

3D POSITION TRACKING FOR ALL-TERRAIN ROBOTS

THÈSE N° 3192 (2005)

PRÉSENTÉE À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Institut d'ingénierie des systèmes

SECTION DE MICROTECHNIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Pierre LAMON

ingénieur en microtechnique diplômé EPF
de nationalité suisse et originaire d'Icogne (VS)

acceptée sur proposition du jury:

Prof. R. Siegwart, directeur de thèse
Prof. P. Fiorini, rapporteur
Dr S. Lacroix, rapporteur
Prof. B. Merminod, rapporteur

Lausanne, EPFL
2005

Table of Contents

Acknowledgments	5
Abstract	7
Version abrégée	9
1 Introduction	11
1.1 Autonomy in rough terrain	11
1.2 The challenges of rough terrain navigation	12
1.2.1 The lack of prior information	12
1.2.2 Perception	12
1.2.3 Locomotion	13
1.3 About this work	13
1.3.1 Context of the research	13
1.3.2 Contributions and structure of this work	14
2 The SOLERO rover	17
2.1 Introduction	17
2.2 The mechanical design	17
2.3 The control architecture	20
2.3.1 Sensors and actuators	22
2.3.1.1 The I2C modules	22
2.3.1.2 Stereovision	23
2.3.1.3 Omnicam	24
2.3.1.4 Inertial Measurement Unit	25
2.3.2 Software architecture	25
2.3.2.1 Solero3D	26
2.3.2.2 The remote control interface	27

2.4 Conclusion	28
3 3D-Odometry	29
3.1 Introduction	29
3.2 3D-Odometry	29
3.2.1 Bogie displacement.	30
3.2.2 3D displacement	33
3.2.3 The contact angles	35
3.3 Experimental results	36
3.4 Conclusion	40
4 Control in rough terrain	41
4.1 Introduction	41
4.2 Quasi-static model of a wheeled rover	42
4.2.1 Mobility analysis	43
4.2.2 A 3D static model	44
4.3 Torque optimization.	45
4.3.1 Wheel slip model	45
4.3.2 Optimization algorithm.	47
4.3.3 Torque optimization for SOLERO	50
4.4 Rover motion	52
4.5 Experimental results	54
4.5.1 Simulation tools	54
4.5.1.1 Wheel slip.	54
4.5.1.2 Wheel-ground contact angles.	54
4.5.2 Experiments	55
4.5.3 Discussion.	58

4.6	Wheel-ground contact angles.....	58
4.7	Conclusion	60
5	Position tracking in rough terrain.....	61
5.1	Introduction	61
5.2	Sensors for outdoors.....	61
5.3	Uncertainties propagation	64
5.3.1	Coordinate systems and transformations	64
5.3.2	Error propagation	66
5.4	Sensor fusion	67
5.4.1	The sensor models.....	70
5.4.1.1	Inertial unit model	70
5.4.1.2	3D-Odometry measurement model	71
5.4.1.3	VME measurement model	72
5.4.2	State prediction model	72
5.5	Experimental results.....	74
5.5.1	Inertial and 3D-Odometry.....	75
5.5.1.1	Setting the state transition covariance matrix Q	76
5.5.1.2	Setting Rimu Rinc and Rodo	77
5.5.1.3	Experimental validation	78
5.5.1.4	Discussion.....	82
5.5.2	Enhancement with VME.....	82
5.5.2.1	Experimental results.....	83
5.6	Conclusion	87
6	Conclusion and outlook	89
6.1	Conclusions	89
6.2	Outlook.....	90

A Parameters and model of SOLERO	93
A.1 Parts of SOLERO	93
A.2 The bogies.....	94
A.3 The main body and the rear wheel	95
A.4 The front fork	95
A.5 Quasi-static model of SOLERO	96
A.5.1 Linear dependence of the wheel torques	97
A.5.2 Equal torques solution	98
B Linearized equations	99
B.1 Accelerometers model.....	99
B.2 Gyroscopes state transition	99
C The Gauss-Markov process	101
D Visual Motion Estimation	105
Literature	107
Curriculum Vitae	113

Acknowledgments

A doctoral thesis is an adventure during which we meet many different people, ready to help, to give good advice and act as a source of inspiration. First of all, I would like to thank my advisor, Roland Siegwart, to have convinced me to do a doctoral thesis at the Autonomous Systems Laboratory. All this has been possible thanks his positive attitude, good advice and support.

During the thesis, I had the chance to spend several months in other labs. Each time the experience was very positive and stimulating. The first exchange was at CMU, where I discovered the world of linux and autonomy applied to rough terrain rovers. I would like to thank Reid Simmons for having accepted to supervise my work, Sanjiv Singh and Dennis Strelow for their help related to the visual motion estimation, Bart Nabbe for the problems related to informatics and Jianbo Shi for his help with the mathematical derivations. The next two exchanges took place at LAAS. In particular, I would like to thank Simon Lacroix and Anthony Mallet for their help with VME and GENOM.

Most of the student projects related to this research provided very good results, which helped to validate the theory through experiments with SOLERO. I would like to thank Ambroise Krebs for his excellent diploma work, which enabled the development of a new approach to slip minimization in rough terrain. Also, I would like to thank Stéphane Michaud for the development of the mechanical structure of SOLERO and the angular sensors, Martin Nyffenegger for the nice remote control interface, Benoît Dagon for the mechanical design of the panoramic vision system and Gabriel Paciotti for the stereovision support.

The advice of my colleagues were precious and helped to develop and debug the various systems. In particular, I would like to thank Grégoire Terrien and Michel Lauria for their expert advice related to the mechanical aspects, Agostino Martinnelli for the mathematics, Daniel Burnier, Ralph Pigué and Gilles Caprari for the electronics and finally Rolf Jordi and Frédéric Pont for the questions related to informatics. The positive atmosphere in the lab provided favorable conditions for an efficient and constructive work. A special thank to Marie-José Pellaud, Nicola Tomatis, Daniel Burnier and my office-mate Gilles Caprari for their psychological

support. Again, I would like to thank everybody in the lab for the great time I've spent during these four years.

Thanks also to the members of the thesis committee, Simon Lacroix, Paolo Fiorini and Bertrand Merminod, for the careful reading of the thesis and for their constructive feedback.

Finally, I would like to thank all the members of my family for their support and my love, Mati.

Abstract

Rough terrain robotics is a fast evolving field of research and a lot of effort is deployed towards enabling a greater level of autonomy for outdoor vehicles. Such robots find their application in scientific exploration of hostile environments like deserts, volcanoes, in the Antarctic or on other planets. They are also of high interest for search and rescue operations after natural or artificial disasters.

The challenges to bring autonomy to all terrain rovers are wide. In particular, it requires the development of systems capable of reliably navigate with only partial information of the environment, with limited perception and locomotion capabilities. Amongst all the required functionalities, locomotion and position tracking are among the most critical. Indeed, the robot is not able to fulfill its task if an inappropriate locomotion concept and control is used, and global path planning fails if the rover loses track of its position. This thesis addresses both aspects, a) efficient locomotion and b) position tracking in rough terrain.

The Autonomous System Lab developed an off-road rover (Shrimp) showing excellent climbing capabilities and surpassing most of the existing similar designs. Such an exceptional climbing performance enables an extension in the range of possible areas a robot could explore. In order to further improve the climbing capabilities and the locomotion efficiency, a control method minimizing wheel slip has been developed in this thesis. Unlike other control strategies, the proposed method does not require the use of soil models. Independence from these models is very significant because the ability to operate on different types of soils is the main requirement for exploration missions. Moreover, our approach can be adapted to any kind of wheeled rover and the processing power needed remains relatively low, which makes online computation feasible.

In rough terrain, the problem of tracking the robot's position is tedious because of the excessive variation of the ground. Further, the field of view can vary significantly between two data acquisition cycles. In this thesis, a method for probabilistically combining different types of sensors to produce a robust motion estimation for an all-terrain rover is presented. The proposed sensor fusion scheme is flexible in that it can easily accommodate any number of sensors, of any kind.

In order to test the algorithm, we have chosen to use the following sensory inputs for the experiments: 3D-Odometry, inertial measurement unit (accelerometers, gyros) and visual odometry. The 3D-Odometry has been specially developed in the framework of this research. Because it accounts for ground slope discontinuities and the rover kinematics, this technique results in a reasonably precise 3D motion estimate in rough terrain.

The experiments provided excellent results and proved that the use of complementary sensors increases the robustness and accuracy of the pose estimate. In particular, this work distinguishes itself from other similar research projects in the following ways: the sensor fusion is performed with more than two sensor types and sensor fusion is applied a) in rough terrain and b) to track the real 3D pose of the rover.

Another result of this work is the design of a high-performance platform for conducting further research. In particular, the rover is equipped with two computers, a stereovision module, an omnidirectional vision system, an inertial measurement unit, numerous sensors and actuators and electronics for power management. Further, a set of powerful tools has been developed to speed up the process of debugging algorithms and analyzing data stored during the experiments. Finally, the modularity and portability of the system enables easy adaptation of new actuators and sensors. All these characteristics speed up the research in this field.

Version abrégée

La robotique tout-terrain est un domaine de recherche très actif et beaucoup d'efforts sont déployés pour rendre les robots totalement autonomes. Les domaines d'application pour de tels robots sont l'exploration d'environnements hostiles comme par exemple, des déserts, des volcans, l'Antarctique, la surface de Mars, ou encore pour des opérations de sauvetage suivant des désastres naturels (tremblements de terre) ou artificiels.

La difficulté de rendre de tels robots autonomes est grande. La tâche nécessite, en particulier, de concevoir des systèmes capables d'évoluer dans des environnements inconnus, sans information à *priori*, avec la difficulté additionnelle de la perception et de la locomotion en terrains accidentés. Parmi toutes les fonctions nécessaires au fonctionnement du système, la locomotion et l'estimation de position sont capitales. En effet, le robot ne pourra pas remplir la tâche qui lui est assignée si un principe de locomotion inadapté est utilisé et ne pourra pas planifier correctement son chemin s'il ne connaît pas sa position actuelle. Cette thèse traite spécifiquement les problèmes de locomotion et d'estimation de position en terrain accidenté.

Le Laboratoire de Systèmes Autonomes a développé un robot tout-terrain appelé Shrimp, qui présente de très bonnes aptitudes de franchissement d'obstacles. Ses performances dépassent celles de la majorité des structures existantes et permettent d'étendre le spectre des régions explorables par des robots tout-terrain. Afin d'améliorer encore les capacités du robot et de minimiser l'énergie consommée, une méthode visant à limiter le glissement des roues a été mise au point dans le cadre de cette thèse. Contrairement à d'autres méthodes de contrôle, notre approche ne nécessite pas l'utilisation de modèles d'interaction roue-sol. Cette propriété permet au système de fonctionner quel que soit le type de sol rencontré durant sa mission. De plus, notre système peut être adapté à tous les robots passifs à roues et peut fonctionner en temps réel.

En terrain accidenté, il est très difficile d'obtenir une bonne estimation de la position d'un robot car celui-ci est soumis à de fortes vibrations et le champ de vision peut changer rapidement. Cette thèse décrit une technique robuste permettant, mal-

gré toutes ces contraintes, d'obtenir une bonne estimation de position en fusionnant des informations provenant de différents capteurs. La méthode proposée est très flexible et permet d'incorporer facilement de nouveaux capteurs. Afin de tester les algorithmes, nous avons choisi d'utiliser les capteurs suivants: de l'odométrie tridimensionnelle, une centrale inertielle (accéléromètres et gyroscopes) et une technique d'odométrie visuelle. La technique d'odométrie 3D a été développée dans le cadre de cette recherche et est appliquée au robot pour estimer son déplacement. La prise en compte de la structure mécanique et des changements brusques de pente permet de produire des estimations de position relativement bonnes compte tenu de la difficulté des terrains rencontrés.

Les expériences de fusion de capteurs ont donné d'excellents résultats et prouvent que l'utilisation de capteurs complémentaires permet d'améliorer substantiellement la précision et la robustesse de l'estimation de position en terrain accidenté. Ce travail se distingue des autres par les éléments suivants: la fusion de capteurs est faite avec plus de deux capteurs (ce qui n'est pas courant dans le domaine), la méthode est appliquée à un robot tout-terrain et finalement la position est estimée en trois dimensions.

Un autre résultat intéressant de ce travail est le développement d'une plateforme de recherche performante. Durant cette recherche, le robot a été équipé de deux ordinateurs, d'un système de stéréovision, d'une caméra omnidirectionnelle, de nombreux capteurs et actionneurs et d'une électronique de gestion de l'énergie. De plus, tout un ensemble d'outils logiciels a été développé pour la mise au point d'algorithmes et l'analyse des données produites durant les expériences. Pour terminer, la modularité et la portabilité du système permet une adaptation facilitée de nouveaux périphériques et d'actionneurs de toute sorte. Toutes ces caractéristiques permettent d'accélérer la recherche dans ce domaine.

1

Introduction

1.1 Autonomy in rough terrain

Making mobile robots move by themselves and take their own decisions is relatively new. Thanks to the efforts of a large research community and the evolution of technology, fully autonomous robots are today ready for applications in structured environments. However their level of autonomy is still very limited and the environments in which they are deployed are generally engineered in order to guarantee reliability. Most successful applications are limited to indoor, office or industry-like environments.

Rough terrain robotics is a fast evolving field of research and a lot of effort is being put towards realizing fully autonomous outdoor robots. Such robots are applied in the scientific exploration of hostile environments like deserts, volcanoes, the Antarctic or other planets. There is also a high level of interest for such robots for search and rescuing after natural or artificial disasters. Two examples of application are given:

- The NASA project “Life in Atacama” aims to search autonomously for life in the Atacama desert in Chile. The first results are very promising but the following extract illustrates the encountered difficulties: “The farthest Zoë ran autonomously was 3.3 kilometers but on average a traverse would terminate after just over 200 meters” (courtesy, [Atacama])
- The recent NASA “Mars Exploration Rover” mission (MER) aims to understand how past water activity on Mars has influenced the red planet's environment over time. Tele-operation of robots from earth is very slow because of the narrow bandwidth of communication and time delay. Thus a high level of autonomy would speed up exploration. However, for safety reasons, autonomous navigation was enabled only on relatively easy terrains and for short traverses.

These examples show that human supervision is still required for operating rovers and that further effort is required in order to enable fully autonomous operation. The aim of the following sections is to describe the challenges of autonomous robotics for rough terrain and to present the contributions of this dissertation.

1.2 The challenges of rough terrain navigation

1.2.1 The lack of prior information

There is a paradox between exploration and localization and the problem is not new. In the past, navigators had to explore and map unknown areas while keeping track of their own position, which is difficult without a consistent map. For a mobile robot, the problem is the same when navigating in a new area. There is no way to guarantee an optimal path between two points without any prior information. In order to reach a distant goal, the robot has to progressively gain knowledge about the explored environment and store it in such a way that it can be used later for planning a path to the final destination.

1.2.2 Perception

A mobile robot uses different types of sensors in order to acquire knowledge about its environment. Unfortunately, all the sensors are error prone and their measurements are uncertain. In comparison with indoor environments, the conditions in natural scenes are even more demanding and the acquired data is more difficult to analyze and understand. For example, changing lighting conditions can strongly affect the quality of the images and the vibrations due to uneven soils lead to noisy signals. In order to illustrate the problems involved in interpreting data, let's consider scans acquired by a 2D laser range finder (360°) from a given position. In two dimensions (e.g. flat ground in a static office-like environment), two scans taken at the same place but with different heading are almost identical. A simple scan matching algorithm confirms that both scans have been taken from the same position. In rough terrain, even a small change in heading can lead to a large change in attitude. Even if the scans have been taken at the exact same position they can be completely different. This example demonstrates the exponential increase of complexity when moving from 2D to 3D and shows the importance of choosing appropriate sensor configurations for outdoor environments.

Another problem of sensing in cluttered terrains is linked to the fact that the field of view is usually limited to a small portion of the environment because of the occlusions generated by the numerous obstacles and slope changes of the terrain. This requires the robot to maneuver frequently to acquire more information and forces it to take more risks while exploring an area. That kind of problem is intrinsic to ground vehicles whereas flying robots are less likely to encounter such constraints because they can adapt their altitude in order to get a global and consistent view of the environment they are exploring.

1.2.3 Locomotion

Indoors, the obstacle map used for navigation is usually composed of obstacle and obstacle-free areas and the robot motion is considered as totally feasible within the obstacle-free regions (in a static case). In rough terrain, this kind of representation is not possible because the obstacle configuration and the types of soils are not precisely known beforehand. The difficulty is to determine if a specific area is traversable or not, if the rover will have to roll on sand or on bare rock, if its mechanical architecture is adapted to the specific obstacle configuration and so on. A single rock of the size of the wheel's diameter can be overcome on a flat terrain but can cause the rover to tilt over in some specific situations, such as on a steep slope. These examples illustrate the complexity of locomotion in rough terrains. Thus, both planning a safe path and actually controlling the rover actuators to execute the requested trajectory are difficult tasks. A powerful all-terrain locomotion concept together with a good wheel controller optimizing traction enables reaching more challenging areas and increases the performance of the system.

1.3 About this work

In this section the contributions of the thesis and its structure are presented. Because the document refers to a lot of different topics, the state of the art is presented in an abstract manner in this section and more deeply in the specific chapters.

1.3.1 Context of the research

A large part of the literature concerning autonomous navigation in rough terrain focuses on high level functionalities such as environment modeling, perception and path planning. In [Singh00], traversability maps are used instead of passable/impassable maps to plan a path through an unknown scene. The D* algorithm is used to dynamically replan an optimal path as the robot acquires more information about the environment. The authors of [Gancet03] propose a unified process considering both perception planning and path planning so that the most relevant perception can be performed regarding the current goal of the robot. This provides a way to both, optimally explore an environment while planning a path to the goal. [Lacroix02] nicely presents the state of the art on autonomous navigation in unknown terrains and proposes solutions to integrate the required functionalities in a consistent way. The publication also insists on the importance and difficulty of localization for autonomous navigation and the necessity to use a set of concurrent and complementary algorithms to produce robust position estimates. [Bonnafous01] focuses on the selection of feasible displacements based on the ki-

nematics constraints of the rover and a digital elevation map. Ensuring the proper execution of the selected motions is still an open challenge.

1.3.2 Contributions and structure of this work

Considering the state of the art and the open challenges of autonomous navigation in rough terrain, this thesis aims to contribute towards gaining a better understanding of the problems involved. In particular, it proposes concrete solutions for improving both *locomotion* and *localization*. Furthermore, a *research platform* has been developed for testing the proposed algorithms in real conditions and to conduct further research in this field.

- *Locomotion*: exploring hazardous environments requires the development of adapted locomotion concepts capable of handling rough terrain. Such structures have generally many degrees of freedom and their control is complex. - We propose both, an efficient mechanical design and a controller for optimizing locomotion in rough terrain (wheel slip minimization).
- *Localization*: the environment cannot be modelled as a 2D traversability map any more and the full 3D problem has to be considered. In rough terrain, the complexity of the position tracking process increases exponentially. - We have implemented a method for fusing the measurements of several sensors in order to robustly track the 3D position of a robot in rough terrain.
- *Research platform*: complete systems including hardware and development tools are not commercially available for rough terrain applications. Such systems are complex and necessitate a good framework for conducting experiments. - We have developed a fully operational all-terrain prototype to conduct research.

Structure of this work

Chapter 2 presents the design of the rough terrain rover SOLERO capable of passively handling obstacles with sizes ranging up to two times the wheel diameter. This design has shown great potential in exploring hazardous environments. In the framework of this research, specific software tools and hardware have been developed for the prototype, making the system fully operational to run experiments in real conditions.

A new method, called 3D-Odometry, is presented in **chapter 3**. When combined with an adapted mechanical structure it produces reliable three dimensional position estimates in rough terrain, thus contributing towards more accurate localization.

A physics-based controller minimizing wheel slip is proposed in **chapter 4**. Minimizing wheel slip not only minimizes odometric errors (localization) but also enhances the climbing performance of the rover (locomotion). This method is generalized and can be applied to any kind of passive mechanical structure with wheels.

Finally, **chapter 5** proposes a set of tools for combining proprioceptive and exteroceptive sensors to robustly track the rover position in three dimensions. The method allows for easy accommodation of any number of sensors, of any kind.

1.3 ABOUT THIS WORK

2

The SOLERO rover

2.1 Introduction

SOLERO (SOLar-Powered Exploration Rover) is the name of a study carried out jointly by Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, and von Hoerner & Sulger GmbH (vH&S), Germany, under a contract of the European Space Agency (ESA). The objective of this activity was to develop a system design for a regional exploration rover, including breadboarding for the demonstration of locomotion capabilities, payload accommodation, power provision and control. More information about this project can be found in [Michaud02].

At the end of this project, one of the breadboard kept at EPFL has been significantly modified in order to accommodate more sensors and computational power. The intent of this chapter is to describe the platform and the tools developed in the framework of this research.

2.2 The mechanical design

The classification we generally use to study locomotion concepts makes the difference between active and passive locomotion. Passive locomotion is based on passive suspensions, that means no additional actuators to guarantee stable movement. On the other hand, an active robot implies a closed loop control to maintain the stability of the system during motion. Active locomotion extends the climbing capability of a robot but increases the complexity of the mechanics and the control. The numerous motors and associated sensors have a negative impact on power consumption, weight and reliability. On the other hand, some well designed passive concepts can offer very good climbing performance without suffering from the drawbacks of active designs. A complete study of locomotion concepts for rough terrain can be found in [Lauria03b].

The mechanical structure of SOLERO is similar to that of Shrimp, an all-terrain rover developed at EPFL in 99 [Siegwart00][Estier00][Siegwart02]. This passive structure shows excellent climbing abilities without any specific active suspension control.

2.2 THE MECHANICAL DESIGN

SOLERO has one wheel mounted on a fork in the front, one wheel attached to the main body at the rear and two bogies on each side (see Fig. 2.1). The parallel architecture of the bogies and the spring suspended fork provide a high ground clearance while keeping all six motorized wheels in ground-contact at any time. This ensures excellent climbing capabilities over obstacles up to two times the wheel diameter and an excellent adaptation to all kinds of terrains.

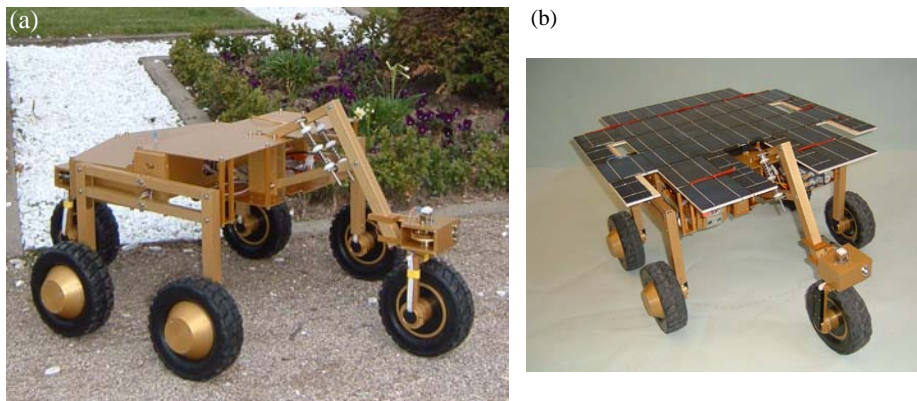


Figure 2.1: SOLERO mechanical structure (a) B-prototype equipped with a solar panel (b).

The front fork has two functions: its spring suspension guarantees ground contact of all wheels and its particular parallel mechanism produces a passive elevation of the front wheel if an obstacle is encountered. As shown in Fig. 2.2b, the front wheel has an instantaneous centre of rotation situated under the wheel axis, which makes it possible to get easily on an obstacle.

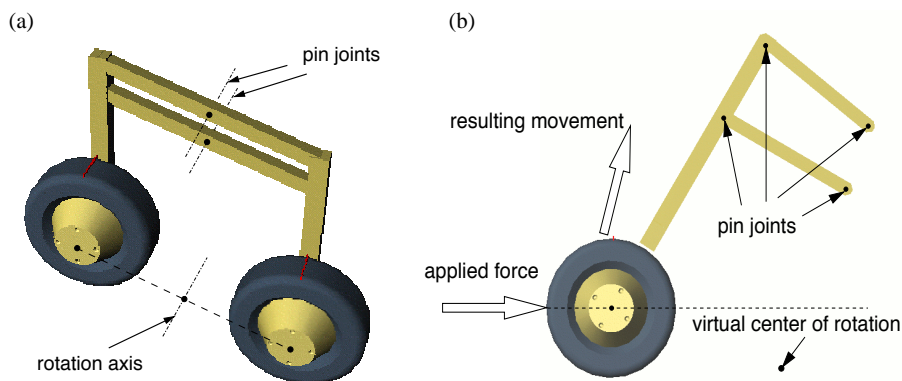


Figure 2.2: Parallel mechanisms a) virtual rotation axis of a bogie b) front fork kinematics. Because the instantaneous rotation center is placed below the wheel axis, the fork passively fold for climbing an obstacle.

The bogies provide lateral stability. To ensure similarly good ground clearance and climbing capabilities, their virtual centre of rotation is set to the height of the wheel axis using the parallel configuration shown on Fig. 2.2a. The steering of the rover is realized by synchronizing the rotation of the front and rear wheel and the speed difference of the bogie wheels (skid-steering).

The following table and figure summarize the overall characteristics of SOLERO. All mechanical variables and parameters are defined in Appendix A.

Table 2.1: *SOLERO main characteristics*

Rover's main body mass (inc. batteries, laptop etc.)	7.4 kg
Wheel mass (inc. motor, gears)	0.7 kg
Steering mechanism mass	0.6 kg
Spring constant	357 N/m
Wheel diameter	0.15 m

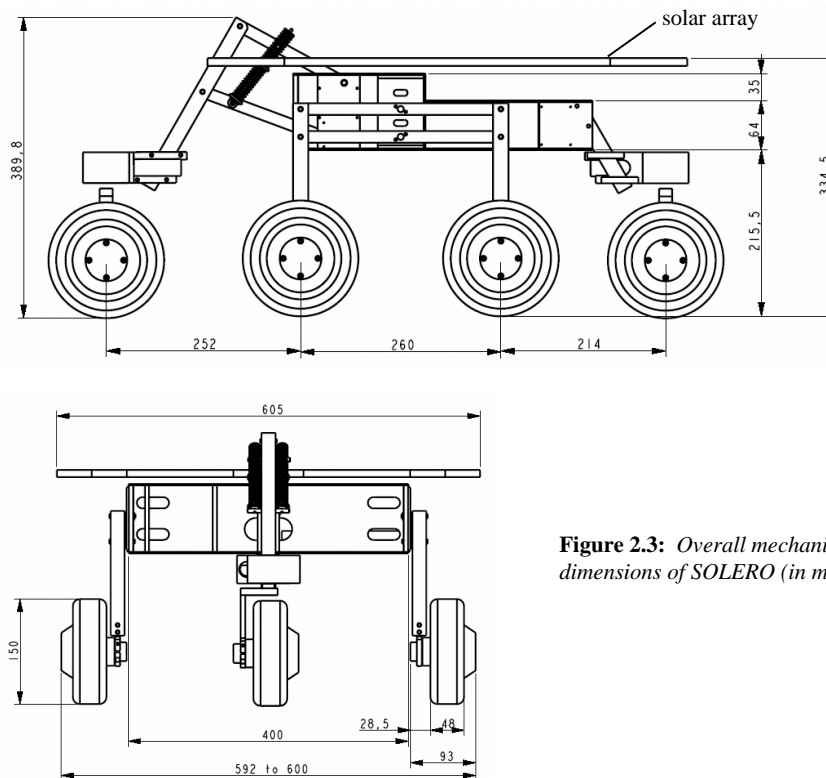


Figure 2.3: *Overall mechanical dimensions of SOLERO (in mm)*

2.3 The control architecture

This section presents the different sensors that have been mounted on SOLERO and the control system i.e. the actuators, the computers and the electronic devices. These components are depicted in Fig. 2.4.

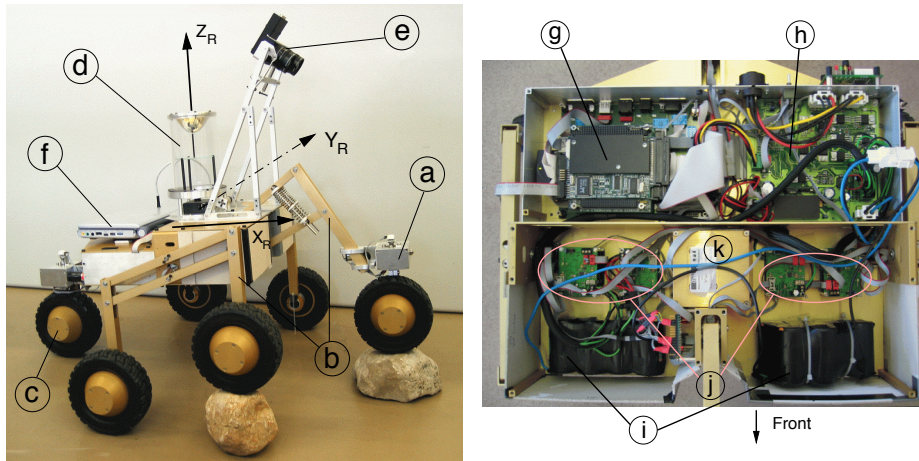


Figure 2.4: Sensors, actuators and electronics of SOLERO. *a)* steering servo mechanism, the same is used for the rear wheel *b)* passively articulated bogie and spring suspended front fork (equipped with absolute angular sensors) *c)* 6 motorized wheels (DC motors) *d)* omnidirectional vision system *e)* stereo-vision module, orientable around the tilt axis *f)* laptop (*solerovaio*) *g)* micro-computer (*soleropc104*) *h)* energy management board *i)* batteries (NiMh 7000 mAh) *j)* I²C slaves modules (motor controllers, angular sensor module, servo controllers etc.) *k)* Inertial Measurement Unit

The global architecture of the robot is presented in Fig. 2.5. The grayed boxes represent computers and the rounded rectangles the sensors and actuators.

SOLERO is equipped with two computers communicating through a crossover ethernet cable. The computer called *solerovaio* is a laptop in charge of image processing. It acquires images from the stereo-vision rig and the omnicam through a firewire bus and transmits processed data to the second computer, called *soleropc104*. This second computer has access to all the other sensors and actuators of the robot. It reads data from an Inertial Measurement Unit through an RS232 port and interfaces an I²C bus through the parallel port. The devices attached to the I²C serial bus are: six wheel controllers, three servo-controllers, one angular sensor module (reading the three suspension angles) and a device for the energy management of the rover. *soleropc104* acts also as a gateway for the rover subnet. A host computer (*soleroap*) can connect to the subnet through a wireless ethernet interface. This allows, for example, to download images, remote control the rover through a graphical user interface and get the rover state online.

