SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria Civile ed Industriale

DIAEE - Dipartimento di Ingegneria Astronautica Elettrica ed Energetica

SIA - Scuola di Ingegneria Aerospaziale

Ph.D degree in "Energy and Environment"

AUTONOMOUS VEHICLE GUIDANCE IN UNKNOWN ENVIRONMENTS

Candidate
Marco Carpentiero

Supervisor
Prof. Giovanni B. Palmerini

A.A 2019/2020

A chi mi ha sempre incoraggiato

Index Of Contents

Introduction ..... **I**

Chapter 1. Autonomous Guidance Exploiting Stereoscopic Vision **1**

1.1 Introduction 1

1.2 Stereoscopic vision concept 3

1.3 The software architecture of the stereo-based navigation system 7

1.3.1 Stereo image acquisition 9

1.3.2 Stereo image processing 9

1.3.3 Obstacles Detection and Localization 13

1.4 Stereo Vision applied to robotic manipulators 14

Chapter 2. A Vehicle Prototype and its Guidance System **19**

2.1 Introduction 19

2.2 The rover RAGNO 20

2.3 The guidance system 24

2.4 Test sessions 27

2.5 Conclusions and further developments 31

Chapter 3. Examples of Possible Missions and Applications **33**

3.1 Introduction 33

3.2 Rover-Drone Coordination 38

3.2.1 Related works 38

3.2.2 RAGNO-UAV proposed architecture 39

3.3 Experiment 41

3.4 Final Remarks 45

Chapter 4. Guidance of Fleet of Autonomous Vehicles 47

4.1 Introduction 47

4.2 STAR: Swarm of Two Autonomous Rovers 49

4.3 Rovers' implementation and trajectory choice 50

4.4 The guidance strategy 51

4.5 Simulations and results 57

4.6 About the Navigation-Communications Subsystems' Connection in Space Exploration Vehicles 59

4.7 Communication constraints 61

4.8 Planetary surface radio-propagation 62

4.9 Navigation 63

4.10 Estimation 66

Final remarks and future perspectives 69

References 72

Introduction

Gaining from significant advances in their performance granted by technological evolution, Autononous Vehicles are rapidly increasing the number of fields of applications. From operations in hostile, dangerous environments (military use in removing unexploded projectiles, survey of nuclear power and chemical industrial plants following accidents) to repetitive 24h tasks (border surveillance), from power-multipliers helping in production to less exotic commercial application in household activities (cleaning robots as consumer electronics products), the combination of autonomy and motion offers nowadays impressive options. In fact, an autonomous vehicle can be completed by a number of sensors, actuators, devices making it able to exploit a quite large number of tasks. However, in order to successfully attain these results, the vehicle should be capable to navigate its path in different, sometimes unknown environments. This is the goal of this dissertation to analyze and - mainly -to propose a suitable solution for this problem. The frame in which this research takes its steps is the activity carried on at the *Guidance and Navigation Lab of Sapienza – Università di Roma*, hosted at the *School of Aerospace Engineering*. Indeed, the solution proposed has an intrinsic, while not limiting, bias towards possible space applications, as it will become obvious in some of the following content. A second bias dictated by the Guidance and Navigation Lab activities is represented by the choice of a sample platform. In fact, it would be difficult to perform a meaningful study keeping it a very general level, independent on the characteristics of the targeted platform: it is easy to see from the rough list of applications cited above that these characteristics are extremely varied. The Lab hosted – before the beginning of this thesis activity – a simple, home-designed and manufactured model of a small, yet performing enough autonomous vehicle, called RAGNO (standing for Rover for Autonomous Guidance Navigation and Observation): it was an obvious choice to select that rover as the reference platform to identify solutions for guidance, and to use it, cooperating to its

improvement, for the test activities which should be considered as mandatory in this kind of thesis work to validate the suggested approaches.

