well as details on the key sub-components necessary for the autonomous navigation solution. Finally, Section 5 gives some concluding remarks.

2. DLR AUTONOMOUS NAVIGATION EXPERIMENT ON PHOBOS

The objective of the DLR Autonomous Navigation Experiment is to navigate safely using navigation autonomy considering the limitations of the small MMX Rover. Such autonomy is required because teleoperation is not feasible

considering the large time delay between Earth and Phobos, limited communication possibilities, and the short lifespan of the rover mission. The lack of knowledge regarding the terrain of Phobos, its very low gravity (which is of the order of 1/1000th of the Earth), as well as sensor, energy, and computational power limitations of the MMX Rover make the task highly challenging. However, the rover's on-board computer (OBC) combining CPU and FPGA architectures provides opportunities to implement highly optimized solutions for performing autonomous navigation. The rover is equipped with a fixed stereo camera setup, which is its primary sensor for localization and perception. Additional challenges arise due to the absence of a camera pan-tilt mechanism and of a high-accuracy inertial measurement unit (IMU), as typically used in terrestrial systems and on higher-gravity planetary bodies such as Mars to estimate a rover's orientation. The proposed solution is based upon the previous research experience of the team at the Robotics and Mechatronics Institute (RM), DLR in the area of autonomous navigation in unstructured planetary-like environments using stereo vision [11], [12], [13], [14], [15]. A rendering of the rover in a simulation environment (developed by the Institute of System Dynamics and Control, DLR) of Phobos as defined by the MMX Environment Requirement Document is shown in Figure 2.

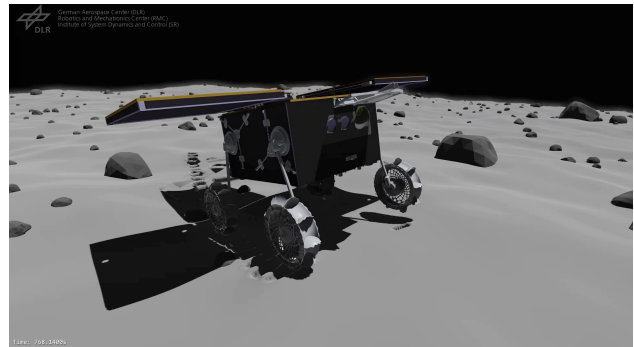


Figure 2. MMX Rover rendering in simulation, courtesy of DLR Institute of System Dynamics and Control.

Time Plan

MMX spacecraft will be launched during the 2024 Earth-Mars launch window. The DLR Autonomous Navigation Experiment part of the MMX Rover project started at beginning of 2019 and will last until the launch. The time of deployment of the rover on Phobos is not yet fixed and is likely to be around 2025-2026. The rover mission is planned for 100 days, whereas the first month will be dedicated to rover commissioning. After this the rover will perform science experiments using the instruments as well as explore Phobos by navigating with and without autonomy. Figure 3 shows the MMX mission timeline. The spacecraft will leave Mars in 2028 and reach Earth in 2029. The communication between the rover and the Earth via telemetry and telecommands will be relayed over the MMX spacecraft. For the communication between the MMX spacecraft and the rover, an S-band RF system will be used [9].

Experiment goals and objectives

The goal of this experiment is to investigate the benefits of on-board semi-local navigation functions for the operation of small rover systems, especially if direct teleoperation is not feasible. Here, semi-local navigation states that the intended modules mainly provide local navigation functionality, like e.g. local motion estimation, relative localization, obstacle

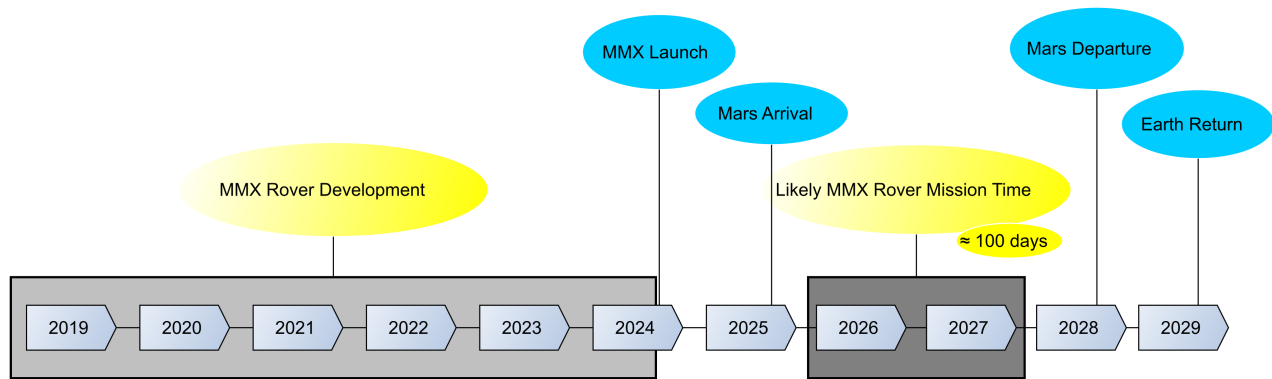


Figure 3. Mission Timeline: MMX Mission launch is scheduled for September 2024; MMX Rover development started in 2019; the rover mission on Phobos spans a period of approximately 100 days and likely will be in 2026 or 2027; MMX Spacecraft is scheduled to return to Earth in 2029 with a sample.

detection and local path planning. These local navigation functions rely only on current sensor readings, but are extended to store a very limited amount of past sensor readings and processing data that allow for e.g. simple path adaption around obstacles. Thus, computational requirements are kept at a minimum. The need for and usage of such autonomous navigation functions, for safe and effective navigation on distant planet bodies, in order to explore and perform science has been successfully shown by previous Mars rovers, like the Mars Exploration Rovers [16].

A higher degree of autonomy and global navigation are not considered for this experiment, as the scientific objectives of the mission and the rover, do not demand it. Moreover, there are constraints like the limited power, memory and computational resources onboard the rover. Other factors like uncertainty about the environment of Phobos and its low gravity also make it risky to go for a higher degree of autonomy.

The experiment campaign will involve a series of module verification tests followed by autonomous navigation experiments. The following functional modules needed for the proposed autonomous navigation solution will be tested first:

- **Semi-Global Matching (SGM):** The on-board calculation of depth images from the stereo camera system is the essential base component for all other navigation modules. We plan to use the Semi-Global Matching (SGM) algorithm [17] as an accelerated FPGA implementation to provide disparity images and depth images. In the experiment, we aim to investigate the robustness and accuracy of the method under different environmental settings.
- **Visual Odometry (VO):** The Visual Odometry (VO) uses the stereo images and the corresponding disparity image for the estimation of the camera motion and thus, of the rover motion.
- **Obstacle Detection:** On-board Obstacle Detection creates an obstacle map from the calculated depth image. The resulting map marks areas that are difficult to pass or impassable.
- **Obstacle Avoidance:** Obstacle Avoidance makes sure that the rover navigates autonomously avoiding all the obstacles in its path.

In the second step, with basic modules verified, we aim to evaluate the robustness and performance of basic autonomy functions for navigation.

- **Semi-local Localization/Pose Estimation by sensor data fusion:** There might be additional information available, capable of refining the pose estimation yielded purely by integrating visual odometry. Such options include an attitude sensor, wheel odometry measurements in low wheel-slip scenarios, feedback from satellite imagery, or the Sun-tracker. If the measurements of said additional sensors would be of sufficient quality, it might be possible to fuse this information together with visual odometry data, to yield a refined pose estimation. The usability of these additional sources of information will be investigated as more details on final selection of sensors becomes known.

- **Semi-Local Mapping:** If computational resources available on board suffice, some level of awareness of the environment around the rover can be stored. Doing so will be investigated.

- **Wheel Slip Detection:** The rover being skid-steered and the unknown properties of regolith on Phobos are two reasons to expect some slippage of wheels. While the wheel motors will contain revolution measurement sensors, when there is wheel slippage, the distance travelled by the rover is not exactly equal to the distance moved by the wheel's circumference. Furthermore, slippage entails the risk of sinkage of the rover wheels. This is only amplified by the fact that the rover's wheels will have relatively large regolith-lifting grousers placed on their circumference [10] [18]. Unobserved wheel slippage may result in the burying of the rover's wheel and rover getting stuck which would jeopardize the locomotion as well as the autonomous navigation experiment. Hence, this experiment focuses on detecting and observing wheel slip by comparison of wheel odometry and visual odometry.

- **Semi-Local Autonomous Navigation:** As the rover will see only up to a distance of 1-2 m in front of itself, driving manually more than this distance in one communication loop is a challenge. Here, we plan to show the advantage of semi-local autonomous navigation in driving a greater distance, for example, 4 m.

3. MMX ROVER DESIGN

Hardware Specification

The rover is box-shaped with a maximum mass of 30kg. The rover chassis height and width are approximately 375 mm and 230 mm respectively. It has four legs which are equipped with the wheels at the end and the rover is skid-steered. The legs are used for uprighting, sun-pointing to recharge using solar panels and for driving. The rover has a targeted driving speed

of 0.1 to 4 mm/s. It has solar arrays, which are unfolded once it is in the upright position on Phobos. It has shutters to avoid dust on the cameras. It has batteries to store power collected using the solar panels. Figure 4 shows an impression of the MMX Rover prototype in a sand testbed used for initial testing of the locomotion sub-system. It is equipped with four scientific instruments:

- NavCAM: to provide images of the environment before the rover for safe locomotion as well as a perception sensor for the autonomous navigation solution.
- WheelCAM: regolith analysis from wheel interaction with the surface.
- miniRAD: for conducting thermal analysis.
- RAX (Raman Spectrometer for MMX): for conducting surface mineralogy.

Refer to [9] and [10] for an overview of the rover system and the scientific instruments carried by it. To obtain more details on the ongoing development of the MMX Rover, especially on the locomotion subsystem, refer to [18].



Figure 4. MMX Rover prototype in the testbed used for locomotion testing, courtesy of DLR Institute of System Dynamics and Control. One can see the wheel’s design and leg’s placement. It doesn’t contain yet the navigation sensors and scientific instruments. Note: This is not a complete representation of the final flight model but representative of it.

NavCAM

Out of the scientific instruments mentioned above, NavCAM, which is a stereo-camera setup, is the primary sensor for the autonomous navigation experiment. It is used both for

mapping of the environment as well as to compute visual odometry in order to estimate the pose of the rover.

It is a stereo-camera setup fixed inside the rectangular body of the rover and does not have a pan-tilt mechanism. This is due to the limitations coming from the size, weight and body-design constraints of the MMX Rover. Some specifications of the NavCAM are shown in Table 1.

Parameter	Value
Camera height above the ground	30 - 35 cm
Stereo camera baseline	6 cm
Downward tilt angle	23 degrees
Horizontal angular field of view	80 degrees
Vertical angular field of view	80 degrees
Camera Sensor Resolution	2048 x 2048 pixels
Pixel pitch	5.5 μm
Focal length	7.8 mm

Table 1. MMX NavCAM Specifications

OBC Architecture

The avionic system of the MMX Rover will be realized by an on-board computer (OBC) consisting of a Xilinx Zynq-7045 system-on-chip integrated on a board with 512 MB of DDR3 RAM with error correcting codes enabled. The OBC is the central control unit and will be responsible for the communication with the orbiter as well the rover’s scientific objectives and operational control, such as navigation, locomotion and power management.

