Received: 25 July 2017 | Revised: 26 April 2018 | Accepted: 29 April 2018

DOI: 10.1002/rob.21790

REGULAR ARTICLE

WILEY

Adaptive localization and mapping with application to planetary rovers

Javier Hidalgo-Carrió[1] | **Pantelis Poulakis[2]** | **Frank Kirchner[3]**

[1]Robotics Innovation Center (RIC), German Research Center for Artificial Intelligence (DFKI), Bremen, Germany

[2]European Space Research & Technology Centre (ESTEC), European Space Agency, AZ Noordwijk, The Netherlands

[3]Robotics Innovation Center (RIC), German Research Center for Artificial Intelligence (DFKI), and Department of Mathematics and Informatics, University of Bremen, Bremen, Germany

Correspondence
Javier Hidalgo-Carrió, Robotics Innovation Center, German Research Center for Artificial Intelligence, Robert-Hooke-Str. 1, 28359, Bremen (Germany).
Email: javier.hidalgo_carrio@dfki.de

Funding information
Deutsches Zentrum für Luft- und Raumfahrt, Grant/Award Numbers: 50RA1406, 50RA1407; European Space Agency, Grant/Award Number: 4000103229/11/NL/PA

Abstract

Future exploration rovers will be equipped with substantial onboard autonomy. SLAM is a fundamental part and has a close connection with robot perception, planning, and control. The community has made great progress in the past decade by enabling real-world solutions and is addressing important challenges in high-level scalability, resources awareness, and domain adaptation. A novel adaptive SLAM system is proposed to accomplish rover navigation and computational demands. It starts from a three-dimensional odometry dead reckoning solution and builds up to a full graph optimization that takes into account rover traction performance. A complete kinematics of the rover locomotion system improves the wheel odometry solution. In addition, an odometry error model is inferred using Gaussian processes (GPs) to predict nonsystematic errors induced by poor traction of the rover with the terrain. The nonparametric GP regression serves to adapt the localization and mapping to the current navigation demands (domain adaptation). The method brings scalability and adaptiveness to modern SLAM. Therefore, an adaptive strategy develops to adjust the image frame rate (active perception) and to influence the optimization backend by including high informative keyframes in the graph (adaptive information gain). The work is experimentally verified on a representative planetary rover under a realistic field test scenario. The results show a modern SLAM systems that adapt to the predicted error. The system maintains accuracy with less number of nodes taking the most benefit of both wheel and visual methods in a consistent graph-based smoothing approach.

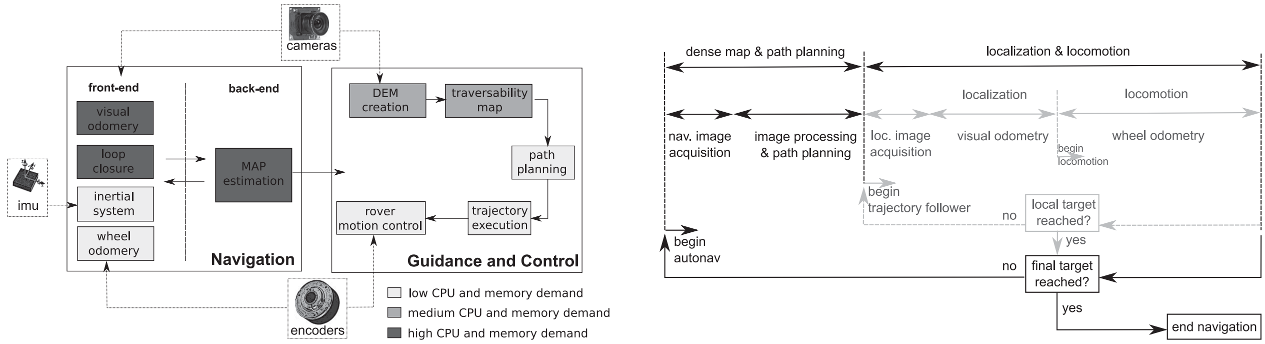
KEYWORDS

mapping, planetary robotics, position estimation

1 | INTRODUCTION

The navigation system is a technological key aspect in mobile planetary robots (i.e., rovers). The system allows to know where the rover is, locate the target, and then guide the robot. These three tasks are essential to perform in situ mission operations in the harsh of a remote environment. The *navigation system* in spacecraft terminology divides into three capabilities as guidance, navigation, and control (GNC). Guidance is the path-planning responsibility. Navigation is the localization and mapping competency, and control is the commanding of the rover locomotion system. These three elements depend on the mission and the requirements affect their design. These requirements are of three types: operational, functional, and resources. Operational imposes the level of autonomy due to a constrained communication bandwidth in space. Functional requirements define the level of performance. Resources establish the sensor type, perception, computational power, and software restrictions. Figure 1a shows a typical GNC system diagram with Simultaneous Localization and Mapping (SLAM) frontend and backend together with the onboard computation demands. Figure 1b depicts the system in two parts. During the first part, the rover acquires images from the navigation cameras, computes a dense map of the surroundings, and calculates the free obstacle path. Consequently, the path is given to the second part to follow the desired trajectory and compute the localization. The localization and locomotion part acquires the images from the localization cameras, computes the image features, extracts the descriptors, and tracks correspondences with respect to the previous pair of images. During locomotion, the rover computes wheel odometry (WO). Feature matching uses the information from the WO to search for potential correspondences in Visual Odometry (VO). The predicted region in the image is calculated using a constant velocity model. The rover locomotion executes the GNC cycle multiple times along the path for the *autonomous driving* until the final location target is reached.

Relative localization strategies, such as WO, are computationally inexpensive and are effective on even terrain with good traction



(a) (b)

FIGURE 1 The GNC system: (a) Diagram of the GNC with the computational demands and (b) GNC cycle for the *autonomous driving* of a planetary rover

properties. Mars Exploration Rovers (MER) accumulated only 3.0% position error over 2 km of driving on level ground (Biesiadecki, Leger, & Maimone, 2007). A more sophisticated visual odometry is not affected by slippage and addresses a significant advance, allowing the rover to navigate through more challenging terrains, increasing the number of science targets to analyze. The application of visual odometry turned out to be essential on MER. Visual odometry allows navigation capability on demanding terrains, typically loose/mixed terrain and/or slopes of 10% and higher, measuring slips as high as 125% when driving up more than 25° slope (Helmick, Cheng, Clouse, & Matthies, 2004; Maimone, Cheng, & Matthies, 2007a). Slippage detection is a complicated task and complex soil parameters are involved. It is described in Biesiadecki et al. (2007), how Spirit reached 100% slippage (no forward progress) on a 16° slope, while only few meters behind had 20% slip on a 19° slope with no discernible difference in the character of the surface.

The penalty of using visual odometry is the computation load and the associated power consumption. Visual odometry takes between 2–3 min to process stereo pair images on the RAD6000 (35 MIPS) processor of the MER rovers, and 60% of overlap between image pairs is required, limiting turning maneuvers. It affects daily operations and degrades the mobility of the whole rover (Biesiadecki et al., 2007; Powell et al., 2006). The localization system has a direct impact on rover trajectory, planning, speed, distance to traverse, ground operations, and scientific return. *Direct driving* speed for MER is about 124 m/h when conducive terrain, and ahead images are available for planning from ground. The rover effective speed decreases to 10 m/h when activating visual odometry. In the case of adding autonomous obstacle avoidance the effective speed reduces to 6 m/h (Biesiadecki et al., 2007). This fact has an important consequence on rover mobility and operations. MER travels 50 m in 25 min using *direct driving* but takes 8 h using visual odometry and obstacle avoidance for the same distance. Therefore, the use of visual odometry and obstacle avoidance was limited considerably in Mars rovers. Different localization solutions were remotely switched from operations to adapt to the mission demands. Visual odometry was used when extremely necessary and *direct driving* on benign terrains. This enhances the use of perception techniques combined with complementary sensor information. Adaptive solutions are desired to reduce human intervention from ground, close the operational loop at the flight segment, and minimize waiting times and communication windows during operation.

The future of space robotics moves the program out of its current technological comfort zone (Maimone, 2014). Future rovers plan to travel longer and faster than past missions. Sample fetching rover (SFR) concepts are studied in the Robotics Exploration Preparation (MREP) program in Europe and NASA post 2020 missions (Exploration & Group, 2011). MER covered ∼15 km in ∼7 years, whereas next rover missions plan a traversal range of ∼20 km in ∼6 months. Traversal requirements drastically affect the locomotion platform and the navigation system design. Figure 2a shows a comparison for mass and primary distance goal for Mars rover missions. Different velocities for past, current, and future Mars rover missions are depicted in Figure 2b. The comparison defines a nominal Mars *locomotion sol* as a driving time of 2.25 h and a *science sol* in which there is not a single rover-driving operation. The *mission velocity* line in Figure 2b is the average mission velocity taking into account *locomotion* and *science sols*. The *direct driving* line computes the average velocity for a *locomotion sol* when *direct driving* is selected (i.e., telecommands with waypoints and no autonomous navigation). The *autonomous driving* line shows the average speed for *locomotion sols* using *autonomous capabilities* (i.e., visual odometry and obstacle avoidance). All lines start at a single point at the Sojourner rover since almost no autonomous surface navigation was available at that time. The curves meet at the SFR concept in which there is no room for speed variation since a Mars return mission imposes such a mission velocity constraint. The plot shows that future rovers will traverse longer distances in a shorter time and depicts how the rover speed is influenced by the surface navigation technology. The navigation strategy is dictated by information available from the previous sol at the ground control station based on surface and soil conditions of the environment. MER maximum average speed is approximately 92 m/sol, which considers a speed peak of 124 m/h in *direct driving* and 96 m/h using *path selection*. Mars Science Laboratory (MSL) speed has been designed to reach almost 100 m/sol in *direct driving* and 43 m/sol considering autonomous capabilities. ExoMars requirements in phase B2 have a nominal speed of 50 m/sol. The total time available in a SFR mission for traversing 15 km in a straight line (22 km effective traversal) in a maximum of 110 sols results in an average rover speed of 200 m/sol.

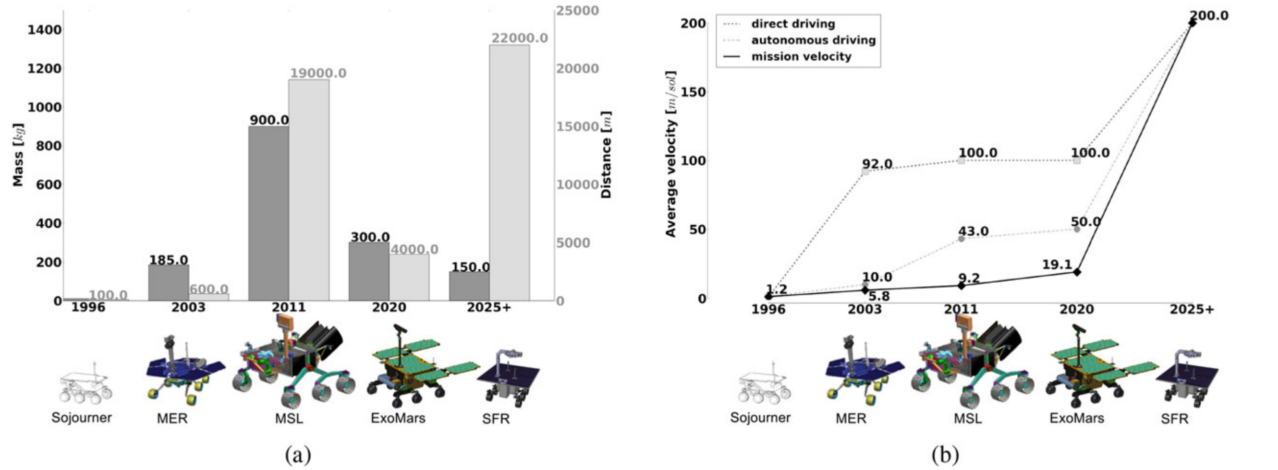


FIGURE 2 Rovers' comparison: (a) mass and primary distance and (b) speed requirements on the past, present, and future Mars rover missions

Future missions are likely to be a sampling fetching mission or a construction site in a moon scenario. Those mission concepts impose the exploration of an unknown environment, collection of samples, and return to the sample capsule. A globally consistent map is required to locate and store scientific material. Without SLAM, the rover interprets the world as an *infinite corridor* making autonomous systems susceptible to fail. Mars missions have demonstrated that the geometric and nongeometric hazards could stop the motion of a rover. These potential hazards are difficult to detect remotely from earth calling the need for an onboard solution (Maimone, Leger, & Biesiadecki, 2007b). The solution would deduce rover position and orientation in a prolonged, optimal, and adaptive manner. The future of SLAM will combine machine-learning techniques with optimization approaches and hardware improvements (e.g., Field Programmable Gate Array (FPGA) and Graphics Processing Unit (GPU)) to enable reliable and fast surface navigation.

The present work proposes several techniques for modern SLAM systems in the context of planetary rovers with the following:

- *An enhanced three-dimensional (3D) odometry model.* Indoor robotics, which traditionally operates in a structured or planar environment, has brought inefficient and simple techniques to the field of WO in outdoor robotics. The estimation of delta displacement through complete motion models (Hidalgo-Carrio, Babu, & Kirchner, 2014) produces a more accurate a priori estimation.
- *Gaussian Process (GP) regression for odometry errors.* The development of a GP to relate the statistical error or uncertainty in WO as a non-systematic error model. The model identifies hazardous areas by learning Bayesian kernels from previous rover's experience. The technique was first introduced in our previous work from Hidalgo-Carrio, Hennes, Schwendner, and Kirchner (2017).
- *Adaptive graph sparsity.* The SLAM is influenced by errors from the WO and brings adaptivity to the solution, that is, adaptive localization and mapping. This is the main contribution of this work.

The paper is organized as follows: A review of related work is presented in Section 2 and a description of the methodology is given in Section 3. First, the enhanced 3D odometry model with a full kinematic solution is given. Second, a machine-learning technique using GP regression for the odometry error is introduced. Third, a localization and mapping solution with an adaptive graph sparsity is defined. Field testing experiments with a planetary rover under representative environment conditions are shown in Section 4. Conclusions are finally discussed in Section 6.

2 | RELATED WORK

The most complete and reliable localization system for planetary rovers up to now is onboard the MSL and the MER. Surface Attitude Position and Pointing (SAPP) is the rover software component in charge of calculating and propagating rover attitude and position estimation and is further explained in Ali et al. (2005). SAPP uses different techniques and combines diverse sensory information. The module carries out the propagation of rover pose depending on three commands defined by the attitude acquisition machine. The system is triggered from ground depending on rover operations. The relative localization computes and propagates attitude using gyro angular rate integration with additional support from Sun elevation information using camera images. Accelerometers data are used in static regimen to verify a correct estimation of the gravity vector. No SLAM solution is currently implemented in space systems. The rover position is propagated using WO, and no accelerometers are used in this step (Maimone et al., 2007a). Visual odometry is triggered when high slippage is expected by previous daily analysis from ground.

2.1 | Wheel odometry

Extensive research with different reasoning is available in the literature. Alexander and Maddocks (1989) presented a planar rigid body model considering a variable number of wheels. Campion, Bastin, and D'Andrea-Novel (1996) classified ordinary mobile robots into five types taking into consideration generic parts of the model equation. Other research into the wheel-ground contact angle and pose estimation of robots moving on uneven surfaces can also be found in Lamon and Siegwart (2007) and Iagnemma and Dubowsky (2000). The kinematics transformation approach for locomotion systems is introduced in Muir and Neuman (1986). Their work provides generality and consistency by applying a series of transformations and Jacobian matrices to relate motion from the joint space to the Cartesian space. First simulations results are investigated for the Rocky7 rover by defining the wheel contact angle and slip vector in Tarokh and Mcdermott (2005) without experiments in a real rover and no results in odometry. A similar method is applied to a 6-DoF (degree of freedom) motion model limited to passive joints in Gajjar and Johnson (2002) but still without field-testing results. The technique is extended to model a fully actuated system (not only passive) in Hidalgo-Carrió and Cordes (2012) and Roehr, Cordes, and Kirchner (2014) using the Sherpa rover. A scout robot, Asguard from Joyeux et al. (2011), propagated the pose with a complete kinematics model in Hidalgo-Carrio et al. (2014) showing the generality of the approach to model leg-wheel hybrid robot. This paper shows the application of the original technique described in Muir and Neuman (1986) beside a traditional wheeled system in outdoor environments. Our previous work already applies a weighted solution showing a more accurate performance that without weighting the contact points.

WO errors are investigated in the literature. Borenstein and Feng (1996) investigated the elimination of systematic odometry errors. Their work focuses on calibration methods to reduce the effect of unequal wheel diameters and uncertainty about the wheelbase. Slippage has been the main nonsystematic error as it causes bad results affecting the final pose. Visual odometry is commonly used to overcome the effect of slippery terrains in Helmick et al. (2004) and Rehder, Gupta, Nuske, and Singh (2012). Fuzzy logic is used to detect wheel slippage by comparing the motor current on the FlexNav architecture in the work presented by Ojeda and Borenstein (2002). They introduced a linearized function to relate electric current and wheel–terrain interaction (Ojeda, Cruz, Reina, Borenstein, & Member, 2006). Rogers-marcovitz, George, Seegmiller, and Kelly (2012) presented a delayed state filter technique in combination with a vehicle system model to correct wheel slip. Their work shows the viability and value of slip modeling.

2.2 | Gaussian processes

They provide a probabilistic approach for learning kernel machines, producing promising results in robotics. The probabilistic nature of GPs makes them attractive to integrate with Bayes-based approaches, either Kalman-based or particle filters. Ko, Klein, Fox, and Haehnel (2007) and Ko and Fox (2008) applied a GP to learn the residuals of the dynamic model of a robotic blimp. The work is afterward applied to dynamic state estimation and control of the blimp with an unscented Kalman filter. In general, GPs have several advantages for robotics since they are a practical tool for solving a diverse set of problems. Mukadam, Yan, and Boots (2016) applied GPs in robotic arm motion planning. In contrast to considering discrete time trajectories, the method represents a continuous time trajectory as a sample from a GP. The use of GPs for mapping is described in Wang and Englot (2016). The technique collects sensor observations and estimates the occupancy map using an octree. The use of GPs in odometry is described in our previous work (Hidalgo-Carrio et al., 2017). The technique uses a GP model for nonlinear regression similarly to Cunningham, Masahiro, Nesnas, Yen, and Whittaker (2017). Our work estimates a 3D slip vector using one GP model for regression, whereas Cunningham et al. (2017) train two GP models, one for longitudinal and one for lateral slip, training the model with telemetry from the Curiosity rover.