This dissertation has been conceived as a focused, detailed description of the results achieved during the period as PhD candidate. It is not intended to offer a global view of the Guidance and Navigation subsystem applicable to autonomous vehicles, that should encompass many more different techniques and algorithms specialized for the different applications. Instead, it has been preferred the studies and findings authored during the period. Such a choice has been due to both (1) the opinion that the global scenario description would have required a really remarkable experience (especially a "hands-on" one to appreciate effective advantages and disadvantages of the past, existing or designed) still unavailable at the PhD level, and indeed (2) the risk to produce a kind of (almost useless) repletion of surveys already available in literature without addition of any significant contribution.

On the other side, the approach selected allowed to duly discuss the peculiarities of the problems tackled during the research and of the proposed – and tested – solutions, indeed at least attempting to offer a maybe useful and certainly original contribution.

The present dissertation has been divided in four main chapters, each of them focused on a relevant topic of the research.

Chapter 1 deals with the main topic of the work, i.e the stereoscopic vision applied for autonomous guidance problem. The concept and theory of the stereovision are presented in detail together with its application in the field of computer vision such as the image processing. The most important features descriptors (BRISK and SURF) are then analyzed to prove, even with experimental results, their effectiveness for objects recognition and localization scope. The last part of the chapter reports an example of application of the stereovision system in the field of the robotics manipulation as it spread from industrial plant (e.g. pick and place machines) to space applications (e.g. on-orbit servicing, visual inspection and so on).

Chapter 2 presents the wheeled rover, RAGNO, involved in the various test sessions to demonstrate both the efficiency and the limits of the implemented guidance algorithms based on the stereovision principles. The first part of the chapter is dedicated to the hardware architecture of the rover focusing on the modular concept adopted during the design and realization phases. In the second part of the chapter the attention moves on the description of the implemented and tested guidance algorithms: the Lyapunov-based strategy and the A-star graph-based guidance law.
The former is a well-known solution adopted for the guidance problem (especially in space applications like probes descent phase and landing) while the latter represents a milestone of the current work since it has been designed and developed during the research period. The results of simulation and tests are reported to demonstrate the advantages, of the A-Star approach with respect to the classic one both from the analytical and computational point of view.

Chapter 3 deals with the experimental campaigns lead by involving RAGNO in a stand-alone or cooperative formation. In addition, the needing of communication rules between members of swarm or formation of autonomous robots has been introduced.

Chapter 4 describes a possible architecture for the guidance of a fleet of autonomous robots operating in an unknown scenario with different tasks, considering the results and limits discussed in the previous chapters. In the first part of the chapter, the STAR (Swarm of Two Autonomous Rovers) rovers configuration is proposed as a possible swarm architecture for planetary exploration. The A-Star guidance algorithm has been adopted and a first attempt of communication protocol has been developed to ensure a coordinated and safe ground exploration and mapping. Simulation results are reported to prove the feasibility of the selected approach and its limits. The second part of the chapter focuses on the intra-platforms communication that is surely the main topic in the case of swarm exploration. The characteristics and constraints of the link, due to the disturbances that affect the planetary surface radio propagation, are analytically

presented. Finally, the last part of the chapter deals with the navigation system architecture designed for the STAR mission with the description Kalman algorithm needed for kinematic state estimation.

Chapter 1.

Autonomous Guidance Exploiting Stereoscopic Vision

1.1 Introduction

Computer vision attained extremely significant advances in recent years, thanks to continuing development of sensors, software and powerful computation resources [1]. Indeed, the applications spread in different areas, and especially in the ones closer to state-of-the-art technology. Among them, space engineering is clearly an elective area, and the Mars exploration carried out by planetary rovers is a very remarkable example for this technology [2].