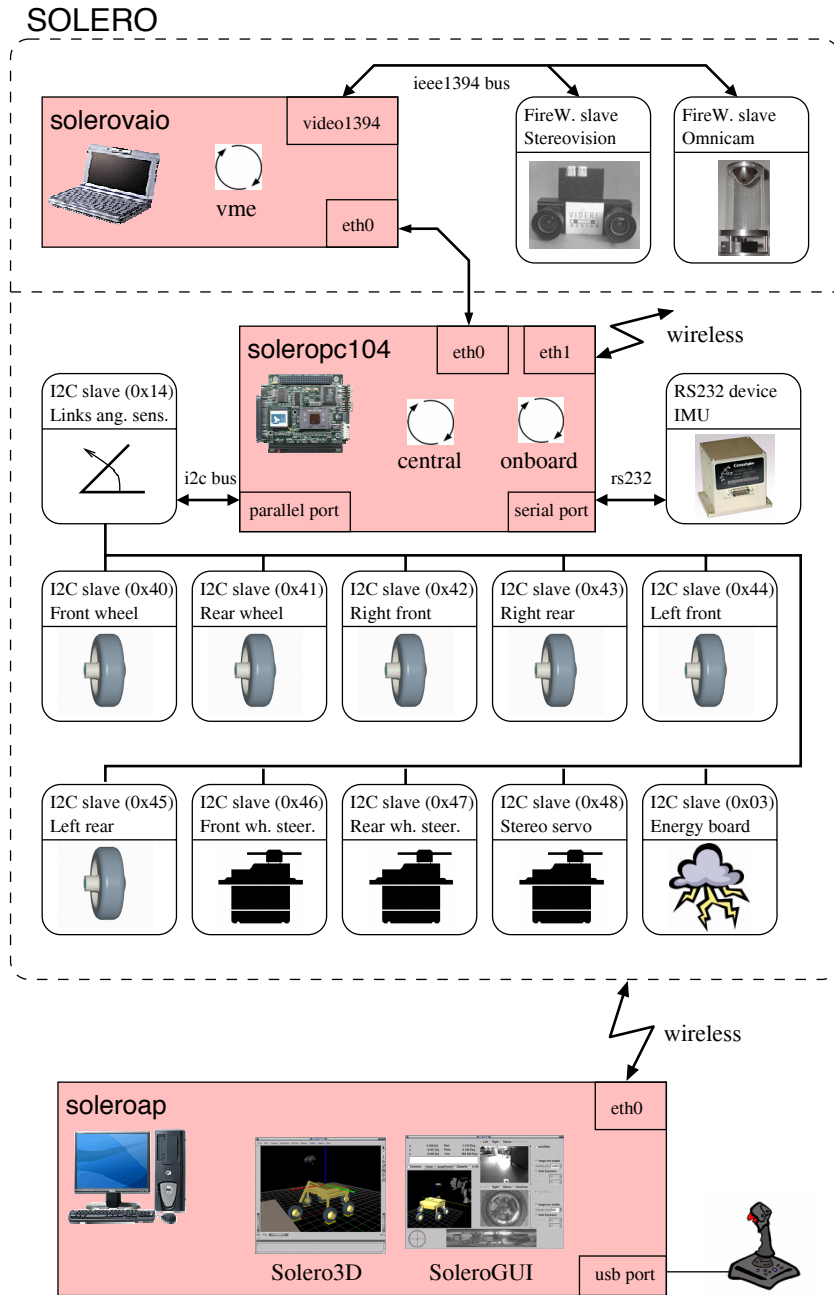


Figure 2.5: Schematics of the control system

2.3.1 Sensors and actuators

Using busses for interfacing the peripherals allows for the easy extension of the number of sensors and actuators. A camera can be easily added to the firewire bus and devices with lower bandwidth needs can be attached to the I²C bus.

2.3.1.1 The I²C modules

The ASL developed various I²C slaves implementing interfaces for different kinds of sensors and actuators i.e. infrared and ultrasound distance sensors, linear camera, inclinometer, GPS, servo controller and DC motor controller. Such architecture allows to attach up to 127 devices. Because the processing load is distributed at the slaves level, there is less computational load for signal processing on the main CPU.

For SOLERO, two new types of I²C slaves have been designed in the framework of this research: an absolute angular sensor and an energy management module.

a. Angular sensor

The angular position of the bogies and the fork relative to the body have to be measured in order to know the rover state during operation. To measure the angle of a joint, a magnet is fixed to the joint axis and the direction of its magnetic field is measured by means of a magneto-resistive bridge (fixed to the main body). This contactless sensing mechanism has the advantage to provide measures with 0.2 degrees of precision, without being too sensitive to temperature and drift. The resulting performance is much better than if a standard potentiometer would have been used. Furthermore, this solution provides absolute angles and does not require initialization every time the system is started. The electronics of the module is depicted in Fig. 2.6.

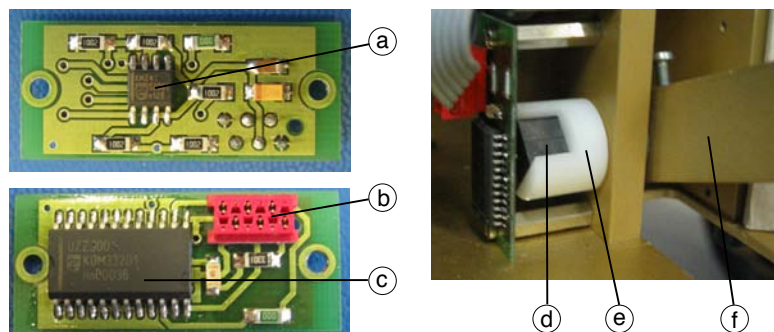


Figure 2.6: Angular sensor for the front fork. **a)** magneto-resistive bridge **b)** communication and power bus **c)** linearization chip **d)** magnet **e)** magnet holder (fixed to the axis) **f)** front fork

b. Energy management board

The energy management board has many features but only the functional level is described in this section. Here is a list of the main board features:

- Two power supplies can be connected: an external power supply (DC voltage between 11 and 15V or a solar panel) and a battery. It is possible to switch between the two sources using an external switch or the I²C interface.
- Delivers regulated 5V (10 Amps) and 12V (1.2 Amps) for the system.
- It is possible to turn on and off the voltage delivered to the motors and the system separately.
- Currents and voltages of all the sources and drains are measured and monitored. This feature has been used to study the energy consumption of the rover.
- The battery voltage is monitored and the board warns the user by means of a blinking led and an acoustic signal. Below a certain voltage the system is turned off automatically in order to protect the battery and the system.
- The battery status can be read from a seven segment display.
- All the functions of the board can be accessed through I²C commands i.e. switch on/off, battery status, currents and voltages, etc.

Fig. 2.7 depicts the energy management board and its physical interfaces.

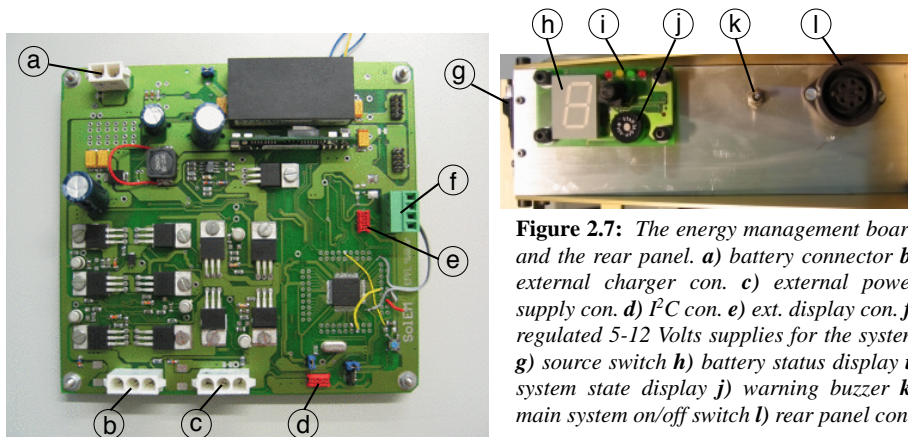


Figure 2.7: The energy management board and the rear panel. **a)** battery connector **b)** external charger con. **c)** external power supply con. **d)** I²C con. **e)** ext. display con. **f)** regulated 5-12 Volts supplies for the system **g)** source switch **h)** battery status display **i)** system state display **j)** warning buzzer **k)** main system on/off switch **l)** rear panel con.

2.3.1.2 Stereovision

The stereovision rig is a MegaD module from VidereDesign. It can acquire gray-scale images up to 1280 x 960 pixels. Equipped with lenses of a focal length of 4.8

2.3 THE CONTROL ARCHITECTURE

mm and with a CMOS of 2/3" it offers a field of view of $85^\circ \times 69^\circ$. It is mounted on top of a mast and can be oriented around the tilt axis (in the vertical plane). This allows to keep ground features in the field of view of the cameras even if the rover is tilted upwards/downwards. Fig. 2.8 shows two views of the mechanism. In order to keep the center of gravity as low as possible, the motor is mounted next to the rover's main body. The rotational motion is transmitted to the stereovision module by mean of a traction pole. A lot of efforts have been deployed to design a system with high stiffness and low mechanical play between the parts. This is important because the relation between the camera coordinate system and the rover system is used by the navigation algorithms. The transformation has to be known precisely because even small inaccuracies can lead to significant error of localization.

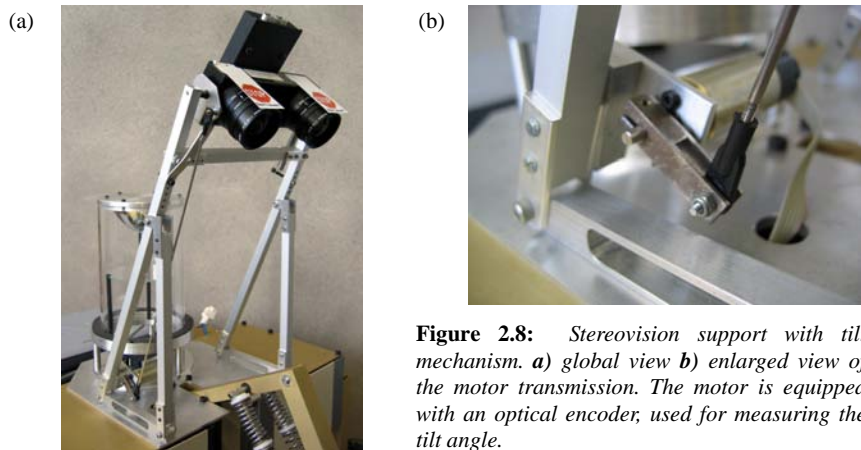


Figure 2.8: *Stereovision support with tilt mechanism. a) global view b) enlarged view of the motor transmission. The motor is equipped with an optical encoder, used for measuring the tilt angle.*

2.3.1.3 Omnicam

The omnicam, depicted in Fig. 2.9, has been especially designed for SOLERO. The imager is the DCAM camera from VidereDesign which has the advantage of being compact and relatively low power. Grayscale and color images up to 640×480 pixels can be acquired. The mirror has a very interesting feature. It is equiangular: that means that each pixel on the image covers the same solid radial angle. As a consequence, when moving radially in the image, the shape of a feature (i.e. a small image subwindow) is less distorted than it would be by using other mirror shapes. This facilitate feature tracking between two consecutive images and data association. More information about that kind of mirror can be found in [Chah197] and [Ollis99].

In order to avoid occlusion and to protect the mirror and the camera from dust, a transparent cylinder is used.

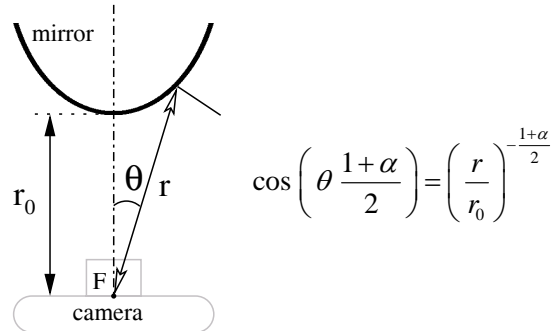


Figure 2.9: The SOLERO omnica. The shape of the mirror is specified by the equation. For this design, the parameters r_0 and α are respectively 14 cm and 11° .

2.3.1.4 Inertial Measurement Unit

The Inertial Measurement Unit (IMU) is the VG400CC-200 device from Crossbow. It is a solid-state inertial measurement system that utilizes MEMS micro-machined sensing technology. It is composed of a triad of accelerometers (velocity rate sensors) and gyroscopes (angular rate sensors), which are combined internally to provide roll and pitch angles in static and dynamic conditions (through a Kalman filter). Furthermore, the calibrated angular rates and accelerations are available and come together with a timestamp. This timing information allows to perform accurate integration of angular and velocity rates over time.

2.3.2 Software architecture

The whole software has been programmed in C and C++ and runs under Linux. However, substantial effort towards portability has been made by choosing cross-platform components and libraries e.g. the widgets, mathematical optimization and communication libraries. The system is divided into five functional modules running as separate processes i.e. *vme*, *central*, *onboard*, *Solero3D* and *SoleroGUI* (see Fig. 2.5). The modules can run on different computers and communicate using the Inter-Process Communication messaging system [IPC]. This IPC library, developed at Carnegie Mellon University, can transparently send and receive complex data structures, including lists and variable length arrays, using both anonymous "publish/subscribe" and "client/server" message-passing paradigms.

In the current configuration *vme* runs on *soleroaio*, *onboard* and *central* on *soleropc104* and *Solero3D* and *SoleroGUI* on a remote computer *soleroap*. However, the architecture can be easily modified to accommodate another hardware

configuration. For example, *vme*, *central* and *onboard* could run on the same machine e.g. *solerovaio*. Because *vme* and *onboard* exchange time critical data, the internal clocks of *solerovaio* and *soleropc104* have to be synchronized. This synchronization is guaranteed by network time protocol daemons (ntpd) running on both computers.

Central acts as a server for the IPC network. It is responsible for routing the messages and holds the system-wide information (such as defined message prototypes). *onboard*, the main program of the architecture, has access to the low level sensors and actuators i.e. the IMU and the I²C modules. Its main tasks are to perform sensor fusion and execute the motion commands coming from a remote control interface such as *SoleroGUI*. On *solerovaio*, the *vme* module has access to the stereovision and the omnicaam through the firewire bus. After images acquisition, it performs some image processing and sends the result to *onboard*. Finally, the two remaining modules *Solero3D* and *SoleroGUI* are described in the following sections.

2.3.2.1 Solero3D

This program has been developed for visualizing and logging data produced by the robot during an experiment. It can also be used for testing and debugging algorithms offline. Fig. 2.10 shows the main window (left) and the data browser (right).

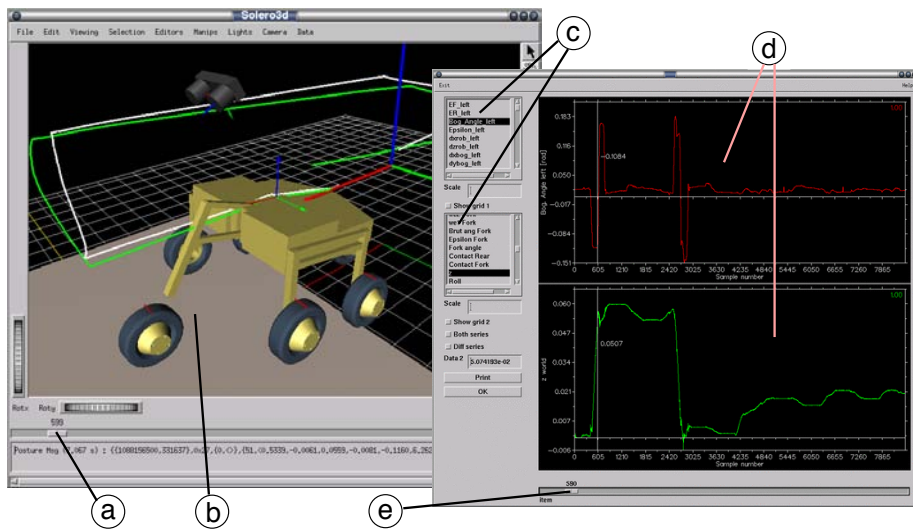


Figure 2.10: The main window (left) and the data browser (right) **a**) data replay slider. By manipulating this slider the scene is updated with the corresponding robot state **b**) 3D rendering area **c**) variable selection lists one and two **d**) plot areas one and two **e**) data browser slider.

The main window integrates a full 3D rendering area allowing the user to change views. By manipulating a slider, the set of data stored during an experiment can be replayed step by step. All the variables such as the robot position, the internal links angles and the pitch angle are plotted in the data browser. This module is a precious tool to test the system, debug and compare algorithms performance.

2.3.2.2 The remote control interface

A dedicated module called *SoleroGUI* has been developed for the tele-operation of SOLERO (Fig. 2.11). In order to ease remote control, the graphical user interface displays the images taken onboard the rover together with a 3D view of the current rover state. Furthermore, stereoscopic information is displayed in the 3D scene, allowing the operator to have a better understanding of the environment in front of the rover and thus properly avoid obstacles. At the bottom of the main window, a panoramic image is displayed. It provides a wide view of the scene and helps to plan a path without needing to turn the rover on the spot. All parameters of the imagers are accessible through dialog boxes.

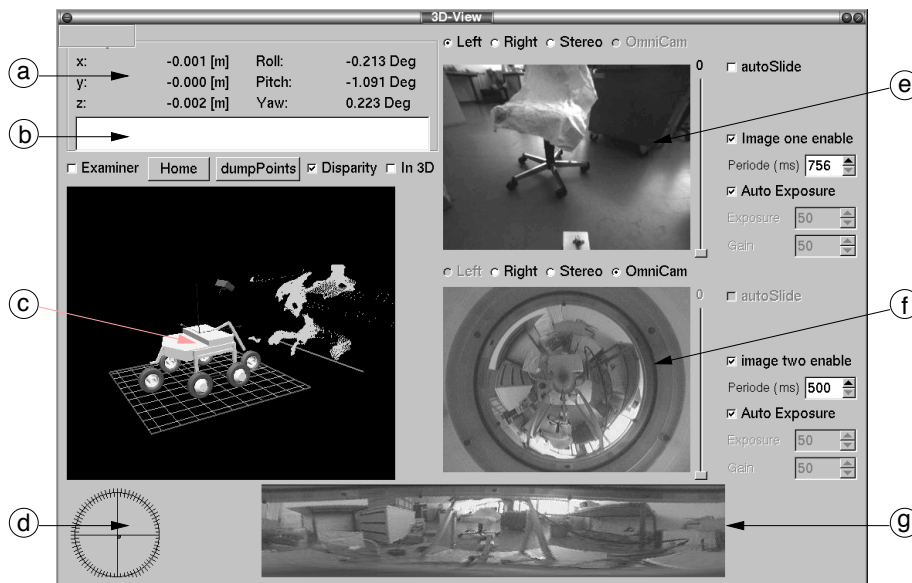


Figure 2.11: The remote control interface for SOLERO. **a)** numerical values of the robot position **b)** text box for warning messages e.g. the pitch angle exceed a predefined value **c)** 3D representation of the robot state and 3D cloud of stereo points. The operator can interactively change the perspective **d)** rover control area. The motion orders are given by clicking and moving the mouse cursor. The robot can be also driven with a joystick **e)** first image area. The user can select the left, right, stereo or omniscam image. All the imagers settings can be modified **f)** second image area **g)** panoramic view area

Other interesting features of the GUI (Graphical User Interface) are listed below

- Warning messages such as “low battery” and “dangerous rover posture” are printed on the screen in order to avoid critical situations, which could damage the rover.
- The user can control the rover with a game pad or the mouse. Smooth trajectories are generated using non-linear optimization accounting for both maximal wheel acceleration and speed.
- Two operation modes are available. The first one is called “coordinated” and allows the robot to drive on any arc of circle. The second mode is called “non-coordinated” and only straight line and point-turning motion are allowed.
- A watchdog timer is implemented to detect communication problems. The GUI sends a signal every second and the rover stops in case the signal is not received.
- The GUI uses cross-platform libraries and can be compiled and run on different OS.

2.4 Conclusion

Because no generic hardware setup exists for such rovers a lot of effort had to be deployed for making SOLERO a performant platform for research. Furthermore, a set of powerful tools has been developed for speeding up the process of debugging the algorithms and analyzing the data stored during the experiments. The modularity and portability of the system allows easy adaptation of new actuators and sensors. For example, a GPS can be easily added to the I²C bus and more cameras attached to the firewire or USB busses. The possibility to access all the schematics and firmware of the I²C sensors allows to have a low-level control of data transmission timings and thus improve the reactivity of the system. Off the shelf components are rarely well documented, especially concerning time-stamping of data, which is of high importance in robotics.

The energetic autonomy of the system running on batteries depends on the intensity and duration of the driving phases. In average, the autonomy is around three hours, allowing to run long sessions of experiments. This autonomy can be doubled by replacing NiMh batteries by LiPo while keeping the same weight.

3

3D-Odometry

3.1 Introduction

Up to recently autonomous mobile robots were mostly designed to run indoor, yet partly structured and flat environments. In rough terrain many new problems arise and position tracking becomes more difficult. Although odometry is widely used indoors (2D), its application is limited in natural environments (3D). The wheels are more likely to slip because of the rough structure of the soil and the error in the position estimation can grow quickly. For these reasons, one generally avoids using odometry in challenging terrains. However, we can look at the problem differently and ask: “Why are the wheels slipping and how could this be avoided?”

There are two different aspects on which we can act directly. The first one is to improve the mechanical structure of the robot. Indeed, a good mechanical design allows the rover to move smoothly across obstacles and thus limits wheel slip. As described in the previous chapter, SOLERO can passively adapt to a large range of obstacles and allows limited wheel slip in comparison with rigid structures such as four-wheel drive rovers. Thus, the odometric information is usable even in rough terrain. A new technique, called *3D-Odometry*, which provides 3D motion estimates of SOLERO is presented in this chapter.

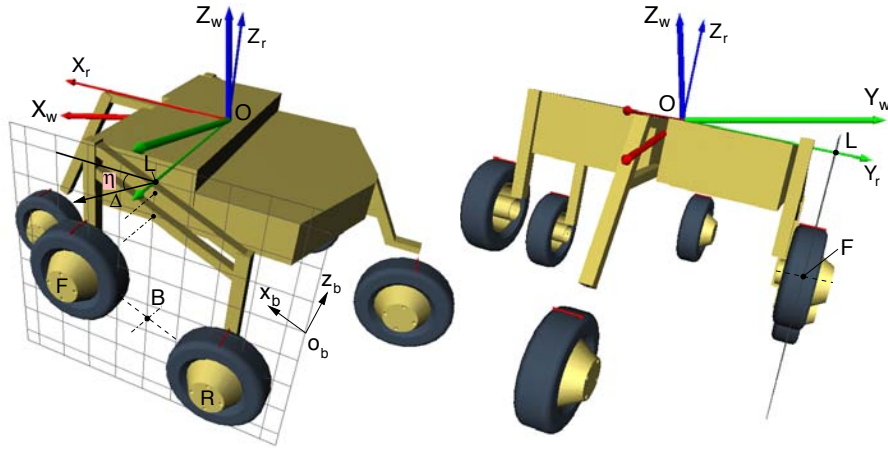
The second action for limiting slip is to improve the way the wheels are controlled. A good balance of the torques and speeds between the wheels is essential to optimize the robot's motion. A torque controller minimizing slip and maximizing traction is presented in the next chapter.

3.2 3D-Odometry

Odometry is widely used for mobile robots moving on flat and even terrains. The equations are well known and allows to estimate the position and the orientation of the robot i.e. $[x_\pi, y_\pi, \psi]^T$ in a plane π . This vector is updated by integrating motion increments between two subsequent robot poses. The error due to integration is minimized by keeping the time-step between the updates as small as possible.

This 2D odometry method can be extended in order to account for slope changes in the environment and to estimate the 3D position in a global coordinate system i.e. $[x, y, z, \phi, \theta, \psi]^T$. This technique uses typically an inclinometer for estimating the roll (ϕ) and pitch (θ) angles relative to the gravity field [Lacroix02]. Thus, the orientation of the plane π , on which the robot is currently traveling, can be estimated. The z coordinate is computed by projecting the robot displacements in π into the global coordinate system. This method, which will be referred later as the *standard method*, works well under the assumption that the ground is relatively smooth and doesn't have too many slope discontinuities. Indeed, the system accumulates errors during transitions because of the planar assumption. In rough terrain this assumption is not verified by definition and the transitions problem must be addressed properly.

The following sections describe a new method, called 3D-Odometry, which takes the kinematics of the robot into account and treats the slope discontinuity problem. The 3D-Odometry computation can be divided into two steps: the displacement estimation of the left and right sides of the robot (section 3.2.1) and the computation of the resulting 3D displacement (section 3.2.2). Fig. 3.1 introduces the used reference frames and variables.



O $X_w Y_w Z_w$ global reference frame (world frame) L projection of O in the bogie plane
 O $X_r Y_r Z_r$ robot's frame (linked to the main body) B left bogie center (rotation center)
 O_b xz bogie frame (in the bogie plane) Δ, η norm/angle of L 's displacement

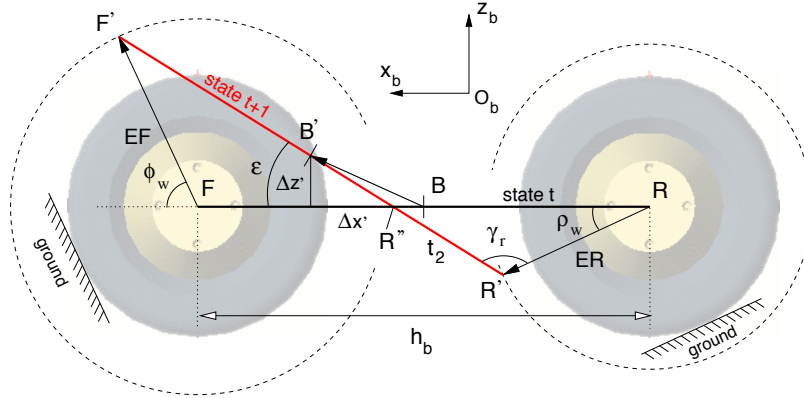
Figure 3.1: Reference frames definition

3.2.1 Bogie displacement

For SOLERO, we have to consider the translations of the left and right bogie to compute the motion of the robot's body. The aim of this section is to describe how

to compute the displacement (Δ and η) of one bogie knowing the translations of the wheels (encoder data ER, EF) and the change of the bogie angle (ε) between the initial and final state (see Fig. 3.1, 3.2 and 3.3). In what follows, the equations have been developed only for the left bogie. However, the same method can be applied for the right bogie and the corresponding equations can be obtained using simple variable and parameter substitutions.

For computing the displacement of L we proceed in two steps: firstly we compute the displacement of B (Fig. 3.2) and then propagate this motion through the bogie's mechanical structure to compute the effective displacement of L (Fig. 3.3).



ER, EF	rear/front wheel displacement	ε	bogie's angular change
ρ_w, ϕ_w	rear/front wheel's direction of motion	h_b	distance between the wheel centers
R, F	initial rear/front wheel center	$\Delta x', \Delta z'$	x/z components of vector BB'
R', F'	final rear/front wheel center	t_1	distance BR'' (not displayed on the figure)
B, B'	initial/final position of the bogie center	t_2	distance $R'R''$

Figure 3.2: Displacement of B between state t and $t+1$. The final position of the rear/front wheel is on a circle of radius ER/EF centered at R/F respectively.

Because the distance between the wheels remains constant one can write the following equations

$$\overline{RF} = \overline{RR'} + \overline{R'F'} + \overline{F'F}$$

$$\begin{pmatrix} h_b \\ 0 \end{pmatrix} = \begin{pmatrix} ER \cos \rho_w \\ -ER \sin \rho_w \end{pmatrix} + \begin{pmatrix} h_b \cos \varepsilon \\ -h_b \sin \varepsilon \end{pmatrix} + \begin{pmatrix} -EF \cos \phi_w \\ EF \sin \phi_w \end{pmatrix} \quad (3.1)$$

These equations can be solved for ϕ_w and ρ_w (with ER, EF and ε as parameters). However, this equation system can be inconsistent in some pathological cases. For example, if ε is zero then ER must be equal to EF because of the constant wheels distance constraint (see Fig. 3.2). In practice, ER and EF can be different because

the wheels can slip and have different speeds. When the set of parameters produces an inconsistent equation system, we simply consider that the total bogie displacement is the average of the displacements of the two wheels.

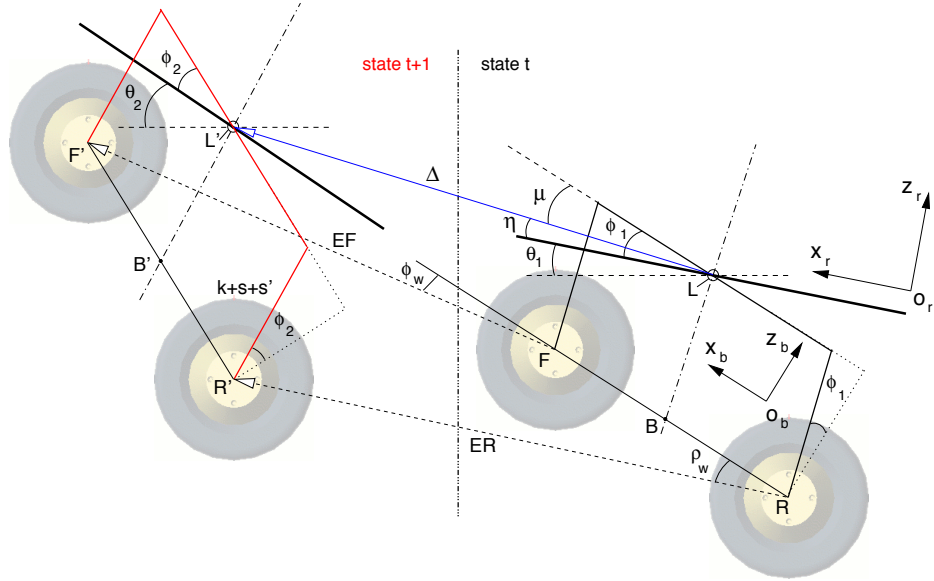
Then, the sine theorem is applied in the $RR'R''$ triangle (see Fig. 3.2) in order to obtain $\Delta x'$ and $\Delta z'$, which are the coordinates of the displacement of B expressed in the bogie's coordinate system $O_b xz$

$$\Delta x' = t_1 + \left(\frac{h_b}{2} - t_2 \right) \cos \varepsilon \quad (3.2) \quad \Delta z' = - \left(\frac{h_b}{2} - t_2 \right) \sin \varepsilon \quad (3.3)$$

with

$$t_1 = \frac{ER |\sin(\pi - \rho_w - \varepsilon)|}{|\sin \varepsilon|} - \frac{h_b}{2} \quad (3.4) \quad t_2 = \frac{ER |\sin \rho_w|}{|\sin \varepsilon|} \quad (3.5)$$

Fig. 3.3 defines the parameters for computing the displacement of L considering the displacement of B and the mechanical structure of the parallel bogie.