2.3 | SLAM using filters

Originally, SLAM initiates the development by using filtering approaches. Filtering provides efficient estimation of the latest state (Civera, Davison, & Montiel, 2008). However, Kalman-based or particle filter SLAM becomes intractable in real-time applications. Extended Kalman Filter (EKF) complexity grows quadratically with the number of features. The state dimension of a particle filter requires an exponential increase in the number of particles. An alternative is presented in Anastasios and Roumeliotis (2006) with an augmented state EKF. The approach keeps the complexity linear with the number of features by marginalizing them out of the state vector. The method implies the clone of past states in the state vector, *stochastic cloning*. The solution has a computational cost of cubic complexity with the number of states and is capable of accurate pose estimation. The cloning does not affect the real-time behavior since the number of states is much lower than the number of features. However, marginalization is a source of errors and outliers can corrupt the filter with irreversible results. Such inconsistency in the filtering estimators entails to perform an observability analysis in vision inertial navigation systems (Hesch, Kottas, Bowman, & Roumeliotis, 2014). The analysis provides modification on the measurement Jacobian matrix in the observable direction of the system (Dong-Si & Mourikis, 2011). In addition, particle-filtering SLAM was introduced by Montemerlo, Thrun, Roller, and Wegbreit (2003) to handle the nonlinearity of the problem. The high dimensionality makes particle filter intractable for real-time applications. The Rao-Blackwellization variant reduces the sample space, making it more suitable for mobile robotics and real-time constraints. Each particle still carries an estimation of the environment making the approach difficult to scale up for long-term applications.

2.4 | Graph SLAM and smoothers

Smoothing can be an alternative to filtering. Fixed-lag smoothers estimate the state within a window of measurements. Smoothing approaches relinearize past measurements providing more accurate estimates. Smoothers are more robust in case of outliers, which make them suitable for long-term localization (Mourikis & Roumeliotis, 2008). However, the smoother needs to take care of consistency analysis and accumulated linearization errors as in filtering methods. To overcome the limitation mentioned above, full smoothers estimate the complete robot trajectory and features as in Kummerle, Grisetti, Strasdat, Konolige, and Burgard (2011). Full-SLAM solves a large optimization problem. The optimization imposes the highest accuracy, but the computation cost quickly grows with time. Incremental smoothing techniques by means of factor graphs and Bayes trees allows fluid relinearization and update only affected nodes of the graph (Grisetti, Kummerle, Stachniss, & Burgard, 2010; Kaess et al., 2011). Forster, Carlone, Dellaert, and Scaramuzza (2015) and Forster, Carlone, Dellaert, and Scaramuzza (2017) proposed preintegration of inertial measurement unit (IMU) measurements to reduce the number of nodes while preserving the manifold structure of the $SO(3)$ rotation group. Carlone, Kira, Beall, Indelman, and Dellaert (2014) defined a set of target variables to deal with smaller graphs. The solution enhances computational efficiency and robustness in the backend. However, the relation with the perception frontend and robot navigation demands to define a smaller graph is missing. A keyframe-based approach is proposed in Leutenegger, Lynen, Bosse, Siegwart, and Furgale (2015) to select graph nodes and improve sparsity in visual-inertial odometry.

2.5 | SLAM architectures

One of the first real-time modern SLAM architectures is the parallel tracking and mapping (PTAM) described in Klein and Murray (2007). PTAM is a dual threading architecture, one thread for features tracking and other for mapping. CD-SLAM (Pirker, Ruther, & Bischof, 2011) and ORB-SLAM (Mur-Artal, Montiel, & Tardos, 2015) include a third thread for loop closing. Engel, Stuckler, and Cremers (2015) introduce LSD-SLAM, a monocular direct (feature-less) SLAM, which minimizes the photometric error between consecutive frames, and it is also able to detect large-scale loop closures. Most of the time SLAM operates on open loop. The error propagates unbounded unless a solution to close the loop by revisiting a place is given. Maintaining data association is key to guarantee a consistent map. Standard matching algorithms fail when areas not observed for long are revisited afterward. Several techniques have been investigated in the past. Loop closures using features in the map are investigated in Clemente, Davison, and Reid (2007). The method looks for a set of similar features (i.e., geometry and appearance) between pairs of nonconsecutive submaps. Williams, Klein, and Reid (2007) find camera poses using the features in previous keyframe maps, similar to a relocalization event. Appearance-based techniques use directly the image sequence as described in Cummins and Newman (2008). The work detects and compares visual words between image pairs. Though not perfect, the method scales well in large data sets and long trajectories. Alternatively, the bag of words results in an effective solution. The technique converts an image into a numerical vector using a visual vocabulary previously created offline (Galvez-Lopez & Tardos, 2012).

2.6 | Exploiting sparsity

Introducing highly informative nodes and nonredundant poses improves computation cost and guarantees consistency. Heuristic strategies based on the distance traveled are investigated in Konolige and Agrawal (2008). The paper shows how the root mean square error (RMSE) increases as a function of the distance between image frames. Their technique is based on a distance-based criterion and does not adapt to rover maneuvers and the characteristics of the terrain. Alternatively, active SLAM solutions prevent wrong measurements by selecting a maximal informative motion command in the control of the platform (Vidal-Calleja, Davison, Andrade-Cetto, & Murray, 2006). Other strategies such as the information-based Pose-SLAM use interval arithmetics and provide informative nodes and relevant poses in the graph (Ila, Porta, & Andrade-Cetto, 2010). Their strategy uses an error function between two poses using a predefined and fixed threshold. Their work emphasizes but does not develop the benefits of adjusting such a threshold according to situation demands, decreasing it as the robot gets lost. This article presents the research to develop a solution that models an error function to provide informative nodes in the graph, as in Ila et al. (2010) but adapting to the current navigation demands dictated by the interaction with the terrain. The scalability of the SLAM solution is the scope of adaptive SLAM.

3 | METHODOLOGY

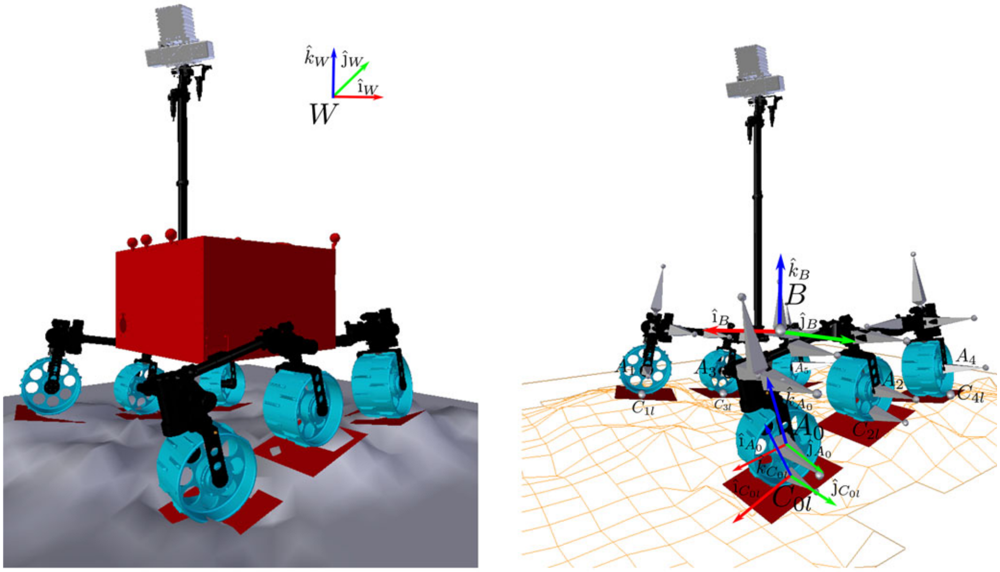
This section expounds the methodology of our research on adaptive localization and mapping in unstructured environments. The design starts from the odometry model followed by the error model propagation until the final adaptive SLAM solution.

3.1 | Kinematics model for WO

WO together with inertial navigation systems are the most common techniques for dead reckoning. Many robotic applications combine both due to consistency over time. The attitude kinematics is self-sufficient with inertial sensors except for the less observable angle (i.e., the heading), and odometry is a reasonable first estimate of the pose as explained in Crassidis and Markley (2003), Hidalgo-Carrió, Poulakis, Köhler, Del-Cerro, and Barrientos (2012) and Hidalgo-Carrio et al. (2014).

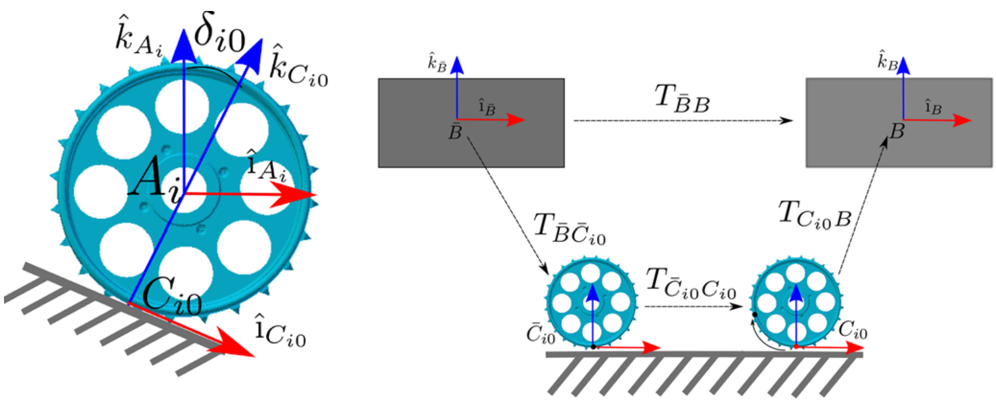
A minimum of two coordinate frames per kinematic chain are required: a robot body frame B attached to the desired rover center and a contact frame C_{il} defined as a single point of contact between the robot and the ground. The other coordinate frames are the minimum required for the computation of the homogeneous transformation matrix $T_{B,C_{il}}$ which relates frame B with frame C_{il}. It will depend on a particular kinematics and the number of joints represented by the vector $\boldsymbol{q} = [q_0\ q_1 \cdots q_n]$, where n is the number of degrees of freedom of the mechanism (see Figure 3 for a visual illustration). The B frame is related to a fixed world frame W by the pose vector $\vec{u} = \boldsymbol{u}_{W,B} = [x_W\ y_W\ z_W\ \phi_W\ \theta_W\ \psi_W]$.

Planetary rovers mostly navigate on uneven terrains. They require the definition of contact angles between the ground and the point in contact. Typically, one contact angle in the direction of motion is necessary for wheeled mobile robots, δ_{il} as depicted in Figure 4. Pure walking machines might require two angles at the point of contact (i.e., the gradients in the lateral and transversal direction). In addition, the motion of the contact point consists of a slip vector ϵ_{il}, which is modeled in three dimensions. A translation in the $\hat{\imath}_{C_{il}}$ axis by ξ_{il}, a lateral slip η_{il} along the $\hat{\jmath}_{C_{il}}$ axis, and a rotational slip ζ_{il} along the $\hat{k}_{C_{il}}$ axis resulting to a vector $\boldsymbol{\epsilon}_{il} = [\xi_{il}\ \eta_{il}\ \zeta_{il}]$.



(a) (b)

FIGURE 3 Illustration of the 3D kinematics. (a) The rover on an uneven terrain. Depicted the world reference frame W. (b) Kinematics modeling. B is the body frame located behind the mast and between the middle wheels, A_i is the wheel frame, and C_{il} the contact point frame. The rover's delta pose displacements are computed as composite equations of wheel Jacobian matrices



(a) (b)

FIGURE 4 Schematic representation of (a) coordinate frames for ith wheel on an inclined terrain and (b) 3D kinematics of movement to induce delta displacement from a single wheel i with one contact point $l = 0$. The wheel rotates clockwise

Mobile robots are commonly commanded by desired velocities. The mapping between using the robot body frame Cartesian space rate vector $\dot{\boldsymbol{u}} = [\dot{x}_B\ \dot{y}_B\ \dot{z}_B\ \dot{\phi}_B\ \dot{\theta}_B\ \dot{\psi}_B]$ and the joint space rate vector with the contact rate angle and slip rate vector is solved by the Jacobian matrix. Velocity kinematics is deduced by derivation of the transformation matrix. The transformation of the rover body at time step $k-1$ (frame $\bar{B}$) to rover body at time step k (frame B) is defined by $T_{\bar{B},B} = T_{\bar{B},\bar{C}_{il}} T_{\bar{C}_{il},C_{il}} T_{C_{il},B}$ as depicted in Figure 4b. The derivative is

$$\dot{T}_{\bar{B},B} = \dot{T}_{\bar{B},\bar{C}_{il}} T_{\bar{C}_{il},C_{il}} T_{C_{il},B} + T_{\bar{B},\bar{C}_{il}} \dot{T}_{\bar{C}_{il},C_{il}} T_{C_{il},B} + T_{\bar{B},\bar{C}_{il}} T_{\bar{C}_{il},C_{il}} \dot{T}_{C_{il},B} \tag{1}$$

In practice, $T_{\bar{B},\bar{C}_{il}}$ is independent of time and the derivative simplifies as $\dot{T}_{\bar{B},B} = T_{\bar{B},\bar{C}_{il}} \dot{T}_{\bar{C}_{il},B}$. It is noticeable that $\dot{T}_{\bar{B},B}$ has the form of a free body in motion with linear and angular velocities, as in Tarokh and Mcdermott (2005):

$$\dot{T}_{\bar{B},B} = \begin{bmatrix} 0 & \dot{\psi}_B & \dot{\theta}_B & \dot{x}_B \\ \dot{\psi}_B & 0 & -\dot{\phi}_B & \dot{y}_B \\ -\dot{\theta}_B & \dot{\phi}_B & 0 & \dot{z}_B \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

The rover configuration rates are expressed in term of joints quantities by substituting Equation (1) into the skew-symmetric matrix of Equation (2). The resulting Jacobian matrix J_{il}, related to a single contact point il, has the form

$$\left[\dot{x}_B\ \dot{y}_B\ \dot{z}_B\ \dot{\phi}_B\ \dot{\theta}_B\ \dot{\psi}_B\right]^{\intercal} = J_{il}[\dot{\boldsymbol{q}}\dot{\boldsymbol{\varepsilon}}_{il}\dot{\boldsymbol{\delta}}_{il}]^{\intercal} \tag{3}$$

It defines the contribution of each kinematic chain to the body motion allowing the analysis of each chain and contact point to the resulting final velocity in $\dot{\boldsymbol{u}}$. Considering a single contact angle, the J_{il} matrix size is $6 \times (n+4)$, where n corresponds to the DoF of the mechanism and 4 to the slip vector and contact angle. The composite rover equations are obtained combining the Jacobian matrices for all kinematics chains (i.e., contact points) into a sparse matrix equation of appropriate dimensions

$$\begin{bmatrix} I_{6\times6} \\ I_{6\times6} \\ \vdots \\ I_{6\times6} \end{bmatrix} \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \\ \dot{\phi}_B \\ \dot{\theta}_B \\ \dot{\psi}_B \end{bmatrix} = J \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\varepsilon} \\ \dot{\delta} \end{bmatrix} \equiv S\dot{\boldsymbol{u}} = J\dot{\boldsymbol{p}} \tag{4}$$

Navigation kinematics relates the rover pose rates to joints and sensed rate quantities. The navigation kinematics is the input for statistical motion models and the basis for dead reckoning estimation. Robots' sensor availability defines sensed and nonsensed quantities, and Equation (4) separates into the following form:

$$\begin{bmatrix} S_s & S_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_s \\ \dot{\boldsymbol{u}}_n \end{bmatrix} = \begin{bmatrix} J_s & J_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_s \\ \dot{\boldsymbol{p}}_n \end{bmatrix} \tag{5}$$

Rearranging into nonsensed (left-side) and sensed (right-side) quantities, the resulting equation is obtained:

$$\begin{bmatrix} S_n & -J_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_n \\ \dot{\boldsymbol{p}}_n \end{bmatrix} = \begin{bmatrix} -S_s & J_s \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_s \\ \dot{\boldsymbol{p}}_s \end{bmatrix} \equiv \Omega \boldsymbol{v} = \boldsymbol{b} \tag{6}$$

where Ω is the matrix whose dimensions depend on the sensing capabilities of the rover and directly influence the existence of a solution. The solution to the overdetermined system above is based on minimizing the error vector $E = \boldsymbol{e}^{\mathsf{T}} \mathcal{W} \boldsymbol{e}$, where $\mathcal{W}$ encodes the individual contribution of each kinematic chain il to the estimated solution

$$E = \boldsymbol{e}^{\mathsf{T}} \mathcal{W} \boldsymbol{e} = (\boldsymbol{b} - \Omega \boldsymbol{v})^{\mathsf{T}} \mathcal{W} (\boldsymbol{b} - \Omega \boldsymbol{v}) \tag{7}$$

The least squares solution of the system in Equation (7) provides a minimum error vector $\boldsymbol{e}$. However, a poor traction from a single wheel might influence the solution by increasing the resulting error. The solution is weighted (adapted) to minimize such influence in the final estimate. The weighting matrix is computed based on the normal force sensed at each wheel. Planetary rovers do not always have a force sensor at the contact point. Therefore, to overcome such limitation a quasi-static force estimation is computed. The lowest point along the wheel circumference is assumed to be always in contact with the ground as depicted in Figure 4. The assumption is valid for most of the cases. However, it does not always hold on highly uneven terrains.

