Adaptive Localization and Mapping for Planetary Rovers

von Javier Hidalgo Carrió

Dissertation

zur Erlangung des Grades eines Doktors der Ingenieurwissenschaften
- Dr.-Ing. -

Vorgelegt im Fachbereich 3 (Mathematik & Informatik)
der Universität Bremen
im Juni 2018

Datum des Promotionskolloquiums: 22.06.2018

Gutachter:

Prof. Dr. Frank Kirchner
Prof. Dr. Udo Frese

Abstract

Future rovers will be equipped with substantial onboard autonomy as space agencies and industry proceed with missions studies and technology development in preparation for the next planetary exploration missions. Simultaneous Localization and Mapping (SLAM) is a fundamental part of autonomous capabilities and has close connections to robot perception, planning and control. SLAM positively affects rover operations and mission success. The SLAM community has made great progress in the last decade by enabling real world solutions in terrestrial applications and is nowadays addressing important challenges in robust performance, scalability, high-level understanding, resources awareness and domain adaptation. In this thesis, an adaptive SLAM system is proposed in order to improve rover navigation performance and demand.

This research presents a novel localization and mapping solution following a bottom-up approach. It starts with an Attitude and Heading Reference System (AHRS), continues with a 3D odometry dead reckoning solution and builds up to a full graph optimization scheme which uses visual odometry and takes into account rover traction performance, bringing scalability to modern SLAM solutions.

A design procedure is presented in order to incorporate inertial sensors into the AHRS. The procedure follows three steps: error characterization, model derivation and filter design. A complete kinematics model of the rover locomotion subsystem is developed in order to improve the wheel odometry solution. Consequently, the parametric model predicts delta poses by solving a system of equations with weighed least squares. In addition, an odometry error model is learned using Gaussian processes (GPs) in order to predict non-systematic errors induced by poor traction of the rover with the terrain. The odometry error model complements the parametric solution by adding an estimation of the error. The gained information serves to adapt the localization and mapping solution to the current navigation demands (domain adaptation). The adaptivity strategy is designed to adjust the visual odometry computational load (active perception) and to influence the optimization back-end by including highly informative keyframes in the graph (adaptive information gain). Following this strategy, the solution is adapted to the navigation demands, providing an adaptive SLAM system driven by the navigation performance and conditions of the interaction with the terrain. The proposed methodology is experimentally verified on a representative planetary rover under realistic field test scenarios. This thesis introduces a modern SLAM system which adapts the estimated pose and map to the predicted error. The system maintains accuracy with fewer nodes, taking the best of both wheel and visual methods in a consistent graph-based smoothing approach.

Acknowledgments

I would like to begin by thanking my supervisor Frank Kirchner for the support and guidance during these years and for giving me the amazing opportunity to be part of one of the greatest robotics institutes. Thanks for being always available to discuss and see my latest results. There is no doubt that I am a better researcher today thanks to Frank. I would also like to express my gratitude to Gianfranco Visentin and Pantelis Poulakis for offering me this great adventure in robotics. They introduced me to the world of space robotics and brought out the best in me at work, and in my personal life.

This thesis is the result of an amazing journey at the Robotics Innovation Center (RIC) of the German Research Center for Artificial Intelligence (DFKI) and at the European Space Research and Technology Center (ESTEC) of the European Space Agency (ESA). I would like to thank all of my colleagues at DFKI for their dedication to make this research possible. Thanks to Sylvain Joyeux for the technical guidance during my early time at DFKI, and for his introduction to the world of robotics software. Thanks to Jakob Schwendner, Alex Duda and Sascha Arnold for the inspirational discussions in SLAM, this thesis would not be the same without them. Thanks to Ajish Babu for the guidance in the mathematics of Jacobians and optimization. Thanks to Daniel Hennes for his knowledge and comments about machine learning, and to Thomas Röhr and Daniel Kühn for their constructive comments and suggestions. I would like to thank Benjamin Freeman for his help with the English grammar. The support from all colleagues at RIC has made this thesis possible.

Thanks to all the guys at ESA, especially Martin Azkarate, who supported me with all the necessary equipment to conduct the experiments. Martin literally gave me a place in his office and in the lab. Thanks to Vassilios Tsounis, Martin Zwick and Robin Nele for growing ExoTeR in to the young, strong and brave rover that it currently is. I will always have good memories of our days and nights at the planetary robotics lab and field testing. My most sincere thanks to them: "Orbit high and rove on!"

My PhD life would have been more productive, but less enjoyable, without my friends at Bremen. Thanks for all the good times, words of kindness and patience. Thanks for giving me all the moral support to make this journey real. "Never so close" to turn the impossible into possible. I am a better person today thanks to them. My warm thanks!

I would like to especially thank my family. To my parents and sister, for their unconditional support and love. Thank you for being there every time I came back from the lab, understanding that distance is not a barrier and feeling that they are always there for me.

Contents

Chapter 1

Introduction

This chapter motivates the thesis and highlights the impact of a localization and mapping system on mission success and operations. It addresses the current problem on the design of sensor fusion and debates the need for more onboard autonomy for planetary rovers. The structure of this document is presented at the end of this chapter.

1.1 Background

Robots have their origin in the ancient world but it is not until the industrial revolution and with the use of electricity when robots began to make an impact on our lives, Siciliano and Khatib (2016). Robotics started its modern development in car factories where the requirements for autonomy were limited to a few sets of commands. When robots were mainly preprogrammed or teleoperated the requirements for autonomy in general and autonomous navigation in particular were very limited. Robotics have finally emerged from the structured environments in factories to the outdoor domain setting higher requirements for autonomous navigation. Mobile robots are present in several field of application such as driverless cars, agriculture, underwater and planetary exploration. The concept of sending robots to explore beyond the realm of Earth has been present since the early 1960s, Gao (2016). The future of exploration is driven by robotics, and autonomous systems are a key element. This is demonstrated in recent mission studies like the MarsFast performed by ESA (2016). In addition, the level of autonomy is not fixed and is diverse from mission to mission. A robot might operate in different levels of autonomy according to mission requirements and communication constraints. The European Cooperation for Space Standardization (ECSS) defines four levels of autonomy in the Requirements & Standards Division (2008). The levels go from E1: real-time ground control or teleoperation; to E4: goal-oriented mission operation or autonomous operation. Beside the ECSS Standard, further information about autonomy and its definition for planetary rovers can be also found in Bajracharya et al. (2008), Schwendner and Joyeux (2010) and Gao (2016). The more mankind aims to explore the universe, more level of autonomy is required due to communication and operations constraints.

The engineering process of a planetary rover is a concurrent design which involves several iterations and phases presenting many challenges during the process. The rover system design is driven by mission requirements and environmental constraints that affect the final rover concept. The design parameters and constraints affect the final set of sensors in order to localize and perceive the surroundings. Different philosophies and approaches are usually taken based on a number of criteria such as redundancy, Technology Readiness Level (TRL), mass, simplicity, reliability and accuracy. During a concurrent design process, different systems contribute to the definition of a suitable rover design: locomotion, power, thermal, On Board Data Handling (OBDH) and many others. An engineering methodology and understanding of performance in detail improve and accelerate such process in the design. Mobile robots in the space domain are sometimes referred to as ground spacecrafts and their design appears as a balanced evaluation. However, there are some differences and peculiarities that make the rover design a unique case in relation to conventional spacecrafts. Among all the systems that constitute a planetary rover, three are critical during the design due to their influence on the mission success: these systems are power, locomotion and navigation. Power needs to be properly scaled since it is a crucial system affecting the performance of the rover as well as scientific instruments. The locomotion and the navigation systems are sometimes mentioned together as *mobility*, emphasizing their importance and interdependence. In particular, 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 capabilities are of exclusive competence of the *navigation system*. They heavily depend on the mission and their requirements affect the design. These requirements are of three types: operational, functional and resources. Operation, because the communication constraint in space imposes a certain level of autonomy. Functional requirements define the level of performance. Resources establish the sensor type, perception, computational power and software restrictions. Fig. 1.1 shows a rover GNC system diagram. The navigation front-end and back-end appears together with the onboard computation demands.

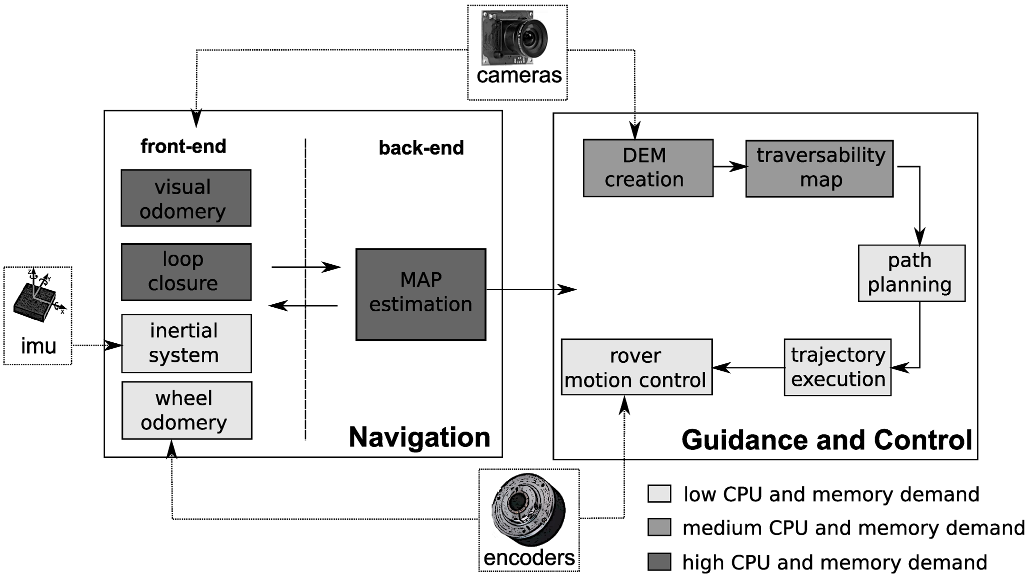


Figure 1.1: Schematic representation of a nominal GNC system for a planetary rover with the different components and the computational demands. The navigation part distinguishes the front-end to process sensory data and the back-end to perform smoothing.

The solution available in actual planetary rovers (i.e. flight missions) and terrestrial robotics (i.e. R&D activities) has a significant technological gap. The reason is due to the extreme conditions of the space environment. Qualified components are costly with a tedious manufacturing process. Space qualification affects robot sensing and computational power. One can say that the current technology available in space is a decade behind consumer products. Nevertheless, such a limitation should not prevent research in laboratory prototypes in order to reduce the technological gap. A planetary rover is a perfect technology-driven example of integrated exploration in extraterrestrial environments. Among all those tasks required for autonomous navigation, localization and mapping appears to be a fundamental element. Makarenko et al. (2002) describe the integrated exploration framework as a tightly coupled system, see Fig. 1.2. The overlap between tasks gives the system the capability to explore. The effectiveness of such exploration strategy depends on a proper balance among the motion control and planning, the localization and the mapping. The performance of the autonomous navigation highly depends on the robustness and reliability of the localization and mapping capabilities, which are responsible for closing the control loop while the rover drives along the planned path, further details are available at Kuroda et al. (2004); Helmick et al. (2004b); Baumgartner (2001); Baumgartner et al. (2004); Huntsberger et al. (2003); Bakambu et al. (2010).

Several different sensors are incorporated on a rover and their individual outputs are combined in a sensor fusion framework in order to estimate the pose with accuracy (i.e. position and attitude), as described in the work by Mitchell (2007); Liggins et al. (2009). The localization and mapping problem uses random variables and probabilistic inference due to the inherent nature of the problem. Sensory information is affected by noise, limiting the deterministic capabilities of the rover to know its localization and surroundings. This lack of precision is described as uncertainty in the measurement. Inaccuracy affects sensor readings, the rover state and the model of the environment which are modeled as random variables. The combination of different sources of information determines a better estimation of the quantities and is referred to as *sensor fusion*. Sensor fusion is the process of combining sensory data or higher levels of information in order to obtain a more meaningful estimation than when individual sensory data are used. The uncertainty and the sensor fusion form the foundations of probabilistic Simultaneous Localization and Mapping (SLAM) techniques. Therefore, SLAM appears as a central element to understand the autonomous navigation process. It provides the current pose of the rover (i.e. localization) and detail elements in the surroundings (i.e. environment). This information can be calculated by different levels of precision and complexity. The resulting quantities are then used in the planning and control part in order to interact with the environment. This is also called the *sensing-acting* loop in robotics agents. The localization and mapping system is typically divided into two parts, the *front-end* and the *back-end* (left side in Fig. 1.1). The former has direct access to the sensor information and produces higher level of sensory data (i.e. delta displacements, image features extraction, data association). In contrast, the back-end contains the collection of optimization techniques and filters (i.e. Kalman filters, Particle filters, nonlinear least squares optimizers) in order to estimate the Maximum a Posteriori (MAP).

Delta displacements relates the current pose of the rover to one or more of its previous poses (i.e. dead reckoning). Absolute position information on the other hand relates the rover position to a globally referenced map of the environment (i.e. global localization). While localization in robotics has some similarities between terrestrial and space applications, they also have some differences. Robotic applications in planetary exploration fail to benefit from GPS and accurate long-term localization systems are needed for the mission success. Due to inherent sensor errors and computation cost, sensor fusion techniques are used to estimate the resulting position and attitude more accurately than if each sensor was considered individually. In order to accomplish such a task, three kinds of sensors are nominally considered in a planetary rover, joint encoders and inertial sensors for proprioceptive information and cameras as exteroceptive sensors. This set of sensors are present in the majority of current rover designs, Borenstein and Feng (1996b); Hidalgo-Carrio et al. (2012b);

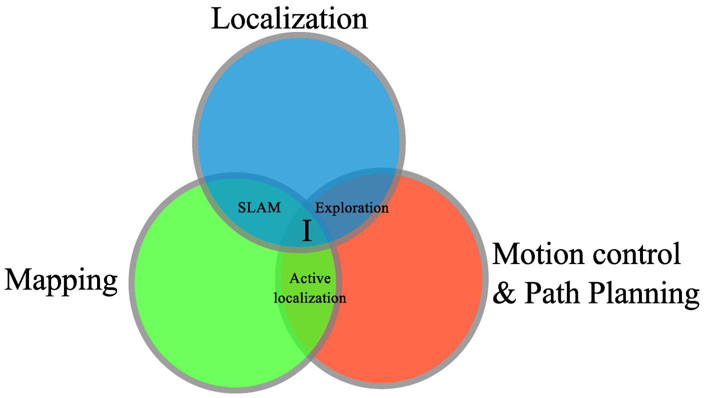


Figure 1.2: The exploration framework, the integrated exploration strategy (I) appears as a tight system coupled with localization, mapping and motion control & path planning. Figure adapted from Makarenko et al. (2002).

BillurBarshan and Durrant-Whyte (1995); NASA (2015, 2004). Additional types of sensors have also been considered for integration in mobile exploration systems such as sun-sensors which are cheap in mass and power requirements. A sun-sensor is not critical since the rover is equip with cameras and its inclusion in the design mainly depends on system redundancy and mission constraints. These sensors can provide information on the heading of the rover without the need to pan the main camera, Eisenman et al. (2002); Furgale et al. (2011); Trebi-ollennu et al. (2001). In addition, range-based exteroceptive sensors, such as LiDAR, can provide dense distance images of the environment with greater accuracy, higher distance and less computational cost compared to stereo processing, Poppinga et al. (2010). They consume more power and mass allocation than conventional stereo cameras and their development currently has a low maturity for space applications.

In summary, the needs of a localization system on board of a planetary rover have several similarities with the ones on terrestrial vehicles as well as conventional spacecrafts. However, they have significant differences in terms of operational constraints and dynamics. The search for accurate and efficient sensor fusion algorithms for robot and vehicle localization has been performed in previous research and takes part in many fields of engineering. It will be introduced in Section 1.2, how different localization and mapping strategies might have negative impacts on mission autonomy and operations. In particular, the process of efficient dead reckoning and adaptive data association depending on the rover navigation demands still remains an open question.

1.2 Motivation

The localization and mapping strategy heavily depends on the sensor technology and operational constraints. The estimation of orientation or attitude is one of the key estimates in robot navigation because it establishes the direction of motion. Accelerometers and gyroscopes are used in the Inertial Measurement Unit (IMU) to measure the relative change in orientation. The integration of angular velocities to produce angular position results in accumulation of error or drift. Accelerometers can correct such a drift in the pitch and roll angles when stationary with the estimation of the gravity vector. The heading or yaw is the most critical angle since it affects the direction of motion. The correction of heading by external means is an open research problem due to the lack of strong magnetic field in non-terrestrial applications, Volpe (1999) and Lambert (2011). Relative localization is the process of determining the pose with respect to some initial coordinate frame. It is calculated using wheel odometry, visual odometry and/or SLAM. Wheel odometry is the most natural technique to compute relative displacements based on counting of wheel and joint revolutions. Visual odometry estimates relative motion by extracting and tracking salient features on successive images. SLAM is the construction of a model of the environment and the pose estimation within it. Afterwards, absolute localization can be achieved by determining the rover pose in a global frame of reference. Panoramic images and digital elevation models are the most used techniques. Orbiter imagery matching uses high resolution images (e.g. HiRISE or LROC) to correct accumulated pose errors in a global frame, Gao (2016). However, such global localization approaches are manual with a strong dependency on human intervention from ground. The accuracy strongly depend on the details of the model, being extremely difficult to achieve GPS-like precision, Gao (2016).

Relative localization strategies, such as wheel odometry, are computationally inexpensive and effective on even terrain with good traction properties. The most complete and reliable localization scheme for planetary rovers is, up to now, on board in the Mars Science Laboratory (MSL), NASA (2015), and the Mars Exploration Rovers (MER), NASA (2004) and Maimone et al. (2007b). Surface Attitude Position and Pointing (SAPP) is the software component in charge of calculating and propagating rover attitude and position estimation for MER, using and combining different techniques and sensory information. SAPP is further explained in Ali et al. (2005) and carries out the propagation of the rover's pose depending on three commands. These commands are defined in the Attitude Acquisition Machine and triggered from the ground control station depending on rover operations. The localization system computes and propagates attitude using gyroscopes integration and additional support from the Sun's elevation information on the camera images. Also accelerometers data in static regimen is provided. The rover position is propagated using wheel odometry and no accelerometers are used in that step, Maimone et al. (2007b). Conventional wheel odometry remains good on simple terrain, flat and level ground. The rover Spirit only accumulated 3 % position error over 2 km of driving on level ground, Biesiadecki et al. (2007). Quantitative ahead images from previous sol (Martian day - 24.6 h) allow further offline terramechanics simulation from the ground, giving valuable aid in wheel odometry and the type of maneuvers to perform. This is referred to as *active localization* in Fig. 1.2. However, it is not possible on complex terrains or when ahead images are not available. The MER mission did not consider visual odometry from camera images in the nominal localization scheme by SAPP. The high slippage observed while driving on Mars surface forced engineers to include visual odometry in successive system updates. Maimone et al. (2007a) describe the use of visual odometry based on structure from motion algorithms using stereo image computation. Se et al. (2004) give also an analysis of applicability of visual odometry in planetary rovers. Visual features are deduced and tuned for corner detection. Stereo matching of the selected features and feature tracking, together with the successive motion estimation, are computed on board the rover. The incorporation of visual odometry into the localization addressed a significant advance. The application of visual odometry turned out to be essential on demanding terrains, typically on loose/mixed terrain and/or slopes of 10 % and higher, measuring slips as high as 125 % when it tried to drive up more than 25° slope, reported in Helmick et al. (2004b); Maimone et al. (2007a). Slippage detection is a complicated task, and 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, Maimone et al. (2007a). Consequently, it affects daily operations and degrades the mobility of the whole rover, Biesiadecki et al. (2007) and 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 difference in speed between using one or another localization solution clearly points to the need for improving and analyzing the design of future localization schemes. This fact has an important consequence on rover mo-

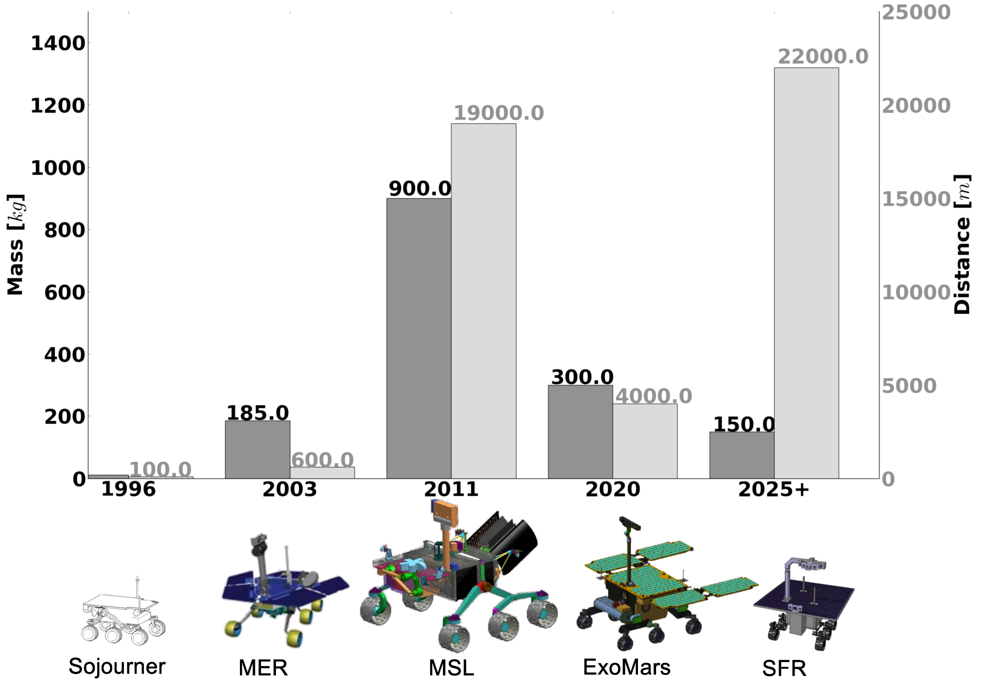


Figure 1.3: Comparison of mass (dark gray) and distance (light gray) to traverse by each rover for past, current and future Mars missions.

bility 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. For this reason the use of visual odometry and obstacle avoidance was limited considerably in Mars rovers. Different localization solutions were remotely switched from operations in order to adapt to the mission demands (task driven inference). Visual odometry was used when extremely necessary and *direct driving* on benign terrains. This fact enhances the necessity for future navigation systems to use perception techniques combined with complementary sensor information. Onboard solutions are required in order to reduce human intervention from ground, close the operational loop at the flight segment and minimize waiting times and communication windows during operation.

Different rover masses and traversal requirements of some current planetary rover missions are summarized in Fig. 1.3. Sample Fetching Rover (SFR) concepts are studied in the Robotics Exploration Preparation (MREP) program in Europe and NASA post 2020 missions as fast mobility rovers, reported in Exploration and Group (2011). MER covered $\sim$ 15 km in $\sim$ 7 years while next rover missions plan a traversal range of $\sim$ 20 km in $\sim$ 6 months. Traverse requirements will drastically drive the design of the platform and its mobility system. As mentioned previously, mobility is used as a term for the locomotion system combined with navigation in order to em-

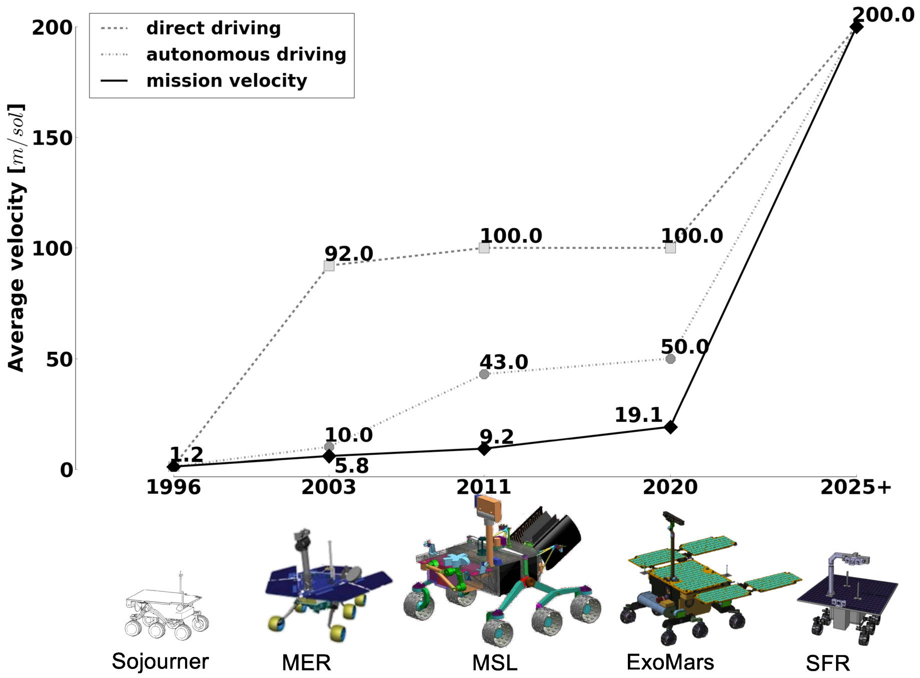


Figure 1.4: Rover velocities for past, current and future Mars missions.

phasize and recognize the interwoven nature of these two systems. For example, a more capable locomotion system decreases the need of obstacle avoidance, decreasing the complexity of the GNC system. Allouis et al. (2010) give a general system engineering overview of the problem. Increasing the size of the locomotion system is a suitable solution to improve mobility and fit the traversal requirements. Mission constraints, in terms of mass and volume, do not always allow a big locomotion chassis and actuators. A more capable locomotion system, such as that of the MSL with $50\,\mathrm{cm}$ wheel diameter, would allow offline fast path planning using the HiRISE imager, with approximately $30\,\mathrm{cm/pixel}$ ground resolution, reducing complexity in the global obstacle avoidance algorithm. Nevertheless, mission constraints always drive the rover design and an imperative improvement of the GNC system is required for future missions where a lighter rover takes part of a multi system mission scenario. While ExoMars rover mass envelope is about $300\,\mathrm{kg}$, Sample Fetching Rover (SFR) is less than $150\,\mathrm{kg}$, having a direct penalty in the size of the locomotion system and increasing the complexity of the GNC. Traverse requirements are much higher with reduced mission lifetime in order to fulfill the cache return operations for a Mars Sample Return (MSR).

Velocities for past, current and future Mars rover missions are depicted in Fig. 1.4. The velocity analysis defines a nominal Mars *locomotion sol* with a driving time of

2.25 h and a *science sol* as a pure scientific measurement activity day without rover driving operations. The *direct driving* line in Fig. 1.4 is 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). The *mission velocity* line is the average mission velocity taking into account *locomotion* and *science sols*. 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 drive longer distances in a shorter time. The different lines depicts how the rover speed is influenced by the surface navigation technology. This is because the navigation strategy is dictated by information available from the previous sol at the ground control station and based on surface and soil conditions of the environment. MER *direct driving* 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*, Biesiadecki et al. (2007). However, autonomous capabilities with visual odometry and obstacle avoidance on MER decreased the speed considerably, up to 6 m/h. MSL speed has been designed to reach up to 100 m/sol in *direct driving* and 43 m/sol for nominal speed when using autonomous capabilities. ExoMars requirements in Phase-B2 have a nominal speed of 50 m/sol in *autonomous driving*, Rusu (2014). 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, ESA (2016). MSL and ExoMars nominal average velocity are set with autonomous navigation capabilities, in contrast to previous missions in which *direct driving* was mainly utilized. SFR should take this improvement one step further.

Understanding the dynamics of the *navigation system* in a planetary rover is desirable in order to motivate this thesis. Fig. 1.5 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 in order to follow the desired trajectory and compute the localization. The localization & 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. Feature matching uses the information from the wheel odometry to search for potential correspondences. 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

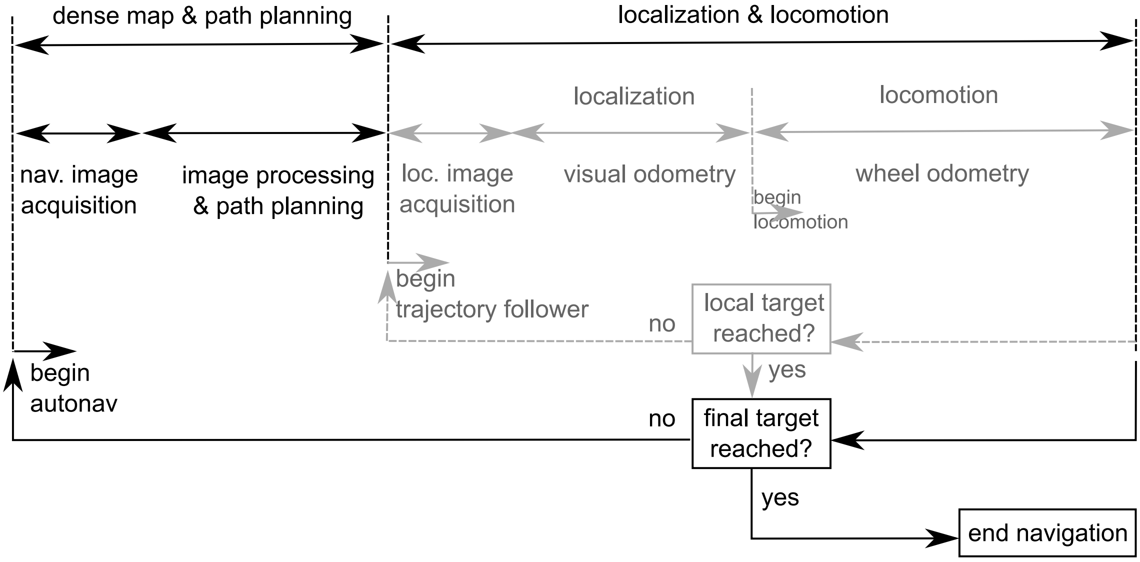


Figure 1.5: The *navigation system* for the *autonomous driving* of a planetary rover.

the final location target is reached.

While Mars undoubtedly is a primary target for exploration missions involving mobile robotic systems, the Moon has also regained attention from the scientific community, as well as several space agencies and private entities that currently pursue exploration missions, Astrobotic (2017). Multiple missions using either orbiters or impactors (SMART-1, SELENE, Chang'e-1, Chandrayaan-1, LCROSS, LRO) have verified the existence of volatiles like water in the lunar south polar region. This is interesting from a scientific point of view as well as for the application for in-situ resource utilization scenarios. ESA has planned a precursor mission to the lunar south pole, mainly to verify technologies in the area of soft precision landing, Homeister et al. (2010). Originally the mission had a rover during the first phases of its design, which was later removed due to mass constraints. Multiple experiments using the rover were already planned in this design phase, which required high localization accuracy of $0.1\,\%$ of the overall distance traveled. The study, called Next-LL, has a scaled down rover in the range of $10\,\mathrm{kg}$ class vehicle, Richter et al. (2011). This rover would only travel a distance of $100\,\mathrm{m}$ away from the lander. However, the rover would need to return to the lander autonomously and thus, autonomously drive the $100\,\mathrm{m}$ without operator intervention. The total distance was planned to be $10\,\mathrm{km}$. This distance requires a consistent localization and mapping component with appropriate accuracy. Recently, Chinese and the Indian Space agencies have plans for lunar rover missions and the Yutu rover from the Chinese National Space Administration (CNSA) is the first rover to operate on the Moon since the Lunokhod 2, Laxman (2012) and Chandrayaan (2017).

Future missions are likely to be a SFR concept 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. Past missions have demonstrated that the geometric and non-geometric hazards could stop the motion of a rover. Those potential hazards are difficult to detect remotely from Earth, calling for an on board solution Maimone et al. (2007b). Therefore, future rovers require data fusion schemes (e.g. SLAM) to deduce rover position and orientation in a prolonged, optimal and adaptive manner. SLAM builds a globally consistent map of the environment to locate the rover and store useful samples. It also provides a framework to study sensors, efficient adaptivity and interprets the environment as a whole, providing loop closures. Without SLAM the rover interprets the world as an *infinite corridor* making long-term autonomous systems inefficient and susceptible to failure. In this aspect, the future of SLAM will combine machine learning techniques with optimization approaches and hardware improvements (e.g. FPGAs) in order to enable more reliable and fast surface navigation.

Biesiadecki et al. (2007) introduce an overview study on how the impact of different navigation and localization modes affects rover autonomy and mission planning. Huntsberger et al. (2003) report upcoming developments at NASA/JPL for future missions with special emphasis on precision navigation over relatively long distances and improvements in mobility operations. Schenker (2006) describes the importance of surface mobility to science, addressing some key problems in advancing performance of future planetary rovers. The impact and influence of the localization and mapping system on the navigation and mission operations motivates this thesis. A motivation towards a better understanding of a dead reckoning process design and adaptiveness for modern SLAM systems in space.

1.3 Objective

The objective of this thesis is the design, development and evaluation of an adaptive SLAM driven by the navigation demands of a planetary rover. Such design, development and evaluation focuses in every level of the localization and mapping problem, from a simple but important dead reckoning to a complete optimization solution. The needs of SLAM on board a planetary rover has significant differences in terms of operational constraints and dynamics (see Section 1.2) with respect to existing solutions. Adaptiveness in SLAM could mitigate, or completely solve, such operational constraints. It is the thesis of this research to understand how adaptive SLAM could improve SLAM with considerations in future planetary rovers.

What is adaptive SLAM? Generally, adaptiveness is the ability of a system to adapt or change. In the context of localization and mapping, adaptiveness is the act of adjusting SLAM to the system demands. Domain adaptation is an important problem in modern SLAM systems and this thesis focuses on the important aspects concerning planetary rovers. More specifically in this thesis, adaptiveness or adaptivity focuses on two domains or parts. Part I is dead reckoning, in which the robot determines its pose by integrating delta displacements with respect to some initial coordinate frame as an infinite corridor. Part II is Simultaneous Localization and Mapping, the estimate of the pose and a model of the environment in which the robot performs exploration. The following defines the essential statements of this thesis.

Thesis statements for the adaptiveness in the design of a dead reckoning process:

- Attitude and Heading Reference System Design. The estimation of the rover's attitude is the first task to investigate in the robot localization problem. Heading defines the direction of motion and attitude determines an important role in tasks such as map reconstruction, path planning and pointing communication antennas. Tightly coupled SLAM systems are not practical SLAM solutions for planetary rovers due to limitations in computational resources and operation constraints, meaning the rover might operate in a simple dead reckoning mode for some periods. The characterization of inertial sensors is of notable interest for this aspect of the localization problem. This part establishes an adaptive design process using the Allan variance, Allan (1966), in order to incorporate any inertial sensor to the inertial navigation system. The analysis of this statement is in Chapter 3.
- Enhanced 3D odometry model. The inheritance from indoor robotics, which traditionally operates a robot in a structured or planar environment, has brought inefficient and simple techniques to the field of odometry in outdoor robotics. The definition of a contact point together with a contact point angle better defines the robot displacement. Therefore, the estimation of delta displacement using a complete motion model is a meaningful technique which produces a more accurate estimation. In addition, the adaptive selection of contact points to estimate the delta pose is a research question. This statement is investigated in Chapter 4.

Statements in the adaptive Simultaneous Localization and Mapping:

- Machine learning for SLAM. This objective is the identification of hazardous areas along the path by learning generalized kernels from previous rover experience. It comprises the development of a Gaussian Process in order to model the statistical error or uncertainty of wheel odometry as a nonlinear regression model. The result is a Gaussian Process over odometry errors to overcome the effect of poor traction on slippery terrains. The development of this statement comprises Chapter 5.

- Adaptive Graph Sparsity. As a consequence of identifying non-systematic errors in wheel odometry using a Gaussian Process model, the SLAM is influenced by the traction performance providing an adaptive solution until now neglected by the SLAM community. This adaptiveness in SLAM achieves more accurate results with less resources by means of an adaptive data association policy, such that the adaptiveness affects the visual odometry and the optimization backend. This statement is mainly evaluated in Chapter 6.

The design, development and evaluation of the above objectives comprise the core of this thesis. Those elements are designed, developed and evaluated independently in the document and combined in an adaptive localization and mapping solution.

1.4 Structure of this thesis

This document is structured in two main parts. Each part consists of two chapters and each chapter describes a consistent piece of work related to each of the thesis statements summarized in Section 1.3. The structure is graphically depicted in Fig. 1.6.

Part I describes the dead reckoning problem: Chapter 3 explains the design and evaluation of an attitude and heading reference system. Chapter 4 develops the enhanced 3D odometry model which is used to improve dead reckoning and serves as the motion model for prediction in the visual SLAM.

Part II conducts Simultaneous Localization and Mapping: Chapter 5 presents a machine learning technique using a Gaussian Process (GP) for odometry error regression. Adaptive SLAM is described in Chapter 6, a localization and mapping solution with active perception and adaptive graph sparsity based on the prediction performance of wheel odometry.

In addition, state-of-the-art in SLAM and a historical evolution of planetary rovers is described in Chapter 2. Finally, a discussion of the objectives of this thesis and future work are presented in Chapter 7.

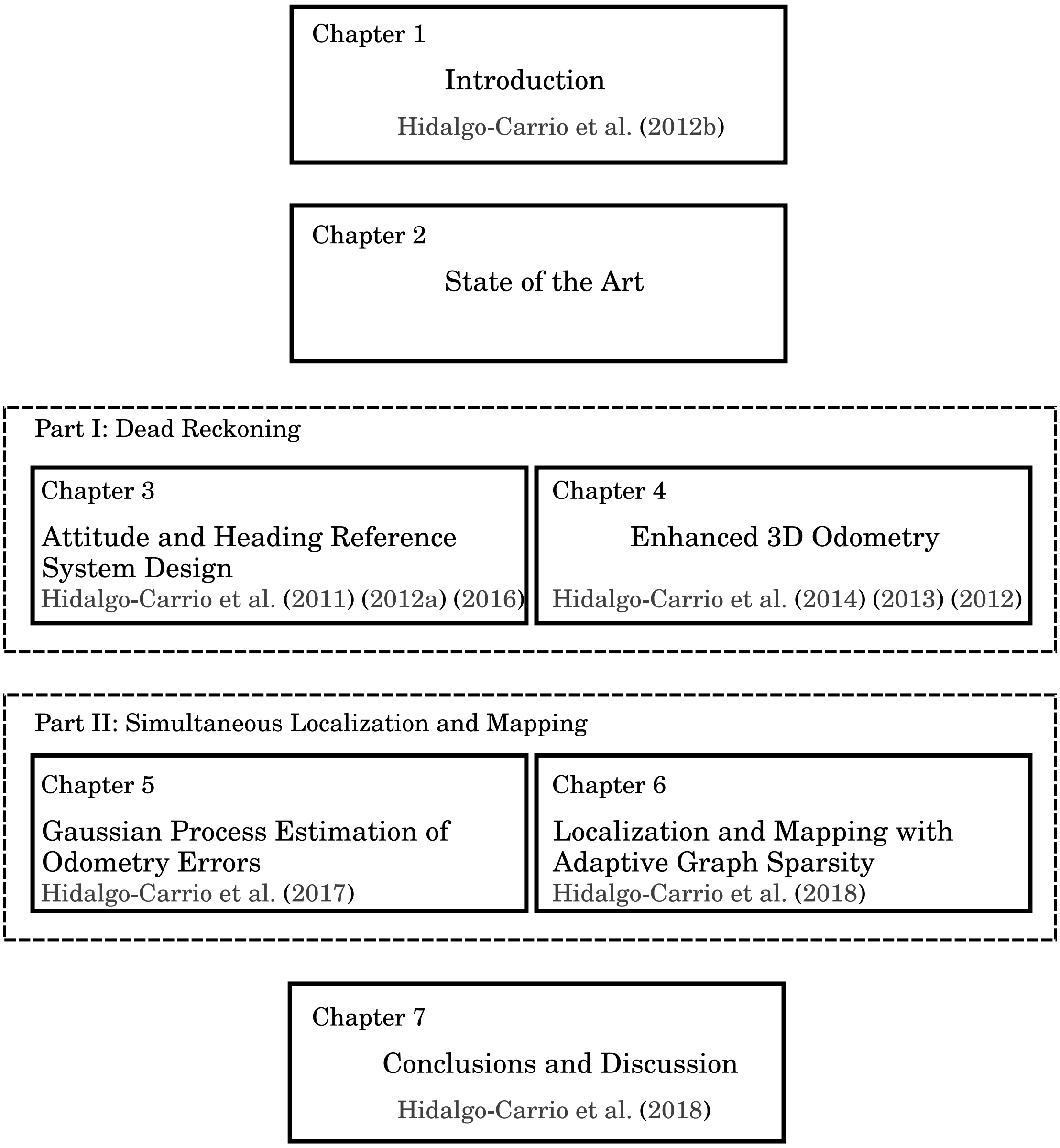

Chapter 1
Introduction
Hidalgo-Carrio et al. (2012b)
Chapter 2
State of the Art
Part I: Dead Reckoning
Chapter 3
Attitude and Heading Reference System Design
Hidalgo-Carrio et al. (2011) (2012a) (2016)
Chapter 4
Enhanced 3D Odometry
Hidalgo-Carrio et al. (2014) (2013) (2012)
Part II: Simultaneous Localization and Mapping
Chapter 5
Gaussian Process Estimation of Odometry Errors
Hidalgo-Carrio et al. (2017)
Chapter 6
Localization and Mapping with Adaptive Graph Sparsity
Hidalgo-Carrio et al. (2018)
Chapter 7
Conclusions and Discussion
Hidalgo-Carrio et al. (2018)

Figure 1.6: Structure of this thesis.

Chapter 2

State of the Art

This thesis is at the point of confluence between the technological needs for future planetary rovers and modern SLAM systems. This chapter gives an overview of the state-of-the-art of SLAM and a historical evolution of planetary rovers over the years.

2.1 Simultaneous Localization and Mapping: A Brief Survey

Though several research groups have worked on mapping and localization since early in the 80's, Moravec and Elfes (1985); Brooks (1985), the first complete SLAM implementation is from 1991, Moutarlier (1991), and the acronym SLAM first appears at the International Symposium on Robotics Research (ISRR) in 1995, Leonard and Durrant-Whyte (1991) and Durrant-Whyte et al. (1995). SLAM consists of estimating the robot pose with on board sensors and the construction of a model of the environment (the map). The popularity of SLAM is related to the emergence of robots from indoor and structure scenarios into increasingly outdoor and unstructured environments. Solving the SLAM problem is a good mix of probabilistic inference, geometry, graph theory and optimization. In practice, the SLAM problem also involves system integration, motion models, sensor characterization and calibration. This section focuses on establishing the foundations and principles to solve the SLAM problem as well as the related work available in the literature.

The SLAM technique has evolved considerably over the years from the first probabilistic formulations into a solid mathematical background. The SLAM problem considers a mobile robot moving without a priori knowledge of the environment. Fig. 2.1 shows the SLAM problem, which consists of a mobile robot (triangle) detecting landmarks or features (stars) in the environment and at the same time the robot uses the map (set of landmarks) to deduce its pose. *Loop closure* is the task when the robot revisits a previous visited area. Task driven adaptive data association and

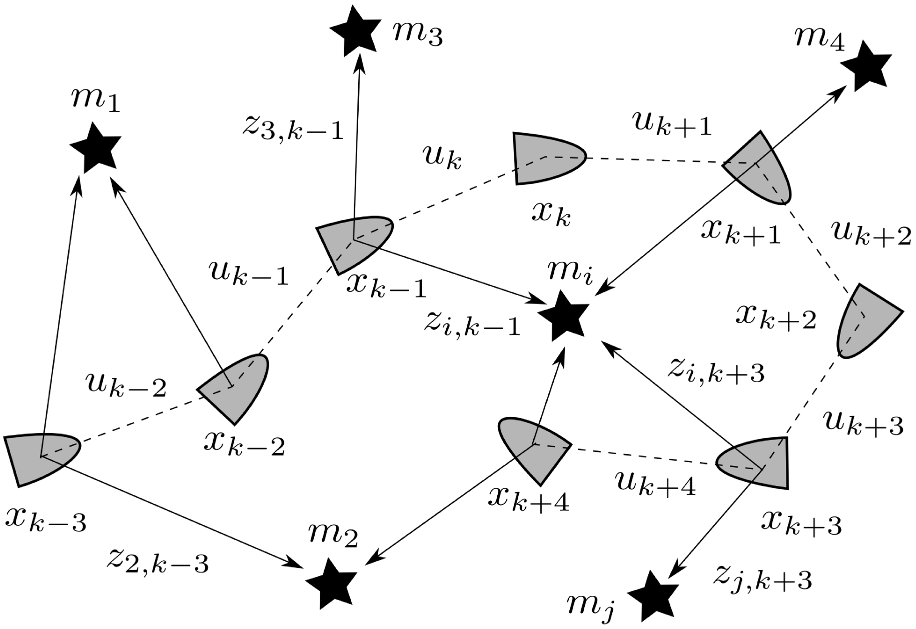


Figure 2.1: The SLAM graphical model representation, the robot (triangle) builds a map of the environment (start landmarks) and simultaneously estimates its location.

robust loop closure techniques are some of the challenges in modern SLAM.

The formulation of SLAM is usually described by means of probabilistic terms due to noisy measurements of the environment. The robot is moving along a trajectory described by a set of random variables $\boldsymbol{x}_k$ at time t. The sequence of 6D poses or three dimensional robot path is the given by

$$\boldsymbol{x}_{0:k} = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_k\} \tag{2.1}$$

where $\boldsymbol{x}_0$ is an arbitrary starting position and $k \to \infty$. The robot moves and consequently acquires a sequence of proprioceptive odometry measurements. Odometry provides relative information between two consecutive locations using a motion model. Let $\boldsymbol{u}_k$ denote the odometry measurement that relates the pose at time $k-1$ and time k. The sequence is denoted as

$$\boldsymbol{u}_{1:k} = \{\boldsymbol{u}_1, ..., \boldsymbol{u}_k\}. \tag{2.2}$$

In a perfect motion model with noise-free measurements, $\boldsymbol{u}_{1:k}$ would be sufficient to deduce the robot trajectory and reconstruct the environment, except for detecting loop closures. However, as described in this thesis, even with precise 3D odometry models, the $\boldsymbol{u}_k$ measurements diverge from the truth. The robot is equipped with exteroceptive sensors to sense objects or features in the environment and establish their relative location. The set of landmarks observations is given by

$$\boldsymbol{z}_{1:k} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_k\} \tag{2.3}$$

where $\boldsymbol{z}_i$ is all observations of landmark $\boldsymbol{m}_i$ from different robot locations $\boldsymbol{z}_i = \{\boldsymbol{z}_{i,0}, ..., \boldsymbol{z}_{i,k}\}$. The fundamental problem is to estimate the model of the world $\boldsymbol{m}$ as a set of landmarks $\{\boldsymbol{m}_1, ..., \boldsymbol{m}_k\}$ and the robot trajectory $\boldsymbol{x}_{0:k}$. The SLAM problem is then defined as the joint posterior probability over $\boldsymbol{x}_{0:k}$ and $\boldsymbol{m}$ using the available data measurements $\boldsymbol{u}_{1:k}$ and $\boldsymbol{z}_{1:k}$. The formulation is given by

$$p(\boldsymbol{x}_{0:k}, \boldsymbol{m} | \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}). \tag{2.4}$$

The variables on the right side of the bar are called observable variables. The ones on the left size of the bar are referred to as hidden, latent or unobservable variables and are the variables of interest. Here the SLAM community distinguishes two forms of SLAM. One is known as the *full* SLAM problem and describes the estimation of a posterior over the complete entire robot trajectory with the map, Grisetti et al. (2010a). This is described in equation (2.4). The other form is the *online* SLAM in which the current rover location is recovered, instead of the complete trajectory, and the map of the environment, Durrant-Whyte et al. (2006). The latter is defined as

$$p(\boldsymbol{x}_k, \boldsymbol{m} | \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}). \tag{2.5}$$

In the literature such online SLAM is known as *filters* and the full SLAM is also called *optimizers* or *smoothers*. To solve either form of the SLAM problem, the technique requires descriptions of the access to the observable variables. These observable variables are measurements of the world through mathematical models. Mathematical functions describe such a model in a noise-free world. The function f describes the movement of the robot or the sensor model. Those mathematical forms accept the last robot location x_{k-1} and sensor measurement x_k as inputs

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k). \tag{2.6}$$

Equation (2.6) function ignores the noise and external perturbation of the world, which entails non-deterministic behaviours. The modeling of such noise adds an uncertainty term ϵ_k and then $\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \epsilon_k$. The model becomes a probability distribution that peaks at the ideal value of $f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k)$. The motion model of the robot is described as state transitions in the form of

$$p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}, \boldsymbol{u}_k). \tag{2.7}$$

The motion model is assumed to be a Markov process in which the next state $\boldsymbol{x}_k$ depends only on the previous state $\boldsymbol{x}_{k-1}$ and the odometry measurement input $\boldsymbol{u}_k$. This document dedicates Chapter 4 to developing a 3D odometry model in order to demonstrate the thesis that a complete motion model improves the state transition estimation.

Likewise the observation model describes the probability of making a landmark measurement when robot pose and landmark locations are known, in the form of

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{m}). \tag{2.8}$$

The model equation indicates that the measurement depends on the robot location and landmark locations. Only in the case where the robot location and the map are known, observations are independent. The Bayes rule transforms the mathematical relationships into a form of equations in order to estimate the posterior probability distribution of the latent variables from the measurement values.

2.1.1 Bayes Filtering

The online SLAM solves the problem in two recursive steps: the prediction and the correction form. The prediction step is the projection of the robot pose given the odometry measurements u_k, in the form of

$$p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_k)\ p(\boldsymbol{x}_{k-1}, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k-1}) d\boldsymbol{x}_{k-1}. \tag{2.9}$$

The prediction equation performs a marginalization which equates to an integral over $\boldsymbol{x}_{k-1}$ in the continuous domain. The chain rule splits the joint probability into the prior $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_k)$ and the prediction $p(\boldsymbol{x}_{k-1}, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k-1})$ deduced from equation (2.8) for the robot state $\boldsymbol{x}_k$ and map $\boldsymbol{m}$ at a time k based on all previous information.

The correction or measurement step uses the Bayes rule in order to estimate the posterior from the prior, deduced from equation (2.9) as

$$p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{m}) p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})}{p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})}. \tag{2.10}$$

The process of building a map is formulated as the conditional density $p(\boldsymbol{m}|\boldsymbol{x}_{0:k}, \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})$ which uses all the observation from different locations to estimate the map $\boldsymbol{m}$. The counterpart is the estimation of the robot location formulated as $p(\boldsymbol{x}_{k|k}, \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}, \boldsymbol{m})$. The online SLAM problem, as described in equation (2.5), estimates both latent variables, robot last pose $\boldsymbol{x}_k$ and the updated map $\boldsymbol{m}$ in a recursive manner.

The remaining formulation is to find an appropriate representation of the noise. The most common representation is an additive Gaussian model, leading to the Extended Kalman Filter (EKF) formulation of the online SLAM, Bailey and Durrant-Whyte (2006). The alternative method is to use non-parametric filtering known as Particle filter. The most common implementation of the Particle filter is the Rao-Blackwellized particle filter, Doucet et al. (2000) and FastSLAM algorithm, Montemerlo et al. (2002) and Montemerlo et al. (2003).