L	projection of the robot's center O in the left bogie plane (initial position at time t)	μ	angle of LL' expressed in $O_b xz$
L'	final position of L (at time $t+1$)	η	angle of LL' expressed in $O_r xz$
Δ	norm of LL'	θ_1, θ_2	initial/final pitch angle
$\Delta x, \Delta z$	coordinates of LL' expressed in $O_b xz$	ϕ_1, ϕ_2	initial/final bogie angle (rel. to the body)
		$k+s+s'$	bogie leg length

Figure 3.3: Real bogie displacement and compression

The effective bogie angle change between state t and $t+1$ is obtained using

$$\varepsilon = \theta_2 + \phi_2 - (\theta_1 + \phi_1) \quad (3.6)$$

Because the relative position of L with respect to B depends on the bogie configuration, the displacement of B and L are not the same. This effect must be taken into account to compute the effective displacement of L . Considering that the angular changes and the translations between t and $t+1$ are small, the incremental corrections are given by¹

$$\begin{aligned} c_x &= -(k + s + s') \cdot (\sin \phi_2 - \sin \phi_1) \\ c_z &= (k + s + s') \cdot (\cos \phi_2 - \cos \phi_1) \end{aligned} \quad (3.7)$$

Then c_x and c_z must be added to $\Delta x'$, $\Delta z'$ to get the effective displacements of point L expressed in the bogie coordinate system $O_b xz$

$$\Delta x = \Delta x' + c_x \quad (3.8) \quad \Delta z = \Delta z' + c_z \quad (3.9)$$

Finally, the norm of the displacement Δ and the motion angle μ are defined as

$$\Delta = \sqrt{\Delta x^2 + \Delta z^2} \quad (3.10) \quad \mu = -\arctan\left(\frac{\Delta z}{\Delta x}\right) \quad (3.11)$$

and the displacement angle expressed in the robot's frame $O_r xz$ is given by

$$\eta = \phi_1 + \mu \quad (3.12)$$

3.2.2 3D displacement

The previous section showed how to compute the translation (Δ and η) of one bogie. The aim of the current section is to derive the equations for computing the 3D displacement of the robot center O using the left and right bogie translations. In what follows, the subscripts l and r are used to denote variables related to the left and right bogie respectively. For example, η_r is the displacement angle of the right bogie defining the plane π_r and Δ_r is the norm of the translation. The main schematics for the 3D-Odometry is depicted in Fig. 3.4.

1. additional definitions of geometric dimensions of SOLERO can be found in Appendix A

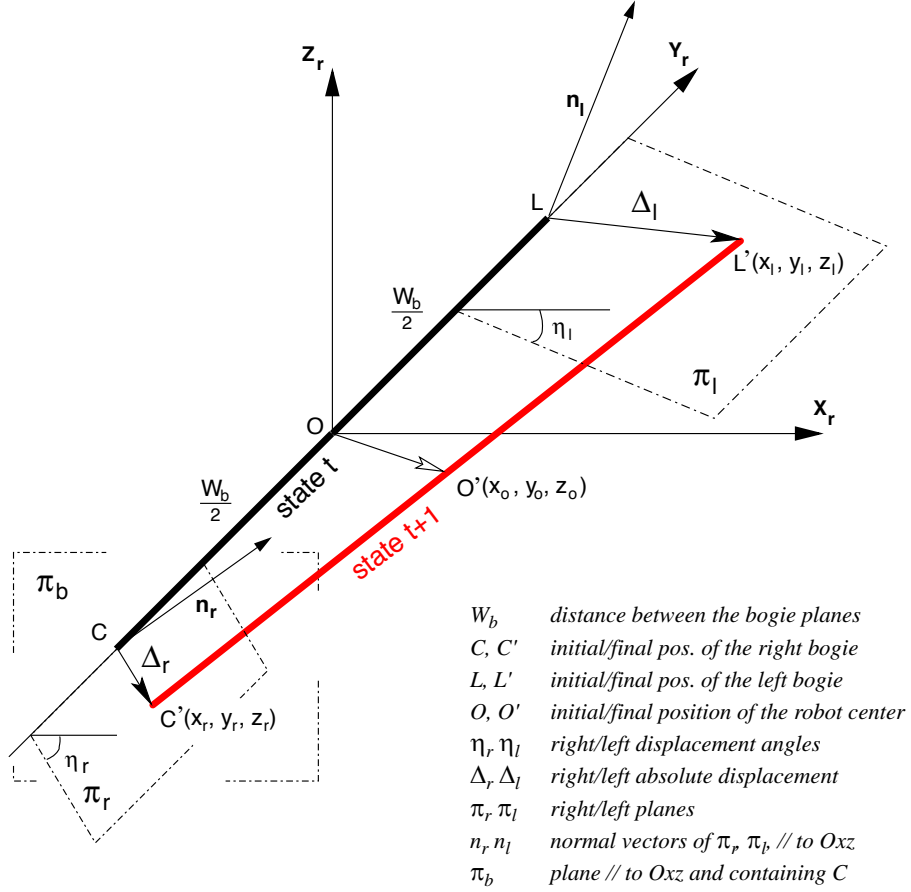


Figure 3.4: 3D-Odometry, variables definition

The angles η_r and η_l define the planes π_r and π_l containing C' and L' . C' and L' are situated on circles centered in C and L with radius Δ_r and Δ_l in the planes π_r and π_l respectively. These considerations lead to the following constraints

$$\vec{n}_r \cdot \overline{CC'} = 0 \quad (3.13) \quad \vec{n}_l \cdot \overline{LL'} = 0 \quad (3.14)$$

$$\Delta_r^2 = x_r^2 + \left(y_r + \frac{W_b}{2} \right)^2 + z_r^2 \quad (3.15) \quad \Delta_l^2 = x_l^2 + \left(y_l - \frac{W_b}{2} \right)^2 + z_l^2 \quad (3.16)$$

SOLERO is non-holonomic and uses skid-steering for turning. Thus, the infinitesimal displacements of the left and right sides of the robot mainly occur in the bogie planes. However, when the robot is turning the norm of the displacement of one side is larger than the other and, because the wheelbase remains constant, that forces a fraction of the motion to occur out of the bogie planes (along Y_r). Because of the non-holonomic constraint, this displacement cannot be measured directly. Thus, we make the approximation that the smallest displacement among Δ_r and Δ_l takes place in the corresponding bogie plane, giving to the other side an additional degree of freedom along Y_r . In the example of Fig. 3.4, Δ_r is smaller than Δ_l therefore \overline{OC} is constrained to remain in the bogie plane π_b . This additional constraint is expressed by equation 3.17. Since the wheelbase W_b remains constant one can write 3.19. Finally, the vector \overline{OD} is obtained using equation 3.18.

$$\left(\frac{W_b}{2}\right)^2 + \Delta_r^2 = x_r^2 + y_r^2 + z_r^2 \quad (3.17) \quad \overline{OD} = \overline{OC} + \frac{\overline{CL'}}{2} \quad (3.18)$$

$$W_b^2 = (x_l - x_r)^2 + (y_l - y_r)^2 + (z_l - z_r)^2 \quad (3.19) \quad d\psi = \frac{x_r - x_l}{W_b} \quad (3.20)$$

Solving the system of nine equations with nine unknowns formed by 3.13 to 3.19 leads to the solutions for \overline{OC} , \overline{OL} and \overline{OD} (the nine unknowns). The yaw angle increment is computed using 3.20. The roll increment $d\phi$ can be computed by substituting x_r and x_l by z_r and z_l in 3.20. However, we have chosen to rely on the value of the roll angle provided by the inclinometer because this is an absolute angle and therefore it is not subject to drift.

3.2.3 The contact angles

The 3D-Odometry technique provides an estimation of the translation in three dimensions and the heading change of the rover. It is interesting to note that the contact angles between the bogie wheels and the ground are also computed by this method (see Fig. 3.2). The contact angles of the fork and the robot's rear wheel are computed using parameter substitution in equation 3.1. To estimate the rear wheel contact angle, ε is replaced by the pitch change of the rover ($d\theta$) and the norm of the robot motion (computed by the 3D-Odometry) is used instead of EF . The same kind of parameter substitutions are used to compute the contact angle of the front wheel. The estimation of these angles is very important because the contact angles are required inputs for a predictive wheel controller minimizing wheel slip. Such a controller is presented in chapter 4.

3.3 Experimental results

In order to test the equations presented above, the robot has been driven forward across an obstacle of known shape and the trajectory computed online with both 3D-Odometry and the *standard method* (see section 3.2). A proportional speed controller has been implemented for the front bogie wheels and an integral term has been added for the rear wheels. This allows the front and back wheels to have slightly different speeds and therefore limits slip during the slope transitions. Some tests with PI controllers on both bogie wheels have been performed but the results were less good than the one presented in what follows.

The system has been tested in two different situations depicted in Fig. 3.5 and 3.6 respectively. The true trajectory is an approximation. It is built with characteristic positions that can be computed knowing the shape of the obstacle, the kinematic model and the state of the robot.

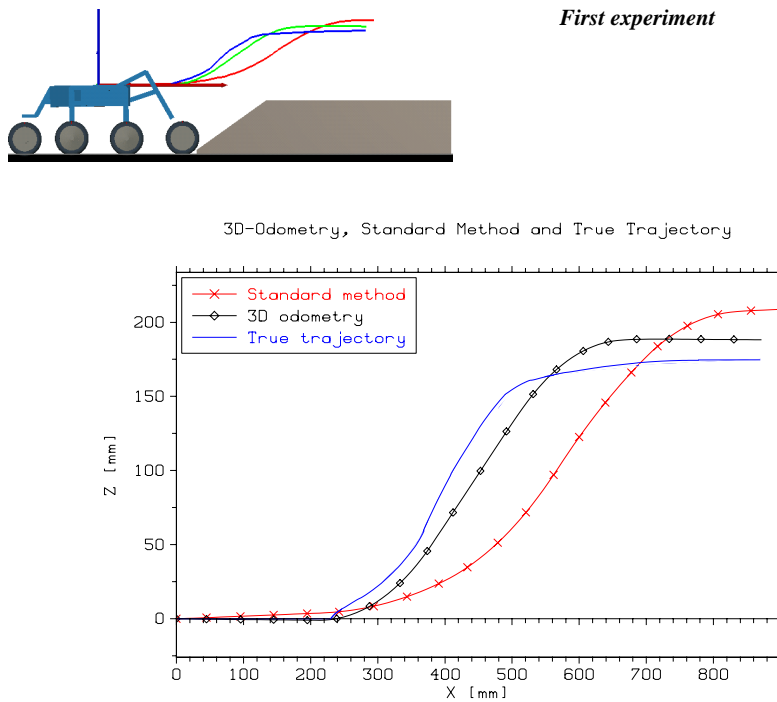


Figure 3.5: *First experiment. The robot starts in front of the obstacle, climbs a 35 degrees slope and stops on top after 870 mm in the x direction. The height of the obstacle is 175 mm. The transitions are relatively smooth for this experiment.*

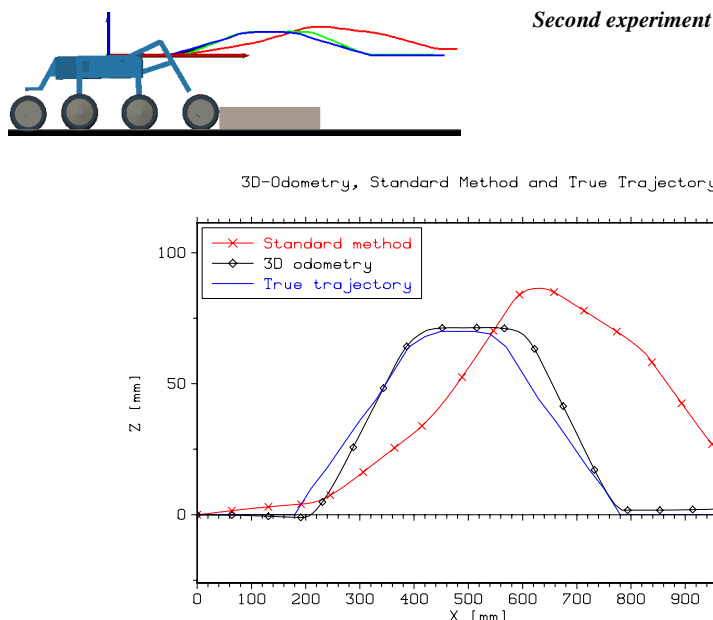


Figure 3.6: *Second experiment. The robot goes over this 300 by 70 mm obstacle and stops after 1 meter. The transitions are sharp and since our method treats the transitions the 3D-Odometry curve respects the obstacle shape.*

For both experiments, the position estimation computed with the standard method diverges quickly. This is due to the fact that the method does not account for the kinematic model of the robot and only considers the attitude of the main body. Whereas the consequences of this approximation are less relevant on smooth terrains (Fig. 3.5), they become disastrous while climbing sharp shaped obstacles (Fig. 3.6).

With the 3D-Odometry, it is possible to detect the slope discontinuities by computing the angle ε (see equation 3.6). This angle together with the wheel encoders data allow to compute the wheel-ground contact angles and the norm **and direction** of motion of a bogie (point L or C). This direction corresponds neither to the pitch of the main body nor to the mechanical angle of the bogie: it is the actual direction of motion of the point L or C . The effective incremental displacement (Δz) along the axis z_r is also computed (see Fig. 3.3). Its variation has been stored during the first experiment and is depicted in Fig. 3.7. It is easy to see when the bogie's first wheel starts climbing the obstacle ($\Delta z > 0$) and when it finally goes on the top of the obstacle ($\Delta z < 0$). The transitions are well detected and this allow to correct the z coordinate and thus better track the robot's position.

3.3 EXPERIMENTAL RESULTS

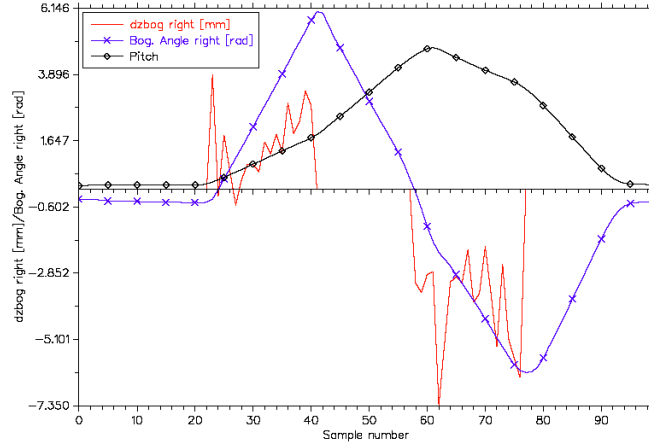


Figure 3.7: *z*-coordinate correction during the transitions of the first experiment. For clarity purpose both Pitch and Bogie Angle curve have been scaled (by a factor of 12 and 8 respectively). The time-step between two samples is about 60ms (computation time for the 3D-

The *x* and *z* coordinates of the robot's final position have been measured for each run. We did five runs for each experiment and computed the relative error. Table 3.1 lists the results corresponding to both experiments. One can see that the 3D-Odometry demonstrates much better performance. The sharper the transitions, the better it does in comparison with the standard method. The errors accumulated by the 3D-Odometry method are due to several reasons. The first one we might think about is wheel slip. In case of slip the calculated distance would be bigger than the measured one and the results presented in Table 3.1 can be interpreted that way.

Table 3.1: Average relative errors for the first and second experiment

First experiment (870 mm)						Second experiment (1000 mm)					
Measured		3D-Odometry		Std. method		Measured		3D-Odometry		Std. method	
<i>x</i>	<i>z</i>	<i>x</i>	<i>z</i>	<i>x</i>	<i>z</i>	<i>x</i>	<i>z</i>	<i>x</i>	<i>z</i>	<i>x</i>	<i>z</i>
864	175	871	188	896	209	993	0	1000	1	1038	19
873	175	876	187	904	210	1010	0	1012	5	1056	24
872	175	877	186	905	209	1015	0	1008	7	1062	25
875	175	878	185	908	208	1009	0	1008	6	1059	25
870	175	873	186	903	208	996	0	1002	4	1046	22
Average error		0.5%	6.4%	3.7%	19.2%	Average error		0.2%	2.7%	5.5%	13.2%

However, wheel slip is not the biggest source of error in these experiments. The errors are mainly in the *z* direction and they are due to the sensors offsets and non-linearities. Although we corrected the bogies angle sensors for offsets we didn't account for non linearity, which is difficult to calibrate. A difference of one degree

leads to an error of around 15 mm in the z direction for a 870 mm horizontal motion. It is approximately the height error for the first experiment. Finally, variation of the wheels' diameters and inaccuracy in the mechanical dimensions are also factors of odometric errors.

The 3D-Odometry produces much better results than the standard method in both experiments. Accounting for transitions improves the position estimation significantly. This is even more obvious when considering sharp transitions like in the second experiment. Since there are a lot of discontinuities in hazardous terrains this will help to provide usable odometric information.

Full 3D experiment

Fig. 3.8 depicts a trajectory computed online with 3D-Odometry and illustrates the full 3D capability of the method. For this experiment, the rover has been remote controlled through the scene. Only the right bogie wheels climbed the first obstacle (a) whereas the other wheels kept ground contact. Then, the rover has been driven over the second obstacle (b). The rover didn't climb the obstacle straight but with an angle of approximately 20° . The interest of such an experiment is that it forced asymmetric bogie configurations and allowed to test the full 3D capability of the method.

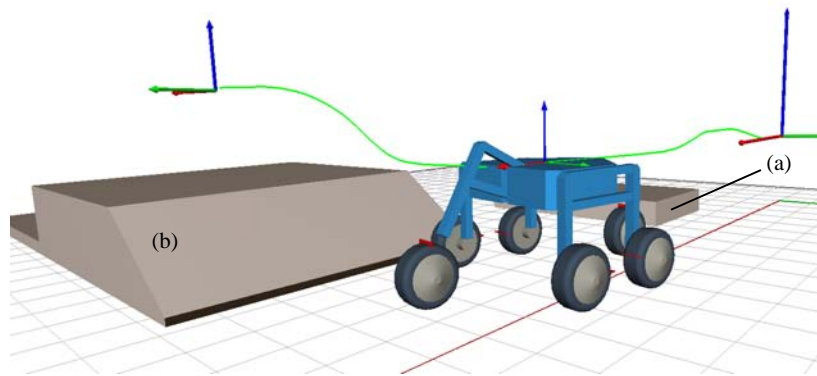


Figure 3.8: A 3D trajectory computed with 3D-Odometry from a real experiment: the obstacles a) and b) are made of wood and the ground is concrete. It illustrates the full 3D capability of the method. The trajectory follows the shape of the obstacle with good fidelity.

The true final position and orientation of the rover has been measured by hand and compared with the computed final position. The absolute position and heading angle are hand-measured as (1.43 m, -0.94 m, 0.175 m, 75°) and calculated by 3D-Odometry as (1.45 m, -0.92 m, 0.18 m, 78°). This leads to a final relative error of (1.4%, 2%, 2.8%, 4%)

3.4 Conclusion

This chapter described a new method called 3D-Odometry¹ showing better performance than the *standard method* used traditionally. The position estimation is significantly improved when the rover overcomes sharp-shaped obstacles because the method accounts for slope discontinuities and the kinematic model of the rover.

SOLERO has a non-hyperstatic mechanical structure that yields a smooth trajectory in rough terrain. As a consequence wheel slip is intrinsically minimized. When combined with 3D-Odometry, such a design allows to use odometry as a mean to estimate the rover motion in rough terrain. Moreover, the quality of odometry can still be significantly improved using a “smart” controller minimizing wheel slip. The description of such a controller is presented in the next chapter.

Of course, the combination of a good mechanical design, a “smart” controller and 3D-Odometry is not sufficient for localization because errors are still integrated over time. In order to improve robustness and decrease error growth, odometry has to be fused with other sensors. This aspect is addressed in chapter 5. However, the 3D-Odometry expands the range of speed and surface roughness over which the rover is able to go and do reasonably precise motion prediction.

1. This work has been published at the ICRA'03 conference [Lamon03].

4

Control in rough terrain

4.1 Introduction

For wheeled rovers, the motion optimization is somewhat related to minimizing wheel slip. Minimizing slip not only limits odometric error but also increases the robot's climbing performance and efficiency. In order to fulfill this goal, several methods have been developed.

Methods derived from the well known Anti-lock Breaking System (ABS) can be used for rough terrain rovers. This technique, essentially developed for the car industry, uses the information of wheel slip to correct individual wheel speed, and thus allows to limit slip. [Burg97] propose to adapt the method to rough terrain by considering the rover attitude and the load on the wheels. A set of accelerometers and encoders are proposed for measuring individual wheel slip. However, because the velocity of rovers in rough terrain is generally very low, the signal/noise ratio of the accelerometers is small and the estimation of wheel slip is not accurate. Another method has been implemented on the Nasa FIDO rover [Baumgartner00]. It is based on a velocity synchronization algorithm which minimizes the effect of the wheels "fighting" each other. The first step of the method consists in detecting which of the wheels are deviating significantly from the nominal velocity profile. Then a voting scheme is used to compute the required velocity set point change for each individual wheel. This technique accounts neither for the kinematics nor for a physical model of the rover. Furthermore, the method attempts to adapt wheel speed when slip already occurred. [Peynot03] propose to avoid creating any slipping situation due to non-relevant wheel speed references. The technique is similar to [Baumgartner00] but provides better performance in rough terrain because it accounts for the kinematic model of the rover.

These methods, which will be referred later as *reactive* methods, have a common point: no wheel-soil interaction models are used. Thus, they are expected to work on various types of terrain. However, performance might be improved by considering both the physical model of the rover and wheel-soil interaction models for specific types of soils. Thus, the traction of each wheel is optimized considering the load distribution on the wheels and the soil characteristics. These methods are called *predictive* methods.

Only a few publications concerning physics-based motion control in rough terrain can be found in the literature. A good overview is presented in [Iagnemma00] [Iagnemma01] [Iagnemma02] and [Hung00]. [Yoshida02] propose a method minimizing slip ratios and thus avoid soil failure due to excessive traction. The method requires to estimate the velocity of each wheel w.r.t the ground, which is difficult to measure in rough terrain.

The physics-based controllers assume that the parameters of the wheel-ground interaction models are known. Unfortunately, these parameters are difficult to estimate and are valid only for a specific type of soil and condition. [Iagnemma02] propose a method for estimating the soil parameters as the robot moves, but it is limited to rigid wheel traveling through deformable terrain. In practice, the rover wheels are subject to roll on different kind of soils, whose parameters can change quickly. Thus, physics based controllers are sensitive and difficult to implement on real rovers.

In this chapter, a method combining the advantages of an ABS-based and physics-based method is proposed. As a consequence, no complex wheel-soil interaction model is required and the load distribution on the wheels is considered. The next section of the chapter describes the approach to model a complex wheeled rover. Then, the model is used to select the optimal set of wheel-torques minimizing slip (section 4.3). Section 4.4 presents the algorithm allowing to avoid the use of complex wheel-soil interaction models. The method is tested and compared with a reactive controller in section 4.5. Finally, the technical aspects related to the wheel-ground contact estimation are addressed in section 4.6 and 4.7 concludes the chapter.

4.2 Quasi-static model of a wheeled rover

The speed of an autonomous rover must be limited in rough terrain in order to avoid high shocks in the structure and for safety reasons. Furthermore, the navigation algorithms are computationally expensive (image processing, path planning, obstacle avoidance, etc.) and the onboard processing power is limited: this requires the rover to move slowly. In this range of speeds (typically 5 to 20 cm/s), the dynamic forces might be neglected and a quasi-static model is appropriate. Such a model can be solved for contact forces and motor torques knowing the state of the robot and the wheel-ground contact angles.

To develop such a model, the mobility analysis of the rover's mechanical structure has to be performed (4.2.1). It ensures to produce a consistent physical model with the appropriate degrees of freedom at each joints. Then the forces are intro-

duced and the equilibrium equations are written for each part composing the rover's chassis (section 4.2.2).

4.2.1 Mobility analysis

The mobility of a rolling robot in straight motion should ideally be one, indicating that the robot can move in a constrained direction. Grubler's Mobility Equation in three dimensions [Mabie87], also known as the Kutzbach Criterion, can be described as

$$MO = 6 \cdot n - 5 \cdot f_1 - 4 \cdot f_2 - 3 \cdot f_3 - 2 \cdot f_4 - f_5 \quad (4.1)$$

where n is the number of mechanical parts and f_j the number of joints of each type ($j = 1, \dots, 5$ for example f_1 : the number of pin joints, f_3 : the number of spherical joints). The mobility equation is a guideline for determining if a system is statically determinate. Many real systems contain redundancy in links and joints resulting in hyperstaticity. A four-legged table, for example, is statically indeterminate if considered rigid. More sophisticated modeling methods are required to analyze the distribution of forces in a hyperstatic system. Another approach is to model selective joints with additional degrees of freedom. Intelligent selection of these joints can minimize the error associated with a quasi-static solution. While the modeled kinematic chain is a simplification, it can be good enough to support motor control.

Mobility analysis of SOLERO

In a first step, one can consider the wheel-ground contacts as spherical joints and all the pin joints in the mechanism as one degree of freedom (DOF) revolute. For the SOLERO, the calculation of the mobility using 4.1 is -20 rather than 1. The system is, therefore, significantly hyperstatic and requires a modified model for a possible quasi-static solution. Two significant modifications to joint degrees of freedom assist the model.

The first one involves the representation of the wheel-ground joint mobility. For a standard wheel without slip, the joint that represents the wheel-ground contact can be modelled as a spherical joint allowing three degrees of freedom (rotations about the three axes). Motor torque on the wheels will directly affect the forces in that contact plane. Lateral forces are not influenced by the motor torque. Therefore, the system was modelled with the lateral forces being carried by the wheel fixed to the body (rear wheel) and the wheel on the front fork. The wheels on the bogies were modelled with no resistance in the lateral direction (4 degrees of freedom). We have chosen such a model because there is not enough information to

assess how the lateral forces are distributed amongst all the wheels and because the error due to the simplification has almost no influence for the controller.

The second modification acts on the representation of the redundant kinematic chains. It is possible to model selected joints on redundant kinematic chains with more degrees of freedom. This results in forces being transmitted through direct flow patterns. Because the model is being used to optimize motor torques, inaccuracies in the internal linkage forces have minimal effect.

Fig. 4.1 shows the resulting kinematic model of the SOLERO. The numbers at the link connections indicate the degrees-of-freedom of that joint.

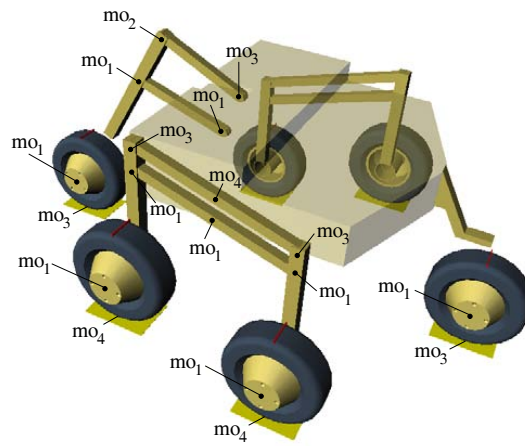


Figure 4.1: Final representation of the mobility of the joints

The resulting model is mechanically equivalent to the real structure. The mobility of the bogies and the fork is one and the final mobility can be calculated using equation 4.1 to produce

$$MO = 6 \cdot 18 - 5 \cdot 14 - 4 \cdot 1 - 3 \cdot 7 - 2 \cdot 6 = 1 \quad (4.2)$$

4.2.2 A 3D static model

For a 3D static model, 6 equations (3 torques and 3 forces) are applied to each body, containing ground reaction forces, gravity forces (weight) and external forces. Dynamic forces are considered to be negligible because the speed is low. The mobility analysis is used to introduce the right number of forces/torques at each joint. In three dimensions, the number of generalized forces (n_g) to introduce follows the rule

$$n_g = 6 - mo \quad (4.3)$$

where mo is the mobility of the joint. For example, five generalized forces are introduced for a pin joint ($mo = 1$): only one rotation is free, all the translations (3 forces) and the remaining rotations (2 torques) are blocked.

Model of SOLERO

SOLERO has 18 parts and is characterized by $6 \times 18 = 108$ independent equations describing the static equilibrium of each part and involving 14 external ground forces, 6 internal wheel torques and 93 internal forces and torques for a total of 113 unknowns. The weight of the fork and the bogies link has been neglected whereas the weight of the main body and the wheels is considered.

Of course, it is possible to reduce this set of independent equations because we have no interest in implicitly calculating the internal forces of the system. The variables of interest are the 3 ground contact forces on the front and the back wheel, the 2 ground contact forces on each wheel of the bogies and the 6 wheel torques. This makes 20 unknowns of interest and the system can be reduced to $20 - (113 - 108) = 15$ equations. This leads to the following matrix equation

$$M_{15 \times 20} \cdot \vec{U}_{20 \times 1} = \vec{R}_{15 \times 1} \quad (4.4)$$

where M is the model matrix depending on the geometric parameters and the state of the robot, U a vector containing the unknowns and R a constant vector. The details of the model together with the mechanical parameters are described in Appendix A.

4.3 Torque optimization

It is interesting to note that there are more unknowns than equations in 4.4. That means that there is an infinite set of wheel-torques guaranteeing the static equilibrium. This becomes obvious when considering that one motorized wheel is enough to make the robot move. This characteristic can be used to control the traction of each wheel and select, among all the possibilities, the set of torques minimizing slip. Other functions, such as energy can be used instead. In this chapter, we focus on slip minimization and this section describes the concepts and the algorithms.

4.3.1 Wheel slip model

The intent is to formulate a holistic model of a robot to control the wheel motor torques in order to minimize wheel slip. Therefore it is helpful to review the governing equations on wheel slip and explain which function should be minimized to reach this goal. The model presented in what follows assumes a rule of proportion-

4.3 TORQUE OPTIMIZATION

ality between the traction and the normal force on the wheel: the more pressure on the wheel the more traction it can carry before slipping. This proportionality rule is not perfectly verified in all circumstances. However, such a model is valid in most of the cases and is appropriate because we are not interested in exactly computing the forces at the interface but minimizing wheel slip.