3.1.1 | Quasi-static force estimation

The quasi-static forces are calculated by defining a body-fixed reference frame B_{CoM} with origin at the center of mass (CoM) and $\boldsymbol{w}$ is the weight of the robot acting along the gravity vector ($\hat{k}_W$ axis). The free body diagram for computation of static forces is given in Figure 5. Let

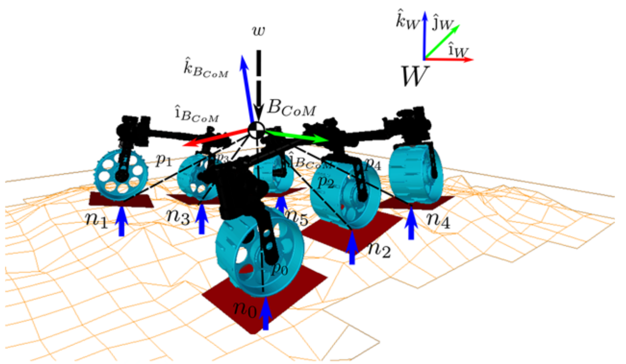


FIGURE 5 Free body diagram for static force computations in a planetary rover

$i = \{0, 1, 2, 3, 4, 5\}$, n_i is the normal reaction force per contact point of wheel i from the ground due to the robot weight and $\boldsymbol{p}_i$ is the position vector to such contact point. Henceforth, the corresponding coordinate systems are added to the representation. A new reference frame B'_{CoM} (not shown in Figure 5) can be defined with the origin coinciding with the CoM of the robot, but aligned to W. The terms for $\boldsymbol{n}_{B'_{CoM}i}$ and $\boldsymbol{w}$ in B'_{CoM} are given by

$$\boldsymbol{n}_{B'_{CoM}i} = \begin{bmatrix} 0 \\ 0 \\ n_i \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{8}$$

where n_i are the scalar reaction forces along the $\hat{k}_{B'_{CoM}}$, m is the mass of the robot and g is the acceleration due to gravity. The objective is to derive the equations for the values of n_i. The equations are developed based on the fact that the robot has three passive links, and these joints cannot transmit any torque. Therefore, the torques in the passive links of the chassis along the free joints are independent. When the robot is quasi-static, the following equations apply:

1. The sum of forces along $\hat{k}_{B'_{CoM}}$ equals the weight of the robot.

$$\Sigma n_i = mg \tag{9}$$

2. The sum of torques along $\hat{\imath}_{B_{CoM}}$ for wheel $i = 0, \ldots, 3$ is zero.

$$\Sigma \left(\boldsymbol{p}_{B_{CoM}i} \times \boldsymbol{n}_{B_{CoM}i} \right) |_{\hat{\imath}_{B_{CoM}}} = 0 \tag{10}$$

3. Sum of torques along $\hat{\imath}_{B_{CoM}}$ for wheel $i = 4, 5$ is zero.

$$\Sigma \left(\boldsymbol{p}_{B_{CoM}i} \times \boldsymbol{n}_{B_{CoM}i} \right) |_{\hat{\imath}_{B_{CoM}}} = 0 \tag{11}$$

4. The sum of torques due to $\boldsymbol{n}_0$ and $\boldsymbol{n}_2$ along $\hat{\jmath}_{B_{CoM}}$ is zero.

$$\left(\boldsymbol{p}_{B_{CoM}0} \times \boldsymbol{n}_{B_{CoM}0} + \boldsymbol{p}_{B_{CoM}2} \times \boldsymbol{n}_{B_{CoM}2} \right) |_{\hat{\jmath}_{B_{CoM}}} = 0 \tag{12}$$

5. The sum of torques due to $\boldsymbol{n}_1$ and $\boldsymbol{n}_3$ along $\hat{\jmath}_{B_{COM}}$ is zero.

$$\left(\boldsymbol{p}_{B_{CoM}1} \times \boldsymbol{n}_{B_{CoM}1} + \boldsymbol{p}_{B_{CoM}3} \times \boldsymbol{n}_{B_{CoM}3} \right) |_{\hat{\jmath}_{B_{CoM}}} = 0 \tag{13}$$

6. The sum of torques along $\hat{j}_{B_{CoM}}$ for all wheels is zero.

$$\Sigma\left(\boldsymbol{p}_{B_{CoM}i}\times\boldsymbol{n}_{B_{CoM}i}\right)|_{\hat{j}_{B_{CoM}}}=0 \tag{14}$$

Let the rotation from B_{CoM} to B'_{CoM} and the position vector $\boldsymbol{p}_{B_{CoM}i}$ be given by

$$R_{B_{CoM}B'_{CoM}}=\begin{bmatrix} r_{00} & r_{01} & r_{02}\\ r_{10} & r_{11} & r_{12}\\ r_{20} & r_{21} & r_{22}\end{bmatrix},\quad \boldsymbol{p}_{B_{CoM}i}=\begin{bmatrix} p_{i\hat{i}}\\ p_{i\hat{j}}\\ p_{i\hat{k}}\end{bmatrix} \tag{15}$$

Using Equation (15), the relationship between $\boldsymbol{n}_{B'_{CoM}i}$ and $\boldsymbol{n}_{B_{CoM}i}$ is given by

$$\boldsymbol{n}_{B_{CoM}i}=R_{B_{CoM}B'_{CoM}}\boldsymbol{n}_{B'_{CoM}i}=\begin{bmatrix} r_{02}\\ r_{12}\\ r_{22}\end{bmatrix}n_i \tag{16}$$

Computing torques from Equations (15) and (16),

$$\boldsymbol{\tau}_{B_{CoM}i}=\boldsymbol{p}_{B_{CoM}i}\times\boldsymbol{n}_{B_{CoM}i}=\begin{bmatrix} p_{i\hat{i}}\\ p_{i\hat{j}}\\ p_{i\hat{k}}\end{bmatrix}\times\begin{bmatrix} r_{02}\\ r_{12}\\ r_{22}\end{bmatrix}n_i \tag{17}$$

$$\boldsymbol{\tau}_{B_{CoM}i}=\begin{bmatrix} p_{i\hat{j}}r_{i\hat{k}}-p_{i\hat{k}}r_{i\hat{j}}\\ p_{i\hat{k}}r_{i\hat{i}}-p_{i\hat{i}}r_{i\hat{k}}\\ p_{i\hat{i}}r_{i\hat{j}}-p_{i\hat{j}}r_{i\hat{i}}\end{bmatrix}n_i \tag{18}$$

Using Equation (18), let

$$\begin{bmatrix} p_{i\hat{j}}r_{i\hat{k}}-p_{i\hat{k}}r_{i\hat{j}}\\ p_{i\hat{k}}r_{i\hat{i}}-p_{i\hat{i}}r_{i\hat{k}}\\ p_{i\hat{i}}r_{i\hat{j}}-p_{i\hat{j}}r_{i\hat{i}}\end{bmatrix}=\begin{bmatrix} t_{i\hat{i}}\\ t_{i\hat{j}}\\ t_{i\hat{k}}\end{bmatrix} \tag{19}$$

Substituting Equations (17)–(19) in Equations (10)–(14) and combining them with Equation (9) gives

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1\\ t_{0\hat{i}} & t_{1\hat{i}} & t_{2\hat{i}} & t_{3\hat{i}} & 0 & 0\\ 0 & 0 & 0 & 0 & t_{4\hat{i}} & t_{5\hat{i}}\\ t_{0\hat{j}} & 0 & t_{2\hat{j}} & 0 & 0 & 0\\ 0 & t_{1\hat{j}} & 0 & t_{3\hat{j}} & 0 & 0\\ t_{0\hat{j}} & t_{1\hat{j}} & t_{2\hat{j}} & t_{3\hat{j}} & t_{4\hat{j}} & t_{5\hat{j}}\end{bmatrix}\begin{bmatrix} n_0\\ n_1\\ n_2\\ n_3\\ n_4\\ n_5\end{bmatrix}=\begin{bmatrix} mg\\ 0\\ 0\\ 0\\ 0\\ 0\end{bmatrix} \tag{20}$$

The system of linear equations (20) can be solved for n_i by

$$\begin{bmatrix} n_0\\ n_1\\ n_2\\ n_3\\ n_4\\ n_5\end{bmatrix}=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1\\ t_{0\hat{i}} & t_{1\hat{i}} & t_{2\hat{i}} & t_{3\hat{i}} & 0 & 0\\ 0 & 0 & 0 & 0 & t_{4\hat{i}} & t_{5\hat{i}}\\ t_{0\hat{j}} & 0 & t_{2\hat{j}} & 0 & 0 & 0\\ 0 & t_{1\hat{j}} & 0 & t_{3\hat{j}} & 0 & 0\\ t_{0\hat{j}} & t_{1\hat{j}} & t_{2\hat{j}} & t_{3\hat{j}} & t_{4\hat{j}} & t_{5\hat{j}}\end{bmatrix}^{-1}\begin{bmatrix} mg\\ 0\\ 0\\ 0\\ 0\\ 0\end{bmatrix} \tag{21}$$

The reaction forces are computed at every time step using Equation (21) and input to the weighted matrix $\mathcal{W}$. The $\mathcal{W}$ matrix in Equation (7) is a diagonal matrix, which consists of six block matrices. Each 6×6 block matrix is a single diagonal matrix associated per each wheel and defines the contact points which are more likely to contribute to the motion of the robot. The block matrices are selected to have the structure $\mathcal{W}_{il}=w_{il}I$, where w_{il} is the contribution of the ilth contact point to the motion of the body. In the ideal case of a balance configuration, all wheels have equal contribution to the robot motion and the value $1/N$ is equally set at each contact point. In practice, the quasi-static force estimator combines the attitude information coming from the IMU and estimates the forces. The values are computed every delta pose, and the instantaneous likelihood of each contact point is calculated accordingly with $\sum_i\sum_l w_{il}=1$.

3.2 | GP regression for odometry errors

The accuracy of odometry is highly influenced by the amount of wheel slippage, which in turn depends on the maximum usable tractive force between the ground and the wheel. We describe here the application of GPs to model a nonlinear regression between the parametric model and the real odometry output. The model serves to identify the challenges of the terrain. GPs are a powerful, nonparametric tool for learning regression functions from sample data. GPs are flexible, work nicely with missing and noisy data and therefore are a practical tool to solve real-world scenarios. A GP is a probability distribution over functions. It can be considered a Gaussian distribution over an infinitely long vector of data (Rasmussen & Williams, 2006). However, an infinite vector is impractical because computer memory is finite. Marginal likelihood or marginalization allows to work with a finite subset without losing generality. Figure 6 shows the train and test schematics of the odometry error model.

Assume we have a training set of data, $D=\langle X,Y\rangle$, where $X=[\boldsymbol{x}_1,\boldsymbol{x}_2,\dots\boldsymbol{x}_n]$ is a matrix containing d-dimensional input examples $\boldsymbol{x}_i$ and $Y=[\boldsymbol{y}_1,\boldsymbol{y}_2,\dots,\boldsymbol{y}_n]$ is a matrix containing o-dimensional training set $\boldsymbol{y}_i$ (i.e., multidimensional output). The GP assumes that data are illustrated with a noisy function such as

$$\boldsymbol{y}_i=f(\boldsymbol{x}_i)+\epsilon \tag{22}$$

where ϵ is a zero-mean Gaussian noise with variance σ^2, that is, $\mathcal{N}(0,\sigma^2)$. The prediction over the noisy output $\boldsymbol{y}$ is a multivariable Gaussian of the input matrix X.

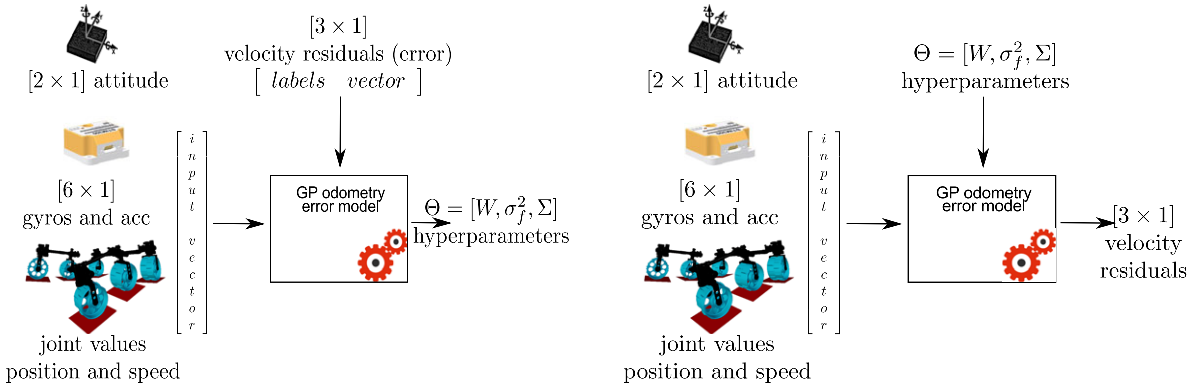
$$p(Y|X)=\mathcal{N}(0,K(X,X)+\Sigma) \tag{23}$$

where $K\equiv K(X,X)$ is the kernel matrix with elements $K_{ij}=k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ defined by the kernel function and $\Sigma=diag(\sigma_1^2I,\dots,\sigma_n^2I)$. The kernel function $k(\boldsymbol{x},\boldsymbol{x}')$ measures the *closeness* between inputs. The most widely used kernel function is the squared exponential also known as radial basis function kernel

$$k(\boldsymbol{x},\boldsymbol{x}')=k_{rbf}(\boldsymbol{x},\boldsymbol{x}')=\sigma_f^2e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')W(\boldsymbol{x}-\boldsymbol{x}')^{\mathsf{T}}} \tag{24}$$

with hyperparameters $\boldsymbol{\Theta}=[W,\sigma_f^2,\Sigma]$. The matrix W contains the length scale per input dimension and σ_f^2 characterizes the signal variances.

Learning a GP is an inductive process that makes a particular reasoning (function) from a set of data. The process of learning defines a finite set of training data to define a function f that makes predictions



(a) (b)

FIGURE 6 GP diagrams for the odometry error model in planetary rovers (a) learning (b) prediction

for all possible inputs in the future. The GP is trained with a selected training data set $D = \langle X, Y \rangle$. The GP defines a predictive distribution over the output Y_* at arbitrary prediction points X_*. The mean function is

$$GP_\mu(X_*, D) = k_*^\intercal [K + \Sigma]^{-1} Y \tag{25}$$

and variance

$$GP_\Sigma(X_*, D) = k(X_*, X_*) - k_*^\intercal [K + \Sigma]^{-1} k_* \tag{26}$$

where k_* is the vector defined by the kernel values between X_* and the training input X as $k(X, X_*)$ and K is the $n \times n$ kernel matrix of the training input values. The prediction uncertainty, captured by the variance GP_Σ, depends on the process noise and the correlation between the test input and the training data. Covariance functions are semipositive definite functions where all the modeling occurs. The covariance function has a set of free parameters $\boldsymbol{\Theta}$, and the learning process optimizes the values given a training set of data.

The hyperparameters $\boldsymbol{\Theta}$ can be learned by maximizing the log likelihood of the training outputs given the inputs,

$$\boldsymbol{\Theta}_{max} = \arg\max\{\log(p(Y|X, \boldsymbol{\Theta}))\} \tag{27}$$

which log term can be expressed as

$$\log(p(Y|X)) = -\frac{1}{2} Y^\intercal (K(X,X) + \Sigma)^{-1} Y - \frac{1}{2} \log \|K(X,X) + \Sigma\| - \frac{n}{2} \log 2\pi \tag{28}$$

3.2.1 | GP modeling of discrete time dynamic processes

Generally, a discrete-time dynamic process can be understood as a series of states at a certain time stamp which evolve over time as

$$\boldsymbol{s}(k+1) = \boldsymbol{s}(k) + g(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k)) \tag{29}$$

where k is the time index and g is the function that described the dynamics of the system (e.g., rover pose rates) given a certain state s and the input vector $\breve{\boldsymbol{u}} = [\dot{\boldsymbol{u}}_s, \dot{\boldsymbol{p}}_s]$ (e.g., odometry inputs). Then, a GP can be used to learn the dynamic process described by the function g (e.g., an odometry model). The result will be a GP, which predicts the delta between two consecutively states $y_k = \boldsymbol{s}(k+1) - \boldsymbol{s}(k)$ given a vector of inputs. To perform such prediction, the output for the parametric model should be part of the training data. This is because the GP assumes a zero-mean function in Equation (23), and the robot odometry is clearly not a zero mean. This is related to the modeling that appears to be in the covariance function. Instead of using the parametric model as input to the GP regression, the GP learns the residual between the parametric model and the expected data (i.e., the odometry error). This is because the residual or error is the value of interest and has a mean close to zero in the ideal case.

3.2.2 | GP modeling of WO errors

Because the parametric 3D odometry model gives reasonable good estimates under reliable ground-traction conditions (see Section 3.1), a zero-mean function better models the odometry residual (the difference between estimates and ground truth). The dynamic system equation for the GP is

$$\boldsymbol{s}(k+1) = \boldsymbol{s}(k) + g(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k)) + f(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k)) \tag{30}$$

where the function g describes the change in state given by the parametric 3D odometry model and the function f is modeled by the GP and describes the odometry residual. The training set D for the GP is a sequence of observed states and inputs. They are used to learn the parameters of the nonlinear function f. The input training data are of the form $\boldsymbol{x}_k = [\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k)]$ and the output residual $\boldsymbol{y}_k = \boldsymbol{s}(k+1) - \boldsymbol{s}(k) - g(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k))$.

The purpose of the GP model is to serve in modern SLAM systems between the frontend (sensor data) and backend (optimization) as depicted in Figure 7. The input vector $\breve{\boldsymbol{u}}(k)$ is composed by pitch and roll orientation angles, joints position and speed, angular velocities, and accelerations measured by the IMU. The GP estimates a 3D output for the residual and its uncertainty in each direction of motion (i.e., odometry error). A different set of collected data, test data, is used to evaluate the accuracy of the estimated residual.

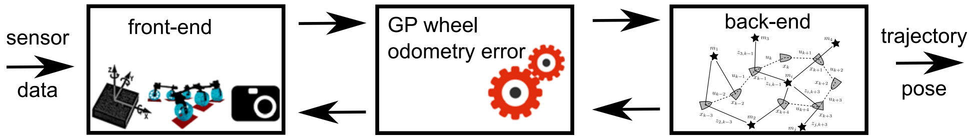


FIGURE 7 Anatomy of a modern SLAM system with the frontend, the backend, and the proposed GP regression model in between

3.3 | Adaptive graph sparsity

In the context of localization and mapping, adaptivity is the act of adjusting SLAM to the system demands. More specifically, in this work, it is the capability to adjust the localization and mapping to the current navigation demands. The adaptivity influences the frontend and backend during operations. A method adapts the frontend to overcome poor WO. Another method adapts the selection of graph nodes or keyframes at the backend. Therefore, two strategies are selected. The former method is an active perception strategy to compute visual odometry. The second method is a keyframe selection policy based on feature matching to determine graph sparsity. It defines graph sparsity or graph node selection with an adaptive information gain.

The rate of adaptation (RoA) is defined here as how fast the system reacts to changes of the WO error. The RoA is the response of the system to change or adapt to new circumstances. The RoA can be a constant function chosen at a design phase. However, the RoA does not necessarily need to be constant with a fixed gradient. The gradient can increase or decrease with the value in the odometry error. A faster adaptation is a higher RoA and here means higher gradient or slope function. Consequently, one solution can be that the bigger the odometry error, the faster the SLAM system adapts. As an illustration, Figure 8 depicts three possible equations to define RoAs. A linear curve maintains a constant slope along the plot which defines a linear RoA. The exponential and quadratic RoA have similar gradient for small odometry errors, but the exponential curve increases the RoA at higher values of the odometry error. The quadratic equation is the chosen method for the adaptive SLAM, as Equations (31) and (32) have a quadratic gradient (explained later). The reason is to maintain a sufficient frequency in visual odometry without drastically increasing the computational load during high WO errors. The choice for one or another equation defines the RoA, which affects the visual odometry load and sparsity of the graph. Ultimately, RoA affects how fast the SLAM system reacts to errors introduced by the WO.