In order to ensure the reliable behavior of the MMX Rover, the system software and all experimental software running on the OBC will be implemented following the ARINC 653 [19] approach. The Xtratum Next Generation (XNG) hypervisor from the vendor FentISS is used to provide time and space (memory) partitioning of the software components. XNG resembles ARINC 653, a standard for virtualizing and partitioning software, though it is not strictly compliant. The strong isolation between components aids in developing the complex software system and shifts the testing burden from integration towards unit testing. This isolation is especially important due to the distributed, multi-agency nature of the rover’s development, and the mixed criticality of the software components. The hypervisor and partition structure of the MMX software is shown in Figure 5.

The hypervisor acts as the lowest software layer, interacting directly with the OBC hardware. The remaining software is divided into partitions, virtualized execution environments, which are allocated CPU time based on a time-slot approach. Each partition has access to a single core of the dual-core processor. Memory accesses between partitions are prevented, except for through a highly structured inter-partition communication interface. This ensures each partition access to both processor time and memory, independent of the functioning (or malfunctioning) of other partitions. A severe failure in a low-criticality section cannot cause resource starvation in a critical section, such as memory management, central control or the telecommand interface. It also allows each partition to operate with its own operating system or real-time executive, according to the software component’s unique requirements.

The navigation software developed by DLR occupies one partition, named NAVDLR, in the hypervisor. This partition uses the open-source RTEMS real-time operating system

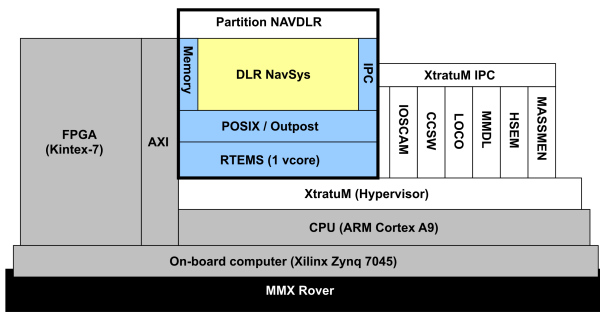


Figure 5. Hypervisor and partition structure of the MMX OBC. The XNG hypervisor interacts directly with the hardware of the Zynq-7045 system-on-chip. Multiple partitions run within the hypervisor. Within the NAVDLR partition the RTEMS operating system is used. Services for inter-partition communication (IPC) are provided by the hypervisor. Not shown are the operating systems within partitions other than NAVDLR.

[20], providing multi-threading, amongst other services, to the application. The real-time performance, space heritage (for example, in the Herschel-Planck [21], Mars Science Laboratory [22] and Eu:CROPIS [23] missions) and availability of source code for evaluation, debugging and modification were important factors in the selection of RTEMS. NAVDLR receives image data from the cameras via IPC requests to the IOSCAM partition, which controls the hardware interface to the camera.

In addition to NAVDLR, the MMX Rover has basic software which provides low-level, mission-independent services. These run in several partitions, such as Mode Management and Data Loading (MMDL). Other partitions contain software for experiments. Other partitions still, such as the Command Control Software (CCSW) carry out mission-specific, critical tasks with overall system responsibility.

4. DLR NAVIGATION SYSTEM

Navigation Pipeline

As already described in Section 2, the main purpose of the DLR navigation system is the evaluation of a collision-free rover navigation towards a waypoint beyond sensor reach. Therefore, a map has to be built from sensor data and the rover has to be localized within the map. The current trajectory of the rover has to be frequently checked for obstacles, using the acquired map. If there is any impassable section, relevant motion commands have to be sent to the locomotion system of the rover.

The main sensor system for the DLR navigation system is the stereo camera pair at the front of the rover (see Section 3). Hence, the navigation system can be modeled as an event-driven pipeline, triggered by the acquisition of new stereo images.

The principal pipeline is illustrated in Figure 6. It provides hazard avoidances and semi-local path planing based on stereo matching and visual odometry, similar to the navigation modules on the MER [24]. With every new stereo image pair, the pipeline starts by calculating a depth image using the Semi-Global Matching (SGM) algorithm [25]. The depth and left camera images are used by visual odometry (VO) to

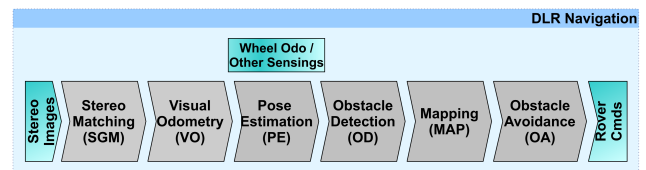


Figure 6. Principal processing pipeline of the DLR navigation system. The pipeline is triggered by newly acquired stereo images and executes consecutively stereo matching, visual odometry, pose estimation, obstacle detection, mapping and obstacle avoidance, resulting eventually in motion commands for the rover. Additional sensor readings are inserted within the pose estimation.

estimate the relative 6D motion. The VO motion estimate can then further be refined with other means of perception, such as the wheel odometry, gyroscope readings, Sun tracker, or an operator input in the pose estimation step (PE), resulting in a drift-reduced relative localization. The depth image is further analyzed for obstacles within the obstacle detection (OD), resulting in a pixel-wise classification. Depth image, obstacle classification and localization information are integrated into a semi-global map within the mapping step (MAP). Finally, the map is used within the obstacle avoidance (OA) to check the current rover path for any impassable sections and to give relevant motion commands to the locomotion system of the rover.

As an optional experiment, a possible slip detection of the wheels should be evaluated. Therefore, the VO and data fusion estimates are compared to the wheel odometry. Larger differences are likely be evidence for slipping wheels.

Navigation System Architecture

As described in Section 3, the DLR navigation system runs on its own partition (NAVDLR) using the RTEMS OS. The navigation system is executed within the OS and accesses system events, sensor data, persistent storage, etc. via inter-partition interfaces (see Figure 5). It can be divided into the following principal modules:

- The navigation pipeline (NavSystem) that contains all components for navigation as well as the necessary control and data flow.
- A pipeline control (NavCtrl) that handles scheduling, data management, pipeline (re-)configuration and failure recovery.
- A system-level control (NavCCSW) that encapsulate the above modules and handles system setup, shutdown as well as health, telemetry logging and tele-command handling

In the following, these modules are further detailed.

Navigation Pipeline—Figure 6 illustrates the principal flow for the navigation pipeline. However, data and control paths are more complicated, update rate, computation time, real-time requirements and embedding into the rover system have to be taken into account. Hence, the navigation pipeline is further detailed as illustrated in Figure 7 (NavSystem part).

In general, stereo matching, visual odometry and local map creation are driven only by camera events, that is calculation is started with every new acquired stereo image. Obstacle avoidance is usually decoupled from map generation and run at a higher rate, in order to check on the rover path more frequently. This decoupling is important for fast rovers,

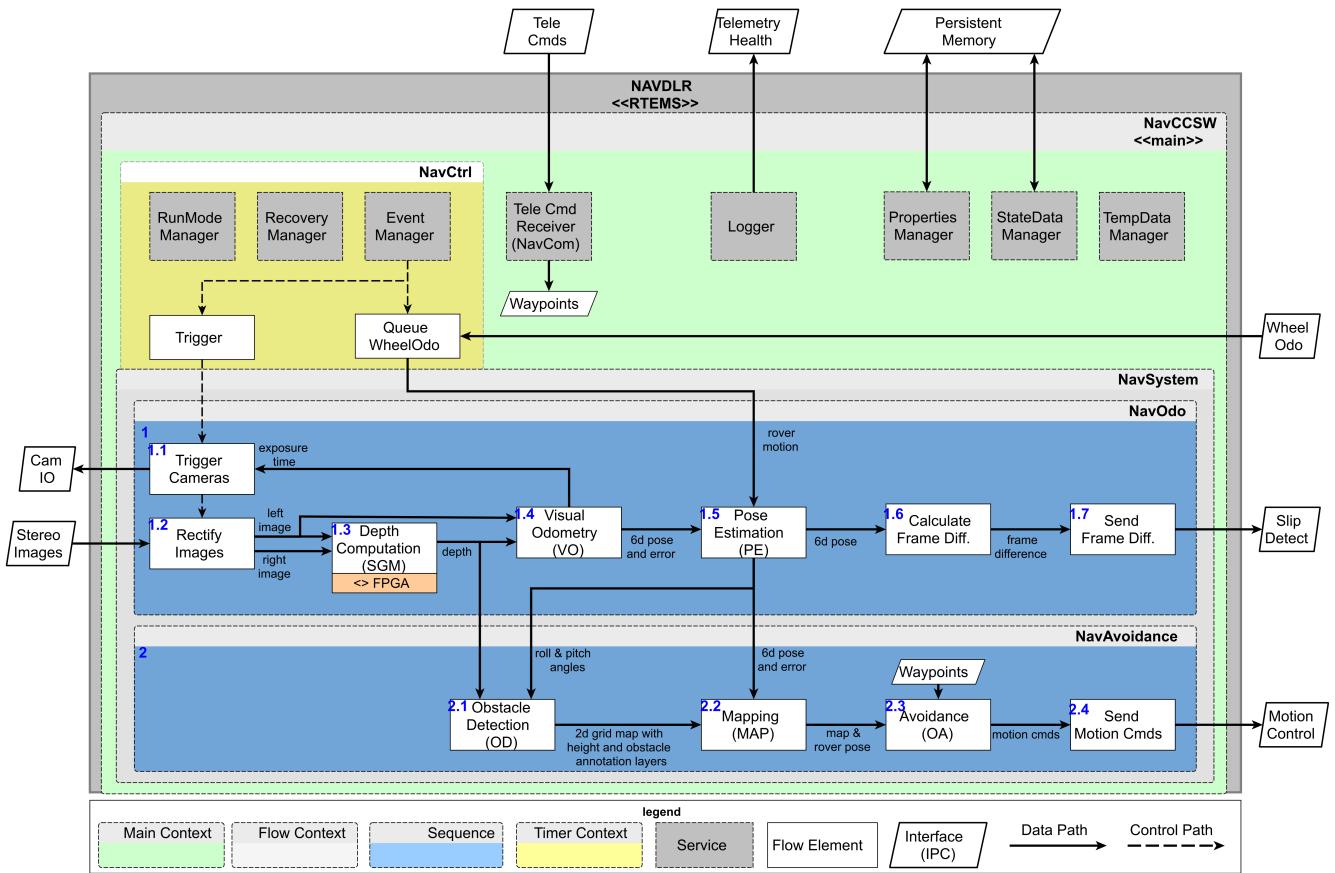


Figure 7. Detailed flow diagram of the DLR navigation solution (NAVDLR) with navigation pipeline (NavSystem) and pipeline control (NavCtrl), embedded into the system-level control (NavCCSW). The pipeline is divided into two sub-sequences, NavOdo and NavAvoidance. The black arrows show the data flow. The sequential control flow is illustrated by the blue numbers in the blocks.

as the frame rate of the camera is low and the calculation time (delay) for the pipeline until map generation is high compared to the movement speed of the system. The MMX Rover, however, has a very low driving speed (see Section 3). Although the frame rate of the cameras is also low (approximately 0.03 Hz), the movement is still small and highly dynamic maneuvers (e.g. tele-operation) are not envisaged. Consequently, for the MMX Rover navigation, the obstacle avoidance can be enqueued in the same thread as the rest of the pipeline, reducing complexity and lowering the overall computation time.