Fig. 4.2 shows the common forces acting on the wheel of a mobile robot.

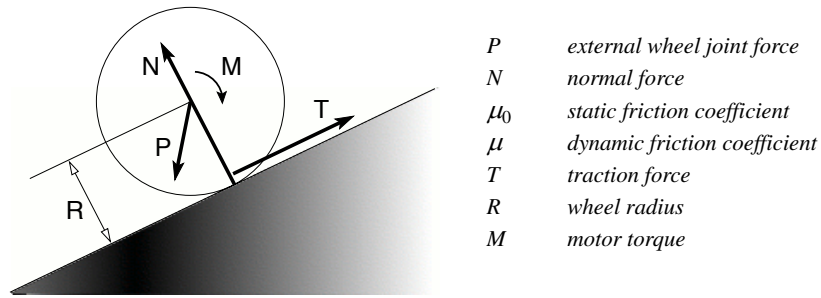


Figure 4.2: Acting forces on a wheel

The wheel is balanced if the friction force fulfils the inequation 4.5. This case represents static friction. If the static friction force can't balance the system, the wheel slips and the friction force is set by equation 4.6.

$$F_{static} \leq \mu_0 \cdot N \quad (4.5) \quad F_{dynamic} = \mu \cdot N \quad (4.6)$$

In order to avoid wheel slip, the friction force which depends directly from the motor torque, M , should satisfy

$$T = \frac{M}{R} \leq \mu_o \cdot N \quad (4.7)$$

The above equations suggest that there are two ways to reduce wheel slip. First, assume that μ_0 is known and set

$$T \leq \mu_o \cdot N \quad (4.8)$$

In fact, it is difficult to know μ_0 precisely because it depends on the kind of wheel-soil interaction. During exploration, the kind of soil interacting with the wheels isn't known which makes μ_0 impossible to pre-determine.

Another way to avoid wheel slip is to first assume that the wheel does not slip. It is then possible to calculate the forces T and N as a function of the torque and the result is optimized in order to minimize the ratio T/N .

Accounting to this assumption

$$\frac{T}{N} = \frac{\mu_n \cdot N}{N} = \mu_n \quad (4.9)$$

μ_n is similar to a friction coefficient. In minimizing this ratio, then minimizing μ_n , we optimize our chances that this coefficient is smaller than the real friction coefficient μ_0 . If this is the truth, there is no slip. Therefore, it is possible to minimize the ratio T/N without knowing the real static friction coefficient. The second method is used here, because it is more robust.

4.3.2 Optimization algorithm

The controllable inputs of the system are the six wheel torques. Since there are five more unknowns than equations it is possible to write an equation expressing the torques as linearly dependant (a proof is presented in Appendix A.5.1). The 14 other equations define the external forces as a function of the torques.

The model of SOLERO is indeterminate because there are less equations than variables and the set of solutions is of dimension five (number of wheels -1). The goal of the optimization is to minimize slip. This can be achieved by maximizing the traction forces, which is equivalent to minimizing the function $\max(T_i/N_i)$ for the wheels. Since it is difficult to do reasoning in five dimensions, a simpler 2D robot referred to as ThreeWheels (see Fig. 4.3) is used to present our optimization algorithm. This process will then be extrapolated for the complete model.

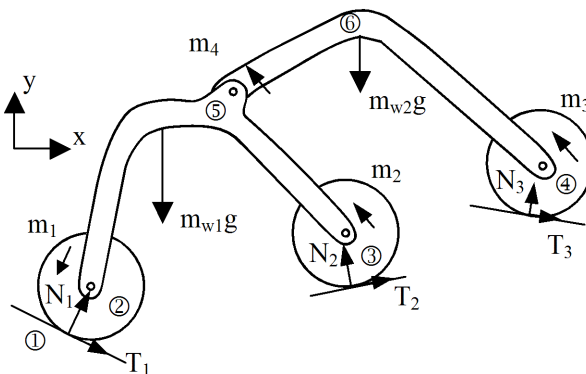


Figure 4.3: The ThreeWheels 2D model. This rover belongs to the passively suspended robots family. m_4 is a non-controllable torque generated by a torsion spring with known

The model of the ThreeWheels rover has nine unknowns: two forces and one torque on each wheel (m_4 is a known spring suspension torque and depends directly on the geometry) and seven equations: three global equations, one torque equa-

4.3 TORQUE OPTIMIZATION

tion for each wheel and one torque equation for the fork. That means that the solutions space is of dimension two. Equation 4.10 express the forces on the wheels and 4.11 the torque of the first wheel as a function of m_2 and m_3 (m_1 , m_2 and m_3 are linearly dependant). α , β , γ , ε and δ are parameters depending on the rover's state and geometry (see A.5).

$$\begin{aligned} N_i &= \alpha_{1i} \cdot m_2 + \beta_{1i} \cdot m_3 + \gamma_{1i} & \text{with } i = 1,2,3 \\ T_i &= \alpha_{2i} \cdot m_2 + \beta_{2i} \cdot m_3 + \gamma_{2i} \end{aligned} \quad (4.10)$$

$$m_1 = \varepsilon_1 \cdot m_2 + \varepsilon_2 \cdot m_3 + \delta \quad (4.11)$$

$$f(m_2, m_3) = \max\left(\frac{T_1}{N_1}, \frac{T_2}{N_2}, \frac{T_3}{N_3}\right) = \max(\mu_1, \mu_2, \mu_3) \quad (4.12)$$

The solution space of the ThreeWheels rover is depicted in Fig. 4.4a. It corresponds to the function f defined by 4.12, which is the function to minimize.

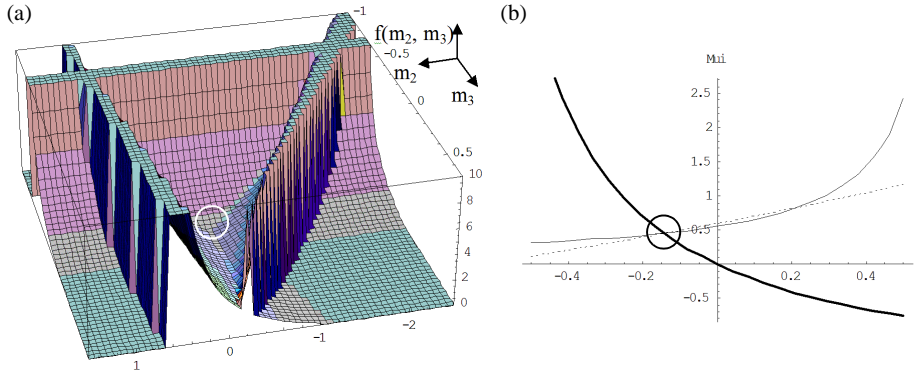


Figure 4.4: Solution space for the ThreeWheels rover. The functions μ_1 , μ_2 and μ_3 are hyperbolic and a linear optimization process is not possible. **a)** optimal solution (circled) minimizing slip and fulfilling the $N_i > 0$ constraint **b)** cross section of figure (a) for $m_2 = -0.3$. One can see that the optimal solution (circled) corresponds to equal μ 's.

Since the system of equations is non-linear, a numerical method is implemented. Our optimization method uses a combination of different algorithms and is depicted in Fig. 4.5. Firstly, the Equal Torques solution (see A.8) is checked versus the following constraints

- Motors saturation: the torques of the optimal solution must be smaller than the maximal possible torque.
- Normal forces: the normal forces N_i must be greater than zero. The asymptotes of the hyperbolic functions in Fig. 4.4a define the sign inversion limit.

If this solution is valid, it is taken as the initial solution for the Fixed Point optimization (A) (see Fig. 4.5). If it doesn't fulfill the constraints, a valid initial solution is computed using the Simplex Method (B). The optimal solution is then provided either by (A) or the Gradient optimization (C). We have chosen this scheme because most of the states are handled by (A), which is computationally very light in comparison with a single, non-linear optimization algorithm.

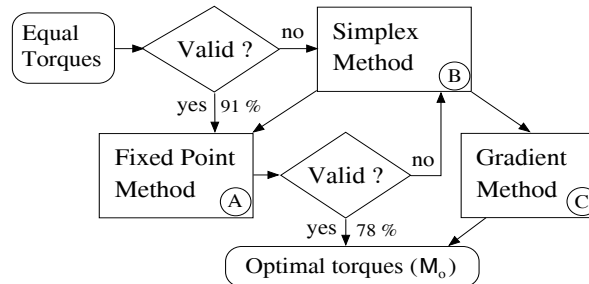


Figure 4.5: Optimization algorithm. The execution times for the algorithms A, B and C are 6 ms, 5 ms and 20 ms respectively (1.5 GHz processor). The worst case is about 31 ms but the majority of the states (70%) are handled in only 6 ms.

A. Fixed Point optimization

This optimization method is based on the fixed point algorithm. The aim of this algorithm is to numerically find an intersection of curves when an analytical solution is difficult to obtain. In our case, the optimal solution corresponds to the intersection of μ_1 , μ_2 and μ_3 . The corresponding flow chart is presented in Fig. 4.6.

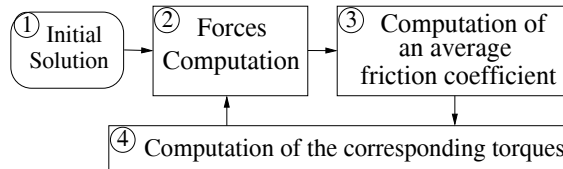


Figure 4.6: Fixed point based algorithm. The quasi-static model (2) is solved with an initial set of torques (1). Block (3) computes an average friction coefficient based on the computed forces (output of block 2). The corresponding torques are computed (4) and fed again in block (2). Twenty iterations are generally sufficient for convergence.

This algorithm is computationally light and provides good results for most cases. Nevertheless, it diverges sometimes and doesn't account for the aforementioned constraints. This can lead to torques that cannot be provided by the motors.

B. Simplex method

This method is based on the Simplex algorithm which solves linear programs in a constrained solution space. The Simplex method tries to maximize an objective

function considering a set of constraints on the variables. In our case, the algorithm is only used to provide a valid initial solution, thus many objective functions can be used. However, in order to get closer to the final optimal solution, we choose the function \mathbf{h} defined in 4.13, which tends to minimize the ratio T_i/N_i .

$$h = \sum N_i \quad (4.13)$$

Furthermore, the function \mathbf{h} is linear because it is a linear combination of the torques. The solution provided by this method is guaranteed to fulfill the constraints and can be used as a starting point for both the Gradient and Fixed point optimization.

C. Gradient optimization

This algorithm seeks for an optimum in the constrained solutions space given a known valid initial solution. The Gradient optimization is similar to the potential field method: at each step the gradient is computed and the next solution is generated following the maximum slope.

4.3.3 Torque optimization for SOLERO

The optimization for the 3 dimensional SOLERO is similar to the method presented in the previous section. The solution space has now five dimensions and one has to account for 18 constraints (with $i = 1, \dots, 6$ and $MaxTrq$ the saturation torque).

$$m_i < MaxTrq \quad m_i > -MaxTrq \quad N_i > 0 \quad (4.14)$$

An example of computed forces and torques is depicted in Fig. 4.7.

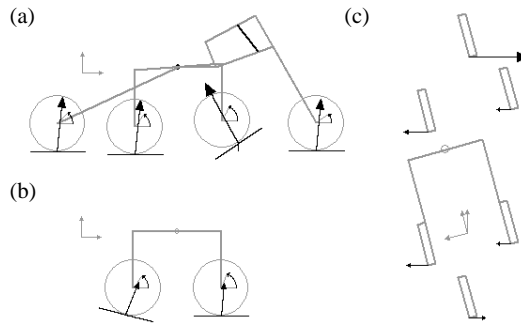


Figure 4.7: Forces and torques computed by the optimization procedure. The forces are expressed in the global coordinate system. The user can interactively change the state of the robot and the contact angles of the wheels **a**) side view: the pitch, the front fork and the left bogie angles can be modified **b**) right bogie view: the angle of the right bogie can be modified in this view **c**) decomposed view from rear: the roll angle can be modified. In this view, the arrows represent the projections of the reaction forces in the global frame of reference.

The optimization algorithm has been tested for around twenty thousand rover postures. The different states have been generated automatically by varying each input parameter (i.e. the wheel-ground contact angles, roll and pitch, the fork, left and right bogies angles) in order to cover most of the robot's configuration space. For each state, the minimal friction coefficient (μ_n) has been computed and fed into the histogram printed in Fig. 4.8.

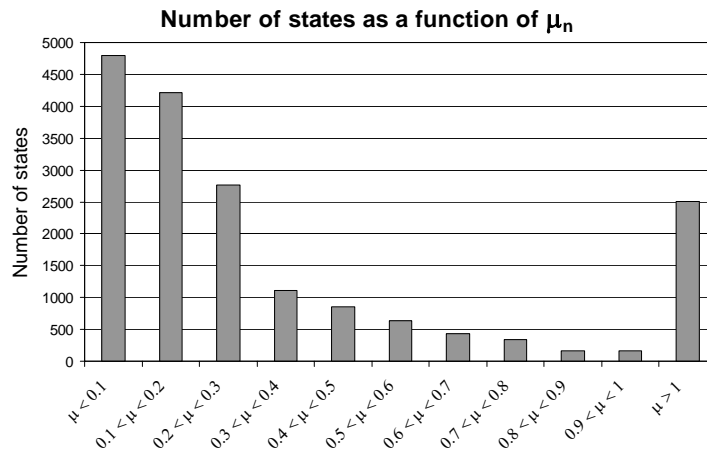


Figure 4.8: Statistics of the optimization. The histogram plots the number of states as a function of the friction coefficient. 80% of the states correspond to a friction coefficient smaller than 0.6.

It is interesting to note that, in 80% of the cases, the friction coefficient μ_n is smaller than 0.6 (the static friction coefficient of a tire on a dry road is around 0.6). As said in section 4.3.1, if μ_n is smaller than the real friction coefficient μ_0 then there is no slip. Thus, there is not slip for 80% of the chassis configurations when the rover is traveling on a terrain with a friction coefficient higher or equal to 0.6 (e.g. a rocky terrain). For more slippery soils it becomes more and more difficult to guarantee no slip. However, the exponential decay of the histogram is favorable and the probability of slip is always minimized whatever the soil type.

The bar of the histogram corresponding to friction coefficients higher than one groups pathological cases reflecting extreme situations where it is difficult to keep static equilibrium. In such circumstances, it is impossible to avoid slip because, in reality, a friction coefficient is always smaller than one. This is not critical in practice because, at a higher lever, the path planner avoids areas in which the rover has a risk to reach such extreme configurations [Bonnafous01]. Thus, such cases can be discarded from the statistics.

4.4 Rover motion

Rolling resistance is another important aspect to the quasi-static model, and is therefore reviewed here. A static model balances the forces and moments on a system to remain at rest or maintain a constant speed. Such a system is an ideal case and does not include resistance to movement. Therefore, an additional torque compensating the rolling resistance torque must be added on the wheels in order to complete the model and guarantee motion at constant speed. This results in a quasi-static model.

Several rolling resistance models are developed in the literature and can be incorporated in the static model to ensure constant speed motion. A rolling resistance model for an elastic wheel on an elastic soil is presented in [Kalker90]. Other models applicable for rigid wheels and deformable soils such as sand or earth can be found in [Bekker56], [Bekker69] and [Andrade98]. In practice, the parameters of these models are generally difficult to estimate and are valid only for a specific type of soil and condition. Furthermore, the behavior of the controller is difficult to predict when wrong parameters and/or models are used: what would happen if a controller designed for sand is used on rock? Because an exploration rover is subject to deal with different types of terrain, using a controller endowing a wheel-ground interaction model specific to one type of soil is generally not appropriate. Fig. 4.9 is a good illustration: when driving on such a terrain some wheels might roll on sand and some others on bare rock. Furthermore, the grit and compactness of the sand changes depending on the local conditions.

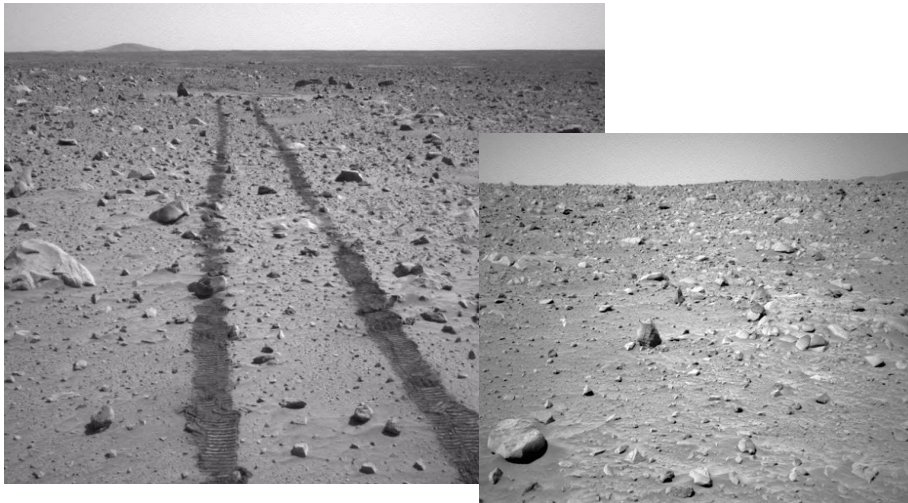


Figure 4.9: *Images of Mars taken by Spirit next to the Bonneville Crater.*

In order to avoid relying on such complex wheel-ground interaction models (whose parameters are unknown), we have introduced a global control loop for estimating the rolling resistance as the robot moves. The final controller, minimizing wheel slip and incorporating rolling resistance, is depicted in Fig. 4.10

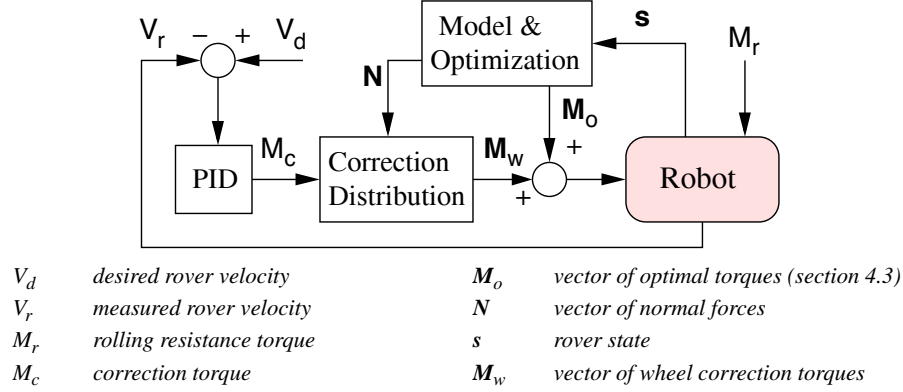


Figure 4.10: Rover motion control loop. The global loop is a speed control loop whereas the local controllers for the wheels are torque controllers. The vector s includes the wheel-ground contact angles, the internal links and the roll and pitch angles.

The kernel of the control loop is a PID controller. It allows to estimate the additional torque to apply to the wheels in order to reach the desired rover velocity V_d and thus, minimizes the error $V_d - V_r$ ¹. M_c is actually an estimate of the global rolling resistance torque M_r , which is considered as a perturbation by the PID controller. The rejection of the perturbation is guaranteed by the integral term I of the PID. We assume that the rolling resistance is proportional to the normal force, thus the individual corrections for the wheels are calculated by

$$M_{w_i} = \frac{N_i}{N_m} \cdot M_c \quad (4.15)$$

where N_i is the normal force on wheel i and N_m the average of all the normal forces. The derivative term D of the PID allows to account for non modeled dynamic effects and helps to stabilize the system. The parameters estimation for the controller is not critical because we are more interested in minimizing slip than in reaching the desired velocity very precisely. For locomotion in rough terrain, a residual error on the velocity can be accepted as long as slip is minimized. Furthermore, the system offers an intrinsic smoothing because the ratio between inertia and motor torques is large.

1. V_r can be estimated using the sensor fusion method presented in chapter 5.

4.5 Experimental results

A simulation phase has been initiated in order to test the algorithms and verify the theoretical concepts and assumptions. The simulation parameters have been set as close as possible to the real operation conditions. However, the intent is not to get exact outputs but to compare different control strategies and detect/solve potential implementation problems.

4.5.1 Simulation tools

Simulations have been realized with the Open Dynamics Engine [ODE]. This engine is a platform independent and open source library that simulates rigid body dynamics in three dimensions. It has advanced joint types and integrates collision detection with friction. The source code being available it is possible to integrate more sophisticated simulation models such as rolling resistance, friction in the joints etc. In this application, a rolling resistance proportional to the normal force on the wheel has been implemented.

The simulation tools allow to test and compare different traction control strategies. In our experiments, wheel slip has been taken as the main benchmark and the performance of our controller (predictive control) has been compared to the controller presented in [Baumgartner00] (reactive control). As said before, the reactive controller implements speed controllers for the wheels whereas the torques of the wheels are controlled for the predictive method.

4.5.1.1 Wheel slip

The slip of wheel i at time step k can be computed with

$$s_k^i = \Delta w_{(k-1,k)}^i - \Delta \theta_{(k-1,k)}^i \cdot R \quad (4.16)$$

where $\Delta w_{(k-1,k)}^i$ is the true wheel displacement, $\Delta \theta_{(k-1,k)}^i$ the angular change and R the wheel radius. The total slip of the rover integrated during an experiment is defined as

$$S = \sum_{i=1}^6 \sum_k s_k^i \quad (4.17)$$

4.5.1.2 Wheel-ground contact angles

The body collision algorithm of ODE provides n contact points around the wheel together with the normal forces. This data is similar to what can be measured with a tactile wheel (the wheel deflection is more or less proportional to the applied

force) and the same method as presented in section 4.6 is applied for computing the contact angles. In some rare cases, no contact point is provided by ODE at time k . It is either that the wheel does not touch the ground or that the body collision algorithm fails to compute contact between the wheel and the ground (a 3D mesh). In these situations, assuming slow motion and a small simulation period (10 ms in our case), we take the same contact angle for time k and $k - 1$.

4.5.2 Experiments

Two sets of simulation experiments have been conducted. The first set comprises different terrain profiles in two dimensions (in the plane x - z) and the second, full 3D environments. In both cases, the nominal speed of the rover is 0.1 m/s and the friction coefficient has been set equal to 0.7.

A. Experiment set of type one

Terrain profiles similar to the one depicted in Fig. 4.11 have been generated and the simulation performed with both torque and speed control. Thanks to the terrain symmetry, the trajectories of the gravity center are the same whatever control type is used.

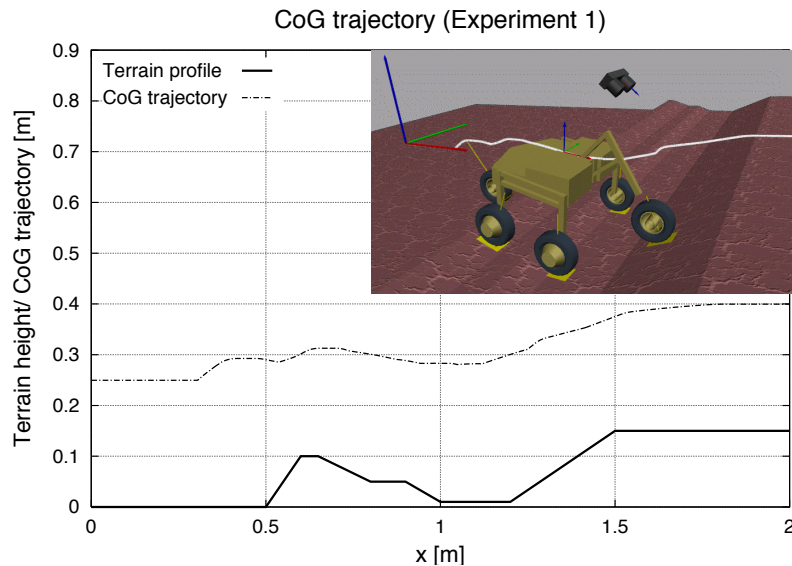


Figure 4.11: Trajectory of the center of gravity for an experiment of type one. That kind of terrain is difficult for a wheeled rover because it includes many sharp slope changes.

Fig. 4.12 depicts typical results that have been obtained on such terrains. For a specific wheel, slip can be locally higher with torque control than with speed control. However, the total slip remains always smaller with torque control for all the experiments. Another interesting result is that the difference between the two methods increases when the friction coefficient gets lower. In other words, the advantage of using torque control becomes more and more interesting as the soil gets more slippery

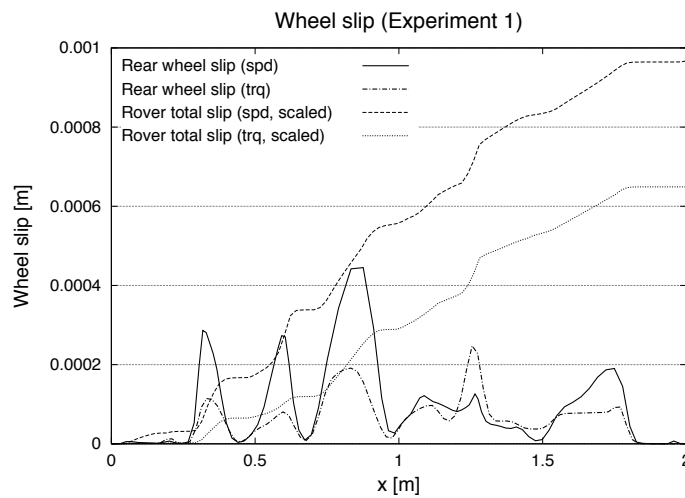


Figure 4.12: Total slip and rear wheel slip for both reactive (*spd*) and predictive (*trq*) control. Total slip is scaled by a factor of 500. Locally wheel slip can be bigger with torque control but the total slip remains always smaller: 31% better than speed control.

B. Experiment set of type two

Here, full three dimensional terrains are used for the experiments. They have been generated randomly with step, sinus, circle and particle deposition functions. This time, because the terrains are not symmetric, the trajectory of the rover depends on the control strategy. Therefore it is difficult to compare performance between predictive and reactive control. However, we have considered an experiment as valid when the distance between the final positions of both trajectories is smaller than 0.1 m for a total distance of around 3.5 m. This distance is small enough to allow performance comparison. For all the valid experiments, predictive control showed better performance than reactive control. In some cases the rover was even unable to climb some obstacles and to reach the final distance when driven with reactive control. Otherwise, the simulations lead to the same conclusions as for the experiments of type one. Fig. 4.13 depicts one of the terrains used for the simulations and Fig. 4.14 the corresponding results.

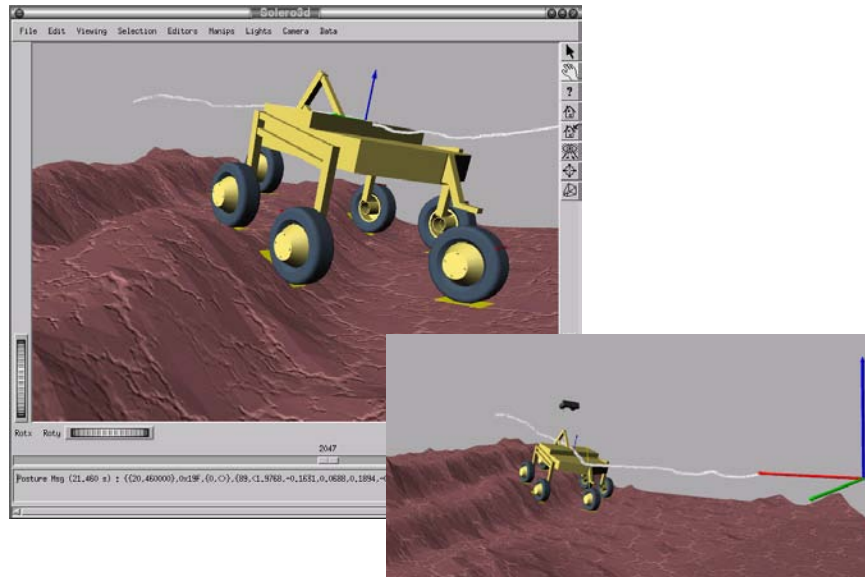


Figure 4.13: Snapshots of an experiment of type two. The total travelled distance along x is 3.5m. That kind of terrain is challenging for a wheeled rover because there is much side slip when the rovers start climbing the slope.

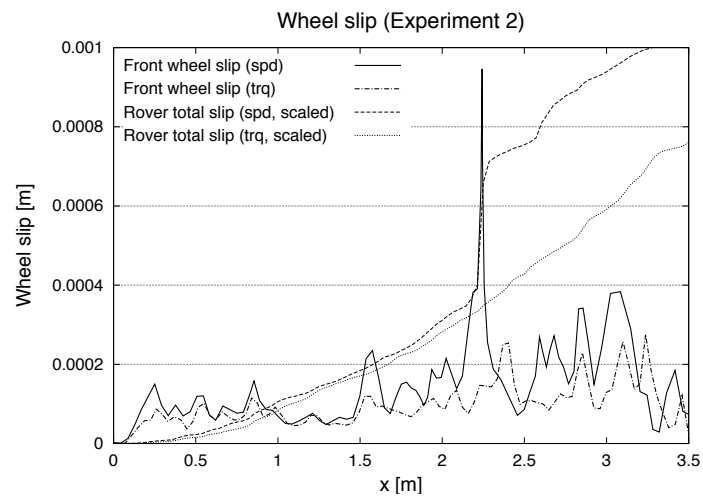


Figure 4.14: Total slip and front wheel slip for an experiment of type two. The difference gets bigger as the rover deals with true rough terrain. Total slip is scaled by a factor of 800. At the end of the experiment, the torque controller performs 26% better than the speed controller.

4.5.3 Discussion

For these experiments, it is difficult to provide a quantitative result to compare the performance of one controller with respect to the other. Indeed, the performance depends on the topography: for easy terrains the performance of both controllers is almost the same, whereas torque control performs better as the terrains become more challenging. However, very interesting behaviors of the torque controller have been systematically observed in all the experiments

- For each wheel, the slip signal is scaled down when using torque control. Such a behaviour can be observed in Fig. 4.12 and 4.14: the peaks are at the same places for both controllers but the amplitude is much smaller for the torque controller.
- The total slip of the rover is always smaller when using torque control.
- Strong assumptions have been used during the development of the torque controller i.e. no slip and the wheels touch the ground all the time. During the experiments both assumptions have been violated but the system was able to recover and keep its stability, even in difficult situations such as depicted in Fig. 4.13.