2.1.1.1 EKF-SLAM

The EKF algorithm represent the observable variables as a multivariate Gaussian distribution. The motion model of the robot is described by state transitions in the form of

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) \approx \mathcal{N}(f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k), \mathcal{R}_k) \tag{2.11}$$

where $\mathcal{N}$ denotes a Normal distribution and $\mathcal{R}$ is the noise covariance. Consequently, the observation model is described in the form

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{m}) \approx \mathcal{N}(h(\boldsymbol{x}_k, \boldsymbol{m}), \mathcal{Q}_k) \tag{2.12}$$

where $h(\cdot)$ describes the observation function and the noise terms in $\mathcal{Q}$ are the measurement errors. The EKF-SLAM formulation estimates the mean

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{k|k} \\ \hat{\boldsymbol{m}}_k \end{bmatrix} = E \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{m} \end{bmatrix} |\boldsymbol{z}_{1:k} \Bigg] \tag{2.13}$$

and covariance

$$P_{k|k} = \begin{bmatrix} P_{xx} & P_{xm} \\ P_{xm}^{\intercal} & P_{mm} \end{bmatrix}_{k|k} = \left[\begin{pmatrix} \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k \\ \boldsymbol{m} - \hat{\boldsymbol{m}}_k \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k \\ \boldsymbol{m} - \hat{\boldsymbol{m}}_k \end{pmatrix}^{\intercal} |\boldsymbol{z}_{1:k} \right]. \tag{2.14}$$

A Gaussian distribution is fully modeled by its mean and covariance. The advantage of this formulation is that the algorithm computes as an algebraic form of multiplying matrices and vectors. The prediction step is then in the form

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k) \tag{2.15}$$

$$P_{xx,k|k-1} = \nabla f P_{xx,k-1|k-1} \nabla f^{\intercal} + \mathcal{R}_k \tag{2.16}$$

where ∇f is the Jacobian of odometry model f evaluated at the robot pose estimate $\boldsymbol{x}_{k-1|k-1}$, Durrant-Whyte et al. (2006). In a static environment there is no need to perform a prediction on the features, however it is required for landmarks that move (i.e. dynamic environments). This is not formulated in this survey since for planetary rovers moving landmarks are removed from the map. Afterwards, the observation step follows the known Kalman update rule

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{k|k} \\ \hat{\boldsymbol{m}}_k \end{bmatrix} = \left[\hat{\boldsymbol{x}}_{k|k-1}\hat{\boldsymbol{m}}_{k-1}\right] + K_k \left[\boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1}, \hat{\boldsymbol{m}}_{k-1})\right] \tag{2.17}$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^{\intercal} \tag{2.18}$$

where K_k is the Kalman gain or matrix, which is computed as

$$K_k = P_{k|k-1} \nabla h^{\intercal} S_k^{-1} \tag{2.19}$$

and the S_k is the measurement prediction covariance matrix derived as

$$S_k = \nabla h P_{k|k-1} \nabla h^{\intercal} + Q_k. \tag{2.20}$$

The ∇h is the Jacobian matrix of the observation function h evaluated at $\hat{\boldsymbol{x}}_{k|k-1}$ and $\hat{\boldsymbol{m}}_{k-1}$. The derived EKF Kalman equations are the standard formulation. Other variants of the filter are the Sigma-Point Kalman filters or the Unscented Kalman Filter (UKF), Merwe et al. (2004).

The observation step updates all the landmarks and joint covariance matrix every time there is a new observation. This entails a computational cost which grows quadratically with the number of landmarks. There has been significant work in developing alternatives to achieve real-time implementations. Most of the variants adopts the *structureless* approach since the landmarks are not present in the state vector. The State Cloning Kalman Filter (SC-KF), Mourikis et al. (2007b), is one of the most mathematically elegant solutions. The approach requires a copy of the state vector at previous time, which is called *stochastic cloning*. The first copy is the state at the latest exteroceptive measurement while the second copy represents the robot current state. At the prediction step of the filter the current state is propagated while the copy stays unchanged. It is at the correction step when the copy is updated and the copy is reset to the new measurement. This approach delays linearization while having complexity only linear in the number of features. An evolution of the algorithm, the Multi-State Cloning Kalman Filter (MSC-KF) described in Mourikis et al. (2007a), takes the concept further to store multiple copies of the sensor pose in the vector state. The concept of cloning makes the covariance more accurate as a monotonically increasing function and therefore a better estimate. However, a structureless filter with no features in the state vector entails marginalization of the landmarks position which is a source of error.

EKF-SLAM performs linearization at every recursive step of the last state vector. Marginalization and erroneous linearization adds spurious information which makes the filter very sensitive to outliers. An effect of the inconsistency is that the estimation becomes over-confident which results in a non-optimal estimation. Convergence and consistency can only be guaranteed in the linear case. The problem has been addressed in the literature to perform the so-called observability analysis by Hernandez et al. (2015). The Observable-Constrained EKF (OC-UKF) described in Kottas et al. (2012); Huang et al. (2009), analyses the unobservable directions which permit the filter to update only in the observable directions. This is because the observability properties of the linearised model are key to determine filter consistency. The methodology alters the computation of the Jacobian in order to achieve the desired observability properties. An elaborate analysis of Visual-Inertial SLAM is available in Martinelli (2012); Hesch et al. (2014).

Those inconsistencies in the EKF-SLAM formulation make the loop closure particularly difficult and intractable for long-term SLAM solutions.

2.1.1.2 Particle Methods

Sequential Monte Carlo filter or Particle methods are non-parametric models that represent multimodal distributions (i.e. non-Gaussian). Particle filters approximate posteriors by a finite number of values or samples instead of a fixed functional form of the posterior $p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})$. The quality of the approximation relies on the number of samples used to represent the posterior. As the number of parameters increases the estimation converges to the true posterior. The approximation is based on a set of N samples $\{\boldsymbol{\alpha}^i\}_{i=1}^N$ referred to as *particles*. The theoretical derivation is based on a general expectation $\mathbf{E}$ for an arbitrary function $g(\cdot)$ over a probability distribution $p(\cdot)$ expressed as

$$\mathbf{E}_{g(\cdot)}(p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})) = \int g(\boldsymbol{x}_k, \boldsymbol{m})p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})d\boldsymbol{x}_k. \tag{2.21}$$

This is known as the indicator function and its integral can be approximated with a collection of particles $\boldsymbol{\alpha^i}$ drawn from the distribution $p(\cdot)$. The approximation formulates as

$$\mathbf{E}_{g(\cdot)}(p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})) \approx \frac{1}{N}\sum_{i=1}^{N} g(\boldsymbol{\alpha}_k^i) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{\alpha}_k^i}(\boldsymbol{x}_k, \boldsymbol{m})_{|N\to\infty} \tag{2.22}$$

where δ stands for the Dirac function at a given sample $\boldsymbol{\alpha}^i$. The speed of convergence of this estimate depends on N the number of particles. In practice, it can be difficult to draw samples from the posterior distribution $p(\cdot)$ because it is multidimensional and only known up to a certain constant value. The alternative is the Bayesian Importance Sampling (IS) method. The IS method uses a *proposal* distribution $\pi(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})$ which approximates the true distribution $p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})$ and the samples are given a set of importance weights $\{\omega^i\}_{i=1}^N$ to compensate for this discrepancy. The weights are given as

$$\omega_k^i = \omega(\boldsymbol{\alpha}_k^i, \boldsymbol{m}^i) = \frac{p(\boldsymbol{\alpha}_k^i, \boldsymbol{m}^i|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})}{\pi(\boldsymbol{\alpha}_k^i, \boldsymbol{m}^i|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})}. \tag{2.23}$$

Therefore, equation (2.22) can now be expressed using the proposal distribution $\pi(\cdot)$ and the weights

$$p(\boldsymbol{x}_k, \boldsymbol{m}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k}) \approx \frac{1}{N}\sum_{i=1}^{N} \omega_k^i \delta_{\boldsymbol{\alpha}_k^i}(\boldsymbol{x}_k, \boldsymbol{m}). \tag{2.24}$$

The high dimensional space of the SLAM problem (i.e. pose and landmarks) makes the direct application of the Particle method infeasible due to the computational cost. The method so-called Rao-Blackwellized, in which the joint state from equation (2.4) and (2.21) is partitioned using the product rule. This technique make sampling methods feasible in computational time. Further information appears in the original work by Doucet et al. (2000). The joint SLAM state divides into the map and the robot pose as

$$p(\boldsymbol{x}_{0:k}, \boldsymbol{m}|\boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}) = p(\boldsymbol{m}|\boldsymbol{x}_{0:k}, \boldsymbol{z}_{1:k})\, p(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}). \tag{2.25}$$

This trick is an important property to speed up the particle filter. The landmarks become independent when related to the complete trajectory $\boldsymbol{x}_{0:k}$ instead of the last pose $\boldsymbol{x}_k$. The map can be computed analytically as a set of independent Gaussian landmarks, with linear complexity instead of a joint map distribution with quadratic cost.

At a certain time the joint state is given by a set of weights, particles and the map as $\{\omega_k^i, \boldsymbol{\alpha}_{0:k-1}^i, p(\boldsymbol{m}|\boldsymbol{\alpha}_{0:k-1}^i, \boldsymbol{z}_{1:k-1})\}_i^N$. The Rao-Blackwellized particle filter for SLAM is implemented as follows.

- When a new odometry measurement is received, compute a proposal distribution for each particle from the particle history,

$$\boldsymbol{\alpha}_k^i \sim \pi(\boldsymbol{x}_k|\boldsymbol{\alpha}_{0:k-1}^i, \boldsymbol{z}_{0:k}, \boldsymbol{u}_k). \tag{2.26}$$

The new particle is joined to the existing set of particles $\boldsymbol{\alpha}^i_{0:k} \triangleq \{\boldsymbol{\alpha}^i_{0:k-1}, \boldsymbol{x}^i_k\}$.

- Weight the samples using the Important Sampling method. The numerator is formed with the measurement model and the motion model,

$$\omega^i_k = \omega^i_{k-1} \frac{p(\boldsymbol{z}_k|\boldsymbol{\alpha}^i_{0:k}, \boldsymbol{z}_{1:k-1})\; p(\boldsymbol{\alpha}^i_k|\boldsymbol{\alpha}^i_{k-1}, \boldsymbol{u}_k)}{\pi(\boldsymbol{x}_k|\boldsymbol{\alpha}^i_{0:k-1}, \boldsymbol{z}_{0:k}, \boldsymbol{u}_k)}. \tag{2.27}$$

  The measurement model differs from the original one in equation (2.12) since the map is marginalized from the observations as

$$p(\boldsymbol{z}_k|\boldsymbol{x}_{0:k}, \boldsymbol{z}_{1:k-1}) = \int p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{m})\; p(\boldsymbol{m}|\boldsymbol{x}_{0:k-1}, \boldsymbol{z}_{1:k-1}) d\boldsymbol{m}. \tag{2.28}$$

- Perform a particle resampling step when necessary. This step selects particles with their associated maps and probability proportional to their weights. Selected particles are given a uniform weight distribution, $\omega^i_k = 1/N$. The frequency of resampling differs between implementations, though the most common is once the weights are below a certain threshold. Some implementations resample every time step or a fixed number of steps.

- Perform a EKF update for each particle with the observed landmark and the known pose. This entails N Kalman updates but is simpler and computational more efficient than propagating particles.

Each particle has the associated map as an independent Gaussian distribution and the resulting map can be computed analytically, given by

$$p(\boldsymbol{m}|\boldsymbol{\alpha}^i_{0:k}, \boldsymbol{z}_{1:k}) = \prod_j^M p(\boldsymbol{m}_j|\boldsymbol{\alpha}^i_{0:k}, \boldsymbol{z}_{1:k}). \tag{2.29}$$

A *fast* implementation of the Rao-Blackwellized filter addressed in the literature is FastSLAM, Montemerlo et al. (2002). The algorithm modifies the proposal distribution directly using the motion model and the last odometry readings. The probabilistic sampling step is easily implemented for any robot for which kinematics can be computed by

$$\boldsymbol{\alpha}^i_k \sim p(\boldsymbol{x}_k|\boldsymbol{\alpha}^i_{k-1}, \boldsymbol{u}_k). \tag{2.30}$$

Consequently, from equation (2.31), the samples are weighted according to the marginalized observation model as

$$\omega_k^i = \omega_{k-1}^i p(\boldsymbol{z}_k | \boldsymbol{\alpha}_{0:k}^i, \boldsymbol{z}_{1:k-1}). \tag{2.31}$$

The Particle filter implemented in FastSLAM has a set of interesting properties: Rao-Blackwellized, conditional independence and particle resampling. Each particle in FastSLAM carries a sample of the complete path but the actual update equation uses the most recent pose, making the FastSLAM a filter. It also makes data association easier since the adoption is made at each particle and not for the complete filter as in the EKF-SLAM algorithm. This property makes particle methods more robust against outliers. Particle filtering can be efficiently implemented using tree methods in order to partition the map. Grid-based maps have been introduced in Grisetti et al. (2007a). In their work, the Gaussian model for the landmarks is replaced by a occupancy grid map. DP-SLAM uses ancestry tree methods to update grid-based maps, reported in Eliazar and Parr (2004). This approach improves map representation using laser range finder and loop closing in deterministic environments. One of the benefits of using sampling methods is the possibility to model non-Gaussian and multimodal distributions. The work described in Stachniss et al. (2007) shows that in around $5\,\%$ of test cases the Gaussian approximation is not sufficient to model the likelihood distribution which might cause divergence of the filter. Therefore, the work proposes a multimodal distribution while maintaining the same efficiency as the Gaussian proposal, Stachniss et al. (2007).

Empirical results have shown that Particle methods can solve the SLAM problem at the cost of high computational requirements. In practice, these methods also suffer from several problems. Loop closure can create particle depletion. This prevents the system from estimating consistent global maps. Some resampling techniques are investigated in Grisetti et al. (2007a); Stachniss et al. (2005) and Grisetti et al. (2007b) which improve but do not completely solve the problem. In addition, the number of particles to compute SLAM is set manually based on an initial guess. In outdoor environments, with a robot moving in a six dimensional space, the particle filter greatly increases the number of samples. Generally, these penalties make Particle methods difficult to scale for long-term SLAM solutions.

2.1.2 Smoothing

Kalman and Particle filters are recursive with the Markov property. As previously noticed, a recursive form involves marginalization of past measurements into the last estimate. This makes filter methods particularly sensitive to spurious information and a single outlying measurement can corrupt the estimate. Smoothing methods incorporate past measurements into the estimation of the current state. The terminology refers to *smoothing* because the estimate is *smoothed* by passing through the measurement history. Most practical approaches use unimodal Bayesian smoothers which models the uncertainty as a Gaussian probability distribution, called Kalman smoothers. Kalman smoothers are *optimal* in the sense that all available information is incorporated to make the best estimate that is mathematically achievable.

Solving the SLAM problem using smoothing starts with the full SLAM problem definition from equation (2.4). Solving the equation consists of finding the Maximum a Posteriori (MAP) estimation of the robot's trajectory $\boldsymbol{x}_{0:k}$ and the map m. A solution to the SLAM problem would not be possible without a well defined structure. The structure used in smoothing is a graph formulation of the SLAM problem shown in Fig. 2.1. Such graph representation is formally derived from a Dynamic Bayesian Network (DBN) which describes a stochastic process with a directed graph structure. Landmarks and robot poses are stored in the nodes while measurements are represented by edges or arcs in the graph. Two consecutive poses $\boldsymbol{x}_{k-1}$, $\boldsymbol{x}_k$ are related together by an edge which represents the odometry measurement $\boldsymbol{u}_k$. Other edges are given by the feature measurement $\boldsymbol{z}_{k,k}$ of a landmark k at a certain pose $\boldsymbol{x}_k$. Building the graph is performed online while the robot is traversing the environment leading to a continuously increasing size. Edges in the graph are soft constraints and solving the optimization problem is also referred as *relaxing* these constraints. Sometimes mentioned in the literature as *relaxation*. Computing the SLAM solution is analogous to computing the state of minimal energy of this model. The goal of a Maximum a Posteriori (MAP) approach is to maximize the log-posterior given as

$$\hat{\boldsymbol{x}}_{0:k}, \hat{\boldsymbol{m}} = \underset{\boldsymbol{x}_{0:k}, \boldsymbol{m}}{\arg\max}\, log(p(\boldsymbol{x}_{0:k}, \boldsymbol{m} | \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})) \tag{2.32}$$

$$\hat{\boldsymbol{x}}_{0:k}, \hat{\boldsymbol{m}} = \underset{\boldsymbol{x}_{0:k}, \boldsymbol{m}}{\arg\max} \prod_{k}^{N} p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}, \boldsymbol{u}_k) \prod_{i}^{N} p(\boldsymbol{z}_{i,k} | \boldsymbol{x}_k, \boldsymbol{m}_i). \tag{2.33}$$

Each constraint of the form $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_k)$ is the motion model given by the odometry input $\boldsymbol{u}_k$. Likewise, $p(\boldsymbol{z}_{i,k}|\boldsymbol{x}_k)$ is the landmark measurement model, and both models correspond to edges in the graph structure. Under a Gaussian noise assumption, the motion model corresponds to equation (2.11) and the measurement or observation model to equation (2.12). The MAP estimate for the trajectory $\hat{\boldsymbol{x}}_{0:k}$ and the map $\hat{\boldsymbol{m}}$ is obtained by minimizing the negative log of the joint probability as

$$\hat{\boldsymbol{x}}_{0:k}, \hat{\boldsymbol{m}} = \underset{\boldsymbol{x}_{0:k}, \boldsymbol{m}}{\arg\min} -log(p(\boldsymbol{x}_{0:k}, \boldsymbol{m} | \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})). \tag{2.34}$$

Substituting the process and measurement models, gives the following nonlinear least squares problem

$$\hat{\boldsymbol{x}}_{0:k}, \hat{\boldsymbol{m}} = \underset{\boldsymbol{x}_{0:k}, \boldsymbol{m}}{\arg\min} \{ \parallel \sum_{k}^{N} f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) - \boldsymbol{x} \parallel_{\mathcal{R}_k}^2 + \parallel \sum_{i}^{N} h(\boldsymbol{x}_k, \boldsymbol{m}_i) - \boldsymbol{z}_{i,k} \parallel_{\mathcal{Q}_i}^2 \} \tag{2.35}$$

where $\parallel \boldsymbol{e} \parallel_{\Sigma}^2 = \boldsymbol{e}^{\intercal} \Sigma^{-1} \boldsymbol{e}$ for the squared Mahalanobis distance with covariance Σ. This is a sparse nonlinear least square system. Sparse because each node in the graph is only connected to a small number of other nodes. The number of constraints in the graph is (at worst) linear in the time elapsed and in the number of nodes in the graph. The odometry model f and the measurement mode h are nonlinear for which a good linearization point is not available. Common optimization methods, such as Levenberg–Marquardt or Gauss–Newton, are typically used to solve a succession of linear approximations to equation (2.35). It is worthwhile to notice, this is similar to EKF-SLAM but iterating repeatedly to approach the minimum, therefore avoiding the problem deduced from incorrectly linearization. The least squares is on a manifold due to the rotational components over the non-Euclidean 3D Special Orthogonal Group $SO(3)$. A smooth manifold is a non-Euclidean mathematical space, but can be seen as Euclidean on a local scale. The manifold structure is an important part of modern SLAM implementations, but conceptually the SLAM problem is equivalent. The reader can find abundant material about the manifold structure of the problem in Hertzberg (2008); Grisetti et al. (2010b).

Graphical SLAM scales better to higher-dimensionality maps in comparison to EKF-SLAM, in which the update time is quadratic with the size of the map. The update time of the graph is constant and the memory required is, for most cases, linear. These two reasons make graphical SLAM methods more suitable for outdoor robotics than Bayes filtering methods. However, computing the optimization can be expensive and depends on the constraints in the graph. For this reason, two approaches for smoothing are commonly discussed and presented here:

- *Fixed-lag smoothing* estimates the states within a given window and perform marginalization with the other nodes in the graph. This approach is more accurate than EKF-SLAM since it takes a bigger number of past measurements. Fixed-lag is also more robust against outliers in the measurements than filters. However, marginalization of states out of the window entails less sparsity which leads to inefficiency in the estimate. Interesting works are available in the literature: Mourikis and Roumeliotis (2008); Sibley et al. (2010); Dong-Si and Mourikis (2011); Leutenegger et al. (2015).

- *Full smoothing* or batch nonlinear least squares approaches are methods that estimate the complete history of poses and landmarks (map). This approach is also called full SLAM and guarantees the highest accuracy. The computational cost quickly becomes infeasible for real-time performance, as the pose and map grows with time. However, there is significant research on the full SLAM problem. Kummerle et al. (2011a,b) is an interesting work which provides online estimation of calibration parameters in a graph SLAM. Recently, other research proposed target variables to deal with smaller graphs, reduce the computational cost and enhance robustness, Carlone et al. (2014). An important development has been the incremental smoothing and mapping from Kaess and Ranganathan (2008); Kaess et al. (2011a,b), which take advantage of factor graphs to maintain efficiency and update a subset of nodes affected by a new measurement using a novel data structure called a Bayes tree. These researches make full smoothers very attractive for real-time SLAM systems.

2.1.3 Relation of Paradigms

The three different methods described here cover most of the work done in SLAM. EKF-SLAM was mainly implemented during the first years of the SLAM problem. It set the mathematical principles of the SLAM problem and helped to analyze the challenges related with optimization and robust data association. At the same time, Particle methods became popular due to their robustness against outliers and multidimensional modeling of the uncertainty distribution. EKF-SLAM and Rao-Blackwellized methods covered most of the implementations during the first 20 years of SLAM. EKF-SLAM has been implemented with a small set of landmarks, mostly in structured environments, since the computational cost is quadratic with the size of the map. Few exceptions, such as the SC-KF mentioned in Section 2.1.1.1, keep the computational cost linear with the number of features. However, filter consistency needs to be ensured with a tedious observability-constrained analysis. Particle methods are mostly used with a reduced dimensionality since the number of particles exponentially grows with the dimension of the state space. Particle depletion makes loop closure difficult in larger maps and outdoor environments. Hierarchical SLAM, which is particle filters with smoothing methods, solves the loop closure limitation as in the Embodied SLAM described in Schwendner et al. (2013), with an increase in robustness and computational cost. Recent research in modern SLAM uses the full SLAM paradigms to solve larger maps in outdoor scenarios, Leutenegger et al. (2015). Full smoothers with graph structures are very suitable to incrementally adding new measurements. A reader familiarized with visual methods might notice some similarities of full SLAM smoothers to Bundle Adjustment (BA) and Structure from Motion (SfM). Both paradigms optimize a 3D structure but smoothers solve the optimization problem incrementally each time a new measurement is included in the graph. Additionally, the measurement model in smoothing methods can be from any type of sensor while cameras models only lead with projective geometry in SfM. In addition, smoothers pay attention to the probabilistic nature of the problem.

In summary, EKF-SLAM filters appear beneficial for small processing costs in confined environments, Particle methods are applicable for problems with a small state dimension, whereas smoothing methods are superior in large environments with promising results towards long-term SLAM.

2.2 Localization and Mapping for Planetary Rovers: A Historical Evolution

Planetary mobile robots, planetary rovers or space exploration vehicles are complex systems designed to perform a main task: to move across the surface of a celestial body or planet. The motivation behind to equippping *a spacecraft on wheels* with such capability widely varies from scientific purpose to cargo support or crew transportation in a human spaceflight mission. A historical evolution of flown planetary rovers is briefly introduced from two perspectives with relevance for this thesis: navigation, and therefore localization and mapping, and operations.

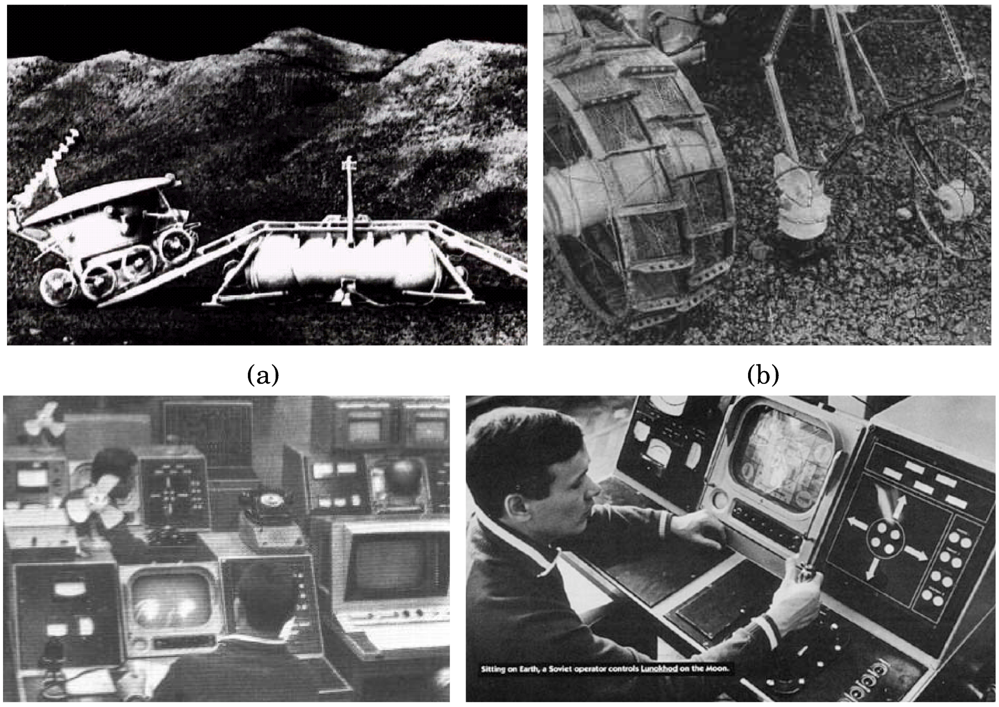
2.2.1 Lunokhod

The series of Soviet robotics mobile vehicles, Lunokhod-1 and Lunokhod-2 explored the Moon surface during the early 1970s. On 17 November 1970 Lunokhod-1 landed on the Moon surface and covered a distance of $10.5\,\mathrm{km}$ in teleoperation beaming thousands of images back to Earth. Less than three years later, on 15 January a replica landed in the Monnier crater covering $39\,\mathrm{km}$, between 16 January 1973 and 22 April 1973. Lunokhod rovers marked the longest distance of any vehicle on the lunar surface. Together they traveled about $50\,\mathrm{km}$ in 414 days on the Moon surface, and a third vehicle on the series was designed but never flew due to funding issues, Carrier (1992); Kemurdjian et al. (1993); Malenkov (2013).

Rovers are $170\,\mathrm{cm}$ long, $160\,\mathrm{cm}$ wide, $135\,\mathrm{cm}$ high and have $30\,\mathrm{cm}$ ground clearance with a total mass of $756\,\mathrm{kg}$. Lunokhod's principal scope was to support future manned missions and to explore the Moon surface, since not much information existed at that time. The chassis has a mass of $105\,\mathrm{kg}$ and consists of eight wheels, independent suspension, motor and brake, Carrier (1992). The rover only requires three wheels to remain operational. The wheel diameter, including the cleats, is $51\,\mathrm{cm}$ and $20\,\mathrm{cm}$ width. No steering capabilities are designed on the rover and skid steering was necessary during the traversal. The rover is equipped with two slow-scan television cameras attached to the front and several scientific instruments as telephotometers, spectrometer and X-ray telescope. The PROP (a Russian acronym for Surface Evaluation Instrument) is the most relevant instrument on board. Although it is a scientific instrument, the PROP is also used as part of the safety system to determine the degree of wheel slippage and to forecast further motion (i.e. dead reckoning). The PROP system consists of a extra free-rolling wheel and a penetrometer. Informally called the *ninth wheel*, it has spikes on its rim so that it could roll without slipping. Wheel odometry coming from the freely rotating wheel provides information on the distance traveled. This wheel is attached to a lever so it could move up and down freely. Its vertical movement provides information on the unevenness of the terrain that the Lunokhod covered (see Fig. 2.2b). The rovers are powered by batteries which were recharged during a lunar day using the solar panel. The opened solar panel provides energy during the two weeks of moon day, and when the lunar night approaches the lid closes down to protect the vehicle from the harsh environmental conditions. Overall the chassis power consumption for moving the rover on Moon surface is $320\,\mathrm{W}$, and a system of gears provided the rover with two speeds, $0.22\,\mathrm{m/s}$ and $0.56\,\mathrm{m/s}$, Kemurdjian (1990). From the learning experience in Lunokhod-1, the second version of the rover has a third television camera mounted high on the front for navigation purposes, since the two front cameras in the first rover design are located too low for a correct guidance.

Operators remotely operated the rover using a joystick and TV images from the cameras on board. Therefore, the navigation-control loop was closed by the operators on Earth. A delay of three seconds in transmitting commands and receiving a response, in addition to the complexity of determining sizes of surface features and distances, required nonstandard modes of driving. The operators had to acquire new habits and Lunokhod driver candidates were carefully trained in simulated conditions, which took account of the psychological and physiological characteristics of every member. As a result, the driver remotely controlled the Lunokhods and they avoided accidents.

Lunokhod-1 was designed for 90 days of operation and explored the lunar surface for 11 months and Lunokhod-2 did it for four months. However, Lunokhod-2 covered seven times the distance of Lunokhod-1, which enhances the contribution of the third camera by improving the navigation-control experience and capabilities to the operator. Lunokhods were equipped with photodetectors to correct its dead reckoning position using a laser supplied from Earth. By the end of the Lunokhod-1 mission, the rover's location was uncertain by at least 5 kilometers and stabs at getting laser returns all failed since early in the mission. Lunokhod-1 location remained unknown for nearly four decades, until the Lunar Reconnaissance Orbiter snapped photographs of the robot's tracks in 2010. Lunokhod-2's reflector is still functional today and its position is known to sub-meter accuracy, Western-Ontario (2010). Fig. 2.2


Sitting on Earth, a Soviet operator controls Lunokhod on the Moon.

(a) (b)

(c) (d)

Figure 2.2: Lunokhod mission (a) Lunokhod artistic impression of the deployment system from the lander. (b) Detail of the wheel and PROP device. (c) Lunokhod rover control station, (d) Lunokhod pilot Vyacheslav Dovgan during a rover teleoperation session. Images credit: Roscosmos

shows rover and operations details of the Lunokhod missions.

2.2.2 Lunar Roving Vehicle

The Apollo Lunar Rover Vehicle (LRV) (depicted in Fig. 2.3) is a four wheeled vehicle to support manned Moon exploration in the last three Apollo missions. The purpose of the rover was to assist astronauts in order to cover longer distance during the short time on the Moon surface. An essential discussion took place between safety and science return during the LRV definition, since the presence of a rover would limit the extra fuel for the Lunar Module (LM) in order to ensure a safe place to land on the moon. The LRV plan started in May 1969 two months before the first manned lunar landing (Apollo 11), with the objective of accomplishing more science in future missions. Apollo 11's crew covered about 230 m during their 21.5 h on the lunar surface, while in Apollo 15 (with the first LRV) the distance was further increased to 27.76 km, able to work 5.0 km away from the lander. The Apollo 17 LRV traveled a cumulative distance of approximately 35.9 km in less than five hours with a greatest distance from the LM of 7.6 km, reported in Boeing (1972).

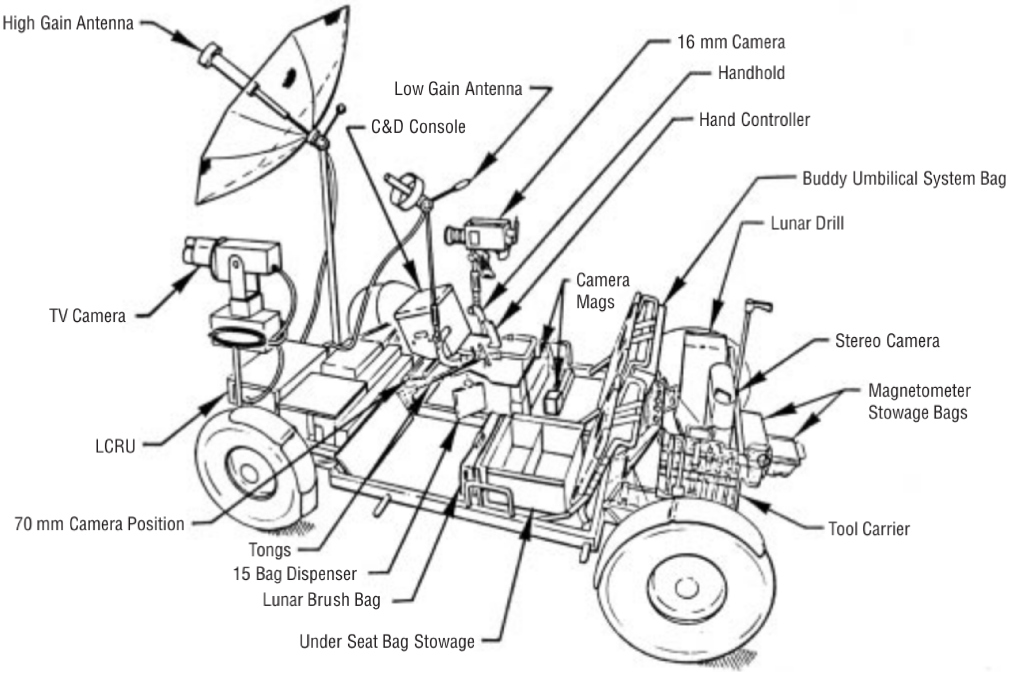


Figure 2.3: Artistic impression of the LRV in a deployed configuration with the instrumentation on board. Figure adapted from Wright et al. (2002).

The LRV passed into history as *moon buggy*, but the reality is that the LRV was a complex sophisticated exploration spacecraft. It was designed to carry two astronauts in their spacesuits and to operate in harsh vacuum and extreme temperatures ($\pm 150° C$) with maximum static and dynamic stability while minimizing weight and volume, able to remain on slopes up to $45°$. All moving parts were protected against lunar dust including sealing the wheel drive motors and traction assemblies. One last major design had to do with the necessity of having a reliable navigation system on board. It was necessary for the astronauts to know not only the position at any moment, but also the direct path back to the LM in case of an emergency.

The locomotion system consists of a chassis separated into three sections constructed on welded aluminum: the forward, center and rear. The forward supports the LRV batteries (two batteries), drive control electronics and navigation electronics as well as mechanical provisions for mounting a power package and antenna television. The center part of the chassis provides support for the astronauts and their seats, foot rests, the hand controller and display console as well as the voice communication antenna. The rear part consists of a platform with mechanical tie-down points for the astronauts' tolls and scientific equipment as well as space for the lunar material to collect. The wheels are approximately $23\,\mathrm{cm}$ wide with a radius of $40\,\mathrm{cm}$ and $5\,\mathrm{kg}$ weight. They are constructed using a spun on aluminum hub attached to a traction drive assembly and a woven mesh of zinc-coated steel wire to which a tread

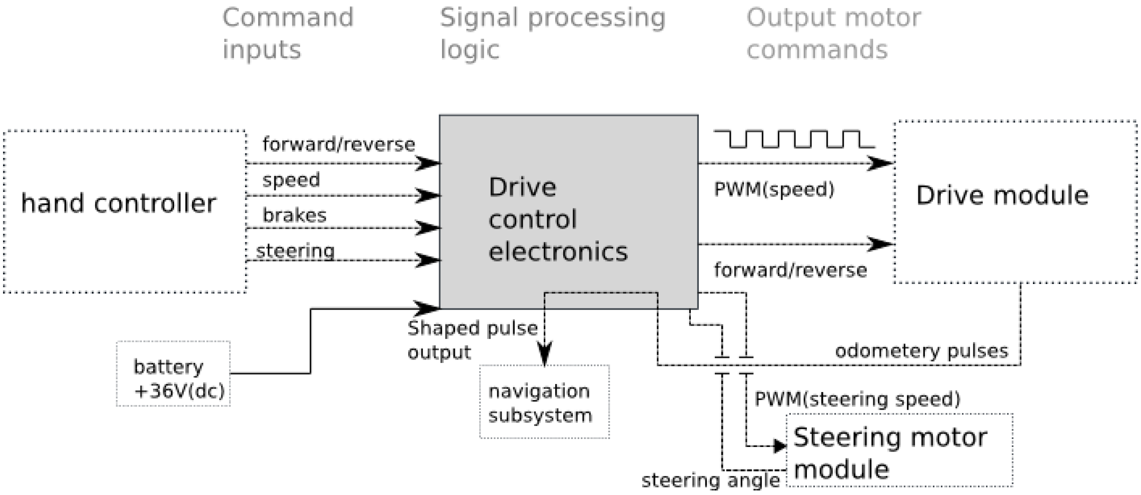


Figure 2.4: LRV drive control electronics operation schematics. Figure adapted from Boeing (1971).

following a chevron pattern was riveted. The wheel design permits climbs of 20° to 25° without significant slippage. The traction drive unit attached to the wheel hub consists of a DC brush-type motor with a harmonic drive reduction. Incorporated into each drive module is a magnetic reed switch that activates nine times per wheel rotation. These pulses are used in the odometry, speed and navigation calculations. The LRV has been tested to support speeds up to $14\,\mathrm{km/h}$ and negotiate rocks with $31\,\mathrm{cm}$ height. Two independent steering systems (front and rear) permit the vehicle to turn within its own length using a double Ackermann steering, providing a maximum maneuverability for locomotion, Boeing (1972).

The drive control electronics were located in the forward part of the chassis and were the electronic heart of the mobility system. The system accepted forward and reverse speed control signals as well as left or right steering commands using Pulse Width Modulation (PWM) (see Fig. 2.4). The system provided signal processing logic to the PWN signal that in turn furnished energy to traction and steering motors, depending on the speed command from the hand controller potentiometers (operated system). The controller was a T-shaped hand controller and display console located between the two astronauts. The controller was spring-loaded, which means it would return to a neutral steering position when released. The Control and Display (C&D) console consisted of an upper portion containing navigation system instruments and a lower portion containing monitoring electric loads. The console had an Attitude Indicator (AI) that provided pitch and roll information within a range of $\pm 25^\circ$, Boeing (1971). This indicator was read by astronauts and also reported to mission control center. The heading of the vehicle was determined using ephemeris data, a sun shadow device located in the upper portion of the console, which helped determine

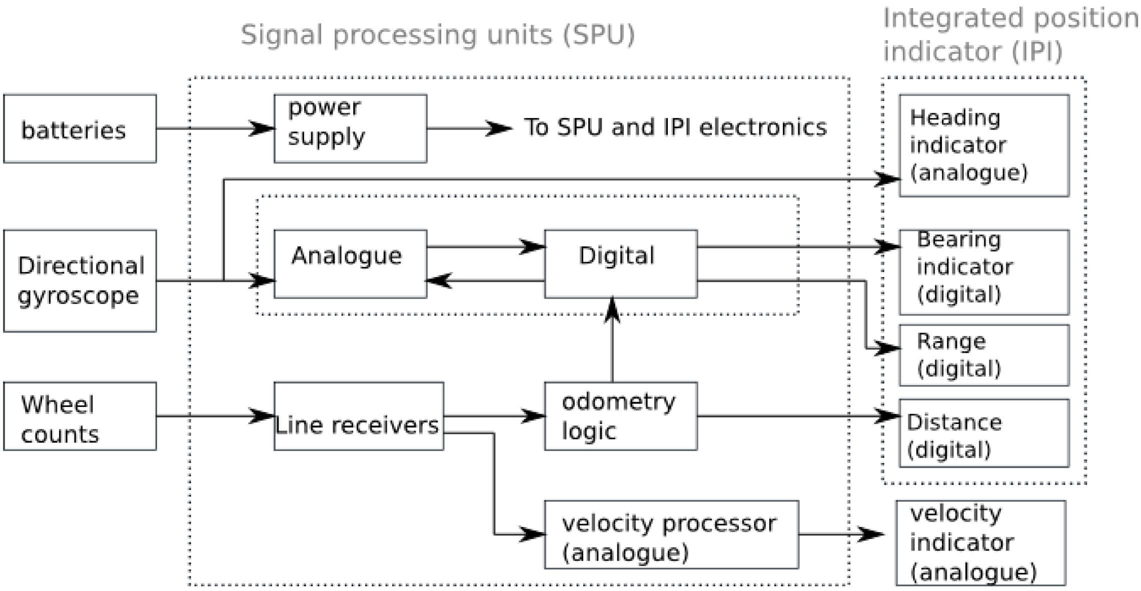

Signal processing units (SPU)
Integrated position indicator (IPI)
batteries
power supply
To SPU and IPI electronics
Heading indicator (analogue)
Directional gyroscope
Analogue
Digital
Bearing indicator (digital)
Range (digital)
Wheel counts
Line receivers
odometry logic
Distance (digital)
velocity processor (analogue)
velocity indicator (analogue)

Figure 2.5: LRV navigation system block. Figure adapted from Boeing (1971).

the LRV heading with respect to the sun. This measurement was compared with a directional or heading gyroscope on board in order to check against drift. Additionally, the attitude data were used by the mission control center during navigation to correct the sun shadow device readings when the vehicle was not level. As a resulting solution, the Heading Indicator (HI) on the console, displayed LRV heading with respect to lunar north. A Bearing Indicator (BI) showed the angle to point directly towards the LM and the Distance Indicator (DI) showed distance traveled in increments of $0.1\,\mathrm{km}$. This information was coming from the navigation signal processor that calculates the north-south and east-west distances using pulse information coming from the drive module (see Fig. 2.5). All four wheels were measured to be sure a correct reading was obtained, even if one or two wheels were slipping in the lunar dust. The computer calculated the median signal from the four wheels. The speed indicator showed velocities between 0 to $20\,\mathrm{km/h}$ using only the information from the right rear wheel.

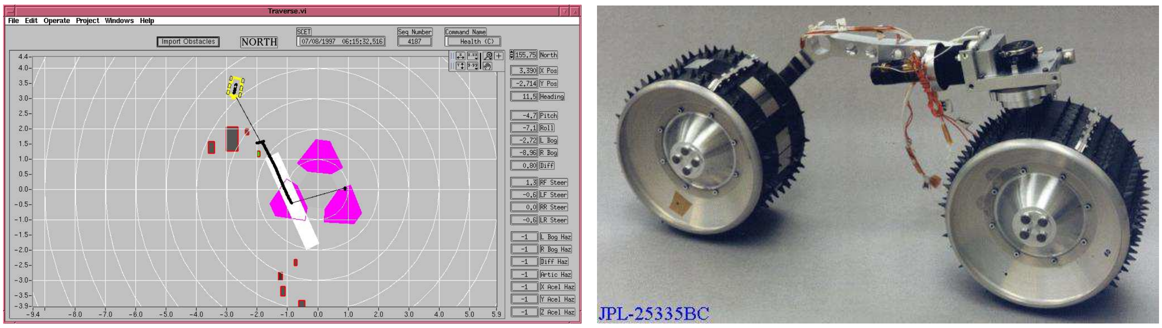
Summarizing, the major components of the navigation system were wheel odometry, one gyroscope and the sun shadow device. The navigation system was based on dead reckoning principles using speed, direction and distance from a known starting position (the LM). The computer on board computes the vehicle position using these data and solves a trigonometric problem in order to provide the relative angle and the distance to the lander. The accuracy of the system was designed to $6°$ in heading and a maximum of $600\,\mathrm{m}$ in a radius of $5\,\mathrm{km}$ from the LM. In reality the LRV navigation system was capable of displaying the vehicle position at any point in the traversal with $\pm 2\%$ accuracy. The pulses sensed on the reversed mode of the driving were considered to be forward by the navigation system, under the assumption that this introduced some error but simplified the design and minimized cost weight and complexity.

Overall the LRV navigation system was simple but sophisticated enough to provide valuable information allowing the astronauts to travel beyond sight of the LM. The exploration vehicle operated with success on the three separate lunar missions, showing that a exploration vehicle truly entails a complete navigation system.

2.2.3 Mars Pathfinder

On December 1996 the National Aeronautics and Space Administration (NASA) launched the Mars Pathfinder spacecraft, carrying a 10.5 kg microrover to explore the immediate vicinity of the lander. The wheeled robotic vehicle, named Sojourner, landed together with the Carl Sagan Memorial Station on July 1997 on Mar's Alles Vallis. The place was chosen because expected past floods channeled and deposited in this area. A tetrahedron lander of 1.5 m wide and 1.0 m high was designed, with a mass of 264 kg, including the rover, a mast with stereo cameras and other science instruments. The cameras were mounted on a pan and tilt unit on top of the mast, located 1.5 m above the ground. The cameras on board the lander were essential to perform rover navigation. They have a baseline of 0.15 m, Charge-Coupled Device (CCD) with 256 × 256 pixels, with 0.001 radian per pixel, which results on a Field of View (FOV) of approximately 14.6°, Matthies et al. (1995); Cooper and B. (1998).

The rover is 630 mm long, 480 mm wide and 300 mm high with 170 mm ground clearance. Sojourner's principal mission was to use the Alpha Proton X-ray Spectrometer (APXS) to determine the composition of rocks. A reliable localization system was required to precisely locate the rover close to the rock to analyze. A passive rocker-bogie chassis with six motorized wheels of 13 cm diameters, and four motorized steering mechanisms formed the locomotion system. As a result, Sojourner was able to traverse obstacles about 1.5 times the wheels diameter. The primary power system for the rover came from the solar panel and non-rechargeable Lithium batteries provided power through the night. The power budget had a peak of 16 W at noon, drove the performance of the navigation system and limited rover operation for about one to two hours around noon. Rover speed was 0.67 cm/s and turning maximal speed of 7 °/s. The rover communicated to the lander over a UHF antenna with a range of 500 m, and the lander directly to Earth. The rover computer on board was a 100 kHz Intel 80C85 with 0.5 MB of RAM with no operating system (100 Mill Instruction Per Second or KIPS). The computer performs I/O to some 90 sensor channels and services the motors, instruments and cameras. The software was a custom executive with a single thread written in C. Therefore, only one major function among driving, steer-



SCET 07/08/1997 06:15:32.516
Seq Number 4187
Command Name Health (C)

155.75	North
3.390	X Pos
-2.714	Y Pos
11.5	Heading
-4.7	Pitch
-7.1	Roll
-2.72	L Bog
-8.96	R Bog
0.80	Diff
1.3	RF Steer
-0.6	LF Steer
0.0	RR Steer
-0.6	LR Steer
-1	L Bog Haz
-1	R Bog Haz
-1	Diff Haz
-1	Artic Haz
-1	X Acel Haz
-1	Y Acel Haz
-1	Z Acel Haz

(a) (b)

Figure 2.6: (a) Screenshot of the navigation software for the Mars Pathfinder mission. Lander petals in a deployed configuration are at the center defining the origin of the navigation frame. (b) Detail of Sojourner front wheels and bogie. Image credit: NASA/JPL.

ing, transmitting hazard detection was able to be performed at a time. Sojourner was a semi-autonomous rover and the GNC system could be decomposed into the following:

- Guidance is divided into three main functions. Goal designation, which identifies the desirable target position. Hazard detection, performing hazard and obstacle identification. Path selection, selecting the most convenient path to reach the goal.

- Rover control performs the appropriate motor commands base on the Guidance and Navigation inputs. Basic movements consist of straight driving and turn-on-spot. The rover motion was commanded by one command at a time: Turn, Move, Go to Waypoint, Find Rock and Position APXS were the programmed commands.

- Navigation or rover localization deduces and propagates real-time rover pose.

The mission operator on Earth designed the goal for the Sojourner rover using stereo imagery from the lander. Goal designation specified 3-D waypoints using stereo cameras on the lander limited to range to $15-20\,\mathrm{m}$ and designation error correlated to the distance from the lander. Sojourner was operated on the basis of a unique local coordinate frame (Navigation or Site frame) centered on the origin of the lander. X and Y axes pointed to Martian North and East respectively forming a right-hand coordinate system. Fig. 2.6 depicts some details of the navigation and locomotion systems.

Sojourner was equipped with two front black and white CCD cameras(768×484 pixels) and one rear color CCD camera close to the APXS instrument for science imaging. The Hazard detection was performed in a minimalistic manner but with good results. Coupled with the front cameras, five infrared laser stripe projectors were available to provide an array of 20 measurement of terrain elevation (four points along each stripe), named active triangulation. On level ground the farthest point on a stripe projected about $30\,\mathrm{cm}$ and $13\,\mathrm{cm}$ centrally in front of the rover and laterally beyond each side of the rover respectively. Sojourner scanned for hazards once every wheel radius of forward motion ($6.5\,\mathrm{cm}$), having a direct impact on the effective rover velocity, slowing it down to $0.22\,\mathrm{cm/s}$ (instead of the $0.65\,\mathrm{cm/s}$ mentioned).

Path selection was a simple control behaviour using steering logic based on the obstacle detected by hazard detection. Briefly, the logic commands the robot to move forward and turn toward the designated goal if any hazard is detected during the detection phase. In case of hazard detection, sojourner turned in spot right or left, depending on the obstacle location, until no hazard was detected. The logic worked well as long as obstacle frequencies are not too high. The definition of *high* was a key issue during the navigation system design. Performance evaluations were tested using the side rock frequency distribution of Viking Lander 2 (VL2) and Pathfinder's candidate landing site. The path selection perfectly worked for terrain between 0.28 and 0.57 $obstacles/m^2$ (full success) and with difficulties at 0.85 $obstacles/m^2$ ($66\,\%$ of success). Pathfinder landing site surface was in a range between 0.65 and 2.0 $obstacle/m^2$, indicating difficulties to safely drive with the proposed methodology, Matthies et al. (1995).

The rover localization uses the dead reckoning and a daily localization update sent to Earth. Dead reckoning was performed counting wheel turns and on board sensors. The navigation sensors consisted of a heading gyroscope and three accelerometers for sensing the X, Y and Z axis motion and wheels encoders in all six wheels for odometry. Passive articulation and wheel steering angle were sensed using potentiometers. A kinematics modeling of the Sojourner chassis was computed on board, processing the chassis configuration information. Sojourner pose was propagated at $2\,\mathrm{Hz}$ by integrating the encoder counts and the gyroscope angular rate. The accelerometers continuously provided tilt and roll information of the rover platform, Mishkin et al. (1998).

Significant heading errors (> $5°$) and the associated error in lateral (y) and forward direction (x) limited the maximum distance to traverse between $5\,\mathrm{m}$ and $10\,\mathrm{m}$. Dead reckoning had a nominal error of $45\,\mathrm{cm}$ in forward and $15\,\mathrm{cm}$ in lateral direction depending on wheel slip. Sojourner traversed a total of $100\,\mathrm{m}$ with $2.7\,\mathrm{m}$ of average distance per locomotion day, having important difficulties maintaining its heading knowledge. The turn errors were greater when the vehicle makes right turns, drifting to the left while turning. In an $8\,\mathrm{m}$ run, up to $15°$ heading error was observed. The result suggests that the turn rate was miscalibrated with a dominant scale factor and its performance degraded when terrain interactions induced vibration in the vehicle, Matthies et al. (1995).

One of the technologies demonstrated was the reconstruction of the actual path, and comparison to the commanded path. The Sojourner mission gave insight for the design and operation of the Mars Exploration Rovers (MER). The mission easily covered the primary and extended mission duration of one and four weeks respectively and exceeded all the expectations and design goals. Mars Pathfinder ended after 83 sols of operations, when the lander lost communication with Earth, most likely due to lander battery failure. In comparison with previous missions, Sojourner marked an important lesson learned in what is called rover autonomy and navigation (being the first rover to navigate on the Mars surface). Some remarkable facts can be learned and are described in the following:

- Limitations on revealed heading estimation and position uncertainty, with typical errors in position and heading of 0.69 m and 8.0° respectively at the end of the trial, representing a 5 % of distance error.

- Autonomy is a key limitation on the number of science operations per day as well as on the distance to navigate.

- Hazard scanning and path selection is directly connected to the net traversal speed of the rover.

- Goal designation on board the lander limited the distances beyond 20 m due to systematics errors caused by cameras angular resolution, mast and pan and tilt angle miscalibration, as well as noisy estimation of the stereo disparity of the waypoints. Future needs for rover autonomy were already identified at this point.

- Sojourner marked the importance of having quantitative information of navigation in order to estimate robotic vehicle performance in the future.

Mars Pathfinder had planned a successor to duplicate the Sojourner rover as part of the Mars Surveyor 2001 mission. The mission was canceled in May 2000, however two rovers succeeded Sojourner, the Mars Exploration Rovers (MER). One of these rovers landed at Mars Surveyor's target site, known as the Meridiani Planum.

2.2.4 Mars Exploration Rovers

MER-A (named Spirit) and MER-B (Opportunity), marked a milestone in space exploration vehicles. Those two twin rovers represent the longest planetary exploration thus far. Spirit landed on Mars on 4 January 2004 on crater Gusev, and its twin three weeks after on the Meridiani Planum. The mission objectives of the Mars Exploration Rovers were to characterize rocks looking for clues of past water on Mars surface. Interesting samples to analyze were minerals deposited by water-related processes. Those objectives drove rover autonomy requirements and mobility capabilities to be able to navigate a considerable minimum distance per Mars day or Sol in complex terrain. The propagation of the rover's pose within a small percentage error of traversal was also necessary. This required a level of intelligence on board, establishing a new benchmark and state-of-the-art on the field. At the time of writing this document, the mission of the rover Opportunity continues with more than 44.66 km traveled. The mission of the rover Spirit was redefined as a stationary research platform after several months attempting to free the rover, since the rover had only four operational wheels. Afterwards, the last communication with the rover was on 22 March 2010 and on 25 May 25 2011 NASA announced the completion of the mission. Spirit did not survive the Martian winter, having excessively cold internal temperatures due to energy inefficiency running the heaters. Independently of the coming news regarding Opportunity, both rovers successfully completed the primary mission which was defined per 90 sols, Maimone et al. (2007b).