Future space missions are likely to continue in exploiting computer vision systems, which are able to provide high accuracy and a rich and comprehensive autonomous perception of the scenarios/environment. Aside from the topic of the rover navigation, spacecraft proximity manoeuvres and in-orbit servicing also offer a wide area for possible applications. Until now, main tasks of computer vision in this field have been the detection and recognition of objects whose shape is already known: this approach was adopted in the automatic rendezvous, carried on by identifying specific markers in acquired images [3]. The improvements in optical sensors' quality, software skills and computational power allow today to attempt a full reconstruction of the observed scene, practically in real time. The capability to detect and avoid obstacles opens the possibility of robotic autonomous operations in the difficult, unprepared scenarios of orbital operations, where visual guidance of the manipulators would be extremely effective [4]. If in-orbit operations already present stricter requirements with respect to planetary rovers in terms of the guidance-navigation-control loop bandwidth, there are landing and very low altitude flights over unexplored, unknown landscapes that will be the ultimate, not yet fully attained goals.

Independently of the application, the performance of autonomous computer vision systems depends on the sensors adopted to gather data, on the software to exploit them,

and on the computation hardware to run these complex codes. With reference to the visual sensor, a classical partition differentiates active instruments (LIDAR, time-of-flight cameras) from passive ones, bounded to make use of external light. Aside from the source of the radiation, a very important asset of active sensors is their capability to detect the evaluate the depth in the captured image, allowing for a 3-D vision. Such an advantage is usually paid in terms of cost and complexity, in addition to mass/volume/power requirements which are sometimes difficult to match with the characteristics of spacecraft. It is not the case of simple, monocular passive viewer, which are instead limited to a 2-D representation. Stereoscopic vision, by combining two images captured from different points of view, has the capabilities to provide information about the depth without increasing too much (or even at all) the issues about sensors. On the other way, a large role is left to the data processing, as the features appearing in the two images captured at the same time need to be identified and coupled in order to provide 3D measurements. However, such a significant limit becomes less rigid in view of the continuing advances (both hardware and software) in computer science.

It is reasonable to claim that stereoscopic vision will have an important role among computer vision techniques for space applications. Impressive advances in real time processing of the image pairs are expected, also – and above all – driven by terrestrial automotive market. As a result, overall architecture advantages of the stereo viewer with respect to other 3D capable solutions are deemed to dominate, especially for small spacecraft (either small rovers or satellites equipped with manipulators).

Indeed, this chapter is aimed to discuss possible applications of the stereoscopic vision. Next paragraph will recall main concepts for this technique. Then, some remarks about use of stereo-vision in manoeuvring space manipulators are presented. Later, considerations on stereo-vision applied to the navigation of a vehicle are reported. Finally, a real-world example is detailed, referring to the small rover RAGNO which main characteristics will be reported in next chapter. The solution for the guidance-

navigation-control loop of this rover, based on a stereo viewer accommodated on board and able to detect obstacles and compute a path avoiding them is presented. Quantitative data from numerical simulations and from road tests are reported, with the aim to show – even within the limited scope of a lab project – the main characteristics of a stereo-vision system.

1.2 Stereoscopic vision concept

An ideal stereoscopic viewer is composed of two identical cameras with parallel optical axes (Figure 1-1). This set-up allows to observe the scene from two points of view and to compute the optical depth of a detected point of the scene (P) by means of a simple triangulation between P and its projections on the right- (P_R) and the left-camera (P_L) image planes. P_L and P_R (which identification in the images and association with P is for now assumed) are also defined as corresponding points.

Once selected a reference frame rigid to the left camera, it is possible to express the triangulation relations considering both cameras like pinholes [5]. For a pinhole camera (Figure 1-2) the relation between a point P belonging to the scene and the relevant image point P_L can be found through the similarity criterion for triangles, stating that two triangles are similar if they have two proportional sides and the angle between these sides equal.