Each pipeline step is modeled as processing block with n inputs and m outputs. The processing blocks are arranged within *container* elements that define the control flow. For pipeline design, different control flow concepts are possible, e.g. sequences, synchronous or asynchronous models. For the navigation pipeline a simple sequence is used. A sequence is a simple flow control that only supports simple feed forward execution of the processing blocks. The processing blocks are executed in a fixed order one after another. Calculated outputs are directly used by subsequent blocks. Hence, data flow is only allowed from blocks that are executed before the receiving block, i.e. feedback loops are not possible. This concept is well suited for blocks with long execution times and low update rates. It also requires less memory, because data outputs do not need buffer swapping.

Furthermore, a *context* is an instance that can be executed in parallel, e.g. as a thread or by manual scheduling². A *flow context* is a *context* that is also a *sequence* and thus is a top-level flow control block for sequences and processing blocks.

Using the above elements, the NavSystem is a flow context that holds the navigation pipeline. Here, the pipeline is further segmented into two sub-sequences, NavOdo that contains the odometry estimation and NavAvoidance that contains mapping and avoidance. The NavOdo sequence comprises the image rectification, depth image computation (SGM), visual odometry (VO) and pose estimation (PE). Further, the difference between visual odometry and wheel odometry is calculated and sent to the locomotion subsystem for detection of wheel slip (see also Section 2). The NavAvoidance sequence comprises obstacle detection (OD), map generation (MAP) and obstacle avoidance (OA). The blue numbers in Figure 7 indicate the control flow or order of the blocks in the sequences. All modules are further detailed in the following sections.

Global and Pipeline Control—The global control (NavCCSW) is responsible for the setup, shutdown and monitoring of the navigation pipeline. It also handles all system events from other partitions and provides health and telemetry in regular intervals. Additionally, it provides abstraction layers

²RTEMS supports multi-threading on the single core partition. Hence, we will likely use the OS-level scheduling rather than manual scheduling.

for OS functions and inter-partition-communication (IPC) to the navigation pipeline (NavSystem) and pipeline control (NavCtrl). Lastly, the global control handles incoming tele-commands.

Pipeline control (NavCtrl) is divided into three aspects: data management, run mode management and FDIR (Fault Detection, Isolation and Recovery).

First, data management is an essential part of system control. It comprises the synchronization of process data with a persistent memory, e.g. for backup, as well as sharing main memory between process parts for the reduction of memory consumption in such small computing platforms. The data used within the processing blocks of the navigation pipeline can be divided into three categories:

- *properties* - The properties are configurable parameters of the processing blocks and the flags that configure the control flow (see state machine below for details). The properties are read-only, i.e. they are only loaded from the persistent memory. See the *tele-operation* paragraph below for details on changing properties.
- *state data* - The state data represents the inner states of the processing blocks, i.e. data that has to be stored between execution steps. This data is usually recovered from persistent memory at system start and backed up regularly to the persistent memory.
- *temporary data* - This is all data that is used within a processing block during its execution but does not need to be stored between two executions. It does not need to be backed up.

Temporary data for each processing block is managed by the corresponding context. Within the sequential flow, the temporary data is used only if a block is active and thus the memory can be shared over all blocks of the sequence, reducing the total amount of memory required. Otherwise, state data of each block and the global properties have to permanently stay in memory. However, their respective management blocks synchronize the data in memory with persistent memory after each iteration via an inter-partition-interface (see Section 3 for details on OBC partitions). This enables recovery of the last state in case of a failure.

The second aspect of the pipeline control is run mode management. For performing the different experiments from Section 2, the active pipeline elements have to be configured and change parameters for the processing blocks during the mission. A certain pipeline configuration is called run mode. They are similar to the *mobility command types* of the MER rover [16] but are tailored for this the mission. The run mode manager controls the run mode and possible transitions to other modes (mode switch). At the moment the following run modes are provided:

- Mode 1 (odometry only) - Only the NavOdo sub-sequence is enabled
- Mode 2 (avoidance, dry-run) - Both NavOdo and NavAvoidance are enabled. However, no motion commands are sent to the locomotion system of the rover but only to telemetry. Instead the rover is operated by setting a goal point via tele-command. The obstacle avoidance can have two different input modes (single image, rolling map) and three different operation modes (emergency stop, steer away, path planning), resulting in six sub-modes.
- Mode 3 (avoidance, operation) - Both NavOdo and NavAvoidance are enabled. Motion commands are generated

by the NavSystem and sent to the rover locomotion subsystem. This mode has the same six sub-modes as mode 2.

The sub-modes are further detailed in the obstacle avoidance section.

The third aspect is *Fault Detection, Isolation and Recovery Techniques (FDIR)*. A sound detection and recovery strategy of possible faults is crucial for operating critical systems. Experiences from earlier projects showed that most failures have an impact on other modules as well and thus can not be recovered within the module. The detection of possible faults is handled inside the respective processing blocks. A fault, e.g. if the VO has not found sufficient features, interrupts the control flow and is signaled to a global *recovery manager*. The recovery manager holds a lookup table of action sets. An action set is a set of actions that are performed in order to restore nominal operation. Hence, for each failure detection from every module there is a corresponding action set. Actions can be module re-configuration, state invalidation, abort of the current processing, etc. As an example, if the VO detects that the current feature set does not match with any key frames, the rover may have moved without NAVDLR being active. Hence, as an action set, the current VO, PE and MAP states have to be reset and the current iteration has to start over.

Tele-Operation—As stated in Section 2, the communication with the rover during mission time is very limited, i.e. long delay, low bitrate, short communication window. Thus, a direct remote-control of the rover, e.g. by joystick from ground station as e.g. in [26], is not possible. Also, a live stream of large amounts of data, like stereo images, depth images or the obstacle map is impossible. Only small regular telemetry data packets and occasional bigger data can be provided with a very long delay (worst case all 24h). Thus, also a *shared navigation support*, as used on the MER rovers [16] is not possible.

Only a next-mission-day style of pre-planning, similar to the MER *human-in-the-loop* approach [16], seems possible. Beyond the communication to other partitions of the rover, the navigation system is able to receive tele-commands that allow for configuring the intended experiments (mission level configuration). Possible tele-commands for the navigation system are:

- **waypoint list** - a list of waypoints, i.e. goal positions relative to the current rover position, that are successively approached.
- **run mode** - a request for switching the run mode of the rover (see NavCtrl above).
- **re-configuration** - a new set of process parameters (properties) that are applied to the modules via property manager.

In the following sub-sections, the main functional blocks of our navigation solution i.e. Stereo Matching (SGM), Visual Odometry (VO), Pose Estimation (PE), Obstacle Detection (OD), Mapping (MAP), Obstacle Avoidance (OA) along with our testing strategy are explained in short.

Stereo Matching (SGM)

Stereo matching is the process of finding the projection of a point of a scenery in a stereo image pair, taken by two side-by-side mounted cameras. Due to the horizontal displacement of the two cameras, the projection of a point is shifted between each stereo image. The closer the object is to the cameras, the higher is the disparity, so there is a direct

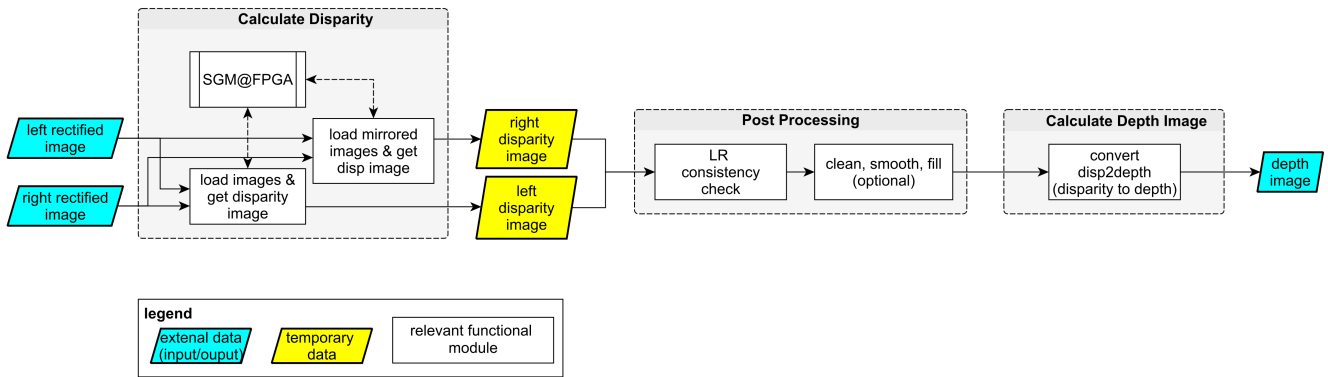


Figure 8. Depth Image Computation Pipeline: Disparity is calculated twice, for normal and for mirrored pair. The results are compared to find invalid regions. After refining the result, distance values are calculated from disparity values.

correlation between distance and disparity. Determining which pixels in both images represent the same point of the scenery is the challenge. Since the displacement is small compared to the distances, the projection of a point and its surrounding area will look very similar in both images. For every pixel the disparity is searched on which a pixel on one image matches a pixel on the other stereo image. The similarity metric is the matching cost function. The more similar the pixels are, the lower is the cost function. Finding similarities for each pixel independently of adjacent pixels is error prone. Lighting changes, occlusions and noise can hide similarities or can create false similarities. There are sophisticated algorithms to mitigate this problem. One of them is Semi-Global Matching (SGM) [17]. The pixels are not examined independently, but the global minimum of the matching costs is searched. Finding this minimum is an NP-complete problem. To still get quick results, the minimum is not searched over the whole image for every pixel, but only over straight paths through the current pixel, or 'semi-global'. Additionally a penalty is introduced to help obtain smooth results. The penalty is added to the costs if the disparity of the current pixel differs from the disparity of adjacent pixels. This reduces outliers because the penalty reduces the chance that the outlying disparity has the minimum cost. However, the penalty must not be so high that it prevents disparity changes on edges. The global optimisation in conjunction with the penalties generates a smooth result with the ability to follow sharp edges. This approach gives good results also on edges, but can still be calculated quickly and efficiently [27]. In this project, SGM was chosen because it is a good trade-off between resource requirements and quality of the results [17].

One important property of the SGM algorithm is its ability to detect invalid results. On Phobos there is no significant atmosphere. The lack of atmospheric scattering leads to challenging harsh light conditions like completely black sky and shadows. These black, homogeneous areas are difficult to handle for depth image calculation, since it relies on comparing features to find corresponding pixels in the stereo image pairs. In homogeneous areas this is impossible, since there are no differences. Although, it is not possible to get correct depth values from inside such regions, the SGM algorithm is able to declare those results as invalid. This is achieved in two separate ways. The first is to calculate a confidence value. The confidence value is the gradient of the costs around the minimum. If the gradient is high, the confidence is high that the minimum is not on one of the adjacent pixels, because they have much higher costs. In

black areas the matching costs will constantly be low, so the gradient and the confidence will be low. The second approach is to calculate the disparity for both images. First the left image is used as the base image, then SGM is processed again with swapped input images. If the disparity in both steps matches, then the result is probably valid, otherwise the result is invalid. This approach, that we call as Left-Right (LR) consistency check, also finds occlusions, which occur when parts of the background are occluded by a closer object and can only be seen from one camera.