In brief, both vehicles are six-wheeled and solar powered rovers, as their predecessor Sojourner. They have six aluminum wheels with a rocker-bogie system in which four of them, front and back wheels, have steering abilities. Therefore, Ackermann steering and turn-on-spot maneuvers define the turning capabilities. Rovers are 1.5 m in height including the camera mast, 2.3 m wide and 1.6 m long. Rover's main body structure called Warm Electronics Box weighs 145.6 kg, and the mass of the rover mobility system is 34.5 kg, making the total mass of the whole rover approximately 180.1 kg. The rover has a ground clearance of 29 cm on a flat surface with 25 cm in diameter per wheel. The mobility system is theoretically capable of negotiating hazards over 35 cm tall. However, due to ground clearance, a hazard is defined an obstacles whose nominal value is 20 cm. Both rovers are statically stable at 45° and capable of traversing slopes as high as 30°, Maimone et al. (2006).

The rovers' sensing ability includes wheel encoders for dead reckoning, potentiometers for the joints of the chassis, inertial attitude sensing and absolute heading determination using cameras and gyroscopes. The Inertial Measurement Unit (IMU), the LN-200 Northrop-Grumman Corporation (2000), is equipped with three accelerometers and three fiber optic gyroscopes. The rovers propagate pose estimation at 8 Hz while driving Maimone et al. (2007a). The position is first estimated using wheel odometry from the encoders. Rover attitude is computed using the accelerometers (during static regimen) and the integration of the gyroscopes (during traversal). High accuracy and low drift have been experienced (less than 2 °/h) in heading. It has entailed a significant improvement in performance with respect

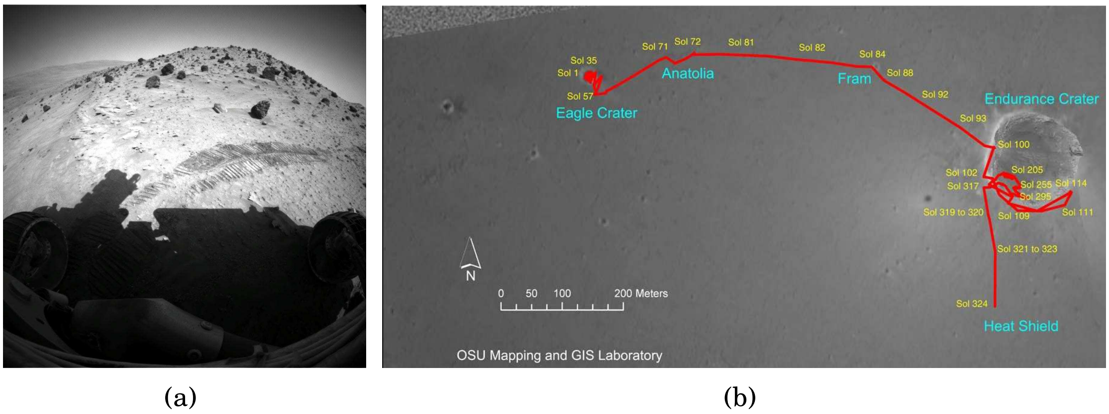


Figure 2.7: (a) Spirit, MER-A's hazard camera with a 120° FOV (b) Opportunity, MER-B's traversal with almost $2\,\mathrm{km}$ for six months of exploration inside Endurance Crater. December 2004. Image credit: NASA/JPL/MSSS/OSU

to their predecessor Sojourner. Heading drift correction is performed using the cameras by looking at the sun once per sol, using the Panoramic cameras (Pancams). Several cameras are on board in order to allow high rover autonomy and sensing for navigation. Two navigation and two panoramic cameras are mounted on a pan and tilt unit on top of the mast at a height of $1.3\,\mathrm{m}$ above the ground. The Navigation cameras (Navcams) have a wider Field of View (FOV) than the Panoramic cameras, 45° FOV and 16° FOV respectively. The reason is that Navcams are used to perform path planning in a short range during traversal, while the Pancams are used for scanning the horizon and resolving far-field terrain in long plan, in which higher resolution is desirable. In addition, two pairs of Hazard detection cameras (Hazcams see Fig. 2.7) are mounted on the rover body closer to the ground, one at the front and one at the back in order to perform obstacle detection and avoidance.

As Sojourner, the rovers have semi-autonomous navigation capabilities being commanded from ground. The GNC system can be decomposed in the following:

- Guidance, divided in two main functions, terrain assessment and path selection.
- Navigation, estimates the rover position and orientation in a full $6\,\mathrm{DoF}$ using the sensing capabilities previously mentioned. Moreover, visual odometry is remotely triggered when expecting to navigate challenging terrains.
- Control. The rover is commanded using different control modes. The low-level control commands the steering and driving actuators. The rover is capable of driving on straight lines, Ackermann and turn-on-spot. Autonomous commands, in which the inputs come from the guidance, are dictated by the navigation, closing the loop in an autonomous manner.

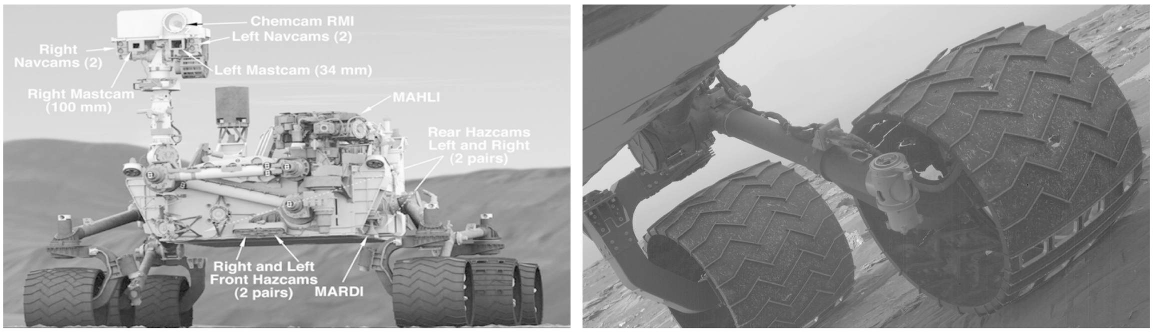
Terrain assessment gives the rovers the capability to estimate the conditions of the nearby terrain in order to evaluate a priori the difficulties of the traversal. Terrain assessment is based on the GESTALT system, Grid-based Estimation of Surface Traversability Applied to Local Terrain (GESTALT), described in Howard and Seraji (2001). The system uses stereo-generated 3D geometric data and evaluates the roughness of the terrain on the grid map. Terrain assessment is beneficial not only during long drives but also in approaching science targets, Biesiadecki and Maimone (2006).

2.2.5 Mars Science Laboratory

Mars Science Laboratory (MSL) or Curiosity rover is nowadays the last launched operational rover on Mars. The rover was launched on November 2011 and successfully landed on Mars on 6 August 2012 with a primary mission time of one Martian year. The rover design inherits much of mobility technologies from the Mars Exploration Rovers, having six-wheel drive, a rocker-bogie suspension, robotic arm and autonomous navigation abilities. The locomotion system is essentially the same but with larger size. Curiosity is twice as big as the MER, weighs 900 kg, has a 50 cm wheel diameter, and is 3 m long and 2.7 m wide allowing the carriage of more scientific instruments on board than its predecessors, Arvidson et al. (2016).

Curiosity has seventeen cameras overall, twelve engineering cameras (Hazcams and Navcams) and five science cameras (see Fig. 2.8a). Similarly to MER, the Hazcams pairs are located on each of the four corners on the body and the Navcams are mounted on the mast at 2.2 m from ground. The Navcams allow the rover to see further with a 42 cm stereo baseline, which produces almost a 100 m disparity map instead of the 25 m for the MER. Curiosity shares with the MER its IMU and orientation estimation software, the hazard detection system and most of the autonomous software. The main update is the visual odometry software which improves robustness, makes multiresolution tracking and reduces the computational time from $2-3$ min in MER to 40 s mainly because the processor on board is 10 times faster than the MER's processor.

Similar to the MER localization capabilities, Curiosity primarily relies its pose on wheel odometry and attitude based on the IMU, triggering visual odometry only when necessary. As of today, Curiosity uses visual odometry with more frequency than in previous missions thanks to the improvements in computational time. Curiosity used visual odometry during 34of the first 40 sols, predicting slip checks every 20 m and autonomously deciding whether to use visual odometry. This technique allows visual odometry to be used when necessary and partially mitigates its impact in operations since the decision is made by the rover. Another navigation improvement on Curiosity



(a) (b)

Figure 2.8: (a) Camera location on the Curiosity rover. (b) Picture of curiosity's left middle wheel with the latest sign of wear and tear. Image credit: NASA/JPL.

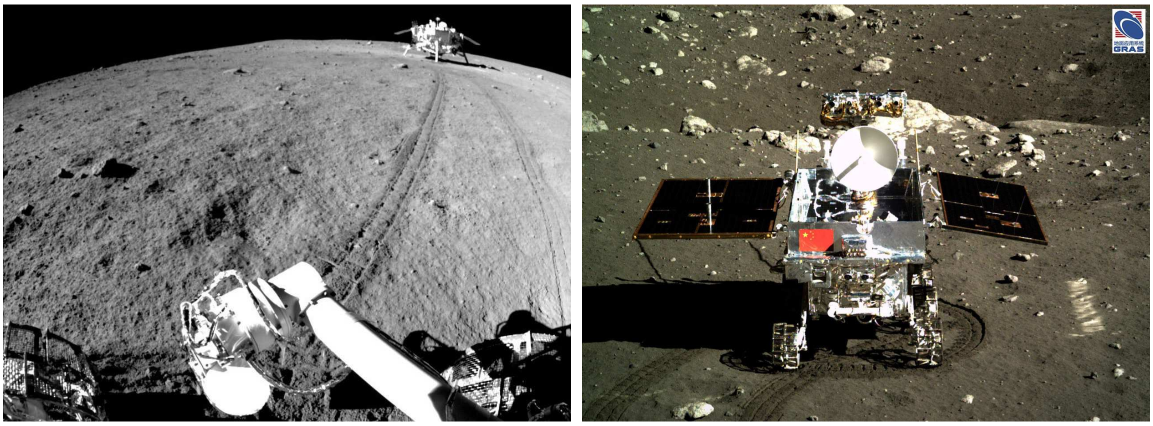
is the geometric hazards detection. The rover is able to detect terrain hazards up to 7 m away from its location which makes the path more effective through the terrain.

On 20 December 2013, NASA reported that Curiosity had successfully upgraded, for the third time since landing, its software programs and is now operating with a new version. The new software is expected to provide the rover with better robotic arm and autonomous driving abilities. The rover got its first extended mission on October 2014 for two years. The rover experiences a significant wear on its wheels (see Fig. 2.8b) which is affecting operations and path selection in order to avoid rougher terrain. Nevertheless, the mission has fulfilled all of its diverse science objectives and primary mission requirements, providing a number of additional discoveries of significance. As of today, MSL progress towards mountain Sharp with an overall traveled distance of 17.32 km on sol 1800 , NASA (2017).

2.2.6 Yutu Rover

China National Space Administration(CNSA) has carried a lunar exploration program since 2002. The Yutu lunar rover landed on 14 December 2013 and was the first soft landing on the Moon since the Lunokhod rovers. The mission objectives are to explore Moon surface materials and resources in order to analyze the physical characteristics of the lunar soil. The rover is an autonomous vehicle of 1.5 m length, 1.0 m width and 1.1 m height, with a mass of 136 kg. The rover has six wheels, a rocker suspension and is equipped with a 3 DoF robotic arm to accomplish scientific samples. Yutu has 3D mapping and path planning abilities along the traversal, Laxman (2012).

The rover completed a first examination of the lunar soil and suffered a circuit malfunction at the end of the task. On 12 February 2014, during the third lunar day, Yutu rover was declared inoperative due to a mechanical control malfunction. The



(a) (b)

Figure 2.9: (a) Hi-res image taken by China's Yutu lunar rover, showing the lander and the surroundings of Mare Imbrium. (b) Picture of Yutu rover executing a point turn maneuver, taken from the lander. Image credit: Chinese Academy of Sciences/ China National Space Administration/ The Science and Application Center for Moon and Deepspace Exploration

rover was unable to move but the scientific instruments were working normally, Howell (2014). The rover was unable to fully charge its batteries and with each lunar night the rover was less functional, Xinhua (2014). Though the scientific instruments were functional, their usability was limited to making the same observations, due to immobile conditions. CNSA mission control kept the rover transmitting data until December 2015, and on 3 August 2016 the mission finished all operations, Aron (2016).

Part I

Dead Reckoning

Chapter 3: Attitude and Heading Reference System Design

Hidalgo-Carrio, J., Arnold, S., and Poulakis, P. (2016). On the Design of Attitude and Heading Reference Systems Using the Allan Variance. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency control*, 63(4):656–65

Hidalgo-Carrio, J., Poulakis, P., Köhler, J., Del-Cerro, J., and Barrientos, A. (2012a). Improving Planetary Rover Attitude Estimation via MEMS Sensor Characterization. *Sensors*, 12(2):2219–2235

Hidalgo-Carrio, J., Poulakis, P., Barrientos, A., and Del-cerro, J. (2011). ESTEC Testbed Capabilities for the Performance Characterization of Planetary Rover Localization Sensors - First Results on IMU Investigations. In *ASTRA 2011 - 11th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, pages 1–8. ESA/ESTEC

Chapter 4: Enhanced 3D Odometry

Hidalgo-Carrio, J., Babu, A., and Kirchner, F. (2014). Static forces weighted Jacobian motion models for improved Odometry. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 169–175. IEEE

Hidalgo-Carrio, J. (2013). Navigation and Slip Kinematics for High Performance Motion Models. In *Symposium on Advanced Space Technologies in Robotics and Automation*

Hidalgo-Carrio, J. and Cordes, F. (2012). Kinematics Modeling of a Hybrid Wheeled-Leg Planetary Rover. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*

Azkarate, M., Zwick, M., Hidalgo-Carrio, J., Nelen, R., Wiese, T., Poulakis, P., Joudrier, L., and Visentin, G. (2015). First Experimental Investigations on Wheel-Walking for Improving Triple-Bogie Rover Locomotion Performances. In *Symposium on Advanced Space Technologies for Robotics and Automation*

Chapter 3

Attitude and Heading Reference System Design

The estimation of a vehicle's attitude is an important task for navigation since it describes the direction of motion. Robots are becoming increasingly more competent at performing autonomous tasks in places and situations where human intervention was the only alternative. Tasks such as autonomous docking, localization in unstructured environments and instrument deployment, to name a few, call for a high-performance Attitude and Heading Reference System (AHRS). Planetary exploration and underwater robots rely solely on onboard localization means due to the absence of global localization in space, Backes et al. (2005); Maimone et al. (2006, 2007b); Ali et al. (2005); Hidalgo-Carrio et al. (2012b, 2014) and underwater Troni and Whitcomb (2013); Hildebrandt et al. (2014), and as such require a particularly good understanding of sensor error sources. For that reason space and underwater robotics share common interests in terms of AHRS.

Inertial sensors are based on different operating principles and fabrication methods, which influence their error characteristics and the dominance of each error source in the final measurement. Traditionally, optical principles of operation senses changes in orientation using the Sagnac effect using an optical fiber. Recent advancements in micro-fabrication and in microsystems technology in general, have led to high quality Micro Electro-Mechanical Systems (MEMS) with improved performances, something that has made them particularly attractive for aerospace and underwater systems. While the use of traditional optical gyroscopes are dominant in both sectors. The use of MEMS-based inertial sensors has made Inertial Navigation Systems (INS) more affordable, with smaller footprints and modest power consumption figures, something that has spawned a new generation of robust, small-scale sensors with great potential for robotic applications, Rehrmann et al. (2011); Schwendner and Hidalgo-Carrio (2012).

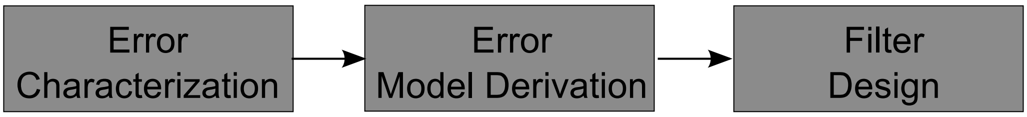
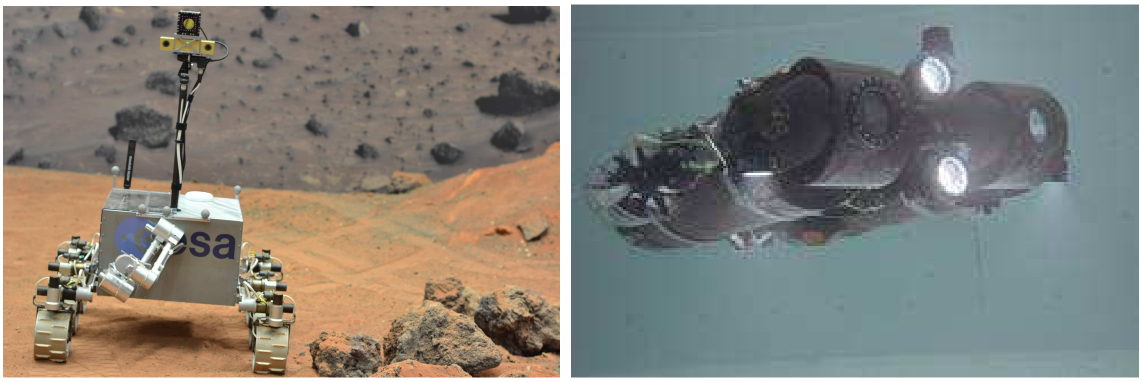


Figure 3.1: Schematic representation of the End to End (E2E) approach for AHRS design. Figure adapted from Hidalgo-Carrio et al. (2012b).

Different techniques can be applied to identify sensor error sources and derive a model which then can be incorporated in the filter. The general concept behind these methods is to analyze the sensor signal in order to characterize the error source. The most popular among them is the Fast Fourier Transform (FFT) of the noise signal, which identifies and distinguishes errors at different frequencies of the spectrum. Another approach is to employ the autocorrelation function to find repetitive patterns in the time-domain of the signal and subsequently using the FFT to derive the Power Spectral Density (PSD). The Fourier transform of the autocorrelation function allows the identification of stochastic processes. When investigating the stability of atomic clocks, David Allan proposed a methodology to separate noise forms of time series measurements, an approach that did not require the conversion of the signal to the frequency domain, Allan (1966). Later this method was adopted by the inertial sensor community due to its simplicity and its capability to identify the long-term noise. In essence, the Allan variance provides direct information of the underlying stochastic processes and has gained momentum due to its computational simplicity and ease of adaptation to a variety of noise types, Allan (1987). Though there are certain limitations in the mapping from the Allan variance to the frequency spectrum, still the method remains powerful and has become an IEEE Standard Specification and Test Procedure for inertial sensors, IEEE (2008).

INS in general, and AHRS in particular, are widely used in aerial, terrestrial and underwater applications, and various techniques for characterizing inertial sensor errors are available in literature as Xiyuan (2003); Aggarwal et al. (2008). N.El-Sheimy et al. (2008) present a thorough methodology for using the Allan variance for the characterization of the noise coefficients of inertial sensors. Their work establishes a relationship between the Allan variance and the noise Power Spectral Density (PSD). Several approaches have also been proposed for modeling inertial sensors and their error behaviour. Xing and Gebre-Egziabher (2008) describe the sensor model for low-cost inertial sensors and in Zhang et al. (2008) the Allan variance is applied to provide a sensor error characterization in detail. Merwe et al. (2004); Han and Knight (2009); Suh (2010); Quinchia et al. (2013b) show that incorporating inertial data in a correct manner in a sensor fusion filter is a challenging process.

The Allan variance Allan (1966) was originally developed to characterize the stability of atomic clocks and has also been successfully applied for the characterization



(a) (b)

Figure 3.2: Two robotic platforms used to validate the AHRS design. (a) ExoMars Test Rover (ExoTeR) at the ESA Planetary Robotics Laboratory. (b) Underwater vehicle Dagon at the DFKI Maritime Exploration Hall.

of inertial sensors. Inertial navigation systems can provide accurate results over short time but might rapidly degrades with time. During the last decade performance of inertial sensors has significantly improved, particularly in terms of signal stability, mechanical robustness and power consumption. Mass and size of inertial sensors has also been significantly reduced, offering system level design and accommodation advantages. This chapter presents a complete methodology for the characterization and modeling of inertial sensors using the Allan variance, with direct application in navigation systems. Although the concept of fusion is relatively straightforward, accurate characterization and sensor information filtering is not a trivial task, and they are essential for good performance. A complete and reproducible methodology utilizing the Allan variance, including all the intermediate steps is described in this chapter.

A schematic overview of this E2E methodology is shown in Fig.3.1. The steps of utilizing an inertial sensor for attitude estimation are addressed: starting from the characterization of the dominant sensor errors, then deriving a suitable sensor error model and finally designing an adequate filter. Two different inertial sensors are used to validate the approach, sensors based on Microelectromechanical systems (MEMS) and a classical optics-based inertial sensor. MEMS are mounted on a planetary rover (Fig. 3.2a) and the optics-based gyroscopes on an underwater vehicle (Fig. 3.2b). The reason is because the optics-based sensor oversizes the allocated space in the planetary rover and requires extra power. The methodology equally applies to both vehicles. It is also important to test the approach with both IMU technologies since both are currently used in modern spacecrafts.

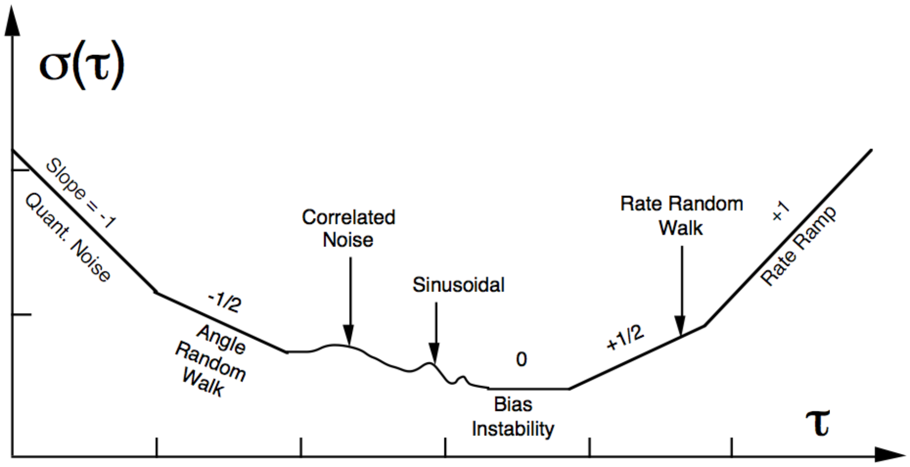


Figure 3.3: Typical Allan variance plot for data analysis. The cluster time of length τ could take different units in time (e.g. microseconds, seconds, minutes or hours) and the standard deviation $\sigma(\tau)$, (e.g. angular velocity rad/s or acceleration $^\circ/h$, g or m/s^2) depending on the sensor type. Figure adapted from IEEE (2008).

3.1 Background

The Allan variance $\sigma^2(\tau)$ extracts information on the form and magnitude of distinct noise terms. First, the method separates static measurement values into clusters. Second, a statistical variance is computed among the clusters of equal dimension. The variance of a cluster time of length τ, is estimated by Allan (1966)

$$\sigma^2(\tau) = \frac{1}{2(N-2n)} \sum_{k=1}^{N-2n} [\bar{\Omega}_{k+1}(t) - \bar{\Omega}_k(t)]^2 \tag{3.1}$$

where N is the complete amount of samples and $\bar{\Omega}(t)$ is the cluster average value of the output for a cluster which starts at the k^{th} data point and contains n samples depending on the length of τ. The Allan standard deviation is plotted in a logarithmic scale (log-log plot) versus the cluster time τ where different noise terms might show in the plot (see Fig. 3.3). The identification of the Power Spectral Density (PSD) function of the stochastic process is required to model the stochastic error. This is done by the unique relationship between the Allan variance (time domain) and the PSD (frequency domain). Such relationship $S_\Omega(f)$ is defined by

$$\sigma^2(\tau) = 4\int_0^\infty S_\Omega(f)\frac{\sin^4(\pi f\tau)}{(\pi f\tau)^2}df. \tag{3.2}$$

The different stochastic processes are identified at various frequencies by simply varying the cluster time τ. Further explanation of typical errors present in inertial sensors is available at IEEE (2008). It is possible to relate the Allan variance and the PSD of a stochastic error using this approach. It is important to note that the Allan variance does not always determine a unique noise spectrum since the mapping from the spectrum to the Allan variance is not bijective (one-to-one). This mapping is given by equation (3.2) and there is no inversion formula. Table 3.1 summarizes these derived relationships for the most dominant inertial sensor errors. The quality of the estimation of the variance is proportional to the number of computed independent clusters. Consequently, the confidence of the estimation improves as the number of cluster increases. It is computationally efficient to calculate the Allan variance at different cluster terms using n as a power of 2.

Modeling sensor errors benefits the overall AHRS performance. A model can be accurate and specific to a particular sensor but lose general applicability. The proposed sensor model is a balance between accuracy and generality and consists of both deterministic and stochastic sensor errors. The sensor model encompasses deterministic errors like the scale factor and the misalignment, while others, like the thermal drift, are assumed to be directly compensated by the transducer. The model does not include scale factor asymmetry, g-sensitive bias, or other errors which mostly appear on a specific sensor and are usually insignificant for most applications Titterton and Weston (2004); Rogers (2003). The sensor model equation is

$$\tilde{\boldsymbol{\omega}}(t) = (I_{3\times3} + M(c,m)) \cdot \boldsymbol{\omega}(t) + \boldsymbol{n}_s(t) \tag{3.3}$$

where $\tilde{\boldsymbol{\omega}}(t)$ is the 3×1 continuous time sensor measurement (i.e: $[p,q,r]^\intercal$ angular velocities from gyroscopes), M is the deterministic error matrix corresponding to the misalignment m and scale factor c and $\boldsymbol{n}_s(t)$ is the stochastic noise process.

The derivation of a stochastic error model requires the identification of the PSD function for which the Allan variance is useful. The model is a linear and time invariant system which uses an input unit white noise to shape itself to the desired output by knowing the PSD function of the stochastic process. This technique is known as the shaping filter approach from Brown and Hwang (2012). Different noise

Table 3.1: Dominant stochastic errors in inertial sensors.

Noise type	Allan variance $\sigma^2(\tau)$	PSD	Coef.
Quantization	$\frac{3Q^2}{\tau^2}$	$(2\pi f)^2 Q^2 \tau$	Q [°]
Random walk	$\frac{N^2}{\tau}$	N^2	N $[°/\sqrt{h}]$
Bias instability	$\approx \frac{2B^2 ln2}{\pi}$	$(\frac{B^2}{2\pi})\frac{1}{f}$	B $[°/h]$
Rate rand.walk	$\frac{K^2\tau}{3}$	$(\frac{K}{2\pi})^2\frac{1}{f^2}$	K $[°/h/\sqrt{h}]$
Rate ramp	$\frac{R^2\tau^2}{2}$	$\frac{R^2}{(2\pi f)^3}$	R $[°/h^2]$

Table 3.2: Accelerometers noise coefficients for the FOG-based IMU.

Coefficient	Acc X	Acc Y	Acc Z	Datasheet
N $[m/s/\sqrt{h}]$	0.0618	0.0802	0.0583	0.0705
K $[m/s^2/\sqrt{h}]$	0.00147	0.00083	0.00073	n/a
B $[m/s^2]$	0.00057	0.00058	0.00053	0.00049

types can be modeled using the Table 3.1, and the spectral function is unique for each stochastic noise type. However, there are certain limiting factors in the derivation of the shaping filter linear equations. For instance, quantization noise cannot be modeled since Kalman filter theory only performs on differential equations driven by white noise, and it will have a noise source which is the derivative of the white noise. In this particular limitation a different approach could be taken using acceptable approximations, like the one presented in Savage (2002). However, it is out of the scope of this chapter to further analyze such limitations since most of the stochastic errors appearing in inertial sensors can be directly incorporated in a Kalman filter.

3.2 Sensor Error Characterization

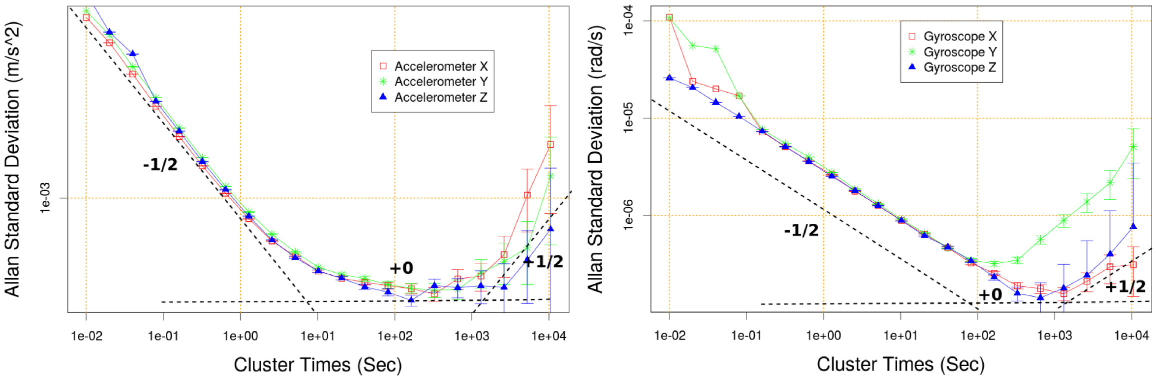
Two different Inertial Measurement Units (IMU) have been characterized using the Allan variance: a miniaturized IMU based on Double-Ended Tuning Fork (DETF) MEMS gyroscopes and a small Fiber Optic Gyro (FOG) based IMU. Both IMUs are equipped with MEMS accelerometers, temperature sensors and the digital electronics on board to acquire sensory data using a serial RS-422 connection.

The Allan variance analysis [1] for the FOG-based IMU is depicted in Fig. 3.4. The plot shows the angle random walk for gyroscopes and velocity random walk for accelerometers as the dominant error for the short cluster times. The curve fits a straight line of slope $-\frac{1}{2}$. The angle random walk coefficient N is obtained by reading the slope line at $\tau = 1$. The coefficient value is solved according to equations in Table 3.1. A straight line of slope $+\frac{1}{2}$ fits the part of the plot for long cluster times.

[1] The allanvar package for R is used: `http://cran.r-project.org/package=allanvar`

Table 3.3: Gyroscopes noise coefficients for the FOG-based IMU.

Coefficient	Gyro X	Gyro Y	Gyro Z	Datasheet
N $[°/\sqrt{h}]$	0.01055	0.01140	0.01032	0.012
K $[°/h/\sqrt{h}]$	0.0881	0.6446	0.1193	n/a
B $[°/h]$	0.0589	0.1078	0.0492	$0.1 - 0.05$



(a) (b)

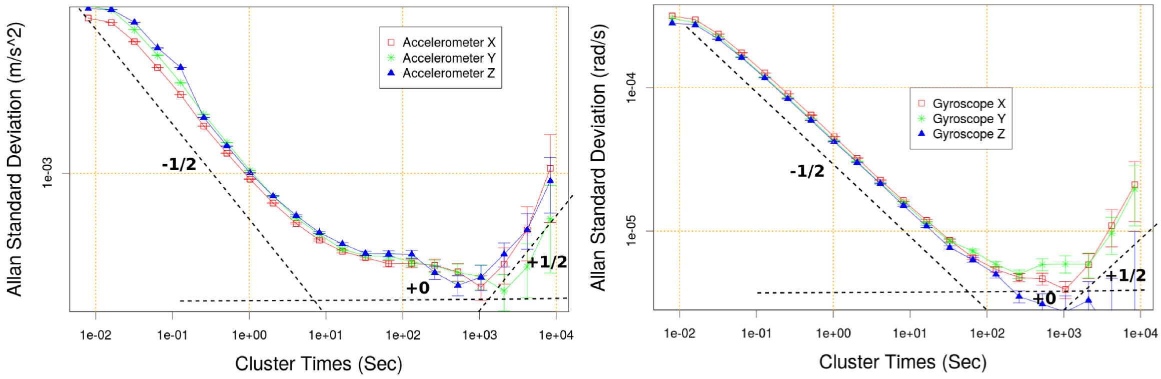
Figure 3.4: Allan variance analysis for the FOG-based IMU. (a) Allan variance plot for accelerometers. (b) Allan variance plot for gyroscopes.

Table 3.4: Accelerometers noise coefficients for the DETF-based IMU.

Coefficient	Acc X	Acc Y	Acc Z	Datasheet
$N\ [m/s/\sqrt{h}]$	0.0570	0.0628	0.0612	0.070
$K\ [m/s^2/\sqrt{h}]$	0.00085	0.00057	0.00086	n/a
$B\ [m/s^2]$	0.00055	0.00063	0.00062	0.00050

The noise coefficient K of the rate random walk is read off the slope line at $\tau = 3$. The corresponding equation from Table 3.1 is also solved. The center of the curve depicts a small flat part with a zero slope. It characterizes a bias instability noise representing the best stability of the run. The conventional unit for the bias instability coefficient B is $^\circ/\mathrm{h}$ for gyroscopes and $\mathrm{m/s^2}$ for accelerometers, IEEE (2001). Table 3.2 and 3.3 summarize the characterized Allan variance noise coefficient of the FOG-based IMU. Further information about categorizing inertial sensors based on noise coefficients can be found in Titterton and Weston (2004); Rogers (2003).

Analysis for the DETF-based IMU was also performed and the result is depicted in Fig. 3.5. The graph shows identical dominant error results for both IMUs. Table 3.4 and 3.5 summarize the characterized noise coefficient for the DETF-based IMU. From the obtained result, it is possible to observe that nowadays FOG-gyros technology performs better than DETF-gyros but with a significant penalty on weight. FOG-based IMU weighs approximately 700 g in comparison to the 55 g of the DETF-based IMU.



(a) (b)

Figure 3.5: Allan variance analysis for the DETF-based IMU. (a) Allan variance plot for accelerometers. (b) Allan variance plot for gyroscopes.

3.3 Error Model Derivation

The Allan variance characterization denoted the angle random walk and the angle rate random walk as the dominant stochastic errors for the gyros of both IMUs. Similarly, the velocity random walk and the velocity rate random walk emerged as the dominant errors for the accelerometers. It is worth noting that random walk is affected by white noise while the sensor bias or drift is coupled with the rate random walk and the bias instability.

The PSD of the random walk is defined by N^2 and the transform function can be directly incorporated into the proposed sensor model. The PSD for the bias instability B is not a function of ω^2 (see Table 3.1). The transfer function cannot be directly obtained by the shaping filter approach and a first order Gauss-Markov process is normally used to model the bias instability Brown and Hwang (2012). The PSD of the angular rate (gyroscope) or acceleration (accelerometer) random walk is defined by K^2/s^2 and is incorporated in the model by K/s, replacing $j\omega$ by s in the Laplace domain. Consequently, the derived inertial sensor model is

Table 3.5: Gyroscopes noise coefficients for the DETF-based IMU

Coefficient	Gyro X	Gyro Y	Gyro Z	Datasheet
N $[°/\sqrt{h}]$	0.1600419	0.1507285	0.1477636	0.15
K $[°/h/\sqrt{h}]$	3.6299	3.2171	1.5450	n/a
B $[°/h]$	1.478	1.557	1.553	0.5

$$\tilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) + M(c,m) + \boldsymbol{b}_g(t) + \boldsymbol{n}_{N_g}(t) \tag{3.4a}$$

$$\tilde{\boldsymbol{a}}(t) = C(\boldsymbol{q}(t))\boldsymbol{g} + \boldsymbol{a}(t) + M(c,m) + \boldsymbol{b}_a(t) + \boldsymbol{n}_{N_a}(t) \tag{3.4b}$$

$$\boldsymbol{b}_\alpha(t) = \boldsymbol{b}_{B_\alpha}(t) + \boldsymbol{b}_{K_\alpha}(t);\ \ \alpha = a, g \tag{3.4c}$$

$$\dot{\boldsymbol{b}}_{B_\alpha}(t) = -\frac{1}{\tau_{B_\alpha}}\boldsymbol{b}_{B_\alpha}(t) + \boldsymbol{n}_{B_\alpha}(t);\ \dot{\boldsymbol{b}}_{K_\alpha}(t) = \boldsymbol{n}_{K_\alpha}(t) \tag{3.4d}$$

where $\tilde{\boldsymbol{\omega}}(t)$ is the gyroscope readout, $\boldsymbol{\omega}(t)$ is the angular velocity, $\tilde{\boldsymbol{a}}(t)$ is the accelerometers readout, $C(\boldsymbol{q}(t))$ is the rotation matrix associated to the attitude quaternion $\boldsymbol{q}(t)$, $\boldsymbol{g}$ is the local gravitational vector, $\boldsymbol{a}(t)$ is the acceleration, $\boldsymbol{b}_\alpha(t)$ is the sensor bias (i.e. for gyroscopes or accelerometers), τ_{B_α} is the correlation time for the Gauss-Markov process and $\boldsymbol{n}_{N_\alpha}(t)$, $\boldsymbol{n}_{B_\alpha}(t)$ and $\boldsymbol{n}_{K_\alpha}(t)$ are independent zero-mean Gaussian noises of an inertial sensor α, meaning g for gyroscopes or a for accelerometers. These noises are defined by

$$E\{\boldsymbol{n}_{N_\alpha}(t)\boldsymbol{n}^\intercal_{N_\alpha}(\iota)\} = I_{3\times3}\sigma^2_{N_\alpha}\delta(t-\iota) \tag{3.5a}$$

$$E\{\boldsymbol{n}_{B_\alpha}(t)\boldsymbol{n}^\intercal_{B_\alpha}(\iota)\} = I_{3\times3}\sigma^2_{B_\alpha}\delta(t-\iota) \tag{3.5b}$$

$$E\{\boldsymbol{n}_{K_\alpha}(t)\boldsymbol{n}^\intercal_{K_\alpha}(\iota)\} = I_{3\times3}\sigma^2_{K_\alpha}\delta(t-\iota) \tag{3.5c}$$

where E denotes expectation, $\delta(t-\iota)$ is the Dirac delta function, σ_{N_α} is the uncertainty associated to the random walk coefficient N, σ_{B_α} is the uncertainty associated to the bias instability coefficient B and σ_{K_α} is the uncertainty associated to the rate random walk coefficient K. The aforementioned uncertainties are actually the standard deviations σ of the associated probability density functions (pdf) and depend on the signal bandwidth Δ_{bn} and the noise coefficients of Tables 3.2-3.5.

3.4 Filter Design

Fusing inertial sensors to estimate a vehicle's attitude is a common practice in navigation systems. State-of-the-art sensor fusion strategies are commonly based on, as described in Chapter 2, the Bayesian approach. The algorithms are, in fact, statistical techniques that offer state estimation by modeling static and dynamic uncertainties. Setting the right values for those uncertainties is important and the stochastic models derived with the Allan variance analysis serve in the filter design. This work employs the Indirect Kalman Filter (IKF), Lefferts (1982); Park (1996), due to its fast response and the fact that the attitude kinematic equations are independent of the filter state. The advantage of IKF is that it has a smaller state vector

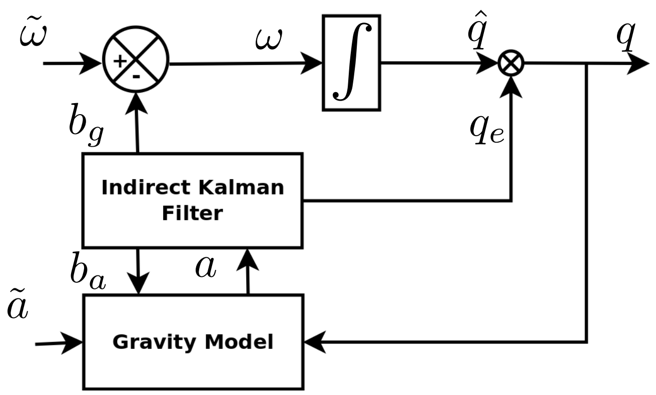


Figure 3.6: Schematic representation of the AHRS using the IKF approach

dimension than the conventional Kalman filter (see Fig. 3.6).

The filter estimates the vehicle's attitude using the error quaternion proposed by Lefferts (1982). The attitude is defined by first applying the current estimated attitude $\hat{\boldsymbol{q}}$ and then is corrected by a small error rotation $\boldsymbol{q}_e$ expressed in the vehicle's body frame. The attitude is computed using the composition (product) of two quaternions $\otimes$ while the state vector estimates the small attitude error so that

$$\boldsymbol{q} = \hat{\boldsymbol{q}} \otimes \boldsymbol{q}_e \tag{3.6}$$

$\boldsymbol{q}_e$ only depends on the sensor error model and is a three element vector assuming that the scalar part is $1 \approx cos(\theta/2)$ for small θ rotations. The error quaternion dynamics are estimated in the filter and the prediction step is determined by the following equation from Creamer (1996)

$$\dot{\boldsymbol{q}}_e(t) \approx -\tilde{\boldsymbol{\omega}}(t) \times \boldsymbol{q}_e(t) - \frac{1}{2}(\boldsymbol{b}_g(t) + \boldsymbol{n}_{N_g}(t)). \tag{3.7}$$

The filter state is defined by $\boldsymbol{x} = [\boldsymbol{q}_e \; \boldsymbol{b}_{B_g} \; \boldsymbol{b}_{K_g} \; \boldsymbol{b}_{K_a}]^\intercal$ which contains the bias of gyroscopes and accelerometers. The following prediction equation is used to propagate the state

$$\dot{\boldsymbol{x}}(\boldsymbol{t}) = A\boldsymbol{x}(t) + B \tag{3.8}$$

with

$$A = \begin{bmatrix} -[\tilde{\boldsymbol{\omega}}\times] & -0.5I_{3\times3} & -0.5I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -\beta I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{3.9a}$$

$$B = \begin{bmatrix} -0.5\boldsymbol{n}_{N_g} \\ \boldsymbol{n}_{B_g} \\ \boldsymbol{n}_{K_g} \\ \boldsymbol{n}_{K_a} \end{bmatrix} \tag{3.9b}$$

where $[\boldsymbol{v}\times]$ represents the cross product matrix of the vector $\boldsymbol{v}$ and β is the inverse of τ_{B_g}, the correlation or cluster time associated to the bias instability. The equations rely on the bias instability and rate random walk to estimate the gyroscope bias with $\boldsymbol{b}_{B_g}$ and $\boldsymbol{b}_{K_g}$ respectively. The accelerometer bias is simply modeled as acceleration random walk $\boldsymbol{b}_{K_a}$. The process noise $\boldsymbol{n}_{N_g}$, $\boldsymbol{n}_{B_g}$, $\boldsymbol{n}_{K_g}$ and $\boldsymbol{n}_{K_a}$ are set using the noise coefficient identified by the Allan variance analysis. The filter process covariance is

$$Q = E\{BB^{\mathsf{T}}\} \triangleq diag(0.25Q_{N_g}, Q_{B_g}, Q_{K_g}, Q_{K_a}) \tag{3.10}$$

where Q is a variance matrix with block matrices as diagonal entries. The standard deviation σ_{N_g} in Q_{N_g} depends on the random walk noise coefficient and the signal bandwidth $f_{bn} = \Delta_{bn}^{-1}$ defined as

$$Q_{N_g} = I_{3\times3}\sigma_{N_g}^2\delta(t-\iota) \tag{3.11a}$$

$$\sigma_{N_g} = N/\sqrt{\Delta_{bn}}. \tag{3.11b}$$

The noise covariance of the Gauss-Markov process for the gyroscopes bias instability model is obtained from the following equation

$$Q_{B_g} = I_{3\times3}\sigma_{B_g}^2\delta(t-\iota) \tag{3.12a}$$

$$\sigma_{B_g} = B\sqrt{2\beta}. \tag{3.12b}$$

The angle rate random walk and acceleration random walk are computed by

$$Q_{K_\alpha} = I_{3\times 3}\sigma^2_{K_\alpha}\delta(t-\iota) \tag{3.13a}$$

$$\sigma_{K_\alpha} = K\sqrt{\Delta_{bn}}. \tag{3.13b}$$

A zero-order hold assumption is selected for discretization where the inertial sensor output is sampled periodically with period Δt and is assumed constant over the time interval. The discrete prediction equations propagate the discrete state $\boldsymbol{x}_k$ using

$$\hat{\boldsymbol{x}}^-_{k+1} = \phi_k\hat{\boldsymbol{x}}_{k+1} \tag{3.14a}$$

$$P^-_{k+1} = \phi_k P_{k+1}\phi_k^\intercal + \hat{Q} \tag{3.14b}$$

where $\phi \triangleq \mathrm{e}^{A\Delta t}$. The discrete form of the noise is assumed to be continuous white noise with variance Q as follows

$$\hat{Q} \triangleq \int_{k\Delta t}^{(k+1)\Delta t} \mathrm{e}^{At}Q\mathrm{e}^{A^\intercal t}\mathrm{d}t. \tag{3.15}$$

The integral can seldom be solved analytically, however a simple approximation can be obtained by applying the Taylor expansion. The solution gives

$$\hat{Q} \approx Q\Delta t + 0.5AQ\Delta t^2 + 0.5QA^\intercal\Delta t^2 \tag{3.16}$$

The measurement updates of the filter use the accelerometers to compensate the pitch and roll angles. The static measurement covariance R_{N_a} is defined here as zero-mean Gaussian noise. The standard deviation is computed similarly to equation (3.11), by applying here the accelerometer random walk noise coefficient N from Table 3.2 or Table 3.4 depending on the sensor.

The accelerometers naturally measure the accelerations exerted on the vehicle, which are considered perturbations when it comes to the estimation of the gravity vector and therefore the gravity model (see Fig. 3.6). The filter estimates the accelerometers residual in order to compensate for these perturbations, as described in detail in Suh (2010). This technique brings better performance in dynamic environments and does not interfere the filter design process described in this work.

3.5 Experimental Results

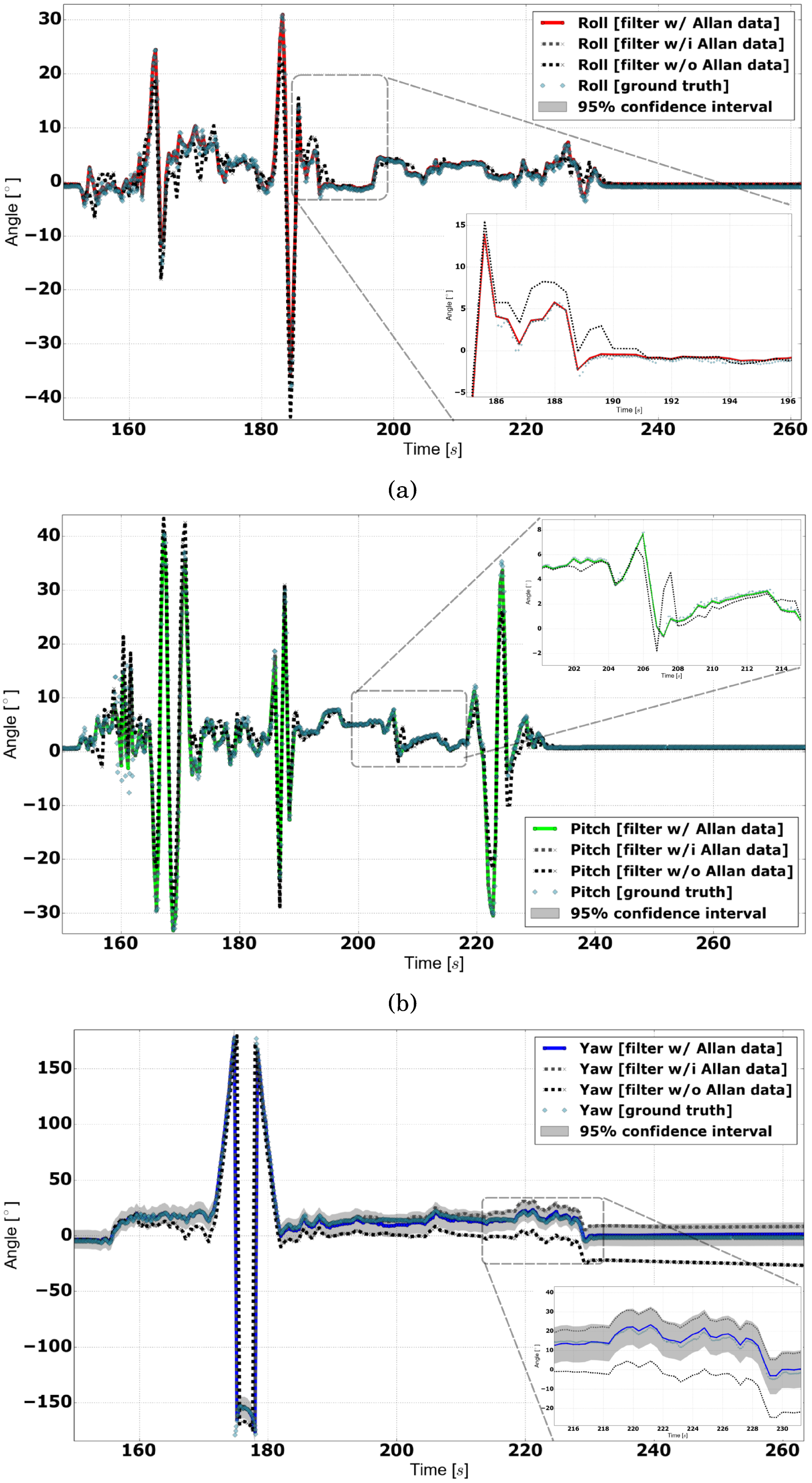
The first experiment using the presented AHRS was performed by manually moving the DEFT-based IMU (i.e. not on board a vehicle) in a laboratory environment. The DEFT-based is lighter than the FOG-based IMU and therefore easier to manually move with one hand.

Objective. In order to evaluate the added value of the sensor characterization in the approach, three filter instances are compared against the measured ground truth. The three instances are identical to each other, with the only difference being that one encompasses the results obtained by the sensor characterization in Section 3.2, while the others do not. The three filter instances run in parallel, having the same system of equations except for the uncertainty values. The objective is to estimate and evaluate the impact of different inertial error models.

Setup. A set of infrared emitting and sensing cameras were used, which covered the testbed area and could track reflective markers attached to the IMU. These cameras are part of a ground truth tracking system, which can deduce the position and orientation of objects equipped with such markers. The IMU was freely moved during several minutes starting from and finishing at static position. The algorithm implemented works as follows: the AHRS estimates the initial attitude by computing the gravity vector during the first few seconds and calculates the relative orientation with respect to the local horizontal plane. This process provides the best heading estimation with respect to the Geographic North at initialization time, i.e. Gyrocompassing, Rogers (2003). Once the initial attitude is estimated, the AHRS fuses the information from the accelerometers and gyroscopes using the filter scheme described in Section 3.4. The filter then estimates the error quaternion and inertial sensors bias in order to correct the attitude quaternion as in equation (3.6).

Table 3.6: Attitude Root Mean Square Error (RMSE) for the experimental results

Filter scheme	Roll [°]	Pitch [°]	Yaw [°]
Filter w / Allan data	0.2875	0.2951	1.7010
Filter w / i Allan data	0.3709	0.3148	5.1360
Filter w / o Allan data	1.2788	1.6265	23.9185



(a)

(b)

(c)

Figure 3.7: The AHRS experimental results using the DETF-based IMU on the manual movement test. (a) roll, (b) pitch and (c) yaw angle, with zoomed areas where details in performance among the three filter instances are visible and more appealing.

Evaluation. The results, together with the ground truth reference, are depicted in Fig. 3.7 using Euler angles (i.e. roll, pitch and yaw). The instance labelled as *filter w/ Allan data* used the results from Section 3.2 and shows the 2σ confidence interval. The *filter w/i Allan data* used *incomplete* sensor error model. It means there is not value for the bias drift, which is model by the bias instability and rate random walk, $\beta = 0$, $\sigma^2_{B_g} = 0.00$ and $\sigma^2_{K_g} = 0.00$. It alters equation (3.9) setting the $I_{3\times3}$ entry of the bias to zero. The initial bias is not used to correct the gyroscopes information. The *filter w/o Allan data* uses the sensor error model but with wrong uncertainty values, $\beta = 1$, $\sigma^2_{B_g} = 0.001$ and $\sigma^2_{K_g} = 1 \cdot 10^{-06}$. It alters the model of the bias drift with a higher correlation time and therefore negatively affecting the evolution of the bias. The pitch and roll angles performed more accurate at dynamic regime when using the values from the Allan variance analysis since a precise sensor bias estimation is computed by the filter. This is critical for INS in highly dynamic environments. Heading suffers from drift in all the filter instances, mainly due to the fact that no other compensating mechanisms exist in the IMUs (e.g. magnetometers). However, the drift is more pronounced for the *filter w/o Allan data*, as it does not include the noise coefficients deduced from the Allan variance analysis. A noticeable drift also appears in the heading angle of the *filter w/i Allan data* which is expected since the filter does not estimate the bias evolution as in equation (3.12) and (3.13). The Root Mean Square Error (RMSE) calculated by comparing the ground truth data versus the attitude estimated by the filter instances is shown in Table 3.6. For all three angles the *filter w/ Allan data* has lower RMSE.