Finally, the simulations showed good results and promising perspectives. Furthermore, they allowed to detect potential problems and address implementation details. This is a step closer to the real application.

4.6 Wheel-ground contact angles

A key parameter required for traction optimization algorithms is the contact angles between the wheels and the ground. There are many ways of sensing or calculating these angles. The method presented in chapter 3 and the one described in [Iagnemma01] are similar in a way that they both consider the displacement/velocity of each wheels for computing the angles. The quality of the estimation provided by such methods strongly depends on wheel slip and terrain profile. In particular, no estimation can be computed when the rover is at stand still and poor results are obtained in slowly changing terrain profiles. [Peynot03] use the information of the global rover motion for the estimation of the contact angles and thus limit the sensitivity to individual wheel slip. However, all these indirect methods for computing the wheel-ground contact angles assume no slip, or account for an accurate rover velocity information. Therefore they are all subject to the chicken and egg paradigm: bad wheel-ground contact angle estimations lead to unadapted motor commands, which implies wheel slip and bad angles. As a consequence, direct measurement of wheel-ground contact angles is required in order to be indepen-

dent of the terrain profile and characteristics and to guarantee the systems stability. An alternative to these indirect estimation methods is to directly measure the forces on the wheels. This can be done using flexible wheels equipped with sensors measuring deflection. It has the advantage to provide the contact points for static conditions also. An example of such a device is depicted in Fig. 4.15 and more information can be found in [Lauria02]

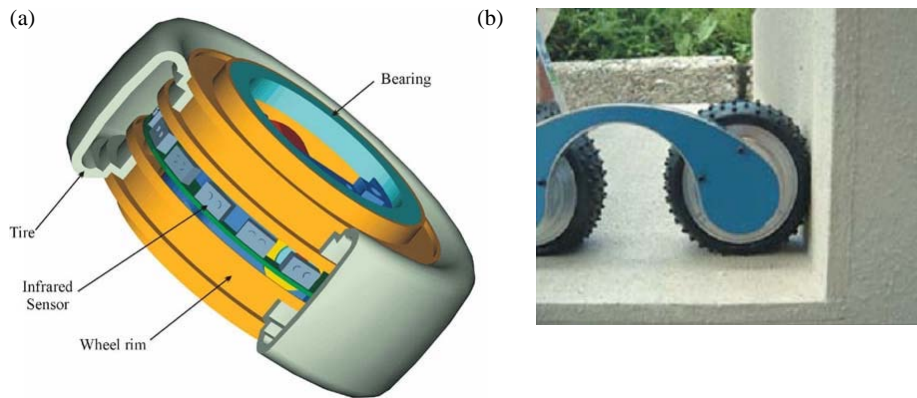


Figure 4.15: *The tactile wheel (developed at EPFL by Michel Lauria). a) Sixteen infrared proximity sensors measure the tire compression all around the wheel. b) picture of the front wheel of the robot Octopus, equipped with tactile wheels.*

With such a wheel, the contact angles are computed with a weighted mean of the sensor signals. This way, a smooth transition is obtained when dealing with difficult terrain profile such as depicted in Fig. 4.15 b.

There are two other advantages of including deflection sensors in the wheels

- improvement of the accuracy of the 3D-Odometry: **a)** direct measurement provides better estimates of the contact angles, better than the one computed using 3.1 **b)** the sensors allow to measure the effective wheel radiuses, which are required inputs for 3D-Odometry (the accuracy of the odometry is very sensitive to these parameters)
- improvement of the torque controller: the deflection of the wheel at a given contact point is an image of the applied force at that point. This information can be incorporated to the model in order to improve the estimates of the normal forces, which are used in 4.15.

4.7 Conclusion

Most of the physics based control methods rely on the knowledge of a specific wheel-soil interaction model. However, in a real application, the parameters of such a model are unknown because the rover is subject to deal with different types of soils i.e. sand, rocks, gravel, grass and a mix of all of them. An error on the parameters estimation has a direct impact on the performance of the controller.

In this chapter, a quasi-static model of a six-wheeled rover together with an optimization method to minimize slip have been presented¹. Unless other control strategies, the proposed method does not require the use of soil models. As a consequence, the rover is able to operate on different types of soils: this is the main requirement for exploration missions. Furthermore, our approach can be adapted to any kind of rover and the needed processing power remains relatively low, which makes online computation feasible. The simulations show promising results and the system is mature enough to be implemented on the rover for real experiments.

An interesting aspect of such a controller is that the normal forces are computed and can be used to associate a slip probability for each wheel: the less pressure on the wheel, the more likely the wheel slips. The slip probability can then be propagated through the 3D-Odometry equations to finally obtain the covariance matrix for the robot's displacement, which is a valuable information for probabilistic multi-sensors fusion (see next chapter).

1. This work has been published at ICRA [Lamon04a] [Lamon05]

5

Position tracking in rough terrain

5.1 Introduction

A good pose estimate is essential for an autonomous mobile robot because position is used by most of the navigation tasks and algorithms running onboard. The first step in localization consists in the integration of high frequency dead reckoning sensors to predict the vehicle motion. The second phase, which is usually activated at a much slower rate, uses an absolute sensing mechanism for extracting relevant features in the environment and updating the predicted position. One of the big challenges of this update is to associate data between the current and previously extracted landmarks and features. This task needs good pose prediction in order to provide reliable results and a minimal number of false matches. This requirement is even more important when the robot travels over cluttered terrain, where the field of view can vary significantly between two features extraction steps.

The intent of this chapter is to develop a method for combining different sources of information in order to provide a robust three-dimensional initial estimate of the six degrees of freedom of a rough terrain rover. This probabilistic method, based on an extended information filter, is presented in section 5.4. Section 5.2 gives a survey of the sensors that can be used in rough terrain and section 5.3 presents the problematic of having sensors distributed at different places on the rover. Section 5.5 presents the experimental results, validating the theory. Finally, section 5.6 concludes the chapter.

5.2 Sensors for outdoors

The aim of this section is to give an overview of sensors that can be used for outdoor applications and emphasize on the difficulties involved in motion perception in unstructured and unknown environments.

The family of 1D/2D distance sensors such as ultrasound and 2D laser scanners are commonly used indoors (structured) but are generally not well adapted for outdoors (unstructured). In structured environments, we can generally assume that a rover moves on a flat ground and that its working space is delimited by walls, per-

pendicular to the soil. This strong assumption allows using simple world representations (2D maps) and distance information to detect obstacles and localize the rover. Simple features such as corners, lines and segments can be easily extracted from raw data and are relevant enough for autonomous navigation [Tomatis01][Arras03]. When dealing with unstructured environments (3D world), these sensors don't provide enough information and the interpretation of data becomes tedious because of the lack of a priori knowledge. However, these distance sensors can be used to detect contingency situations: for example, the case when the rover gets too close to an obstacle.

Monocular vision (a single camera) provides a lot of information and has the advantage of covering a large field of view: from very close distances to the horizon (this is not the case for distance sensors, which are limited to a predefined range). Many applications use monocular vision as a source of information for localization. When enhanced with a parabolic or equiangular shaped mirror, the field of view of a camera can be extended to 360°. That kind of panoramic vision system is used in [Strelow01] for tracking the six degrees of freedom of a robotic platform. In [Cozman00], the skylines extracted from panoramic views are used to localize a mobile robot, provided a topographic map is available. However, monocular vision provides only scaleless information. It is enough for topological localization but the information has to be completed with metric data for metric localization.

Stereovision provides range images and is today the most used sensor for outdoor applications. It allows for the creation of traversability maps and provides estimation of the rover's ego-motion (visual odometry). The principle of visual odometry is to compute an estimate of the six displacement/orientation parameters between two stereo frames on the basis of a set of 3D point-to-point matches (see Appendix D). The matches are established by tracking the corresponding pixels in the image sequence acquired while the robot moves. However, the use of stereovision has some limitations: it works well only in environment with enough texture, its range is limited and the images are subject to be affected by bad illumination conditions, making the visual odometry unavailable.

As presented in chapter 3, 3D-Odometry is a method of motion estimation for an all-terrain rover. It has been shown that it can be used on uneven terrains if the mechanical structure of the rover is adapted. However, because of wheel slip, the position estimation error can grow quickly and this technique cannot be used alone.

An IMU (Inertial Measurement Unit) is a device measuring accelerations and rotation rates in three dimensions. In the presence of a known gravity field, the attitude of the rover can be estimated without any drift. The double integral of the accelerations and single integral of angular rates are computed to track the position

and the orientation of the body on which it is mounted. However, the pose estimation diverges quickly because the signals are affected by biases and errors integrated over time. Thus, such a sensor cannot be used alone to estimate motion and has to be combined with other sensors to update the biases and scaling errors.

Heading sensors such as compasses are of high interest because they provide absolute heading information and therefore are not subject to drift. However, the magnetic field is generally not homogeneous. For example, the large amount of iron ore on the surface of Mars strongly distorts the magnetic field.

The sun sensors and star sensors use respectively the sun and the stars as absolute references (the design of a sun sensor is presented in [Treb01]). Both provide absolute heading whereas the star sensors also provide the latitude and the longitude. These sensors can only be used when the rover is perfectly still and requires good meteorological conditions. Furthermore, a sun sensor can be utilized only during the day whereas the measurements of a star sensor are only available during the night. However, they are of high interest for global localization. For example, the absolute position acquired by a star sensor during the night can be used to globally relocate the rover after a long traverse.

Three things can be inferred from the above mentioned discussion

- No single sensor is perfect and can provide all the required information. All of them have their own drawbacks and advantages. In general, the quality of the provided information is inversely proportional to the sampling rate e.g. an IMU can provide data at 100Hz but the heading estimation diverges quickly, whereas a sun sensor provides absolute heading but requires the rover to remain at rest and more measurement time.
- Since the data provided by absolute sensors contains no drift, it has more value than the one acquired using dead reckoning sensors.
- Because a small angle error (e.g. heading error) leads to a large position error, it is more important to have precise information about the angles than about the distances.

It is obvious that the use of complementary sensors is required for robust position tracking. In this chapter, we use three different sources of information to test the proposed sensor fusion method, but it can be easily extended to more sources

- *Wheel encoders*: as discussed in chapter 3, odometry can be used to predict the robot displacement and a reasonably accurate motion estimation can be obtained in rough terrain. Moreover, chapter 4 proposes algorithms to limit

wheel slip and enhance the accuracy of the odometry. This motion estimation technique provides data at a relatively high rate (10 Hz) and is not affected by bias errors.

- *IMU*: an Inertial Measurement Unit directly measures the dynamics of the system. When a gravity field is present, the attitude can be estimated without any drift, which is a very valuable information. Furthermore, the measurements are available at a high frequency (75 Hz).
- *Stereovision*: a method similar to [Mallet00] and [Olson01] is used to estimate the six degrees of freedom of the rover. Instead using pixel tracking, interest points are extracted for each frame and are matched using the algorithm presented in [Jung01] and [Jung03]. This technique, which is called Visual Motion Estimation¹ (VME), usually provides better pose estimation than the other sensors but at a much lower rate (0.5 Hz). However, it does not provide any information if features are not properly tracked between successive frames.

For this research, absolute sensors such as GPS have not been considered because the rover is supposed to track its position in an unknown environment without the help of any artificial beacons (e.g. exploration of Mars).

5.3 Uncertainties propagation

A mobile robot is generally equipped with several sensors positioned at different places on the chassis. In order to fuse their measurements, a common reference frame has to be determined and the position of each sensor has to be expressed in this reference coordinate system. The following sections develop the equations for propagating the uncertainties associated to the sensors measurements (incremental motion information) through coordinate system transformations.

5.3.1 Coordinate systems and transformations

The position of a sensor S expressed in the reference coordinate system is determined using an homogeneous transformation matrix (equation 5.1). Such a matrix includes both rotation and translation in three dimensions. The rotation is a three by three direction cosine matrix expressed in terms of the Euler angles² ϕ , θ , ψ and the components of the translation are x_t , y_t and z_t . Thus, such a transformation is parametrized by only six parameters. It is important to note that a homogeneous

1. The author would like to thank S. Lacroix and I.K. Jung for the code and the help related to VME (Appendix D).
 2. The use of the Euler angles can be hazardous because a singularity appears for a pitch of 90° when propagating the angles trough time. Anyway, in our application the pitch is limited ($< 30^\circ$) and will never reach this singular value.

matrix (formed using six parameters) can also be used to describe motion information between two poses or even describe the robot's pose.

$$H = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & x_i \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & y_i \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta & z_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

The different coordinate systems used in the sensor fusion scheme are depicted in Fig. 5.1.

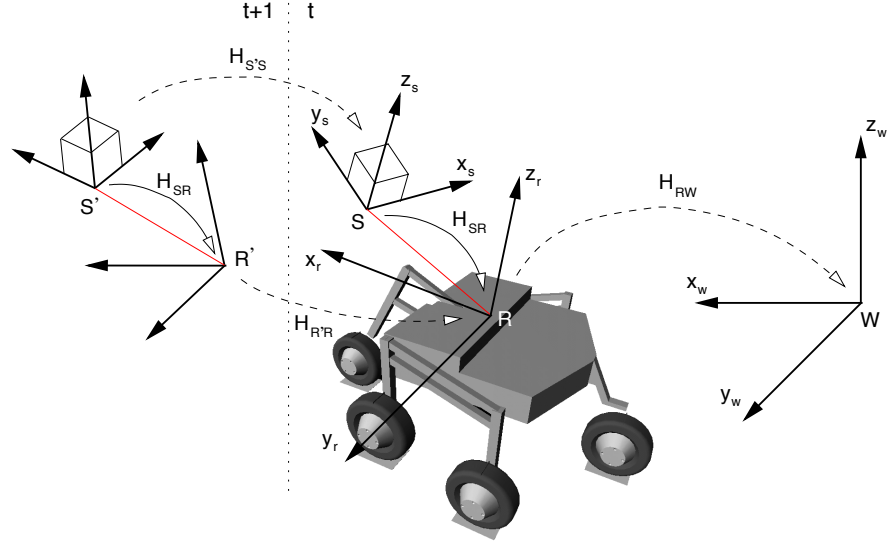


Figure 5.1: Transformations between the different coordinate systems. W , R and S are respectively the centers of the coordinate systems linked to the World, the Robot and a Sensor onboard the rover. The homogeneous matrix H_{ij} allows for the transformation of coordinates expressed in the coordinate system i to that in system j . “prime” signs added to variable names (ex p') denote values related to time step $t+1$.

Between time t and $t+1$, the sensor moves from point S to S' . The 3D transformation reflecting the sensor motion is given by $H_{S'S}$. Thus, the six parameters defining $H_{S'S}$ include all the motion information between t and $t+1$, as perceived by the sensor. Let's define \mathbf{p}_S as being a vector of the parameters defining of $H_{S'S}$

$$\mathbf{p}_S = [\phi_S \quad \theta_S \quad \psi_S \quad x_S \quad y_S \quad z_S]^T \quad (5.2)$$

As said at the beginning of this section, we need to express this transformation in a common coordinate system, which is the coordinate system linked to the robot's body in our case (see Fig. 5.1). In other words, we need to compute the mo-

tion with respect to the robot's center, i.e. $H_{R'R}$. Considering that the position of the sensor relative to the robot's frame remains constant ($H_{SR} = H_{S'R'}$), one can write

$$x_R = H_{SR} \cdot H_{S'S} \cdot x_{S'} = H_{R'R} \cdot H_{SR} \cdot x_{S'} \quad (5.3)$$

where $x_{S'}$ is a point in the sensor frame at time $t+1$ and x_R its coordinates in the robot's frame at time t . Using 5.3, the motion of the robot expressed in the robot's coordinate system is given by

$$H_{R'R} = H_{SR} \cdot H_{S'S} \cdot H_{SR}^{-1} \quad (5.4)$$

and the corresponding vector of parameters is defined as

$$p_R = [\phi_R \quad \theta_R \quad \psi_R \quad x_R \quad y_R \quad z_R]^T \quad (5.5)$$

Using a similar approach, the pose of the robot at time $t+1$, expressed in the world coordinates frame, is given by

$$H_{R'W} = H_{RW} \cdot H_{R'R} \quad (5.6)$$

We define p_W and $p_{W'}$ as being the parameters vectors of H_{RW} and $H_{R'W}$ respectively

$$\begin{aligned} p_W &= [\phi_W \quad \theta_W \quad \psi_W \quad x_W \quad y_W \quad z_W]^T \\ p_{W'} &= [\phi_{W'} \quad \theta_{W'} \quad \psi_{W'} \quad x_{W'} \quad y_{W'} \quad z_{W'}]^T \end{aligned} \quad (5.7)$$

5.3.2 Error propagation

For sensor fusion, we need to assess how the uncertainties associated with the sensor measurement p_S propagate through the coordinate system transformation $H_{R'R}$. The uncertainties associated to the transformation H_{SR} have been neglected because it can be calibrated with high accuracy. In what follows, we define C_S and C_R as the covariance matrices associated to the vectors p_S and p_R respectively. In order to propagate the uncertainties, the function \mathbf{q} (a set of six functions), expressing p_R as a function of p_S has to be derived. One can find \mathbf{q} using 5.4 and the properties of the homogeneous transformation matrices

$$\begin{aligned} q_0(p_S) &= \arctan(H_{R'R}(3,2)/H_{R'R}(3,3)) = \phi_R \\ q_1(p_S) &= \arcsin(-H_{R'R}(3,1)) = \theta_R \\ q_2(p_S) &= \arctan(H_{R'R}(2,1)/H_{R'R}(1,1)) = \psi_R \\ q_3(p_S) &= H_{R'R}(4,1) = x_R \\ q_4(p_S) &= H_{R'R}(4,2) = y_R \\ q_5(p_S) &= H_{R'R}(4,3) = z_R \end{aligned} \quad (5.8)$$

Then the covariance matrix associated to \mathbf{p}_R is given by¹

$$\mathbf{C}_R = \mathbf{J}_S \mathbf{C}_S \mathbf{J}_S^T \quad \text{with} \quad \mathbf{J}_S = \frac{\partial \mathbf{q}}{\partial \mathbf{p}_S} \quad (5.9)$$

Now, we are interested in computing the uncertainty associated to the robot's pose at time $t+1$. For that, we need to combine the uncertainty of the pose at time t and the uncertainty associated to the incremental motion \mathbf{p}_R . The function \mathbf{q}' (a set of six functions), expressing \mathbf{p}_W as a function of \mathbf{p}_W and \mathbf{p}_R is obtained using 5.6 and the properties of the homogeneous transformation matrices

$$\begin{aligned} q'_0(\mathbf{p}_W, \mathbf{p}_R) &= \arctan(H_{R'W}(3,2)/H_{R'W}(3,3)) = \phi_{W'} \\ q'_1(\mathbf{p}_W, \mathbf{p}_R) &= \arcsin(-H_{R'W}(3,1)) = \theta_{W'} \\ q'_2(\mathbf{p}_W, \mathbf{p}_R) &= \arctan(H_{R'W}(2,1)/H_{R'W}(1,1)) = \psi_{W'} \\ q'_3(\mathbf{p}_W, \mathbf{p}_R) &= H_{R'W}(4,1) = x_{W'} \\ q'_4(\mathbf{p}_W, \mathbf{p}_R) &= H_{R'W}(4,2) = y_{W'} \\ q'_5(\mathbf{p}_W, \mathbf{p}_R) &= H_{R'W}(4,3) = z_{W'} \end{aligned} \quad (5.10)$$

Finally, the covariance matrix of the pose at time $t+1$ is given by²

$$\mathbf{C}_W^{t+1} = \mathbf{J}_R \mathbf{C}_R \mathbf{J}_R^T + \mathbf{J}_W \mathbf{C}_W^t \mathbf{J}_W^T \quad (5.11)$$

where \mathbf{C}_W^t is the covariance matrix associated to \mathbf{p}_W and

$$\mathbf{J}_R = \frac{\partial \mathbf{q}'}{\partial \mathbf{p}_R} \quad (5.12) \quad \mathbf{J}_W = \frac{\partial \mathbf{q}'}{\partial \mathbf{p}_W} \quad (5.13)$$

5.4 Sensor fusion

Probabilistic data fusion is the most used method for combining uncertain information. All the probabilistic filters such as the *Hidden Markov Models*, the *Partially Observable Markov Decision Processes* or the *Kalman filter* are derived from the Bayes rule and provide a framework to fuse uncertain data. The choice of one or an other depends on the application. For continuous state spaces, the *Kalman filter* is the preferred choice for sensor fusion. Even if this method can be applied to fuse the measurements acquired by any number of sensors, most of the applications found in the literature generally use only two sensors. The most commonly used pairs are: odometry/laser scanner, odometry/inertial measurement unit

1. An introduction to error propagation is proposed in [Manyika94][Arras98]
2. Although these equations seem to be simple, their implementation generates very complicated expressions.

[Borenstein96], inertial measurement unit/vision [Strelow03][Roumeliotis02] [Vieville93], compass/inertial measurement unit [Roumeliotis99], inertial/GPS [Nebo97] etc. Furthermore, even for rough terrain rovers, **only the 2D case** (x, y, ψ) is generally considered. In this section, a method to estimate the **six degrees of freedom** ($x, y, z, \phi, \theta, \psi$) using the measurements of **three** different sensors is presented.

In our approach an extended information filter (EIF) is used to combine the information coming from the sensors. This formulation of the Kalman filter has very interesting features: its mathematical expression is well suited to implement a distributed sensor fusion scheme and allows for easy extension of the system in order to accommodate any number of sensors, of any kind [Manyika94]. In this application, the observation and transition equations are not linear and a non-linear information filter is required. The observation equation assumes additive zero mean Gaussian noise and is written

$$z_j(k) = \mathbf{h}_j[k, x(k)] + v_j(k) \quad (5.14)$$

where z_j is the measurement vector of sensor j and \mathbf{h}_j the non-linear observation model transforming the state vector $x(k)$ from the state space to the observation space. We define the measurement covariance matrix as being the expectation of the measurement noise: $R_j = E\{v_j v_j^T\}$. Similarly, the non-linear state transition equation can be written as

$$x(k) = \mathbf{f}[k, x(k-1)] + w(k) \quad (5.15)$$

where \mathbf{f} is the non-linear state transition model describing the transition of the state from one time-step to another as a non-linear function of the state. The covariance matrix of the state transition is defined as $Q = E\{w w^T\}$. The first order non-linear information filter is similar to the linear information filter if the following substitutions are made

$$\nabla_x \mathbf{h}_j[k, \hat{x}(k)] \equiv H_j(k) \quad (5.16) \quad \nabla_x \mathbf{f}[k, \hat{x}(k)] \equiv F(k) \quad (5.17)$$

The information filter, the information state vector \mathbf{y} and the information matrix, which is the inverse of the covariance matrix P , are updated according to

$$\hat{\mathbf{y}}(k|k) = \hat{\mathbf{y}}(k|k-1) + \sum_{j \in S} H_j(k)^T R_j^{-1}(k) z_j'(k) = P^{-1}(k|k) \hat{\mathbf{x}}(k|k) \quad (5.18)$$

$$P^{-1}(k|k) = P^{-1}(k|k-1) + \sum_{j \in S} H_j(k)^T R_j^{-1}(k) H_j(k) \quad (5.19)$$

with $S = \{\text{imu, inc, odo, zup, vme}\}$ (see Fig. 5.2) and

$z'(k) = z(k)$ for the linear filter, otherwise

$$z'_j(k) = z_j(k) - (h_j[k, \hat{x}(k|k-1)] - \nabla_x h_j[k, \hat{x}(k|k-1)] \hat{x}(k|k-1)) \quad (5.20)$$

The covariance matrix and the information vector are predicted as

$$P(k|k-1) = F(k)P(k-1|k-1)F(k)^T + Q(k) \quad (5.21)$$

$$\hat{y}(k|k-1) = P^{-1}(k|k-1)F(k)P(k-1|k-1)\hat{y}(k-1|k-1) \quad (5.22)$$

The state vector may be obtained from

$$\hat{x}(k|k) = P(k|k)\hat{y}(k|k) \quad (5.23)$$

Fig. 5.2 depicts the schematics of the sensor fusion process.

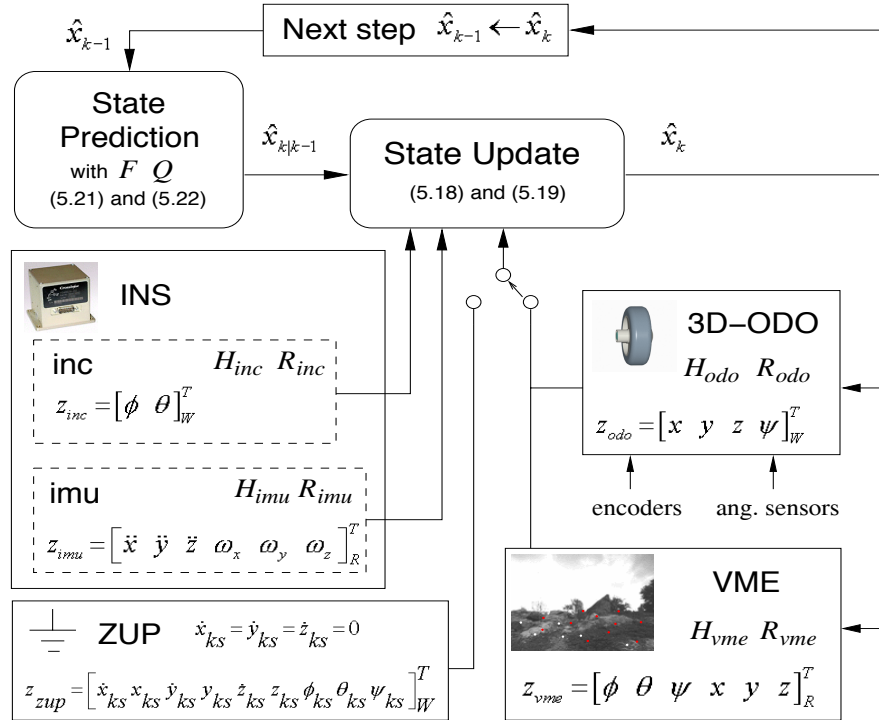


Figure 5.2: The EIF sensor fusion scheme is easily extensible to more sensors. The INS is divided into two sensors: an inclinometer (inc) and an inertial measurement unit (imu) (the unit on SOLERO has a DSP that estimates the attitude and provides both accelerations and angular rates). When the robot is stopped, at time k_s , the ZUP (Zero Update Position) becomes active. This allows to ensure fast convergence of the IMU biases and no drift while the robot is stopped.

It is interesting to note that a sensor j can be easily incorporated in the sensor fusion scheme if the observation model h_j , the error model R_j and the measurement vector z_j are known. With the information filter, the update of the information vector and the information matrix take the form of simple equations (5.18 and 5.19). The update of the information vector can be interpreted that way: the information at time k is equal to the information at time $k-1$ plus the total amount of the information provided by the sensors.

5.4.1 The sensor models

To implement such a filter, the relation between the measurement vectors and the state vector has to be determined for all the sensors. The measurement models h_j together with their linear matrix form H_j are presented in this section whereas the methodology for setting up the corresponding covariance matrices R_j is discussed in the experimental results section. Indeed, the values in R_j are specific to the sensors used in the experiments. The definitions of the measurement vector and matrices for all the sensors are available in Fig. 5.2.

5.4.1.1 Inertial unit model

The position, velocity and attitude can be computed by integrating the readings from the IMU

$$z_{imu} = [\ddot{x} \quad \ddot{y} \quad \ddot{z} \quad \omega_x \quad \omega_y \quad \omega_z]_R^T \quad (5.24)$$

However, both accelerometers and gyros can be influenced by bias errors. Even if these offsets are small they will cause an unbounded growth of the error of integrated measurements. The velocity and the attitude error diverge proportionally over time and the position to the square of time. The measurements of the accelerometers are thus modeled as

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}_R^z = \Gamma_{WR} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}_W + \begin{bmatrix} b_{ax} \\ b_{ay} \\ b_{az} \end{bmatrix}_R + \mathbf{v}_a \quad (5.25)$$

and the gyros as

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}_R^z = \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}_W + \begin{bmatrix} b_{\omega x} \\ b_{\omega y} \end{bmatrix}_R + \mathbf{v}_\omega \quad (5.26)$$

$$\omega_z^z = (1 + \Delta_{\omega z}) [\omega_z + b_{\omega z}] + v_{\omega z} \quad (5.27)$$

Γ_{WR} is the direct cosine rotation matrix that transforms values expressed in the world-fixed coordinates system W into the robot's coordinates system R . This matrix is a function of the angles ϕ (roll), θ (pitch) and ψ (yaw). The b 's and v 's are the biases and the measurements noises of the signals respectively. Unlike roll and pitch, the heading of the rover is not periodically updated by absolute data. Therefore, in order to limit the error propagation, a special provision is included in the z-gyro model: a more accurate modeling, incorporating the scaling error Δ_{ω_z} .

The equations 5.25 and 5.27 are not linear and the first order Taylor expansion is used to provide

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}_R = \bar{\Gamma}_{WR} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}_W + \nabla [\bar{\Gamma}_{WR} \cdot \bar{a}] \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} + \begin{bmatrix} b_{ax} \\ b_{ay} \\ b_{az} \end{bmatrix}_R + \mathbf{v}_a \quad (5.28)$$

$$\text{with } \bar{a} = [\bar{\ddot{x}} \quad \bar{\ddot{y}} \quad \bar{\ddot{z}} - g]^T$$

$$\omega_z^i = (1 + \bar{\Delta}_{\omega_z}) \omega_z + (1 + \bar{\Delta}_{\omega_z}) b_{\omega_z} + (\bar{\omega}_z + \bar{b}_{\omega_z}) \Delta_{\omega_z} + v_{\omega_z} \quad (5.29)$$

where the bars denotes operating point values and g is the gravitational constant, which has to be removed before integrating the accelerations. Then, the matrix H_{imu} can be constructed using 5.26, 5.28 and 5.29 (the detailed linearized equations for the IMU are developed in Appendix B). H_{inc} is the identity matrix because the inclinometer directly measures ϕ and θ (the attitude of the robot).

Since the IMU is not placed exactly at the center of the robot, it is subject to centripetal accelerations due to the angular velocities. These perturbations have to be subtracted from the measurements in order to consider the accelerations related to the center of the robot, which is used as the reference point by all sensors. The centripetal contribution \mathbf{c}_i for each accelerometer is

$$\mathbf{c}_i = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_i) + \dot{\boldsymbol{\omega}} \times \mathbf{r}_i \quad (5.30)$$

$$\text{with } \boldsymbol{\omega} = [\omega_x \quad \omega_y \quad \omega_z]^T \quad \text{and } i \in \{x \ y \ z\}$$

where \mathbf{r}_i is the position of each accelerometer i with respect to the robot's center.