The whole disparity calculation flow is shown in Figure 8. The disparity calculation by Semi-Global Matching is calculated for the stereo image pair and the mirrored stereo image pair. Both results are compared to find invalid areas in the disparity image. The disparity can further be refined to remove noise or outliers, for example, before disparity values are converted into distance values in the final step.

The SGM algorithm is implemented in the FPGA of the Zynq-7045 system-on-chip. An FPGA can process much data in parallel. The FPGA implementation can be optimized for calculating necessary data like matching costs (e.g. from eight adjacent pixels) in a single clock cycle. Additionally, the FPGA has many (up to hundreds of) memory cells which can be accessed in parallel. Instead of reading and writing temporary data from and into an external memory through a single memory interface, temporary data can be stored in multiple independent RAM cells with independent memory interfaces. This significantly increases throughput, because data can be accessed as quickly it can be processed and avoids the bottleneck of a single interface.

Visual Odometry (VO)

In order to navigate autonomously on Phobos, the rover must be able to localize itself in its local environment. In many terrestrial applications Global Navigation Satellite System (GNSS) is used to solve this task, however such systems are not available in space. Therefore, another approach has to be used. One solution, which has been applied more often in recent years, and is also used for the MMX project, is the localization based on visual data provided by a camera system, which is referred to as visual odometry (VO). Using a visual odometry has been successfully applied in previous planetary robotic missions [28], [29]. The task of the VO module is to calculate the 3D translation and orientation (6D pose) of a camera within an environment by means of visual

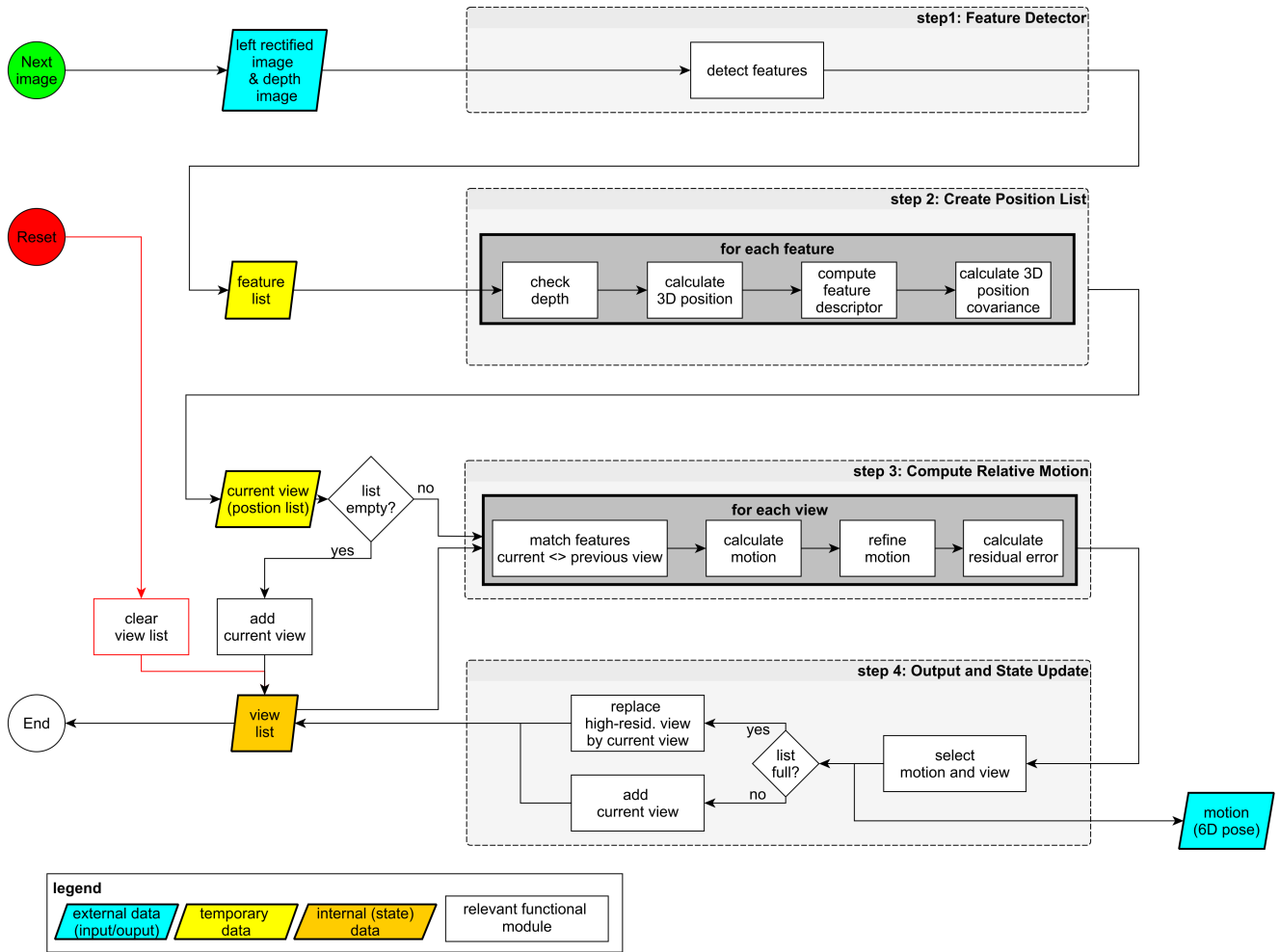


Figure 9. Flow Diagram of Visual Odometry Pipeline. The VO uses the left camera image and the corresponding depth image as input. In step 1, image features are detected in the provided image. Step 2 creates a list of valid image features and calculates their 3D position. Current features and previous features are matched and used to estimate the current pose in step 3. In step 4, internal states are updated and the results of the VO is prepared for the output.

information provided by the camera images. Therefore, a lack of features due to very low intensity gradients will cause VO to fail. Also, images that contain features only in a small region will result in larger errors of the estimated motion. To be able to calculate a metric scale for the pose, we employ our stereo-VO for this project [25], [12], [11], illustrated in detail in Figure 9. This method was successfully tested and evaluated in multiple missions [15], [30], variety of robots, roving [31] and flying [32], and in a variety of different environments. Therefore, the input of our VO is the image of one camera and the corresponding depth image, calculated by the "Stereo Matching" module. To estimate the motion between image frames, the VO uses corner features to identify distinctive image points in one image frame, which can also be found in the other image frame. With that list of corresponding image features the module can calculate the relative pose. Furthermore, an estimated pose error in the form of a covariance matrix can be provided. In order to increase the accuracy and robustness of the pose estimation, the VO uses keyframes (labeled 'views' in Figure 9). In this approach, the pose is usually not calculated with respect to the most recent image frame, but to one that may lie further back in time, which has many advantages: It can achieve

higher accuracy due to lower accumulated drift, in particular if motion is small between consecutive image frames and is more robust with respect to tracking losses as the best keyframe can be chosen from a set of keyframes. Our VO keeps a list of keyframes of fixed maximum size. The motion of the current image frame to each keyframe is calculated and the one that yields the smallest residual error of matching 3D feature points is selected. In addition, the current image frame is added to the list of keyframes and the keyframe with the highest residual error is removed if the maximum number of keyframes has already been reached.

Pose Estimation (PE)

The PE component integrates inputs from one or multiple sensors measuring the rover's egomotion into a joint estimation of the rover's pose. Figure 10 shows the pose estimation's data flow model.

The GNSS, a sensor due to its qualities oftentimes considered to be the primary navigation sensor in terrestrial applications, is out of question, as discussed in the VO section. Visual Odometry is a dead reckoning means of navigation, meaning that the uncertainty of the estimation necessarily grows over

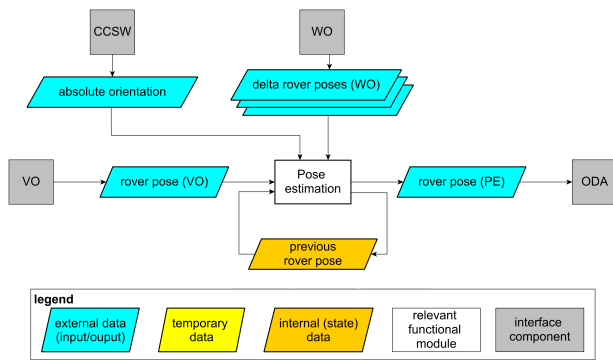


Figure 10. Pose estimation component data flow model. Visual odometry is the primary input for the pose estimation. Wheel odometry might be, according to the settings, fused in, discarded or compared against the complementary inputs to estimate the wheel slip. An occasional absolute orientation input from the superordinate Command-and-control software partition would be beneficial. Here, ODA refers to the combination of Obstacle Detection and Obstacle Avoidance modules which are explained in the following sub-sections.

the time. Furthermore, it may perform suboptimally in numerous cases of challenging visual conditions. Despite all its limitations, this application considers the VO to be its primary sensoric input.

Wheel odometry, that is estimation of system's egomotion obtained as the readings of the wheel revolution encoders, shall be available. It tends to be particularly reliable in the automotive industry, where the contact between the rubber tire wheels and asphalt road is particularly good. In contrast with that, our experience with wheel-based ground robotics in an environment with loose terrain shows, that the Wheel odometry tends to be very unreliable in cases when a large fraction of wheel revolutions goes into the wheel slippage rather than into translatory motion forwards. Due to lacking knowledge of the Phobos' environment upfront, our design will allow multiple ways of treating the Wheel odometry inputs:

- In case of trustworthy measurements in a low-slip environment, WO can be considered as a valid sensoric input.
- In case of untrustworthy measurements but in environments where the risk imposed by wheel slippage is not very high, Wheel odometry inputs can be chosen to be discarded.
- In case of needing to operate in environments with high wheel slippage posing a risk to the mission due to the possibility of self-burying, we may be asked to provide a hazard avoidance service by comparing the measured WO against other means of egomotion estimation, in order to estimate wheel slip as the difference of those two. According to Phobos ERD (Environment Requirements Document) [33], Phobos' surface is expected to be mostly covered with fine, loose and low-density regolith. That, in combination with the skid-steered character of rover's locomotion, makes this high wheel slip scenario a likely one. We made an experience with the LRU on Etna, that a wheel slippage translates into a jerky motion of the rover body, which can be observed in the accelerometers of the IMU [34]. When it does, it can be detected rapidly, adapting the rover driving speed accordingly, which can even resolve the problem. However, as the MMX Rover will likely have no access to accelerometer readings, the only viable option of estimating the wheel slip seems to

be the comparison between the WO and the VO.

There seem to be, additionally, two feasible ways to occasionally estimate absolute orientation of the rover - its roll, pitch and yaw in the planetary coordinate system. The first one would be the rover's onboard capability of Attitude sensing, which in itself would be a fusion of gyroscope measurements and the visual Sun sensing. The second one would be to get a 3D map of the environment with the rover in it, possibly obtained from the MMX orbiter. While neither of these options is at the moment certain to be available for us, given how important it is for the rover to know its roll and pitch with respect to the gravity vector in order to detect terrain hazards and find drivable paths reliably, our design remains open to occasionally receiving the absolute orientation of the rover as an input from the superordinate Command-and-Control software.