3.6 Navigation Results

The two characterized IMUs are part of the navigation systems of two distinctly different robotic platforms, see Fig. 3.2. The DETF-based IMU is mounted on the ExoMars Test Rover (ExoTeR) due to restrictions in volume. ExoTeR is a laboratory rover prototype that resembles the ExoMars rover mobility configuration in scale, Poulakis et al. (2015). ExoTeR's sensor suite also includes a stereo camera pair, a Time of Flight (TOF) camera and the actuator encoders and potentiometers. ExoTeR weighs 25 kg and has a ground clearance of 20 cm. The track width and wheel base are 62 cm and 53 cm respectively, see Appendix A for further details. The FOG-based IMU operates on board the Dagon Autonomous Underwater Vehicle (AUV), which is also equipped with a stereo camera system, a Doppler Velocity Log (DVL), a mechanical scanning sonar and a pressure sensor. Dagon has a mass of 75 kg, a length of 110 cm and is equipped with six thrusters to provide high maneuverability. The information from the DVL, the pressure sensor and the IMU are fused to obtain a dead reckoning pose as described in Hildebrandt et al. (2012, 2014).

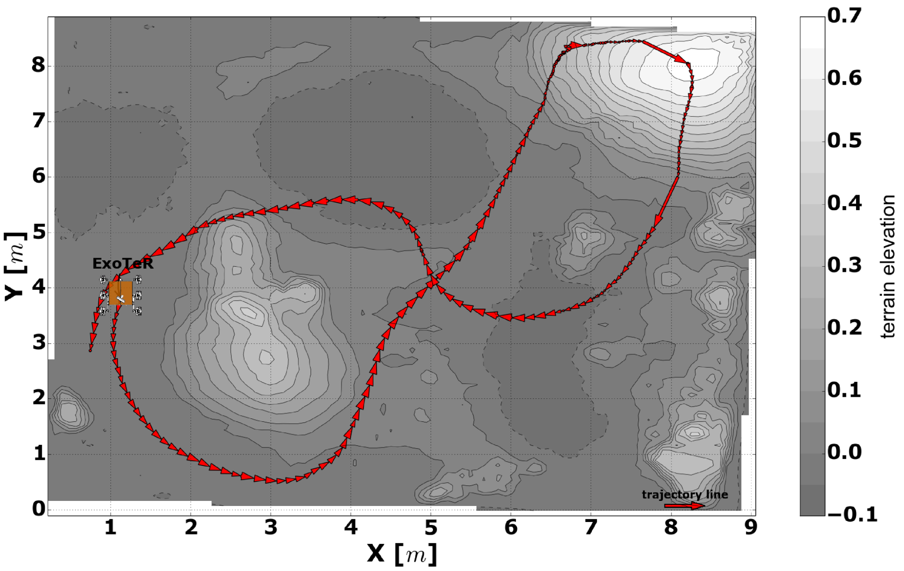


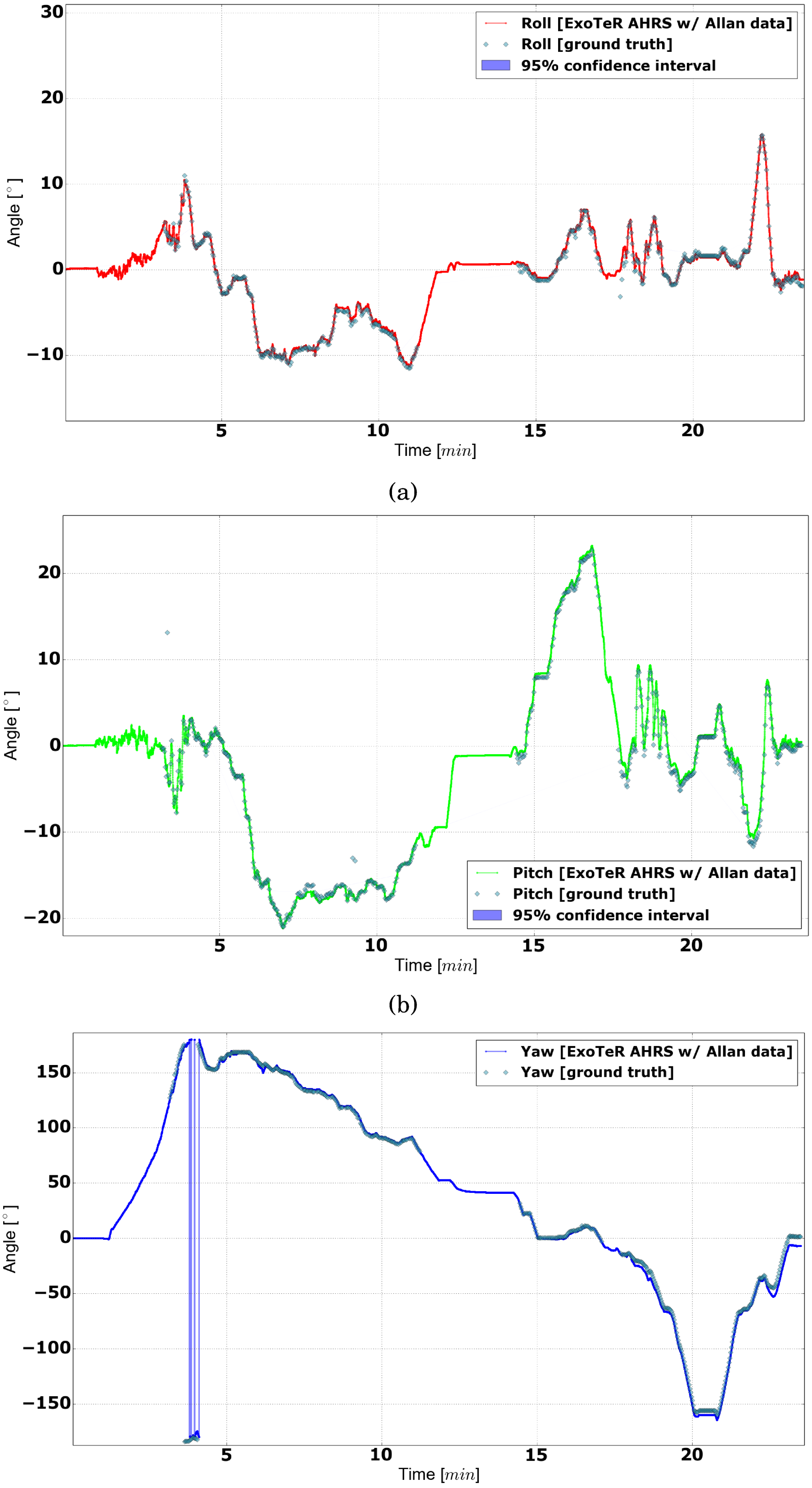
Figure 3.8: Trajectory path performed by ExoTeR in the ESA/ESTEC Planetary Robotics Lab.

3.6.1 Planetary rover

Objective. The performance onboard a planetary rover is worthy of investigation. After understanding the real benefit of following a systematic design approach, the solution onboard ExoTeR is presented. The objective is to measure the performance in order to combine the AHRS in a dead reckoning solution with a motion model in Section 4.

Setup. The experiments with ExoTeR were performed on a Mars-like testbed in the Planetary Robotics Lab (PRL) of the European Space Research and Technology Centre (ESTEC) - the largest site of the European Space Agency (ESA). The testbed consists of a $9\,\mathrm{m} \times 9\,\mathrm{m}$ test area with different soil types and a set of twelve Vicon cameras for the ground truth measurements Hidalgo-Carrio et al. (2011). ExoTeR performed a mapping and localization task while executing a figure eight trajectory of an accumulated track of approximately $25\,\mathrm{m}$, see Fig. 3.8.

Evaluation. The attitude results of the AHRS, running onboard the rover, versus the measured ground truth are depicted in Fig. 3.9. The filter performs accurately for the pitch and roll angles with only $0.5°$ RMSE, while it is satisfactory for the yaw. The difference in the RMSE of the pitch and roll between this and the standalone IMU experiments, see Table 3.6, is due to the difference in accuracy of the ground truth measurements. The accuracy of the camera-based ground truth systems largely depends on the number of cameras and the area covered.



(a)

(b)

(c)

Figure 3.9: ExoTeR attitude results using the DETF-based IMU and the proposed AHRS using the Allan variance to characterize the noise coefficient. (a) roll, (b) pitch and (c) yaw angle

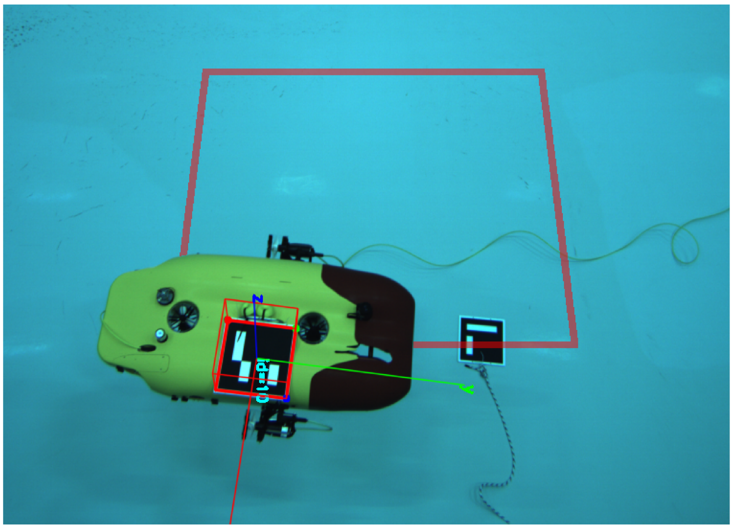


Figure 3.10: The AUV Dagon equipped with an ArUco marker Garrido-Jurado et al. (2014) in the maritime exploration facility of the DFKI Robotic Innovation Center (RIC).

In the case of the ESA PRL, due to the larger size of the testbed, the number of camera-per-m^2 is lower than the laboratory where the standalone IMU experiment was conducted. Finally, the sudden change in heading close to 4 min in Fig. 3.9c is due to the mathematical representation of the depicted angle in degrees between $180°$ and $-180°$.

3.6.2 Underwater vehicle

Objective. It is of interest to evaluate the performance using the FOG-based IMU onboard the underwater vehicle. The availability of a reliable ground truth in water conditions is still an open question in maritime robotics. Very few commercial applications are accessible on the marker Qualysis (2017). The measurement of *truth* position is challenging in underwater scenarios and the measurement of attitude is even more difficult. Here, an *ad-hoc* ground truth solution to conduct the experiment and measure the heading is presented. The goal is to measure the heading in order to evaluate the AHRS for vehicle navigation in a GPS-denied environment.

Setup. The Dagon AUV experiments were performed in the Maritime Exploration Facility of the DFKI - Robotics Innovation Centre (RIC) at Bremen. Dagon was set to autonomously follow a square-like trajectory of $2\,\mathrm{m} \times 2\,\mathrm{m}$, always facing to the next way-point. The commanded trajectory is shown as a red line in Fig. 3.10. Starting from one of the corners of the square, the AUV performed one complete closed-trajectory in one direction, then changed direction and followed the same trajectory

Table 3.7: Attitude Root Mean Square Error (RMSE) for the navigation results

Filter scheme	Roll [°]	Pitch [°]	Yaw [°]
ExoTeR AHRS w/ Allan data	0.4180	0.5954	4.0412
Dagon AHRS w/ Allan data	-	-	1.9083

back to the starting point. The pattern was repeated for a duration of one hour. The ArUco marker [2] on the bottom of the underwater testbed was used by the AUV to correct the error in position when passing by the marker, Garrido-Jurado et al. (2014). The ground truth pose of the AUV was obtained by tracking the marker attached on Dagon using a statically mounted camera on the side of the basin. The coordinate frame of the detected marker can be seen in Fig. 3.10, while Fig. 3.11 depicts the first and last trajectory performed during the one-hour experiment.

Evaluation. The estimated heading angle is shown in Fig. 3.12. The AHRS with the FOG-based gyroscopes estimated a reliable heading without external correction for a long run. The RMSE error for this test is also summarized in Table 3.7. Dagon's AHRS presented a small heading RMSE since such gyroscopes provided a lower drift in comparison with the DETF-based gyroscopes. It should be noted that only heading or yaw is computed in this experiment since a commercial ground truth attitude measurement, for measuring the three Euler angles, is not available at the Maritime

[2] ArUco: a minimal library for Augmented Reality applications based on OpenCv http://sourceforge.net/projects/aruco/

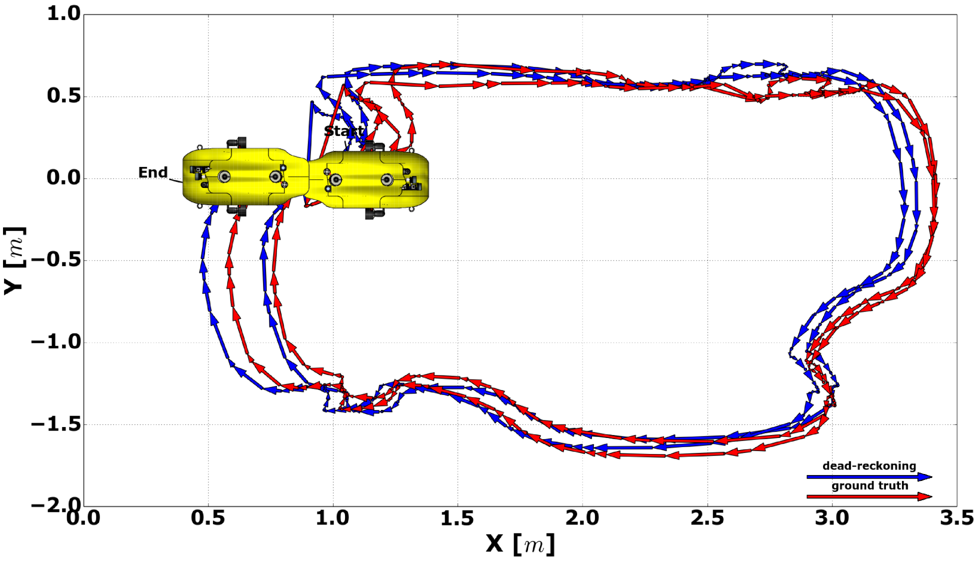


Figure 3.11: Dagon's and ground truth paths performing a square-like trajectory.

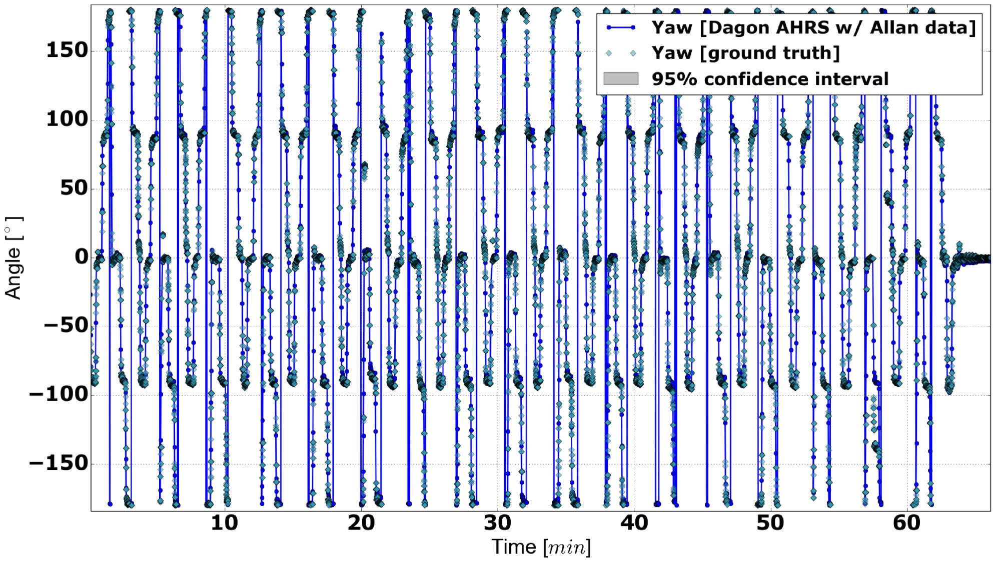


Figure 3.12: Dagon's AHRS and ground truth heading while performing a square trajectory for one hour duration at the Maritime exploration facility.

Exploration Facility.

3.7 Conclusion

The Allan variance and its derivative, the Modified Allan variance from Allan and Barnes (1981), have been successfully applied in a wide range of applications since its inception in 1966. The method has been adopted in 2008 by the inertial sensor community due to its simplicity and effectiveness, IEEE (2008). The allanvar [3] package is implemented by the author of this thesis in the R language and used in the design of an AHRS. The time domain and visual technique in log-log plots serve to analyze the signal quality as well as characterize the most dominant errors underlying the sensory data. The mapping between the frequency and time domain expressed by equation (3.2) significantly helps to characterize the stochastic processes and their coefficients. As a consequence, the design of AHRS are nowadays difficult to conceive without the use of the Allan variance and most sensors manufacturers characterize their products using this method. Though relevant noise coefficients are usually available in sensor datasheets, a more detailed characterization is recommended because it improves accuracy. Datasheets are frequently ambiguous or incomplete on some of the specifications and sensor aging effects can have a strong impact on the error coefficients. These coefficients are essential to properly size the uncertainties in a Bayesian filter approach.

[3] Allan variance R package: `http://cran.r-project.org/package=allanvar`

Specifically, this chapter focuses on the End to End approach for AHRS design using the Allan variance methodology. The work describes the design aspects of a general AHRS which can be adapted to any inertial sensor in order to improve the performance, independently of the sensor technology and its application. The generality of the approach is evaluated by integrating two different inertial measurement units, a DETF-based and a FOG-based IMU, on the AHRS of two robotic platforms. A set of reproducible steps from the inertial sensor characterization until the AHRS design and final evaluation is presented. The influence of the sensor noise model on the filter equations is described. The results show and improvement of 1° RMSE in attitude, pitch and roll, and by several degrees in heading during a freely moving experiment (see Table 3.6). The chapter also presents the effect of partial sensor error modeling on the final accuracy. The delta in attitude estimated by the AHRS is of importance in a 6 DoF odometry as it will be mentioned in Chapter 4.

In future work, low-cost inertial sensors might have noises with complex spectral structures since several random processes are superimposed. A solution based on nonlinear fitting with constraints is proposed by Quinchia et al. (2013a). Their work describes a nonlinear least squares fitting on the log-log plot of the Allan variance. The integration of this nonlinear modeling in a GPS/INS is subject to a future applied research for low-cost navigation.

Chapter 4

Enhanced 3D Odometry

Articulated mobile robots have complex chassis and require a complete model in comparison to skid-steer models (planar assumption). This chapter presents the necessary steps to develop a complete motion model for real-time inertial-aided odometry. The presented methodology relates normal forces with the probability of a contact point to slip. This work uses the mathematical principle of the general transformation approach, first introduced by Muir and Neuman (1986). The method is applied in two wheel mobile platforms with a significantly diverse locomotion system, the ExoMars Test Rover: a research planetary rover and Asguard: a simple yet capable leg-wheel hybrid robot. The performance of the approach is demonstrated during extensive field testing within different unstructured environments. The results show that the accuracy is increased by weighting the least squares solution using static force predictions. Error analysis and comparison with state-of-the-art planar and contact point odometry are discussed, resulting in a richer modeling technique and a more accurate localization.

4.1 Introduction

Localization focuses on determining the pose (i.e. position and orientation) with respect to a global coordinate frame and typically within a map. Probabilistic localization frameworks, as variants of Bayes filters, have been used to solve the localization problem, Montemerlo et al. (2002); Thrun et al. (2005); Schwendner et al. (2013). Those frameworks fuse sensory data to propagate robot's pose while moving. The process is separated into *prediction* and *correction* of the pose using proprioceptive and exteroceptive sensors respectively.

The estimation of motion in the *prediction* step is commonly performed using a motion model. Motion models are means for mobile robot odometry, have real-time capabilities and are inexpensive in comparison with sophisticated map matching techniques. The inheritance from indoor robotics, which operated in structured and planar environments, brings efficient and simple techniques like the skid-steer methods, see Chapter 29, *Inertial Sensing, GPS and Odometry* at Siciliano and Khatib (2016). However, this simplicity causes a performance degradation specially while localizing on complex uneven terrains like planetary surfaces. As a consequence, extensive sensing capabilities have to overcome poor odometry performance, but before applying sensing means, the evaluation of a complete 6 DoF motion model in a 3D unstructured environment is worthy of investigation.

In this chapter a method that is able to *optimally* [1] combine the motion induced at each contact point is presented. The approach is motivated by the observation that the robot has different normal forces at each contact point while moving. The primary contribution of the work is fusing, in a unified framework, normal forces with the probability of each contact to slip. A performance analysis also demonstrates that a complete motion model is more accurate than skid-steer and previous contact point models as Schwendner et al. (2013). The influence of using normal forces is also investigated. This chapter structures as follow: related work is presented in the next section. The details of the proposed technique are described in Section 4.3. Section 4.4 contains the results from challenging large-scale unstructured environments. The method is applied in two rovers, the Asguard platform, a capable but simple skid-steer leg-wheel robot, Joyeux et al. (2011), and the ExoTeR, a laboratory rover prototype that resembles the ExoMars rover in scale, Poulakis et al. (2015). Conclusions and limitations of the approach are mentioned in brief at the end of this chapter.

4.2 Related Work

Odometry has been widely studied since many robots only rely on dead reckoning processes for basic localization. Starting from a known pose, odometry involves the calculation of a robot's body configuration from encoder readings. The disadvantage of dead reckoning systems is very well-known. The localization uncertainty grows unbounded due to accumulation of errors. Considerable research has been done in order to reduce the undesirable effect of poor odometry performance. The literature focuses on three specific types of errors, Borenstein and Feng (1996b) and Siciliano and Khatib (2016):

- Systematic errors, such as misalignment of actuators and uncertainty about effective link dimensions.

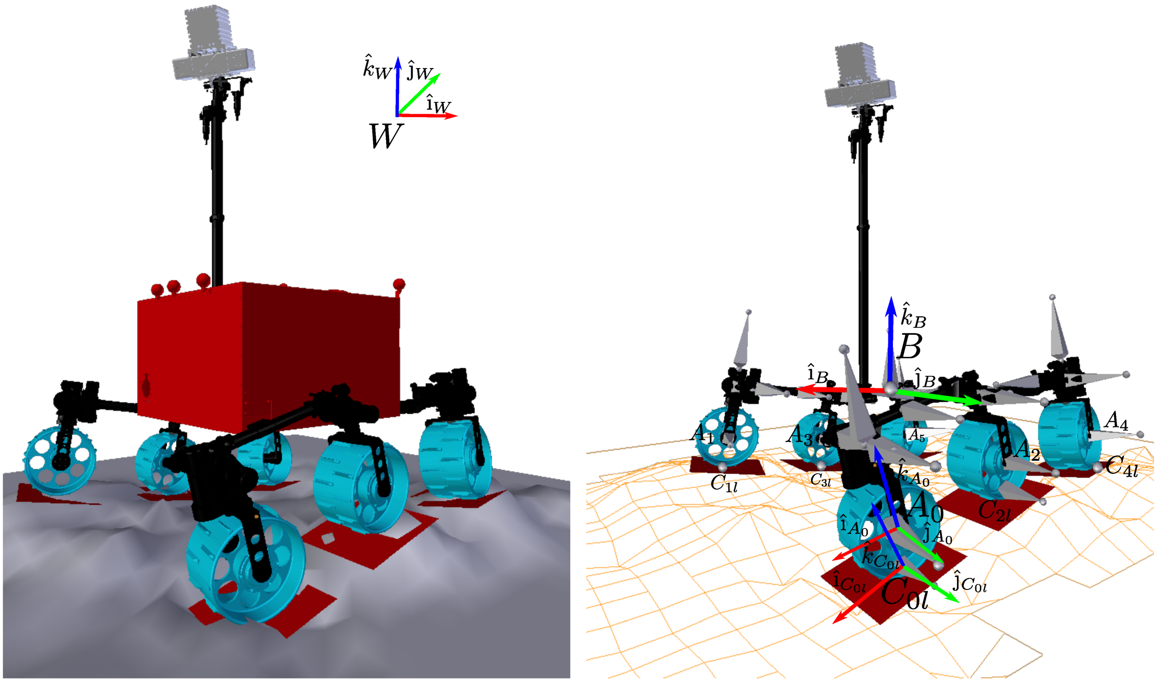
[1] Optimally here refers to the best mathematically achievable from a nonlinear least squares perspective.

- Non-systematic errors, which include slippage, dragging, forces and multi-point wheel contact models with the ground.
- Numerical drift and linearization errors, due to discrete-time integration of delta poses.

The elimination of systematic errors is described from early research in Borenstein and Feng (1996b). Calibration methods were applied for specific trajectories in order to reduce the effect of unequal wheel diameter and uncertainty about the wheelbase (i.e. deterministic errors). A general method based on least squares is proposed in Antonelli et al. (2005) with no limitation to a particular predefined path. Differential-drive platforms are commonly discussed for indoor scenarios because they are mechanically simple to build. Position tracking in challenging terrains was also investigated in Lamon and Siegwart (2007). Simulations results are investigated for the Rocky7 rover in Tarokh and McDermott (2005) defining the wheel contact angle and slip vector.

Slippage correction has been the main non-systematic error as it causes bad results affecting the final pose. Visual odometry techniques are used to overcome the effect of slippery terrains, Helmick et al. (2004a); Rehder et al. (2012). Fuzzy logic has also been used to detect wheel slippage by comparing the motor current on the Flex-Nav architecture, Ojeda and Borenstein (2002). Their work corrects wheel-slippage based on motor current measurements. They introduce a linearized function to relate electric current and wheel-terrain interaction, Ojeda et al. (2006). Rogers-Marcovitz et al. (2012) present a delayed state filter technique in combination with a vehicle system model to correct wheel slip. Adaptive odometry by means of terrain classification using inertial data has been investigated in Reinstein et al. (2013). A regression function is trained offline to directly output the adaptive correction coefficient of the odometry model. Terrain classifiers are presented in Giguère and Dudek (2008) based on clustering methods with field testing results using the hexapod robot. Performance analysis for omnidirectional robots in rough terrain is available in Ishigami et al. (2011). Slip compensation based on wheel velocity differences is analyzed for a traditional skid-steer kinematics in circular trajectories.

Despite all efforts, the dead reckoning problem requires a more elaborate analysis and understanding to identify the impact of motion models. A complete motion model cannot correct non-systematic errors, which require sensing means as visual odometry. However, systematic errors, e.g. one single wheel slips, can be mitigated by effectively modeling the robot's locomotion system. As of today, the improvement evaluation of such impact is still an open question. A field testing comparison of a more sophisticated motion model for robot odometry motivates the research described in this chapter.



(a) (b)

Figure 4.1: Illustration of the 3D kinematics for ExoTeR. (a) The rover on an uneven terrain. Depicted is 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.

4.3 Odometry Motion Models

Motion models and Attitude and Heading Reference Systems (AHRS) are the two principal techniques for basic odometry. Many robotic applications combine both for consistency over time. The attitude kinematics is self-sufficient with inertial sensors except for the less observable angle (i.e. the heading or yaw), Crassidis and Markley (2003); Hidalgo-Carrio et al. (2012b). The heading angle is only observable by gyroscopes since it is not directly deduced from the gravity vector. Motion models are more accurate than double integration of accelerometer readings and are the basics for dead reckoning in mobile robotics.

4.3.1 Kinematic Modeling

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

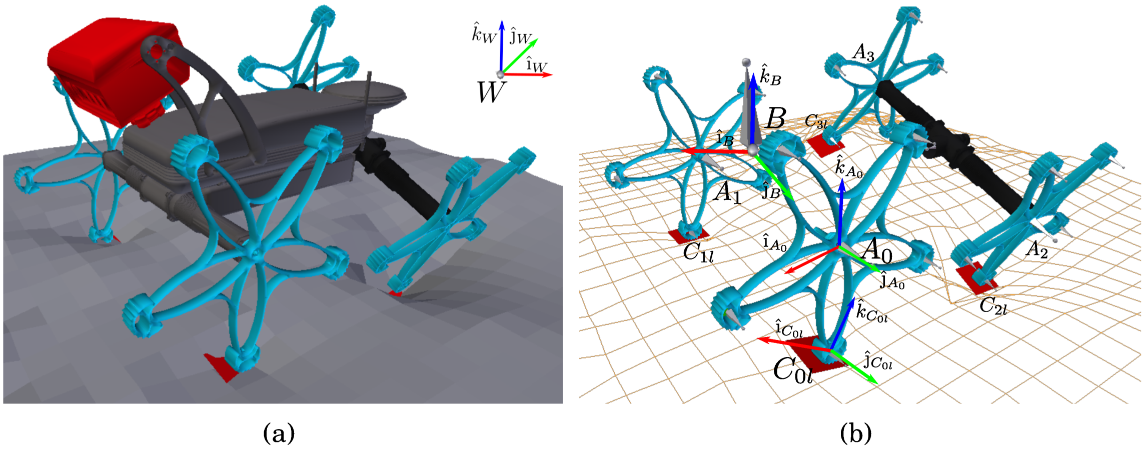


Figure 4.2: Illustration of the 3D kinematics for Asguard. (a) The robot on an uneven terrain. Depicted is the world reference frame W. (b) Kinematics modeling. B is the body frame located in the front axis between the middle wheels, A_i is the wheel frame and C_{il} the contact point frame. The rover's delta pose displacements are computed as composite equations of wheel Jacobian matrices.

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 $\boldsymbol{q} = [q_0\ q_1\ \ldots\ q_n]$ where n is the number of degrees of freedom of the mechanism, see Fig. 4.1 and Fig. 4.2 for a visual illustration. The B frame is related to a fixed world frame W by the pose

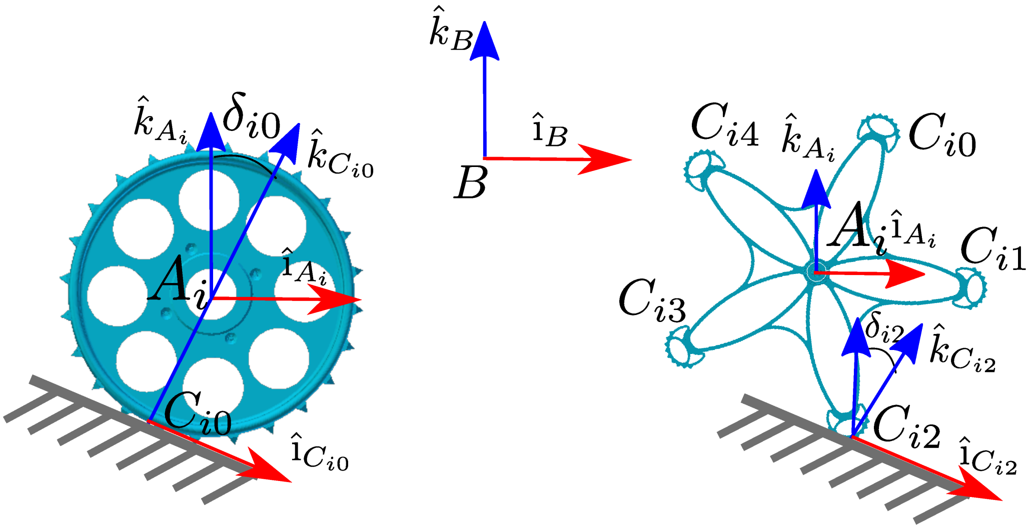


Figure 4.3: Schematic representation of coordinate frames for a wheel i on an inclined terrain for conventional wheels and hybrid leg-wheel systems. A single point of contact $l = 0$ is defined for a conventional wheel (left) and multiple contact points $l = 0, \cdots, 4$ are defined in a hybrid leg-wheel (right). The contact angle is a notable distinction between indoor and outdoor motion models.

vector $\vec{u} = \boldsymbol{u}_{W,B} = [x_W \ y_W \ z_W \ \phi_W \ \theta_W \ \psi_W]$.

Planetary rovers 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}, see Fig. 4.3. Pure walking machines might require two angles at the point in contact (i.e. the gradients in the lateral and transversal direction). In addition, the motion of the contact point consists of a slip vector $\boldsymbol{\varepsilon}_{il}$, which is modeled in three dimensions. A translation in the $\hat{\imath}_{C_{il}}$ axis by ξ_{il}, lateral slip η_{il} along the $\hat{\jmath}_{C_{il}}$ axis and rotational slip ζ_{il} along the $\hat{k}_{C_{il}}$ axis, $\boldsymbol{\varepsilon}_{il} = [\xi_{il} \ \eta_{il} \ \zeta_{il}]$.

Mobile robots are commonly commanded by desired velocities. The mapping between using the robot body frame Cartesian space rate vector $\dot{\boldsymbol{u}} = \left[\dot{x}_B \ \dot{y}_B \ \dot{z}_B \ \dot{\phi}_B \ \dot{\theta}_B \ \dot{\psi}_B\right]$ and the joint space rate vector with the contact rate angle $\dot{\delta}_{il}$ and slip rate vector $\dot{\boldsymbol{\varepsilon}}_{il}$ 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 Fig. 4.4. The derivative is

$$\begin{aligned}\dot{T}_{\bar{B},B} = \dot{T}_{\bar{B},\bar{C}_{il}} T_{\bar{C}_{il},C_{il}} T_{C_{il},B} +& \\ T_{\bar{B},\bar{C}_{il}} \dot{T}_{\bar{C}_{il},C_{il}} T_{C_{il},B} +& \\ T_{\bar{B},\bar{C}_{il}} T_{\bar{C}_{il},C_{il}} \dot{T}_{C_{il},B}&\end{aligned} \tag{4.1}$$

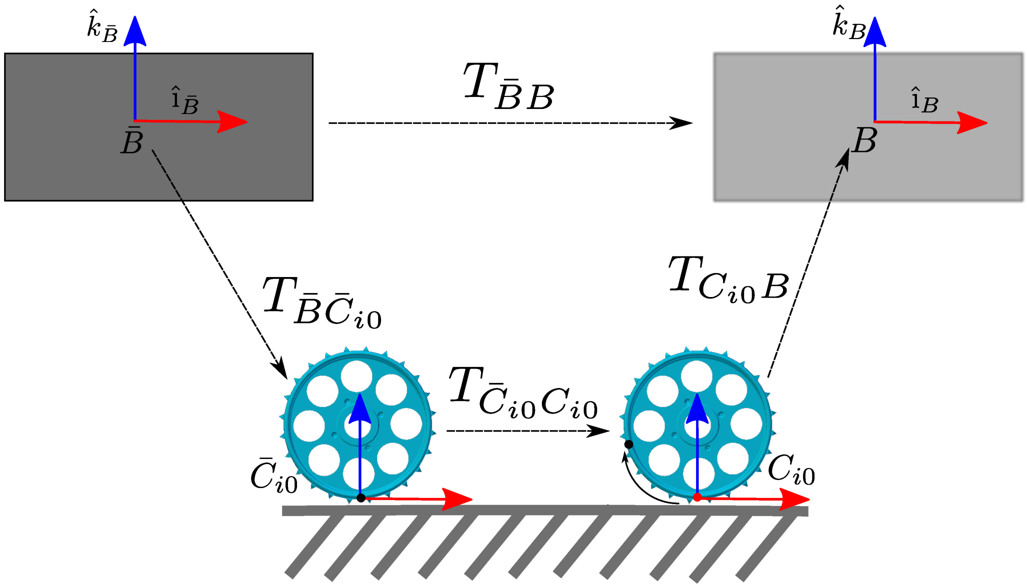


Figure 4.4: 3D kinematics of movement to induce the delta displacement from a single wheel i with one contact point $l = 0$. The wheel rotates clockwise.

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 appreciable that $\dot{T}_{\bar{B},B}$ has the form of a free body in motion with linear and angular velocities, Tarokh and McDermott (2005).

$$\dot{T}_{\bar{B},B} = \begin{bmatrix} 0 & \dot{\psi}_{\bar{B}} & \dot{\theta}_{\bar{B}} & \dot{x}_{\bar{B}} \\ \dot{\psi}_{\bar{B}} & 0 & -\dot{\phi}_{\bar{B}} & \dot{y}_{\bar{B}} \\ -\dot{\theta}_{\bar{B}} & \dot{\phi}_{\bar{B}} & 0 & \dot{z}_{\bar{B}} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.2}$$

It is possible to obtain the rover configuration rates $\dot{\boldsymbol{u}}$ expressed in term of joints quantities by substituting equation (4.1) into the skew-symmetric matrix of equation (4.2). After rearranging the terms and equating the elements on both sides of the equation the mapping is obtained. The resulting Jacobian matrix J_{il}, related to a contact point il, has the form

$$\begin{bmatrix} \dot{x}_{\bar{B}} \; \dot{y}_{\bar{B}} \; \dot{z}_{\bar{B}} \; \dot{\phi}_{\bar{B}} \; \dot{\theta}_{\bar{B}} \; \dot{\psi}_{\bar{B}} \end{bmatrix}^{\mathsf{T}} = J_{il} \begin{bmatrix} \dot{\boldsymbol{q}} \; \dot{\boldsymbol{\varepsilon}}_{il} \; \dot{\boldsymbol{\delta}}_{il} \end{bmatrix}^{\mathsf{T}} \tag{4.3}$$

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.

$$\begin{bmatrix} I_{6\times 6} \\ I_{6\times 6} \\ \vdots \\ I_{6\times 6} \end{bmatrix} \begin{bmatrix} \dot{x}_{\bar{B}} \\ \dot{y}_{\bar{B}} \\ \dot{z}_{\bar{B}} \\ \dot{\phi}_{\bar{B}} \\ \dot{\theta}_{\bar{B}} \\ \dot{\psi}_{\bar{B}} \end{bmatrix} = J \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\varepsilon}} \\ \dot{\boldsymbol{\delta}} \end{bmatrix} \equiv S\dot{\boldsymbol{u}} = J\dot{\boldsymbol{p}} \tag{4.4}$$

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 non-sensed quantities and equation (4.4) separates into the following form

$$\begin{bmatrix} S_s & S_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_s \\ \dot{\boldsymbol{u}}_n \end{bmatrix} = \begin{bmatrix} J_s & J_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_s \\ \dot{\boldsymbol{p}}_n \end{bmatrix} \tag{4.5}$$

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

$$\begin{bmatrix} S_n & -J_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_n \\ \dot{\boldsymbol{p}}_n \end{bmatrix} = \begin{bmatrix} -S_s & J_s \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_s \\ \dot{\boldsymbol{p}}_s \end{bmatrix} \equiv \Omega \boldsymbol{\nu} = \boldsymbol{b} \tag{4.6}$$

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 $E = \boldsymbol{e}^{\intercal}\mathcal{W}\boldsymbol{e}$, where $\mathcal{W}$ encodes the individual contribution of each kinematics chain to the estimated solution

$$E = \boldsymbol{e}^{\intercal}\mathcal{W}\boldsymbol{e} = (\boldsymbol{b} - \Omega\boldsymbol{\nu})^{\intercal}\mathcal{W}(\boldsymbol{b} - \Omega\boldsymbol{\nu}). \tag{4.7}$$

The least squares solution of the system in equation (4.7) provides a minimum error vector $\boldsymbol{e}$. However, poor traction from a single wheel might influence the solution by increasing the resulting error. The solution is weighted (adapted) in order 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. Consequently, the motor current signal is proportional to the torque and therefore related to the traction force. However, the traction force relation from the torque is not straight forward since in the case of Asguard multiple wheels and foot locations affect the measurement. A dedicated identification model would be required in order to estimate such complex relation, in the direction of the work described in Ojeda and Borenstein (2002). Alternatively, quasi-static forces are simpler to compute, and precise in static regimen with less accuracy at dynamic range. Therefore, in order to overcome such limitations a quasi-static force estimation is explained in the following. In addition, the lowest point along the wheel circumference is assumed to be always in contact with the ground as depicted in Fig. 4.4. The assumption is valid for most of the cases. However, it does not always hold on highly uneven terrains. Accurate estimations are only possible with additional sensors added to the wheels or feet in order to detect contact and forces more realistically, Fondahl et al. (2012).

4.3.2 Quasi-Static Forces Estimation

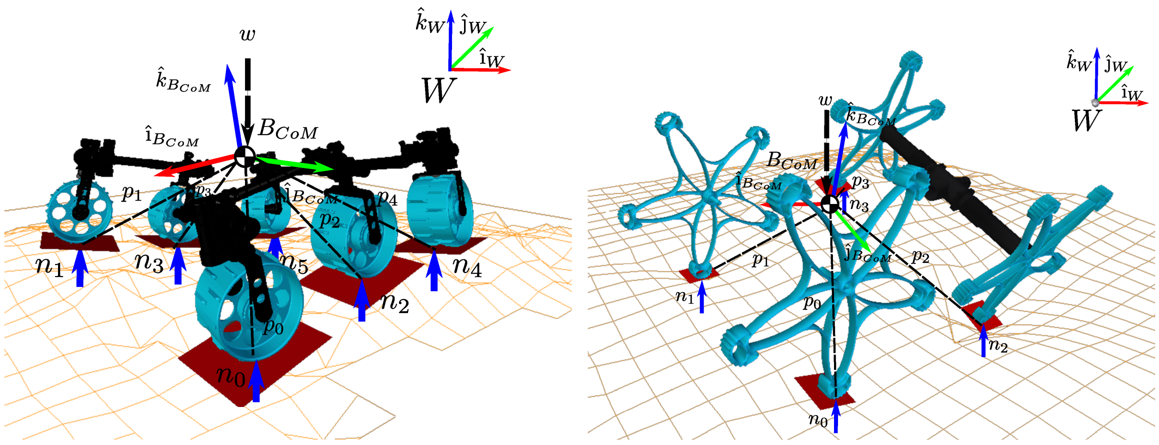
Computation of odometry is highly influenced by the amount of wheel slip, which in turn depends on the maximum usable tractive force between ground and the wheel. The maximum available traction, assuming a simple friction model is proportional to the normal force on each wheel, Wong (2001). The normal force distribution among the wheels of the robot is based on static reaction forces caused by the robot's weight. The computation is performed every sample time, based on the contact point positions calculated by the transformation matrices, hence called quasi-static force estimation.

For the case of Asguard it is assumed that the leg which is the lowest, in relation to the robot body plane, is in contact with the ground and coincident with the contact point frame C_{il}. On highly uneven surfaces, it is possible that the higher foot is in contact with the ground and the lowest is not. For ExoTeR a unique contact point $l = 0$ is equally defined at the lowest point of the wheel circumference. The lowest point assumption is mostly valid for relatively uneven surfaces. Multiple points can be in contact simultaneously as well, but mainly a single contact point per wheel is leading the motion.

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). 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 Fig. 4.5) is defined with the origin coinciding with the CoM of the robot, but aligned to the world frame W. The terms for $\boldsymbol{n}_{B'_{CoM}i}$ and $\boldsymbol{w}$ in B'_{CoM} are given by,

$$\boldsymbol{n}_{B'_{CoM}i} = \begin{bmatrix} 0 \\ 0 \\ n_i \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{4.8}$$

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. Let the rotation from B_{CoM} to B'_{CoM} and the



(a) (b)

Figure 4.5: Free body diagram for static forces computation in (a) ExoTeR and (b) Asguard.

position vector $\boldsymbol{p}_{B_{CoM}i}$ be given by

$$R_{B_{CoM}B'_{CoM}} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}, \boldsymbol{p}_{B_{CoM}i} = \begin{bmatrix} p_{i\hat{\imath}} \\ p_{i\hat{\jmath}} \\ p_{i\hat{k}} \end{bmatrix}, \tag{4.9}$$

Using equation (4.9), the relationship between $\boldsymbol{n}_{B'_{CoM}i}$ and $\boldsymbol{n}_{B_{CoM}i}$ is given by,

$$\boldsymbol{n}_{B_{CoM}i} = R_{B_{CoM}B'_{CoM}} \boldsymbol{n}_{B'_{CoM}i} = \begin{bmatrix} r_{02} \\ r_{12} \\ r_{22} \end{bmatrix} n_i \tag{4.10}$$

Computing torques from equation (4.9) and equation (4.10),

$$\boldsymbol{\tau}_{B_{CoM}i} = \boldsymbol{p}_{B_{CoM}i} \times \boldsymbol{n}_{B_{CoM}i} = \begin{bmatrix} p_{i\hat{\imath}} \\ p_{i\hat{\jmath}} \\ p_{i\hat{k}} \end{bmatrix} \times \begin{bmatrix} r_{02} \\ r_{12} \\ r_{22} \end{bmatrix} n_i \tag{4.11}$$

$$\boldsymbol{\tau}_{B_{CoM}i} = \begin{bmatrix} p_{i\hat{\jmath}} r_{i\hat{k}} - p_{i\hat{k}} r_{i\hat{\jmath}} \\ p_{i\hat{k}} r_{i\hat{\imath}} - p_{i\hat{\imath}} r_{i\hat{k}} \\ p_{i\hat{\imath}} r_{i\hat{\jmath}} - p_{i\hat{\jmath}} r_{i\hat{\imath}} \end{bmatrix} n_i \tag{4.12}$$

Using equation (4.12) let,

$$\begin{bmatrix} p_{i\hat{\jmath}} r_{i\hat{k}} - p_{i\hat{k}} r_{i\hat{\jmath}} \\ p_{i\hat{k}} r_{i\hat{\imath}} - p_{i\hat{\imath}} r_{i\hat{k}} \\ p_{i\hat{\imath}} r_{i\hat{\jmath}} - p_{i\hat{\jmath}} r_{i\hat{\imath}} \end{bmatrix} = \begin{bmatrix} t_{i\hat{\imath}} \\ t_{i\hat{\jmath}} \\ t_{i\hat{k}} \end{bmatrix} \tag{4.13}$$

Since the actual force calculation at each contact point is highly dependent to the chassis and number of contact points, the equations for the ExoTeR planetary platform and the Asguard robot are described separately in the following.

4.3.2.1 ExoTeR Equations

The free body diagram for computation of static forces is depicted in Fig. 4.5a. The objective is to derive the equations for the values of n_i in the case of the ExoTeR kinematics. 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. Sum of forces along $\hat{k}_{B'_{CoM}}$ equals the weight of the robot.

$$\Sigma n_i = mg \tag{4.14}$$

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

$$\Sigma \left(\boldsymbol{p}_{B_{CoM}i} \times \boldsymbol{n}_{B_{CoM}i} \right) |_{\hat{\imath}_{B_{CoM}}} = 0 \tag{4.15}$$

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

$$\Sigma \left(\boldsymbol{p}_{B_{CoM}i} \times \boldsymbol{n}_{B_{CoM}i} \right) |_{\hat{\imath}_{B_{CoM}}} = 0 \tag{4.16}$$

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

$$\left(\boldsymbol{p}_{B_{CoM}0} \times \boldsymbol{n}_{B_{CoM}0} + \boldsymbol{p}_{B_{CoM}2} \times \boldsymbol{n}_{B_{CoM}2} \right) |_{\hat{\jmath}_{B_{CoM}}} = 0 \tag{4.17}$$

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

$$\left(\boldsymbol{p}_{B_{CoM}1} \times \boldsymbol{n}_{B_{CoM}1} + \boldsymbol{p}_{B_{CoM}3} \times \boldsymbol{n}_{B_{CoM}3} \right) |_{\hat{\jmath}_{B_{CoM}}} = 0 \tag{4.18}$$

6. Sum of torques along $\hat{\jmath}_{B_{CoM}}$ for all wheels is zero.

$$\Sigma \left(\boldsymbol{p}_{B_{CoM}i} \times \boldsymbol{n}_{B_{CoM}i} \right) |_{\hat{\jmath}_{B_{CoM}}} = 0 \tag{4.19}$$

Substituting equations (4.11), (4.12) and (4.13) in equations (4.15) - (4.19) and combining them with equation (4.14) gives,

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ t_{0\hat{\imath}} & t_{1\hat{\imath}} & t_{2\hat{\imath}} & t_{3\hat{\imath}} & 0 & 0 \\ 0 & 0 & 0 & 0 & t_{4\hat{\imath}} & t_{5\hat{\imath}} \\ t_{0\hat{\jmath}} & 0 & t_{2\hat{\jmath}} & 0 & 0 & 0 \\ 0 & t_{1\hat{\jmath}} & 0 & t_{3\hat{\jmath}} & 0 & 0 \\ t_{0\hat{\jmath}} & t_{1\hat{\jmath}} & t_{2\hat{\jmath}} & t_{3\hat{\jmath}} & t_{4\hat{\jmath}} & t_{5\hat{\jmath}} \end{bmatrix} \begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{bmatrix} = \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.20}$$

The system of linear equations (4.20) can be solved for n_i by,

$$\begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ t_{0\hat{\imath}} & t_{1\hat{\imath}} & t_{2\hat{\imath}} & t_{3\hat{\imath}} & 0 & 0 \\ 0 & 0 & 0 & 0 & t_{4\hat{\imath}} & t_{5\hat{\imath}} \\ t_{0\hat{\jmath}} & 0 & t_{2\hat{\jmath}} & 0 & 0 & 0 \\ 0 & t_{1\hat{\jmath}} & 0 & t_{3\hat{\jmath}} & 0 & 0 \\ t_{0\hat{\jmath}} & t_{1\hat{\jmath}} & t_{2\hat{\jmath}} & t_{3\hat{\jmath}} & t_{4\hat{\jmath}} & t_{5\hat{\jmath}} \end{bmatrix}^{-1} \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.21}$$

The reaction forces computes at every time step using equation (4.21) and inputs to the weighted matrix $\mathcal{W}$ (see Section 4.3.3).

4.3.2.2 Asguard Equations

The free body diagram for computation of static forces is depicted in Fig. 4.5b. The objective is to derive the equations for the values of n_i in the case of the Asguard kinematics. The equations are developed based on the fact that the robot has a free joint in the rear axis and this joint cannot transmit any torques. Therefore, the torques in the front and the rear part of the robot along this free joint are independent when the robot is quasi-static. In the following the equations for the Asguard robot are derived. When the robot is quasi-static the following equations apply.

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

$$\Sigma n_i = mg \tag{4.22}$$

2. Sum of torques along $\hat{\jmath}_{B_{CoM}}$ is zero.

$$\Sigma \left(\boldsymbol{p}_{B_{CoM}i} \times \boldsymbol{n}_{B_{CoM}i}\right) |_{\hat{\jmath}_{B_{CoM}}} = 0 \tag{4.23}$$

3. Sum of torques due to $\boldsymbol{n}_0$ and $\boldsymbol{n}_1$ along $\hat{\imath}_{B_{CoM}}$ is zero.

$$\left(\boldsymbol{p}_{B_{CoM}0} \times \boldsymbol{n}_{B_{CoM}0} + \boldsymbol{p}_{B_{CoM}1} \times \boldsymbol{n}_{B_{CoM}1}\right) |_{\hat{\imath}_{B_{CoM}}} = 0 \tag{4.24}$$

4. Sum of torques due to $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ along $\hat{\imath}_{B_{CoM}}$ is zero.

$$\left(\boldsymbol{p}_{B_{CoM}2} \times \boldsymbol{n}_{B_{CoM}2} + \boldsymbol{p}_{B_{CoM}3} \times \boldsymbol{n}_{B_{CoM}3}\right) |_{\hat{\imath}_{B_{CoM}}} = 0 \tag{4.25}$$

Substituting (4.11), (4.12) and (4.13) in (4.23), (4.24) and (4.25), and combining them with (4.22) gives,

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ t_{0\hat{\jmath}} & t_{1\hat{\jmath}} & t_{2\hat{\jmath}} & t_{3\hat{\jmath}} \\ t_{0\hat{\imath}} & t_{1\hat{\imath}} & 0 & 0 \\ 0 & 0 & t_{2\hat{\imath}} & t_{3\hat{\imath}} \end{bmatrix} \begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.26}$$

The set of linear equations (4.26) can be solved for n_i by,

$$\begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ t_{0\hat{\jmath}} & t_{1\hat{\jmath}} & t_{2\hat{\jmath}} & t_{3\hat{\jmath}} \\ t_{0\hat{\imath}} & t_{1\hat{\imath}} & 0 & 0 \\ 0 & 0 & t_{2\hat{\imath}} & t_{3\hat{\imath}} \end{bmatrix}^{-1} \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.27}$$

The reaction forces computes at every time step using equation (4.27) and inputs to the weighted matrix $\mathcal{W}$ (see Section 4.3.3).