5.4.1.2 3D-Odometry measurement model

The robot used for this research is a partially skid-steered rover and the natural and controlled motion is mainly in the forward direction. Thus, the motion estimation errors due to wheel slip and wheel diameter variations have much more effect in the x-z plane of the rover than along the lateral direction y. Therefore, scaling

errors (Δ_{ox} and Δ_{oz}), modeling wheel slip and wheel diameter change, have been introduced only for the x and z-axes.

3D-Odometry provides an incremental measurement of the rover's motion between time t and $t+1$: $\mathbf{p}_{odo} = [d_{ox} \ d_{oy} \ d_{oz} \ d_{o\psi}]^T$ (expressed in the robot's coordinate system). Thus, the corresponding transformation matrix $H_{R'R}$ (see Fig. 5.1) is obtained by making the following substitution in equation 5.1

$$x_t = (1 + \Delta_{ox})d_{ox} \quad y_t = d_{oy} \quad z_t = (1 + \Delta_{oz})d_{oz} \quad \phi = 0 \quad \theta = 0 \quad \psi = d_{o\psi} \quad (5.31)$$

We set the roll and pitch increments to zero because the information about these angles is not explicitly provided by odometry. As the odometry is updated at a relatively high rate, we can consider the small angles approximation. Thus, setting these angles to zero has minimal impact on the incremental motion error.

The position in the world coordinate system is computed as shown in equation 5.6, using the robot pose at time t and the incremental motion $H_{R'R}$. Finally, 5.10 is used to find the relations between the state vector and the measurement vector. These expressions are not linear and the Jacobian has to be developed to finally obtain H_{odo} .

5.4.1.3 VME measurement model

VME computes the incremental camera motion between two stereo pairs acquisitions i.e. $\mathbf{p}_{vme} = [d_{vx} \ d_{vy} \ d_{vz} \ d_{v\phi} \ d_{v\theta} \ d_{v\psi}]^T$. This transformation, expressed in the camera coordinate system, is first converted into the robot's coordinate system using 5.4. Then the same method as presented in section 5.4.1.2 is applied to derive H_{vme} .

5.4.2 State prediction model

The state prediction model determines the transition of the state vector from one time-step to another. In our case, it has the following form

$$\begin{bmatrix} \mathbf{x}_x \\ \mathbf{x}_y \\ \mathbf{x}_z \\ \mathbf{x}_{ba} \\ \mathbf{x}_\omega \\ \mathbf{x}_{b\omega} \\ \mathbf{x}_\Delta \end{bmatrix}_{k+1} = \begin{bmatrix} F_x & & & & & & \\ & F_y & \dots & & & & \mathbf{0} \\ & & F_z & & & & \\ & \vdots & & F_{ba} & & & \vdots \\ & & & & F_\omega & & \\ \mathbf{0} & & \dots & & F_{b\omega} & & \\ & & & & & F_\Delta & \end{bmatrix} \begin{bmatrix} \mathbf{x}_x \\ \mathbf{x}_y \\ \mathbf{x}_z \\ \mathbf{x}_{ba} \\ \mathbf{x}_\omega \\ \mathbf{x}_{b\omega} \\ \mathbf{x}_\Delta \end{bmatrix}_k + \begin{bmatrix} w_x \\ w_y \\ w_z \\ w_{ba} \\ w_\omega \\ w_{b\omega} \\ w_\Delta \end{bmatrix}_k \quad (5.32)$$

with

$$\begin{aligned} \mathbf{x} &= [\mathbf{X}_x \ \mathbf{X}_y \ \mathbf{X}_z \ \mathbf{X}_{b_a} \ \mathbf{X}_\omega \ \mathbf{X}_{b_\omega} \ \mathbf{X}_\Delta]^T \\ &= [\ddot{x} \ \dot{x} \ x, \ \ddot{y} \ \dot{y} \ y, \ \ddot{z} \ \dot{z} \ z, \ b_{ax} \ b_{ay} \ b_{az}, \ \omega_x \ \phi \ \omega_y \ \theta \ \omega_z \ \psi, \ b_{\omega_x} \ b_{\omega_y} \ b_{\omega_z}, \ \Delta_{\omega_z} \ \Delta_{\omega_x} \ \Delta_{\omega_z}]^T \end{aligned}$$

The angular rates, biases, scaling errors and accelerations are random processes which are affected by the motion commands of the rover, time and other unmodeled parameters. However, they cannot be considered as pure white noise exclusively because they are highly time correlated. In order to illustrate this statement, let us assume that the rover is subject to an acceleration of $2g$ at time t . At time $t+1$, the acceleration cannot reach $-2g$ because the rover has a certain inertia and the elapsed time between t and $t+1$ is small. Thus, the signals are time correlated and cannot be considered as white noise. Instead they can be modeled as first order Gauss-Markov processes¹ whose auto-correlation function is

$$R_p(t) = \sigma^2 e^{-\tau|t|} \quad (5.33)$$

where $1/\tau$ is the time constant defining the correlation time and σ^2 is the variance of the process. Such a process can also be considered as a low pass filter, with τ being the time constant. The discrete differential equations of the first and second integral of such a process are computed using the inverse Laplace operator

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}_{k+1} = \begin{bmatrix} e^{-\tau h} & 0 & 0 \\ (1-e^{-\tau h})/\tau & 1 & 0 \\ (\tau h - 1 + e^{-\tau h})/\tau^2 & h & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}_k = \Phi(\tau, h) \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}_k \quad (5.34)$$

where p_2 and p_3 are respectively the first and second integral of the Gauss-Markov process p_1 and h is the sampling time. It is interesting to note that if τ tends towards zero and if p_1 corresponds to an acceleration then 5.34 becomes

$$\begin{bmatrix} \ddot{x} \\ \dot{x} \\ x \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 \\ h & 1 & 0 \\ h^2/2 & h & 1 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \dot{x} \\ x \end{bmatrix}_k \quad (5.35)$$

This corresponds to the well known set of equations that represents uniformly accelerated motion. Thus, the state propagation along x between k and $k+1$ is nothing more than an accelerated motion using the best estimate of the acceleration at time step k .

1. The detailed derivation of the equations related to the first and second integral of a Gauss-Markov process is available in Appendix C. In particular, the covariance matrix associated to such a process is developed.

The covariance matrix Q_p associated with a Gauss-Markov process and the terms in its integral are derived by computing the individual expectations $E\{p_i p_j\}$ with $i, j = 1, \dots, 3$. Thus, because the accelerations, biases and scaling errors are modeled as Gauss-Markov processes, one can write

$$\begin{aligned}
F_x &= \Phi(\tau_x, h) & F_y &= \Phi(\tau_y, h) & F_z &= \Phi(\tau_z, h) \\
Q_x &= E\{w_x w_x^T\} & Q_y &= E\{w_y w_y^T\} & Q_z &= E\{w_z w_z^T\} \\
F_{ba} &= \text{diag}(e^{-\tau_{bax}}, e^{-\tau_{bay}}, e^{-\tau_{baz}}) & Q_{ba} &= E\{w_{ba} w_{ba}^T\} \\
F_{b\omega} &= \text{diag}(e^{-\tau_{b\omega x}}, e^{-\tau_{b\omega y}}, e^{-\tau_{b\omega z}}) & Q_{b\omega} &= E\{w_{b\omega} w_{b\omega}^T\} \\
F_\Delta &= \text{diag}(e^{-\tau_{\Delta ax}}, e^{-\tau_{\Delta ay}}, e^{-\tau_{\Delta az}}) & Q_\Delta &= E\{w_\Delta w_\Delta^T\}
\end{aligned} \tag{5.36}$$

where $\text{diag}(a, b, c)$ refers to a diagonal matrix composed of the elements a, b and c .

The derivation of F_ω is more tedious because the dynamics of \mathbf{x}_ω is non-linear. Furthermore, the small-angle approximation cannot be made because the robot moves on rough terrain, where angular variations can be of high amplitude. Equation 5.37 describes the non-linear state transition of \mathbf{x}_ω

$$\begin{aligned}
\omega_{x_{k+1}} &= \omega_{x_k} e^{-\tau_{\omega x} h}, & \omega_{y_{k+1}} &= \omega_{y_k} e^{-\tau_{\omega y} h}, & \omega_{z_{k+1}} &= \omega_{z_k} e^{-\tau_{\omega z} h} \\
\phi_{k+1} &= q_1(\mathbf{x}_\omega) = \phi_k + h \left((\omega_y \sin(\phi) + \omega_z \cos(\phi)) \tan(\theta) + \omega_x \right) \\
\theta_{k+1} &= q_2(\mathbf{x}_\omega) = \theta_k + h \left(\omega_y \cos(\phi) - \omega_z \sin(\phi) \right) \\
\psi_{k+1} &= q_3(\mathbf{x}_\omega) = \psi_k + h \left(\omega_y \sin(\phi) + \omega_z \cos(\phi) \right) / \cos(\theta)
\end{aligned} \tag{5.37}$$

Finally, the linearized 6x6 matrix F_ω is obtained by computing the Jacobian of the \mathbf{q} functions (see B.2).

5.5 Experimental results

The aim of this section is to describe the methodology to define the state transition and measurements covariance matrices (Q and R_i) and to validate the theory through experiments conducted on SOLERO. In order to better illustrate how each sensor contributes to the pose estimation and in which situation, the experiments have been divided into two parts. The first part describes the results of sensor fusion between Inertial sensor and Odometry only, whereas the second part involves all the three sensors i.e. Odometry, Inertial sensor and VME (Visual Motion Estimation).

5.5.1 Inertial and 3D-Odometry

The Inertial Navigation Systems (INS) provides direct measurements of the dynamics of a system and is self-contained. For these reasons, it is used in many applications to predict the robot's position and orientation. INS's were first used in aerospace applications and a large part of the literature uses them in this context (see [Titterton97] for theory and application of INS). The availability of integrated low-cost and low-power solid-state sensors enabled the usage of the INS for ground applications such as mobile robots. Nevertheless, these sensors provide less accurate position information and their implementation on ground vehicles is more difficult than on aircrafts. Indeed, trajectories are less smooth on the ground where the system is subjected to strong vibrations.

Many research works are related to road vehicles applications, in which an INS is used to provide higher update rate of the position between two consecutive GPS data acquisition. In particular, this combination of sensors is used to estimate the wheel diameter changes and the vehicle sideslip in [Wada00][Bevly01]. [Barshan95] showed that a low-cost INS can improve the system performance and can be applied to mobile robotics if an accurate sensor model is provided. A method for combining data from gyroscopes and odometry is presented in [Borenstein96]. The authors of [Scheding99] present interesting results for an underground mining vehicle. They show clearly how inertial sensors can be used to correct non-systematic errors due to soil irregularities when fused with other sensors such as wheel encoders and laser scanners. In [Dissanayake01], the authors propose to use the nonholonomic constraints that govern the motion of a vehicle on a surface, to align the INS. 3D-simulation results of a sensor fusion between an INS and a compass are presented in [Roumeliotis99]. The paper states that the system can be used to localize a Mars rover prototype. Unfortunately, the position error grows quickly when localization is done on the sole basis of acceleration integration. Furthermore, a compass cannot be used on Mars because of the high density of iron.

Most of the published works involving INS on ground vehicles present results in two dimensions and deal with the estimation of the planar position and orientation only. Furthermore, the target environment is generally flat and the structure of the soil can be known beforehand. This allows developing relatively accurate vehicle models, which lead to good odometric information. The situation in rough terrain is more challenging and these assumptions are not applicable. In particular, no accurate wheel-ground interaction model can be developed and the planar assumption cannot be considered. In this section, the experimental results show how INS and 3D-Odometry can be combined to provide better position estimates in three dimensions.

5.5.1.1 Setting the state transition covariance matrix Q

Most of the parameters in the state transition covariance matrix Q are estimated based on the datasheet values and experimental data. The parameters associated with Q_x , Q_y , and Q_z are based on assumptions on how the vehicle is driven and on the general terrain type: the robot has ground contact at any time, it is non-holonomic and the roll and pitch angles are limited to relatively small values ($< 30^\circ$). These constraints limit the robot motion to remain in the 2.5D space. In other words, the z coordinate of the robot is a function of x and y. These considerations are used to adjust the values in the different covariance matrices. The noise sequences of \mathbf{x}_x , \mathbf{x}_y , and \mathbf{x}_z are dependant from each other. Indeed, when the rover is accelerating in the x-y plane both accelerations along x and y are affected. Modeling of this cross-correlation is highly complex because it is a function of nonlinear transformations, which are in turn functions of time. However, in order to avoid excessive complexity and benefit from a simpler filter, we have assumed no cross-correlation between these axes.

Some simple heuristics can be applied for estimating how certain parameters are related to each other and how they are expected to behave as a function of time. For example, the bias affecting an accelerometer changes slower than the acceleration itself. Finally, taking some margin on the variances of the processes allows accounting for a larger range of situations, avoiding the filter to diverge. Table 5.1 lists all the parameters together with the heuristics that have been used in each case. They might not be the optimal parameters but they have proven to give good filter performance.

Table 5.1: State transition parameters

$\tau_{\ddot{x}} = \tau_{\ddot{y}} = 0.6, \tau_{\ddot{z}} = 0.1$	The experiments show that the z-axis is more subject to vibration when the rover is driving. Thus, it has to be filtered to a greater extent as compared to x and y.
$\sigma_{\ddot{x}}^2 = \sigma_{\ddot{y}}^2 = 0.008, \sigma_{\ddot{z}}^2 = 0.003$	The acceleration along the z axis is generally smaller than the acceleration along x and y axes because the motors of the rover directly affect the acceleration in x and y. The acceleration along z is only due to slope changes in the terrain.
$\tau_{b_{ax}} = \tau_{b_{ay}} = 0.016, \tau_{b_{az}} = 0.002$	The biases change slower than the accelerations, over time. Thus, their time constants are set shorter.
$\sigma_{b_{ax}}^2 = \sigma_{b_{ay}}^2 = 0.2, \sigma_{b_{az}}^2 = 0.11$	These values are set being equal to the square of the half of the maximum biases of the accelerometers (2σ), the values of which are given in the INS datasheet.
$\tau_{\omega_x} = \tau_{\omega_y} = 1, \tau_{\omega_z} = 3$	ω_z is directly governed by the command signals to the rover. It is thus subject to change more rapidly than $\omega_{x,y}$.

$\sigma_{box}^2 = \sigma_{boy}^2 = 0.0006$ $\sigma_{boz}^2 = 0.012$	These values are set being equal to the square of half the maximum biases for the gyros (2σ), the values of which can be found in the INS datasheet.
$\tau_{\Delta w_z} = 3 \cdot 10^{-4}, \sigma_{\Delta w_z}^2 = 3 \cdot 10^{-5}$	The same reasoning used for setting the acceleration biases is applicable here. According to the INS datasheet, the scaling factor is less than 1%. So, we took the square of half of this value to set the variance for the scaling factor.
$\tau_{\Delta ox} = \tau_{\Delta oz} = 2$ $\sigma_{\Delta ox}^2 = \sigma_{\Delta oz}^2 = 1$	These values have been determined experimentally. However, the filter is not very sensitive to their variation.

5.5.1.2 Setting R_{imu} , R_{inc} and R_{odo}

In order to set the variances for the IMU, the rover has been driven forward at different velocities and on different types of soil while collecting data and computing statistics. The experiments showed that the variance of the signals doesn't change significantly with change in velocity or terrain type. This can be attributed to the passive mechanical structure of SOLERO, which allows for filtering and smoothing of the trajectories. Thus, the worst-case set of variances has been selected to set the matrix R_{imu} . For the inclinometer, the variances of the roll and pitch angles have been set to the square of half the value given in the INS datasheet ($2\sigma = 1^\circ$). These values correspond to the diagonal elements of R_{inc} .

The sensor model for the odometry is much more tedious to assess because the robot is subject to drive across all kind of terrain and soil such as sand, rock and grass. It is very difficult to classify all types of terrains and configurations and to associate the corresponding variances. Instead, we set the uncertainty of the odometric information as being proportional to the acceleration undergone by the rover (see 5.38). Indeed, slip mostly occurs in rough terrain, when negotiating an obstacle, while the robot is subject to accelerations. Furthermore, at constant speed, the acceleration is zero and thus acceleration does not bring much information. In this particular case, position estimation can rely only on odometry. For the same reasons, the variance for the yaw angle has been set proportional to the angular rate (ω_z). Thus, the covariance matrix associated to the 3D-Odometry is written as

$$C_R = \begin{bmatrix} k_x(1 + \ddot{x}_R^z - g_x) & \dots & 0 \\ \vdots & k_y(1 + \ddot{y}_R^z - g_y) & \vdots \\ 0 & \dots & k_z(1 + \ddot{z}_R^z - g_z) \\ & & & k_\psi(1 + \omega_z^z) \end{bmatrix} \quad (5.38)$$

5.5 EXPERIMENTAL RESULTS

where k_x , k_y , k_z and k_ψ are constants set empirically and g_x , g_y and g_z are the gravitational components in the rover-fixed frame. The constants k_x and k_z have been set larger than k_y because SOLERO is a skid-steered rover: the motion commands affect the x-z position, y is not directly controllable. This set of constants has been tested and validated during the experiments performed on different types of terrains and obstacles. Finally, the same equations presented in section 5.3.2 are applied to propagate the covariance matrix C_R through the coordinate system transformation and obtain R_{odo} .

5.5.1.3 Experimental validation

In order to test the sensor fusion algorithm, the robot has been driven forward across different experimental setups during a fixed interval of time. Then, the pure 3D-odometry and filtered trajectories have been compared. By filtered trajectory, we mean the trajectory built out of the position estimates computed by the EIF filter. We have repeated the same experiment several times and measured the final position of the robot in each run. Fig. 5.3 depicts the most difficult obstacle configuration the rover has to climb during the experiments.

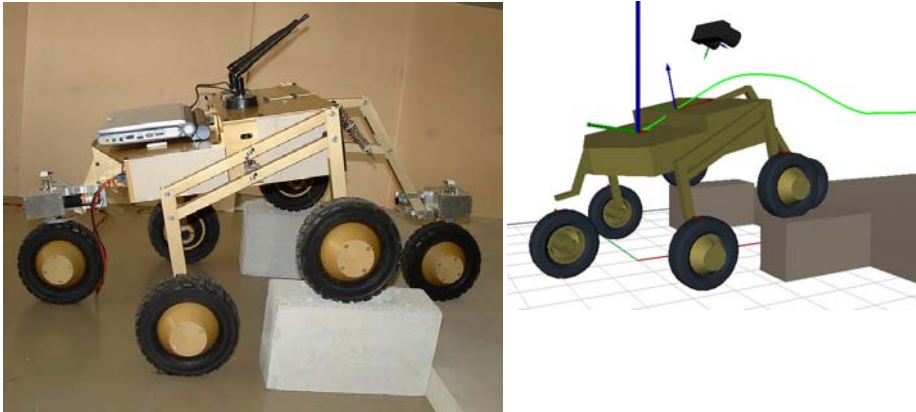


Figure 5.3: *Picture of one of the experimental setups along with the corresponding 3D model used to analyse the results. Because the dimensions of the obstacles are known, we can measure precisely the true maximal and final position heights. In this case the maximal height is 135mm and the final height is 45mm. This kind of obstacle is very difficult to negotiate for a wheeled rover because of the sharp edges and the low friction coefficient.*

Due to the low friction coefficient between the wheels and the obstacle, a lot of slip occurs during the step climbing. Furthermore, the robot literally bounces on the ground when the rear bogie wheels go down from the obstacle. The shock occurring during the experiment are easily identified when looking at the z-accelerometer plot in Fig. 5.4.

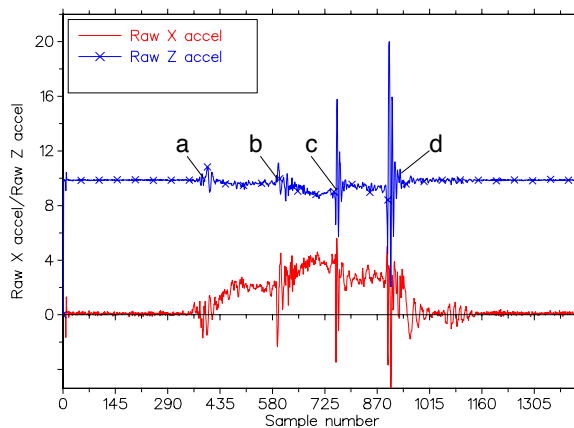


Figure 5.4: Raw x and z -accelerometer signals (in the robot's coordinate system). The amplitude of the accelerations can reach values higher than $1g$ (a) front bogie wheels climbing on the obstacle (b) rear bogie wheels climbing (c) front bogie wheels going down the obstacle (d) rear bogie wheels going down.

Table 5.2 reports the final measurements together with the final position errors. The third run, highlighted in the table, is used as the reference experiment for the next two figures.

Table 5.2: Experimental measurements

Experimental values (mm)								
Measured			3D-Odometry only			Filtered		
x	y	z	x	y	z	x	y	z
1020	4	45	1150	88	40	1160	17	44
1025	7	45	1149	66	40	1152	38	40
1030	5	45	1182	58	38	1184	18	45
1030	2	45	1149	31	33	1150	29	34
1025	1	45	1152	35	36	1152	16	37
Average error			130	52	8	131	20	5

The error along the x -axis is the same for both the 3D-odometry and the filtered trajectories. This result can be explained: it is because wheel slip mainly occurs when the robot starts climbing the obstacle at constant speed, while the trajectory is smooth. During this phase, the accelerometers don't detect velocity change and therefore cannot help in correcting the position (see Fig. 5.4a). On the other hand, when the rover goes down the obstacle (Fig. 5.4c and d), the variance of the odometry increases (5.38) and the z -accelerometer information allows for correction of the trajectory. Thus, the error along the z -axis is only 5 mm instead of 8 mm. Fig.

5.5 depicts this correction nicely. For all the experiments the filtered final z-coordinate is always closer to the true height of 45mm.

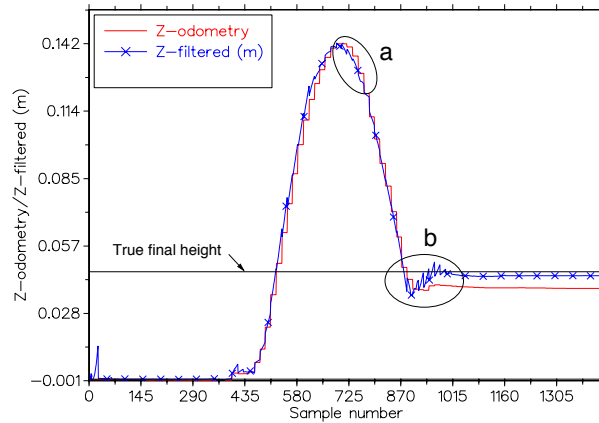


Figure 5.5: The z trajectories for the third run (see Table 5.2). The ellipses (a) and (b) show the correction occurring when the front, respectively rear, bogie wheels go down from the step. These corrections correspond to the zones c and d of Fig. 5.4.

The error in the y -direction is mostly due to the heading (yaw) error occurring during asymmetric wheels slip. The odometry is very sensitive to this effect and the yaw estimation can vary significantly even for limited slip. Fig. 5.6 shows how the yaw gyro helps correcting the heading. The result is a noticeable diminution of the error along the y -axis (see Table 5.2).

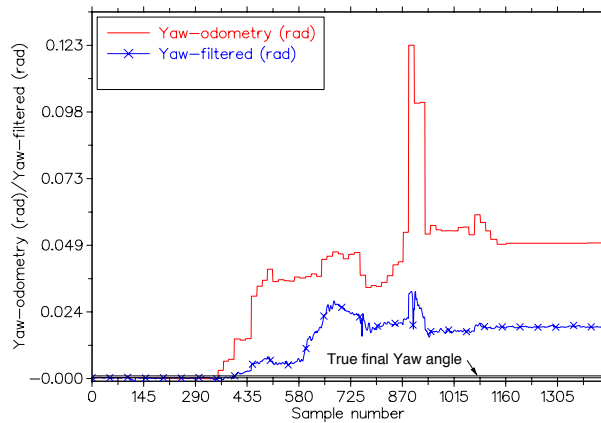


Figure 5.6: The yaw angles estimates for the third run (the true final angle is close to zero). The yaw gyroscope (measuring the angular rate around z) allows to correct for asymmetric slip.

The errors along the x-axis being the same, it is interesting to consider the absolute errors in the y-z plane. Fig. 5.7 shows that the final positions, computed with the sensor fusion algorithm, are systematically closer to the true position than the pure 3D-odometry estimations.

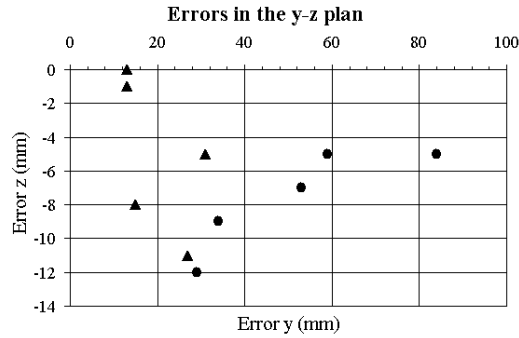


Figure 5.7: Errors in the y-z plane. The triangles represent the filtered values and the circles, the pure 3D-odometry estimations. The total travelled distance along x is one meter.

For testing the system in a more general case, the rover has been driven twenty times across the scene depicted in Fig. 5.8. Each time, the operator remote controlled the rover in order to close the loop. For each run, the final error of the filtered trajectory was smaller than that given by pure-odometry. An example is depicted in Fig. 5.8

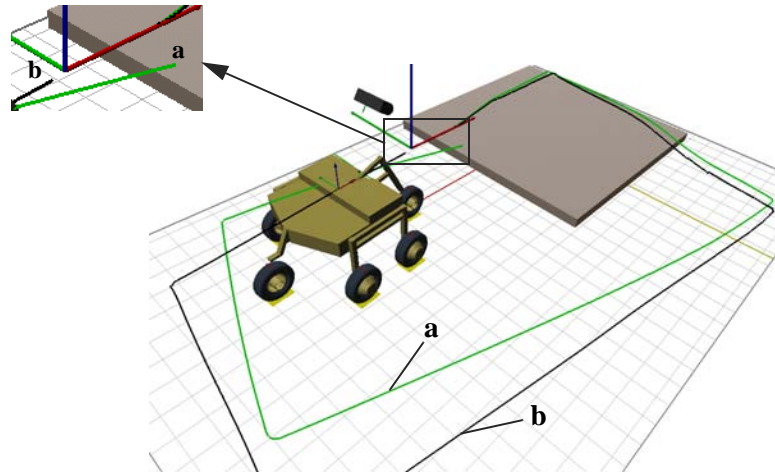


Figure 5.8: Comparison between (a) pure odometry and (b) filtered trajectory. The final error $[\epsilon_x, \epsilon_y, \epsilon_z, \epsilon_\psi]$ is respectively $[0.16, 0.142, 0.014, 18^\circ]$ and $[0.06, 0.029, 0.012, 1.2^\circ]$ for this

5.5.1.4 Discussion

The experimental results show that the inertial navigation system helps to correct odometric errors and significantly improves the pose estimate. The main contributions occur locally when the robot overcomes sharp-shaped obstacles and during asymmetric wheel slip. In all the experiments, the fusion between odometry and inertial sensor provided better motion estimates than with odometry alone. The improvement brought by the sensor fusion process becomes more and more pronounced as the total path length increases.

In comparison with other research projects which integrate inertial sensors to mobile robots, this work¹ has the following interesting features

- an error model for the 3D-Odometry is established based on the INS measurements: accelerations and angular rates.
- the INS is used in rough terrain, where the ratio signal/noise is low.
- it has been shown that the integration of the accelerations can be used to locally correct the robot's position.
- this work addresses the full 3D case.

5.5.2 Enhancement with VME

In the previous section, only proprioceptive sensors have been integrated to estimate the robot's position. Even if the inertial sensor helps to correct odometric error, there are situations where this combination of sensors does not provide enough information. For example, the situation where all the wheels are slipping is not detected by the system. In this case, only the odometric information is integrated to produce erroneous position estimates. Thus, in order to increase the robustness of the localization and to limit the error growth, it is necessary to incorporate exteroceptive sensors. These sensors allow to compute ego-motion by tracking characteristic features in the environment and thus complete the missing information. In this section, experimental results integrating an Inertial Sensor, 3D-Odometry and Visual Motion Estimation are presented.

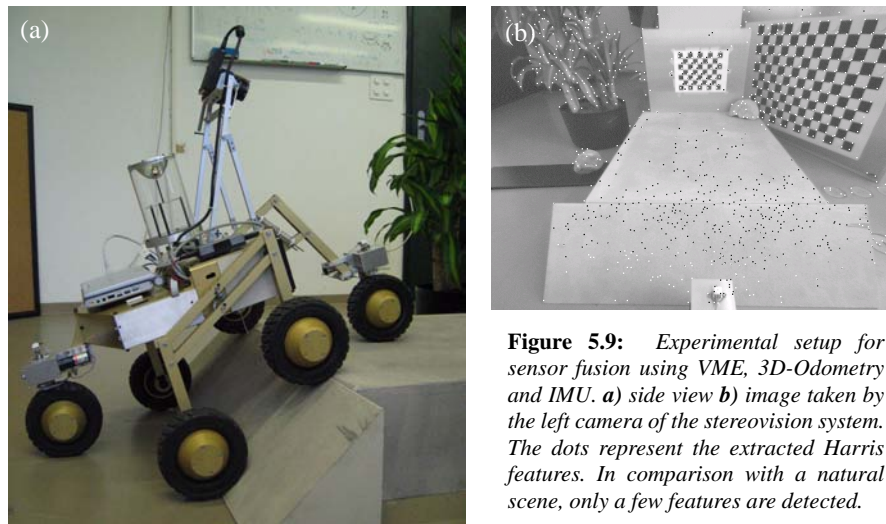
The matrices R_{imv} , R_{inc} and R_{odo} have already been determined in the previous experiments. Here, only R_{vme} remains to be defined. The uncertainty model of VME is based on an error model of stereovision, which uses the assumption that there is a strong correlation between the shape of the similarity score curve around its peak and the standard deviation of the disparity. More details about the error model of VME are available in

1. This work has been published at IROS '04 (see [Lamon04b])

[Jung03]. Finally, the equations presented in section 5.3 are used to propagate the covariance matrix of VME, expressed in the camera frame, in the robot's frame.

5.5.2.1 Experimental results

The setup used to test the sensor fusion is depicted in Fig. 5.9. The use of obstacles of known shape enables to both the pre-calculation of a reference trajectory (ground truth) and to the determination of the exact final height of the rover.