3.3.1 | Adaptive visual odometry

Selecting the image frame rate adapts the visual odometry computational load. A high frame rate produces unnecessary computational load when the odometry model correctly estimates the pose. However, a low frame rate might cause inadequacy in feature tracking, which implies to trigger *relocalization* to maintain a functional visual odometry. Relocalization computes by an extensive search in the features space of the last keyframe. The balance between an excessive visual odometry load and loss of tracking is given in Equation (31). Therefore, the sample period τ to compute visual odometry is given by

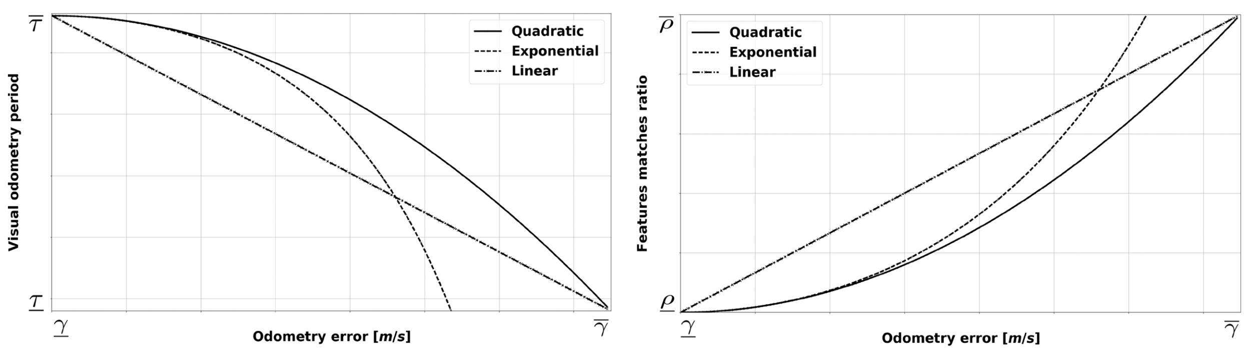
$$\tau = \frac{\underline{\tau} - \overline{\tau}}{(\overline{\gamma} - \underline{\gamma})^2} \|\check{\mathbf{y}}_*\|^2 + \overline{\tau} \tag{31a}$$

$$\check{\mathbf{y}}_* = \mathbf{y}_* + \Sigma_{\mathbf{y}_*} \tag{31b}$$

where $\underline{\overline{\tau}} = [\underline{\tau}, \overline{\tau}]$ are the minimum and maximum period allowed by the camera sensor, $\underline{\overline{\gamma}}$ are the minimum and maximum of the accepted WO error, according to a threshold explained later, and $\mathbf{y}_*$ is the predictive mean of the WO error given by the Gaussian process $GP_\mu(\mathbf{x}_*, D)$ with variance $\Sigma_{\mathbf{y}_*} = \mathrm{diag}(GP_\Sigma(\mathbf{x}_*, D))$. Figure 8a depicts the quadratic curve of Equation (31).

3.3.2 | Adaptive node selection

Keyframes are the selected frames to take as reference in the feature tracking and local bundle adjustment. They are also incorporated as nodes in the graph for global optimization. A distance criterion is the simplest method to select a keyframe. However, distance does not perform well during drastic movements, bumps, or turning maneuvers. The criterion to select a keyframe is based on feature tracking instead of traversed distance. The criterion establishes a minimum number of



(a) (b)

FIGURE 8 Comparison of three possible equations for the RoA (a) visual odometry period (b) feature matching ratio

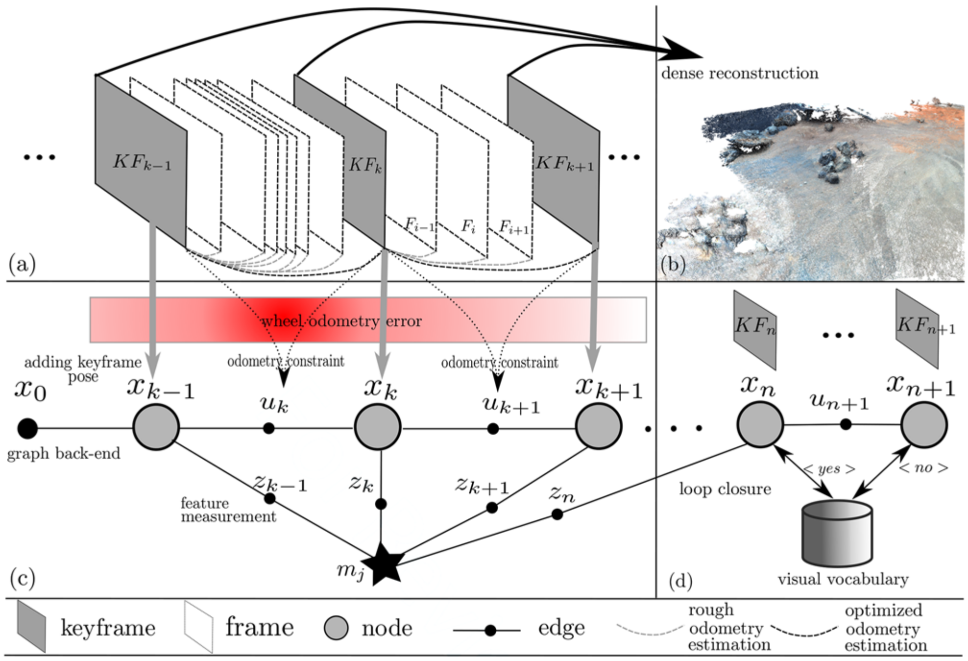


FIGURE 9 Anatomy of the adaptive localization and mapping system: (a) visual odometry, (b) dense map reconstruction, (c) optimization backend, and (d) loop closures engine

features which must change in the scene to insert a new keyframe, and consequently a new node in the graph. The following equation defines the adaptive strategy related with the predicted WO error, as

$$\rho = \frac{\overline{\rho} - \underline{\rho}}{(\overline{\gamma} - \underline{\gamma})^2} \, \|\check{\mathbf{y}}_*\|^2 + \underline{\rho} \tag{32a}$$

$$\check{\mathbf{y}}_* = \mathbf{y}_* + \Sigma_{\mathbf{y}_*} \tag{32b}$$

where $\underline{\overline{\rho}}$ are the minimum and maximum ratio of feature matching, respectively. $\underline{\overline{\rho}}$ defines a [0,1] interval where 0 means no features overlap and 1 defines overlap of all the features between consecutive keyframes. A value near zero reduces the number of features with the resulting loss in tracking. On the contrary, a high ratio imposes a new keyframe for each new image frame. The adaptive feature matching defines a minimum visual change depending on wheel traction performance. Figure 8b depicts the quadratic curve of Equation (32). A novel method to relate the performance of the WO to graph node selection in the backend.

3.3.3 | WO error threshold

The thresholds for the WO error $\underline{\overline{\gamma}}$ are defined as a percentage of the commanded rover velocity. The threshold changes between a hypothetical perfect odometry solution, 0% slippage, to a worst case scenario with no forward movement, 100% slippage. In addition, two intermediate threshold values are selected for evaluation. Generally, a successful localization system for planetary rovers has an error between 1.0% and 2.5% of the total distance traveled. For this reason, the selected thresholds in velocity are 10% and 25%, respectively. These two values with the lowest and upper bounds are the threshold of interest. To summarize, a total of four thresholds are selected, 0% to compute SLAM without adaptiveness, 100% to analyze whether adaptive SLAM could lose the trajectory in the worst case scenario and two intermediate values, 10% and 25%. The following equation defines the accepted WO error according to such slippage threshold by

$$\underline{\overline{\gamma}} = [\underline{\gamma} = 0, \overline{\gamma} = \hat{p}v] \tag{33a}$$

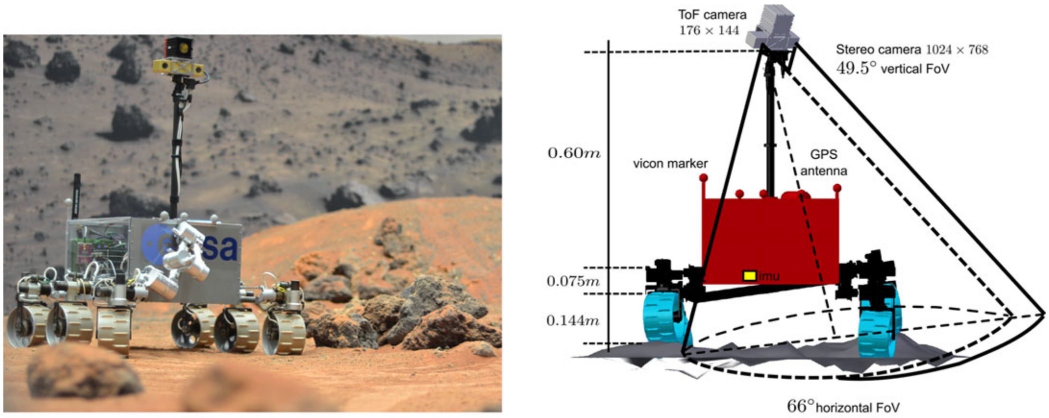
$$\hat{p} = 0, 0.10, 0.25, 1.0 \tag{33b}$$

where $\hat{p}$ is the desirable percentage among those four values explained above and v is the commanded rover velocity.

3.3.4 | Adaptive SLAM architecture

The adaptive localization and mapping is depicted in Figure 9. The diagram shows four different blocks, which comprise the SLAM architecture. The block (a) describes the image features tracking module, which computes a *rough* visual odometry using the initial guess from the delta pose estimated by the WO. The visual tracking uses a constant velocity model to estimate the rover displacement. The visual tracking computes Oriented FAST and rotated BRIEF (ORB) features, which are multiscale Features from Accelerated Segment Test (FAST) keypoints and Binary Robust Independent Elementary Features (BRIEF) visual descriptors (Rublee, Rabaud, Konolige, & Bradski, 2011). The visual odometry uses RANdom SAmple Consensus (RANSAC) to remove outliers. A local bundles adjustment optimizes the delta pose, depicted as *optimized odometry estimation* lines in Figure 9. The visual odometry rate changes depending on the WO error using Equation (31), depicted in a red-colored bar in Figure 9. The block (b) performs stereo dense reconstruction with the stereo camera pair and builds a local dense map. Stereo dense uses an efficient large-scale stereo-matching technique, Libelas* (Geiger, Roser, & Urtasun, 2010). A local map is

* Library for Efficient Large-scale Stereo Matching, http://www.cvlibs.net/software/libelas



(a) (b)

FIGURE 10 The ExoMars Test Rover: (a) during the test at the Planetary Robotics Laboratory at ESA and (b) sensors configuration and dimensions

processed in an EnviRe† item and stored in an EnviRe graph. EnviRe is a representation model that facilitates real-world reconstruction using a connected directed graph. The third block, labeled (c), is the optimization backend composed of the graph and the smoother which estimates the final pose and the global map. The selection of keyframes uses Equation (32). This module is based on the G2O‡ graph SLAM from Kummerle et al. (2011). The EnviRe graph uses the optimized constraints from the G2O graph and optimally combines the local maps in the EnviRe items. The connection between both graphs, G2O and EnviRe, is performed with pointers to make the computation more efficient. Finally, the block (d) matches distinguishable features using a visual vocabulary created offline from ORB features. The module estimates loop closure constraints based on bag of words place recognition using DBoW§ from Galvez-Lopez and Tardos (2012). Visual words are a discrete version of the descriptor space, known as the visual vocabulary. Every time a new keyframe is inserted, a visual word vector is computed and compared with the existing visual words to identify a revisited place.

4 | EXPERIMENTS

To evaluate the feasibility of the approach, a set of field experiments have been performed with a representative research platform. The experiments cover the 3D odometry model, the odometry error prediction using the GP regression, and ultimately the adaptive SLAM.

4.1 | ExoMars test rover

The ExoMars Test Rover (ExoTeR) is the selected platform to conduct the experiments. ExoTeR is representative of a Mars-like planetary rover for research activities at the European Space Agency (ESA). The chassis resembles in scale the ExoMars rover mobility configuration (Poulakis et al., 2015). ExoTeR is shown in Figure 10, the rover has six wheels, a mass of 25 kg, and a ground clearance of 20 cm. Track width and wheel base are 62 and 53cm, respectively. Six-wheel rovers have a significantly better obstacle negotiation capability than four wheels. The triple bogie system is able to keep the rover body almost at a horizontal position when driving over rocks. ExoTeR's locomotion formula is $6+6+4$, which includes six driving, six walking, and four steering motors. There are no steering motors in the middle wheels, making the chassis lighter but preventing the rover to crab. The wheel walking motors are part of the deployment mechanism to stow the rover in a compact configuration. The wheel walking motors also serve to investigate peristaltic motion modes as the wheel walking (Azkarate et al., 2015).

The sensor suite includes a stereo camera pair, an IMU, and actuator encoders and potentiometers at the joints. The STIM300 IMU measures accelerations and angular velocities which are filtered using a Kalman-based attitude and heading reference system (AHRS). The stereo camera setup has a baseline of 12 cm with a resolution of 1024×768 pixels per image. ExoTeR runs the Rock¶ real-time framework on a Linux operating system installed on a Core2 Duo at 1.86 GHz. The inertial system module, which runs the AHRS, provides the orientation and inertial readings required by the odometry model described in Section 3.1. Joint positions and velocities are collected by the sensor drivers and dispatched together to the odometry model. Angular velocities $[\dot{\phi}_B, \dot{\theta}_B, \dot{\psi}_B]$ and joints measurements $\dot{\boldsymbol{q}}$ are sensed rate quantities in Equation (5). The vector $\dot{\boldsymbol{q}}$ has dimension 25×1, six for driving, six for wheel walking, four for steering, and nine passive joints for the triple bogie. Linear velocities $[\dot{x}_B, \dot{y}_B, \dot{z}_B]$, slip vector $\dot{\epsilon}$, and contact angles $\dot{\boldsymbol{\delta}}$ are nonsensed quantities. The complete vector $\dot{\boldsymbol{p}}$ has a dimension of 49×1. A system of equations is obtained by rearranging nonsensed (left-side) and sensed (right-side) quantities as in Equation (6). ExoTeR has a vector $\boldsymbol{v}$ of sensed quantities with dimension 28×1 and a vector $\boldsymbol{b}$ of nonsensed quantities with dimension 27×1. The solution of the parametric model for 3D odometry is obtained by finding the least squares solution to Equation (7) and setting the weighting matrix $\mathcal{W}$ to the calculations from Equation (21). These computations run in a

† Environment Representation, https://github.com/envire

‡ A General Framework for Graph Optimization, https://github.com/RainerKuemmerle/g2o

§ Hierarchical Bag of Words library, https://github.com/dorian3d/DBoW2

¶ The Robot Construction Kit, http://www.rock-robotics.org

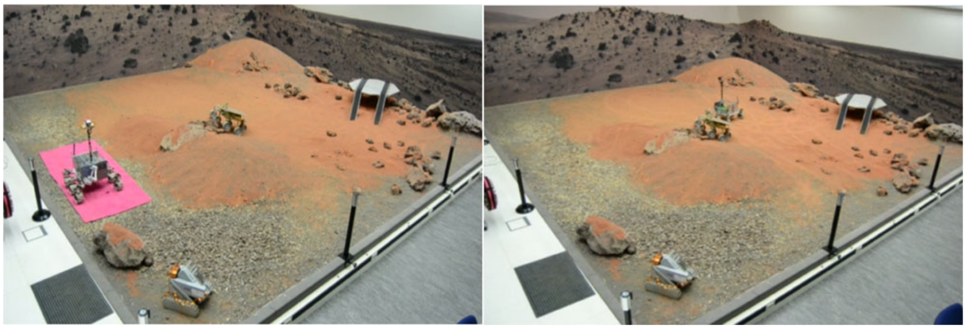


FIGURE 11 Photographs during the tests for collecting sensory data with ExoTeR on the Mars-like terrain at ESA's Planetary Robotics Laboratory. The training data are used to learn the kernel function of the GP nonlinear regression

dedicated Rock task and deployed as a single process on the onboard operating system.

Once the parametric model for the 3D odometry outputs the estimated delta poses, a dedicated process computes the error model. First, the learning of the nonparametric odometry error model is computed offline. GP learning is performed by finding the set of hyperparameters (see Section 3.2). The training data D are collected from driving the ExoTeR in a number of relevant experiments. The training data inputs X are the pitch and roll angles (attitude) computed by the AHRS, the filtered inertial gyroscopes and accelerometers, and the position and speed joint measurements. To reduce the input dimensionality, the wheel walking joints are not used for training the GP. The training outputs Y are obtained by comparing the estimated delta pose output from the 3D odometry model with the delta pose from a ground truth measurement, $\boldsymbol{s}(k+1) - \boldsymbol{s}(k)$. Four training tests were conducted in a representative scenario. Figure 11 shows some screenshots of the filmed laboratory setup. The learned kernel encapsulates in a single Rock task and predicts the current odometry error for each new input data $\boldsymbol{x}_*$ that arrives at the communication port.

The adaptive SLAM runs on a multithreading architecture in a Rock single deployment or process. One thread runs the stereo visual odometry, a second thread runs the graph optimization, a third one executes the loop closing, and a final one computes the dense map. SLAM receives the delta pose estimated by the WO model with the odometry error as uncertainty in a 6×6 covariance matrix. The stereo camera driver delivers image pairs to the tracking module and to the dense map reconstruction task. The quadratic equations, Equations (31) and (32), adapt the SLAM solution to the current navigation demands. The equations take the GP predictions to adapt the visual odometry processing and keyframe selection strategy. The minimum and maximum camera periods are $\underline{\overline{\tau}} = [0.4\text{s}, 2.0\text{s}]$ in Equation (31). It means that under poor traction circumstances the visual odometry is triggered up to five times faster than during good WO conditions. The boundaries of the feature matching ratio change from 30%, under good traction conditions to 75% overlapping when the odometry error is maximum. This gives a parameter $\underline{\overline{\rho}} = [0.3, 0.75]$ for Equation (32).

4.2 | Mars-like terrain

ESA's Planetary Robotics Laboratory comprises, among other facilities, a 9 m $\times$ 9 m Mars-like terrain that resembles a planetary surface. Around the terrain, a set of 12 infrared emitting and sensing cameras are mounted to the walls, which sense reflective markers mounted on the rover platform. These cameras are part of the Vicon-tracking system, which can deduct and track position and orientation of objects equipped with such reflective markers. The Vicon system tracks rover's pose with centimeter accuracy in position and few tenths of a degree in attitude. Two experiments, *Test#1* and *Test#2*, are conducted to evaluate the feasibility of the approach. The objective is to evaluate each step of the methodology in a relevant scenario.

4.2.1 | Wheel odometry

WO performs a dedicated dead reckoning calculation to estimate the quality of a full 3D odometry model. The pose results are compared with a classical skid odometry implementation. Skid odometry is a planar kinematics approach, and it does not model the full kinematics of the chassis. The skid odometry takes rover wheelbase and track width with the wheel radius to compute the displacements. Instead, a 3D odometry model allows to identify each contact point and compute a weighted solution to estimate the delta pose. The delta pose is computed by minimizing a least squares error. The odometry trajectories produced by the dead reckoning pose from both odometry models are shown in Figure 12. They are compared with the ground truth from the Vicon system, and resulting errors are given in Table 1. It is appreciable that any dead reckoning accumulates errors unbounded, but a 3D odometry model performs more accurately in both tests. The final error is not always the maximum error, due to the circular shape of the trajectory. The RMSE is the most accepted metric to evaluate the performance and the value is lower for the 3D odometry model. In addition, *Median E.* defines the statistical median of the error, *Max E.* depicts the maximum error in meters along the path, *Final E.* is the error at the end of the trajectory and *Max E.* (%) is the percentage error per distance traveled by taking the maximum error. The *Distance* is the total distance of the traversal in meters.