In terrestrial applications, the PE is usually designed as a predictor-corrector, having both a way to predict system state based upon current knowledge and to update it by newly-arriving information from the constituent sensors. This design allows for asynchronous arrival of measurements and frequent state polling, and is thus especially advantageous for multi-sensor highly-dynamic robots, such as e.g. the flying ones. In contrast with those, our system is slower in general, doesn't need to respond that quickly to environmental hazards, and the robustness and formal verifiability of implementation seem to be of greater importance. Our design thus considers single update-and-decide passes. These are steered by the availability of new stereo camera image. This is first processed up to the VO step. VO, together with a queue of unprocessed WO measurements (if applicable), is used to update the rover state. Still on the same pass, the environment awareness and hazard avoidance parts of the pipeline follow.

Obstacle Detection (OD)

In order to perform autonomous and safe navigation in an unknown environment on Phobos, the rover needs to know the obstacles in its path. Using this environment and obstacle map, the rover can avoid obstacles as well as plan a safe path to navigate. This is done by the obstacle detection, mapping and avoidance modules. Due to the limited memory and computation available onboard, these algorithms need to be computationally and memory efficient.

The obstacle detection module detects the obstacles using the rover's perception data acquired through its sensors. As mentioned before, MMX NavCAM is the primary sensor used for navigation and the depth image computed from the NavCAM is used as the primary sensor input for obstacle detection.

The obstacle mapping solution being developed for this purpose is based on the existing obstacle detection module [13] that is being used on the Lightweight Rover Unit (LRU) [35], a space rover prototype that was developed at the Institute of Robotics and Mechatronics, DLR. The solution will be adapted and reimplemented considering the Phobos environment, mechanical and sensor setup of the MMX Rover as well as the power, computational and memory constraints.

There are three main components that drive the design of the obstacle detection and avoidance solution. They are the environment in which the rover has to navigate, the rover and its locomotion capabilities and the perception sensor. Phobos is still unexplored in this sense as no certain information is available on the terrain of Phobos, at least at a

resolution desired by the navigation solution of a rover. So, we consider the likely cases of the terrain as defined in the Phobos ERD [33] for the MMX Rover mission. Secondly, the size and the locomotion mechanism of the rover define what makes an obstacle in the environment. Lastly, the perception sensors define what data is available in order to detect these obstacles.

The environment features that are considered as obstacles are the ones that are usually dangerous for a mobile robot in an unstructured environment. These are rocks bigger than a specific size, slopes, and negative edges. Considering all these aspects, we consider the following as obstacles:

- rocks: From preliminary locomotion analysis studies, we consider
 - rocks of height less than 1/4 the wheel radius (30mm) as not being obstacles.
 - rocks of height less than 1/2 wheel radius (55mm) as low risk obstacles.
 - rocks of height less than wheel radius (110mm) as high risk obstacles.
 - rocks of height greater than wheel radius (110mm) as impassable obstacles.
- steep slopes: From some preliminary locomotion analysis studies, we consider
 - slopes less than 10° are safe to navigate.
 - slopes less than 20° are with some risk.
 - slopes greater than 20° are high risk obstacles.
- negative edges like cliffs or edge of a crater as impassable obstacles.

The obstacle detection module takes in the depth map as input and detects obstacles. Using the depth map and camera calibration parameters, 3D points are calculated corresponding to the pixels in the depth map. This information is used to detect the predefined classes of obstacles. The output of the module is a 2.5D elevation map with annotated obstacle traversability classification results. The processing of the stereo data to produce a grid-based local traversability map is similar to the Grid-based Estimation of Surface Traversability Applied to Local Terrain (GESTALT) system used on Mars Exploration Rovers [36], [16]. Figure 11 shows the functional block diagram of the whole Obstacle Detection and Obstacle Avoidance modules along with their input and output data.

Mapping (MAP)

The mapping module takes 2.5D elevation maps with obstacle traversability classification results (from OD) as well as the respective 6D robot pose estimates (from PE) as input. Its output is a local map to be used by the extended version of the obstacle avoidance (OA) module.

In order to create this local map, the mapping module aggregates the single-shot maps from OD along the rover trajectory estimated by PE into a 2.5D map with traversability classification annotations, similar to the local map described in [14, Chapter 5.1]. For this, in each step, the single-shot map from the OD module is first transformed into the MAP coordinate frame based on the rover's respective pose. It is then merged with the existing map data. The result is a so-called rolling map, which has a fixed maximum size. It needs to get periodically re-centered with respect to the robot and pruned to stay within its bounds. Figure 12 shows an overview of the data flow in the MAP module, including the transformations between the start of the pose estimation (PE), the map frame, and the robot frame. Splitting the transformation between the PE frame and the robot frame

into two parts, i.e., PE frame to map frame and map frame to robot frame, allows to decouple the frequency of map updates and map recenter operations from the robot movements and thus gives more options to save computational resources if necessary.

The resolution and size of the map result from a trade-off between computational effort, memory consumption and the accuracy and extent of the map needed for OA. While the exact values still need to be determined, in this early design phase we consider grid map resolutions of approx. 1-5 cm and sizes of at least $4\text{ m} \times 4\text{ m}$. The choice of a grid-based map representation is motivated by our successful evaluation in terrestrial tests with a space exploration rover prototype, including autonomous navigation at a Moon-analogue site on the volcano Mt. Etna [14], [30]. It allows for a straightforward integration of the results from the OD module while filtering noise over time, e.g., via a low-pass filter for data integration. The fixed size of the map helps to ensure that the computational effort in each update step is bounded.

Obstacle Avoidance (OA)

Taking the obstacle maps from OD and MAP, the rover obstacle avoidance module generates path/motion commands to avoid the obstacles therein and ensure a safe navigation. Four levels of obstacle avoidance are considered, at an increasing level of functionality but also complexity, dependencies on other modules, and requirements for computational resources (see Figure 11).

- *Reactive Behavior*
 - *Emergency Stop*: Rover stops movement in case an obstacle is detected directly in front of it or in its commanded path.
 - *Steer Away*: Rover turns in case a obstacle is detected directly in front of it or in its path; the direction to turn could be the one with the smallest number of obstacles or the one closer to its original goal.
- *Path Planning*
 - *Path Planning on Single Image*: Rover plans a (short) path to avoid local obstacles based on the obstacle map from a single depth image.
 - *Path Planning on (Rolling) Map*: Rover plans a (potentially longer) path to avoid obstacles based on the local rolling obstacle map.

Testing

The testing of NAVDLR intends to answer several questions. In general, testing in our case is a multi-dimensional task that has to consider the aspects of hardware and software integration, environmental conditions of Phobos, and different means of testing – experimental, dataset based, or simulative testing. As it is the adaption of the DLR-RM navigation pipeline that is used on our planetary rover prototypes [31] and similarly on our MAVs [32][11], we already have a baseline regarding robustness and accuracy to compare it against. The questions to be answered by testing are:

- Does NAVDLR as a reimplement of DLR-RM Nav provide the same accuracy in a known environment?
- Is NAVDLR as robust to environmental disturbances as DLR-RM Nav?
- Are the individual software components and the overall navigation pipeline of NAVDLR robust with respect to the expected environmental conditions on Phobos?
- Can NAVDLR be correctly integrated with the MMX Rover OBC and how is the software's performance on it?

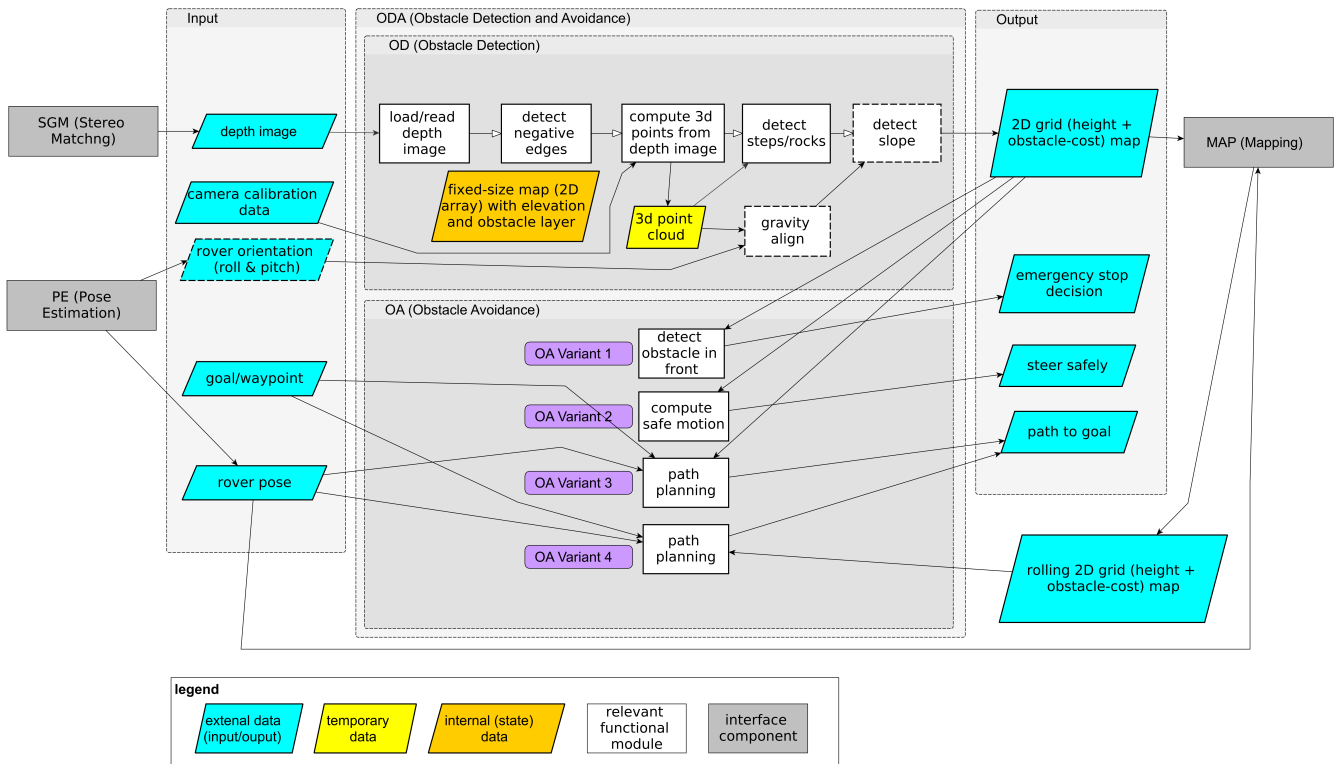


Figure 11. Obstacle Detection and Obstacle Avoidance. The OD module takes depth image and rover orientation as input and computes the 2D grid map with height and annotated obstacle costs. The Mapping module aggregates these local obstacle and height maps to rolling/global larger maps. The OA module takes the obstacle avoidance maps and provides safe navigation features like emergency stop, reactive steering away from obstacles as well as safe paths for autonomously navigating to a goal location. The reliable slope estimation depends on the availability of accurate roll and pitch of the rover in the beginning. At the moment, quality of the estimation of this on a low-gravity body like Phobos considering the MMX rover sensor and power constraints is unclear.