4.3.3 Weighted Least Squares Optimization

A limitation of the kinematic model comes from the residual error of the least squares in equation (4.7). Bad measurements might have a big penalty when minimizing the sum of the square of the errors. The measurements that are more informative are given more weight, and those that are less informative are given less weight. This case, e.g. a wheel is producing bad measurements because of poor traction, can be minimized by adjusting the matrix $\mathcal{W}$ in equation (4.7). The solution might suffer from heteroscedasticity due to different units of $SE(3)$ in the $\mathcal{W}$ matrix. However, it is not too impacted by heteroscedasticity since the information is computed at every delta pose for a small period of time. A rule of thumb in heteroscedasticity for least squares is that the influence is small as long as the maximum variance is not greater than four times the minimum variance.

The $\mathcal{W}$ matrix in equation (4.7) is a diagonal matrix which comprises block matrices. Each 6×6 block matrix is a single diagonal matrix associated to each wheel and defines the contact points which are more likely to contribute to the motion of the robot. These block matrices are selected to have the structure $\mathcal{W}_{il} = w_{il}I$ where w_{il} is the contribution of the il^{th} contact point to the resulting body motion. In the ideal case of a balance configuration, all wheels have equal contribution to the robot motion, 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 Inertial Measurement Unit (IMU) and estimates the forces. The estimated forces are computed every delta pose and the *instantaneous* likelihood of each contact point is calculated accordingly with $\sum_i \sum_l w_{il} = 1$.

4.4 Experimental Results

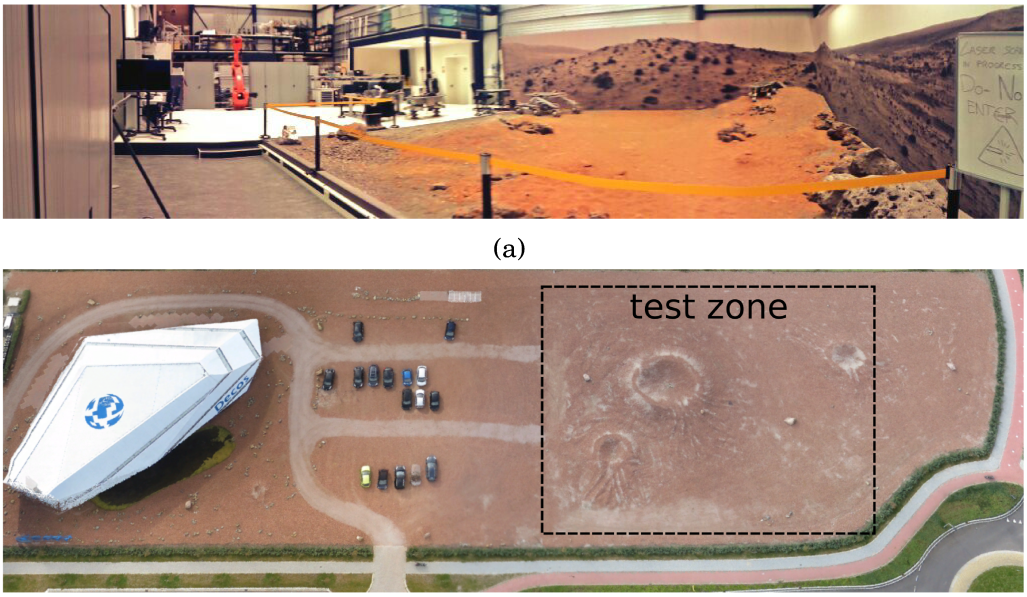
The methodology is applied in two platforms, Asguard and ExoTeR. Asguard is a simple yet highly capable leg-wheel system that is intended to serve as the scout rover unit in a multi-robot exploration scenario. Asguard is able to navigate in complex obstacles while having a simple locomotion system. It is equipped with optimal encoders in each wheel and an absolute potentiometer for the passive joint, Joyeux et al. (2011).

ExoMars Test Rover (ExoTeR), is a laboratory rover prototype for the ExoMars mission. ExoTeR is equipped with a stereo camera pair, a Time of Flight (TOF) camera, actuator encoders and potentiometers in the three passive joints. Both platforms are equipped with GPS and a Sensonor STIM300 IMU (DETF-based IMU) including inclinometers, accelerometers and gyroscopes as described in Chapter 3. A reference trajectory was acquired using an absolute measurement system, Vicon for the tests at indoor and GPS in outdoor scenarios. When possible, GPS readings were corrected with a base station for more accurate ground truth, applying Real Time Kinematic (RTK).

Joint measurements and inertial readings are sensed quantities in equation (4.6). The slip vector $\dot{\boldsymbol{\varepsilon}}$ and contact point angles $\dot{\boldsymbol{\delta}}$ are non-sensed values. Non-sensed or unknown quantities of the vector $\dot{\boldsymbol{u}}$ are $\dot{x}$, $\dot{y}$ and $\dot{z}$, which are the variables of interest for odometry. The slip vector $\dot{\boldsymbol{\varepsilon}}$ is simplified and modeled as only rotation along its $\hat{k}_{C_{il}}$ axis (ζ_{il}) since it is assumed that contact points slip with *non-holonomic* constraints. The rearrangement of the system of equations is obtained as in equation (4.6). The solution of the enhanced 3D odometry is computed by solving equation (4.7).

4.4.1 ExoMars Test Rover

The experiments consisted of two different odometry calculations, 3D odometry and skid odometry, Siciliano and Khatib (2016), using data information from encoders and the AHRS while driving on uneven terrains. A comparison with the simple contact point odometry is left for the Asguard robot, because the technique implemented in Schwendner et al. (2013) does not adapt to wheeled robots like ExoTeR. Their methodology does not apply the transformation approach for general motion models. The lack of modeling makes it impossible to adapt such contact point odometry to a robot moving on wheels. The only alternative would be to reformulate the method in Schwendner et al. (2013) using the transformation approach from Muir and Neuman (1986) which ultimately will result in the methodology presented here except for the quasi-static forces estimation.



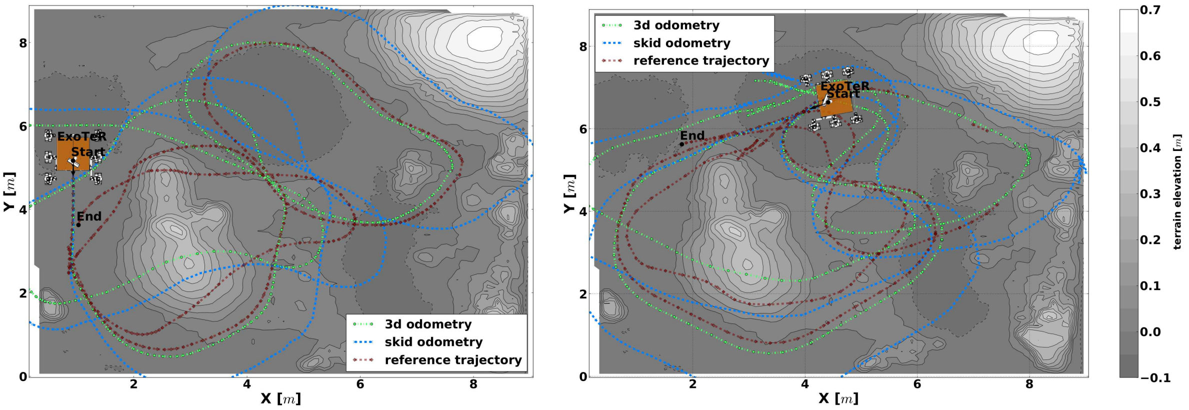
(a)

(b)

Figure 4.6: Test scenarios for ExoTeR. (a) Panoramic view of the ESA Planetary robotics laboratory. (b) Aerial picture of the Decos terrain with the test zone squared in black.

Objective. The purpose is to evaluate the performance of the odometry model presented in this work using the ExoTeR. The approach is compared with a classical skid odometry as the method described in Siciliano and Khatib (2016). Skid odometry is a planar kinematics approach and it does not model the full kinematics of the chassis. The skid odometry takes the rover wheelbase and track width with the wheel radius to compute the displacements. Instead, a 3D odometry model identifies each contact point and compute a weighted solution to estimate the delta pose. The delta pose is computed by minimizing a least squares error.

Setup. The tests with ExoTeR are conducted in two different locations, the Mars-like terrain at the Planetary Robotics laboratory and the Decos terrain, see Fig, 4.6. ESA's planetary robotics laboratory consists of, among other facilities, a $9\,\mathrm{m} \times 9\,\mathrm{m}$ Mars-like terrain that resembles a planetary surface. Mars-like terrain is equipped with a Vicon tracking system which can deduct and track position and orientation of objects. Decos terrain is located in Noordwijk, The Netherlands, in proximity to the European Space Research and Technology Center (ESTEC). It is a terrain imitating a rocky Mars environment with one prominent crater in the middle and a smaller ground depression next to it. Two experiments, *Test#1* and *Test#2*, are conducted at ESA's planetary laboratory to evaluate the feasibility of the approach and a single test in conducted at the outdoor Decos terrain. ExoTeR is commanded remotely to



(a) (b)

Figure 4.7: 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.

perform a figure eight trajectory connecting two consecutive circles. A bigger circle along the main crater and another circle bordering the ground depression. The skid and 3D odometry models run in two separated Rock [2] tasks onboard of ExoTeR's main computer.

Evaluation. Mars-like terrain. The odometry trajectories produced by the dead reckoning pose from both odometry models are shown in Fig. 4.7. They are compared with the ground truth from the Vicon system and resulting errors are given in Table 4.1. 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 root mean square error (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.

Evaluation. Decos terrain. 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. Fig. 4.8 shows the trajectory for the dead reckoning for both odometry models. Dead reckoning accumulates errors unbounded, but due to the double circle trajectory the maximum error does not appear at the end of the

[2] The Robot Construction Kit (Rock) http://www.rock-robotics.org

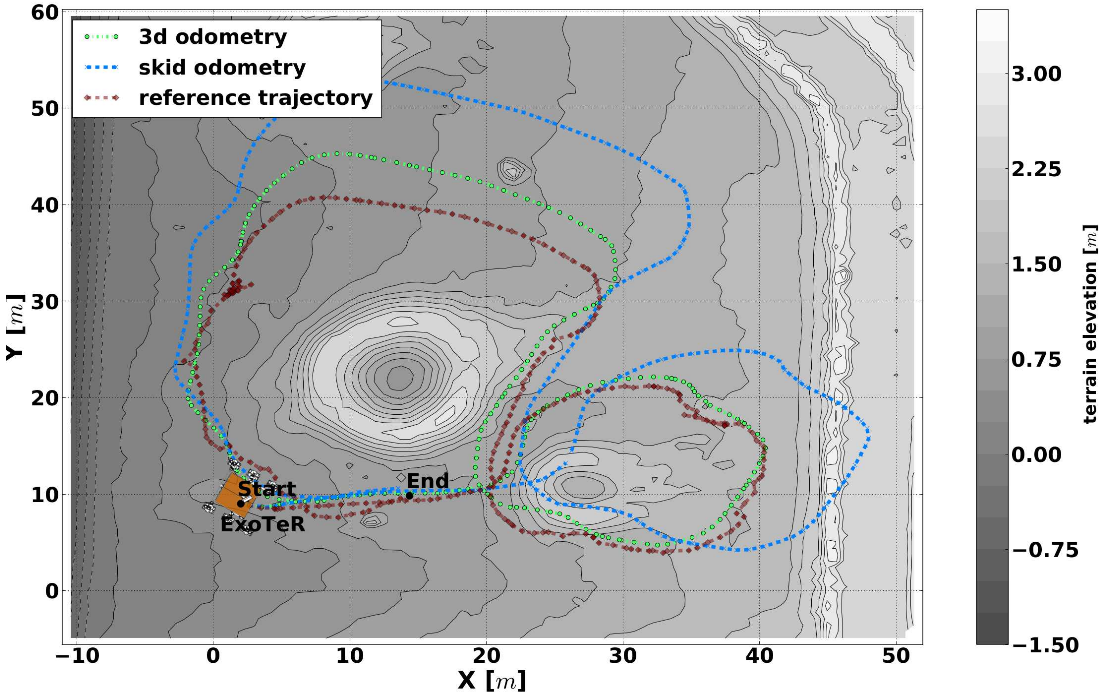


Figure 4.8: Resulting trajectories of the 3D odometry and skid odometry model with ExoTeR on the Decos terrain.

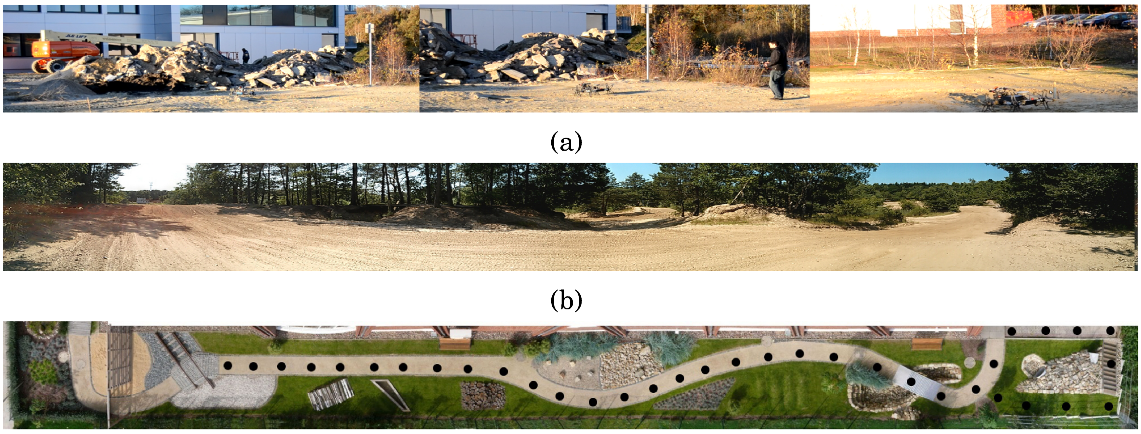
drive. Metric information is given in Table 4.2. Skid odometry is an extensively used odometry averaging left and right wheel velocities. This simplification entails inaccuracies in the pose and motivates the effort to fully model the chassis. The metrics show that a complete model of the chassis reduces the percentage error per distance traveled by half at the Decos test.

Table 4.1: 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

Table 4.2: 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



(a)

(b)

(c)

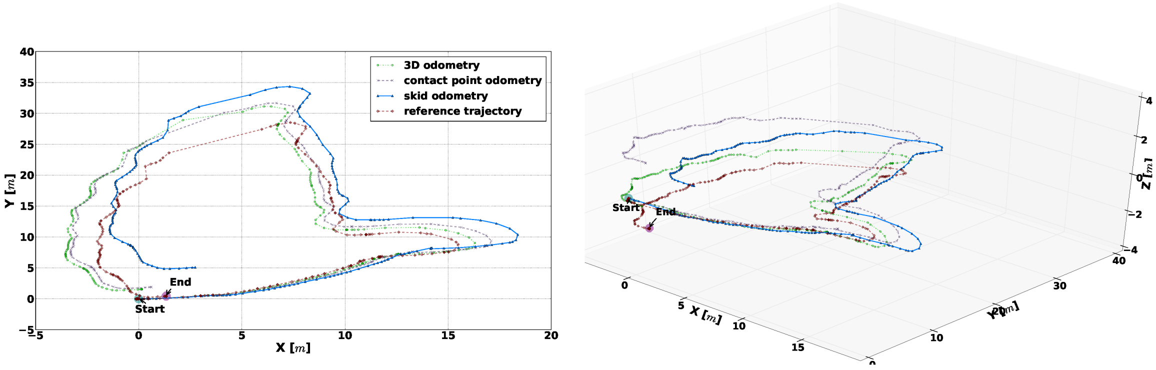
Figure 4.9: Test scenarios for the Asguard robot. (a) Pictures of the sandfield. (b) Panoramic view of the motocross track. (c) DFKI's outdoor test track with the trajectory depicted as a dotted line, negotiating the bridge and the stairs.

4.4.2 Asguard Robot

Objective. The purpose is to evaluate the performance of the enhanced 3D odometry model in Asguard. In addition to previous tests, the performance is evaluated with the contact point odometry from Schwendner et al. (2013).

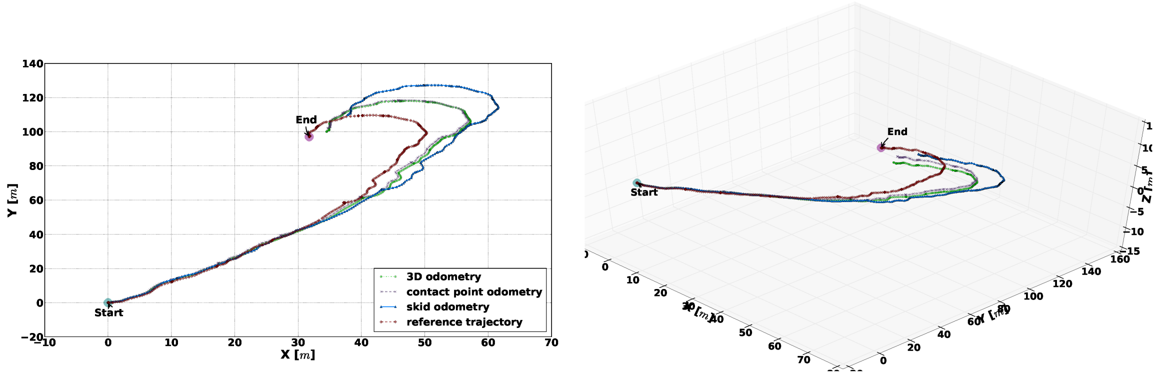
Setup. Field testing with a rich variety of uneven terrains, see Fig. 4.9, were selected to demonstrate the approach using the Asguard robot. The experiments consist of three different odometry calculations, skid odometry, contact point odometry and the proposed enhanced 3D odometry, using on board sensors readings from the encoders and the IMU. RTK, when available, was only used to post-process the accuracy of the test. The enhanced 3D odometry uses the Jacobian calculations and computes the weighting quasi-static forces. This is compared with a contact point odometry which equally computes the 3D position of the contact point using the presented technique but it combines contact points and calculates the resulting delta pose as the technique described in Schwendner et al. (2013). As in the previous Section, both techniques are compared with a state-of-the-art skid odometry which computes a skid-steer delta pose similar to the kinematic model described in Siciliano and Khatib (2016).

Evaluation. Fig. 4.10 and 4.11 depict qualitative results for the experiments in the sandfield and motocross respectively. As expected, skid odometry suffers from the lack of modeling while the 3D odometry estimates body motion more accurately. Though the contact point odometry improves a conventional skid-steer approach,



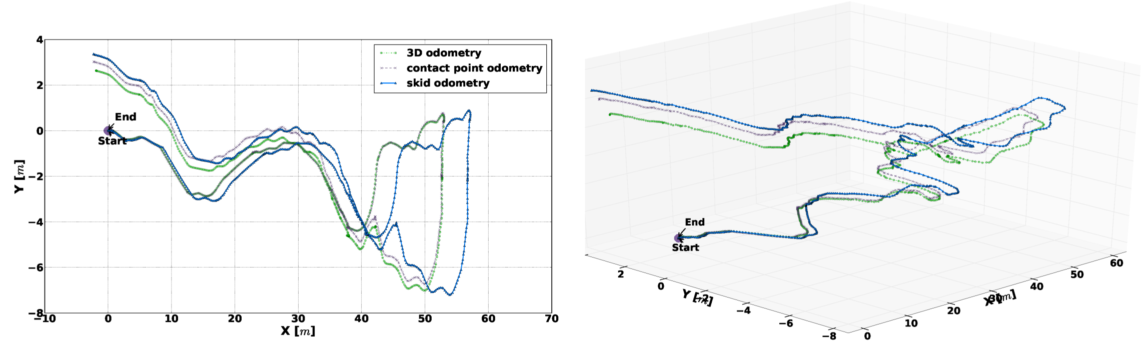
(a) (b)

Figure 4.10: Sandfield test trajectory. (a) Top view. (b) 3D view.



(a) (b)

Figure 4.11: Motocross test trajectory. (a) Top view. (b) 3D view.



(a) (b)

Figure 4.12: DFKI's test track trajectory. (a) Top view. (b) 3D view.

it still is negatively impacted by single wheel slippage and dragging. The enhanced 3D odometry better overcomes this negative effects by using least squares from equation (4.7). It is also appreciable in Fig. 4.12 that the pitch drift becomes

Table 4.3: Sandfield results for the different odometry models with Asguard

Odometry model	RMSE $[m]$			Final E.$[m]$			Final E. $[\%]$	Distance $[m]$
	x	y	z	x	y	z		
skid odometry	0.94	3.39	0.93	1.43	4.65	1.95	6.15	85
contact point odometry	0.89	1.59	2.14	0.73	1.47	3.18	4.21	85
3D Odometry.	0.84	1.08	0.39	1.18	1.10	0.78	2.11	85

dominant at DFKI's test track, however the impact in position is still less dominant for the enhanced 3D odometry. Table 4.3 and 4.4 show the Root Mean Square Error (RMSE) and final error values for the sandfield and motocross tests. These results show that the 3D odometry has a more accurate estimate.

The error is reduced because the robot kinematics is completely modeled instead of implementing a planar assumption, i.e. skid odometry, with less accurate estimate. The weighted 3D approach decreases the RMSE in all directions in the sandfield, see Table 4.3. Contact point odometry is more accurate than skid odometry for x-y direction but less accurate in the z direction. The reason is because the skid odometry only estimates translation velocities in the x-y directions, i.e. planar assumption. Therefore, only the drift in attitude influences the error in elevation. However, the error is significantly reduced in the 3D odometry by weighting the composite equations using quasi-static forces. It is important to mention that RTK was not available at the motocross test and only GPS signal was available. Conversely, as a matter of the Geometry Dilution of Precision (GDOP) in the GPS, the error in altitude is difficult to evaluate in the motocross test. Therefore, the error in z direction show in Table 4.4 appears difficult to analyze due to the poor quality of the measurements.

However, an accurate ground truth was acquired at the sandfield test where RTK

Table 4.4: Motocross track results for the different odometry models with Asguard

Odometry model	RMSE $[m]$			Final E.$[m]$			Final E. $[\%]$	Distance $[m]$
	x	y	z	x	y	z		
skid odometry	7.38	11.78	1.37	5.96	12.55	1.46	8.33	167
contact point odometry	4.05	5.64	1.44	2.83	4.78	1.82	3.48	167
3D odometry.	4.32	5.15	1.99	2.55	2.56	2.90	2.78	167

Table 4.5: DFKI's test track results for the different odometry models with Asguard

Odometry model	RMSE $[m]$			Final E.$[m]$			Final E. $[\%]$	Distance $[m]$
	x	y	z	x	y	z		
skid odometry	-	-	-	2.12	3.48	4.67	5.06	120
contact point odometry	-	-	-	2.10	3.13	4.42	4.74	120
3D odometry	-	-	-	1.74	2.72	3.95	4.17	120

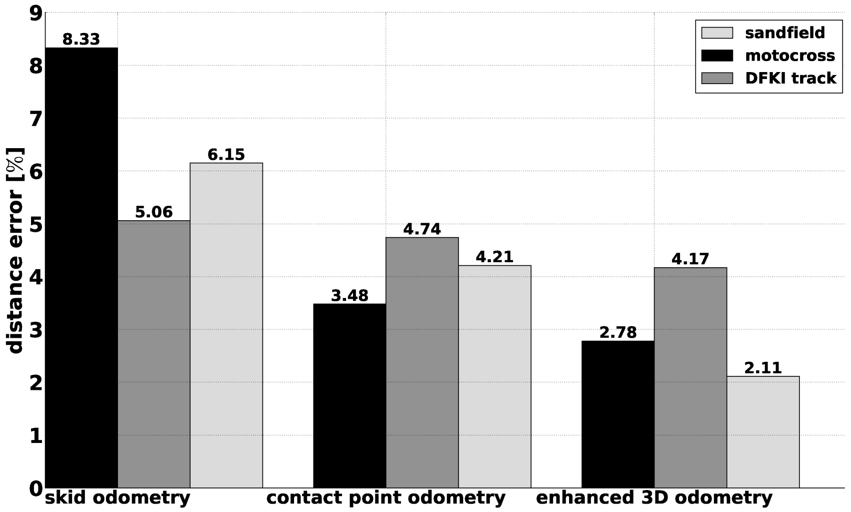


Figure 4.13: Percentage error of the distance traveled for each test with Asguard.

fixed signal was available along all the trajectory with centimeter accuracy. For the sandfield test the ground truth value for the z direction was stable, with a higher GDOP. This allows analysis of the error in elevation with more precision.

Finally, the results for the last test conducted at the DFKI's test track are shown in Table 4.5. Poor satellite signal was sensed due to the narrow view of the test track between buildings and as a consequence GPS is not available during this test. The results still show improvements for the 3D odometry at the final pose. The improvement is less significant than for the sandfield and motocross tests since the terrain was mainly flat. Nevertheless, when negotiating the bridge and climbing up the stairs the enhanced 3D odometry improves the localization performance, accumulating less error.

4.5 Conclusion

Motivated by the research during recent years at DFKI and seeking for accurate and more flexible kinematics modeling, the work presented in this chapter improves the state-of-the art of odometry solutions. The performance of the enhanced 3D odometry, which is a 6 DoF motion model, is compared with a classical skid odometry (a planar model) and a contact point odometry (a state-of-the art 3D model) in representative field testing datasets. The enhanced 3D odometry, while still simplistic in its modeling, brings adaptiveness to the solution and shows a reduction of the localization error on uneven terrains.

Fig. 4.13 shows the percentage error per distance traveled for each different technique and test scenario. The bar chart shows the percentage error for each test as

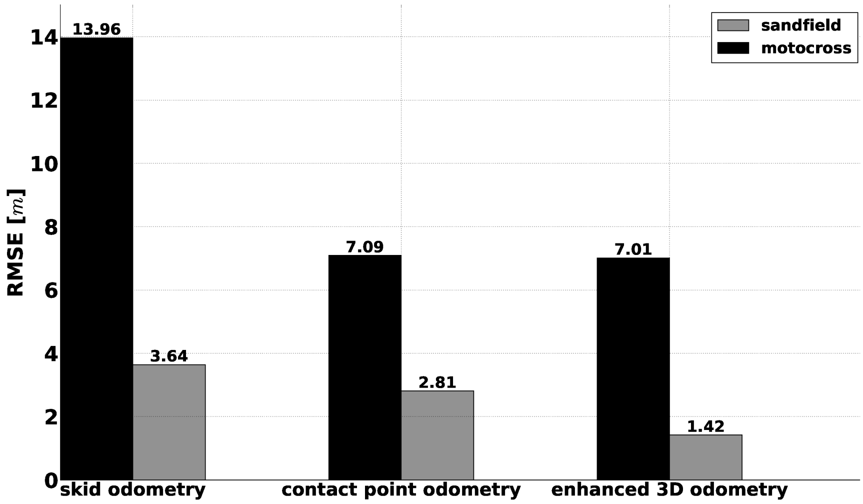


Figure 4.14: Comparison of the Root Mean Square Error (RMSE) for the sand field and motocross tests (i.e. GPS available tests) with the Asguard robot.

a function of the distance traveled. The skid odometry has 8.33 % final error for the motocross test with a total distance of 167 m. The contact point and enhanced 3D solution have 3.48 % and 2.78 % final error respectively for the same test. The sandfield test comprises a total distance of 85 m, and at this test, the final error decreases from 6.1 % for the skid odometry to 4.21 % for the contact point and 2.1 % for the enhanced 3D odometry. The improvement is less significant but still appreciable at the DFKI's test tack, with 4.17 % final error for a total distance traveled of 120 m. The error induced in pitch is dominant for DFKI's test track dataset due to climbing the stairs.

The RMSE comparison for the sandfield and motocross tests are shown in the bar chart of Fig. 4.14. The conclusion is that the RMSE significantly improves when a 3D odometry is developed. The error is more stable for the sandfield test since the RTK fixed signal improves the accuracy of the metrics, making the sandfield test the more reliable dataset for Asguard. The RMSE for the sandfield decreases from 3.64 % using skid odometry to 1.42 % using the enhanced 3D odometry. In addition, Table 4.2 emphasises that there is no doubt a six wheel rover with a passive link chassis, as a planetary rover, benefits from a complete 3D odometry model, decreasing the metrics of the error in all tests.

Independently of the metrics and accuracy, it is important to notice the value of the modeling by itself. A complete and adaptable model based on the transformation approach and Jacobian matrices bring flexibility to the design while *ad-hoc* solutions like the work in Lamon and Siegwart (2007) and Schwendner et al. (2013) make their technique still valuable but difficult to generalize to other robots. Pros and cons of developing a 6 DoF motion model for a 3D odometry are summarized at the end of the

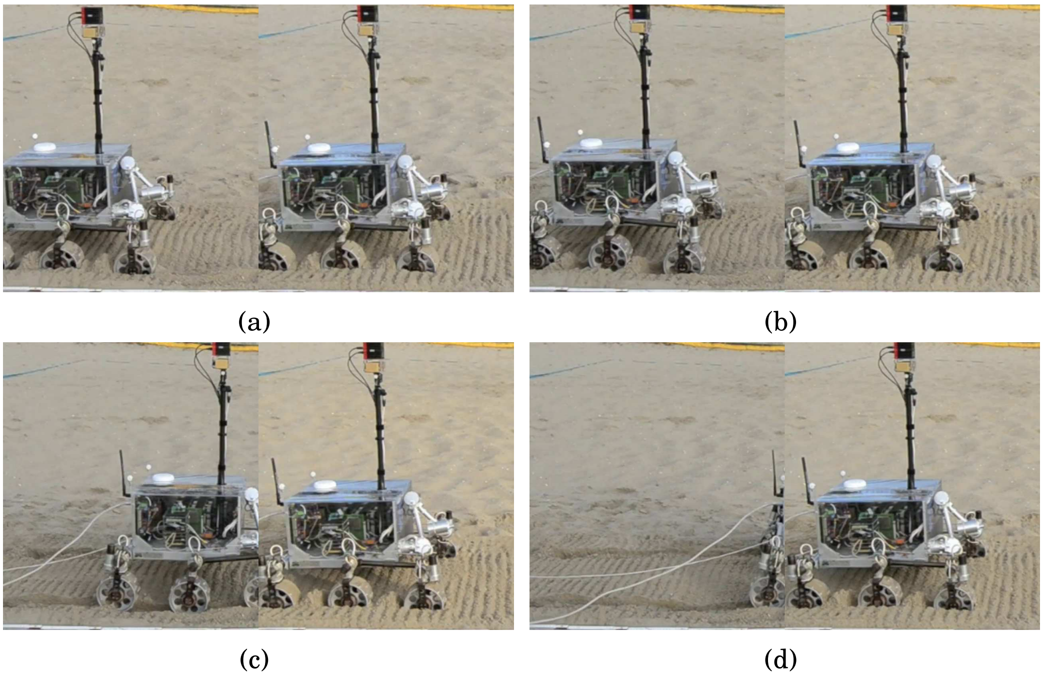


Figure 4.15: ExoTeR performing a wheel walking maneuver (left) versus normal driving (right) during an entrapment situation. (a) staring time, (b) situation after $15s$, (c) $30s$ and (d) $45s$. Figure from Azkarate et al. (2015).

document in Chapter 7.

In addition, the presented motion model can be inverted and used for controlling the robot. An accurate and complete robot control strategy at body level can only be achieved with a $6\,\mathrm{DoF}$ motion model. This is the case of the wheel walking [3], a set of sophisticated but very efficient locomotion strategies that optimally combine the movement induced at each joint to achieve a peristaltic motion. The development of a complete motion model is the guarantee of a correct and synchronized motion. Such advanced maneuvers are of great advantage during entrapment situations as shown in Fig. 4.15. The pictures show how ExoTeR did not escape from the entrapment situation using a normal driving while escaping in less than $45\,\mathrm{s}$ using wheel walking. Further development of the subject is available in Azkarate et al. (2015) and Wiese (2017).

[3] ExoTeR wheel walking video online: `https://youtu.be/qkOKzFq1SpY`

Part II

Simultaneous Localization and Mapping

Chapter 5: Gaussian Process Estimation of Odometry Errors

Hidalgo-Carrio, J., Hennes, D., Schwendner, J., and Kirchner, F. (2017). Gaussian Process Estimation of Odometry Errors for Localization and Mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore. IEEE

Chapter 6: Localization and Mapping with Adaptive Graph Sparsity

Hidalgo-Carrio, J., Poulakis, P., and Kirchner, F. (2018). Adaptive localization and mapping with application to planetary rovers. *Journal of Field Robotics*

Chapter 5

Gaussian Process Estimation of Odometry Errors

Since early in robotics, the performance of wheel odometry has been of constant research for mobile robots. This is due to its direct influence in localization. The pose error grows unbounded in dead reckoning systems and its uncertainty has negative impacts on localization and mapping (i.e. SLAM). The dead reckoning performance in terms of residuals, i.e. the difference between the expected and the real pose state, is related to the statistical error or uncertainty in probabilistic motion models. A novel approach to model wheel odometry errors using Gaussian processes (GPs) is introduced in this chapter. The methodology trains a GP on the residual between the nonlinear parametric motion model and the ground truth training data. The result is a GP over wheel odometry residuals which provides an expected value and its uncertainty in order to enhance the belief with respect to the parametric model. The localization and mapping benefits from a comprehensive GP model of the wheel odometry error. The approach is applied to a planetary rover in an unstructured environment. This work shows that the presented approach predicts wheel odometry errors towards the enhancement of a visual SLAM system.

5.1 Introduction

The use of proprioceptive and exteroceptive sensors to localize a robot and map its surroundings is a common practice in robotics. The technique is mathematically well established in robotics and has received special attention in GPS-denied environments, such as planetary robotics. The design of Inertial Navigation Systems (INS) have been investigated in Chapter 3 and the work shows the generality of the design approach in two different IMU technologies. Measuring and counting the joint displacement of the locomotion system gives an initial guess to derive the pose as described in Chapter 4. The work proves the flexibility of the transformation approach on modeling two different locomotion systems. Asguard and ExoTeR have versatile but different locomotion principles. Both techniques are combined in dead reckoning, Hidalgo-Carrio and Cordes (2012); Hidalgo-Carrio et al. (2014), and their understanding has been covered in Part I of this document. Consequently, Part II focuses on a higher level of abstraction by building SLAM on a previous solid understanding of dead reckoning processes. Experiments in Chapter 4, and Chapter 6 focuses on performing more tests with the planetary rover platform, ExoTeR.

SLAM optimizes dead reckoning methods in combination with landmarks in order to correct the robot state initially predicted by odometry. Three types of methods are typically applied: filtering, fixed-lag smoothing and full smoothing, Forster et al. (2015). They are slightly diverse in methodology but all use some degree of odometry in order to estimate an initial guess of the robot's pose. The importance of having a good guess of the odometry estimation is necessary to establish the selection of keyframes or nodes in case of a graph. This can be a difficult task since wheel odometry performance is not deterministic, due to its uncertainty from the interaction with the ground. Wheel odometry may perform well (i.e. $1-2\,\%$ error per distance traveled) in flat terrains but it can rapidly degrade on slippery terrains.

This chapter focuses on the methodology of training a GP on the wheel odometry residuals and its application to robot localization and mapping. A GP on the residuals estimates the quality of wheel odometry. The difference between the parametric model and ground truth serves to train a GP regression model. The purpose of the GP model is to serve in modern SLAM systems between the front-end (sensor data) and back-end (optimization) as depicted in Fig. 5.1. The contributions of this chapter are:

- To train a Gaussian process on wheel odometry residuals to model poor traction performance (i.e. odometry errors).
- To assess the kernel for the Gaussian process regression and therefore the hyperparameters.
- First investigations on the adaptive keyframes selection based on the GP re-

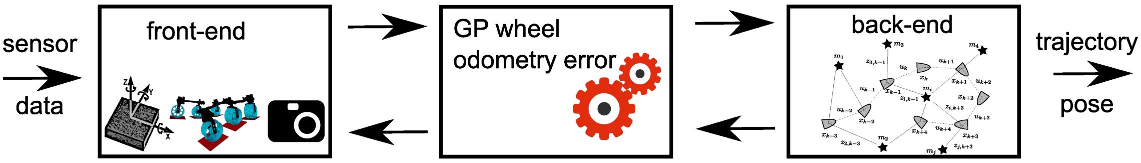


Figure 5.1: Anatomy of a modern SLAM system with the front-end, the back-end and the proposed GP regression model in between.

gression model.

The parametric model for computing the 3D odometry was introduced in Chapter 4. Section 5.2 reviews the related work in the context of odometry for localization and mapping and GPs applied in robotics. The Gaussian process regression is explained in Section 5.3. Afterwards, prediction quality of the learned process is presented in Section 5.4 in order to address the feasibility of the approach. The Gaussian process is used in a state-of-the-art visual simultaneous localization and mapping to estimate the robot pose. Experiments with a planetary rover in a representative environment are presented. Finally, Section 5.5 describes a conclusion about this part of the work.

5.2 Related Work

Odometry errors are investigated comprehensively in literature. Borenstein and Feng (1996b) investigate 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. Those systematic errors are also well characterized in the literature, Borenstein and Feng (1996a); Ojeda and Borenstein (2004). However, non-systematic errors are complex, difficult to predict and not possible to fully correct unless other perception means are present. Slippage has been the main non-systematic 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. (2004a) and Rehder et al. (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 introduce a linearised function to relate electric current and wheel-terrain interaction, Ojeda et al. (2006). Rogers-Marcovitz et al. (2012) presents 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.

Wang (1988) studies the uncertainty modeling and analysis for wheel odometry. Their solution is motivated by the lack of any exact method for calculating the covariance matrix. Their research finds a good approximate covariance matrix for the location estimator, limited to two-wheeled systems in planar environments. Basically, their work presents an adjustment factor to approximate the covariance matrix. Lamon and Siegwart (2007) associate odometry covariance with body acceleration. Kelly (2004) studies linearised techniques for error propagation. Their work is concerned with the nonlinear problem of propagating the uncertainty from a moving body frame to a fixed world frame, but it does not focus on the uncertainty modeling. Schwendner et al. (2013) defines a fixed configuration matrix for the odometry model in the novel *Embodied* localization and mapping technique, see Chapter 2 in Schwendner (2013). Learning methods are applied in Abbeel et al. (2005) to estimate the covariance of Kalman filters. A more sophisticated approach is presented in Kummerle et al. (2011a). Their research simultaneously calibrates the kinematic parameters into SLAM in order to adapt to different robot configurations. Uncertainty estimation is also investigated for aerial robots in Muller et al. (2012). A probabilistic velocity estimation uses flow sensors in autonomous airships. The technique fuses the information in a particle filter, with approximately 500 particles. The identification of process noise parameters of process models for mini drones is also discussed in Bry et al. (2012). Their method projects the process noise forward over multiple time steps. The process model only includes inertial sensors, which are sufficient for flying robots but not directly applicable to ground vehicles.

Machine learning is used in SLAM to increase the robustness and it seems just the beginning of a new perception era. For example, an EKF-SLAM with machine learning image processing is described in Casarrubias-Vargas et al. (2010) to increase robustness. GPs provide a probabilistic approach for learning kernel machines, producing promising results in robotics, but are not exploited further. The probabilistic nature of GPs makes them attractive to integrate with Bayes-based approaches, either Kalman-based or smoothers. Ko et al. (2007) and Ko and Fox (2008) apply 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 (GP-UKF). In general, GPs have several advantages for robotics since they are a practical tool for solving a diverse set of problems. Mukadam et al. (2016) applies 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 GP for mapping is described in Wang and Englot (2016). The technique collects sensor observations and estimates the occupancy map using an octree. Peretroukhin et al. (2016) use Gaussian kernels to predict visual feature uncertainty. Their work shows the application of generalized kernel estimation to improve the uncorrelated and static Gaussian error models in stereo visual odometry. As a result, the feature tracking estimation and robustness improve in localization by inferring a more accurate error model.

GPs have been used for nonlinear regression, but to the best of the author's knowledge their application to modeling the odometry error as residuals between the parametric model and a realistic wheel odometry output has not yet been addressed in the literature. This chapter introduces a novel manner to model wheel odometry errors based on the interaction with the ground.

5.3 Gaussian Processes for 3D Odometry

Odometry accuracy 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. This section describes the application of GPs to model a nonlinear regression function between the parametric model and the real odometry output.

5.3.1 Gaussian Processes

GPs are a powerful, non-parametric tool for learning regression functions from sample data. GPs are flexible, work nicely with missing and noisy data and therefore are very practical for solving 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 and Williams (2006). However, an infinite vector is impractical because computer memory is finite. Marginal likelihood or marginalization allows work in statistics with a finite subset without losing generality. Fig. 5.2 shows the train and test schematics of the odometry error model.

Assume we have a training set of data, $D =< X, Y >$, where $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_n]$ is a matrix containing d-dimensional input examples $\boldsymbol{x_i}$ and $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n]$ 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

$$\boldsymbol{y}_i = f(\boldsymbol{x}_i) + \boldsymbol{\epsilon} \tag{5.1}$$

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

$$p(Y|X) = \mathcal{N}(0, K(X, X) + \Sigma) \tag{5.2}$$

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

$$\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{k}_{rbf}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')W(\boldsymbol{x}-\boldsymbol{x}')^\intercal} \tag{5.3}$$

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 for all possible inputs in the

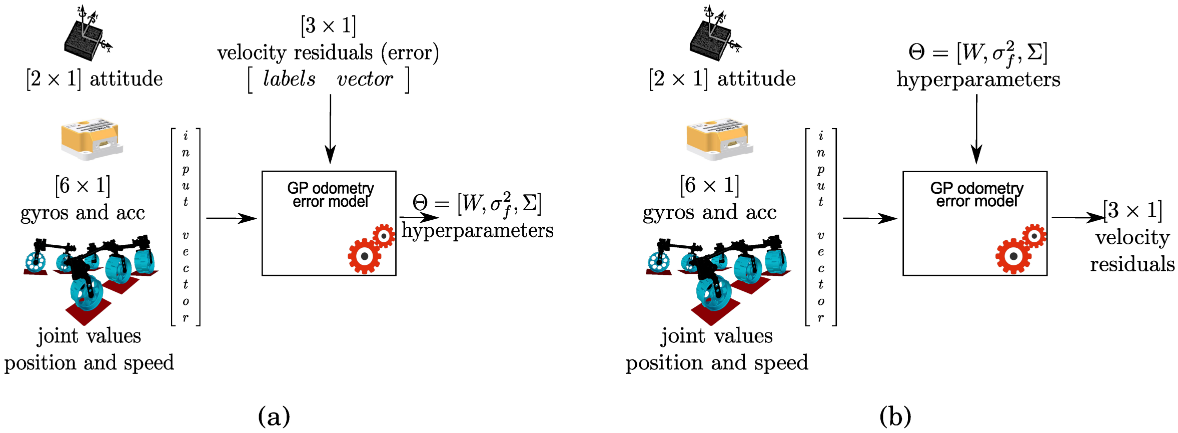


Figure 5.2: Gaussian Process diagrams for odometry error model in planetary rovers (a) learning (b) prediction.

future. The GP is trained with a selected training data set $D =< X, Y >$. The GP defines a predictive distribution over the output Y_* at arbitrary prediction points X_*. The mean function is

$$GP_\mu(X_*, D) = \boldsymbol{k}_*^\intercal [K + \Sigma]^{-1} Y \tag{5.4}$$

and variance

$$GP_\Sigma(X_*, D) = \boldsymbol{k}(X_*, X_*) - \boldsymbol{k}_*^\intercal [K + \Sigma]^{-1} \boldsymbol{k}_* \tag{5.5}$$

where $\boldsymbol{k}_*$ is the vector defined by the kernel values between X_* and the training input X as $\boldsymbol{k}(X, X_*)$ 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 semi-positive 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 learnt by maximizing the log likelihood of the training outputs given the inputs,

$$\boldsymbol{\Theta}_{max} = \arg\max\{\log(p(Y|X, \boldsymbol{\Theta}))\} \tag{5.6}$$

where the log term can be expressed as

$$\log(p(Y|X)) = -\frac{1}{2} Y^\intercal (K(X,X) + \Sigma)^{-1} Y - \frac{1}{2} \log \|K(X,X) + \Sigma\| - \frac{n}{2} \log 2\pi \tag{5.7}$$

5.3.2 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

$$\boldsymbol{s}(k+1) = \boldsymbol{s}(k) + g(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k)) \tag{5.8}$$

where k is the time index and g is the function which describes 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). A Gaussian process can be used to learn the dynamic process described by the function g. The result will be a GP which predicts the delta between two consecutive 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 Gaussian process assumes a zero-mean function in equation (5.2) and the robot odometry is clearly not a zero-mean. This is related to the modeling, which appears 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. This is because the residual or odometry error is the value of interest and has a mean close to zero in the ideal case.

5.3.3 Odometry Residuals from Gaussian Processes

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

$$\boldsymbol{s}(k+1) = \boldsymbol{s}(k) + g(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k)) + f(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k)) \tag{5.9}$$

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 which 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 $\boldsymbol{y}_k = \boldsymbol{s}(k+1) - \boldsymbol{s}(k) - g(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k))$.

5.4 Experimental Results

As in previous experiments, the ExoMars Test Rover (ExoTeR) a laboratory rover prototype that resembles in scale the ExoMars rover, is used to evaluate the methodology Azkarate et al. (2015); Poulakis et al. (2015). ExoTeR's sensor suite includes a stereo camera pair, an Inertial Measurement Unit (IMU) and actuator encoders and potentiometers. The experiments take place on a Mars-like testbed in the Planetary Robotics Laboratory (PRL) of the European Space Research and Technology Centre (ESTEC) - the largest site of the European Space Agency (ESA). The testbed is a 9×9 m test area with different soil types as described in Chapter 4.

A set of reflective markers are located on the ExoTeR in order to track its position and orientation during the training experiments. The Vicon system captures the ground truth data. The accuracy of the system is around $1\,\mathrm{cm}$ in position and $0.2\,\mathrm{degree}$ in orientation depending on the number of cameras tracking the markers. ExoTeR is remotely driven in the testbed area describing all possible maneuvers, e.g. forward, backward, turn-on-spot and Ackermann, at widely different inclinations and terrain characteristics. Training data are collected by driving the rover at different speeds, possible maneuvers and a diverse set of terrain conditions such as ground, soft carpet, loose soil and rocky terrain, see Fig. 5.3. The absolute position and heading are eliminated from the training data. The input vector $\boldsymbol{x}_k$ is composed by pitch and roll orientation angles, joints position and speed, angular velocities sensed by gyroscopes and linear accelerations from the accelerometers. The error in robot's linear velocities is the training vector of outputs $\boldsymbol{y}_k$. The Vicon system delivers rover position at a frequency of $100\,\mathrm{Hz}$. Training data are calculated from the rover's linear velocities by low-pass filtering the delta position readings from the Vicon system. The hyperparameters of the Gaussian process are optimized using conjugated gradient optimization to solve equation (5.6), i.e. the Levenberg–Marquardt algorithm, Rasmussen and Williams (2006). The GP estimates a three dimensional output vector $\boldsymbol{y}_*$ of the velocity error in each direction of motion $[\mu_{\dot{x}_{\hat{i}}}, \mu_{\dot{y}_{\hat{j}}}, \mu_{\dot{z}_{\hat{k}}}]$ (odometry error). A different set of collected data, the test data, is used to evaluate the accuracy of the estimated residuals.

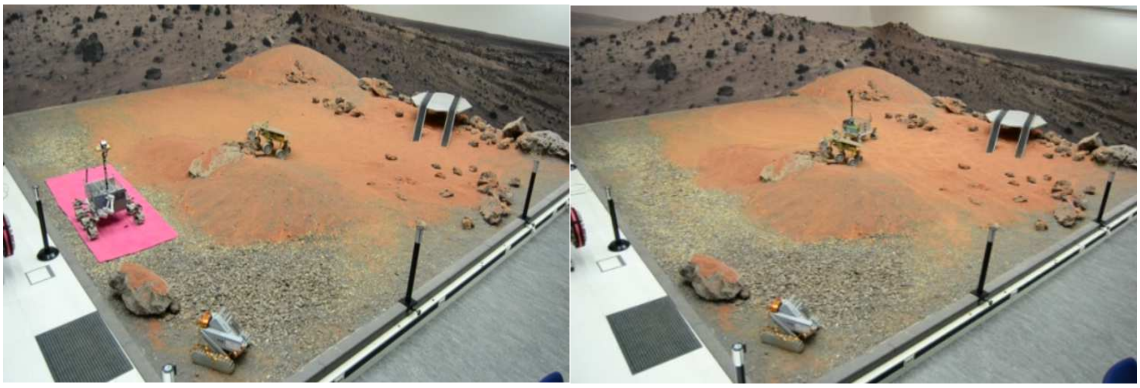


Figure 5.3: Photographs during the tests for collecting training data with ExoTeR on the Mars-like terrain at ESA's planetary robotics laboratory. The training data are used to learn the kernel function of the Gaussian Process nonlinear regression.

Table 5.1: Error per kernel evaluated with the test data.

Kernel	RMSE$[m/s]$	MAE$[m/s]$
rbf: $[k_{rbf}]$	**0.004494**	**0.002127**
rbf + linear: $[\breve{k}_{rbf}]$	0.004909	0.002588
matern 5/2: $[k_{m52}]$	0.005562	0.002635
matern 5/2 + linear: $[\breve{k}_{m52}]$	0.004909	0.002588

The Gaussian process is trained offline using GPy (2012). First, several kernels are evaluated in order to select the kernel which best predicts the regression model. The results are described in Section 5.4.1 using some test data from the Mars-like terrain. The test data are a subset of the training data not used during the GP learning process.

5.4.1 Kernel Selection

Three kernels, linear, squared exponential and Matern 5/2, Rasmussen and Williams (2006), are trained to determine which combination of kernels fits the regression more accurately. The kernels are separated and combined as $\breve{k}_{ker} = k_{ker} + k_{lin}$ where k_{ker} is a squared exponential (rbf) or Matern 5/2 kernel and

$$k_{lin}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_l^2 \boldsymbol{x} \cdot \boldsymbol{x}'. \tag{5.10}$$

Objective. The purpose is to select the kernel function that better fits the regression model among a set of different kernel combinations.

Setup. The kernels are fitted with normalized training data. This enforces feature scaling, making gradient descent converge more easily. Four training tests were conducted in a representative scenario. Fig. 5.3 shows pictures of the filmed laboratory setup. Test data, not included in the training data, are used to verify the prediction quality of the Gaussian process odometry residuals.

Evaluation. Table 5.1 shows the error metrics for each of the kernels. The squared exponential fits the residuals with lower root mean square error (RMSE) and mean absolute error (MAE) than the Matern 5/2. In addition, the metrics shows that the linear kernel k_{lin} does not improve the estimates of the squared exponential or Matern5/2. The square exponential kernel is selected for the rest of experiments in the thesis.

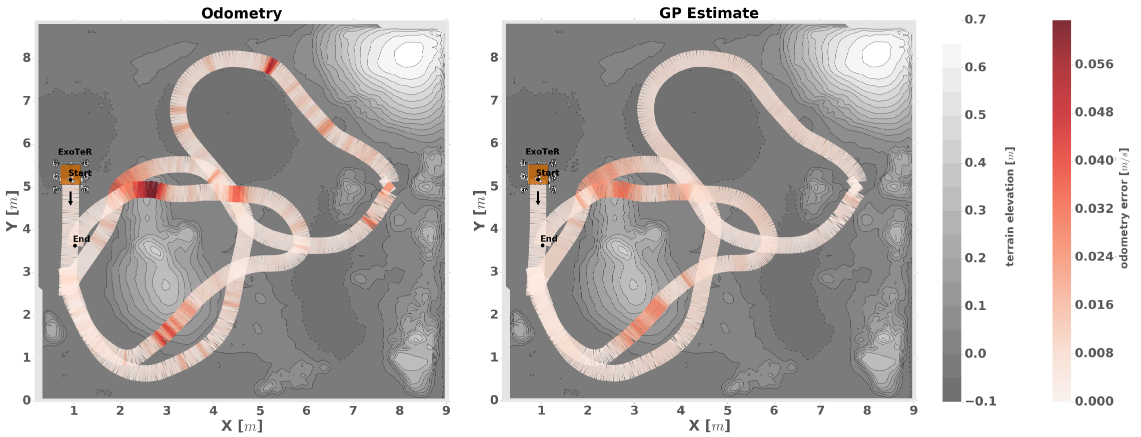


Figure 5.4: Ground truth odometry residual (left) and GP estimate (right) for Test#1. Traversed trajectory and Digital Elevation Map (DEM) of the Mars-like testbed are depicted together with the odometry error (red color-bar).