4.2.2 | GP error model

GP predictions are compared with the *true* odometry error and depicted in Figure 13 for Test#1 and Figure 14 for Test#2. The *true* odometry error is computed by comparing the delta poses from the 3D odometry model and the ground truth data. The trajectory is depicted with the contour map to facilitate the interpretation of the results.

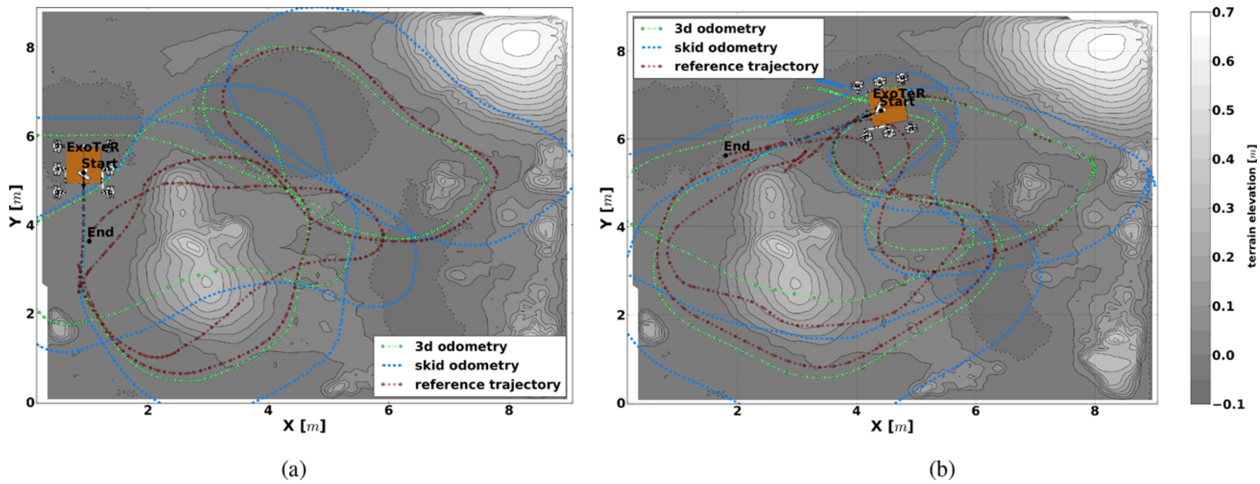


FIGURE 12 Trajectories of the 3D odometry and skid odometry model with ExoTeR for (a) Test#1 and (b) Test#2 at the ESA's Planetary laboratory

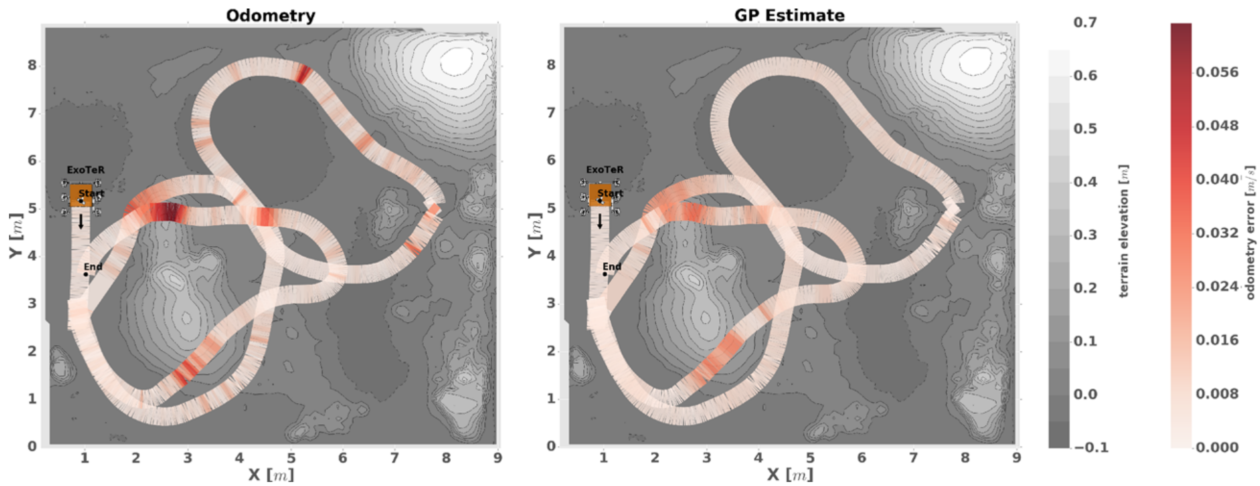


FIGURE 13 Ground truth odometry residual and GP estimate for Test#1. Traversed trajectory and digital elevation map (DEM) of the Mars-like testbed are depicted together with the odometry error (red color bar)

TABLE 1 Mars-like terrain pose results for the different odometry models with ExoTeR

Odometry model	Test case	RMSE (m)	Median E. (m)	Max E. (m)	Final E. (m)	Max E. (%)	Distance (m)
Skid odometry	Test#1	1.81	1.18	4.34	3.25	9.54	45.5
3D odometry	Test#1	1.14	0.38	2.96	2.84	6.51	45.5
Skid odometry	Test#2	1.14	0.78	2.75	2.20	5.50	50.0
3D odometry	Test#2	0.87	0.29	2.64	2.14	5.20	50.0

ExoTeR is commanded at a nominal velocity of 6 cm/s, and most of the odometry errors are due to poor traction on high slope areas or sandy terrain. It can be noted that for most of the driven areas the GP correctly predict the errors in odometry. The highest odometry error in Test#1 occurred when traversing the sandy dune located at the middle left part of the testbed located around point (2.5, 5.0) in the map of Figure 13 (at time 21 h and 03 min in Figure 15). The rover almost encounters 100% slippage (no forward movement) at that location. ExoTeR experiences a similar behavior in Test#2 but at a different location on the same dune, a rocky area with a high slope angle located around point (2.5, 2.0) in the map of Figure 14 (at time 18 h and 10 min in Figure 16). It is worthwhile to notice how the GP predicts different values even when passing the same area at different instances. Adjacent terrain areas might have similar soil properties, but the rover behaves differently in traction performance. This gives the understanding that odometry performance is dependent not only on the soil characteristics but also on the rover velocity, acceleration, chassis configuration, and attitude. As a result, the interaction of the rover with the terrain alters those propioceptive values depending on soil characteristics. The GP model uses exactly the proprioceptive information to predict the nonparametric model of the error, resulting in different predictions, even when driving at the same spot.

Figure 15a compares the odometry velocity with the ground truth velocity, and Figure 15b shows the truth and predicted odometry error

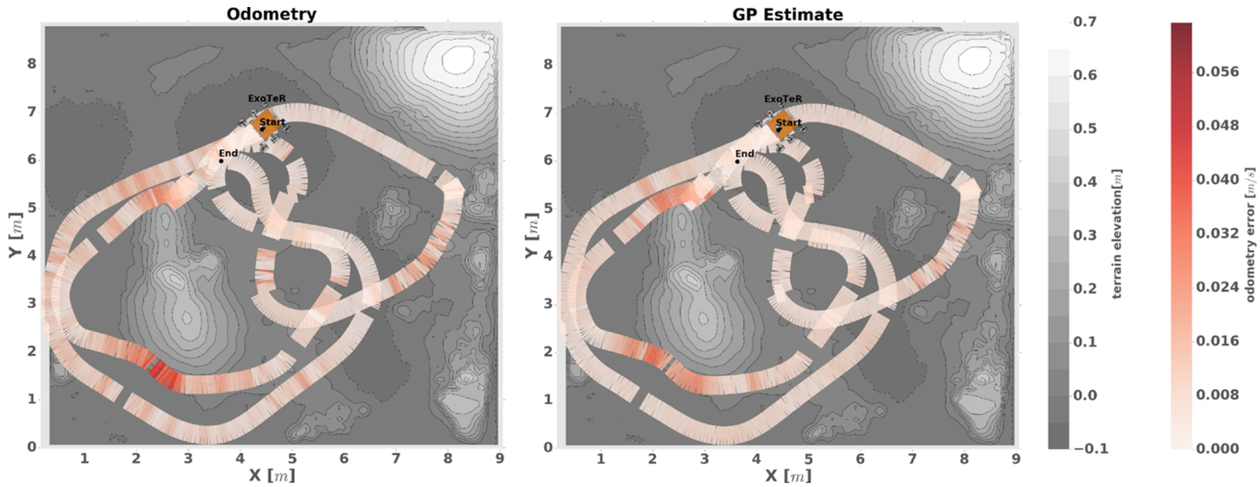
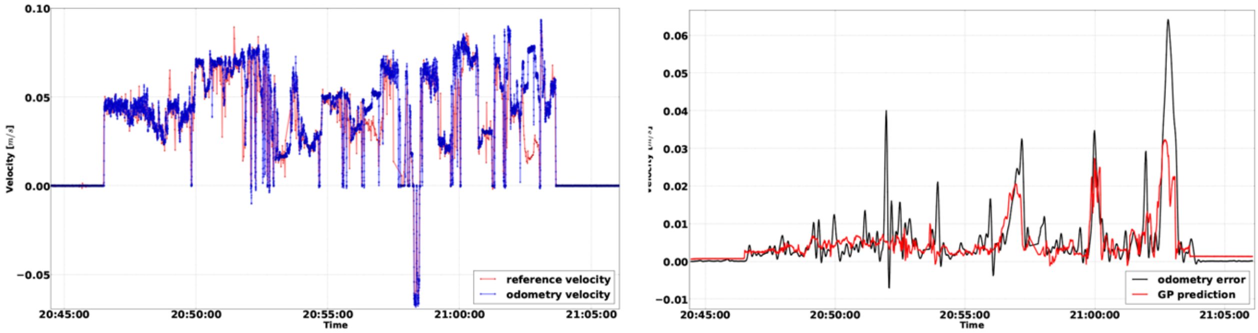
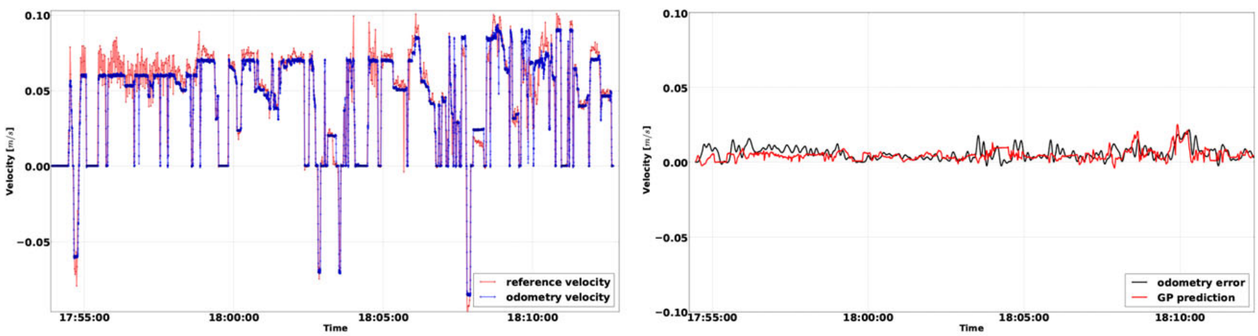


FIGURE 14 Ground truth odometry residual and GP estimate for Test#2. Traversed trajectory and DEM of the Mars-like testbed are depicted together with the odometry error (red color bar)



(a) (b)

FIGURE 15 Test#1 results for ExoTeR: (a) odometry velocity and ground truth and (b) *truth* and GP-estimated error



(a) (b)

FIGURE 16 Test#2 results for ExoTeR: (a) odometry velocity and ground truth and (b) *truth* and GP-estimated error

in a single graph with respect to time. The predicted odometry error follows the truth odometry error in most part of the experiment. The GP model predicts a maximum of 70% slippage, which correspond to 4 cm/s odometry error when driving the sandy dune area previously mentioned for Test#1 (see Figure 15b at time 21 h and 3 min). Most of the GP modeling occurs in the covariance function described in Equation (26), which tends the predictions toward a zero mean function. This is because GPs assume a zero mean prior in Equation (23), and the mean function $GP_{\mu}(X_{*}, D)$ tends toward zero as the distance between samples of the training data increases. This makes the choice of training data important, with a diverse set of values. The GP accurately predicts the transitions in slippage, which are more valuable than the absolute error number. It is actually the important aspect since the objective of the GP regression model is to detect slippage, and not correct it, to adapt the SLAM solution in a successive localization and mapping step.

4.2.3 | Adaptive SLAM

The purpose is to evaluate the performance of the adaptive SLAM in the Mars-like terrain. GP prediction is the trigger mechanism

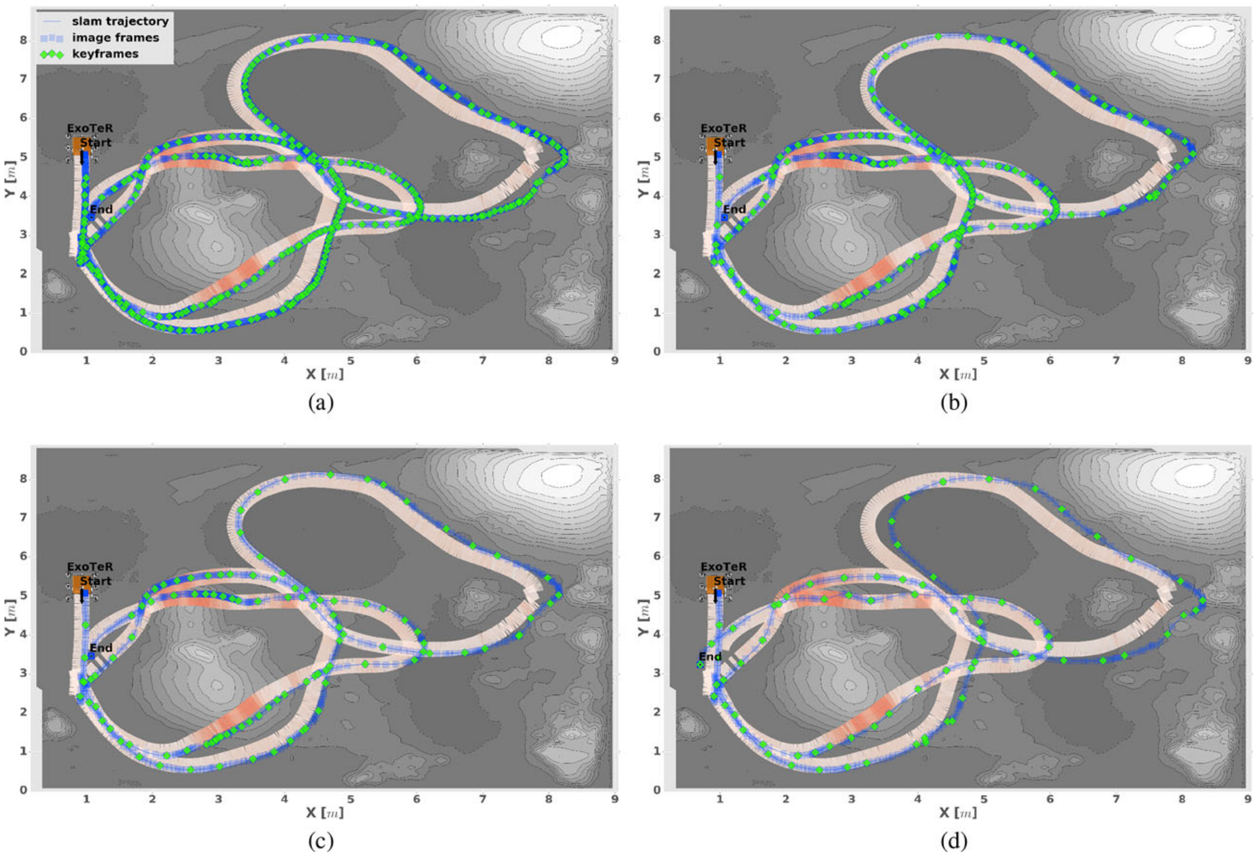


FIGURE 17 Adaptive localization and mapping for different odometry error thresholds for the data in Test#1: (a) SLAM w adaptivity at $\underline{\tau} = \overline{\tau} = 0.4$s and $\underline{\rho} = \overline{\rho} = 0.75$, (b) adaptive SLAM 10% [$\overline{\gamma} = 0.0063$ ms^{-1}], (c) adaptive SLAM 25% [$\overline{\gamma} = 0.016$ ms^{-1}], and (d) adaptive SLAM 100% [$\overline{\gamma} = 0.063$ ms^{-1}]

TABLE 2 ExoTeR's pose results for the different SLAM schemes running Test#1

Adaptiveness	Number of frames	Number of keyframes	RMSE (m)	Max E. (m)	Final E. (m)	Max E. (%)	Distance (m)	Number ofloops
Without adaptivity	2,619	291	0.14	0.45	0.063	0.98	45.5	3
10% threshold	1,206	181	0.15	0.50	0.058	1.09	45.5	2
25% threshold	716	111	0.15	0.51	0.064	1.12	45.5	2
100% threshold	480	78	0.34	0.85	0.47	1.87	45.5	1

for the adaptive visual odometry. Equation (31) decides whether an image frame has to be included in the VO tracking. In the meantime, delta poses are preintegrated in an inter frame dead reckoning process. When the adaptivity policy decides to trigger the image frame, the preintegrated delta pose is used to track the correspondence features in the image. Afterwards, a more accurate and refined delta pose is computed by the VO module. Concurrently, Equation (32) separately adapts the necessity to incorporate a keyframe in the SLAM backend and perform an optimization step in the backend.