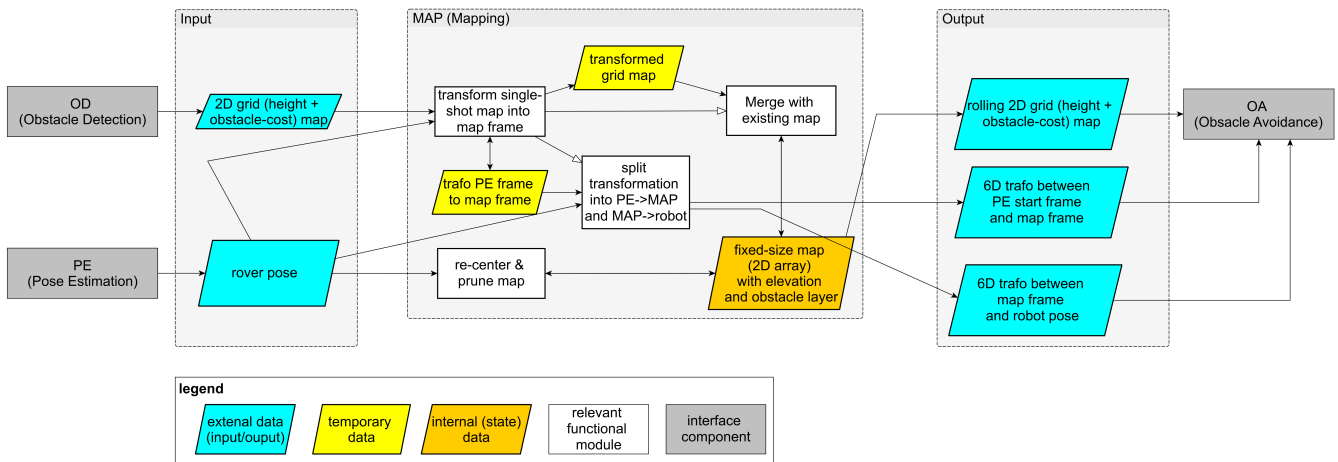


Figure 12. Mapping component data flow model. The MAP module takes grid maps created from single stereo image pairs from the OD module and integrates them along the robot trajectory estimated by PE. It creates a rolling map of a fixed size that is periodically re-centered w.r.t. the rover's pose. This map with elevation and obstacle annotation layers will then be used for local path planning in the extended version of the OA module.

- Does the interfacing between the NAVDLR partition and other MMX Rover software components work correctly?

Regarding the complexity of the tests, we define several test levels as:

T0: **Component Verification** - Unit testing and basic functionality tests. Artificially created input signals and dataset-based inputs are used to test basic functionalities and interfacing between components.

T1: **Component Validation** - Test of individual component performance under ideal operation conditions (i.e. feature

rich soil with strong illumination that additionally features diffuse lighting) with the DLR-RM Nav as reference baseline. **T2: Component Testing for Expected and Corner-Case Phobos Conditions** - Test of individual component performance under challenging operation conditions, i.e. the expected conditions on Phobos. We consider each component individually with ideal input from previous components in the pipeline.

T3: NAVDLR Pipeline Validation - Test of the performance of all software components combined into a pipeline, under ideal operation conditions. The DLR-RM Nav performance and other state of the art navigation algorithms can be used as baseline.

T4: NAVDLR Pipeline Testing for Expected and Corner-Case Phobos Conditions - We consider similar tests as T2 with the challenging Phobos environment conditions and beyond, but in this case with the complete NAVDLR pipeline.

Regarding the hardware, we try to implement an agile, parallelized testing approach that addresses all these questions. It is understood that the final hardware components and the final MMX Rover OBC setup will only be available to us for testing late in the project. Therefore, other means of testing will be exploited early on. Figure 14 shows the stages of hardware and software integration, with the possibility of parallelization in testing.

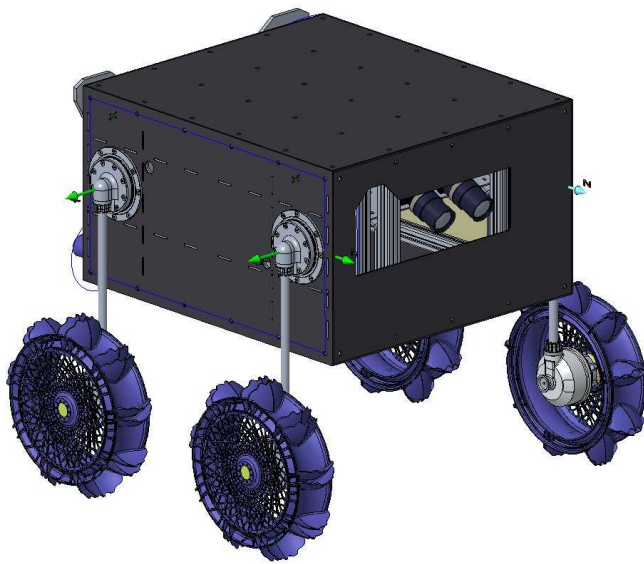


Figure 13. Design of the MMX Rover mock-up for testing on Mt. Etna. It features the representative locomotion subsystem and a representative COTS camera system for experimenting and data recording.

All software components can initially be tested independent of the MMX Rover system, a testing step that we call Software-in-the-Loop (SiL). There, we run the algorithms on generic Linux host systems to focus mostly on unit testing, the navigation accuracy of the algorithms, as well as its robustness regarding the Phobos environment conditions. SiL is therefore used in all test level, from T0 to T4.

To have a representative hardware setup at early stages of the project, we set up an *agile testing track*. Agile refers to the fact that we will build an exclusively NAVDLR internal testing platform, that features similar specifications as the final flight system but uses commercial of-the-shelf (COTS) components. These COTS components feature a similar Zynq

Processor with FPGA, cameras with the same sensor chip and camera lenses that provide a similar field of view. There, we first run tests with the representative processor (Processor in the Loop - PiL) to test the algorithm's compatibility with the respective processor architecture. Later on, we add peripherals such as representative cameras (Hardware in the Loop - HiL), and finally the same hardware with the Xtratum Hypervisor as so-called System in the Loop (SYSiL). The SYSiL tests are of special importance as we mimic there the other partitions to test NAVDLR for interfacing and inter-partition communication. The test levels for the agile track are T0-T4, with a focus on the T0 and T1 test levels.

Later in the project, the original hardware and software for the MMX Rover is expected to become available. There, we use the same incremental testing philosophy starting at PiL and continuing via HiL until the SYSiL, this time referred to as the *flight track*. Due to the expected limited test time on the flight track systems, we consider in that case mostly integration tests, i.e. test level T0.

As we use continuous integration for our software development, unit tests are continuously executed at SiL and Agile-PiL levels.

The NAVDLR solution is not only to be tested with the respective hardware but also with respect to the Phobos environment. From our experience with the DLR-RM Nav software over the recent years, we can define an environment where we can assure good navigation performance of the already existing DLR-RM Nav solution. These *ideal* operation conditions feature

- soil that is composed of coarse grains (at least 2-5 mm grain size);
- rocks that are distributed throughout the field of view of the rover;
- illumination with not only direct but also partially diffuse illumination;
- a static scene.

These conditions are used as first step to comparatively evaluate the NAVDLR performance on T1 and T3 levels, before we start considering more challenging cases based on the expected Phobos environment.

There is a high uncertainty about the surface conditions on Phobos, as the highest available image data is at a maximum resolution of 1.5 m/px [33]. However, [33] provides estimates of the surface conditions as a reference for development by using available data and by analyzing similar celestial bodies, such as Earth's Moon.

Considering the information in [33], we identify several risks that are imposed on our navigation pipeline by the specific environmental conditions on Phobos. These risks shall be considered for testing with incrementally more challenging settings and it should be determined at which threshold the navigation component fails.

The following environment related challenges for NAVDLR are reflected in tests on component level (T2 test level) and in tests on navigation pipeline level (T4 test level):

- Shadows: Phobos is tidally locked with Mars and therefore has a rotation with respect to the Sun equal to its orbital period $T_{orb} = 27566$ s. We therefore expect shadows on the surface

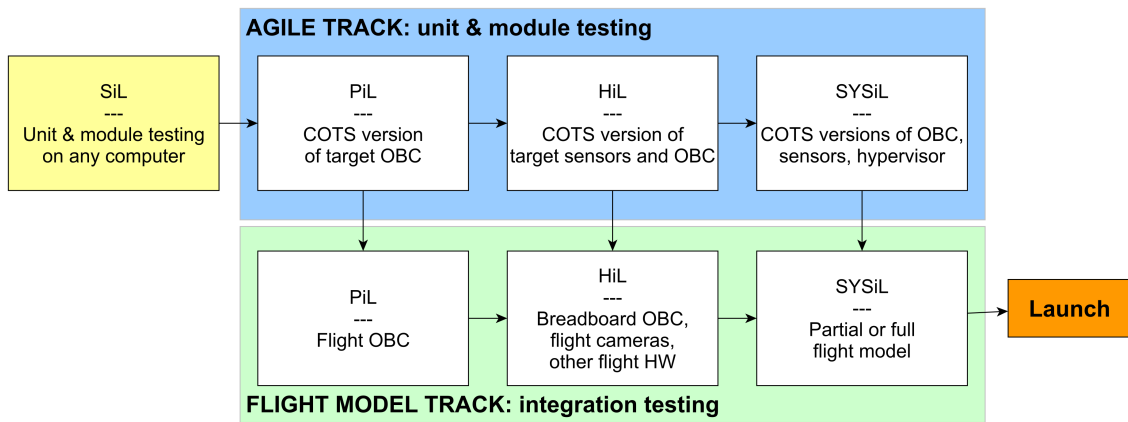


Figure 14. Test Tracks - we feature two different test tracks. The agile track aims at providing a representative testbed for early testing and the flight track represents the integration into the target hardware.

of Phobos that move with a similar angular velocity:

$$\omega_{\text{shadows}} \approx \omega_{\text{phobos}} = 360^\circ / T_{\text{orb}} = 47.01^\circ / \text{h} = 0.013^\circ / \text{s}$$

The resulting linear velocities of the shadows on the ground can violate the assumption of a static scene. We need to test the impact that these shadows have on the visual odometry. In case that shadow-induced position drift occurs, the software needs to be adapted for higher robustness. For testing, the moving shadows will be emulated using a strong and mobile artificial light source.

- **Illumination:** Furthermore, direct sunlight reaches the surface of Phobos unfiltered and diffuse illumination is limited [33]. We therefore test the navigation algorithms for robustness with respect to image saturation and underexposure of areas of the camera images due to the intense illumination conditions expected on Phobos.

- **Rock distribution and grain size:** There is a high uncertainty on the distribution and size of rocks and grains on Phobos. As visual features are mostly found on non-smooth surfaces, a lack of visible grains and rocks presents a risk for navigation. Using the upper and lower bound for rock and grain distribution on Phobos as specified in Phobos ERD [33], we expect between 50 – 1500 objects of sufficient size such that they are visible in the camera image. Note that close to the cameras, even small grains provide useful features for navigation, whereas further away, only bigger stones are relevant. Observing a terrain with lower bound features can cause the VO to fail, as not enough useful (and evenly distributed) features are detected in the scene.