The experiment consisted in driving the rover on top of the obstacle and run the sensor fusion algorithm to compute the rover's trajectory. A sequence of snapshots taken during the experiment is shown in Fig. 5.10.

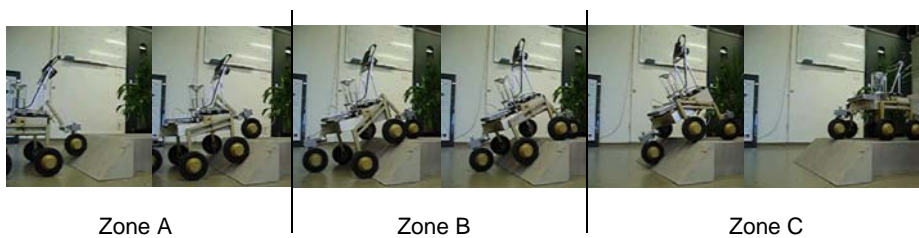


Figure 5.10: Trajectory of SOLERO (decomposed in three zones)

The graph in Fig. 5.11 plots four trajectories i.e. the pure 3D-Odometry, the VME, the Estimated trajectory and the Reference trajectory. The Estimated trajectory is the result of the sensor fusion of all the three sensors. The Reference trajec-

tory has been computed considering the kinematics of SOLERO and the known shape of the obstacle (the final x coordinate has been measured at the end of the experiment). As the rover did not deviate significantly from straight motion, this trajectory is considered as the ground truth.

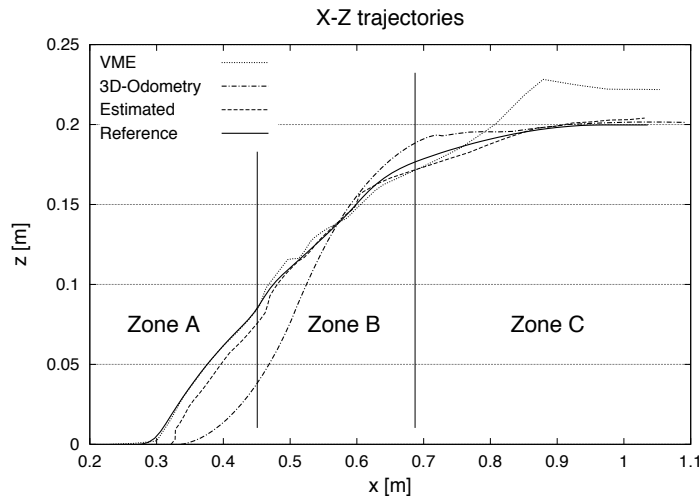


Figure 5.11: X-Z trajectories for the sensor fusion experiment. In zone A, the VME trajectory is very close to the reference trajectory and the odometry provides very divergent information. As a consequence the estimated trajectory is not perfectly aligned on the reference trajectory.

The graph is divided into three zones, characterizing different situations (see Fig. 5.10) and filter behaviors. In zone A, the VME trajectory is almost exactly superposed on the Reference trajectory whereas the Estimated trajectory is a slightly offset. This is mainly due to the fact that odometry provides very divergent position estimates in that situation, which is in turn due to wheel slip. Furthermore, because the trajectory of the robot is smooth in that part of the path, the IMU did not detect significant acceleration along z and thus could not bring valuable information. Even if the variance in the odometry is smaller than that of the VME it nevertheless contributed towards moving the Estimated trajectory away from the Reference.

In zone B, the Estimated trajectory is closer to the reference trajectory than VME (see Fig. 5.12). In this part of the experiment, the VME started to produce less accurate motion estimations due to a lower number of feature matches between successive frames. As a consequence, the VME trajectory comprises of steps and hence the uncertainty associated with the estimations increases. This explains why these steps are filtered and almost invisible in the Estimated trajectory.

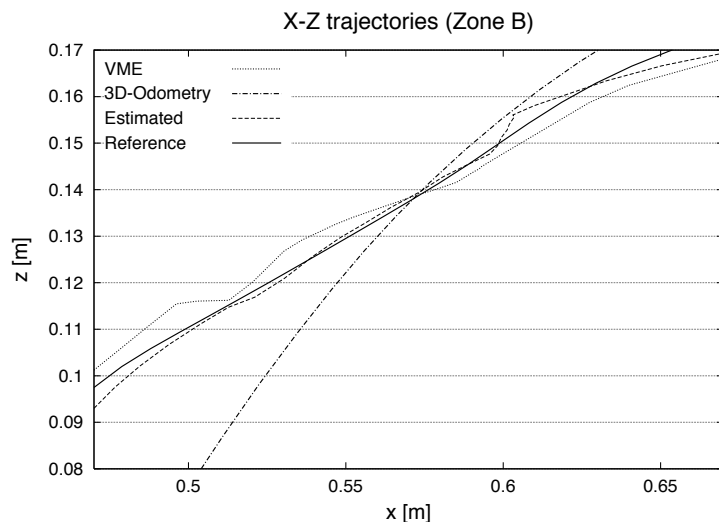


Figure 5.12: Enlarged view of zone B (see Fig. 5.11). On this graph the Estimated trajectory is very close to the Reference trajectory whereas VME comprises an offset.

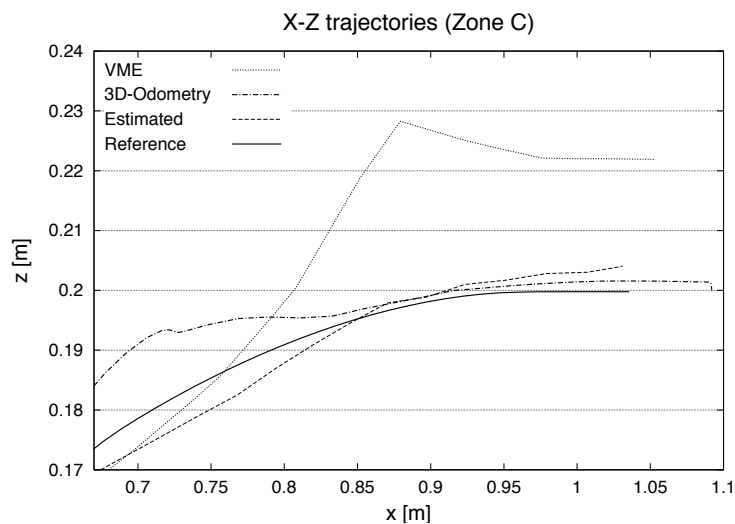


Figure 5.13: Enlarged view of zone C. Less than 30 features have been matched between image 31 and 32. This leads to a bad estimate of the VME (associated to a large uncertainty). Thanks to the sensor fusion, the system successfully filtered this information to provide good position estimation.

5.5 EXPERIMENTAL RESULTS

In zone C (Fig. 5.13), less than thirty features have been matched between image number 31 and 32. The difficulty to find matches between these two images is due to a high discrepancy between the views: when the rear wheel finally climbs the obstacle, it causes the rover to tilt rapidly. As a consequence, the VME provided a very bad motion estimate with a high uncertainty (see Fig. 5.13 and Fig. 5.14). In this situation, less weight is given to VME and the sensor fusion could perfectly filter this bad information to produce a reasonably good estimate using the odometry and the IMU instead. Finally, the estimated final position is very close to the measured final position. A final error of four millimeters for a trajectory longer than one meter (0.4%) is very satisfactory, given the difficulty of the terrain.

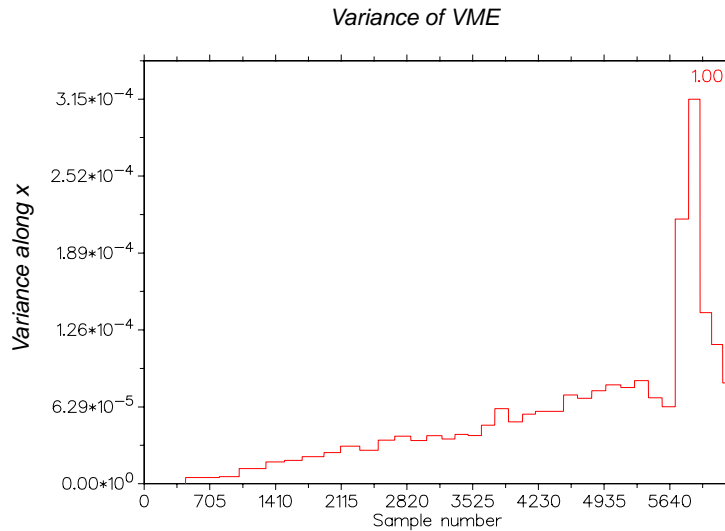


Figure 5.14: Variance associated to the VME estimations along the x axis. The uncertainty increases suddenly at image 32 because only a few features have been matched.

It is interesting to plot the variance of the position estimation along the x direction as a function of the time. As shown in Fig. 5.15, the variance globally increases, as a function of time. This is because no absolute information is available to reset the position in a global reference frame. The “saw” shape of the function at the global level is due to the VME updates whereas the “saw” shape at a local level is due to the odometry updating the position estimated on the sole base of the inertial measurement unit. In other words, the estimations of VME have the biggest weight, followed respectively by the 3D-Odometry and the IMU. In Fig. 5.15, we can also observe the effects of the updates when uncertain VME estimations are provided. When the uncertainties of VME are large (see Fig. 5.14), the estimations

have less weight and the variance along x remains high. Such a behavior, is expected and prove that the filter provides consistent estimates.

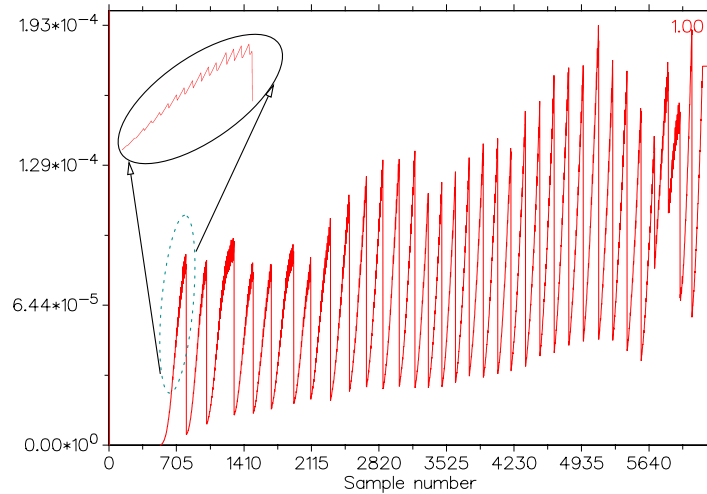


Figure 5.15: *The variance of the estimate along the x axis. Because no absolute sensor is used to reset the position, the variance globally increases over time. The variance significantly decreases each time a VME estimate is available. At a lower level, the odometry (see the enlarged view) periodically resets the inertial estimation and this corrects the biases.*

5.6 Conclusion

In this chapter, a method for combining different sensors to produce a robust motion estimate has been presented. The sensor fusion scheme is flexible and can easily accommodate any number of sensors of any type. To test the method, different experiments have been performed and proved that the use of complementary sensors increases the robustness and accuracy of the results emphatically.

This work distinguishes itself from other similar research projects in the following respects:

- The sensor fusion is performed with more than two sensor types (usually only two sensors are used).
- Sensor fusion is applied in a rough terrain to track the 3D pose of the rover. Most of other research works assume flat environments and only track the position in 2D.

5.6 CONCLUSION

6

Conclusion and outlook

6.1 Conclusions

The challenge of realizing autonomous all terrain rovers warrants the development of systems able to deal with a lack of a priori information, the problems of perception and locomotion. One of the main difficulties encountered, in comparison with 2D indoor robotics, is that it is more difficult to decouple the various functionalities involved. For example, a trajectory planner cannot be easily ported from one platform to another because it has to account for the specific kinematic structure and climbing capabilities of the rover. Another example is related to the sensors: it has been possible to use inertial information on SOLERO because it is a passive architecture which intrinsically limits the ratio noise/signal. Extracting motion information from inertial sensors is very difficult if they are mounted on a fully active structure or a four-wheeled rover, for example. These two examples show that the methods cannot be generalized and applied to any kind of robotic structure. Finally, we can attempt to formulate the following rule: the more challenging the terrain, the more specific the solutions.

The intent of this thesis is to extend the range of possible areas a robot can explore. The contributions focus mainly on locomotion and localization.

- In chapter 2, the design of a performant platform for research has been presented. In particular, the rover is equipped with two computers, a stereovision module, an omnidirectional vision system, an inertial measurement unit, numerous sensors and actuators and electronics for power management. Furthermore, a set of powerful tools has been developed to speed up the process of debugging the algorithms and analyzing the data stored during the experiments. Finally, the modularity and portability of the system allows easy adaptation of new actuators and sensors.
- In chapter 3, 3D-Odometry has been developed for SOLERO. Because it accounts for the kinematics of the rover, it provides better odometric estimates than a simpler method accounting only for the attitude of the main body (roll and pitch). An interesting feature of the method is that it internally

computes the contact angles between the wheels and the ground, which are required inputs for the proposed wheel controller.

- In chapter 4, a quasi-static model of a six-wheeled rover together with an optimization method for minimizing slip have been presented. Unlike other control strategies, the proposed method is independent of soil models, whose parameters are unknown in real applications. Indeed, the rover drives on different types of soils during exploration missions. Furthermore, the approach can be adapted to any kind of passive wheeled rover and the optimization can be run online.
- In chapter 5, a sensor fusion scheme, extensible to any types/number of sensors has been developed. Experiments involving inertial, 3D-Odometry and visual odometry have been performed. It has been shown that the use of complementary sensors improves the accuracy and the robustness of the motion estimation. In particular, the system was able to properly discard inaccurate and uncertain visual motion information.

Technically, doing research in this field is difficult because there is almost no all terrain platform available in the market. Most of the rovers are prototypes and require specific tools and developments. There are a lot of constraints preventing the use of standard technology and special care must be taken when choosing a specific sensor or actuator. In particular, a lot of efforts is requested to keep the weight and the energy consumption as low as possible. Thus, in comparison with other fields of research, a large part of this work is devoted to engineering and implementation. However, as mentioned before, understanding the specific kinematics and the physics of the structure is important to develop appropriate algorithms, which can actually be implemented on a real rover. Being aware of these constraints promotes bottom up solutions instead of top down approaches, often leading to solutions that cannot be implemented because they use unavailable information about the environment or require too much processing power etc.

6.2 Outlook

Even if the experiments provided promising results, there are still some aspects that can be improved to provide better performance

- For sensor fusion, the uncertainty in the odometry has been set proportional to the accelerations. However, this simple model can be improved by also accounting for the kind of wheel-ground interaction. Thanks to the quasi-static model of SOLERO, the traction and normal forces can be computed and used to set a slip probability for each wheel: the less the pressure on the wheel, the more likely it slips (an alternative to this method is to measure

the torque of the wheels). These uncertainties can then be propagated using the kinematic model of the rover to produce the global motion uncertainty for the odometry.

- The state transition model for the sensor fusion can be refined. Until now, it does not account for the inputs of the system, which are the torques/speeds of the wheels. Again, accounting for the kinematics of the rover, the inputs can be used to better predict the next state.
- The accuracy of the position estimates can be improved by integrating more sensors to the system. Because the position errors (x,y,z) are very sensitive to a heading error it is important to give special care to the estimation of this angle. For this purpose, an omniscam is an interesting option because it offers a panoramic view that allows for tracking features all around the rover at the same time, without suffering from the problem of the lateral image borders. Ego-motion algorithms can be applied to the raw images to estimate the six degrees of freedom. However, this information is difficult to integrate to the sensor fusion scheme because it is difficult to establish an uncertainty model and the translation are given scaleless. This can be solved by introducing a scale factor in the sensor's observation model: the scale factor is introduced in the state vector and is estimated using all the other sensors e.g. odometry, inertial and VME.
- In this thesis, we have proposed a set of tools and algorithms to improve the accuracy of the position estimation and limit the error growth as much as possible. Periodically, the position of the rover has to be reset using absolute sensors such as star-sensor, sun sensor etc. In theory, we can avoid using these absolute sensors by doing SLAM (Simultaneous Localization and Mapping). Basically, this method consists in simultaneously estimating the rover position while creating a map, composed of relevant features of the environment. By constantly re-observing and matching the same features it is then possible to bound the position error. However, in practice, the method is very difficult to apply in rough terrain. The main constraints include:
 - a. Due to the problem of perception in rough terrain, it is difficult to constantly re-observe and match the same features as the robot moves. The main difficulties are related to occlusions and potentially important field of view change between two data acquisition steps. Furthermore, even if the rover is placed at the same position, the view can be very different depending on the orientation (even with an omniscam). The perception of the environment can be extremely different when going from A to B or B to A, this makes the problem of feature matching extremely difficult. Finally, most of the time, the rover never comes back at the same place when exploring

6.2 OUTLOOK

an area (e.g. the MER's) and thus, SLAM does not provide a bounded positioning error.

- b. SLAM becomes computationally expensive as the number of landmarks increases. When exploring a large area a lot of landmarks have to be stored and this problem appears quickly.

In spite of all these limits, SLAM can nevertheless be used locally to refine the motion estimates. Indeed, even if the features are re-observed only a few times and discarded when they disappear from the field of view of the robot, these multiple observations help to limit the error growth in the position estimate.

- In hazardous terrains, the rover has to negotiate obstacles instead of avoiding them. As a consequence of this, the task of planning a trajectory in 3D through the scene and that of controlling the rover's motion become highly complex. For trajectory planning, a Digital Terrain Model (DTM) is required to select an optimal path considering the kinematic model of the rover. In challenging environments, the 3D information about the terrain in front of the rover is sparse because of the shadow effect (occlusions caused due to the presence of obstacles). Trajectory planning with incomplete information is difficult. Once a path is selected, the system has to generate motor commands in order to properly execute the trajectory, this is tedious because of the wheel-slip phenomenon and the inability to have complete information about the terrain characteristics. Regardless of the problems mentioned above, a controller minimizing wheel slip and robust 3D position tracking are required functionalities for trajectory execution. This thesis contributes towards both these critical aspects of autonomous rover navigation.

A

Parameters and model of SOLERO

This appendix defines all the constants and variables of the mechanical structure of SOLERO and includes the complete mathematical expression of the quasi-static model used for a wheel motor control minimizing slip. The units of all the parameters follow the SI convention (also known as MKSA convention). This formalism is very useful because it allows different people to communicate and easily integrate their work. In order to increase lisibility, all the internal forces of the joints have been omitted in the figures. Only the relevant forces and torques are depicted.

For the coordinate systems, r refers to the robot and W to the global frame (world).

A.1 Parts of SOLERO

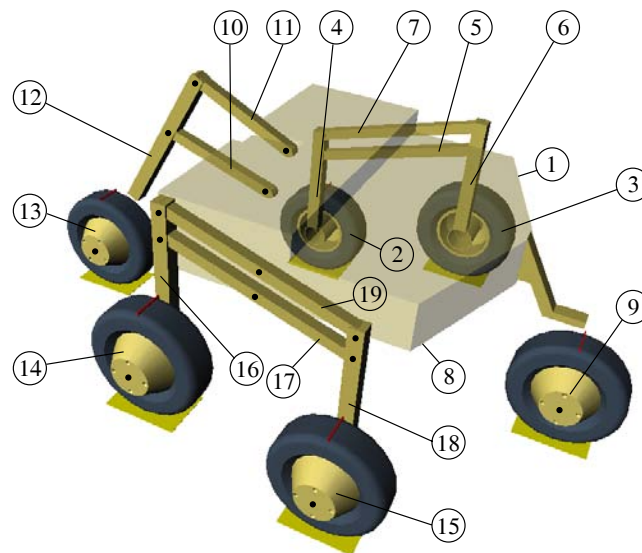


Figure A.1: *Parts numbering for SOLERO (part 1 is the main body)*

A.2 The bogies

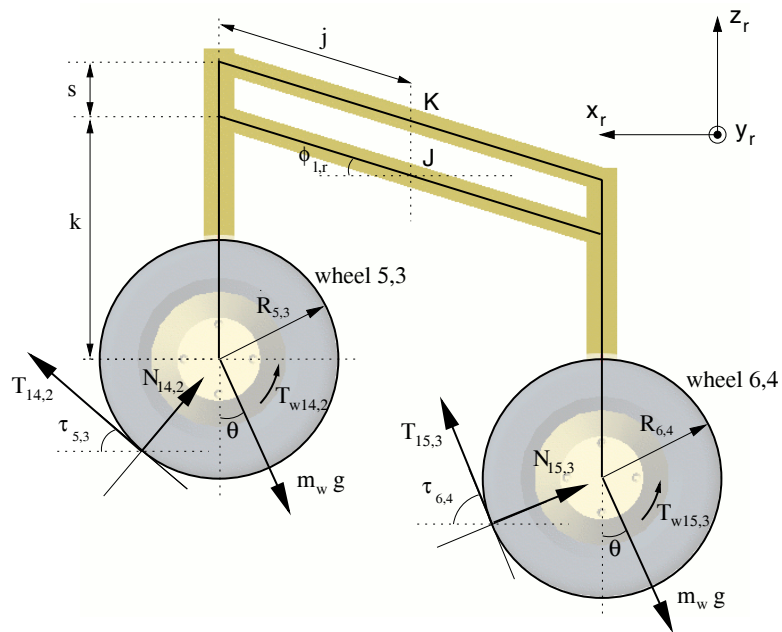


Figure A.2: Variables and dimensions of the bogie. l and r denote the left and right bogie respectively. The bogies are attached to the main body through pin joints placed at points J and

A.3 The main body and the rear wheel

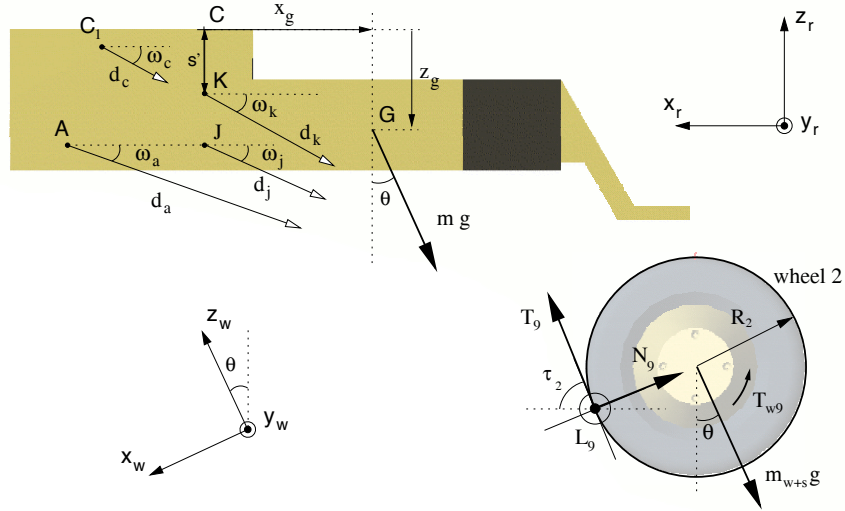


Figure A.3: Variables and dimensions of the main body and the rear wheel

A.4 The front fork

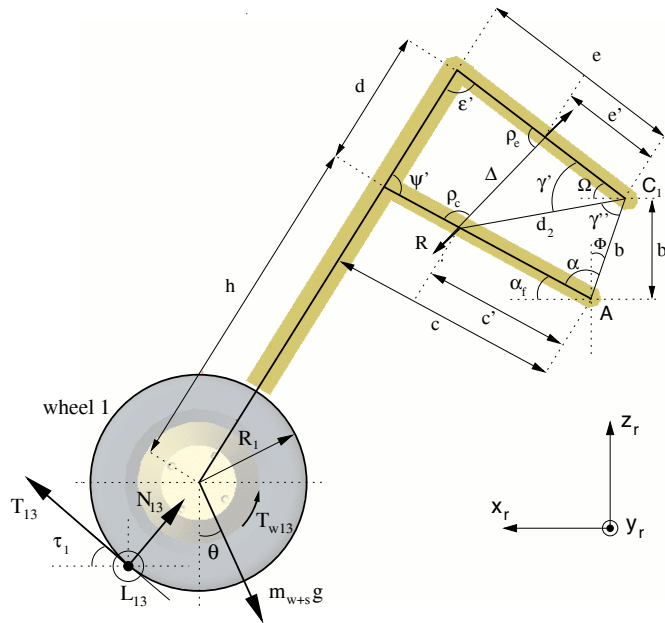


Figure A.4: Dimensions of the fork. A and C₁ are the interfaces with the main body.

A.5 Quasi-static model of SOLERO

$$\begin{aligned}
& -R_1 T_{13} + T_{W_{13}} = 0 \\
& -R_2 T_9 + T_{W_9} = 0 \\
& -R_3 T_2 + T_{W_2} = 0 \\
& -R_4 T_3 + T_{W_3} = 0 \\
& -R_5 T_{14} + T_{W_{14}} = 0 \\
& -R_6 T_{15} + T_{W_{15}} = 0 \\
& 2k \cdot mxBR + F_{78x} \cdot s + T_{W_2} + T_{W_3} - k (N_2 \cos(\tau_3) + N_3 \cos(\tau_4) + T_2 \sin(\tau_3) + T_3 \sin(\tau_4)) = 0 \\
& k \cdot \cos(\phi_j) \cdot (-T_2 \cos(\tau_3) + T_3 \cos(\tau_4) + N_2 \sin(\tau_3) - N_3 \sin(\tau_4)) + (2k \cdot mxBR + F_{78x} \cdot s + T_{W_2} + T_{W_3} - 2k \cdot (N_2 \cos(\tau_3) + T_2 \sin(\tau_3))) \sin(\phi_j) = 0 \\
& 2k \cdot mxBL - F_{819x} \cdot s + T_{W_{14}} + T_{W_{15}} - k(N_{14} \cos(\tau_5) + N_{15} \cos(\tau_6) + T_{14} \sin(\tau_5) + T_{15} \sin(\tau_6)) = 0 \\
& k \cos(\phi_j) (-T_{14} \cos(\tau_5) + T_{15} \cos(\tau_6) + N_{14} \sin(\tau_5) - N_{15} \sin(\tau_6)) + (2k \cdot mxBL - F_{819x} \cdot s + T_{W_{14}} + T_{W_{15}} - 2k(N_{14} \cos(\tau_5) + T_{14} \sin(\tau_5))) \sin(\phi_j) = 0 \\
& T_{810x} + h \cdot (L_{13} + myF) \sin(\xi) - c \cdot ((L_{13} + myF) \sin(\alpha_f) + L_{13} \cdot R_1 \cdot \sin(\tau_1)) = 0 \\
& T_{810z} + c \cdot (L_{13} + myF) \cos(\alpha_f) + L_{13} \cdot R_1 \cdot \cos(\tau_1) + h \cdot (L_{13} + myF) \cos(\xi) = 0 \\
& d \cdot (F_{811z} - F_{spz}) \cos(\xi) + d \cdot F_{811x} \sin(\xi) + h \cdot N_{13} \sin(\tau_1 + \xi) - (T_{W_{13}} + h \cdot mzF \cos(\xi) + h \cdot T_{13} \cos(\tau_1 + \xi) + d \cdot F_{spz} \sin(\xi) + h \cdot mxF \sin(\xi)) = 0 \\
& e \cdot F_{811z} \cos(\Omega) + (e' - e) \cdot F_{spz} \cos(\Omega) - e \cdot F_{811x} \sin(\Omega) - (e' - e) \cdot F_{spz} \sin(\Omega) = 0 \\
& (c' \cdot F_{spz} + c \cdot (F_{811z} + mxF)) \sin(\alpha_f) - ((c' \cdot F_{spz} + c \cdot (F_{811z} - F_{spz} + mzF)) \cos(\alpha_f) + c \cdot (T_{13} \cos(\alpha_f - \tau_1) + F_{spz} \sin(\alpha_f) + N_{13} \sin(\alpha_f - \tau_1))) = 0 \\
& - (k (Mx + mxR) + F_{819x} \cdot s) + F_{78x} \cdot s + T_{W_2} + T_{W_3} + T_{W_{14}} + T_{W_{15}} + k (N_{13} \cos(\tau_1) + N_9 \cos(\tau_2) + T_{13} \sin(\tau_1) + T_9 \sin(\tau_2)) = 0 \\
& N_{13} \sin(\tau_1) + N_9 \sin(\tau_2) + N_2 \sin(\tau_3) + N_3 \sin(\tau_4) + N_{14} \sin(\tau_5) + N_{15} \sin(\tau_6) - (Mz + 2 \cdot mzBL + 2 \cdot mzBR + mzF + T_{13} \cos(\tau_1) \\
& \quad + T_9 \cos(\tau_2) + T_2 \cos(\tau_3) + T_3 \cos(\tau_4) + T_{14} \cos(\tau_5) + T_{15} \cos(\tau_6)) = 0 \\
& L_{13} + L_9 + My + 2 \cdot myBL + 2 \cdot myBR + myF + myR = 0 \\
& 2 \cdot B' \cdot mzBR + T_{810x} + My \cdot (c_b + z_g) + L_9 \cdot R_2 \sin(\tau_2) + B'(T_2 \cos(\tau_3) + T_3 \cos(\tau_4) + N_{14} \sin(\tau_5) + N_{15} \sin(\tau_6)) - (2k(myBL + myBR) \\
& \quad + 2 \cdot B' \cdot mzBL + Mz \cdot y_g + B'(T_{14} \cos(\tau_5) + T_{15} \cos(\tau_6) + N_2 \sin(\tau_3) + N_3 \sin(\tau_4)) + d_j (L_9 + myR) \sin(\omega_j)) = 0 \\
& B'((F_{78x} + F_{819x})s + T_{W_2} + T_{W_3}) + k \cdot (T_{810z} + Mx \cdot y_g) + d_j \cdot k(L_9 + myR) \cos(\omega_j) - (d_{ju} \cdot k \cdot (L_{13} + myF) \\
& \quad + B' \cdot (T_{W_{14}} + T_{W_{15}}) + k \cdot My \cdot x_g + k \cdot L_9 \cdot R_2 \cos(\tau_2)) = 0 \\
& F_{78x} \cdot s + Mx \cdot (c_b + z_g) + d_j \cdot T_9 \cos(\tau_2 - \omega_j) + d_j \cdot mzR \cos(\omega_j) + d_{ju} \cdot N_{13} \sin(\tau_1) - (b' \cdot F_{811x} + b'' \cdot F_{811z} + d_{ju} \cdot mzF + F_{819x} \cdot s \\
& \quad + T_{W_9} + Mz \cdot x_g + d_{ju} \cdot T_{13} \cos(\tau_1) + d_j \cdot N_9 \sin(\tau_2 - \omega_j) + d_j \cdot mxR \sin(\omega_j)) = 0
\end{aligned}$$

with

$$\begin{aligned}
R_x &= \cos(\pi - \alpha_f - \rho_c) \cdot R \\
R_z &= -\sin(\pi - \alpha_f - \rho_c) \cdot R \\
mxF &= m_{w+s} \cdot g \sin(\theta) \\
myF &= -m_{w+s} \cdot g \sin(\phi) \cos(\theta) \\
mzF &= -m_{w+s} \cdot g \cos(\phi) \cos(\theta) \\
mxR &= m_{w+s} \cdot g \sin(\theta) \\
myR &= -m_{w+s} \cdot g \sin(\phi) \cos(\theta) \\
mzR &= -m_{w+s} \cdot g \cos(\phi) \cos(\theta) \\
mxBR &= m_w \cdot g \sin(\theta) \\
myBR &= -m_w \cdot g \sin(\phi) \cos(\theta) \\
mzBR &= -m_w \cdot g \cos(\phi) \cos(\theta) \\
mxBL &= m_w \cdot g \sin(\theta) \\
myBL &= -m_w \cdot g \sin(\phi) \cos(\theta) \\
mzBL &= -m_w \cdot g \cos(\phi) \cos(\theta) \\
Mx &= (M - 4 \cdot m_w - 2 \cdot m_{w+s}) \cdot g \sin(\theta) \\
My &= -(M - 4 \cdot m_w - 2 \cdot m_{w+s}) \cdot g \sin(\phi) \cos(\theta) \\
Mz &= -(M - 4 \cdot m_w - 2 \cdot m_{w+s}) \cdot g \cos(\phi) \cos(\theta)
\end{aligned}$$

This set of equations still contains several internal forces. They are later removed using the Gauss-Jordan elimination.