5.4.2 Prediction Quality

Once the hyperparameters of the selected GP kernel are learned, the prediction is made online on the rover based on sensed current inputs x_*. The GP prediction and the localization and mapping are executed on board the ExoTeR rover running the Rock real-time framework [1]. Two experiments, *Test#1* and *Test#2*, are conducted at ESA's planetary laboratory to evaluate the prediction quality of the GP regression model. The results are described here.

Objective. The goal is to evaluate the accuracy performance of the GP regression model, Test#1 and Test#2, with respect to the ground truth odometry error.

Setup. ExoTeR is remotely commanded to navigate in the Mars-like terrain. ExoTeR's trajectory is tracked by the Vicon system to collect ground truth data.

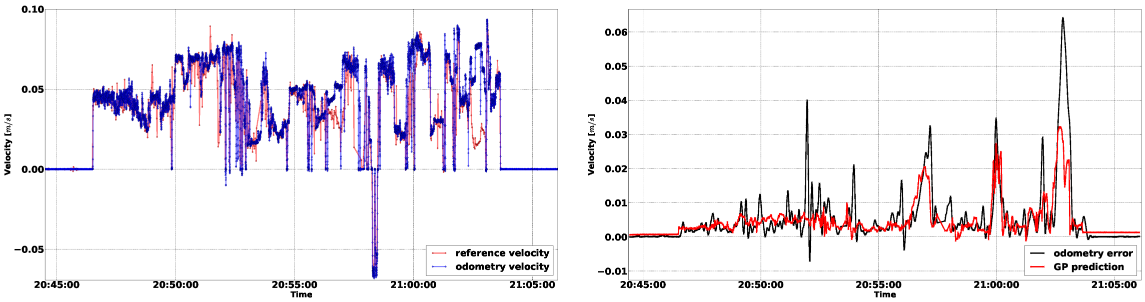
Evaluation. GP predictions are compared with the *true* odometry error and depicted in Fig. 5.4 for Test#1 and Fig. 5.6 for Test#2. The *true* odometry error is computed by comparing the delta poses from the 3D odometry model and the ground truth data from the Vicon system. The trajectory is depicted with the contour map to facilitate the interpretation of the results.

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 predicts the errors in odometry.

[1] The Robot Construction Kit (Rock) http://www.rock-robotics.org

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 Fig. 5.4 (at time $21\,\mathrm{h}$ and $3\,\mathrm{min}$ in Fig. 5.5). The rover almost encounters $100\,\%$ slippage (no forward movement) at that location. ExoTeR experiences a similar behaviour 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 Fig. 5.6 (at time $18\,\mathrm{h}$ and $10\,\mathrm{min}$ in Fig. 5.7). 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 not only dependent 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 in order to predict the non-parametric model of the error, resulting in different predictions, even when driving at the same spot.

Fig. 5.5a compares the odometry velocity with the ground truth velocity and Fig. 5.5b shows the truth and predicted odometry error in a single graph with respect to time. The predicted odometry error follows the truth odometry error in most parts of the experiment. The GP model predicts a maximum of $70\,\%$ slippage which corresponds to $4\,\mathrm{cm/s}$ odometry error when driving over the sandy dune area previously mentioned for Test#1, see Fig. 5.5b at time $21\,\mathrm{h}$ and $3\,\mathrm{min}$. Most of the GP modeling occurs in the covariance function described in equation (5.5) which tends the predictions towards a zero mean function. This is because GPs assume a zero mean prior in equation (5.2) and the mean function $GP_{\mu}(X_{*}, D)$ tends towards 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 transi-



(a)

(b)

Figure 5.5: Test#1 results for ExoTeR (a) odometry velocity and ground truth (b) *truth* and GP estimated error.

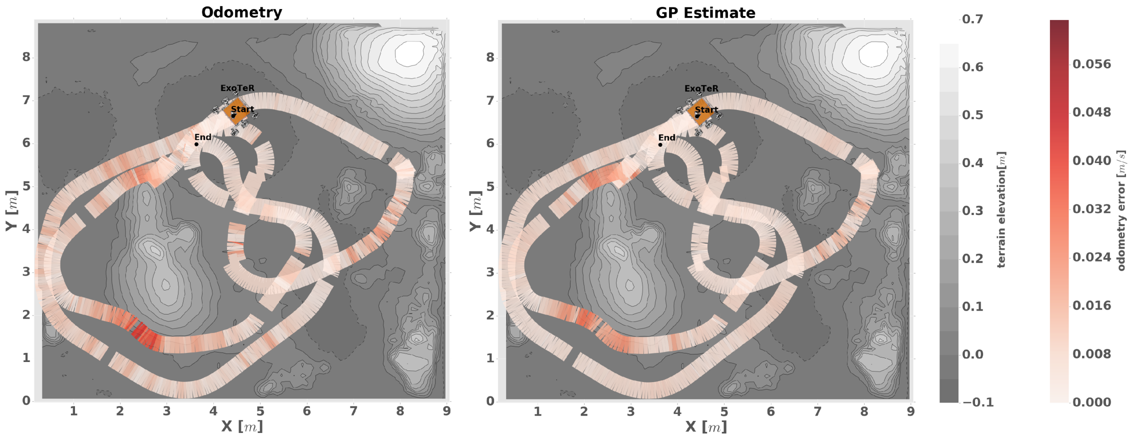
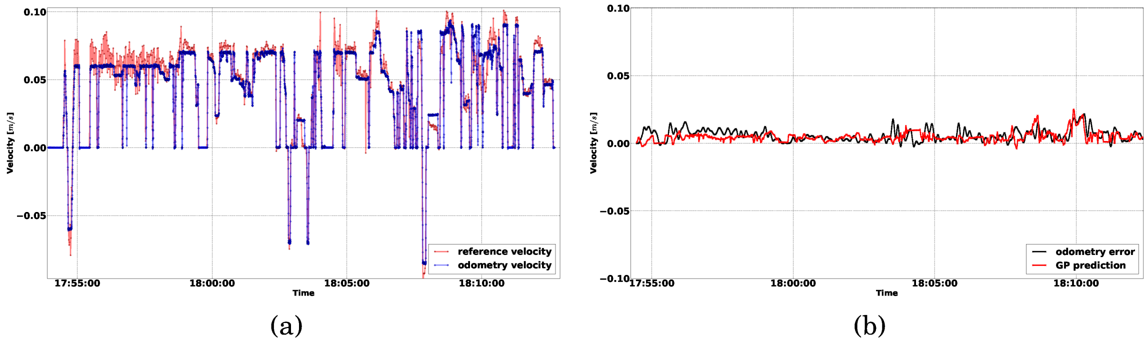


Figure 5.6: Ground truth odometry residual (left) and GP estimate (right) for Test#2. Traversed trajectory and Digital Elevation Map (DEM) of the Mars-like testbed are depicted together with the odometry error (red color-bar).

tions in slippage, which are more valuable than the error number. It is actually the most important aspect since the GP model adapts the SLAM solution by detecting slippage and not correcting it. Design of an adaptive SLAM system is discussed in Chapter 6. Nevertheless, a first investigation with a state-of-the-art visual SLAM is presented in the following.

5.4.3 Application to Localization and Mapping

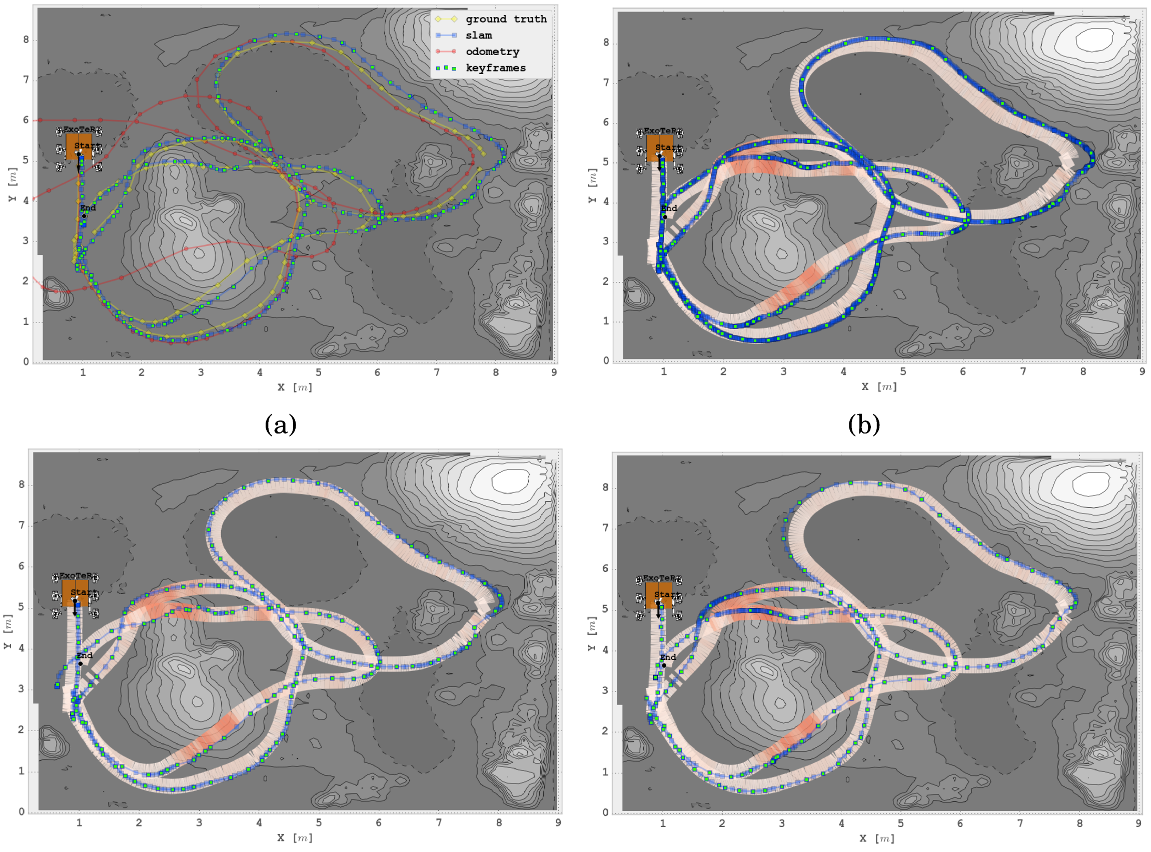
Wheel odometry performs as well as visual odometry when residuals are small. Residuals provide information about the wheel odometry error and helps to identify when the visual odometry is required. The wheel odometry can equally support feature tracking by providing an initial guess between two consecutive image frames, e.g. constant velocity model.



(a)

(b)

Figure 5.7: Test#2 results for ExoTeR (a) odometry velocity and ground truth (b) *truth* and GP estimated error.



(a) (b)

(c) (d)

Figure 5.8: SLAM results: (a) test experiments and trajectories comparison. (b) shows the distribution of images and keyframes without GP (c) still computes more image frames with worse performance than (d) which uses the GP odometry error model.

Objective. The purpose of this preliminary test is to evaluate the benefit of using GPs in SLAM, as a motivation for Chapter 6. It is expected to have similar or better performance than state-of-the-art SLAM but with fewer of nodes in the graph. It also involves to compute visual odometry only when necessary without losing the track of features. Ideally, the number of losses in camera poses would be reduced and robot relocalization strategies would not be required as often as when only using visual means.

Setup. The test is conducted at the Mars-like testbed of ESA's PRL with the same setup as in previous experiments. The experiment is conducted using the well-known ORB-SLAM, Mur-Artal et al. (2015). ORB-SLAM is robust against outliers, accurate and uses ORB binary features descriptors.

Table 5.2: ExoTeR's pose results for the different SLAM schemes.

Scheme	#Frames	#Keyframes	RMSE $[m]$	Max E.$[m]$	Final E.$[m]$	Final E. $[\%]$	Distance $[m]$
SLAM 2.5fps	2582	182	0.144	0.455	0.204	0.48	42.35
SLAM 0.5fps	**500**	**150**	0.204	0.729	**0.520**	**1.22**	42.35
SLAM with GP	**484**	**135**	0.145	0.468	**0.264**	**0.62**	42.35

Evaluation. The information gained by the GP is used to inform the localization and mapping system to compute visual odometry only when necessary. It furthermore reduces the keyframes in the optimization back-end. This technique applies to any full SLAM approach. Table 5.2 shows the results of calculating ExoTeR's pose running three different SLAM schemes for a test data trajectory. *SLAM 2.5 fps* computes the localization and mapping by processing a new image frame at a rate of 2.5 fps. *SLAM 0.5fps* computes the localization and mapping with a lower frequency of 0.5 fps. Without the GP prediction the image frames are equally distributed along the trajectory, which is not a desired feature since the wheel odometry performs accurately except for highly slippery areas. *SLAM with GP* is the modified approach using the GP between the front-end and the back-end to select the keyframes and frame frequency accordingly, see the schematic in Fig.5.1. The GP provides extra information to the SLAM and allows unequally distributing the image frames along the path, having better performance with the same computational effort. It means that with the same size of graph and number of processed image frames the *SLAM with GP* is superior in accuracy. Fig. 5.8 visually depicts the trajectory for the evaluated approaches together with the map from the testbed. The number of frames and keyframes can be seen along the path with the total traversal of 42.35 m. *SLAM with GP* adapts to the wheel odometry reducing the number of image frames in the visual odometry. The *SLAM with GP* approach computes five times fewer of frames without a significant penalty in accuracy, see Table 5.2, and augmenting graph sparsity.

5.5 Conclusion

This formulation describes the first insight into using GPs to model odometry errors and its first application in localization and mapping. This work is the first demonstration that wheel odometry errors, slippage and poor traction performance can be predicted by training a GP using robot propioceptive data. The outcome entails a significant result because the expected wheel odometry error is finally understood and the localization and mapping system can adapt to it. The chapter focuses on modeling the odometry error, selecting the kernel function, identifying the hyperparameters, training the GP regression model and demonstrating the importance of keyframes selection in SLAM.

GPs, as Support Vector Machines (SVMs), use kernel functions which satisfy the Mercer's Theorem, see Chapter 4 in Rasmussen and Williams (2006) for further reading. The selected kernel is the RBF function due to slightly more accurate performance that Matern $5/2$. The test results also show that a linear kernel does not improve the accuracy of the regression model (see Table 5.1), showing the nonlinearity of the problem. It is worthwhile to mention that the presented GP training uses feature scaling which accelerates the gradient optimization to solve in equation (5.6). This is due to the wide range in scale of different features in data, e.g. acceleration vs joints position. Finally, the methodology has been demonstrated using a planetary rover navigating on an unstructured environment and loose terrain. The GP information is used to selectively identify the image keyframe in visual odometry and preintegrate the delta displacement given by the wheel odometry.

Though simple, this chapter shows the first results towards a more efficient computation of frames and keyframes and serves as an introduction to further develop the adaptive SLAM in Chapter 6. Until now, the adaptation rate for image frames processing and selection of keyframes is set as fixed parameters. Following methods describe a SLAM architecture with dense map reconstruction, loop closure, adaptive visual odometry and selection of highly informative keyframes. Localization and mapping with adaptive graph sparsity is developed in next, based on navigation performance, adaptive criteria and further analysis.

Chapter 6

Localization and Mapping with Adaptive Graph Sparsity

The Gaussian Process (GP) complements the parametric solution by adding an estimation of the odometry error. The information gained by the non-parametric GP model serves to adapt the localization and mapping solution to the current navigation demands. An adaptive strategy is selected to adjust the image frame rate (active perception) and to influence the optimization back-end by including highly informative keyframes in the graph (adaptive information gain). Following this strategy, the resources adapt to the navigation demands providing an adaptive SLAM driven by the navigation performance and conditions of the interaction with the terrain. The proposed methodology is experimentally verified on the ExoTeR, a representative planetary rover, under a realistic field test scenario. The results examine the benefits of the approach, bring adaptive sparsity to the graph and provide insightful aspects towards a long-term solution.

6.1 Introduction

SLAM optimizes a robot's pose and builds a consistent representation of the environment. A globally consistent map is required to locate and store useful samples as well as to return to the lander. SLAM provides the framework to study efficient sensor fusion (e.g. full smoothers) and interprets the environment as a whole, providing loop closures. Without SLAM the rover interprets the world as an *infinite corridor* making long-term autonomous systems inefficient with partly human intervention. Future rovers require data fusion solutions (e.g. SLAM) to deduce rover position and orientation in a prolonged, optimal and adaptive manner. Mars missions have demonstrated that the geometric and non-geometric hazards could stop the motion of a rover. Those potential hazards are difficult to detect remotely from Earth, calling for an on board solution, Maimone et al. (2007b). In this aspect, the future of SLAM will combine machine learning techniques with optimization approaches and hardware improvements (e.g. FPGAs and GPUs) in order to enable more reliable and fast surface navigation.

A typical architecture for SLAM has two parts: the *front-end* and the *back-end*. The front-end perceives the environment to translate data into models in order to estimate the state. The back-end uses the models to perform Maximum a Posteriori (MAP) estimation and inference. The SLAM community often identifies robotics as the application to hinder advances in SLAM since robots are complex systems to work with. However, robotics benefits from SLAM and SLAM benefits from robotics. Mobile robots are mobile perception systems which might enrich the SLAM solution. The information collected by the platform can be used to complement the typical visual SLAM. In summary, robotics helps to tackle the following open issues in SLAM.

- Performance. Wheeled mobile robots are in permanent contact with the terrain through the locomotion system. The information acquired from the interaction with the ground can be modeled using machine learning techniques in order to brings high levels of understanding. Robotics can benefit SLAM.

- Resources. Robots are embedded systems with limited computational resources. SLAM is limited by perception means and the computational load available onboard. Active perception in robotics provide means to adjust the computational load according to the navigation demands.

- Inference. The connection of SLAM with other systems, e.g. path planning or trajectory control, produces more useful information. The information might be used for risk-awareness and operations.

- Testing. Robotics brings the possibility for SLAM to leave the laboratory environment, collect enormous amounts of real data and test the approach under demanding situations. Robotics provides the platforms to connect perception with control actions.

This thesis proposes several techniques in accordance with the research goals described in Chapter 1. The techniques developed in this thesis are integrated in an adaptive SLAM and presented in this chapter as the following.

- The enhanced 3D odometry model developed in Chapter 4 is integrated in SLAM. The reason is that indoor robotics, which traditionally operates in a structured or planar environment, has brought inefficient and simple techniques to the field of odometry in outdoor robotics. The estimation of delta displacement through complete motion models produces a more accurate a priori estimate.

- Machine learning in SLAM. The GP regression model in Chapter 5 is used as a non-parametric error model. The model identifies hazardous areas by learning Bayesian kernels from the previous experiences.

- Adaptive data association. The SLAM is influenced by the odometry error allowing adaptivity to the solution, i.e. adaptive localization and mapping. The design, development and verification testing of the adaptive SLAM is the subject of this chapter.

The chapter is organized as follows. A dedicated review of SLAM systems is addressed in Section 6.2 with special attention in graph sparsity and data association. Section 6.3 gives an overview and describes the adaptive SLAM system. Field testing experiments with the ExoTeR planetary rover under representative environment conditions are shown in Section 6.4. The potential benefit of the adaptive SLAM on a real planetary mission is described in Section 6.5. Conclusions are discussed in brief at the end of this chapter.

6.2 Related Work

Originally, SLAM was developed by using filtering approaches. Filtering provides an efficient estimation of the latest state, Civera et al. (2008). However, Kalman-based or particle filter SLAM becomes intractable in real-time applications. EKF complexity grows quadratic with the number of features. An alternative is presented in Mourikis et al. (2007a) with an augmented-state EKF. Their 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, called *stochastic cloning*. The solution has a computational cost of cubic complexity with the number of states and is capable of accurately estimate the pose. The cloning does not affect the real-time behaviour 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 very easily with irreversible results. Such inconsistency in the filtering estimators entails to develop an observability analysis in vision inertial navigation systems, Hesch et al. (2014). The analysis provides modification of the measurement Jacobian matrix in the observable direction of the system as described in Dong-Si and Mourikis (2011).

The state dimension of a particle filter requires an exponential increase of the number of particles. Research in particle filtering SLAM was introduced by Montemerlo et al. (2003) to handle the nonlinearity of the SLAM problem. The high dimensionality makes particle filters intractable for real-time applications. The Rao-Blackwellization variant reduces the sample space, making it more suitable for mobile robotics and real-time constraints. Nevertheless, each particle still carries an estimation of the environment making the approach difficult to scale up for long-term applications.

Smoothing is an alternative to filtering. Fixed-lag smoothers estimate the state within a window of measurements. Smoothing approaches relinearise past measurements providing more accurate estimates. Smoothers are more robust in case of outliers, which make them suitable for long-term localization, Mourikis and Roumeliotis (2008). However, the fixed-lag smoother needs to take care of consistency analysis and accumulated linearization errors as in filtering methods.

In order to overcome the limitation mentioned above, full smoothers estimate the complete robot trajectory and features as in Kummerle et al. (2011b). This is the full SLAM which 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 updates only affected nodes in the graph, Grisetti et al. (2010a) and Kaess et al. (2011b). Forster et al. (2015) and Forster et al. (2017) propose preintegration of IMU measurements in order to reduce the number of nodes while preserving the manifold structure of the $SO(3)$ rotation group. Carlone et al. (2014) defines a set of target variables to deal with smaller graphs. The solution enhances computational efficiency and robustness in the back-end. However, the relation with the perception front-end and robot navigation demands is still missing in robotics. This is the proposed research in this chapter of this thesis. A keyframe-based approach is proposed in Leutenegger et al. (2015) in order to select graph nodes and improve sparsity in visual-inertial odometry.

Multithreading is the common architecture in modern SLAM. 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 feature tracking and other thread for mapping. CD-SLAM Pirker et al. (2011) and ORB-SLAM Mur-Artal et al. (2015) includes a third thread for loop closing. Engel et al. (2015) introduces 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. The method is informally known as *direct approaches* and they have been gaining popularity in the last couple of years. Embodied SLAM combines particles filters and graph optimizers in a Hierarchical multi-process SLAM architecture, Schwendner et al. (2013). A Particle filter fuses embodied information in a multimodal probability distribution. A graph optimizer overcomes the limitations of particles filters for dense map reconstruction and loop closure.

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. Standard matching algorithms fail when areas which are 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 et al. (2007). The method looks for a set of similar features, i.e. geometry and appearance, between pairs of non-consecutive sub-maps. Williams et al. (2007) finds camera poses using the features in previous keyframe maps, similar to a relocalization event. A probabilistic approach to the problem of recognizing places based on their appearance is described in Cummins and Newman (2008). Their work detects and compares visual words between image pairs. Though not perfect, the method scales well in large datasets 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 and Tardos (2012). Embodied localization and mapping uses contact point robot information to build local segments in a global graph SLAM, Schwendner et al. (2013). However, the challenges of a robust loop closure remain open.

Introducing highly informative nodes and non-redundant poses improves computation cost, guarantees consistency, produces accurate results and speeds up data association. Within this context, heuristic strategies based on the distance traveled are investigated in Konolige and Agrawal (2008). The paper shows how the RMSE increases as a function of the distance between image frames. However, a distance-based criterion does not adapt to rover turning 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 et al. (2006). Other strategies such as information-based PoseSLAM use interval arithmetic and provide informative links and relevant poses in the graph, Ila et al. (2010). Their strategy exploits the benefits of using an error function in the graph back-end. The error function determines the distance 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 the situation demands. Adaptive SLAM proposes a novel architecture and the methodology to explore this solution. The methodology of this thesis identifies relevant keyframes and adapts the localization and mapping solution to the current navigation demands, dictated by the interaction with the ground. Adaptive SLAM uses more resources when the robot might get lost, avoid unnecessary computation and select the right moment to incorporate a new node in the graph.

6.3 Methodology

This section expounds the methodology of the 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 SLAM solution. Each contribution has been previously described in a dedicated chapter of this document. However, for the sake of consistency, the general overview is outlined here.

6.3.1 Overview

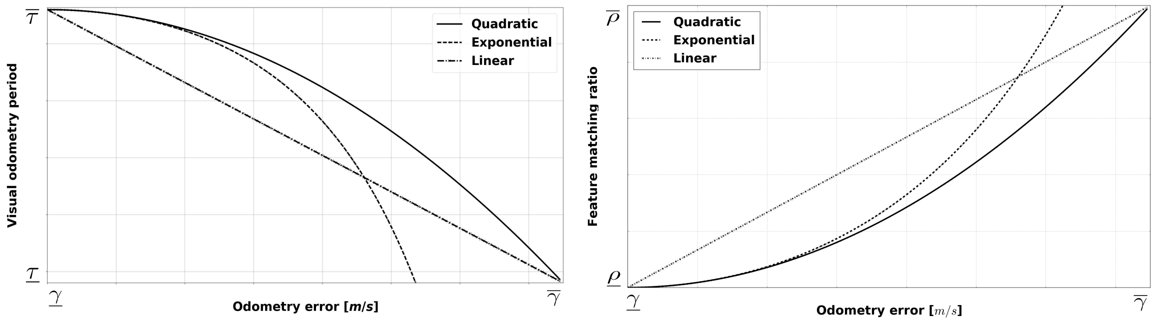
The estimation of motion in the *prediction* step is commonly performed using an odometry model. As previously mentioned, odometry combines motion models and inertial information in order to compute variations of the pose. Indoor robotics, which operates in a structured or planar environment, entails simplifications in the model. The simplicity causes a performance degradation, especially when porting the solution into planetary rovers. The 3D odometry model of an articulated chassis is described in Chapter 4. The relative delta displacements are computed. The parametric model is also enhanced with an adaptive odometry technique. This is based on a weighted least squares optimization which combines the delta displacement for all wheels. A simple friction model is assumed since many planetary rovers lack force sensors. The technique assumes the normal force on each wheel is proportional to the traction. The distribution of normal forces among the wheels of the robot is based on static reaction forces caused by the robot's weight.

The accuracy of the odometry is highly influenced by the amount of wheel slippage, which depends on the maximum usable tractive force between ground and the wheel. Odometry errors are modeled as the difference between the expected and the real delta poses, i.e. odometry residuals. Machine learning is used to train a Gaussian Process offline to estimate odometry errors, as defined and validated in Chapter 5.

Finally, the robot state prediction model is obtained with a realistic estimation of the uncertainty. The accuracy of the odometry is used by the SLAM architecture: the front-end adaptively acquires image frames depending on the dead reckoning performance (active perception) and adjusts the criterion to select keyframes in the back-end (adaptive information gain). Traction demands dictate the results based on the characteristics of the terrain. This methodology and the SLAM architecture is explained in the following.

6.3.2 Adaptive Visual Odometry and Graph Node Selection

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



(a) (b)

Figure 6.1: Comparison of three possible equations for the RoA (a) visual odometry period (b) feature matching ratio.

the localization and mapping to the current navigation demands. The adaptivity influences the front-end and back-end during operations. A method adapts the front-end in order to overcome poor wheel odometry. Another method adapts the selection of graph nodes or keyframes at the back-end. 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 in order 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 wheel odometry error. RoA is the response of the system to change or adapt to new circumstances. RoA can be a constant function chosen at design phase. However, 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, Fig. 6.1 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 equation (6.1) and equation (6.2) 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 wheel odometry 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 wheel odometry.

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* in order 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 lost of tracking is given in equation (6.1). Therefore, the sample period τ to compute visual odometry is given by

$$\tau = \frac{\underline{\tau} - \overline{\tau}}{(\overline{\gamma} - \underline{\gamma})^2} \ \|\breve{\boldsymbol{y}}_*\|^2 \ + \ \overline{\tau} \tag{6.1a}$$

$$\breve{\boldsymbol{y}}_* = \boldsymbol{y}_* + \Sigma_{\boldsymbol{y}_*} \tag{6.1b}$$

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 wheel odometry error, according to a threshold explained later, and $\boldsymbol{y}_*$ is the predictive mean of the wheel odometry error given by the Gaussian Process $GP_\mu(\boldsymbol{x}_*, D)$ with variance $\Sigma_{\boldsymbol{y}_*} = diag(GP_\Sigma(\boldsymbol{x}_*, D))$. Fig. 6.1a depicts the quadratic curve of equation (6.1).

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

$$\rho = \frac{\overline{\rho} - \underline{\rho}}{(\overline{\gamma} - \underline{\gamma})^2} \ \|\breve{\boldsymbol{y}}_*\|^2 \ + \ \underline{\rho} \tag{6.2a}$$

$$\breve{\boldsymbol{y}}_* = \boldsymbol{y}_* + \Sigma_{\boldsymbol{y}_*} \tag{6.2b}$$

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. Fig. 6.1b depicts the quadratic curve of equation (6.2). A novel method to relate the performance of the wheel odometry to graph node selection in the back-end.

The thresholds for the wheel odometry 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 $1.0\,\%$ to $2.5\,\%$ of the total distance traveled. For this reason the selected thresholds in velocity are $10\,\%$ and $25\,\%$ respectively. Those 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 wheel odometry error according to such slippage threshold by

$$\underline{\overline{\gamma}} = \left[\underline{\gamma} = 0, \overline{\gamma} = \hat{p}v\right] \tag{6.3a}$$

$$\hat{p} = 0, 0.10, 0.25, 1.0 \tag{6.3b}$$

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

The adaptive localization and mapping is depicted in Fig. 6.2. 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 wheel odometry. The visual tracking uses a constant velocity model in order to estimate the rover displacement. The visual tracking computes ORB features which are multi-scale FAST keypoints and BRIEF visual descriptors, Rublee et al. (2011), see Fig. 6.3. The visual odometry uses RANdom SAmple Consensus (RANSAC) to remove outliers. A local Bundles Adjustment (BA) optimizes the delta pose, depicted as *optimized odometry estimation* lines in Fig. 6.2. The visual odometry rate changes depending on the wheel odometry error using equation (6.1), depicted in a red colored bar in Fig. 6.2. 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, so-called Libelas, [1] Geiger et al. (2010). A local map is processed in an EnviRe [2] item and stored in an EnviRe graph. EnviRe is a representation model that facil-

[1] Library for Efficient Large-scale Stereo Matching `http://www.cvlibs.net/software/libelas`

[2] Environment Representation `https://github.com/envire`

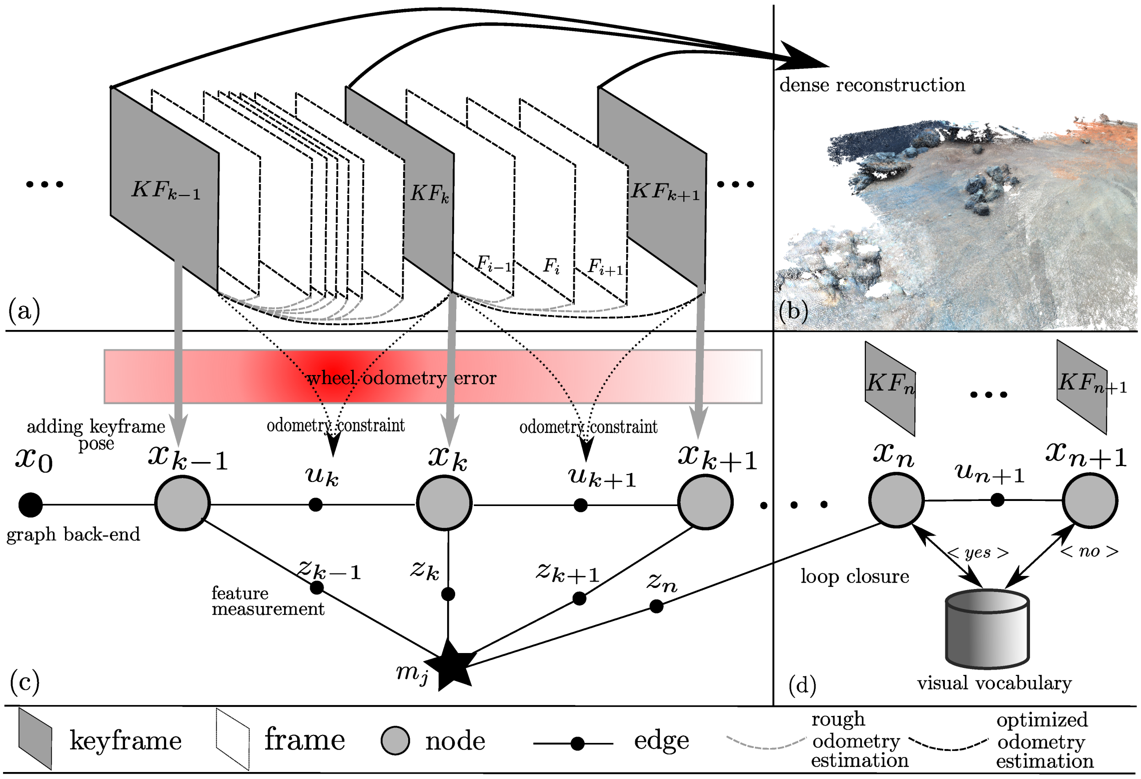


Figure 6.2: Anatomy of the adaptive localization and mapping system (a) visual odometry (b) dense map reconstruction (c) optimization back-end (*d*) loop closures engine.

itates real-world reconstruction using a connected directed graph. The third block, labeled (c), is the optimization back-end comprised of the graph and the smoother which estimates the final pose and the global map. The selection of keyframes uses equation (6.2). This module is based on the G2O [3] graph SLAM from Kummerle et al. (2011b). 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 [4] 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.

The inertial system module, which runs the AHRS explained in Chapter 3, pro-

[3] A General Framework for Graph Optimization `https://github.com/RainerKuemmerle/g2o`

[4] Hierarchical Bag of Words library `https://github.com/dorian3d/DBoW2`

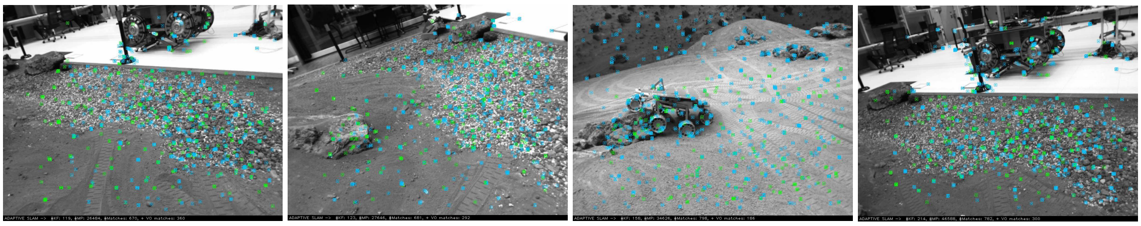


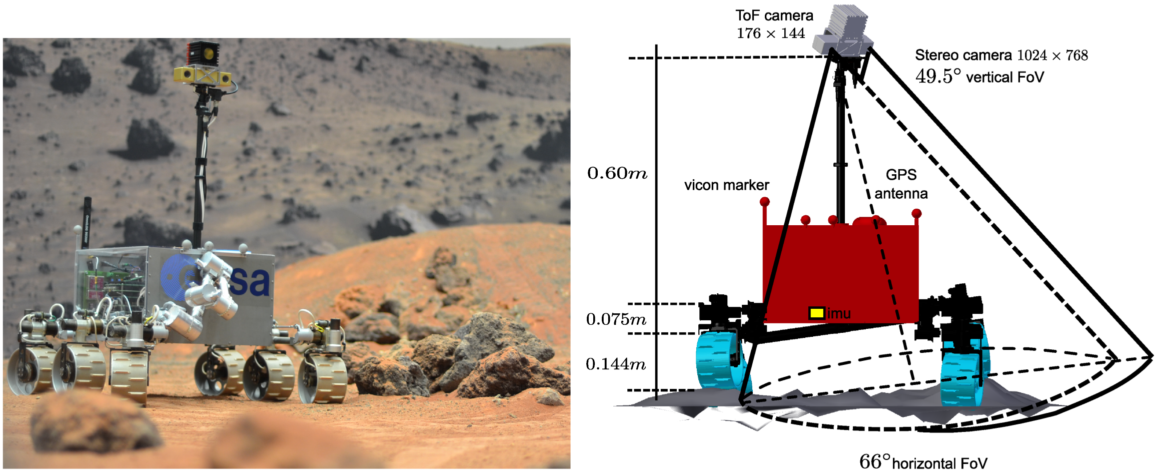
Figure 6.3: Camera images along the Mars-like terrain, revisiting locations to detect loop closures.

vides the orientation and inertial readings required by the odometry model. Joint positions and velocities are collected by the sensor drivers and dispatched together to the odometry model, as described in Chapter 4. Angular velocities $[\dot{\phi}_B, \dot{\theta}_B, \dot{\psi}_B]$ and joint measurements $\dot{\boldsymbol{q}}$ are sensed rate quantities of equation (4.5). 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 $\dot{\boldsymbol{\varepsilon}}$ and contact angles $\dot{\boldsymbol{\delta}}$ are non-sensed quantities. The complete vector $\dot{\boldsymbol{p}}$ in equation (4.4) has a dimension of 49×1. A system of equations is obtained by rearranging non-sensed (left-side) and sensed (right-side) quantities as in equation (4.6). ExoTeR has a vector of sensed $\boldsymbol{\nu}$ and non-sensed $\boldsymbol{b}$ quantities with dimension 28×1 and 27×1 respectively. The solution of the parametric model for 3D odometry is obtained by finding the least squares in equation (4.7) and setting the weighting matrix $\mathcal{W}$ from equation (4.21). Those computations run in a dedicated Rock [5] task and deployed as a single process on the 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 non-parametric odometry error model is computed offline. Gaussian Process inference is performed by finding the set of hyperparameters. The training data D are collected from driving the ExoTeR in a number of training experiments, described in Chapter 5. The training data inputs X are the pitch and roll angles (attitude) computed by the AHRS, the filtered inertial gyroscopes and accelerometers data and the position and speed joint measurements. In order 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 (velocity) with the delta pose from the ground truth. This is $\boldsymbol{y}_k = \boldsymbol{s}(k+1) - \boldsymbol{s}(k) - g(\boldsymbol{s}(k), \breve{\boldsymbol{u}}(k))$ solved for equation (5.9).

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 com-

[5] The Robot Construction Kit (Rock) http://www.rock-robotics.org



(a) (b)

Figure 6.4: The ExoMars Test Rover (ExoTeR) (a) during the test at the Planetary Robotics laboratory at ESA (b) sensors configuration and dimensions.

putes the dense map. SLAM receives the delta pose estimated by the wheel odometry 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, equation (6.1) and equation (6.2), 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 $\overline{\underline{\tau}} = [0.4s, 2.0s]$ in equation (6.1). This means that under poor traction circumstances the visual odometry is triggered up to five times faster than during good wheel odometry 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 $\overline{\underline{\rho}} = [0.3, 0.75]$ in equation (6.2).

6.4 Experimental Results

The odometry model and the odometry error rely on the proprioceptive information. The approach explained in this thesis is of course not limited to planetary rovers and it can be used in any ground vehicle with known kinematics. It can become especially useful for any vehicle traversing a challenging terrain on uneven surfaces. As in previous tests, the ExoTeR is the desired platform due to its resemblance to Mars planetary rovers. ExoTeR is shown in Fig. 6.4 and has been introduced in previous chapters. ExoTeR's technical details are described in Appendix A. Six wheels rovers have 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. The wheel walking motors are part of the deployment mechanism to stow the rover in a compact configuration. The wheel walking motors also serve for research on peristaltic motion modes, as the wheel walking in Azkarate et al. (2015). There are no steering motors in the middle wheels, making the chassis lighter but preventing the rover from crabbing.

The STIM300 IMU measures accelerations and angular velocities which are filtered using a Kalman-based Attitude and Heading Reference System (AHRS) described in Chapter 3. The stereo camera setup has a baseline of $12\,\mathrm{cm}$ 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 $1.86\,\mathrm{GHz}$.

6.4.1 Mars-like Terrain

These tests are conducted at the Planetary Robotics laboratory of ESA. They are described in the following.

Objective. The purpose is to evaluate the performance of the adaptive SLAM in the Mars-like terrain, a realistic and versatile testbed for planetary rovers. The goal is to understand how the adaptive thresholds alter the SLAM solution and to identify which values bring successful results.

Setup. ESA's planetary robotics laboratory comprises, among other facilities, a $9\,\mathrm{m} \times 9\,\mathrm{m}$ Mars-like terrain that resembles a planetary surface. As in previous experiments, the area is equipped with a set of twelve infrared emitting and sensing cameras mounted to the walls, which sense reflective markers on the rover platform. These cameras are part of the Vicon tracking system which can deduct and track position and orientation of objects 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. Fig. 6.5 shows the real-time VizKit 3D [6] visualization of the dense map reconstruction during one of the experiments.

The GP kernel runs in a single Rock task and predicts the current odometry error for each new input data x_* that arrives at the communication port. GP prediction is the trigger mechanism for the camera driver and the visual odometry module. Delta poses are incorporated by the SLAM system in order to track the features in the

[6] 3D Vizualization Kit https://www.rock-robotics.org/stable/documentation/graphical_user_interface/index.html

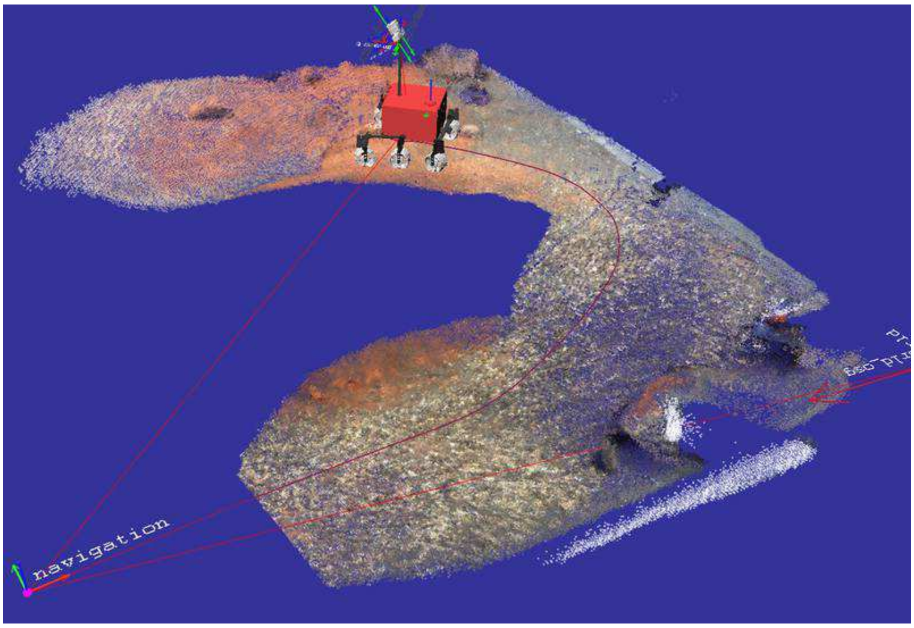
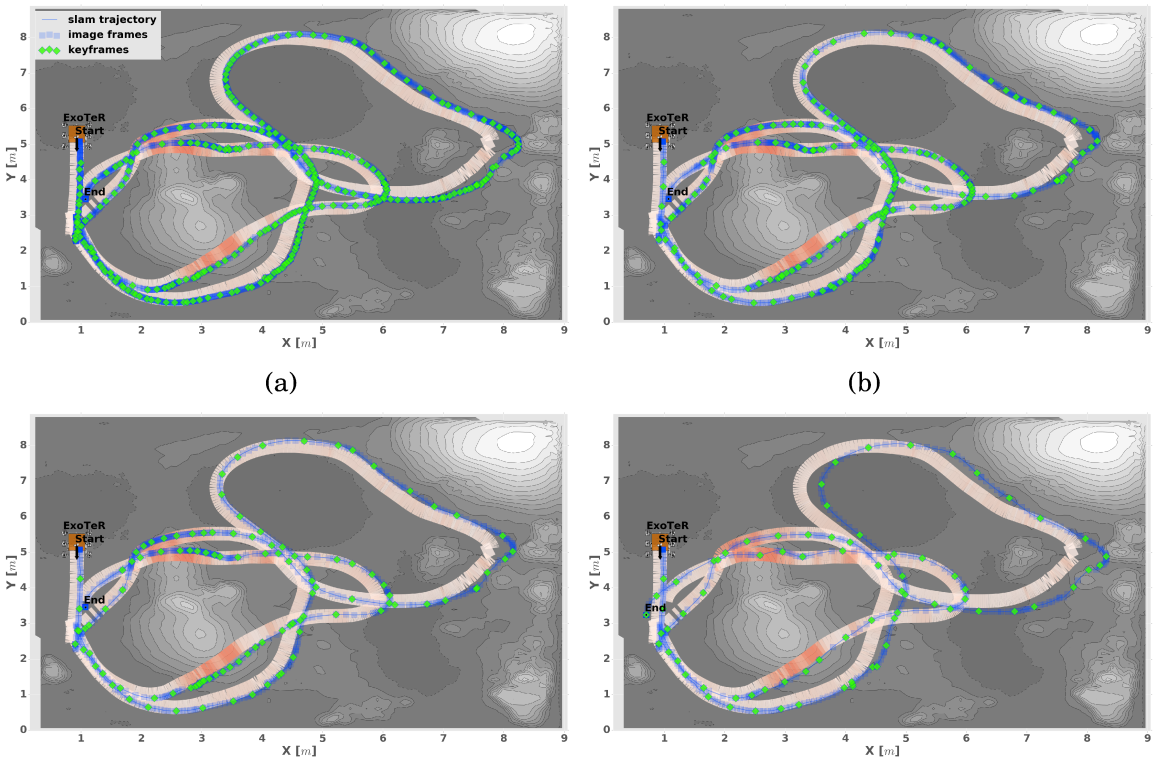


Figure 6.5: Real-time visualization of ExoTeR and the 3D map reconstruction during Test#1 at the Mars-like terrain of ESA' planetary robotics laboratory.

VO module, see Fig 6.2. The tracking module uses a constant velocity model calculated by the delta poses of the wheel odometry. The adaptive equation (6.1) decides whether an image frame has to be included in the VO tracking. In the meantime the odometry delta poses are pre-integrated between frames. When the adaptivity policy decides to trigger the image frame, the pre-integrated 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 (6.2) separately adapts the necessity to incorporate a keyframe in the SLAM back-end and perform an optimization step in the G2O graph.

Evaluation. Several maximum odometry errors $\overline{\gamma}$ are selected in order to test the feasibility of the approach. $\overline{\gamma}$ depends on the maximum allowed threshold in equation (6.3). 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 $0.063\,\mathrm{m/s}$ ($6.3\,\mathrm{cm/s}$) results in $\overline{\gamma}$ equal to $0.016\,\mathrm{m/s}$. Fig. 6.6 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 $\underline{\tau} = \overline{\tau} = 0.4s$ and a constant feature matching ratio $\overline{\underline{\rho}}$ of $75\,\%$ overlapping. The other three cases show the SLAM solution



(a)

(b)

(c)

(d)

Figure 6.6: Adaptive localization and mapping for different odometry error thresholds for the data in Test#1 (a) SLAM w/o adaptivity at $\underline{\tau} = 0.4s$ and $\underline{\rho} = 0.75$ (b) adaptive SLAM $10\,\%$ [$\overline{\gamma} = 0.0063ms^{-1}$] (c) adaptive SLAM $25\,\%$ [$\overline{\gamma} = 0.016ms^{-1}$] (d) adaptive SLAM $100\,\%$ [$\overline{\gamma} = 0.063ms^{-1}$].

for $10\,\%$, $25\,\%$ and $100\,\%$ slippage threshold with $\overline{\gamma}$ equal to $0.0063\,\mathrm{m/s}$, $0.016\,\mathrm{m/s}$ and $0.063\,\mathrm{m/s}$ respectively. The values are used in the adaptivity equations to perform different adaptiveness in the SLAM.

Note that in Fig. 6.6b and Fig. 6.6c the image frames and keyframes have a high spatial frequency in areas where the rover encounters poor traction performance, whereas in Fig.6.6a it is not. The solution depicted in Fig. 6.6d has a minimum number of keyframes in order to maintain SLAM consistency. Adaptive SLAM has a significant correlation between the odometry error and the graph structure. Adaptive SLAM overcomes the poor estimation of wheel odometry as desired. The adaptivity

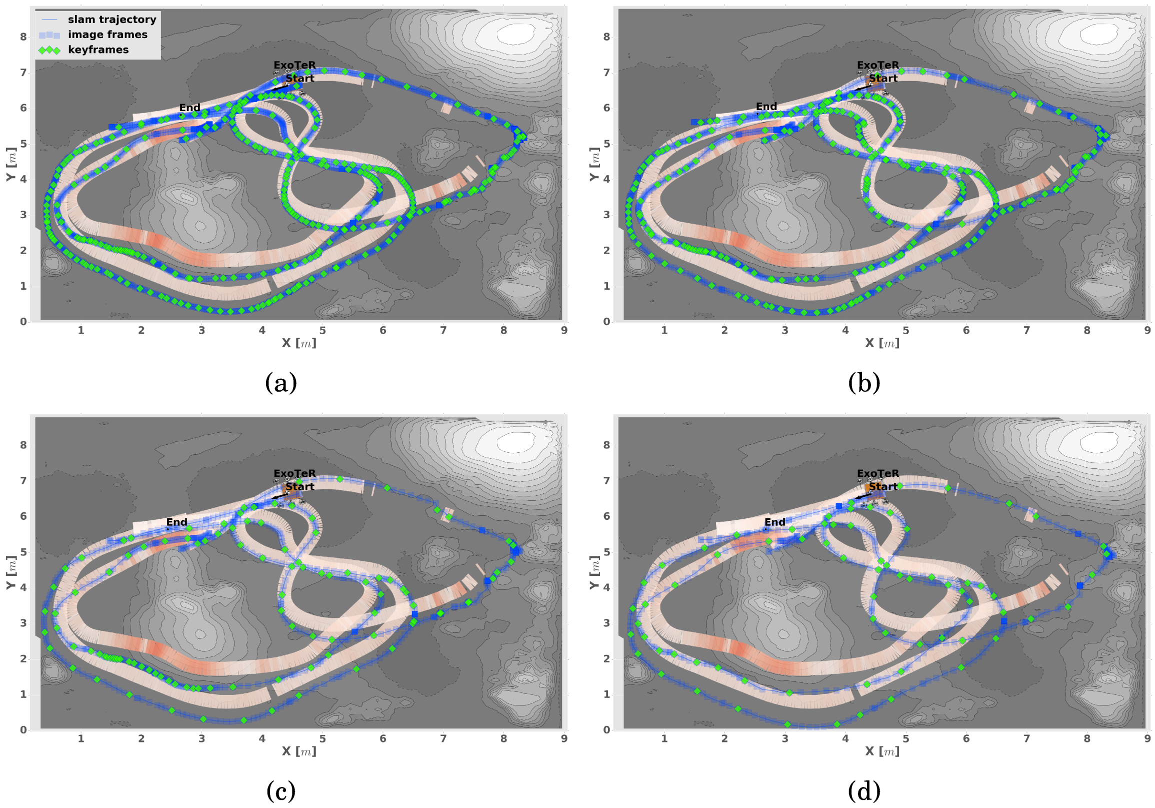
Table 6.1: ExoTeR's pose results for the different SLAM schemes running Test#1.

Adaptiveness	#Frames	#Keyframes	RMSE [m]	Max E. [m]	Final E. [m]	Max E. [%]	Distance [m]	#Loops
w/o adaptivity	2619	291	0.14	0.45	0.063	0.98	45.5	3
$10\,\%$ threshold	1206	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

threshold has an impact on the quality of the estimates. Table 6.1 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 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 since 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 6.1. This is because the adaptivity equation reduces image frames when wheel odometry performs reasonably well and 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. Fig. 6.7 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 during turning maneuvers than in straight parts of the trajectory. The keyframe selection criteria previously mentioned in Section 6.3.2 apply as in Test#1. The criterion is based on feature matching ratio instead of distance traveled, which generates more keyframes when ExoTeR is turning. The equation also adapts the density in areas with poor wheel odometry results and reduces the number of keyframes when ExoTeR drives with good wheel odometry 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 location due to partial occlusion of the reflective markers. ExoTeR does not encounter, in term of absolute numbers, as much slippage as in Test#1. The rover does not reach 100 % slippage during Test#2. 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 the traction performance. Such information is interpreted by the adaptive SLAM and adapts the parameters online as depicted in the plots. Table 6.2 summarizes the error value depending on the adaptive criteria.