The threshold is a percentage of the nominal rover velocity, resulting in a maximum odometry error or slippage. For instance, a 25% slippage threshold in a rover driving at 0.063 m/s (6.3 cm/s) results in $\overline{\gamma}$ equal to 0.016 m/s. Figure 17 depicts the localization and mapping trajectory of four different cases depending on the maximum slippage allowed during the traversal. The first case shows the solution by running SLAM without adaptivity, 0% slippage threshold. This case uses the maximum amount of sensory data and resources available and turns out to be a fixed classical SLAM system with an image frame rate $\underline{\tau} = \overline{\tau} = 0.4$s and a constant feature matching ratio $\underline{\rho}$ of 75% overlapping. The other three cases show the SLAM solution for 10%, 25%, and 100% slippage threshold with $\overline{\gamma}$ equal to 0.0063, 0.016, and 0.063 m/s, respectively. The values are used in the adaptivity equations to perform different adaptiveness in the SLAM. Note in Figure 17 that the image frames and keyframes have a high spatial frequency in areas where the rover encounters poor traction performance. There is a significant correlation between the odometry error and structure of the graph, to overcome the poor estimation of odometry, as designed. The adaptivity threshold has an impact on the quality of the estimates. Table 2 summarizes the error value of each approach. Note that the number of frames and keyframes reduces as the threshold increases. It can also be noted how the number of keyframes is almost

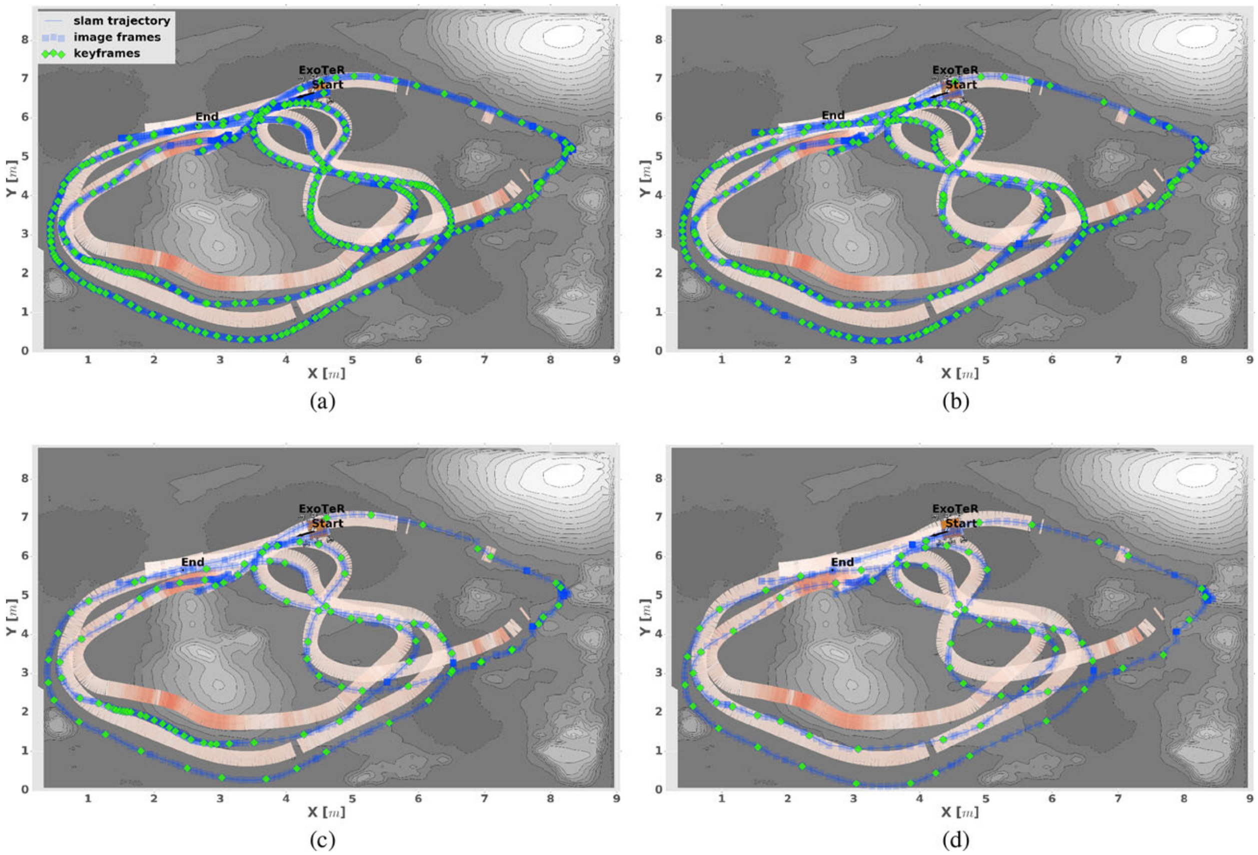


FIGURE 18 Adaptive localization and mapping for different odometry error thresholds for the data in Test#2: (a) SLAM without adaptivity at $\underline{\tau} = \overline{\tau} = 0.4$ s and $\underline{\rho} = \overline{\rho} = 0.75$, (b) adaptive SLAM 10% [$\overline{\gamma} = 0.0063$ ms^{-1}], (c) adaptive SLAM 25% [$\overline{\gamma} = 0.016$ ms^{-1}], and (d) adaptive SLAM 100% [$\overline{\gamma} = 0.063$ ms^{-1}]

TABLE 3 ExoTeR's pose results for the different SLAM schemes running Test#2

Adaptiveness	Number of frames	Number of keyframes	RMSE (m)	Max E. (m)	Final E. (m)	Max E. (%)	Distance (m)	Number of loops
Without adaptivity	2849	299	0.30	0.65	0.13	1.30	50.0	2
10% threshold	1837	245	0.31	0.72	0.16	1.44	50.0	2
25% threshold	797	109	0.36	0.78	0.24	1.56	50.0	2
100% threshold	569	71	0.50	1.21	0.20	2.42	50.0	2

three times less in the 25% slippage (111 keyframes) than in the fixed SLAM without adaptivity 0% slippage (291 keyframes). However, the reduction of keyframes does not have a significant penalty in the error because any of the computed errors are three times higher than the solution without adaptivity. From 0% to 25% slippage, only an increase of 0.14% in the percentage error per distance traveled is noticeable (see Table 2). This is because Equation (31) reduces image frames when WO performs reasonably well, and Equation (32) eliminates redundant keyframes resulting in a sparser graph.

The same thresholds are selected to study the influence in the adaptivity and the impact on the quality of the solution in Test#2. Figure 18 depicts the localization and mapping trajectory of four different cases depending on the maximum slippage allowed during the traversal. It is appreciable in the figures that the keyframes have a higher spatial frequency in turning maneuvers than in straight parts of the trajectory. The keyframe selection criteria previously mentioned in Section 3.3 applies as in Test#1. The criterion is based on the matches overlapping ratio instead of distance traveled, which generates more keyframes when ExoTeR is turning. Equation (32) adapts the density in areas with poor WO results and reduces the number of keyframes when ExoTeR drives with good WO conditions. It can also be noted that the ground truth trajectory interrupts at the top-right area of the maps. The Vicon systems could not properly reconstruct the rover pose at that locations due to partial occlusion of the reflective markers. ExoTeR does not encounter, in terms of absolute numbers, as much slippage as in Test#1. The rover does not reach 100% slippage along the Test#2 drive. Nevertheless, slippage always occurs on sandy terrains with metallic wheels and as long as there is a certain amount of slippage the GP model is able to predict poor traction performance. Such information is interpreted by the adaptive SLAM and adapts the parameters online as depicted in the plots. Table 3 summarizes the error value depending on the adaptive criteria.

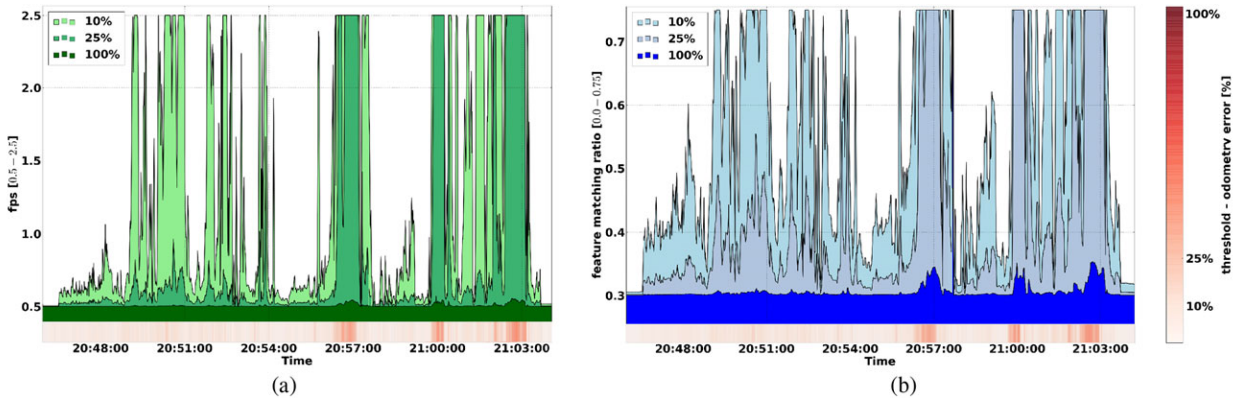


FIGURE 19 Adaptive evolution for Test#1 of (a) the image processing at $1/\tau$ frames per second and (b) feature matching ratio ρ for the 10%, 25%, and 100% scheme

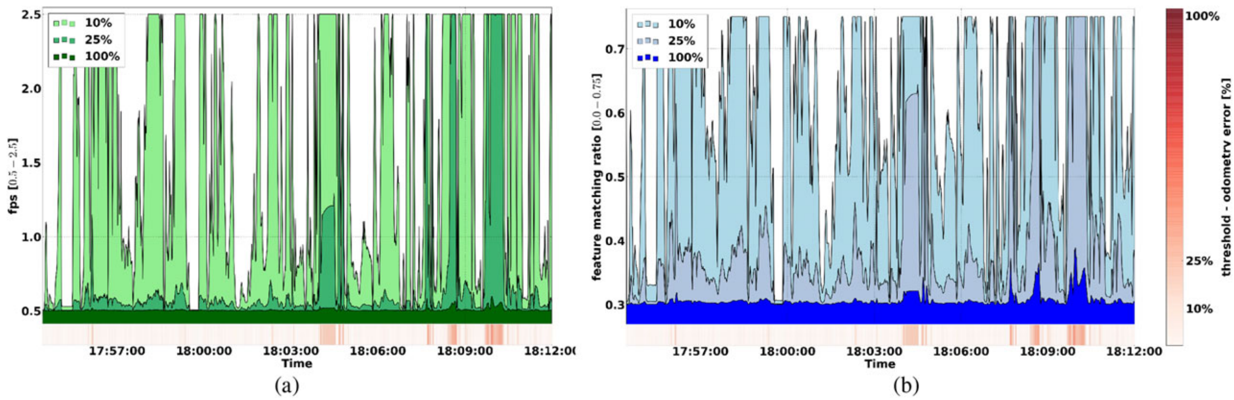


FIGURE 20 Adaptive evolution for Test#2 of (20) the image processing at $1/\tau$ frames per second and (20) feature matching ratio ρ for the 10%, 25%, and 100% scheme

It is important to mention that the camera during the Test#2 is looking forward and not being tilted as in Test#1. It penalizes the accuracy in visual odometry but increases the robustness in the loop closure. The number of loop closures decreases in the Test#1, from a maximum of three in the SLAM without adaptivity to one detected loop closure at 100% slippage threshold. The number of loop closures in the Test#2 is two and stays constant independently of the adaptiveness threshold. The reason is that the camera is looking 30° downwards in Test#1. The lesson learned from these two experiments in relation to loop closure is that adaptivity negatively affects the number of detected loops, but it is not a key factor. The camera configuration has a stronger influence on the loop closure. Generally, a camera looking 30° downwards produces a more accurate delta pose visual odometry and terrain maps, whereas a camera looking forward increases the loop closure robustness due to a wider view of the scene. This argumentation also depends on the scene and available persistent visual features.

The number and spatial distribution of frames and keyframes significantly change along the trajectory. This is due to the variation of the frame period and feature matching ratio. Figures 19 and 20 show the time evolution of these values correlated with the odometry error (red color bar located at the bottom of the plot). A significant variation of the area covered by the curves can be noticed, which depends on the selected threshold. The threshold selected at each experiment is labeled on the left side of the graph together with the color bar of the legend. The steep variation in the curves is related with two factors, the odometry error predicted from the GP model and the quadratic function described in Equation (31) for the frame period and Equation (32) for the feature matching ratio. The area covered by the plots in Figures 19 and 20 can be interpreted as the amount of computational effort required for the adaptive SLAM per each adaptive threshold. With this interpretation, an adaptive SLAM with 0% slippage threshold covers the complete space (resources) and the adaptive SLAM with 100% slippage threshold the smallest area in the plot (dark blue and dark green). In between, there are the 10% and 25% selected thresholds as depicted in the plots.

SLAM not only estimates the rover pose but also creates a map of the surroundings. The dense map reconstruction (see Figure 9) takes the stereo frame images and performs a disparity image to estimate the depth per pixel. The depth value is combined with the color information to produce a colored point cloud. The collection of point clouds, one point cloud per frame pair, is locally merged at the latest keyframe. Afterwards, the set of local maps is combined together using the EnviRe graph structure to generate the global map of the environment. It is expected to have some distortion and inaccuracies in the map as the adaptivity threshold increases. Figure 21 depicts the resulting error map for the adaptive localization and mapping approach with different thresholds. An error metric is defined here, which is the absolute Euclidean distance of each point in the voxel to the ground

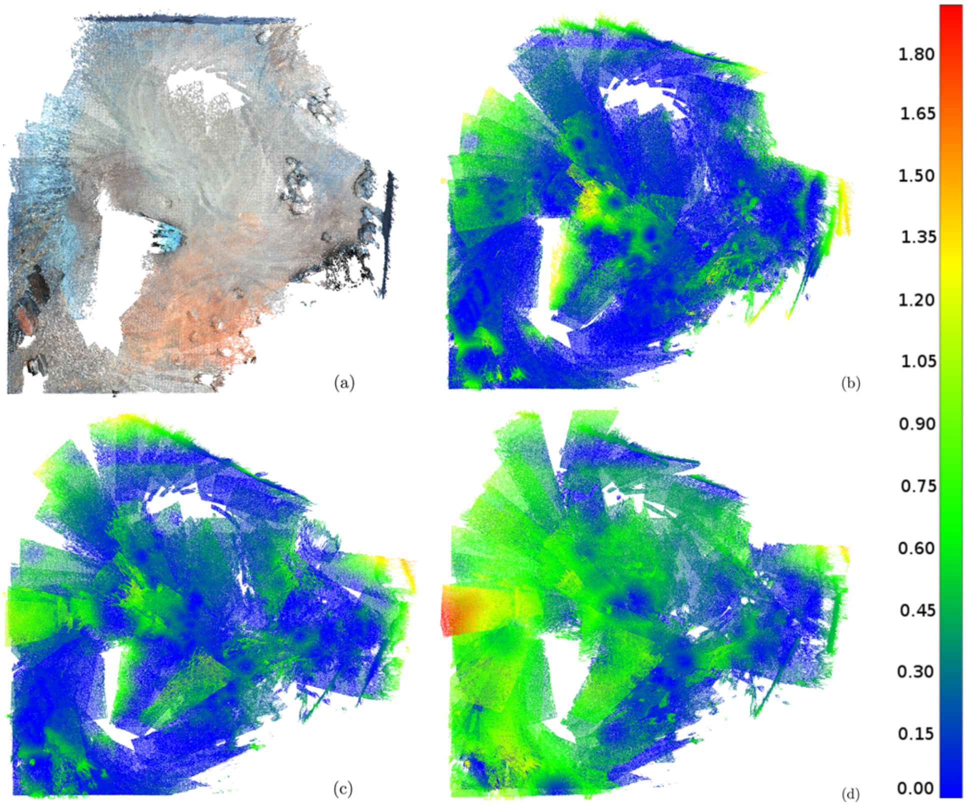


FIGURE 21 Adaptive map quality for different odometry error thresholds in Test#1: (a) colored reference dense map reconstruction, (b) error map with respect to the reference map for 10% threshold in adaptiveness, (c) error map for 25% threshold in adaptiveness, and (d) error map for 100% threshold in adaptiveness. The colored error legend in meters is depicted at the right side

truth map. The ground truth map is created by the same dense map reconstruction technique but using the ground truth position (e.g., the Vicon system). It is worth noticing how the map error is affected by the threshold. The heading drift is the main factor for point cloud misalignment (ground truth vs. SLAM). The drift originates from the IMU and increases as the threshold in SLAM increases. Also, the error does not uniformly affect all points in each local map. The error in the point cloud is directly proportional to the distance from the sensor focal point. Therefore, the pose error affects points further from the camera more than points closer to it. So, in mapped areas far from where the rover has been the Euclidean distance error is higher. The biggest map error of 1.92 m with respect to the ground truth map is encountered by the adaptive SLAM with 100% slippage threshold (see Figure 21) at the middle left zone of the map. This is coincident with the highest slippage occurring at the sandy dune, denoting the penalty of odometry errors in the quality of the final map.

4.3 | Decos terrain

The Decos terrain is located in Noordwijk (The Netherlands) in proximity to the European Space Research and Technology Center (ESTEC). Figure 22 shows the ExoTeR located at the base camp with the necessary equipment and the remote control station. It is a terrain imitating a rocky Mars environment with one prominent crater in the middle and a smaller ground depression next to it. The test zone with

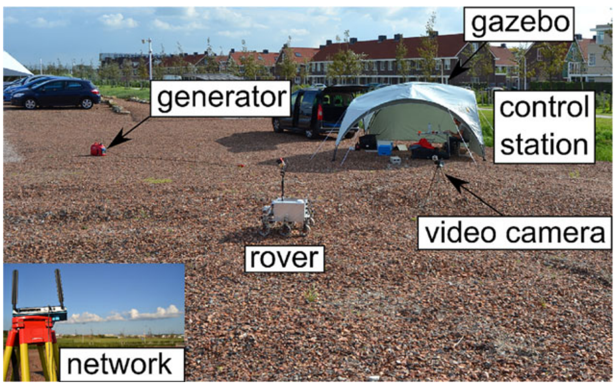


FIGURE 22 Decos experiment: rover control station at the experimental environment for the Decos terrain

the dimension of 50 m × 80 m consists of medium size rocks and red broken bricks. ExoTeR is equipped with a GPS antenna to acquired ground truth measurements along the drive. An autonomous drone is used to capture high-resolution aerial images and produce a DEM of the target zone. The generated DEM is used to render the ground truth map and contour lines in the resulting figures from the experiment.

4.3.1 | Wheel odometry

ExoTeR performs an eight shape trajectory connecting two consecutive circles: a bigger circle along the crater and another circle border-

TABLE 4 Decos terrain pose results for the different odometry models with ExoTeR

Odometry model	RMSE (m)	Median E. (m)	Max E. (m)	Final E. (m)	Max E. (%)	Distance (m)
Skid odometry	7.90	6.17	16.44	1.17	9.28	177
3D odometry	3.17	2.08	7.32	2.80	4.13	177

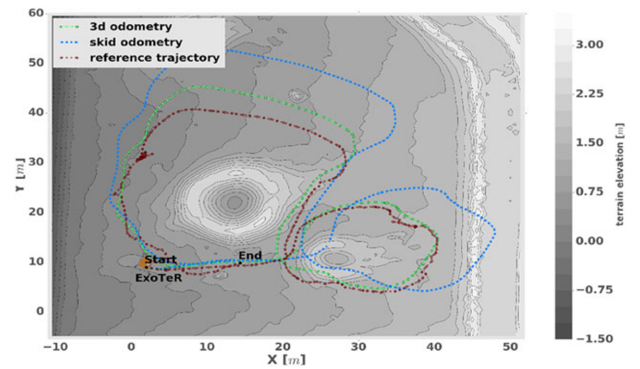


FIGURE 23 Decos experiment: resulting trajectories of the 3D odometry and skid odometry model with ExoTeR on the Decos terrain

ing the ground depression next to it. Similarly to the test in the Mars-like terrain, the stand-alone dead reckoning uses the 3D odometry model and compares it with a classical skid odometry implementation. Figure 23 shows the trajectory for the dead reckoning for both odometry models. Dead reckoning accumulates errors unbounded but due to the shape of double circle trajectory the maximum error does not appear at the end of the drive. Metric information is given in Table 4. Skid odometry is a very common odometry used in SLAM. The calculation averages left and right wheel velocities and estimates a delta pose using the attitude information from the AHRS. This simplification entails inaccuracies in the pose and motivates the effort to fully model the rover chassis and develop a 3D odometry model. The metrics show that a complete model of the chassis reduces the percentage error per distance traveled by half during the Decos test.