For testing, we consider several sources for representative sensor input. First of all, we consider on-site experimentation in the DLR-RM Planetary Exploration Laboratory, which features a 5 m × 10 m area of soil for driving and a high precision ground truth tracking system. It will further be modified to provide a mobile illumination source.

Furthermore, we plan to participate as a side-experiment in the 2021 ARCHES field test [30] on Mt. Etna in Italy to perform testing experiments in a Moon analogue location [37] using a representative MMX Rover mock-up, as shown in Figure 13. During all on-site experiments, datasets can be recorded for continuous off-site testing of NAVDLR.

As an additional source of planetary-analogue data, we consider the MADMAX dataset [38]. It provides data from a Mars-analogue scenario in the desert of Morocco that does

not fit the conditions of Phobos precisely, but features long navigation trajectories and results from state-of-the-art algorithms as benchmarks.

Finally, we consider a high-level simulation of the MMX rover to create depth images as input data which can be used for open-loop testing of the NAVDLR obstacle detection and mapping. This simulation is a specific redesign of [39] for the MMX Rover. It allows the creation of Phobos-specific terrain with the MMX Rover model inside, and enables rendering of the camera sensor input together with the corresponding depth images. The simulation additionally allows steering of the MMX Rover in the virtual environment, allowing testing of the obstacle avoidance in a closed loop simulation.

5. CONCLUSION

This paper presented an introduction to the DLR Autonomous Navigation Experiment that is planned to be conducted with the MMX Rover on Phobos. The rover will fly as a payload on the MMX spacecraft that is scheduled to launch in the third quarter of 2024. The rover will help to understand the surface properties of Phobos for a safe landing of the lander later. In addition to this, it will gather scientific data and act as a technology demonstrator. In this context, DLR is preparing to perform an autonomous navigation experiment to showcase the challenges as well as advantages of autonomous navigation for planetary exploration. We presented a short summary of our experiment objectives as well as an initial design of our solution for this problem considering the limitations resulting from the low gravity unknown environment, small rover, limited electrical and computing powers as well as the large communication delays which make teleoperation infeasible.

ACKNOWLEDGMENTS

The MMX Rover is a joint contribution of DLR and CNES and we would like to thank all the people involved in this challenging endeavor. In addition, we would also like to thank all the people coming from different agencies, institutes and countries involved in the MMX mission lead by JAXA and for providing us an opportunity to be part of this exciting mission.

REFERENCES

- [1] S. R. Taylor, C. M. Pieters, and G. J. MacPherson, "Earth-Moon System, Planetary Science, and Lessons Learned," *Reviews in Mineralogy and Geochemistry*, vol. 60, no. 1, pp. 657–704, 2006.
- [2] J. Zhang, N. Dauphas, A. M. Davis, I. Leya, and A. Fedkin, "The proto-Earth as a significant source of lunar material," *Nature Geoscience*, vol. 5, no. 4, pp. 251–255, 2012.
- [3] T. C. Duxbury, A. V. Zakharov, H. Hoffmann, and E. A. Guinness, "Spacecraft exploration of Phobos and Deimos," *Planetary and Space Science*, vol. 102, pp. 9–17, 2014.
- [4] M. Pajola, M. Lazzarin, C. Dalle Ore, D. Cruikshank, T. Roush, S. Magrin, I. Bertini, F. La Forgia, and C. Barbieri, "Phobos as a D-type captured asteroid, spectral modeling from 0.25 to 4.0 μm ," *The Astrophysical Journal*, vol. 777, no. 2, p. 127, 2013.
- [5] M. Pajola, M. Lazzarin, I. Bertini, F. Marzari, D. Turri, S. Magrin, F. La Forgia, N. Thomas, M. Küppers, R. Moissl, F. Ferri, C. Barbieri, H. Rickman, and H. Sierks, "Spectrophotometric investigation of Phobos with the Rosetta OSIRIS-NAC camera and implications for its collisional capture," *Monthly Notices of the Royal Astronomical Society*, vol. 427, no. 4, pp. 3230–3243, 2012.
- [6] S. Murchie, N. Thomas, D. Britt, K. Herkenhoff, and J. F. Bell III, "Mars Pathfinder spectral measurements of Phobos and Deimos: Comparison with previous data," *Journal of Geophysical Research: Planets*, vol. 104, no. E4, pp. 9069–9079.
- [7] K. Kuramoto, Y. Kawakatsu, and M. Fujimoto, "Martian Moons eXploration (MMX) : an overview of its science," in *European Planetary Science Congress*, Sep. 2018, pp. EPSC2018–1036.
- [8] T. Usui, K.-i. Bajo, W. Fujiya, Y. Furukawa, M. Koike, Y. N. Miura, H. Sugahara, S. Tachibana, Y. Takano, and K. Kuramoto, "The Importance of Phobos Sample Return for Understanding the Mars-Moon System," *Space Science Reviews*, vol. 216, p. 49, 2020.
- [9] S. Ulamec, P. Michel, M. Grott, U. Böttger, H.-W. Hübers, N. Murdoch, P. Vernazza, K. Özgür, J. Knollenberg, K. Willner, M. Grebenstein, S. Mary, P. Chazalnoel, J. Biele, C. Krause, T.-M. Ho, C. Lange, J. T. Grundmann, K. Sasaki, M. Maibaum, O. Küchemann, J. Reill, M. Chalon, S. Barthelmes, R. Lichtenheldt, R. Krenn, M. Smířek, J. Bertrand, A. Moussi, C. Delmas, S. Tardivel, D. Arrat, F. Ijpelaan, L. Melac, L. Lorda, E. Remeteau, M. Lange, O. Mierheim, S. Reershemius, T. Usui, M. Matsuoko, T. Nakamura, K. Wada, H. Miyamoto, K. Kuramoto, J. LeMaitre, G. Mas, M. Delpéch, L. Celine, A. Raffleau, H. Boirad, R. Schmitter, C. Virmondois, C. Cenac-Morthe, A. Besson, and F. Rull, "A rover for the JAXA MMX Mission to Phobos," in *70th International Astronautical Congress*, 2019.
- [10] J. Bertrand, S. Tardivel, F. Ijpelaan, E. Remeteau, A. Torres, S. Mary, M. Chalon, F. Buse, T. Obermeier, M. Smířek, A. Wedler, J. Reill, and M. Grebenstein, "Roving on Phobos: Challenges of the MMX rover for Space Robotics," in *15th Symposium on Advanced Space Technologies in Robotics and Automation*, 2019.
- [11] M. G. Müller, F. Steidle, M. J. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomić, and W. Stürzl, "Robust Visual-Inertial State Estimation with Multiple Odometries and Efficient Mapping on an MAV with Ultra-Wide FOV Stereo Vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.
- [12] A. Stelzer, H. Hirschmüller, and M. Görner, "Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 381–402, 2012.
- [13] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa, "Stereo-vision based obstacle mapping for indoor/outdoor SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1846–1853.
- [14] M. J. Schuster, "Collaborative Localization and Mapping for Autonomous Planetary Exploration: Distributed Stereo Vision-Based 6D SLAM in GNSS-Denied Environments," Ph.D. dissertation, University of Bremen, 2019.
- [15] A. Wedler, M. Vayugundla, H. Lehner, P. Lehner, M. J. Schuster, S. G. Brunner, W. Stürzl, A. Dömel, H. Gmeiner, B. Vodermayr, B. Rebele, I. L. Grixia, K. Bussmann, J. Reill, B. Willberg, A. Maier, P. Meusel, F. Steidle, M. Smisek, M. Hellerer, M. Knapmeyer, F. Sohl, A. Heffels, L. Witte, C. Lange, R. Rosta, N. Toth, S. Völk, A. Kimpe, P. Kyr, and M. Wilde, "First results of the robex analogue mission campaign: Robotic deployment of seismic networks for future lunar missions," in *68th International Astronautical Congress (IAC)*, ser. 68th International Astronautical Congress (IAC), vol. 68. International Astronautical Federation (IAF), September 2017. [Online]. Available: <https://elib.dlr.de/118237/>
- [16] M. Maimone, J. Biesiadecki, E. Tunstel, Y. Cheng, and C. Leger, "Surface navigation and mobility intelligence on the Mars Exploration Rovers," *Intelligence for space robotics*, 01 2006.
- [17] H. Hirschmüller, "Stereo Processing by Semi-Global Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.
- [18] H. Sedlmayr, S. Barthelmes, R. Bayer, W. Bertleff, M. Bihler, F. Buse, M. Chalon, D. Franke, F. Ginner, V. Langofer, R. Lichtenheldt, T. Obermeier, A. Pignède, J. Reill, J. Skibbe, and S. Tardivel, "MMX - Development of a Rover Locomotion System for Phobos," in *IEEE Aerospace Conference*, 2020, pp. 1–10.
- [19] A. I. Activites, "Avionics Application Software Standard Interface, Part 0, Overview of ARINC 653," *ARINC Specification 653P0-2*, 2019.
- [20] RTEMS. URL: <https://www.rtems.org/>.
- [21] M. Mikučionis, K. G. Larsen, J. I. Rasmussen, B. Nielsen, A. Skou, S. U. Palm, J. S. Pedersen, and P. Hougaard, "Schedulability analysis using uppaal: Herschel-planck case study," in *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*. Springer, 2010, pp. 175–190.
- [22] P. Burkhart, T. Ely, C. Duncan, E. Lightsey, T. Campbell, and A. Mogensen, "Real-Time EDL Navigation Performance Using Spacecraft to Spacecraft Radiometric Data," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6566.