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



(a)

(b)

(c)

(d)

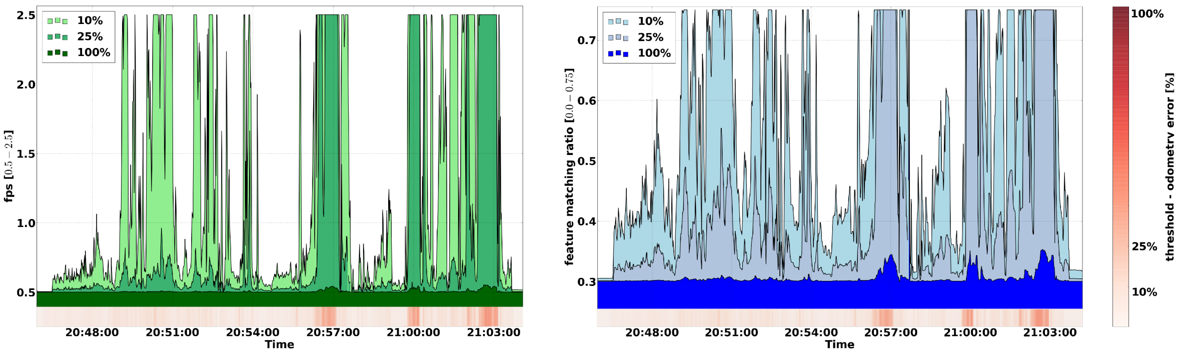
Figure 6.7: Adaptive localization and mapping for different odometry error thresholds for the data in Test#2 (a) SLAM w/o adaptivity at $\overline{\tau} = 0.4s$ and $\overline{\rho} = 0.75$ (b) adaptive SLAM $10\,\%$ [$\overline{\gamma} = 0.0063ms^{-1}$] (c) adaptive SLAM $25\,\%$ [$\overline{\gamma} = 0.016ms^{-1}$] (d) adaptive SLAM $100\,\%$ [$\overline{\gamma} = 0.063ms^{-1}$].

has a stronger influence in the loop closure. Generally, a camera looking 30° downwards produces a more accurate delta pose visual odometry and terrain maps, while 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 changes along the trajectory. This is due to the adaptive variation of the frame period and feature matching ratio. Fig. 6.8 and Fig. 6.9 show the time evolution for the frame period and feature matching ratio correlated with the odometry error (red color

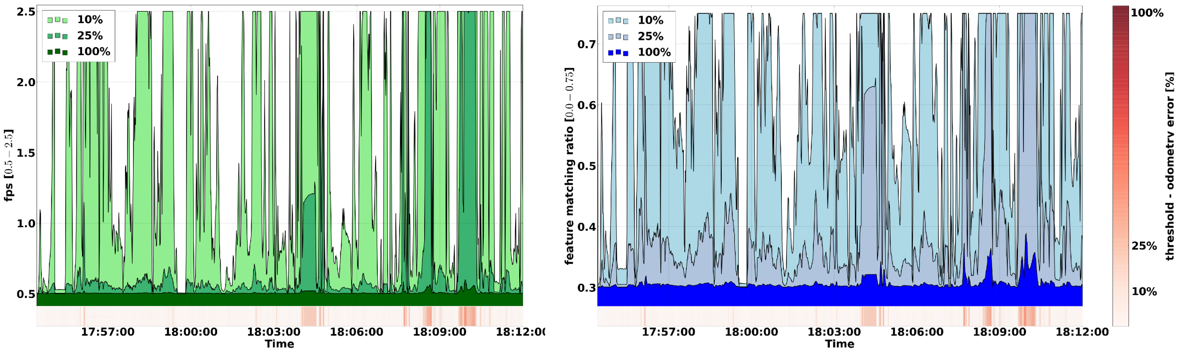
Table 6.2: ExoTeR's pose results for the different SLAM schemes running Test#2.

Adaptiveness	#Frames	#Keyframes	RMSE [m]	Max E. [m]	Final E. [m]	Max E. [%]	Distance [m]	#Loops
w/o 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



(a) (b)

Figure 6.8: Adaptive evolution at Test#1 for (a) the image processing at $1/\tau$ frames per second and (b) feature matching ratio ρ for the $10\,\%$, $25\,\%$ and $100\,\%$ scheme.



(a) (b)

Figure 6.9: Adaptive evolution at Test#2 for (a) the image processing at $1/\tau$ frames per second and (b) feature matching ratio ρ for the $10\,\%$, $25\,\%$ and $100\,\%$ scheme.

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 thresholds selected at each experiment labelled on the left side of the graph together with the color bar of the legend on the right side. The slope variation of the curves is related to two factors. The odometry error predicted from the GP model, and the quadratic equations described in equation (6.1), for the frame period or equation (6.2), for the feature matching ratio. The area covered by the plots in Fig. 6.8 and Fig. 6.9 can be interpreted as the amount of computational effort required for the adaptive SLAM for each percentage threshold. With this interpretation, an adaptive SLAM with $0\,\%$ slippage threshold covers the complete space, all computational resources. The adaptive SLAM with $100\,\%$ slippage threshold covers 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

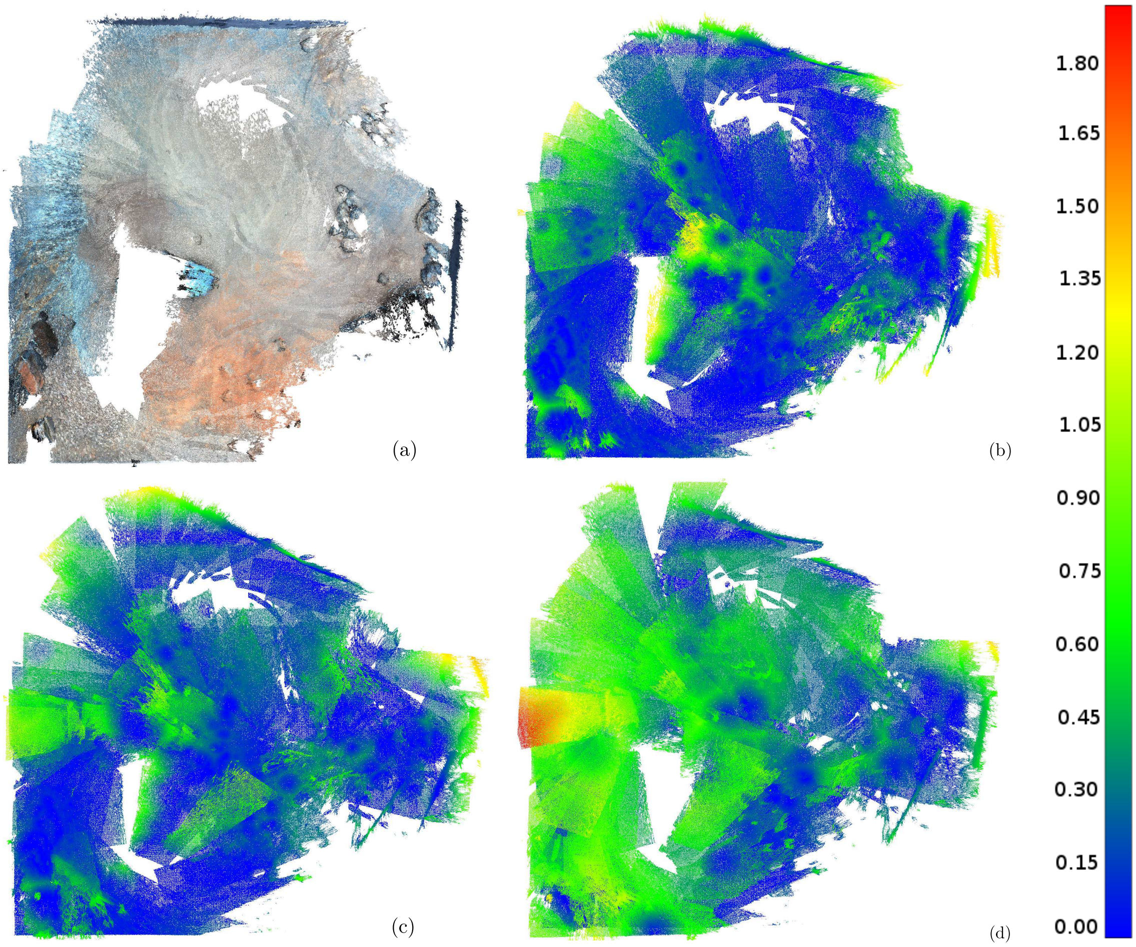


Figure 6.10: Adaptive map quality for different odometry error thresholds in Test#1 (a) colored reference dense map reconstruction, (b) error map with respect to the reference map for 10 % threshold in adaptiveness, (c) error map for 25 % threshold in adaptiveness and (d) error map for 100 % threshold in adaptiveness. The colored error legend in meters is depicted on the right side.

surroundings. The dense map reconstruction, see Fig. 6.2, 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. Fig. 6.10 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 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 Fig. 6.10, 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.

6.4.2 Decos Terrain

Objective. The purpose is to test the performance of the adaptive SLAM outdoors, on a different terrain from the Mars-like terrain. The goal is to conduct a longer drive, in a different environment with sunlight conditions. For that reason the Decos terrain is selected due to its resemblance to a Mars landscape. However, the diversity of the Decos terrain and variety of soil types is not as realistic as in the PRL testbed which mimics Martian soil.

Setup. Decos terrain is located in Noordwijk, The Netherlands, in proximity to the European Space Research and Technology Center (ESTEC). 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 dimensions 50 m × 80 m comprises of medium size rocks and red broken bricks. ExoTeR is equipped with a GPS antenna in order to acquire ground truth measurements along the drive. The stereo camera sensor is also tilted 30° downwards as in Test#1. An autonomous drone is used to capture high-resolution aerial images and produce a Digital Elevation Map (DEM) of the target zone. The generated DEM is used to render the ground truth map and contour lines for the resulting figures of the experiment. Fig. 6.11 shows the ExoTeR located at the base camp with the necessary equipment and the remote control station.

Table 6.3: ExoTeR's pose results for the different SLAM schemes at Decos terrain.

Adaptiveness	#Frames	#Keyframes	RMSE [m]	Max E. [m]	Final E. [m]	Max E. [%]	Distance [m]	#Loops
w/o adaptivity	6133	797	1.06	5.41	0.48	3.05	177	0
10 % threshold	2982	530	1.07	5.56	0.53	3.14	177	0
25 % threshold	1352	243	1.29	5.61	0.67	3.16	177	0
100 % threshold	1224	215	1.63	6.30	1.19	3.56	177	0

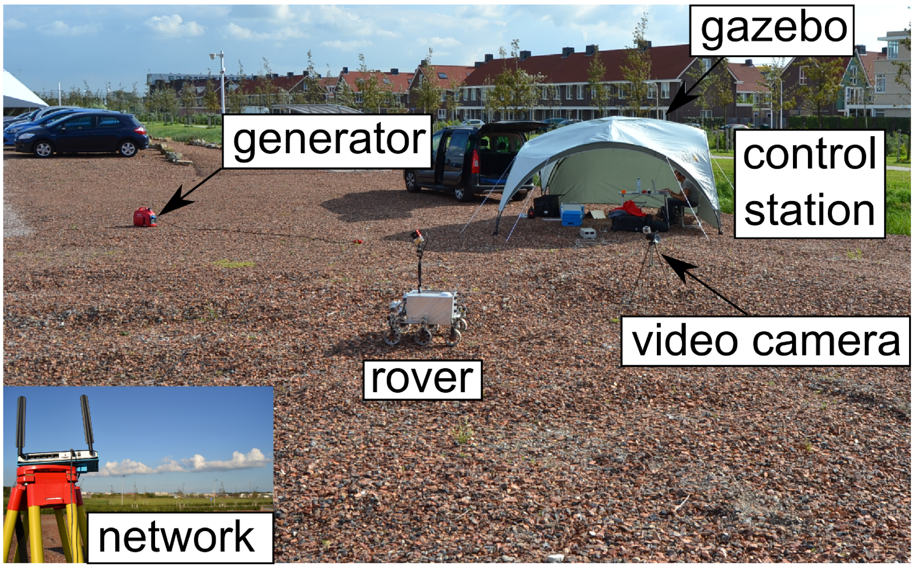
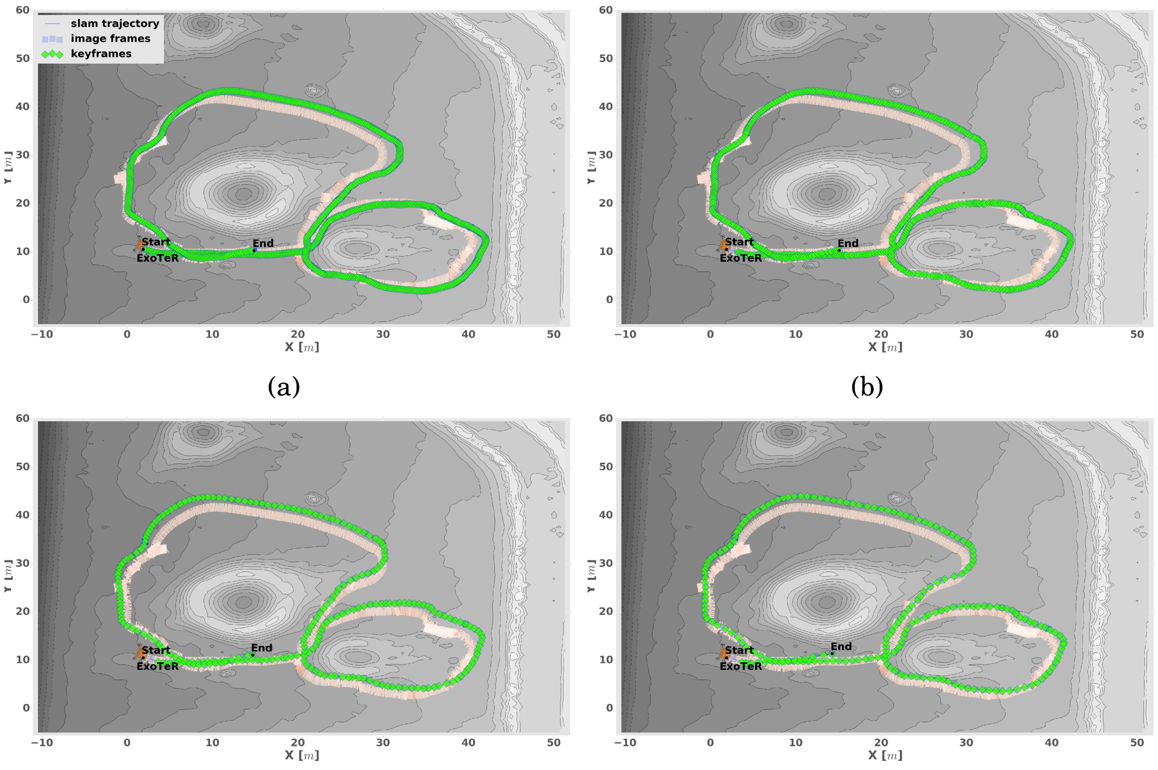


Figure 6.11: Field-test control station next to the Decos test area.

Evaluation. Similar to the previous tests, the wheel odometry error dictates the adaptivity of the SLAM. The delta poses from the odometry module are pre-integrated between image frames and used to track visual features. The equation (6.1) adapts the VO odometry frequency and equation (6.2) 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 $10\,\mathrm{cm/s}$. The same four thresholds are selected to compare the adaptivity of the approach.

Fig. 6.12 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 $0.01\,\mathrm{m/s}$, $0.025\,\mathrm{m/s}$ and $0.10\,\mathrm{m/s}$ respectively. Table 6.3 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 wheel odometry 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. Fig. 6.13 shows the time evolution for the frame period and Fig. 6.14 shows feature matching ratio correlated with the odometry error



(a)

(b)

(c)

(d)

Figure 6.12: Adaptive localization and mapping for different odometry error thresholds for the data in Decos terrain (a) SLAM w/o adaptivity at $\underline{\tau} = 0.4s$ and $\underline{\rho} = 0.75$ (b) adaptive SLAM $10\,\%$ $[\overline{\gamma} = 0.010ms^{-1}]$ (c) adaptive SLAM $25\,\%$ $[\overline{\gamma} = 0.025ms^{-1}]$ (d) adaptive SLAM $100\,\%$ $[\overline{\gamma} = 0.10ms^{-1}]$.

(red color bar located at the bottom of the plot). Decos test shows that even having the same quadratic equations the variation in the plot is more constant than in the Mars-like terrain. For instance, there are fewer drastic variations in the plot of Fig. 6.13 and Fig. 6.14 for the Decos test, than in the plot of Fig. 6.8 for Test#1 at the Mars-like terrain. This denotes a constant slope at lower frequency for the frame period and the feature matching ratio when the terrain has less diverse (same soil and roughness along the path). However, there is still some differences in the area covered among the curves for 10 %, 25 % and 100 % slippage threshold. Additionally, note that the gap at time 18 h and 30 min and after time 18 h and 50 min in both figures is because the rover stops for few minutes and therefore the GP odometry error is near zero.

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

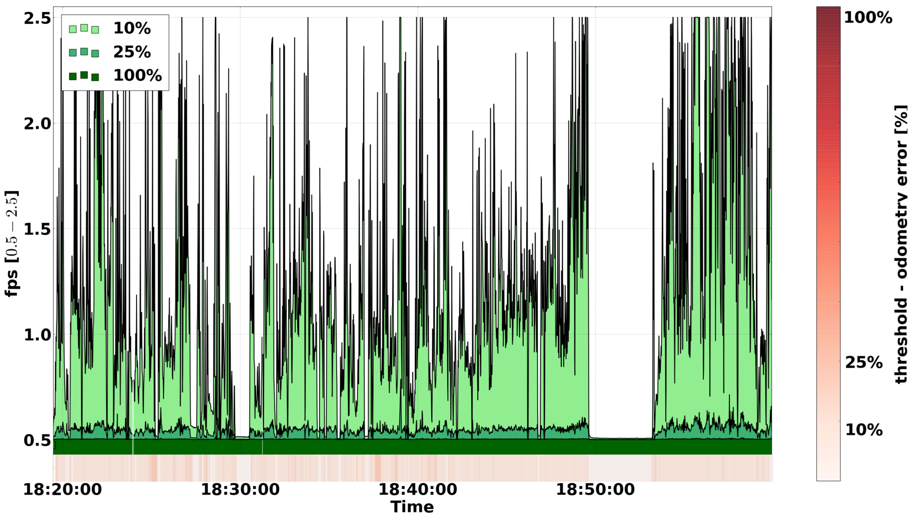


Figure 6.13: Adaptive evolution at Decos test for the image processing at $1/\tau$ frames per second for the $10\,\%$, $25\,\%$ and $100\,\%$ scheme.

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. Fig. 6.15 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

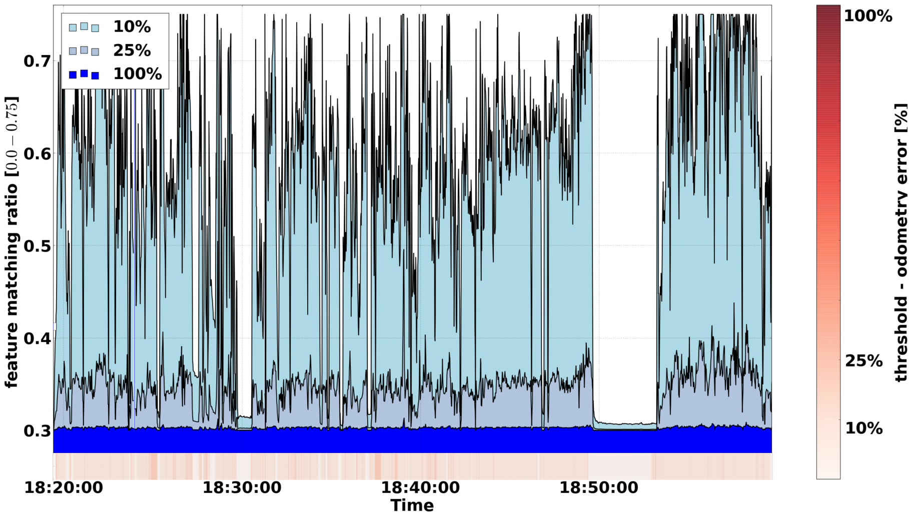


Figure 6.14: Adaptive evolution at Decos test for the feature matching ratio ρ for the $10\,\%$, $25\,\%$ and $100\,\%$ scheme.

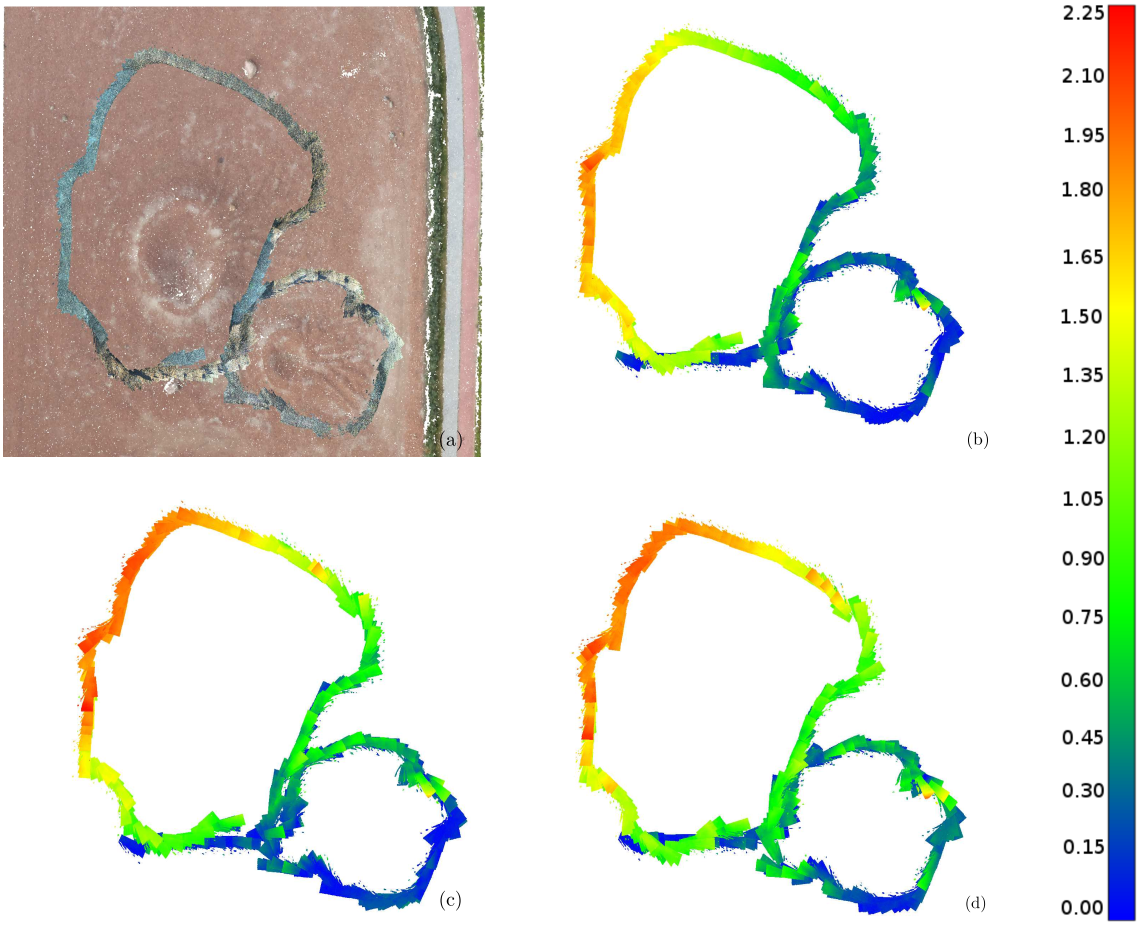


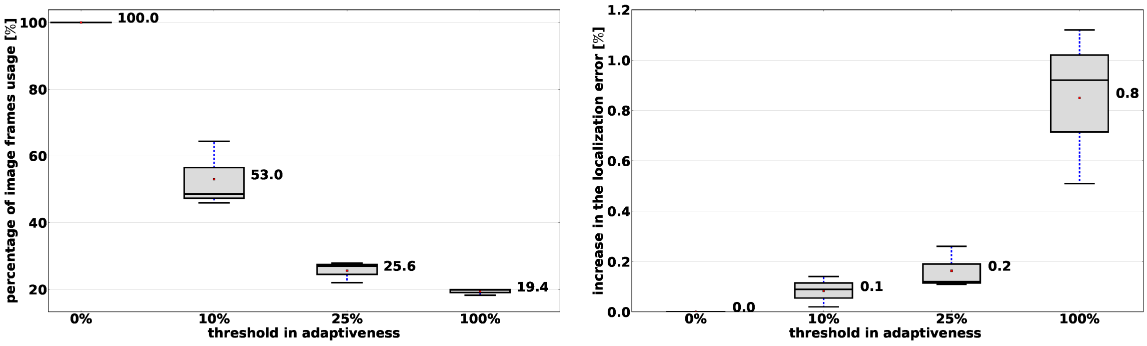
Figure 6.15: Adaptive map quality for different odometry error thresholds in Decos terrain (a) colored reference dense map reconstruction, (b) error map with respect to the reference map for 10 % threshold in adaptiveness, (c) error map for 25 % threshold in adaptiveness and (d) error map for 100 % threshold in adaptiveness. The colored error legend in meters is depicted on the right side.

rest of the visuals shows 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.

6.5 Influence in a Planetary Mission

This section analyses the impact of including the adaptive SLAM into the navigation system of a potential planetary rover. The argumentation connects the motivation of this thesis in Chapter 1 to the final results of the adaptive SLAM. The purpose is to elaborate an analysis on how the adaptive SLAM influences a real mission scenario.

Fig. 6.16a shows the influence of adaptive SLAM in the number of image frames



(a) (b)

Figure 6.16: Adaptive localization and mapping (a) influence of the threshold in adaptiveness to the number of computed image frames p_{us} (b) penalty of the threshold in adaptiveness to the percentage error per distance traveled.

processed by the visual tracking module. Fig. 6.16b 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 Fig. 6.16b. A small number of image frames is always required, independently of the adaptiveness, in order 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 Fig. 1.5 is explained also here in order to evaluate the impact of the adaptive SLAM. For this purpose the GNC of the ExoMars rover is used as reference. Fig. 6.17 depicts the system in two parts. A first part in which 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 in order to follow the desired trajectory and compute the localization. This is repeated every $2\,\mathrm{m}$ until the final target is reached. Such target is usually selected from aerial images. The localization & 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 $33\,\mathrm{cm}$. With these dynamics, the ExoMars rover stops six times every $2\,\mathrm{m}$ of traversed distance. The idea is to embed the adaptive SLAM into the *navigation system* and evaluate the impact on the distance traveled.

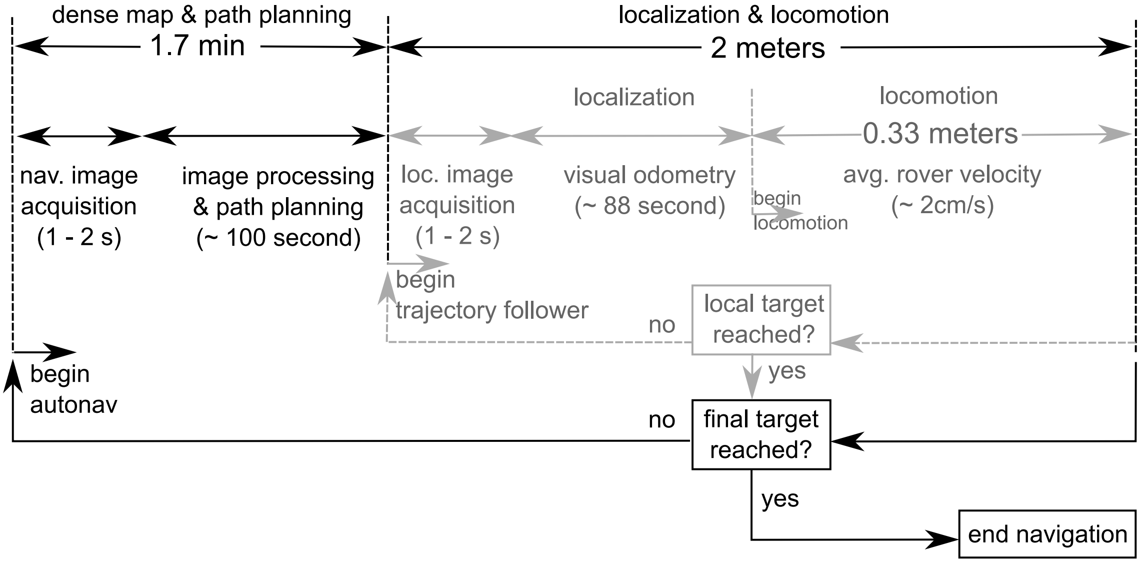

dense map & path planning
1.7 min
localization & locomotion
2 meters
localization
locomotion
0.33 meters
nav. image acquisition (1 - 2 s)
image processing & path planning (~ 100 second)
loc. image acquisition (1 - 2 s)
visual odometry (~ 88 second)
begin locomotion
avg. rover velocity (~ 2cm/s)
begin trajectory follower
no
local target reached?
yes
begin autonav
no
final target reached?
yes
end navigation

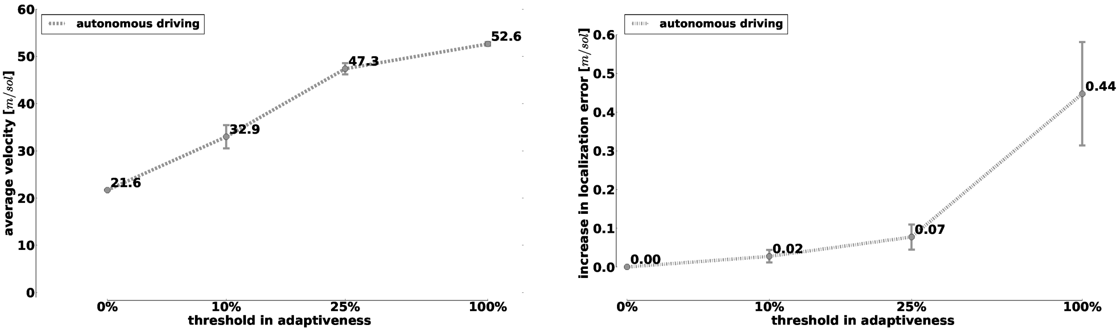
Figure 6.17: The *navigation system* for the *autonomous driving* of the ExoMars rover.

The average distance per sol for past, current and future Mars rovers is introduced in Chapter 1. ExoMars rover has a maximum requirement of $50\,\mathrm{m/sol}$ during Phase-B2. The navigation system requires 25 navigation image pairs every sol in order to compute a DEM every two meters of traversal. The computational time is $t_{nav} = 1.7min$, which involves a total of $42.5\,\mathrm{min}$ every sol in order 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 $t_{vo} = 1.5min$. 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 Fig. 6.16a.

$$T = \left[\frac{d}{l_{loc}} t_{nav}\right] + [n_f \cdot p_{us} \cdot t_{vo}] + t_{loc} \tag{6.4}$$

where d is the distance to navigate ($50\,\mathrm{m}$), l_{loc} is the locomotion & localization part ($2.0\,\mathrm{m}$), n_f is the number of localization image frames in a distance d (151 image pairs for $50\,\mathrm{m}$), t_{loc} is the time to traverse a distance d at the nominal rover velocity of $2.0\,\mathrm{cm/s}$ and p_{us} changes according to the values depicted in Fig. 6.16a. The result for maximum $p_{us} = 1.0$ ($100\,\%$) is a total time T of $5.19\,\mathrm{h}$ in order to navigate $50\,\mathrm{m}$. A minimum $p_{us} = 0.194$ ($19.4\,\%$) gives a total time T of $2.13\,\mathrm{h}$ 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 $2.25\,\mathrm{h}$. This is used to define a *locomotion sol* in Chapter 1 and to cal-



(a)

(b)

Figure 6.18: Adaptive SLAM analysis on a planetary rover mission scenario (a) the rover's velocity per different thresholds in adaptiveness (b) the increase in error per different thresholds in adaptiveness.

culate the maximum time that a solar powered rover can effectively drive on Mars, i.e. around noontime. Fig. 6.18a depicts the results of an average rover velocity per sol considering the *navigation system* from Fig. 6.17 and using $2.25\,\mathrm{h}$ of driving time. The values are calculated using the total time T from equation (6.4). 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 $50\,\mathrm{m/sol}$ 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. Fig. 6.18b shows the impact of each threshold to the final localization error per sol. The error bars associated to each value represent the 1σ standard deviation with the data from the experiments described in this chapter. The adaptive SLAM with $100\,\%$ threshold has a $0.85\,\%$ increase in the percentage error. This value translated to effective meters after traveling $52.6\,\mathrm{m}$ (see Fig. 6.18a) results in an increment of $0.44\,\mathrm{m}$ in the final error. This means, that in case the final error after traversing $52.6\,\mathrm{m}$ is $1.0\,\mathrm{m}$ with a classical SLAM, the final error with

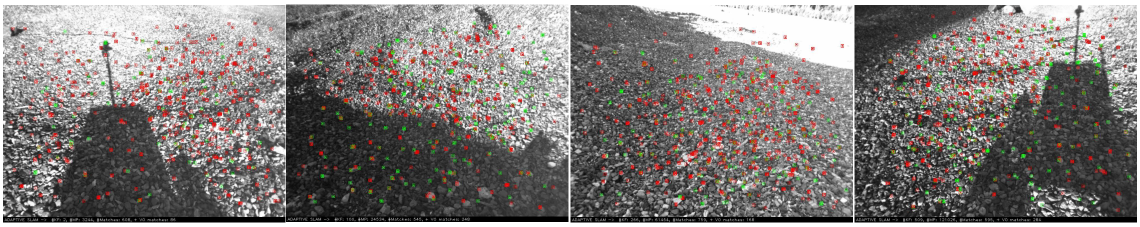


Figure 6.19: Camera images along the Decos terrain showing a perceptual aliasing which entails difficulties to detect a loop closure.

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, i.e. terrain, gravity and illumination conditions. Nevertheless, it gives a valuable information on how it could result in a real mission scenario on Mars.

6.6 Conclusion

This chapter introduces a novel method to positively influence the SLAM with an adaptive graph sparsity. The adaptive SLAM is possible due to optimization and machine learning techniques which bring to the next step the interpretation of raw information generated by the rover. The technique is based on the rover interaction with the traversed terrain by means of odometry errors. The research demonstrates that the information from the interaction with the terrain provides valuable data which are useful in order to adapt the localization and mapping solution. Fig. 6.18a shows how the rover velocity increases as a function of adaptiveness with a small penalty in the localization error, see Fig. 6.18b. More specifically, the average rover velocity doubles in the adaptive SLAM with 25 % slippage threshold in comparison with a state-of-the-art SLAM (without adaptivity).

Perceptual aliasing can influence the loop closing under certain situations in a planetary scenario using sensors with a limited field of view. Fig. 6.19 shows image frames during the Decos test with similar appearance. The perceptual aliasing produces indistinguishable bag of word vectors with multiple candidates in the dataset. It makes multiples options without a clear candidate to determine when a place is revisited. A solution investigated here is to increase the view of the cameras by not tilting the sensor unit, see results for *Test#2* at the Planetary Robotics laboratory. Further ideas and alternatives to minimize the perceptual aliasing are given in Chapter 7.

Future long-term SLAM systems will require adaptive graph sparsity, such as the method presented here. The technique improves scalability. Ideally, an adaptive SLAM system should be complemented with faster computational hardware, such as FPGAs or GPUs, in order to boost the computational speed, increase parallelization and reduce the value for the slippage threshold. With such hardware improvements a moderate value between $10\,\%$ and $25\,\%$ percentage threshold in equation (6.3) should allow a non-stop, accurate and faster traversal. Future planetary missions can benefit from adaptive SLAM to fulfill the requirements of a Sample Fetching Rover (SFR) with $200\,\mathrm{m/sol}$ as nominal velocity.

Chapter 7

Conclusions and Discussion

This chapter summarizes the results of this thesis and presents the lessons learned during the process. Some insights for future work regarding long-term localization and mapping are described at the end.

7.1 Thesis Summary

This thesis is at the conceptual line between an adaptive data association in modern SLAM systems and the mission requirements for planetary rovers. Within this context the thesis contributes to the improvement of dead reckoning processes and the adaptiveness in SLAM. This thesis develops a bottom-up approach to solve the localization and mapping problem in planetary rovers. It starts from basic means in an attitude and heading estimation, continues with a 3D odometry, defines a Gaussian Process (GP) regression and finishes with integration in a modern SLAM system. Each block builds on top of its predecessor to develop a novel SLAM solution which serves the needs of a potential planetary rover. The thesis statements described in Section 1.3 of Chapter 1 identify the missing gaps in four different domains: attitude estimation, motion models, machine learning and full SLAM systems.

Part I focuses on dead reckoning as the elementary technique to localize the rover from a starting position. An inquisitive reader might question the dedication of a complete part in this thesis to dead reckoning processes. Mainly because a full SLAM is the state-of-art solution. One could directly implement a tightly coupled nonlinear optimization-based SLAM as a monolithic approach where all sensory information is optimized at once. This could be a valid approach, except for the fact that planetary rovers are very constrained systems where autonomy diverges from mission to mission or from operation to operation mode within the same mission. The autonomous capabilities are remotely selected from Earth depending on mission demands. Therefore, a rover could operate in dead reckoning mode for a period of time, when scientific analyses are more relevant that navigation performance, and switch to a modern SLAM approach when a longer traversal distance is required. The second reason to have a dedicated section to dead reckoning, is that a modern SLAM should not prevent to full characterization and analysis of the unbounded errors. This helps to better understand the benefits of sensor fusion, the contributions of SLAM, mapping and loop closure versus dead reckoning in a particular platform, such as a planetary rover. Conclusions for Part I are summarized in the following.

The research in the AHRS design studies the impact of inertial sensor errors in the estimated robot's attitude. Nowadays, the characterization of inertial sensors cannot be understood without the use of the Allan variance. The work extends previous research on time series analysis and incorporates the technique into the indirect form of a Kalman filter. The presented research describes a process to follow in order to design a robust AHRS. The design, depicted in Fig. 3.1, is divided into three steps: error characterization, model derivation and filter design. This research provides an End to End approach that directly relates stochastic errors with the inaccuracies in the estimation of the robot attitude. The methodology provides a general design which adapts to most of the inertial sensors used in the aerospace sector. This is the reason why the methodology is applied in two complete different set of inertial sensors, a DETF-based and a FOG-based IMU. The presented methodology provides a design process for AHRS with improvements in accuracy.

Investigations on enhanced 3D odometry describe the basics of a generic and complete odometry model for outdoor robots. The method fuses information from the locomotion system and uses it in the dead reckoning of a localization scheme. The research is motivated in Chapter 1 by describing how MER rovers accumulated only $3\,\%$ position error over $2\,\mathrm{km}$ of driving on soil, flat and level ground using conventional odometry. The primary objective of the wheel odometry is to capture the whole kinematics of the locomotion system by using a complete model. A skid odometry model might be attractive to compute due to the simplicity of the solution. However, the motivation for this research was to demonstrate that this simplicity has a penalty in accuracy. Nevertheless, independently of the performance, the use of transformation matrices is more convenient, practical and generic than previous *ad-hoc* contact-point approaches as in Lamon and Siegwart (2007) and Schwendner et al. (2013) which does not easily adapt to wheeled robots. The technique described in this thesis is generic and adapts to any open kinematic chain by means of transformation matrices. The set of contact points is combined in a least squares optimization, which minimizes the error. Despite non-systematic errors, a 3D odometry model shows improvements in accuracy. The absolute improvement might depend on the rover trajectory and maneuvers but always benefits on uneven terrains. The odometry is able to propagate the movement taking into consideration the full kinematics constraints and the expected traction forces. The methodology is a 6 DoF motion model which decreases the percentage error per distance traveled from 8.3 % to 2.7 % on uneven and soil terrains during test with the Asguard rover and from 9.54 % to 6.51 % with the ExoTeR. The model reduces by half the percentage error per distance traveled in ExoteR in trajectories with considerable turning maneuvers as shown in the Decos scenario, see Table 4.2. In addition, the benefits of developing a complete motion model are also significant to control the rover. Azkarate et al. (2015) and Wiese (2017) apply the motion model approach to control the rover with the wheel walking maneuvers.

Though the results in Part I are already an achievement in the field, it does not fit the navigation requirements for planetary rovers (robust against slippage, loop closure, adaptivity, etc..). Machine learning and sensor fusion techniques are applied to combine with visual odometry at a higher level of data fusion (SLAM). Visual odometry is an important piece of work in localization and mapping. The enhanced 3D odometry aims to complete the visual odometry rather than substitute it. Parallel computing, either running in a GPU or FPGA, will bring the technique to a faster computational mean. The enhanced 3D odometry exploits the information available from the locomotion system and the contact with the ground to fuse meaningful information from the kinematics. The question raised here is how to fuse visual and wheel odometry in order to take the most from both techniques, minimizing redundancy and computational cost and maximizing efficiency and robustness. The answer given in this thesis is adaptive SLAM. A methodology to predict the odometry error in order to dynamically adapt the visual odometry load and influence the system back-end. Such research constitutes Part II and builds on top of the methodologies developed in Part I. The conclusions of Part II are summarized in the following.

Gaussian processes (GPs) are a practical machine learning tool for robotics and are investigated and applied in this thesis. The modeling of non-systematic odometry errors, affected by a poor traction performance, is an open research problem in robotics. The reason is the direct impact on robot control, planning and localization. Instead of modeling the odometry of the rover, the approach model the error with respect to the estimation from a parametric model. This is very convenient for GPs since all the modeling is in the covariance function. The odometry error model does not require extra sensors beyond what is usually available in planetary rovers (inertial sensors and joint encoders). The results have shown reasonably good predictions on representative terrains as shown in Fig. 5.4 and Fig. 5.6. GPs are able to learn the hyperparameters of a kernel function in order to derive a non-parametric model of the odometry error. GP predictions 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 predicts the odometry error with extraordinary results. Contrary to a high odometry error, a low odometry error might be inestimable by the method due to a low signal to noise ratio. This situation might occur when the minimum sensing level of the proprioceptive sensors overlaps with the odometry error. Finally, it could be interesting for future work to entirely model the wheel odometry using GPs instead of the residuals. It would entail inserting the output of a parametric odometry as input data into the nonlinear regression model. However, this thesis proposes the solution to model the error and do not become dogmatic about learning. Why to learn an odometry model when there is an accurate parametric motion model available? Specially, when the values of interest are the errors or inaccuracies of the model.

An offline learning approach is presented in Chapter 5, where the model is learned on training runs and evaluated in a test environment. The capability of the rover to predict the odometry error using proprioceptive data is an important achievement not only for SLAM but for machine learning. However, the strategy can be improved since online learning might be required on highly dynamic terrains, where traction performance change significantly from one location to another. Future work includes online learning and using visual odometry as target inputs instead of ground truth data. Initially visual odometry will be queried frequently, and as model uncertainty decreases the frequency of visual odometry can be reduced and computational effort can be saved. The learning step, which is more computationally expensive, can be computed over night, when the rover is stopped, waiting for new commands for the next day. Techniques such as Incremental Local Gaussian Regression, Meier et al. (2014), would allow online learning and adapting the localization and mapping, as well as the visual odometry, to the dynamics of the environment. Odometry errors might also provide a cue to identify the terramechanics of the terrain and inform the path planning component about potential hazards.

Adaptive SLAM in Chapter 6 finally encompasses the efficiency and purpose of this thesis. 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 wheel odometry has the same accuracy as visual odometry (approx $1-2\,\%$ error). Errors in both techniques increase unbounded unless a loop closing strategy applies. The ability to use the odometry error to encompass the classical visual SLAM approach is interesting for planetary rovers. The ability to adapt the SLAM solution autonomously and in real-time as the rover navigates the environment, is the contribution of adaptive SLAM. This contribution allows to adapt SLAM computational load to the current navigation demands. Table 5.2 shows how *SLAM with GP* can double the accuracy in comparison to SLAM without GP with a graph which has the same number of nodes, approximately $150\,\mathrm{nodes}$. The requirements depend on a design criterion, and the criteria are driven by the mission. This thesis presents two criteria, an active perception and an informative keyframes selection strategy. Firstly, active perception autonomously adapts the visual odometry computational load to the situation's demands. This minimizes the computational cost while still providing reliable results. Secondly, informative keyframes selection provides an efficient distribution of keyframes and therefore nodes in the graph. The adaptive SLAM with 10 % threshold reduces the computational cost of VO by half (from 2619 frames to 1206 frames) with respect to a classical SLAM and with 100 nodes fewer in the graph, obtaining similar RMSE in position, see Table 6.1. An average error of approximately 35 cm in the quality of the point clouds in mapping is achieved for the adaptive SLAM with 10 % threshold. In addition, Fig. 6.18a shows how the rover velocity increases as a function of the adaptiveness with a small penalty in the localization error. This analysis extrapolates the benefit of using adaptive SLAM into a potential planetary mission.

7.2 Lessons Learned

The goal of this thesis is to design, develop and evaluate an adaptive localization and mapping solution driven by the demands of a planetary rover. Several lessons have been learned during this journey.

- Dead reckoning. When possible, mitigate errors at the dead reckoning level. A bad dead reckoning process will negatively affect the complete SLAM system with catastrophic results. This applies to the Attitude and Heading Reference System, the wheel odometry and the visual odometry.

- Wheel odometry. A 3D model has pros and cons. Mobile robots with a simple locomotion system, e.g. skid robots, do not justify the development of a more complex 3D odometry approach; especially when the robot only navigates on flat surfaces. A 3D odometry primarily captures the 6 DoF delta displacement induced by the locomotion system but the justification for its development requires a minimum of complexity in the locomotion system as well as by the unevenness on the terrain. Planetary rovers definitely justify the approach as evaluated in Chapter 4.

- Data is king. The information gathered by the robotic platform in general and the locomotion system in particular, is an effective manner to achieve graph SLAM sparsity. The process requires machine learning techniques, such as the GP regression described in Chapter 5. This will be a key aspect for long-term SLAM systems of the future.

- SLAM scalability. Graph sparsity is a fundamental property for the accuracy of the final SLAM solution. Table 5.2 shows that with the same number of frames and keyframes the final error can be reduced by selecting highly informative nodes. This emphasizes the importance of graph node selection or sparsity for adaptive SLAM.

- Dense perception. Sparse SLAM should not prevent a dense perception. Dense map and 3D world reconstructions are required for path planning and science operations. The combination of dense perception and sparse optimization is the desired solution, see Chapter 6.

- Loop closure. The number of loop closures is affected by the adaptiveness in the SLAM. Adaptive SLAM is capable of closing the loop several times in the Mars-like terrain. The number of detected loops is influenced by the adaptivity threshold as shown in Table 6.1. However, perceptual aliasing and sensor orientation, as shown in Fig. 6.19, becomes the big issue due to the surroundings, especially when the rover is only equipped with a stereo camera pair.

  Mars-like terrain experiments show that the adaptivity threshold affects the loop closing. However, the threshold does not play a prominent role when the perceptual aliasing is a main factor. The Decos terrain experiment in Section 6.4.2 shows that adaptiveness does not play a prominent role when there is aliasing. Perceptual aliasing is primarily affected by the environment and adaptivity only emphasizes its impact. Decos environment shows a perceptual aliasing which is not affected by any of the adaptive SLAM schemes. In this context active perception can minimize perceptual ambiguities as described in the work by Forster et al. (2014).

- SLAM needs robotics as much as robotics needs SLAM. Robots enrich the SLAM solution by the information gained from their interaction with the environment. A robotic platform complements a typical visual SLAM approach. This is a key issue, since under bad conditions, both visual odometry or wheel odometry perform equally badly. Vision is very sensitive to texture, image blur and light conditions, and the locomotion system is sensitive to the traction performance. A rover is a perfect platform to combine both approaches, and adaptive SLAM is presented in this thesis as a feasible, robust and scalable approach.

- System evaluation. A SLAM system needs to be evaluated at application level. The performance and requirements of a SLAM system are diverse from application to application. The design of a SLAM solution for a planetary rover is a good exercise.

- The future of space exploration will depend on the performance of modern SLAM systems. Autonomous robots in space will require SLAM in order to work for longer runs in outdoor environments.

- Learn to adapt. Adaptation, ignorance and awareness based on prior robot knowledge increase the robustness of a SLAM system. This thesis investigates adaptation, showing a small step ahead in long-term SLAM. Learning to adapt reduces parameter tuning and increases the lifelong of the system.

- The limit of learning. SLAM should not become dogmatic about learning everything, but pragmatic in its use. Learning is a general term which involves an optimization problem. The right question is what to optimize using learning methods. This thesis proposes that when there is a formalization of the problem, machine learning should not substitute it. It means that machine learning should only overcome the limitation of the mathematical model to adapt to the real word and it should never substitute the complete formalization. Otherwise, machine learning becomes hard to address challenges, difficult to detect failures and the number of data required to learn the optimization function increase considerably, making the learning process unmanageable.

7.3 Future Work

We are at the beginning of the era of autonomous robots, an era where intelligent machines will coexist with humans at work, at home and to explore the universe. The development is based on three pillars: data, artificial intelligence and classical robotics. This thesis combines those pillars with the belief that a modern SLAM system will be one day on board a planetary rover. There are very interesting following lines of research for future development.

- Regarding Attitude and Heading Reference Systems (AHRS), improvements in the direction of dynamically fusing magnetic information are desired for future work. The research will be more relevant in terrestrial application, since celestial bodies not always have a reliable magnetic field that could serve for navigation purposes. Some recent research in this direction is in Christensen et al. (2017).

- In the context of wheel odometry, the estimation of how much each contact point contributes to the rover movement can be significantly improved in the future. The contribution of each contact point is currently estimated using a quasi-static calculation of reaction forces. An improvement of the 3D odometry is expected by incorporating force-torque sensor at the contact point. However, most existing planetary rovers lack such sensors in the wheels due to mass and power restrictions. It is foreseen to have such sensors in the future when they become smaller, lighter and more efficient.

- A Gaussian Process estimation of odometry errors might have several applications in robotics, not only for localization and mapping. The odometry error model can assist rover motion control, path planning and online terrain characterization in remote environments. Future techniques to improve the accuracy of the model include correlating the predictions with visual odometry from the cameras instead of using offline ground truth data. The technique combines well with image segmentation in the direction of semantic SLAM which would make the necessary solution for on board traversability prediction in planetary rovers. Recent work with GPs in this direction is done by Cunningham et al. (2017).

- Adaptive SLAM. In this thesis the quadratic equations for visual odometry adaptivity and graph node selection are studied. Other form of equations is interesting to consider in future work.

- Semantic reasoning. Open issues, such as robust loop closure will require a semantic understanding of the scene. High level object recognition, geometry and semantics will be part of future SLAM. This might eventually mitigate the penalty of perceptual aliasing induced by the environment.

- Long-term SLAM. Learning techniques, such as Bayesian inference or deep learning, exploit the information sensed by the rover. Deep learning brings robustness in the future of perception systems. Online learning and adaptiveness will be essential to the new perception era.

- Event-based vision will entail a significant change in visual localization and SLAM. Recent development in Dynamic Vision Sensors (DVS) will allow visual odometry in the harshest environments with a higher level of robustness. DVS have faster response, lower latency, do not suffer image blur and automatically adapt to different lighting conditions. The sensor works based on events instead of frames, imitating the visual system of animals. This characteristic brings a new concept to perform localization and mapping based on events. The SLAM community is already adapting to the new paradigm, Censi and Scaramuzza (2014); Rebecq et al. (2017). There is substantial research to pursue in the field. The adaptation of the technology into the aerospace sector is an exciting development with great potential.

- Active SLAM. A coupled SLAM system between perception, optimization and control of the robot is still far from providing actionable and compact models of the environments as used by humans. This remains an open question in this line of research. An inquisitive reader might find recent results using quadrotors in Vidal et al. (2017).