A.5.1 Linear dependence of the wheel torques

With a Gauss-Jordan elimination, this equation system can be simplified. In particular all the internal variables are removed. The final system has 15 equations with 20 unknown and can be written as

$$Q_{20 \times 14} F_{14 \times 1} = A_{20 \times 1} - B_{20 \times 6} M_{6 \times 1} \quad (\text{A.1})$$

with F a vector containing the unknown forces and M the vector containing the torques. The matrices Q , A and B contain the information about the gravity, the rover's geometry and state. The forces can be expressed as a function of the torques

$$F_{14 \times 1} = \text{pinv}(Q)_{14 \times 20} [A_{20 \times 1} - B_{20 \times 6} M_{6 \times 1}] \quad (\text{A.2})$$

with $\text{pinv}(Q)$, the pseudo-inverse of Q . Now the linear dependence of the torque is proven using the null space property.

Null space definition

M is a linear application. One call kernel (or null space) of M, the set of vectors whose image by M is the null vector

$$C_{n \times m} \cdot \text{null}(C)_{m \times 1} = \vec{0}_{n \times 1} \quad (\text{A.3})$$

One can write

$$[C \cdot \text{null}(C)]^T = \text{null}(C)^T \cdot C^T = \vec{0}_{l \times n} \quad (\text{A.4})$$

and then

$$[C^T \cdot \text{null}(C^T)]^T = \text{null}(C^T)^T \cdot C = \vec{0}_{n \times l} \quad (\text{A.5})$$

Using the property of A.5 applied to A.1 one can write

$$\text{null}(Q^T)^T \cdot Q \cdot F = \vec{0} = \text{null}(Q^T)^T \cdot [A - B \cdot M] \quad (\text{A.6})$$

rewriting A.6 we obtain

$$\text{null}(Q^T)^T \cdot B \cdot M = \text{null}(Q^T)^T \cdot A \quad (\text{A.7})$$

$$B'_{1 \times 6} \cdot M_{6 \times 1} = A'_{1 \times 1}$$

Equation A.7 proves that the torques are linearly dependant. Furthermore, this confirms that the solution space is of dimension $m-1$ where m is the number of wheels.

A.5.2 Equal torques solution

For SOLERO the solution space is of dimension 5. Among all the possible solutions, the set of torque defined by A.8 is a solution of the system.

$$E_{6 \times 1} = \frac{A'_{1 \times 1}}{\sum_{i=1}^6 B'_{1 \times 6}(1, i)} \cdot [1 \ 1 \ 1 \ 1 \ 1 \ 1] \quad (\text{A.8})$$

B

Linearized equations

The non-linear equations of chapter 5 are linearized in this appendix. In what follows c and s correspond to the cosine and sine functions, the bars on the symbols denote of nominal values and h is the sampling time.

B.1 Accelerometers model

The Jacobian for the linearized accelerometers model has the following form

$$\nabla \left[\bar{\Gamma}_{WR} \cdot \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} - g \end{bmatrix} \right] = \begin{bmatrix} 0 \\ (\bar{z} - g)c\bar{\theta}c\bar{\phi} + \bar{y}(-c\bar{\psi}s\bar{\phi} + c\bar{\phi}s\bar{\theta}s\bar{\psi}) + \bar{x}(c\bar{\phi}c\bar{\psi}s\bar{\theta} + s\bar{\phi}s\bar{\psi}) \\ -(\bar{z} - g)c\bar{\theta}s\bar{\phi} + \bar{x}(-c\bar{\psi}s\bar{\theta}s\bar{\phi} + c\bar{\phi}s\bar{\psi}) + \bar{y}(-c\bar{\phi}c\bar{\psi} - s\bar{\theta}s\bar{\phi}s\bar{\psi}) \\ -(\bar{z} - g)c\bar{\theta} - \bar{x}c\bar{\psi}s\bar{\theta} - \bar{y}s\bar{\theta}s\bar{\psi} \\ \bar{x}c\bar{\theta}c\bar{\psi}s\bar{\phi} - (\bar{z} - g)s\bar{\theta}s\bar{\phi} + \bar{y}c\bar{\theta}s\bar{\phi}s\bar{\psi} \\ \bar{x}c\bar{\theta}c\bar{\phi}c\bar{\psi} - (\bar{z} - g)c\bar{\phi}s\bar{\theta} + \bar{y}c\bar{\theta}c\bar{\phi}s\bar{\psi} \\ \bar{y}c\bar{\theta}c\bar{\psi} - \bar{x}c\bar{\theta}s\bar{\psi} \\ \bar{y}(c\bar{\psi}s\bar{\theta}s\bar{\phi} - c\bar{\phi}s\bar{\psi}) + \bar{x}(-c\bar{\phi}c\bar{\psi} - s\bar{\theta}s\bar{\phi}s\bar{\psi}) \\ \bar{x}(c\bar{\psi}s\bar{\theta}s\bar{\phi} - c\bar{\phi}s\bar{\psi}) + \bar{y}(c\bar{\phi}c\bar{\psi}s\bar{\theta} + s\bar{\phi}s\bar{\psi}) \end{bmatrix}$$

B.2 Gyroscopes state transition

$$\begin{bmatrix} \omega_x \\ \phi \\ \omega_y \\ \theta \\ \omega_z \\ \psi \end{bmatrix}_{k+1} = F_\omega \cdot \begin{bmatrix} \omega_x \\ \phi \\ \omega_y \\ \theta \\ \omega_z \\ \psi \end{bmatrix}_k \quad \text{with}$$

B.2 GYROSCOPES STATE TRANSITION

$$F_\omega = \begin{bmatrix} e^{-\tau_{\omega_x} h} & 0 & 0 & 0 & 0 & 0 \\ h & 1+h(\bar{\omega}_y c\bar{\phi}-\bar{\omega}_z s\bar{\phi})\tan\bar{\theta} & hs\bar{\phi}\tan\bar{\theta} & \frac{h}{c^2\bar{\theta}}(\bar{\omega}_z c\bar{\phi}+\bar{\omega}_y s\bar{\phi}) & hc\bar{\phi}\tan\bar{\theta} & 0 \\ 0 & 0 & e^{-\tau_{\omega_y} h} & 0 & 0 & 0 \\ 0 & h(-\bar{\omega}_z \cos\bar{\phi}-\bar{\omega}_y \sin\bar{\phi}) & hc\bar{\phi} & 1 & -hs\bar{\phi} & 0 \\ 0 & 0 & 0 & 0 & e^{-\tau_{\omega_z} h} & 0 \\ 0 & \frac{h}{c\bar{\theta}}(\bar{\omega}_y c\bar{\phi}-\bar{\omega}_z s\bar{\phi}) & h\frac{s\bar{\phi}}{c\bar{\theta}} & \frac{h}{c\bar{\theta}}(\bar{\omega}_z c\bar{\phi}+\bar{\omega}_y s\bar{\phi})\tan\bar{\theta} & h\frac{c\bar{\phi}}{c\bar{\theta}} & 1 \end{bmatrix}$$

C

The Gauss-Markov process

The aim of this appendix is to derive the equations for a double integrated Gauss-Markov process. A Gauss-Markov process is a stochastic process with zero mean and whose autocorrelation function is

$$R_p(t) = \sigma^2 e^{-\tau|t|} \quad (\text{C.1})$$

where $1/\tau$ is the time constant defining the correlation time of the process and σ^2 its variance. The power spectral density function of P is

$$S_p(j\omega) = \int_{-\infty}^{\infty} R_p(t) \cdot e^{-j\omega t} dt = \frac{2\sigma^2\tau}{\omega^2 + \tau^2} \quad (\text{C.2})$$

A Gauss-Markov process can also be considered as a white noise being filtered by a low pass filter with transfer function

$$H(j\omega) = \frac{\sqrt{2\sigma^2\tau}}{j\omega + \tau} \quad (\text{C.3})$$

which is derived from the following relationship

$$S_p(j\omega) = |H(j\omega)|^2 S_U(j\omega) \quad (\text{C.4})$$

where $S_U(j\omega)$ is a unity white noise signal.

Fig. C.1 depicts the double integration of a Gauss-Markov process p_1 . $u(t)$ is a unity white noise and p_2 and p_3 are respectively the first and second integral of p_1 .

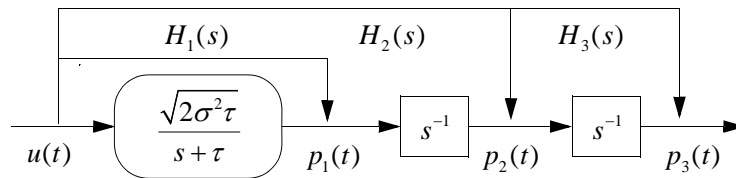


Figure C.1: Double integration of a Gauss-Markov process

The transfer functions between u and p_1, p_2 and p_3 are respectively

$$H_1(s) = \frac{\sqrt{2\sigma^2\tau}}{s + \tau} \quad H_2(s) = \frac{\sqrt{2\sigma^2\tau}}{s(s + \tau)} \quad H_3(s) = \frac{\sqrt{2\sigma^2\tau}}{s^2(s + \tau)} \quad (\text{C.5})$$

and the impulse responses are

$$\begin{aligned} h_1(t) &= \sqrt{2\sigma^2\tau} e^{-\tau t} \\ h_2(t) &= \sqrt{\frac{2\sigma^2}{\tau}} (1 - e^{-\tau t}) \\ h_3(t) &= \sqrt{\frac{2\sigma^2}{\tau^3}} (\tau t - 1 + e^{-\tau t}) \end{aligned} \quad (\text{C.6})$$

The continuous state transition model is in this case

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \end{bmatrix} = \begin{bmatrix} -\tau & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + \begin{bmatrix} \sqrt{2\sigma^2\tau} \\ 0 \\ 0 \end{bmatrix} u(t) \quad (\text{C.7})$$

In order to derive the state transition matrix Φ and the corresponding covariance matrix Q , the discrete form of C.7 need to be derived. Using the inverse of the Laplace function the state transition matrix is

$$\begin{aligned} \Phi &= \mathcal{L}^{-1}[(sI - F)^{-1}] = \mathcal{L}^{-1} \begin{bmatrix} s + \tau & 0 & 0 \\ -1 & s & 0 \\ 0 & -1 & s \end{bmatrix}^{-1} = \mathcal{L}^{-1} \begin{bmatrix} 1/s + \tau & 0 & 0 \\ 1/s(s + \tau) & 1/s & 0 \\ 1/s^2(s + \tau) & 1/s^2 & 1/s \end{bmatrix} \\ &= \begin{bmatrix} e^{-\tau h} & 0 & 0 \\ (1 - e^{-\tau h})/\tau & 1 & 0 \\ (\tau h - 1 + e^{-\tau h})/\tau^2 & h & 1 \end{bmatrix} \end{aligned} \quad (\text{C.8})$$

where h is the sampling time.

The determination of Q is more difficult because the expectations have to be computed. The covariance between two sequences can be derived as

$$E[p_i p_j] = \int_0^h \int_0^h h_i(\lambda) h_j(\varepsilon) E[u(\lambda)u(\varepsilon)] d\lambda d\varepsilon \quad (\text{C.9})$$

If the signal u is a unity white noise C.9 can be simplified to

$$E[p_i p_j] = \int_0^h \int_0^h h_i(\lambda) h_j(\varepsilon) \delta(\lambda - \varepsilon) d\lambda d\varepsilon = \int_0^h h_i(\lambda) h_j(\lambda) d\lambda \quad (\text{C.10})$$

Using C.10 all the elements of the covariance matrix Q can be computed. Only $E[p_2 p_2]$ and $E[p_2 p_3]$ are derived here. The other terms are obtained in a similar way

$$\begin{aligned} E[p_2 p_2] &= \int_0^h h_2(\lambda) h_2(\lambda) d\lambda = \int_0^h \left[\sqrt{\frac{2\sigma^2}{\tau}} (1 - e^{-\tau\lambda}) \right]^2 d\lambda \\ &= \frac{2\sigma^2}{\tau} \left[h - \frac{2}{\tau} (1 - e^{-\tau h}) + \frac{1}{2\tau} (1 - e^{-2\tau h}) \right] \end{aligned} \quad (\text{C.11})$$

$$\begin{aligned} E[p_2 p_3] &= \int_0^h h_2(\lambda) h_3(\lambda) d\lambda = \int_0^h \sqrt{\frac{2\sigma^2}{\tau}} (1 - e^{-\tau\lambda}) \cdot \sqrt{\frac{2\sigma^2}{\tau^3}} (\tau\lambda - 1 + e^{-\tau\lambda}) d\lambda \\ &= \frac{2\sigma^2}{\tau^2} \left[\frac{\tau}{2} h^2 - h (1 - e^{-\tau h}) + \frac{1}{\tau} (1 - e^{-\tau h}) - \frac{1}{2\tau} (1 - e^{-2\tau h}) \right] \end{aligned} \quad (\text{C.12})$$

Finally Q becomes

$$Q = \begin{bmatrix} E[p_1 p_1] & E[p_1 p_2] & E[p_1 p_3] \\ E[p_2 p_1] & E[p_2 p_2] & E[p_2 p_3] \\ E[p_3 p_1] & E[p_3 p_2] & E[p_3 p_3] \end{bmatrix} \quad (\text{C.13})$$

D

Visual Motion Estimation

This appendix presents the principle of the Visual Motion Estimation technique (VME, also known as visual odometry), which allows to compute an estimate of the six displacement parameters between two stereo pairs acquisitions [Mallet00] [Olson01]. The technique we use on SOLERO has been developed at LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes) and ported on SOLERO thanks the help of Simon Lacroix and Il-Kyun Jung. The following figure summarizes the approach

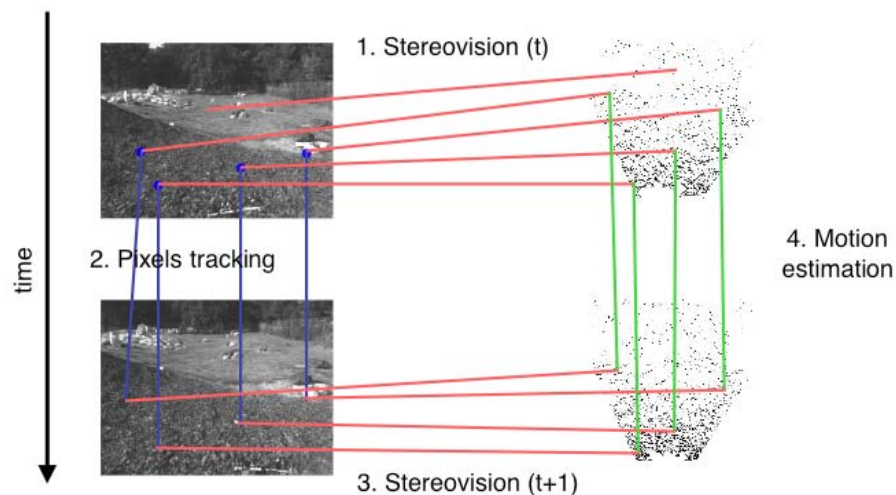


Figure D.1: : Principle of the Visual Motion Estimation (the illustration comes from LAAS)

1. At time t a stereo pair is acquired and interest points are extracted in both images. The interest points extraction phase uses the Harris corner detector [Harris88]. Then, the interest point matching technique presented in [Jung01] is used to both find correspondences between the interest points in both images and reject false matches. Finally, the cloud of 3D points corresponding to the interest points is obtained using stereovision and a second outliers rejection

tion cycle is performed [Jung03]. For speeding up VME, stereo is only computed for the interest points.

2. At time $t+1$, a new stereo pair is acquired and the Harris points are again extracted from both images. Then, the correspondences between the interest point extracted in the left images (acquired at time t and $t+1$) are searched using the same technique as presented in 1.
3. The stereovision is used to compute the cloud of 3D points at time $t+1$.
4. Finally, the six displacement parameters between t and $t+1$ are computed using the least square minimization technique presented in [Haralick89].

More details about the Visual Motion Estimation technique can be found in [Jung03]. In particular, the error model associated to the estimation of the six displacement parameters is presented in details. This error model is required for sensor fusion.

Literature

- [Andrade98] Andrade G., Amar F.B., Bidaud P., Chatila R., “Modeling robot-soil interaction for planetary rover motion control”, IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, Canada, 1998.
- [Arras98] Arras K. O., “An Introduction To Error Propagation: Derivation, Meaning and Example of Equation $Cy = Fx Cx Ftx$ ”, Technical Report, EPFL-ASL-TR-98-01 R3.
- [Arras03] Arras K. O., “Feature-based robot navigation in known and unknown environments”, Thèse n° 2765, Département de Microtechnique, École Polytechnique Fédérale de Lausanne, 2003.
- [Atacama] <http://www.frc.ri.cmu.edu/atacama>
- [Barshan95] Barshan B., Durrant-Whyte H. F., “Inertial Navigation Systems for Mobile Robots”, IEEE Transaction on Robotics and Automation, 1995.
- [Baumgartner00] Baumgartner E. T., Aghazarian H., Trebi-Ollennu A., Huntsberger T. L., Garrett M. S., “State Estimation and Vehicle Localization for the FIDO Rover”, Sensor Fusion and Decentralized Control in Autonomous Robotic Systems III, SPIE Proc. Vol. 4196, Boston, USA, 2000.
- [Bekker56] Bekker G., “Theory of Land Locomotion”, University of Michigan, Ann Arbor, 1956.
- [Bekker69] Bekker G., “Introduction to Terrain-Vehicle Systems”, University of Michigan Press, MI, 1969.
- [Bevly01] Bevly D.M., Sheridan R., Gerdes J.C., “Integrating INS sensors with GPS velocity measurements for continuous estimation of vehicle sideslip and tire cornering stiffness”, Proceedings of the American Control Conference, Volume 1, 2001.

- [Bonnafous01] Bonnafous D., Lacroix S., Simeon T., “Motion generation for a rover on rough terrains”, In the proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001.
- [Borenstein96] Borenstein J., Feng L., “Gyrodometry: A new method for combining data from gyros and odometry in mobile robots”, IEEE International Conference on Robotics and Automation, Minneapolis, USA, 1996.
- [Burg97] van der Burg J., Blazevic P., “Anti-Lock Braking and Traction Control Concept for All-Terrain Robotic Vehicles”, In the proceedings of the 1997 IEEE International Conference on Robotics and Automation, USA, 1997.
- [Chahl97] Chahl J.S., Srinivasan M.V., “Reflective surfaces for panoramic imaging”, Applied Optics, 1997.
- [Dissanayake01] Dissanayake G., Sukkarieh S., Nebot E., Durrant-Whyte H., “The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications”, IEEE Transactions on Robotics and Automation, Oct. 2001.
- [Cozman00] Cozman F., Krotkov E., gustrin C., “Outdoor Visual Position Estimation for Planetary Rovers”, Autonomous Robots, Volume 9, Issue 2, Kluwer Academic Publishers, Hingham, MA, USA, 2000.
- [Estier00] Estier T., Crausaz Y., Merminod B., Lauria M., Piguat R., Siegwart R., “An Innovative Space Rover with Extended Climbing Abilities”, In Proceedings of Space & Robotics, the Fourth International Conference and Exposition on Robotics in Challenging Environments, Albuquerque, USA, 2000.
- [Gancet03] Gancet J., Lacroix S., “PG2P: a perception-guided path planning approach for long range autonomous navigation in unknown natural environments”, IEEE/RSJ International Conference on Intelligent Robots and Systems, USA, 2003.
- [Haralick89] R. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V.G. Vaidya, M.B. Kim., “Pose estimation from corresponding point data”, IEEE Transactions on Systems, Man and Cybernetics, 19(6):1426-1446, Nov/Dec 1989.
- [Harris88] C. Harris and M. Stephens, “A combined corner and edge detector”, In Alvey Vision Conference, page 147-151, 1988.

- [Hung00] Hung M.-H., Orin D. E., Waldron K. J., "Efficient Formulation of the Force Distribution Equations for General Tree-Structured Robotic Mechanisms with a Mobile Base", IEEE Transactions on Systems, man and Cybernetics, Part B: Cybernetics, Vol 30, No 4, August 2000.
- [Iagnemma00] Iagnemma K., Dubowsky S., "Mobile robot rough-terrain control (RTC) For planetary exploration", Proceedings ASME Design Engineering Technical Conferences, Baltimore, Maryland, USA, 2000.
- [Iagnemma01] Iagnemma K., Shibly H., Rzepniewski A., Dubowsky S., "Planning and Control Algorithms for Enhanced Rough-Terrain Rover Mobility", Proceedings of the Sixth International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS, 2001.
- [Iagnemma02] Iagnemma K., Shibley H., Dubowsky S., "On-Line Terrain Parameter Estimation for Planetary Rovers", IEEE International Conference on Robotics and Automation, Washington D.C, USA, 2002.
- [IPC] <http://www-2.cs.cmu.edu/~IPC/>
- [Jung01] Jung I-K., Lacroix S., "A robust Interest Point Matching Algorithm", In International Conference on Computer Vision, Vancouver, Canada, 2001.
- [Jung03] Jung I-K., Lacroix S., "Simultaneous Localization and Mapping with Stereovision", International Symposium on Robotics Research, Siena, 2003.
- [Kalker90] Kalker J.J., "Three dimensional elastic bodies in rolling contact", Kluwer Academic Publishers, Dordrecht, 1990.
- [Lacroix02] Lacroix S., Mallet A., Bonnafous D., Bauzil G., Fleury S., Herrb M., Chatila R., "Autonomous rover navigation on unknown terrains: functions and integration", International Journal of Robotics Research, 2002.
- [Lamon03] Lamon P., Siegwart R., "3D-Odometry for rough terrain - Towards real 3D navigation", IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003.
- [Lamon04a] Lamon, P., Krebs, A., Lauria, M., Shooter, S. and Siegwart, R., "Wheel torque control for a rough terrain rover", IEEE International Conference on Robotics and Automation, New Orleans, USA, 2004.

- [Lamon04b] Lamon P., Siegwart R., “Inertial and 3D-odometry fusion in rough terrain – Towards real 3D navigation”, IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 2004.
- [Lamon05] Lamon P., Siegwart R., “Wheel torque control in rough terrain - modeling and simulation”, IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, *accepted*.
- [Lauria02] Lauria M., Piguat Y. and Siegwart R., “Octopus - An Autonomous Wheeled Climbing Robot”, In Proceedings of the Fifth International Conference on Climbing and Walking Robots, Published by Professional Engineering Publishing Limited, Bury St. Edmunds and London, UK, 2002.
- [Lauria03a] Lauria M., Shooter S., Siegwart R., “Topological Analysis of Robotic N-Wheeled Ground Vehicles”, In Proceedings of the 4th International Conference on Field and Service Robotics, Yamanashi, Japan, 2003.
- [Lauria03b] Lauria M., “Nouveaux concepts de locomotion pour véhicules tout-terrain robotisés”, Doctoral Thesis Nr. 2833, EPFL, Lausanne, 2003.
- [Mabie87] Mabie H.H., Reinholtz C.F., “Mechanisms and Dynamics of Machinery”, 4th Edition, John Wiley and Sons, New York, 1987.
- [Mallet00] Mallet A., Lacroix S., Gallo L., “Position estimation in outdoor environments using pixel tracking and stereovision”, IEEE International Conference on Robotics and Automation, San Francisco, USA, April 2000.
- [Manyika94] Manyika J., Durrant-Whyte H., “Data fusion and sensor management: A decentralized information-theoretic approach”, Ellis Horwood Limited, 1994.
- [Michaud02] Michaud S., Schneider A., Bertrand R., Lamon P., Siegwart R., van Winnendael M., Schiele A., “SOLERO: Solar Powered Exploration Rover”, In the Proceedings of the 7 ESA Workshop on Advanced Space Technologies for Robotics and Automation, The Netherlands, 2002.
- [Nebot97] Nebot E., Sukkarieh S., Durrant-Whyte H., “Inertial navigation aided with GPS information”, In the proceedings of the Fourth Annual Conference of Mechatronics and Machine Vision in Practice, Sept. 1997.
- [ODE] Open Dynamic Engine, <http://ode.org>

- [Ollis99] Ollis M., Hermann H., Singh S., "Analysis and Design of Panoramic Stereo Vision Using Equi-Angular Pixel Cameras", Technical Report, CMU-RI-TR-99-04.
- [Olson01] Olson C.F., Matthies L.H., Schoppers M., Maimone M.W., "Stereo ego-motion improvements for robust rover navigation", IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001.
- [Peynot03] Peynot T., Lacroix S., "Enhanced Locomotion Control for a Planetary Rover", In the proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Las Vegas, USA, 2003.
- [Roumeliotis99] Roumeliotis S.I., Bekey G.A., "3D Localization for a Mars Rover Prototype", In 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '99), ESTEC, The Netherlands, 1999.
- [Roumeliotis02] Roumeliotis S.I., Johnson A.E., Montgomery J.F., "Augmenting inertial navigation with image-based motion estimation", IEEE International Conference on Robotics and Automation, Proceedings, Washington, USA, 2002.
- [Scheding99] Scheding S., Dissanayake G., Nebot E.M., Durrant-Whyte H., "An experiment in autonomous navigation of an underground mining vehicle", IEEE Transactions on Robotics and Automation, Volume: 15, Issue: 1, Feb. 1999.
- [Shiller91] Shiller Z., Gwo Y.-R., "Dynamic motion planning of autonomous vehicles", IEEE Transactions on Robotics and Automation, Vol. 7, Issue: 2, 1991.
- [Singh00] Singh S., Simmons R., Smith T., Stentz A., Verma V., Yahja A., Schwehr K., "Recent Progress in Local and Global Traversability for Planetary Rovers", IEEE International Conference on Robotics and Automation, San Francisco, USA, 2000.
- [Siegwart00] Siegwart R., Estier T., Crausaz Y., Merminod B., Lauria M., Piguet R., "Innovative Concept for Wheeled Locomotion in Rough Terrain", In Proceedings of the Sixth International Conference on Intelligent Autonomous Systems, Venice, Italy, 2000.
- [Siegwart02] Siegwart R., Lamon P., Estier T., Lauria M., Piguet R., "Innovative design for wheeled locomotion in rough terrain", Journal of Robotics and Autonomous Systems, Elsevier, vol 40/2-3 p151-162.

- [Strelow01] Strelow D., Mishler J., Singh S., Herman H., "Extending shape-from-motion to noncentral omnidirectional cameras", IEEE/RSJ International Conference on Intelligent Robots and Systems, Hawaii, USA, 2001.
- [Strelow03] Strelow D., Singh S., "Online Motion Estimation from Image and Inertial Measurements", The 11th International Conference on Advanced Robotics, Portugal, 2003.
- [Titterton97] Titterton D. H., Weston J. L., "Strapdown inertial navigation technology", Stevenage, United Kingdom: Institution of Electrical Engineers, cop. 1997.
- [Tomatis01] Tomatis N., "Hybrid, Metric - Topological, Mobile Robot Navigation", Thèse n° 2444, Département de Microtechnique, École Polytechnique Fédérale de Lausanne, 2001.
- [Trobi01] Trobi-Ollennu A., Huntsberger T., Yang Cheng, Baumgartner E.T., Kennedy B., Schenker P., "Design and analysis of a sun sensor for planetary rover absolute heading detection", IEEE Transactions on Robotics and Automation, Issue: 6, Dec. 2001.
- [Vieville93] Vieville T., Romann F., Hotz B., Mathieu H., Buffa M., Robert L., Façao P.E.D.S., Faugeras O.D., Audren J.T., "Autonomous navigation of a mobile robot using inertial and visual cues", IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 1993.
- [Wada00] Wada M., Kang Sup Yoon, Hashimoto H., "High accuracy multisensor road vehicle state estimation", 26th Annual Conference of the IEEE Industrial Electronics Society, IECON, 2000.
- [Yoshida02] Yoshida K., Hamano H., Watanabe T., "Slip-Based Traction Control of a Planetary Rover", In the proceedings of the 8th International Symposium on Experimental Robotics, ISER, Italy, 2002.

Curriculum Vitae

Born 25 December 1974, I grew up in Sierre, VS, Switzerland. In 1994, I graduated from the *Lycée Collège des Creusets de Sion*, high school in science (Type C). After one year in physics in the *Swiss Federal Institute of Technology Zurich* I changed orientation and entered the *Swiss Federal Institute of Technology Lausanne* in the section of micro-engineering. I completed my study (Dipl. Ing EPFL) in spring 2000 with the diploma work entitled “Deriving and matching image fingerprint sequences for mobile robot localization”. This work has been done at Carnegie Mellon University (CMU, PA, USA) and received an award (*Approche originale en informatique pouvant intéresser le milieu industriel*). Then, I started a PhD at the Autonomous System Laboratory under the supervision of Roland Siegwart. During the PhD, I’ve spent four months at CMU for implementing an autonomous navigation software on Shrimp and at LAAS for ten weeks for the implementation a visual motion estimation algorithm. In parallel to the thesis, I was also lecturer in micro-informatics (third year students).