- [23] F. Dannemann and F. Greif, "Software Platform of the DLR Compact Satellite Series," in *Proceedings of 4S Symposium*, 2014.
- [24] M. Maimone, P. Leger, and J. Biesiadecki, "Overview of the Mars Exploration Rovers' Autonomous Mobility and Vision Capabilities," 2007.
- [25] H. Hirschmüller, P. R. Innocent, and J. M. Garibaldi, "Fast, Unconstrained Camera Motion Estimation from Stereo without Tracking and Robust Statistics," in *7th International Conference on Control, Automation, Robotics and Vision*, 2002, pp. 1099–1104.
- [26] P. Schmaus, D. Leidner, R. Bayer, B. Pleintinger, T. Krüger, and N. Y. Lii, "Continued Advances in Supervised Autonomy User Interface Design for METERON SUPVIS Justin," in *IEEE Aerospace Conference*. IEEE Computer Society, March 2019.
- [27] H. Hirschmüller, "Memory Efficient Semi-Global Matching," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I-3, pp. 371–375, 2012.
- [28] Yang Cheng, Mark Maimone, and Larry Matthies, "Visual odometry on the mars exploration rovers," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2005, pp. 903–910 Vol. 1.
- [29] A. E. Johnson, S. B. Goldberg, Yang Cheng, and L. H. Matthies, "Robust and efficient stereo feature tracking for visual odometry," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 39–46.
- [30] M. J. Schuster, M. G. Müller, S. G. Brunner, H. Lehner, P. Lehner, R. Sakagami, A. Dömel, L. Meyer, B. Vordermayer, R. Giubilato, M. Vayugundla, J. Reill, F. Steidle, I. von Bargen, K. Bussmann, R. Belder, P. Lutz, W. Stürzl, M. Smíšek, M. Maier, S. Stoneman, A. F. Prince, B. Rebele, M. Durner, E. Staudinger, S. Zhang, R. Pöhlmann, E. Bischoff, C. Braun, S. Schröder, E. Dietz, S. Frohmann, A. Börner, H. Hübers, B. Foing, R. Triebel, A. O. Albu-Schäffer, and A. Wedler, "The ARCHES Space-Analogue Demonstration Mission: Towards Heterogeneous Teams of Autonomous Robots for Collaborative Scientific Sampling in Planetary Exploration," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5315–5322, 2020.
- [31] M. J. Schuster, M. G. Müller, S. G. Brunner, H. Lehner, P. Lehner, A. Dömel, M. Vayugundla, F. Steidle, P. Lutz, R. Sakagami, L. Meyer, R. Belder, M. Smíšek, W. Stürzl, R. Triebel, and A. Wedler, "Towards Heterogeneous Robotic Teams for Collaborative Scientific Sampling in Lunar and Planetary Environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (Workshop on Informed Scientific Sampling in Large-scale Outdoor Environments)*, 2019.
- [32] P. Lutz, M. G. Müller, M. Maier, S. Stoneman, T. Tomić, I. von Bargen, M. J. Schuster, F. Steidle, A. Wedler, W. Stürzl, and R. Triebel, "ARDEA – an MAV with skills for future planetary missions," *Journal of Field Robotics*, vol. 37, no. 4, pp. 515–551, 2020.
- [33] T. Simon, B. Jens, B. Fabian, H. Andreas, L. Roy, M. Patrick, O. Kazunori, S. Stefanus, W. Konrad, W. Friederike, and Z. Ramona, "Phobos Environment Requirement Document for the MMX Rover Mission," CNES, DLR, et al., Tech. Rep., 2019, internal.
- [34] K. Bussmann, L. Meyer, F. Steidle, and A. Wedler, "Slip Modeling and Estimation for a Planetary Exploration Rover: Experimental Results from Mt. Étna," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 2449–2456.
- [35] M. J. Schuster, S. G. Brunner, K. Bussmann, S. Büttner, A. Dömel, M. Hellerer, H. Lehner, P. Lehner, O. Porges, J. Reill, S. Riedel, M. Vayugundla, B. Vordermayer, T. Bodenmüller, C. Brand, W. Friedl, I. Grixia, H. Hirschmüller, M. Kaßecker, Z.-C. Márton, C. Nissler, F. Ruess, M. Suppa, and A. Wedler, "Towards Autonomous Planetary Exploration: The Lightweight Rover Unit (LRU), its Success in the SpaceBotCamp Challenge, and Beyond," *Journal of Intelligent & Robotic Systems*, Nov. 2017.
- [36] J. J. Biesiadecki and M. W. Maimone, "The mars exploration rover surface mobility flight software driving ambition," in *2006 IEEE Aerospace Conference*, 2006, pp. 15 pp.–.
- [37] L. Preston, M. Grady, and S. Barber, "CAFE. Concepts for Activities in the Field for Exploration. TN2: The Catalogue of Planetary Analogues," The Planetary and Space Sciences Research Institute, The Open University, UK, Tech. Rep., Dec. 2012, under ESA contract: 4000104716/11/NL/AF.
- [38] L. Meyer, M. Smíšek, A. F. Villacampa, L. O. Maza, D. Medina, M. J. Schuster, F. Steidle, M. Vayugundla, M. G. Müller, B. Rebele, A. Wedler, and R. Triebel, "The MADMAX dataset for visual-inertial rover navigation on Mars," 2020, (in press).
- [39] M. Hellerer, S. Barthelmes, and F. Buse, "The DLR Rover Simulation Toolkit," in *14th Symposium on Advanced Space Technologies in Robotics and Automation*, E. van Beekhuizen, Ed. ESA's Automation and Robotics group, June 2017.

BIOGRAPHY



Mallikarjuna Vayugundla is a researcher in the department of "Perception and Cognition" at the German Aerospace Center (DLR). He is working on autonomous navigation algorithms for rovers. He joined DLR as a full-time researcher in 2016. He received his Masters degree in Autonomous Systems from the Bonn-Rhine-Sieg University of Applied Sciences in 2015.



Tim Bodenmüller is senior researcher at the Institute for Robotics and Mechatronics at the German Aerospace Center (DLR). He joined DLR as a full-time researcher in 2001. and is working on 3D-sensing, robotic middleware and software design. He further is member of the institutes software engineering group. He received his Dipl.-Ing. degree in electrical and information engineering in 2001 from the Technical University of Darmstadt and his PhD in electrical engineering in 2009 from the Technical University of Munich.



Martin J. Schuster is senior researcher and leader of the Planetary Exploration Operations team at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). He received his Ph.D. (Dr.-Ing.) degree in Computer Science (CS) from the University of Bremen in 2019, his first Master's degree in CS from the Georgia Institute of Technology in Atlanta, USA in 2010,

his Bachelor's and second Master's degree in CS from the Technische Universität München (TUM) in 2008 and 2011, and his Bachelor's degree in Philosophy from the Ludwig-Maximilians-Universität München (LMU) in 2014. His research focus is on multi-robot SLAM in GNSS-denied areas.



Marcus G. Müller is a researcher and Ph.D. student in the department of "Perception and Cognition" at the German Aerospace Center (DLR) since 2016. He is the leader of the XRotor-Team at the Institute of Robotics and Mechatronics (DLR-RM), where he is working on autonomous navigation algorithms for MAVs. Before joining DLR he conducted research at the Jet Propulsion Laboratory (JPL) of NASA in Pasadena, USA, where he worked on visual inertial navigation for MAVs and on radar signal processing. Marcus received his Master's and Bachelor's degree in Electrical Engineering from the University of Siegen, Germany.

laboratory (JPL) of NASA in Pasadena, USA, where he worked on visual inertial navigation for MAVs and on radar signal processing. Marcus received his Master's and Bachelor's degree in Electrical Engineering from the University of Siegen, Germany.



Lukas Meyer is a researcher and Ph.D. student in the department of "Perception and Cognition" at the German Aerospace Center (DLR). He works in the field of planetary robotics with a focus on testing navigation algorithms. His work includes the testing of navigation solutions during planetary analogue campaigns on a volcano or in the desert. Lukas received his Master's

and Bachelor's degree in Mechanical Engineering and Engineering Science from the Technical University of Munich, Germany.



Patrick Kenny is a researcher in the On-board Software Systems group, Institute for Software Technology at the German Aerospace Center (DLR). Patrick has worked in the medical device and automotive industries before joining the DLR in 2020, where his work includes robust, distributed software platforms for spacecraft. He received his Bachelor's degree in Engineering (Mechatronics) and his

Master's degree (Engineering Science) from the University of Auckland, New Zealand.



Florian Schuler is a researcher at the Institute of Robotics and Mechatronics. After receiving his Master's degree in Computer Engineering from the Leibniz University Hannover, he started his career at the DLR as an FPGA developer in the field of image processing for the MMX space mission.



Markus Bihler received his M.Sc. degree in computer science from the University of Applied Science Augsburg, Germany in 2015. Since then, he has been with the German Aerospace Center, Institute of Robotics and Mechatronics. His main research focus is the field of FPGAs, communication and System-on-Chip architecture inside robots. Since October 2017, he is group leader for digital electronics within the department Mechatronic Systems.

digital electronics within the department Mechatronic Systems.



Wolfgang Stürzl is a senior research scientist in the department of "Perception and Cognition" at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). His research interests include computer vision for mobile robots, in particular using multi-camera and wide-angle imaging systems, and bio-inspired visual navigation of flying systems.



Bernhard-Michael Steinmetz received his M.Sc. degree in Electrical Engineering from TU Munich (Germany) in 1989. He joined the DLR Institute of Robotics and Mechatronics in 1989 as a project and research engineer. Over the years he was responsible for system design and operation in both national and international projects (e.g. ROTEX, ETS-VII, PSPE, ROKVISS, Kontur-1..2).



Jörg Langwald is senior research scientist at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). His research interests include computer vision for mobile robots, especially FPGA based image processing and embedded computing. He received his Dipl.-Ing. degree in electrical engineering in 1995 from the Ilmenau University of Technology, Germany.



Andreas Lund is a Post-Doc researcher in the Onboard Software Systems group, Institute for Software Technology at the German Aerospace Center (DLR). He received his PhD in Computer Science in 2019 from Goethe-University Frankfurt and has been since then with the DLR. His work includes the research of fault-tolerant and distributed embedded software systems in the aerospace domain.



Riccardo Giubilato is a Post-Doc researcher at the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Weßling, Germany. In 2020 he received a PhD from Centre of Studies and Activities for Space (CISAS) "G. Colombo," University of Padova. His current research interests include sensor fusion for vision systems and range sensors, visual SLAM, multiagent localization and mapping.

tion and mapping.



Armin Wedler is a senior researcher and head of the planetary exploration group at the institute of Robotics and Mechatronics of the German Aerospace Center (DLR). He received his Ph.D. (Dr.-Ing.) degree in mechanical engineering from the University of Hanover in 2010, his diploma and bachelor degrees in robotics and mechatronics in 2004 also from the University of

Hanover. Mr. Wedler worked in several space and research projects such as Exomars, Lunar Lander, Dexhand and the light weight rover unit (LRU). He is project leader or spokesperson for several projects and is also active in the ISCEG. His research focuses on the developments of intelligent mobile platforms for exploration and terrestrial usages.



Rudolph Triebel received his PhD in 2007 from the University of Freiburg in Germany. The title of his PhD thesis is "Three-dimensional Perception for Mobile Robots". From 2007 to 2011, he was a postdoctoral researcher at ETH Zurich, where he worked on machine learning algorithms for robot perception within several EU-funded projects. Then, from 2011 to 2013 he worked in

the Mobile Robotics Group at the University of Oxford, where he developed unsupervised and online learning techniques for detection and classification applications in mobile robotics and autonomous driving. Since 2013, Rudolph works as a lecturer at TU Munich, where he teaches master level courses in the area of Machine Learning for Computer Vision. In 2015, he was appointed as leader of the Department of Perception and Cognition at the Robotics Institute of DLR.



Michal Smíšek is a researcher in the department of "Perception and Cognition" at the German Aerospace Center (DLR). He is responsible for the development of the MMX Navigation solution. Michal received his M.Sc. and PhD. degrees in Applied Mathematics from the Slovak University of Technology.



Markus Grebenstein received the Dipl. Ing. degree in mechanical engineering in 1997 from the Technical University in Munich, Germany and the Doctor of Science degree from ETH Zurich, Switzerland in 2013. Engaged at DLR Institute of Robotics and Mechatronics in 1996, he worked on the development of electromechanical passenger vehicle brake systems until 1998. Since then,

his focus has turned to the research and development of anthropomorphic robotic hands and he developed DLR Hand II and DAVID's antagonistically driven Awiwi-hand. He was leading the development of the DLR Hand Arm System until 2012. From 2018 he is project leader at DLR for the MMX Rover mission to the Martian moon Phobos. Starting in 2000, he has served as the Head of Mechanical Development. Since 2010 he has been Head of the Department of Mechatronic Components and Systems. In 2015 he was appointed Deputy Director of the DLR Institute of Robotics and Mechatronics.