4.3.2 | GP odometry error

The GP model predicts the odometry error for the Decos terrain test using the same model as in the previous test at the planetary laboratory. The *true* odometry error is computed by comparing the delta poses from the 3D odometry model and the ground truth data acquired by the GPS. During the experiment, ExoTeR is commanded at a nominal velocity of 10 cm/s and drives most of the time on even terrain and conducive conditions. There are no significant sandy areas, dunes, or steep inclinations along the trajectory. This characteristic makes a relatively constant and small odometry error during the traverse. It is worthwhile to notice that the maximum odometry error measured along the drive is 2.5 cm/s, which corresponds to 25% of the commanded velocity. This is a notable improvement in traction with respect to the planetary robotics laboratory on which the slippage reaches almost 100%. This consequently has two impacts. First, the odometry error induced by the 3D odometry model is lower in the Decos terrain than in the Mars-like terrain. The metrics are shown in Table 4. Second, the odometry error predicted by the GP model is less notable in Decos (less error in absolute numbers) than in the Mars-like tests.

4.3.3 | Adaptive SLAM

Similar to the previous tests, the WO error dictates the adaptivity of the SLAM. The delta poses from the odometry module are preintegrated between image frames and used to track visual features. Equation (31) adapts the VO frequency and Equation (32) the criterion to select a new keyframe. The maximum allowed odometry error is also selected on a slippage threshold. The threshold is a percentage of the nominal rover velocity which in this case is 10 cm/s. The same four thresholds are selected to compare the adaptivity of the approach. Figure 24 depicts the localization and mapping trajectory for those different adaptive SLAM schemes. The first trajectory shows the result for a fixed SLAM without adaptivity. The rest of the trajectories show the SLAM solution for 10%, 25%, and 100% slippage thresholds with $\overline{\gamma}$ equal to 0.01, 0.025, and 0.10 m/s, respectively. Table 5 shows the metrics for the different SLAM solutions. The results denote that the number of processed image frames and keyframes can be reduced notably without significantly affecting the accuracy of the solution. In comparison to previous experiments at the planetary laboratory, there is not a significant change in graph sparsity along the drive. The reason is a more constant WO error and therefore less adaptation. In the case that the rover encounters a constant error, less adaptivity is required and the distribution of keyframes is equally maintained. Note that the adaptivity is based on how the traction performance is estimated by the GP odometry error model, and the soil properties are constant in the Decos area.

The loop closing module is not able to close the two loops performed by the rover along the trajectory at Decos. Neither the classical SLAM nor the adaptive SLAM is able to detect a revisited place. It is due to strong perceptual aliasing of the run. This is because the stereo camera sensor is tilted 30° downwards, and only features on the ground are perceived. This is a good setup for dead reckoning using visual odometry but makes difficult to close the loop using DBoW.

Adaptiveness also influences the quality of the map as shown in the previous experiments. The dense map reconstruction works with the same principle as explained before. In this case, an open loop experiment denotes higher map errors at the end of the traversal that at the beginning. Figure 25 shows a top down view of the different maps for the different adaptive SLAM schemes. The first image shows the color map, which overlaps the digital model acquired from the drone as a ground truth map. The rest of the visuals show the resulting maps colored with error information with respect to the digital model. The error is more pronounced at the left part of the bigger circle, which is the end of the trajectory.

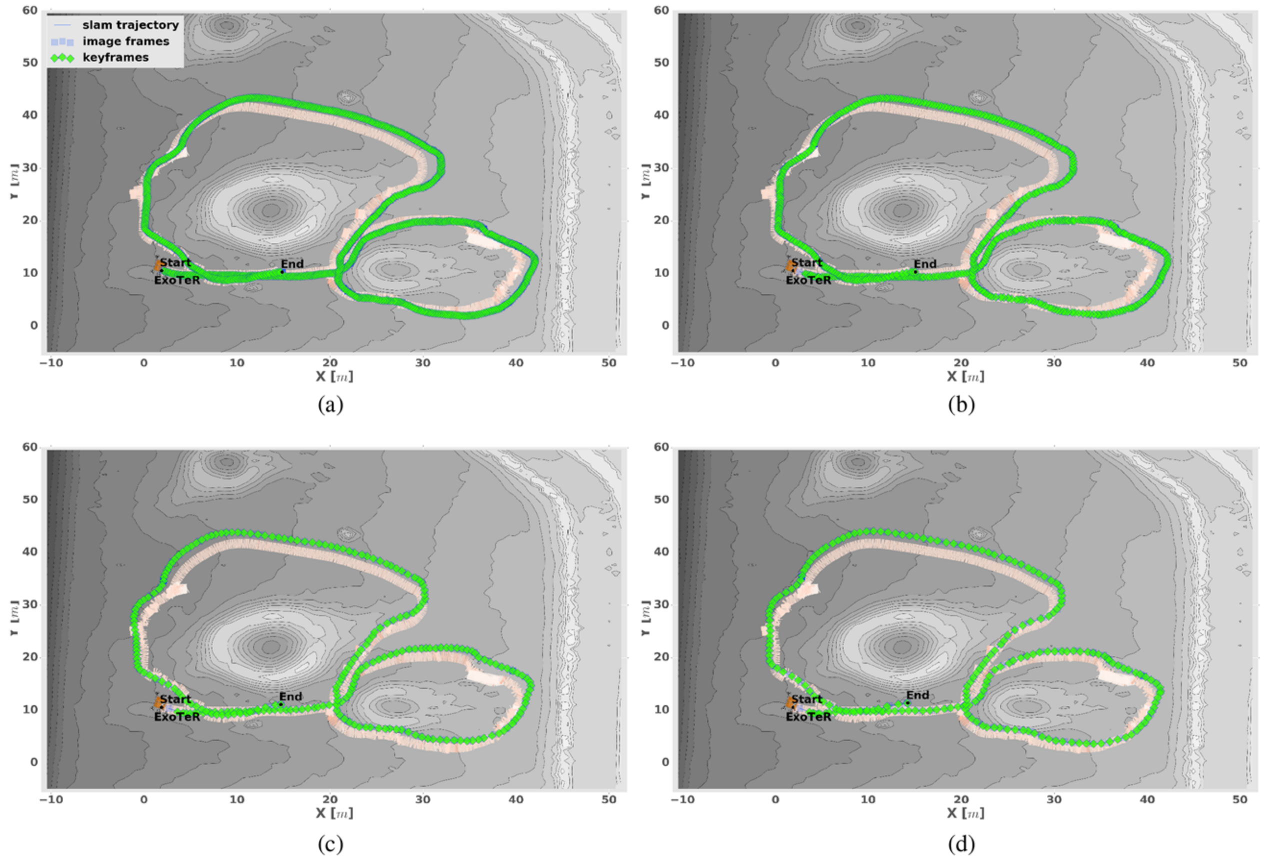


FIGURE 24 Adaptive localization and mapping for different odometry error thresholds for the data in the Decos terrain: (a) SLAM without adaptivity at $\underline{\tau} = \overline{\tau} = 0.4$ s and $\underline{\rho} = \overline{\rho} = 0.75$, (b) adaptive SLAM 10% [$\overline{\gamma} = 0.010$ ms^{-1}], (c) adaptive SLAM 25% [$\overline{\gamma} = 0.025$ ms^{-1}], and (d) adaptive SLAM 100% [$\overline{\gamma} = 0.10$ ms^{-1}]

TABLE 5 ExoTeR's pose results for the different SLAM schemes at Decos terrain

Adaptiveness	Number of frames	Number of keyframes	RMSE (m)	Max E. (m)	Final E. (m)	Max E. (%)	Distance (m)	Number of loops
Without adaptivity	6,133	797	1.06	5.41	0.48	3.05	177	0
10% threshold	2,982	530	1.07	5.56	0.53	3.14	177	0
25% threshold	1,352	243	1.29	5.61	0.67	3.16	177	0
100% threshold	1,224	215	1.63	6.30	1.19	3.56	177	0

5 | INFLUENCE ON A PLANETARY MISSION

This section analyzes the impact of including the adaptive SLAM into the navigation system of a potential planetary rover. The argumentation connects with the motivation given in Section 1. The purpose is to elaborate an analysis on how the adaptive SLAM could influence a real mission scenario.

Figure 26a shows the influence of adaptive SLAM in the number of image frames processed by the visual tracking module. Figure 26b shows the increase in the percentage error. Both figures show a different perspective on the effect of adaptiveness in the previous experiments. For instance, with a 25% threshold the adaptive SLAM triggers 53% (as average of the three experiments) of the maximum number of image frames, resulting in 47% fewer images processed by the visual-tracking module. It is worthwhile to notice that the number of image frames reduces with a small penalty in the pose error due to the selection of highly informative keyframes (see Figure 26b). A small number of image frames is always required, independently of the adaptiveness, to track a minimum number of features. This is the reason why at the maximum (100%) threshold, adaptive SLAM still uses 19.4% of the image frames. This is the minimum number of image frames to keep SLAM functional.

The dynamics of the *navigation system* used in Figure 1b is explained also here to evaluate the impact of the adaptive SLAM. For this purpose, the GNC of the ExoMars rover is used as reference. Figure 27 depicts the system in two parts. A first part in which the rover acquires

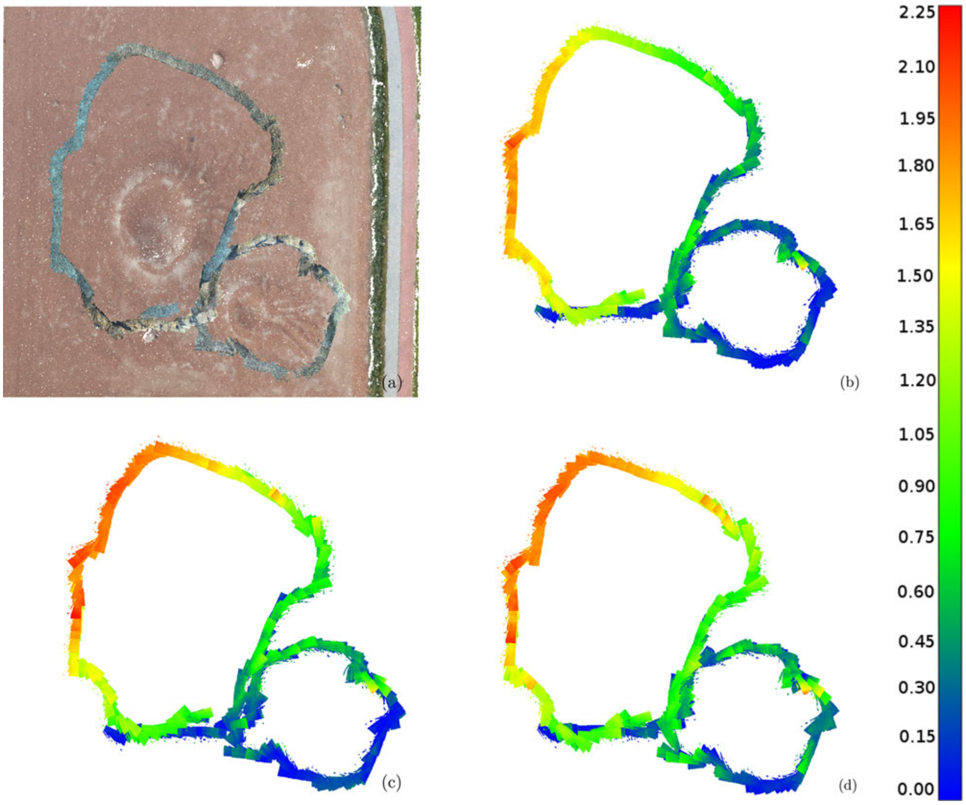


FIGURE 25 Adaptive map quality for different odometry error thresholds in the Decos terrain: (a) colored reference dense map reconstruction, (b) error map with respect to the reference map for 10% threshold in adaptiveness, (c) error map for 25% threshold in adaptiveness, and (d) error map for 100% threshold in adaptiveness. The colored error legend in meters is depicted at the right side

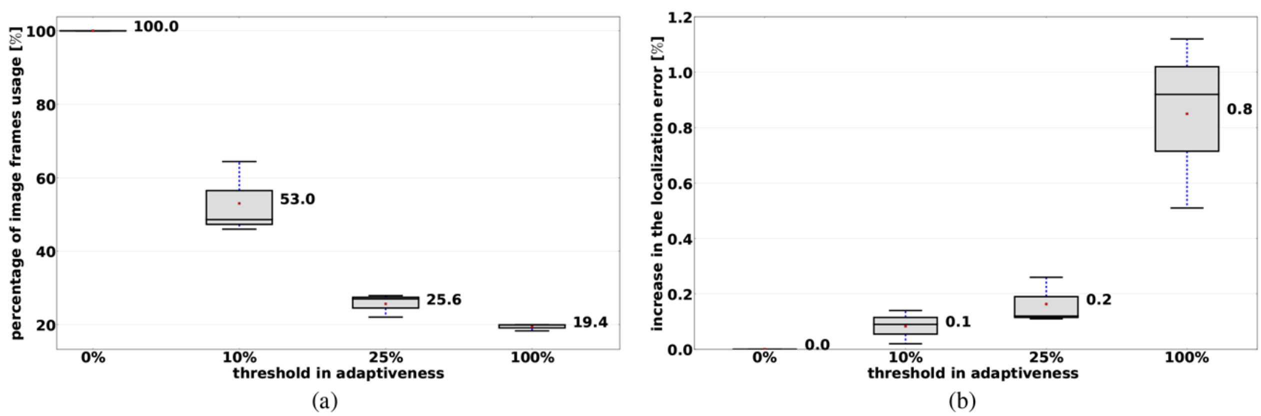


FIGURE 26 Adaptive localization and mapping: (a) influence of the threshold in adaptiveness to the number of computed image frames p_{us} and (b) penalty of the threshold in adaptiveness to the percentage error per distance traveled

images from the navigation cameras, computes a dense map of the surroundings, and calculates the free obstacle path. Consequently, the path is given to the second part to follow the desired trajectory and compute the localization. This is repeated every 2 m until the final target is reached. Such target is usually selected from aerial images. The localization and locomotion part acquires the images from the localization cameras, computes the image features, extracts the descriptors, and tracks correspondences with respect to the previous pair of images. The rover locomotion begins and executes such a cycle approximately every 33 cm. With these dynamics, the ExoMars rover stops six times every 2 m of traversed distance. The idea is to embed the adaptive SLAM into the *navigation system* and evaluate the impact on the distance traveled.

The average distance per sol for past, current, and future Mars rovers is introduced in Figure 2b. ExoMars rover has a maximum requirement of 50 m/sol during Phase B2. The navigation system requires 25 navigation image pairs every sol to compute a DEM every 2 m of traversal. The computational time is t_{nav} = 1.7 min, which

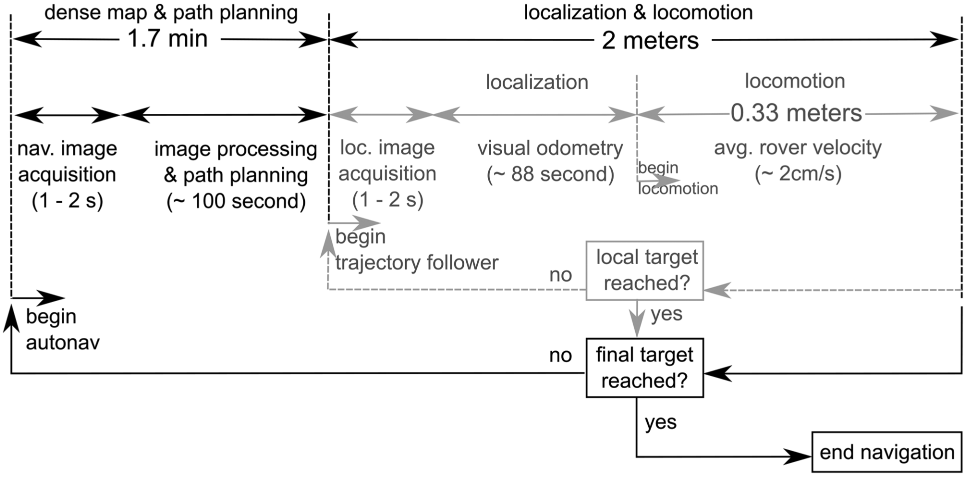
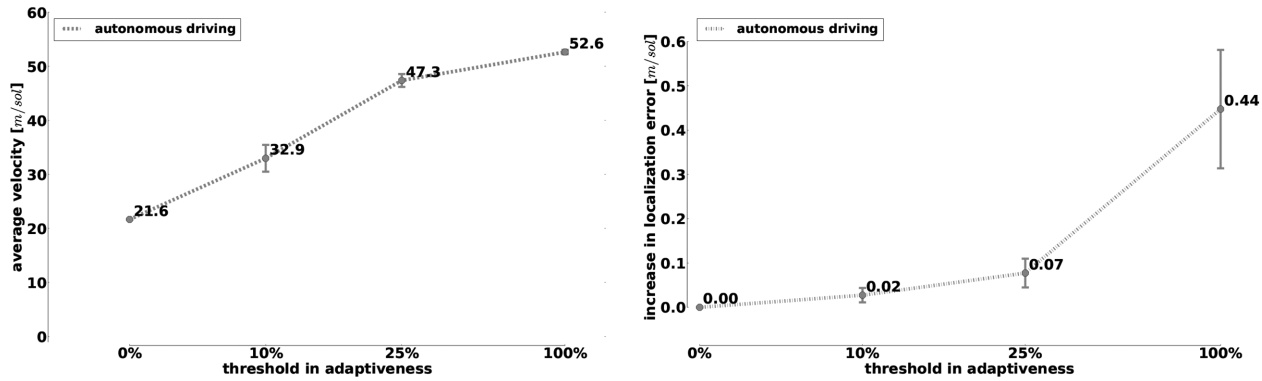


FIGURE 27 The GNC cycle for the *autonomous driving* of the ExoMars rover



(a) (b)

FIGURE 28 Adaptive SLAM analysis on a planetary rover mission scenario: (a) the rover's velocity per different thresholds in adaptiveness and (b) the increase in error per different thresholds in adaptiveness

involves a total of 42.5 min every sol to compute the map and perform the path planning on a LEON 2 processor. Similarly, the computational time to process a stereo pair of localization images for visual odometry is $t_{vo} = 1.5$ min. The following equation calculates the total time T required to drive a certain distance d as a function of the percentage of image frame usage p_{us} calculated from Figure 26a.

$$T = \left[\frac{d}{l_{loc}} t_{nav}\right] + \left[n_f \cdot p_{us} \cdot t_{vo}\right] + t_{loc} \tag{34}$$

where d is the distance to navigate (50 m), l_{loc} is the locomotion and localization part (2.0 m), n_f is the number of localization image frames in a distance d (151 image pairs for 50 m), t_{loc} is the time to traverse a distance d at the nominal rover velocity of 2.0 cm/s, and p_{us} changes according to the values depicted in Figure 26a. The result for maximum $p_{us} = 1.0$ (100%) is a total time T of 5.19 h to navigate 50 m. A minimum $p_{us} = 0.194$ (19.4%) gives a total time T of 2.13 h for the same distance. These values give a clear insight about the benefit of adaptive SLAM to the effective traversal.