Appendix A

ExoMars Testing Rover - ExoTeR

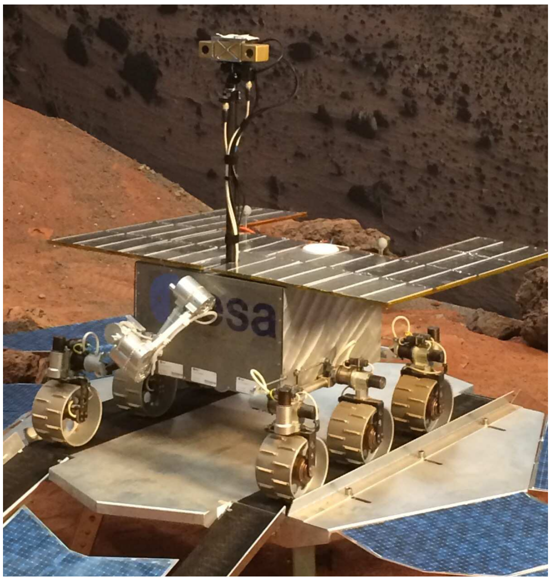


Figure A.1: ExoMars Testing Rover at the Planetary robotics laboratory in 2015

The rover platform used for most of the described experiments is an ExoMars-like scaled down laboratory prototype. The ExoMars Testing Rover (ExoTeR) mimics the locomotion configuration of ExoMars (according to its design in 2007), a.k.a. triple-bogie passive suspension, with a parallelogram structure on top of each bogie Fig. A.1 illustrates the locomotion system of ExoTeR together on the descent module. The locomotion system comprises 6 wheels and 16 actuated joints, more precisely, 6 driving, 4 steering and 6 deployment (or walking) motors. Motion control electronics are a network of servo-drives, namely Elmo Whistles, connected in a CAN Bus together with the On Board Computer (OBC). A driver module in the OBC acts as a CAN Master implementing the CANOpen protocol and sends joint commands timely synchronised to perform a certain locomotion maneuver. Each servo-drive takes care of the close-loop control of one active joint to reach the commanded (position and/or velocity) set point.

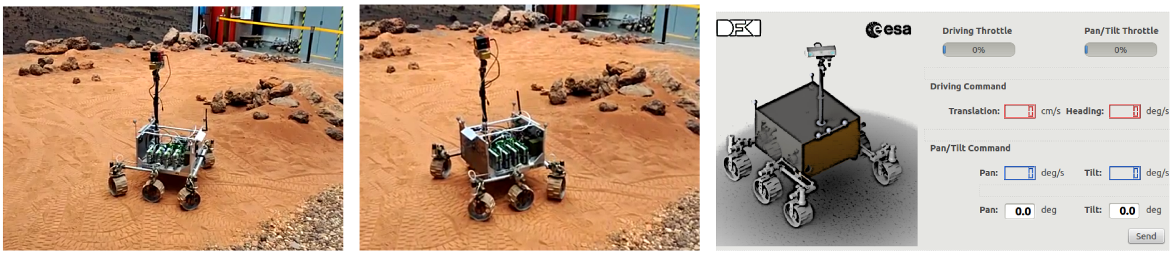
The rover is equipped with solar panels, a stereo camera, a 5 DoF robotic arm and the necessary onboard electronics. The general dimensions of the rover are summarised in Table B.1.

The power electronics are mainly comprised with the Elmo drivers with a weight of 5 kg including the batteries. The computer is based on a PC-104 standard, equipped with several extension modules. ExoTeR has a wireless connection, Solid State Harddisk, a Trimble GPS module receiver. The stereo camera setup has a baseline of 12 cm with a resolution of 1024 × 768 pixels per image. ExoTeR runs the Rock [1] real-time framework on a Linux operating system installed on a Core2 Duo at 1.86 GHz. Fig. A.2 shows ExoTeR performing a point-turn maneuver and the control user interface.

Table A.1: ExoTeR technical data

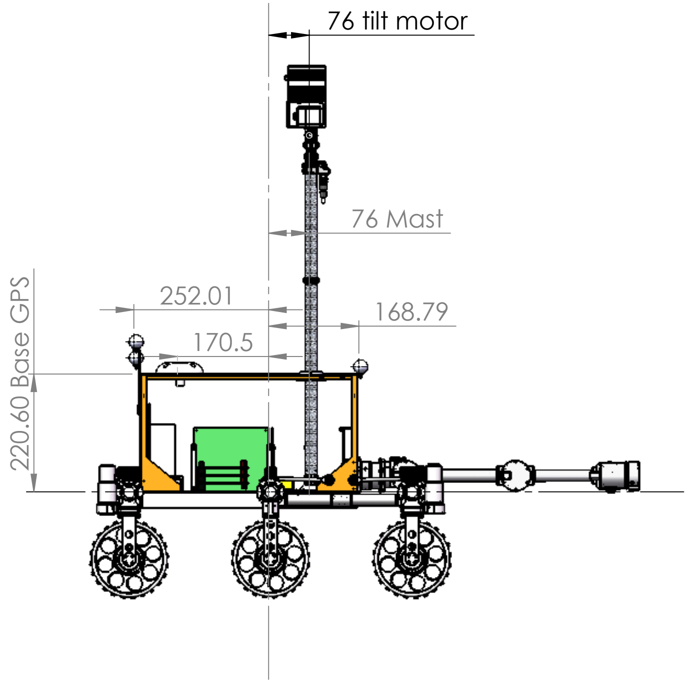
Total Mass	24.08 kg
Dimensions	L 0.7 × W 0.7 × H 0.4 m
Chassis	tripple-bogie passive
Locomotion	6 DC traction motors, 4 DC steering, 6 DC wheel walking
Wheel diameter	15.0 cm
Wheel width	9.0 cm
Computer	Embedded PC Core2 Duo
System	Ubuntu x86 + Rock
Power supply	Lithium Polymer Batteries
Camera	Bumblebee 2
3D	TOF Mesa Swissranger SR4000
IMU	Sensonor STIM300

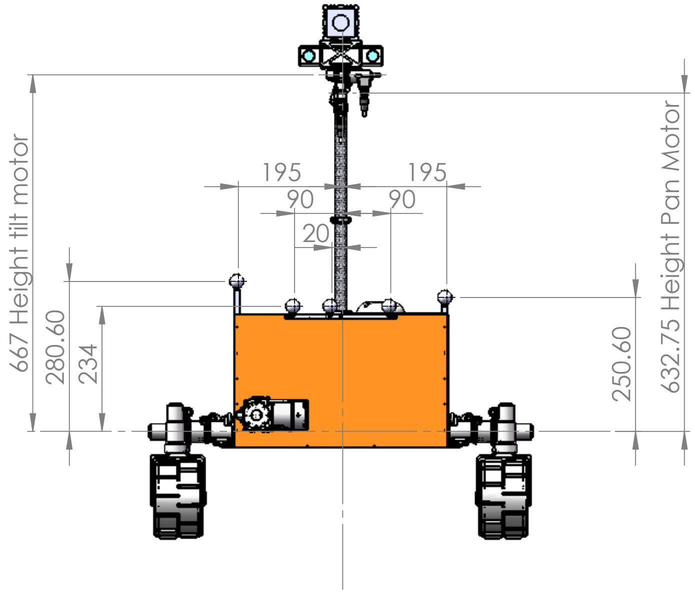
[1] The Robot Construction Kit (Rock) http://www.rock-robotics.org

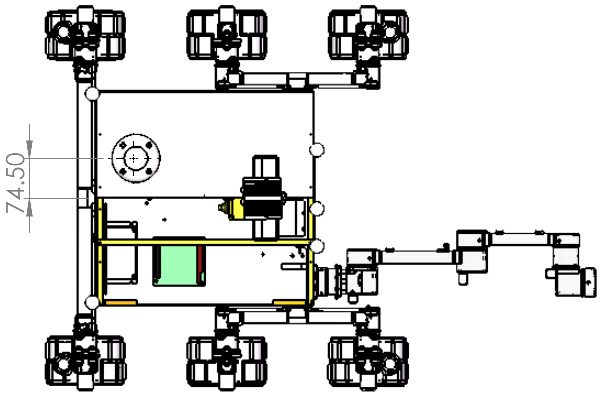


(a) (b) (c)

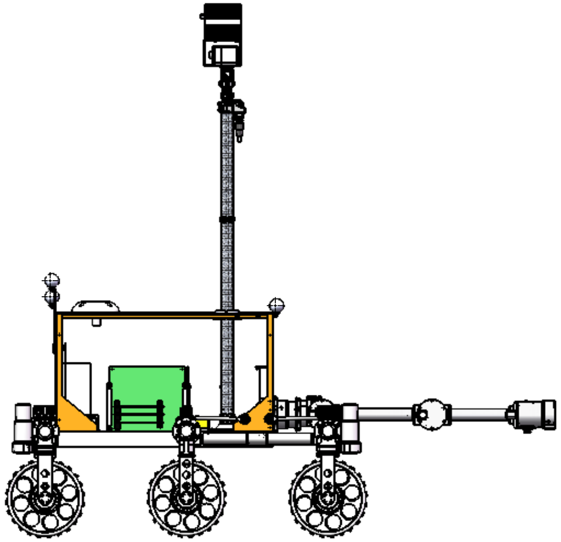
Figure A.2: ExoTeR at the Mars-like terrain in the Planetary robotics laboratory. (a) and (b) sequence of a point-turn trajectory. (c) graphical user interface of the control station.

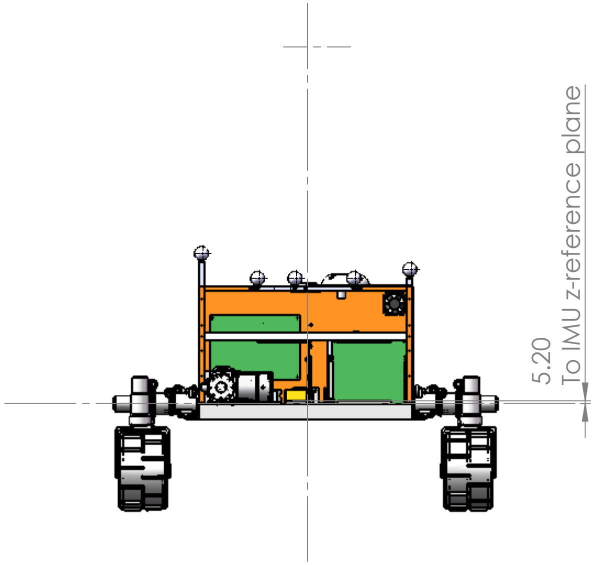


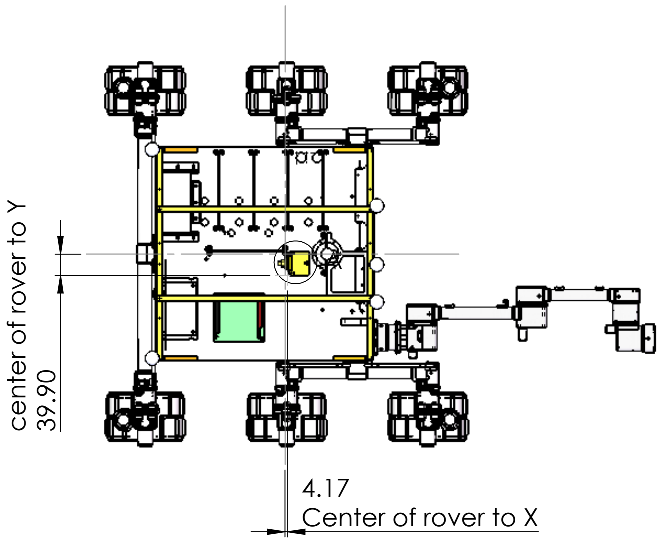


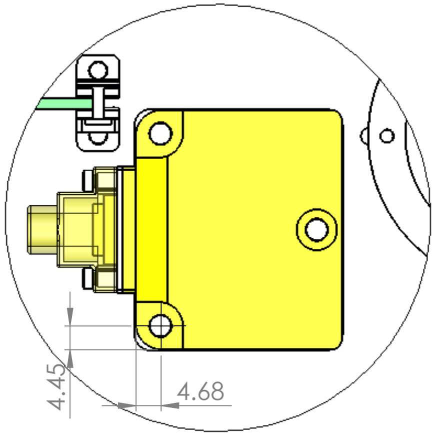


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:		DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
				ESA	
	NAME	SIGNATURE	DATE	TITLE:	
DRAWN	RN			ExoTeR	
CHK'D					
APPV'D					
MFG					
Q.A			MATERIAL:	DWG NO. dim sensors	A3
				SCALE:1:1	SHEET 1 OF 3

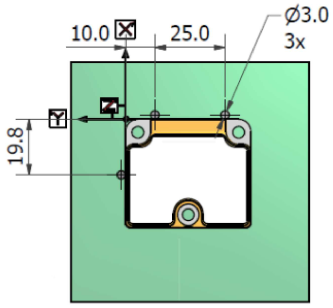








Location X and Y



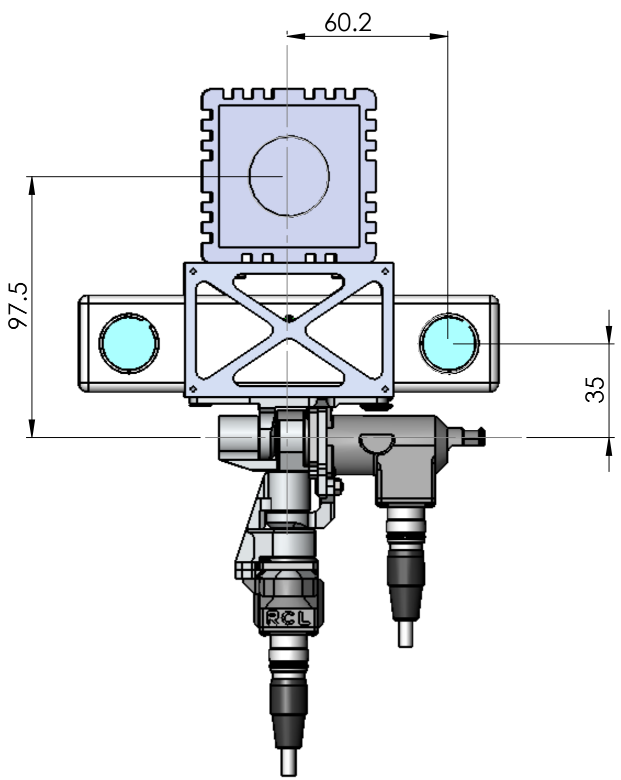
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
			ESA	

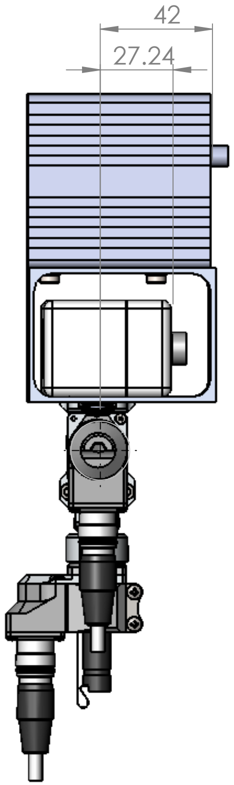
	NAME	SIGNATURE	DATE			
DRAWN	RN					
CHK'D						
APPV'D						
MFG						
Q.A				MATERIAL:		

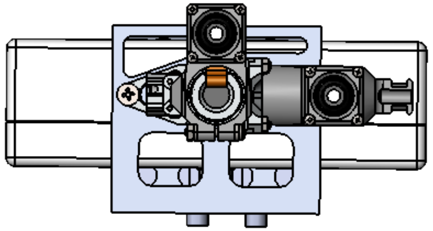
TITLE:

ExoTeR

DWG NO.	dim sensors	A3
SCALE:1:1	SHEET 2 OF 3	







UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:				DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION ESA
	NAME	SIGNATURE	DATE				TITLE:
DRAWN	RN						ExoTeR
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:			DWG NO. dim sensors A3
							SCALE:1:1 SHEET 3 OF 3

Appendix B

Advanced Security Guard - Asguard

Asguard III is a mobile robot with 4 actuated legged wheels and a passive body degree of freedom, Fig. B.1. The wheel shape allows the robot to negotiate uneven terrain and climb stairs. This robot is based on the Asguard II platform that was equipped with additional sensors and a very powerful onboard computer. The sensor head features a laser scanner, a stereo camera system and an inertial measurement unit.

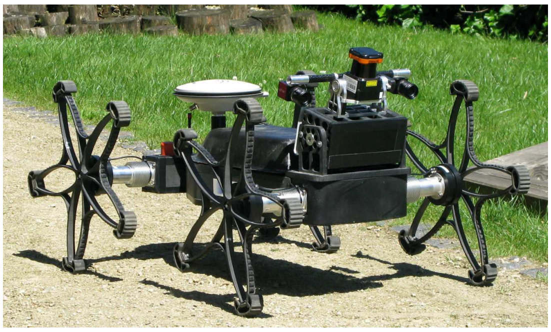
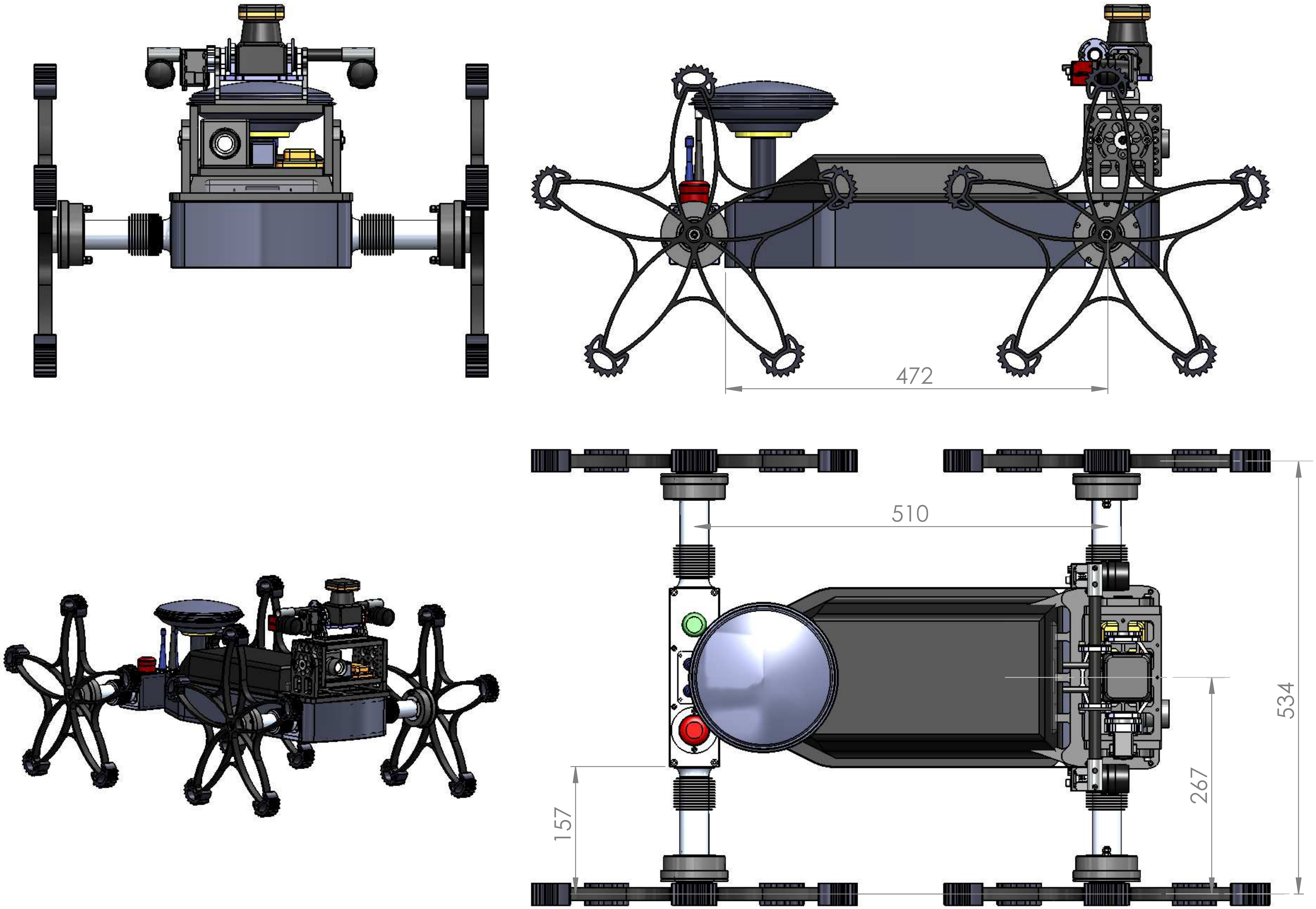


Figure B.1: Asguard III at the DFKI test track.

Table B.1: Asguard III technical data

Total Mass	14.0 kg
Dimensions	L 0.54 × W 0.53 × H 0.9 m
Chassis	legged wheels
Locomotion	4 × 80 watts Faulhaber DC motors – planetary gear 46:1
Wheel diameter	42.0 cm
Wheel width	4.0 cm
Computer	Embedded PC Core2 Duo
System	Ubuntu x86 + Rock
Power supply	Lithium Polymer Batteries
Camera	2 × Guppy F-036C / objective: TS4124-4mm Pentax
3D	Hokuyo UTM-30LX Laser Scanner
IMU	Sensonor STIM300


472
510
534
267
157

List of Figures

List of Tables

Bibliography

Abbeel, P., Coates, A., Montemerlo, M., Ng, A. Y., and Thrun, S. (2005). Discriminative Training of Kalman Filters. *Proceedings of Robotics: Science and Systems*.

Aggarwal, P., Syed, Z., Niu, X., and El-Sheimy, N. (2008). A standard testing and calibration procedure for low cost MEMS inertial sensors and units. *The Journal of Navigation*, 61(02):323–336.

Ali, K. S., Vanelli, C. A., Biesiadecki, J. J., Maimone, M. W., Cheng, Y., Martin, M. S., and Alexander, J. W. (2005). Attitude and Position Estimation on the Mars Exploration Rovers. In *IEEE Conference on Systems, Man, and Cybernetics*.

Allan, D. (1966). Statistics of atomic frequency standards. *Proceedings of the IEEE*, 54(2):221–230.

Allan, D. (1987). Time and Frequency (Time-Domain) Characterization, Estimation, and Prediction of Precision Clocks and Oscillators. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 34(6):647–654.

Allan, D. and Barnes, J. (1981). A Modified "Allan Variance" with Increased Oscillator Characterization Ability. In *Thirty Fifth Annual Frequency Control Symposium*, pages 470–475. IEEE.

Allouis, E., Jorden, T., Patel, N., Senese, S., Magnani, P., Spörri, R., Kapellos, K., Ag, R. A., and Rudolfspoerriruagcom, E. (2010). End-to-end Design of a Robotic System for Collecting and Transferring Samples on Mars Automation and Robotics Section , ESA 2 The Mars Surface Sample Transfer and Manipulation. In *isairas*, pages 178–185.

Antonelli, G., Chiaverini, S., and Fusco, G. (2005). A Calibration Method for Odometry of Mobile Robots Based on the Least-Squares Technique : Theory. *IEEE Transactions on Robotics*, 21(5):994–1004.

Aron, J. (2016). China's Jade Rabbit moon rover dead after 31 months on surface. New Scientist.

Arvidson, R. E., Iagnemma, K. D., Maimone, M., Fraeman, A. A., Zhou, F., Heverly, M. C., Bellutta, P., Rubin, D., Stein, N. T., Grotzinger, J. P., and Vasavada, A. R. (2016). Mars Science Laboratory Curiosity Rover Megaripple Crossings up to Sol 710 in Gale Crater. *Journal of Field Robotics*, pages 495 – 518.

Astrobotic (2017). https://www.astrobotic.com.

Azkarate, M., Zwick, M., Hidalgo-Carrio, J., Nelen, R., Wiese, T., Poulakis, P., Joudrier, L., and Visentin, G. (2015). First Experimental Investigations on Wheel-Walking for Improving Triple-Bogie Rover Locomotion Performances. In *Symposium on Advanced Space Technologies for Robotics and Automation*.

Backes, P., Diaz-Calderon, a., Robinson, M., Bajracharya, M., and Helmick, D. (2005). Automated rover positioning and instrument placement. *2005 IEEE Aerospace Conference*, pages 60–71.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous Localization and Mapping - SLAM: Part II. *IEEE Robotics and Automation Magazine*, pages 108–117.

Bajracharya, M., Maimone, M. W., and Helmick, D. (2008). Autonomy for Mars Rovers : Past, Present and Future. *IEEE Computer Society*.

Bakambu, J. N., Langley, C., Pushpanathan, G., Maclean, W. J., Mukherji, R., and Corporation, M. D. A. (2010). Planetary Rover Visual Motion Estimation improvement for Autonomous , Intelligent , and Robust Guidance , Navigation and Control 2 Test Scenario and Field Test Terrain. *Scenario*, pages 552–559.

Baumgartner, E. T. (2001). Rover localization results for the FIDO rover. In *Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, volume 4571, pages 34–44. Spie.

Baumgartner, E. T., Bonitz, R. G., Melko, J. P., Shiraishi, L. R., and Leger, P. C. (2004). The Mars Exploration Rover Instrument Positioning System. In *IEEE Conference on Aerospace*, pages 1–19.

Biesiadecki, J. and Maimone, M. (2006). The Mars Exploration Rover Surface Mobility Flight Software: Driving Ambition. In *2006 IEEE Aerospace Conference*, pages 1–15. IEEE.

Biesiadecki, J. J., Leger, P. C., and 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.

BillurBarshan and Durrant-Whyte, H. F. (1995). Inertial Navigation Systems for Mobile Robots - B.Barshan, D.Durrant-Whyte [IEEE Trans. on Robotics & Automation].pdf. *IEEE Transactions on Robotics*, 11:328–342.

Boeing (1971). Lunar Roving Vehicle - Operations Handbook. Technical report, Boeing, LRV Systems Engineering.

Boeing (1972). Lunar Rover Vehicle. Technical report, Boeing, Apollo XVII.

Borenstein, J. and Feng, L. (1996a). Measurement and Correction of Systematic Odometry Errors in Mobile Robets. *IEEE Transactions on Robotics*, 12:869–880.

Borenstein, J. and Feng, L. (1996b). Measurement and Correction of Systematic Odometry Errors in Mobile Robots. *IEEE Transactions on Robotics*, 12:869–880.

Brooks, R. (1985). Visual map making for a mobile robot. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 824–829. Institute of Electrical and Electronics Engineers.

Brown, R. G. and Hwang, P. Y. C. (2012). *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*. John Wiley & Sons.

Bry, A., Bachrach, A., and Roy, N. (2012). State estimation for aggressive flight in GPS-denied environments using onboard sensing. In *2012 IEEE International Conference on Robotics and Automation*, pages 1–8. IEEE.

Carlone, L., Kira, Z., Beall, C., Indelman, V., and 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)*, pages 4290–4297. IEEE.

Carrier, W. D. (1992). Soviet rover systems. In *Space Programs and Technologies Conference*, Reston, Virigina. American Institute of Aeronautics and Astronautics.

Casarrubias-Vargas, H., Petrilli-Barcelo, A., and Bayro-Corrochano, E. (2010). EKF-SLAM and Machine Learning Techniques for Visual Robot Navigation. In *2010 20th International Conference on Pattern Recognition*, pages 396–399. IEEE.

Censi, A. and Scaramuzza, D. (2014). Low-latency event-based visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–710. IEEE.

Chandrayaan (2017). http://www.isro.gov.in/pslv-c11-chandrayaan-1. Asian Scientist.

Christensen, L., Krell, M. M., and Kirchner, F. (2017). Learning Magnetic Field Distortion Compensation for Robotic Systems. *In Intelligent Robots and Systems (IROS)*.

Civera, J., Davison, A., and Montiel, J. (2008). Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945.

Clemente, L., Davison, A., and Reid, I. (2007). Mapping Large Loops with a Single Hand-Held Camera. *Science and Systems*.

Cooper and B. (1998). Driving on the Surface of Mars Using the Rover Control Workstation.

Crassidis, J. L. and Markley, F. L. (2003). Unscented filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 26(4):536–542.

Creamer, G. (1996). Spacecraft attitude determination using gyros and wide field-of-view star cameras. In *Guidance, Navigation, and Control Conference*, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics.

Cummins, M. and 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., and Whittaker, W. L. (2017). Locally-Adaptive Slip Prediction for Planetary Rovers Using Gaussian Processes. In *International Conference on Robotics and Automation (ICRA)*.

Dong-Si, T.-C. and Mourikis, A. I. (2011). Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In *2011 IEEE International Conference on Robotics and Automation*, pages 5655–5662. IEEE.

Doucet, A., de Freitas, N., Murphy, K., and Russell, S. (2000). Rao-blackwellised Particle Filtering for Dynamic Bayesian Networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, pages 176–183, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Durrant-Whyte, B. Y. H., Bailey, T. I. M., Cheeseman, P., Crowley, J., and Durrant, H. (2006). Simultaneous Localization and Mapping : Part I. *Robotics*.

Durrant-Whyte, H., Rye, D., and Nebot, E. (1995). Localisation of automatic guided vehicles. In *The 7th International Symposium on Robotics Research (ISRR)*. Springer Verlag.

Eisenman, A., Liebe, C., and Perez, R. (2002). Sun sensing on the Mars exploration rovers. In *IEEE Conference in Aerospace*, volume 5, pages 2249 – 2262. IEEE.

Eliazar, A. and Parr, R. (2004). DP-SLAM 2.0. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, pages 1314–1320 Vol.2. IEEE.

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

ESA (2016). *CDF Study Report: MarsFast, Assesment of an ESA Fast Mobility Mars Rover*. ESA Concurrent Design Facility, TEC-SYE.

Exploration, I. S. and Group, C. (2011). The Global Exploration Roadmap. Technical Report September, International Space Exploration Coordination Group.

Fondahl, K., Kuehn, D., Beinersdorf, F., Bernhard, F., Grimminger, F., Schilling, M., Stark, T., and Kirchner, F. (2012). An adaptive sensor foot for a bipedal and quadrupedal robot. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 270–275. IEEE.

Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2015). IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In *Robotics: Science and Systems*.

Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2017). On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. *IEEE Transactions on Robotics*, 33(1):1–21.

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicle.

Furgale, P., Enright, J., and Barfoot, T. (2011). Sun Sensor Navigation for Planetary Rovers : Theory and Field Testing. *IEEE Transactions On Aerospace And Electronic Systems*, 47(3).

Galvez-Lopez, D. and Tardos, J. D. (2012). Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197.

Gao, Y. (2016). *Contemporary Planetary Robotics, An Approach Toward Autonomous Systems.* Wiley-VCH Verlag GmbH & Co.

Garrido-Jurado, S., Muñoz Salinas, R., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292.

Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient Large-Scale Stereo Matching. In *Asian Conference on Computer Vision (ACCV)*.

Giguère, P. and Dudek, G. (2008). Clustering Sensor Data for Terrain Identification using a Windowless Algorithm. In *Robotics: Science and Systems*.

GPy (since 2012). GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`.

Grisetti, G., Kummerle, R., Stachniss, C., and Burgard, W. (2010a). A Tutorial on Graph-Based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43.

Grisetti, G., Kummerle, R., Stachniss, C., Frese, U., and Hertzberg, C. (2010b). Hierarchical optimization on manifolds for online 2D and 3D mapping. In *2010 IEEE International Conference on Robotics and Automation*, pages 273–278. IEEE.

Grisetti, G., Stachniss, C., and Burgard, W. (2007a). Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, 23(1):34–46.

Grisetti, G., Tipaldi, G. D., Stachniss, C., Burgard, W., and Nardi, D. (2007b). Fast and accurate SLAM with Rao–Blackwellized particle filters. *Robotics and Autonomous Systems*, 55(1):30–38.

Han, S. and Knight, N. (2009). Using Allan Variance to Determine the Calibration Model of Inertial Sensors for GPS/INS Integration. In *6th International Symposium on Mobile Mapping Technology, Presidente Prudente*, pages 1–8, Sao Paulo, Brazil.

Helmick, D., Cheng, Y., and Roumeliotis, S. (2004a). Path following using visual odometry for a Mars rover in high-slip environments. *IEEE Aerospace Conference Proceedings*, 2:772–789.

Helmick, D. M., Cheng, Y., Clouse, D. S., and Matthies, L. H. (2004b). Path Following using Visual Odometry for a Mars Rover in High-Slip Environments. *IEEE Aerospace Conference Proceedings*.

Hernandez, J., Tsotsos, K., and Soatto, S. (2015). Observability, identifiability and sensitivity of vision-aided inertial navigation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2319–2325. IEEE.

Hertzberg, C. (2008). *A Framework for Sparse, Non-Linear Least Squares Problems on Manifolds*. PhD thesis, Universität Bremen.

Hesch, J. A., Kottas, D. G., Bowman, S. L., and Roumeliotis, S. I. (2014). Consistency Analysis and Improvement of Vision-aided Inertial Navigation. *IEEE Transactions on Robotics*, 30(1):158–176.

Hidalgo-Carrio, J. (2013). Navigation and Slip Kinematics for High Performance Motion Models. In *Symposium on Advanced Space Technologies in Robotics and Automation*.

Hidalgo-Carrio, J., Arnold, S., and Poulakis, P. (2016). On the Design of Attitude and Heading Reference Systems Using the Allan Variance. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency control*, 63(4):656–65.

Hidalgo-Carrio, J., Babu, A., and Kirchner, F. (2014). Static forces weighted Jacobian motion models for improved Odometry. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 169–175. IEEE.

Hidalgo-Carrio, J. and Cordes, F. (2012). Kinematics Modeling of a Hybrid Wheeled-Leg Planetary Rover. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

Hidalgo-Carrio, J., Hennes, D., Schwendner, J., and Kirchner, F. (2017). Gaussian Process Estimation of Odometry Errors for Localization and Mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore. IEEE.

Hidalgo-Carrio, J., Poulakis, P., Barrientos, A., and Del-cerro, J. (2011). ESTEC Testbed Capabilities for the Performance Characterization of Planetary Rover Localization Sensors - First Results on IMU Investigations. In *ASTRA 2011 - 11th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, pages 1–8. ESA/ESTEC.

Hidalgo-Carrio, J., Poulakis, P., and Kirchner, F. (2018). Adaptive localization and mapping with application to planetary rovers. *Journal of Field Robotics*.

Hidalgo-Carrio, J., Poulakis, P., Köhler, J., Del-Cerro, J., and Barrientos, A. (2012a). Improving Planetary Rover Attitude Estimation via MEMS Sensor Characterization. *Sensors*, 12(2):2219–2235.

Hidalgo-Carrio, J., Schwendner, J., and Kirchner, F. (2012b). Planetary Rover Localization Design: Antecedents and Directions. In *IEEE Intelligent Vehicles Symposium Workshops*, pages 1 – 8.

Hildebrandt, M., Gaudig, C., Christensen, L., Natarajan, S., Carrio, J. H., Paranhos, P. M., and Kirchner, F. (2014). A Validation Process for Underwater Localization Algorithms. *International Journal of Advanced Robotic Systems*, page 1.

Hildebrandt, M., Gaudig, C., Christensen, L., Natarajan, S., Paranhos, P., and Albiez, J. (2012). Two years of experiments with the AUV dagon - a versatile vehicle for high precision visual mapping and algorithm evaluation. In *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 1–9. IEEE.

Homeister, M., Thaeter, J., Scheper, M., Apeldoorn, J., and Koebel, D. (2010). Nextlunar lander -an opportunity for a close look at the lunar south pole. In *38th COSPAR Scientific Assembly*.

Howard, A. and Seraji, H. (2001). Vision-based terrain characterization and traversability assessment. *Journal of Robotic Systems*, 18(10):577–587.

Howell, E. (2014). China's Yutu rover is still alive, reports say, as lunar panorama released. Universe Today.

Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I. (2009). On the complexity and consistency of UKF-based SLAM. In *2009 IEEE International Conference on Robotics and Automation*, pages 4401–4408. IEEE.

Huntsberger, T., Cheng, Y., Baumgartner, E. T., Robinson, M., Schenker, P. S., and Drive, O. G. (2003). Sensory Fusion for Planetary Surface Robotic Navigation , Rendezvous , and Manipulation Operations. *International Conference on Advanced Robotics*.

IEEE (2001). IEEE Standard for Inertial Sensor Terminology.

IEEE (2008). IEEE Standard Specification Format Guide and Test Procedure for Single- Axis Interferometric Fiber Optic Gyros.

Ila, V., Porta, J., and Andrade-Cetto, J. (2010). Information-Based Compact Pose SLAM. *IEEE Transactions on Robotics*, 26(1):78–93.

Ishigami, G., Pineda, E., Overholt, J., Hudas, G., and Iagnemma, K. (2011). Performance analysis and odometry improvement of an omnidirectional mobile robot for outdoor terrain. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4091–4096. IEEE.

Joyeux, S., Schwendner, J., Kirchner, F., Babu, A., Grimminger, F., Machowinski, J., Paranhos, P., and Gaudig, C. (2011). Intelligent Mobility - Autonomous Outdoor Robotics at the DFKI. *KI - Kuenstliche Intelligenz*.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., and Dellaert, F. (2011a). iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *2011 IEEE International Conference on Robotics and Automation*, pages 3281–3288. IEEE.

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

Kaess, M. and Ranganathan, A. (2008). iSAM: Incremental Smoothing andMapping. *IEEE Transactions on Robotics*, 24(6):1365 – 1378.

Kelly, A. (2004). Linearized Error Propagation in Odometry. *The International Journal of Robotics Research*, 23(2):179–218.

Kemurdjian, A. L. (1990). From the Moon rover to the Mars rover. *The Planetary Report*, 10(4):4 – 11.

Kemurdjian, A. L., Gromov, V. V., Kazhukalo, I. F., Kozlov, G. V., Komissarov, V. I., Korepanov, G. N., Martinov, B. N., Malenkov, V. I., Mitskevich, A. V., and Mishkinyuk, V. K. (1993). Soviet developments of planet rovers in period of 1964 - 1990. *In CNES, Missions, Technologies, and Design of Planetary Mobile Vehicles p 25-43 (SEE N94-23373 06-91)*.

Klein, G. and Murray, D. (2007). Parallel Tracking and Mapping for Small AR Workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10. IEEE.

Ko, J. and 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*, pages 3471–3476. IEEE.

Ko, J., Klein, D. J., Fox, D., and 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*, pages 742–747. IEEE.

Konolige, K. and Agrawal, M. (2008). FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077.

Kottas, D. G., Hesch, J. A., Bowman, S. L., and Roumeliotis, S. I. (2012). On the consistency of Vision-aided Inertial Navigation. In *13th International Symposium on Experimental Robotics*, Quebec City.

Kummerle, R., Grisetti, G., and Burgard, W. (2011a). Simultaneous calibration, localization, and mapping. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3716–3721. IEEE.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011b). G2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE.

Kuroda, Y., Kurosawa, T., Tsuchiya, A., and Kubota, T. (2004). Accurate localization in combination with planet observation and dead reckoning for lunar rover. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, pages 2092–2097 Vol.2. IEEE.

Lambert, A. J. (2011). *Visual Odometry Aided by a Sun Sensor and an Inclinometer*. PhD thesis, University of Toronto.

Lamon, P. and Siegwart, R. (2007). 3D Position Tracking in Challenging Terrain. *The International Journal of Robotics Research*, 26(2):167–186.

Laxman, S. (2012). Chang'e-3: China To Launch First Moon Rover In 2013. Asian Scientist.

Lefferts, E. J. (1982). Kalman Filtering for Spacecraft Attitude Estimation. In *AIAA 20th Aerospace Sciences Meeting*, Orlando, Florida.

Leonard, J. and Durrant-Whyte, H. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, pages 1442–1447. IEEE.

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

Liggins, M. E., Hall, D. L., and Llinas, J. (2009). *Handbook of Multisensor Data Fusion Theory and Practice*. CRC Press.

Maimone, M., Biesiadecki, J., Tunstel, E., Cheng, Y., and Leger, C. (2006). Surface Navigation and Mobility Intelligence on the Mars Exploration Rovers. In *Intelligence for Space Robotics*, chapter chapter 3, pages 45–69. TSI Press, San Antonio, TX, tsi press edition.

Maimone, M., Cheng, Y., and 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., and 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*, pages 1 – 8. IEEE Press.

Makarenko, A. A., Williams, S. B., Bourgault, F., and Durrant-Whyte, H. F. (2002). An experiment in integrated exploration. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 1, pages 534–539. IEEE.

Malenkov, M. I. (2013). Creation of Lunokhod-1 as an outstanding scientific and technological achievement of the XX century. *Solar System Research*, 47(7):610–617.

Martinelli, A. (2012). Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination. *IEEE Transactions on Robotics*, 28(1):44–60.

Matthies, L., Gat, E., Harrison, R., Wilcox, B., Volpe, R., and Litwin, T. (1995). Mars microrover navigation: performance evaluation and enhancement. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 433–440. IEEE Comput. Soc. Press.

Meier, F., Hennig, P., and Schaal, S. (2014). Incremental Local Gaussian Regression. In *Advances in Neural Information Processing Systems*, pages 972–980.

Merwe, R. V. D., Wan, E. A., and Julier, S. (2004). Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation. In *Proceedings of the AIAA Guidance, Navigation & Control Conference*, pages 2004–5120.

Mishkin, A., Morrison, J., Nguyen, T., Stone, H., Cooper, B., and Wilcox, B. (1998). Experiences with operations and autonomy of the Mars Pathfinder Microrover. In *IEEE Aerospace Conference Proceedings (Cat. No.98TH8339)*, volume 2, pages 337–351. IEEE.

Mitchell, H. B. (2007). *Multi-Sensor Data Fusion: An Introduction*. Springer Publishing Company, Incorporated, 1st edition.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Eighteenth National Conference on Artificial Intelligence*, pages 593–598, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Montemerlo, M., Thrun, S., Roller, D., and 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*, pages 1151–1156.

Moravec, H. and Elfes, A. E. (1985). High Resolution Maps from Wide Angle Sonar. In *IEEE International Conference on Robotics and Automation*, pages 116 – 121.

Mourikis, A. I. and 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*, pages 1–8. IEEE.

Mourikis, A. I., Roumeliotis, S. I., Anastasios Mourikis, and Roumeliotis, S. I. (2007a). A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565 – 3572. Dept. of Computer Science and Engineering. University of Minnesota, IEEE.

Mourikis, A. I., Roumeliotis, S. I., and Burdick, J. W. (2007b). SC-KF Mobile Robot Localization: A Stochastic Cloning Kalman Filter for Processing Relative-State Measurements. *IEEE Transactions on Robotics*, 23(4):717–730.

Moutarlier, P. (1991). *Modélisation autonome de l'environnement par un robot mobile*. PhD thesis, Toulouse.

Muir, P. F. and Neuman, C. P. (1986). Kinematic Modeling of Wheeled Mobile Robots. Technical report, CamegieMellon University.

Mukadam, M., Yan, X., and Boots, B. (2016). Gaussian Process Motion planning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9–15. IEEE.

Muller, J., Paul, O., and Burgard, W. (2012). Probabilistic velocity estimation for autonomous miniature airships using thermal air flow sensors. In *2012 IEEE International Conference on Robotics and Automation*, pages 39–44. IEEE.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163.

NASA (2017). Where is Curiosity?: https://mars.nasa.gov/msl/mission/whereistherovernow.

NASA, M. E. R. (2004). https://www.nasa.gov/mission_pages/mer/index.html.

NASA, M. S. L. (2015). https://www.nasa.gov/mission_pages/msl/index.html.

N.El-Sheimy, Hou, H., and Niu, X. (2008). Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149.

Northrop-Grumman Corporation (2000). LN-200 Fiber Optic Inertial Measurement Unit. Technical report.

Ojeda, L. and Borenstein, J. (2002). FLEXnav : Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain BY. In *IEEE International Conference on Robotics and Automation*, pages 1–6.

Ojeda, L. and Borenstein, J. (2004). Methods for the Reduction of Odometry Errors in Over-Constrained Mobile Robots. *Autonomous Robots*, pages 273–286.

Ojeda, L., Cruz, D., Reina, G., Borenstein, J., and Member, S. (2006). Current-Based Slippage Detection and Odometry Correction for Mobile Robots and Planetary Rovers. *IEEE Transactions on Robotics*, 22(2):366–378.

Park, K. (1996). Dead reckoning navigation of a mobile robot using an indirect Kalman filter. *1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems (Cat. No.96TH8242)*, pages 132–138.

Peretroukhin, V., Vega-Brown, W., Roy, N., and Kelly, J. (2016). PROBE-GK: Predictive robust estimation using generalized kernels. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 817–824. IEEE.

Pirker, K., Ruther, M., and Bischof, H. (2011). CD SLAM - continuous localization and mapping in a dynamic world. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3990–3997. IEEE.

Poppinga, J., Birk, A., and Pathak, K. (2010). A Characterization of 3D Sensors for Response Robots. In *RoboCup 2009: Robot Soccer World Cup XIII*, pages 264–275. Springer, Berlin, Heidelberg.

Poulakis, P., Vago, J. L., Loizeau, D., Vicente-Arevalo, C., A.Hutton, McCoubrey, R., Arnedo-Rodriguez, J., Smith, J., Boyes, B., Jessen, S., Otero-Rubio, A., Durrant, S., Gould, G., Joudrier, L., Yushtein, Y., Alary, C., Zekri, E., Baglioni, P., Cernusco, A., Maggioni, F., Yague, R., and Ravera, F. (2015). Overview and development status of the Exomars rover mobility subsystem. In *Advanced Space Technologies for Robotics and Automation*, Noordwijk, The Netherlands.

Powell, M. W., Crockett, T., Fox, J. M., Joswig, J. C., Norris, J. S., and Rabe, K. J. (2006). Targeting and Localization for Mars Rover Operations. In *IEEE International Conference on Information Reuse and Integration*, pages 23–27.

Qualysis (2017). https://www.qualisys.com/applications/sports/swimming/.

Quinchia, A., Ferrer, C., Falco, G., and Dovis, F. (2013a). Constrained Non-linear Fitting for Stochastic Modeling of Inertial Sensors. In *Design and Architectures for Signal and Image Processing (DASIP), 2013 Conference on*, pages 119–125.

Quinchia, A. G., Falco, G., Falletti, E., Dovis, F., and Ferrer, C. (2013b). A comparison between different error modeling of MEMS applied to GPS/INS integrated systems. *Sensors (Basel, Switzerland)*, 13(8):9549–88.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.

Rebecq, H., Horstschaefer, T., Gallego, G., and Scaramuzza, D. (2017). EVO: A Geometric Approach to Event-Based 6-DOF Parallel Tracking and Mapping in Real Time. *IEEE Robotics and Automation Letters*, 2(2):593–600.

Rehder, J., Gupta, K., Nuske, S., and Singh, S. (2012). Global pose estimation with limited GPS and long range visual odometry. *2012 IEEE International Conference on Robotics and Automation*, pages 627–633.

Rehrmann, F., Schwendner, J., Cornforth, J., Durrant, D., Lindegren, R., Selin, P., Hidalgo-Carrio, J., Poulakis, P., and Köhler, J. (2011). A Miniaturised Space Qualified MEMS IMU for Rover Navigation Requirements and Testing of a Proof of Concept Hardware Demonstrator. In *Advanced Space Technologies for Robotics and Automation*, number 1, pages 1 – 8.

Reinstein, M., Kubelka, V., and Zimmermann, K. (2013). Terrain adaptive odometry for mobile skid-steer robots. In *2013 IEEE International Conference on Robotics and Automation*, pages 4706–4711. IEEE.

Requirements & Standards Division, E. (2008). *Space segment operability*. European Cooperation for Space Standardization, ECSS, Noordwijk, The Netherlands.

Richter, L., Apfelbeck, M., Claasen, F., Haarmann, R., Hofmann, P., Klinkner, S., and Schwendner, J. (2011). Mobile Payload Element: Concept Study for a Sample Fetching Rover for the ESA Lunar Lander Mission. In *ISTVS Conference*.

Rogers, R. (2003). *Applied mathematics in integrated navigation systems*. American Institute of Aeronautics and Astronautics, Inc, 2nd edition.

Rogers-Marcovitz, F., George, M., Seegmiller, N., and Kelly, A. (2012). Aiding Off-Road Inertial Navigation with High Performance Models of Wheel Slip. In *IEEE International Conference on Intelligent Robots and Systems*.

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

Rusu, A. (2014). *Path Planning and Autonomous Navigation for a Planetary Exploration Rover*. PhD thesis, University of Toulouse.

Savage, P. G. (2002). Analytical Modeling of Sensor Quantization in Strapdown Inertial Navigation Error Equations. *Journal of Guidance, Control, and Dynamics*, 25(5):833–842.

Schenker, P. (2006). Advances in Rover Technology for Space Exploration. *2006 IEEE Aerospace Conference*, pages 1–23.

Schwendner, J. (2013). *Embodied Localisation and Mapping*. PhD thesis, Universitaet Bremen.

Schwendner, J. and Hidalgo-Carrio, J. (2012). Terrain aided navigation for planetary exploration missions. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 1 – 8.

Schwendner, J. and Joyeux, S. (2010). Classifying Autonomy for Mobile Space Exploration Robots. *International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

Schwendner, J., Joyeux, S., and Kirchner, F. (2013). Using Embodied Data for Localization and Mapping. *Journal of Field Robotics*, pages 263 – 295.

Se, S., Ng, H.-k., Jasiobedzki, P., and Moyung, T.-j. (2004). Vision Based Modeling and Localization for Planetary Exloration Rovers. In *International Astronautical Congress*.

Sibley, G., Matthies, L., and Sukhatme, G. (2010). Sliding window filter with application to planetary landing. *Journal of Field Robotics*, 27(5):587–608.

Siciliano, B. and Khatib, O. (2016). *Handbook of robotics*. Springer-Verlag Berlin Heidelberg.

Stachniss, C., Grisetti, G., and Burgard, W. (2005). Recovering Particle Diversity in a Rao-Blackwellized Particle Filter for SLAM After Actively Closing Loops. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 655–660. IEEE.

Stachniss, C., Grisetti, G., Burgard, W., and Roy, N. (2007). Analyzing gaussian proposal distributions for mapping with rao-blackwellized particle filters. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3485–3490. IEEE.

Suh, Y. S. (2010). Orientation Estimation Using a Quaternion-Based Indirect Kalman Filter With Adaptive Estimation of External Acceleration. *IEEE Transactions on Instrumentation and Measurement*, 59(12):3296–3305.

Tarokh, M. and McDermott, G. J. (2005). Kinematics Modeling and Analyses of Articulated Rovers. *IEEE Transactions on Robotics*, 21(4):539–553.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, intelligen edition.

Titterton, D. and Weston, J. L. (2004). *Strapdown Inertial Navigation Technology*. The Institution of Electrical Engineers, 2nd edition.

Trebi-ollennu, A., Huntsberger, T., Cheng, Y., Baumgartner, E. T., Kennedy, B., and Schenker, P. (2001). Design and Analysis of a Sun Sensor for Planetary Rover Absolute Heading Detection. *IEEE Transactions on Robotics*, 17(6):939–947.

Troni, G. and Whitcomb, L. L. (2013). Preliminary experimental evaluation of a Doppler-aided attitude estimator for improved Doppler navigation of underwater vehicles. In *2013 IEEE International Conference on Robotics and Automation*, pages 4134–4140. IEEE.

Vidal, A. R., Rebecq, H., Horstschaefer, T., and Scaramuzza, D. (2017). Hybrid, Frame and Event based Visual Inertial Odometry for Robust, Autonomous Navigation of Quadrotors. *Computer Vision and Pattern Recognition*.

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

Volpe, R. (1999). Mars rover navigation results using sun sensor heading determination. In *IEEE International Conference on Intelligent Robots and Systems.*, volume 1, pages 460–467. IEEE.

Wang, C. M. (1988). Location estimation and uncertainty analysis for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 1230–1235.

Wang, J. and 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)*, pages 1003–1010. IEEE.

Western-Ontario (2010). Russian Lunar Rover Found: 37-Year-Old Space Mystery Solved. Science Daily.

Wiese, T. (2017). 3D Kinematic Modeling and Evaluation of Rough-Terrain Locomotion Modes for an ExoMars-like Mobility Subsystem.

Williams, B., Klein, G., and Reid, I. (2007). Real-Time SLAM Relocalisation. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE.

Wong, J. Y. (2001). *Theory of Ground Vehicles, 3rd Edition*. Wiley-Interscience.

Wright, M., Jaques, B., Morea, S., and Editor, T. (2002). *A Brief History of the Lunar Roving Vehicle*. NASA, Marshall Space Flight Center.

Xing, Z. and Gebre-Egziabher, D. (2008). Modeling and bounding low cost inertial sensor errors. In *Position, Location and Navigation Symposium, 2008 IEEE*, pages 1122–1132.

Xinhua (2014). Chinese lunar rover alive but weak. SpaceDaily.

Xiyuan, C. (2003). Modeling random gyro drift by time series neural networks and by traditional method. In *International Conference on Neural Networks and Signal Processing*, volume 1, pages 810–813.

Zhang, C. X., Li, Y., Mumford, P., and Rizos, C. (2008). Allan Variance Analysis on Error Characters of MEMS Inertial Sensors for an FPGA-based GPS/INS System. In *Int. Symp. on GPS/GNSS*, pages 127 – 133.