An interesting comparison is to analyze the consequences of the adaptive SLAM system to the average rover velocity per sol. A Mars planetary rover has a nominal driving time of 2.25 h. This is used to define a *locomotion sol* in Figure 2b and to calculate the maximum time that a solar-powered rover can effectively drive on Mars, that is, around noontime. Figure 28a depicts the results of an average rover velocity per sol considering the *navigation system* from Figure 27 and using 2.25 h of driving time. The values are calculated using the total time T from Equation (34). As a result, the influence of the adaptive SLAM system in the traversed distance is significant. With the proposed *navigation system* and a classical SLAM, 0% slippage threshold, the rover would not be capable of traversing 50 m/sol fully autonomously. With the classical SLAM approach, the rover has to alternate *autonomous driving* with *direct driving* with the consequent intervention from the mission control on earth, making difficult to accomplish the task due to communication constraints. However, with adaptive SLAM the rover is able to minimize the computational time acquiring and processing localization images only when necessary. The cycle in the *navigation system* adapts to the localization demands, reducing unnecessary information, keeping the graph sparse, guaranteeing consistency, and minimizing the penalty in the pose error. For instance, with the 25% threshold in adaptiveness the rover achieves double the distance than with 0% threshold. This evaluation explains the benefit of adaptive SLAM and the importance of having adaptiveness onboard the rover.

Consequently, an increase in adaptiveness results in a bigger percentage error per distance traveled. Figure 28b shows the impact of each threshold to the final localization error per sol. The error bars associated with each value represent the 1σ standard deviation with the data from the experiments. The adaptive SLAM with 100% threshold has a 0.85% increase in the percentage error. This value translated to effective meters after traveling 52.6 m (see Figure 28a) results in an increment of 0.44 m in the final error. This means, that in case the final error after traversing 52.6 m is 1.0 m with a classical SLAM, the final error with the adaptive SLAM at 100% is 1.44 m. The deducted analysis is performed with the available data from the conducted ExoTeR experiments on earth, that is, terrain, gravity and illumination conditions. Nevertheless, it gives a valuable information on how it could result in a real mission scenario on Mars.

6 | CONCLUSION

The main contribution of this work is the adaptiveness in the localization and mapping for planetary rovers. Adaptiveness in SLAM is a higher level of abstraction with respect to conventional SLAM approaches. This is because optimization and machine-learning techniques bring scalability to the solution. We have shown an adaptive criterion to perform SLAM. The criterion is based on the rover interaction with the traversed terrain by means of the odometry error. We have shown that the interaction with the environment provides information which is useful to adapt the localization and mapping solution.

The primary objective of the WO is to capture the whole kinematics of the chassis by using a complete model. A skid odometry model might be attractive to compute due to the simplicity of the solution. However, our motivation for this research was to demonstrate that this simplicity has a penalty in accuracy. The use of transformation matrices is more convenient, practical, and generic than previous *ad hoc* contact-point approaches which does not easily adapt to wheeled robots (Schwendner, Joyeux, & Kirchner, 2013). The 3D odometry is generic and adapts to any open kinematic chain by means of transformation matrices. We show in previous research the generality of the approach (Hidalgo-Carrio et al., 2014). In addition, the set of contact points is combined in a least squares optimization to improve the estimate which minimizes an error. Despite nonsystematic errors, a 3D odometry model shows improvements in accuracy. The absolute improvement might depend on the rover trajectory and maneuvers, but it always benefits on uneven terrains. The 3D odometry model reduces by half the percentage error per distance traveled for trajectories with considerable turning maneuvers as shown in Table 4. The average improvement of the WO error among the presented experiments is 2.82% of the distance traveled.

A complete motion model also benefits the locomotion control of the rover. An accurate and complete robot control strategy at body level can only be achieved with a complete motion model. This is the case of our wheel-walking[#] research in Azkarate et al. (2015) and Wiese (2017). The control achieves constant body velocity while wheel walking. This maneuver would not be possible without a motion model by means of Jacobians. A complete motion model also has its pros and cons. Mobile robots with simple chassis or unexciting locomotion systems (e.g., skid robots) do not justify the development of a 3D odometry approach, especially when the robot only navigates on flat surfaces. Planetary rovers definitely justify the approach as shown in this work.

The modeling of nonsystematic odometry errors, affected by a poor traction performance, is a key issue in robotics. The reason is the direct impact on robot control, planning, and localization. Our research presents a novel methodology based on GPs. Instead of modeling the odometry of the rover, our approach model the error with respect to the estimation from a parametric model. This is very convenient for GPs because all the modeling is in the covariance function. The odometry error model does not require extra sensors beyond what is usually available in exploration systems (inertial sensors and joint encoders). Results have shown excellent predictions on representative terrains at Mars-like environments as shown in Figures 13 and 14. GPs are able to learn the hyperparameters of a kernel function to derive a nonparametric model of the odometry error. GP regression have an accuracy of 70% in the worst case scenario as shown in Test#1 on a Mars-like terrain. Even though the model cannot predict the absolute value in some extreme cases (i.e., 100% slippage), the model is able to detect errors in the odometry.

The technique to adapt the SLAM solution autonomously and on real-time is the main contribution of this paper. Wheel and visual odometry are supplementary techniques that accurately complement each other to estimate the rover pose. It has been demonstrated that under good traction conditions WO has the same accuracy as visual odometry (approximately 1–2% error). Both techniques increase error unbounded unless a loop closing strategy is applied. The ability to use the odometry error to encompass the classical visual SLAM approach has value in planetary rovers. Adaptive SLAM maintains accuracy and reduces nodes in the graph by more than 50% when using SLAM 25%. The solution exploits sparsity and scale the graph to the situations demands. Adaptive SLAM provides an efficient distribution of keyframes along the traversal. The methodology also adapts computational load to the current demands.

Future planetary rovers such as Mars Sample Return missions will start to use more and more autonomy onboard and SLAM will be a key aspect. The future of long-term SLAM will face robustness, resource awareness, driven performance, and high-level understanding. Our research focuses on a driven performance and resource awareness without affecting the robustness. The motivation of our approach is the balance between robustness and efficiency toward more capable navigation systems. It will contribute to faster navigation solutions as discussed in Section 5. We also believe that the work presented here provides insight in two directions. First, SLAM needs robotics as much as robotics needs SLAM. Robots enrich the SLAM solution by the information gained from their interaction with the ground. Second, the future

[#] ExoTeR wheel-walking video online, https://youtu.be/qkOKzFq1SpY of SLAM requires machine learning. Learning techniques exploit the information sensed by the rover. Online learning and scalability are essential for future long-term SLAM.

ACKNOWLEDGMENTS

This work presented here is part of the European Space Agency (ESA) Networking Partnering Initiative (NPI), grant no. 4000103229/11/NL/PA and the Entern project funded by the Space Agency of the German Aerospace Center with federal funds of the Federal Ministry of Economics and Technology (BMWi) in accordance with the parliamentary resolution of the German Parliament, grant numbers 50RA1406 and 50RA1407. The authors would like to thank all members of the Planetary Robotics Laboratory at ESA, for their help during the experiments. Special thanks to Gianfranco Visentin and Martin Azkarate for providing all the necessary support.

ORCID

Javier Hidalgo-Carrió http://orcid.org/0000-0001-6709-9285

REFERENCES

Alexander, J., & Maddocks, J. (1989). On the kinematics of wheeled mobile robots. *The International Journal of Robotics Research*, 8(5), 15–27.

Ali, K. S., Vanelli, C., Biesiadecki, J. J., Maimone, M. W., San Martin, A., & Alexander, J. W. (2005). Attitude and position estimation on the mars exploration rovers. *IEEE International Conference on Systems, Man and Cybernetics*, 1, 20–27.

Anastasios, M., & Roumeliotis, S. I. (2006). *A multi-state constraint Kalman filter for vision-aided inertial navigation* (Technical Report). Minneapolis, MN: Department of Computer Science and Engineering, University of Minnesota.

Azkarate, M., Zwick, M., Hidalgo-Carrio, J., Nelen, R., Wiese, T., Poulakis, P., … Visentin, G. (2015). First experimental investigations on wheel-walking for improving triple-bogie rover locomotion performances. Paper presented at Advanced Space Technologies for Robotics and Automation, (ASTRA), Noordwijk, The Netherlands.

Biesiadecki, J. J., Leger, P. C., & Maimone, M. W. (2007). Tradeoffs between directed and autonomous driving on the Mars exploration rovers. *The International Journal of Robotics Research*, 26(1), 91–104.

Borenstein, J., & Feng, L. (1996). Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics*, 12, 869–880.

Campion, G., Bastin, G., & D'Andrea-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics*, 12, 47–62.

Carlone, L., Kira, Z., Beall, C., Indelman, V., & Dellaert, F. (2014). Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, People's Republic of China (pp. 4290–4297). Piscataway, NJ: IEEE.

Civera, J., Davison, A., & Montiel, J. (2008). Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5), 932–945.

Clemente, L., Davison, A., Reid, I. Neira, J. L., Tardós, J. D. (2007). Mapping large loops with a single hand-held camera. Paper presented at Robotics: Science and Systems, Atlanta, GA.

Crassidis, J. L., & Markley, F. L. (2003). Unscented filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 26(4), 536–542.

Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6), 647–665.

Cunningham, C., Masahiro, O., Nesnas, I., Yen, J., & Whittaker, W. L. (2017). Locally-adaptive slip prediction for planetary rovers using Gaussian processes. In *2017 International Conference on Robotics and Automation (ICRA)*, Singapore (pp. 5487–5494). Piscataway, NJ: IEEE.

Dong-Si, T.-C., & Mourikis, A. I. (2011). Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In *2011 IEEE International Conference on Robotics and Automation*, Shanghai, People's Republic of China (pp. 5655-5662). Piscataway, NJ: IEEE.

Engel, J., Stuckler, J., & Cremers, D. (2015). Large-scale direct SLAM with stereo cameras. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany (pp. 1935–1942). Piscataway, NJ: IEEE.

Exploration, I. S., & Group, C. (2011). *The global exploration roadmap* (Technical Report). International Space Exploration Coordination Group.

Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2015). IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. Paper presented at Robotics: Science and Systems XI, Rome, Italy.

Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2017). On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1), 1–21.

Gajjar, B., & Johnson, R. (2002). Kinematic modeling of terrain adapting wheeled mobile robot for Mars exploration. In *Proceedings of the Third International Workshop on Robot Motion and Control*, Bukowy Dworek, Poland (pp. 291–296). Piscataway, NJ: IEEE.

Galvez-Lopez, D., & Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5), 1188–1197.

Geiger, A., Roser, M., & Urtasun, R. (2010). Efficient large-scale stereo matching. In *Proceedings of the 10th Asian Conference on Computer Vision (ACCV)*, Queenstown, New Zealand (Vol. 1, pp. 25–38). Berlin, Germany: Springer.

Grisetti, G., Kummerle, R., Stachniss, C., & Burgard, W. (2010). A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4), 31–43.

Helmick, D. M., Cheng, Y., Clouse, D. S., & Matthies, L. H. (2004). Path following using visual odometry for a Mars rover in high-slip environments. In *IEEE Aerospace Conference Proceedings*, Big Sky, MT (Vol. 2, pp. 772–789). Piscataway, NJ: IEEE.

Hesch, J. A., Kottas, D. G., Bowman, S. L., & Roumeliotis, S. I. (2014). Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1), 158–176.

Hidalgo-Carrió, J., & Cordes, F. (2012). Kinematics modeling of a hybrid wheeled-leg planetary rover. Paper presented at International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS-2012), Turin, Italy.

Hidalgo-Carrió, J., Poulakis, P., Köhler, J., Del-Cerro, J., & Barrientos, A. (2012). Improving planetary rover attitude estimation via MEMS sensor characterization. *Sensors*, 12(2), 2219–2235.

Hidalgo-Carrio, J., Babu, A., & Kirchner, F. (2014). Static forces weighted Jacobian motion models for improved odometry. In *2014 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, Chicago, IL (pp. 169–175). Piscataway, NJ: IEEE.

Hidalgo-Carrio, J., Hennes, D., Schwendner, J., & Kirchner, F. (2017). Gaussian process estimation of odometry errors for localization and mapping. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore (pp. 5696–5701). Piscataway, NJ: IEEE.

Iagnemma, K., & Dubowsky, S. (2000). Vehicle wheel-ground contact angle estimation: With application to mobile robot traction control. In *International Symposium on Advances in Robot Kinematics* (pp. 137–146). Dordrecht, The Netherlands: Springer.

Ila, V., Porta, J., & Andrade-Cetto, J. (2010). Information-based compact pose SLAM. *IEEE Transactions on Robotics*, 26(1), 78–93.

Joyeux, S., Schwendner, J., Kirchner, F., Babu, A., Grimminger, F., Machowinski, J., … Gaudig, C. (2011). Intelligent mobility - autonomous outdoor robotics at the DFKI. *KI - Künstliche Intelligenz [German Journal on Artificial Intelligence]*, 25(2), 133–139..

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2011). iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2), 216–235.

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *2007 Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan. (pp. 1–10). Piscataway, NJ: IEEE, 1–10.

Ko, J., & Fox, D. (2008). GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France (pp. 3471–3476). Piscataway, NJ: IEEE.

Ko, J., Klein, D. J., Fox, D., & Haehnel, D. (2007). Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy (pp. 742–747). Piscataway, NJ: IEEE.

Konolige, K., & Agrawal, M. (2008). FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5), 1066–1077.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). G^2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China (pp. 3607–3613). Piscataway, NJ: IEEE.

Lamon, P., & Siegwart, R. (2007). 3D position tracking in challenging terrain. *The International Journal of Robotics Research*, 26(2), 167–186.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visualinertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3), 314–334.

Maimone, M. (2014). C++ on Mars: Incorporating C++ into Mars rover flight software. Presented at *The C++ Conference*, Seattle, WA.

Maimone, M., Cheng, Y., & Matthies, L. (2007a). Two years of visual odometry on the Mars exploration rovers. *Journal of Field Robotics*, 24(3), 169–186.

Maimone, M. W., Leger, P. C., & Biesiadecki, J. J. (2007b). Overview of the Mars exploration rovers' autonomous mobility and vision capabilities. In *IEEE International Conference on Robotics and Automation, Space Robotics Workshop* (pp. 1–8). Piscataway, NJ: IEEE Press.

Montemerlo, M., Thrun, S., Roller, D., & Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *IJCAI International Joint Conference on Artificial Intelligence*, Acapulco, Mexico (pp. 1151–1156). San Francisco, CA: Morgan Kaufmann.

Mourikis, A. I., & Roumeliotis, S. I. (2008). A dual-layer estimator architecture for long-term localization. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK (1–8). Piscataway, NJ: IEEE.

Muir, P. F., & Neuman, C. P. (1986). *Kinematic modeling of wheeled mobile robots* (Technical Report). Pittsburgh, PA: Carnegie Mellon University.

Mukadam, M., Yan, X., & Boots, B. (2016). Gaussian process motion planning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden (pp. 9–15). Piscataway, NJ: IEEE.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5), 1147–1163.

Ojeda, L., & Borenstein, J. (2002). FLEXnav: Fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain. In *IEEE International Conference on Robotics and Automation*, Washington, DC (pp. 1–6). Piscataway, NJ: IEEE.

Ojeda, L., Cruz, D., Reina, G., Borenstein, J., & Member, S. (2006). Current-based slippage detection and odometry correction for mobile robots and planetary rovers. *IEEE Transactions on Robotics*, 22(2), 366–378.

Pirker, K., Ruther, M., & Bischof, H. (2011). CD SLAM—continuous localization and mapping in a dynamic world. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA (pp. 3990–3997). Piscataway, NJ: IEEE.

Poulakis, P., Vago, J. L., Loizeau, D., Vicente-Arevalo, C., Hutton, A., McCoubrey, R., … Ravera, F. (2015). Overview and development status of the Exomars rover mobility subsystem. Paper presented at 13th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2015), Noordwjk, The Netherlands.

Powell, M. W., Crockett, T., Fox, J. M., Joswig, J. C., Norris, J. S., & Rabe, K. J. (2006). Targeting and localization for Mars Rover operations. In *IEEE International Conference on Information Reuse and Integration*, Waikoloa Village, HI (pp. 23–27). Piscataway, NJ: IEEE.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Rehder, J., Gupta, K., Nuske, S., & Singh, S. (2012). Global pose estimation with limited GPS and long range visual odometry. *2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN (pp. 627–633). Piscataway, NJ: IEEE.

Roehr, T. M., Cordes, F., & Kirchner, F. (2014). Reconfigurable integrated multirobot exploration system (RIMRES), heterogeneous modular reconfigurable robots for space exploration. *Journal of Field Robotics*, 31(1), 3–34.

Rogers-Marcovitz, F., George, M., Seegmiller, N., & Kelly, A. (2012). Aiding off-road inertial navigation with high performance models of wheel slip. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal (pp. 215–222). Piscataway, NJ: IEEE.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision* (pp. 2564–2571). Piscataway, NJ: IEEE.

Schwendner, J., Joyeux, S., & Kirchner, F. (2013). Using embodied data for localization and mapping. *Journal of Field Robotics*, 31(2), 263–295.

Tarokh, M., & Mcdermott, G. J. (2005). Kinematics modeling and analyses of articulated rovers. *IEEE Transactions on Robotics*, 21(4), 539–553.

Vidal-Calleja, T., Davison, A., Andrade-Cetto, J., & Murray, D. (2006). Active control for single camera SLAM. In *Proceedings of 2006 IEEE International Conference on Robotics and Automation (ICRA 2006)*, Orlando, FL (pp. 1930–1936). Piscataway, NJ: IEEE.

Wang, J., & Englot, B. (2016). Fast, accurate Gaussian process occupancy maps via test-data octrees and nested Bayesian fusion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden (pp. 1003–1010). Piscataway, NJ: IEEE.

Wiese, T. (2017). *3D kinematic modeling and evaluation of rough-terrain locomotion modes for an ExoMars-like mobility subsystem* (Ph.D. thesis). University of Munich, Munich, Germany.

Williams, B., Klein, G., & Reid, I. (2007). Real-time SLAM relocalisation. In *2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil (pp. 1–8). Piscataway, NJ: IEEE.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Hidalgo-Carrió J, Poulakis P, Kirchner F. Adaptive localization and mapping with application to planetary rovers. *J Field Robotics*. 2018;35:961–987. https://doi.org/10.1002/rob.21790

VIDEO ATTACHMENT

The multimedia file is a video of ExoTeR performing *Test#1* at the Planetary Robotics Laboratory.