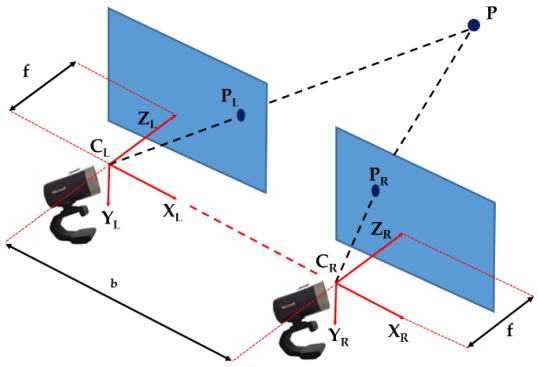


Figure 1-1. Stereoscopic vision system arrangement

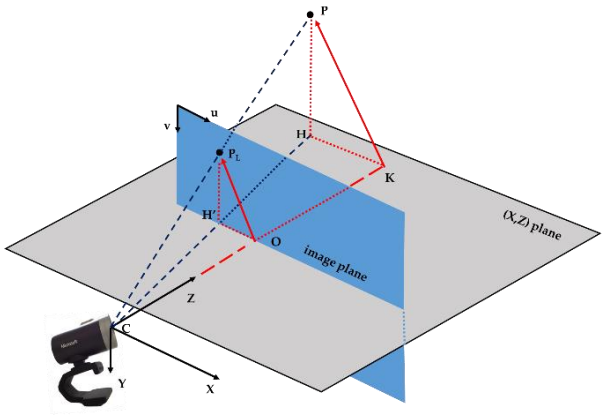


Figure 1-2: Pinhole camera and similar triangles; X, Y, Z are the coordinates of the captured point P in a frame centered in the camera, u, v the corresponding coordinates of its projection in the image plane.

Applying this similarity criterion to the ⊿(CKH)-⊿(COH′) and ⊿(CHP)-⊿(CH′P′) pairs of triangles it is possible to write:

$$\begin{cases} u = f\dfrac{X}{Z} + p_u \\ v = f\dfrac{Y}{Z} + p_v \end{cases}$$

Eq. 1-1: Image coordinates of point P

Eq. 1-1 can be rewritten in matrix form by means of homogeneous coordinates:

$$Z\,P_L = Z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underline{\underline{K}}\,P = \begin{pmatrix} f & 0 & p_u \\ 0 & f & p_v \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Eq. 1-2: Homogeneous coordinates of point P

Eq. 1-2 is also labelled *perspective law* because it projects the point P from the reference frame characteristic to the scenario to the image reference frame. The $\underline{\underline{K}}$ matrix is defined as the intrinsic matrix of the camera and its elements can be determined through a calibration procedure [6].

The perspective law (Eq. 1-2) includes two equations with three unknowns (X, Y, Z), thus resulting unsolvable. This is the reason why it needs a second camera to obtain the optical depth of the point P. The application of the perspective law to the stereo viewer leads to the following relations:

$$\begin{cases} Z\,P_L = \underline{\underline{K}}_L P \\ Z\,P_R = \underline{\underline{K}}_R\ {}^L\underline{\underline{R}}_R^T\ (P - C_R^T) \end{cases}$$

Eq. 1-3: perspective law in left-camera frame

where R and C matrices take into account the relative position and orientation of the second camera. In the ideal case sketched in Figure 1-1, with the baseline between the cameras orthogonal to the optical axes, which are parallel, and a similar arrangement for image planes and resolution, it follows that the intrinsic matrices are equal and that the rotation matrix corresponds to the identity.

Indeed Eq. 1-3 becomes:

$$\begin{cases} Z\,P_L = \underline{\underline{K}}\ P \\ Z\,P_R = \underline{\underline{K}}\ (P - C_R^T) \end{cases}$$

to be expanded as:

$$\begin{cases} u_L = f\,\dfrac{X}{Z} + p_u \qquad v_L = f\,\dfrac{Y}{Z} + p_v \\ u_R = f\,\dfrac{X\text{-}b}{Z} + p_u \qquad v_R = f\,\dfrac{Y}{Z} + p_v \end{cases}$$

Eq. 1-4: Pixel coordinates of point P in left and right camera image plane

Since cameras are supposed identical, the second and fourth equations of Eq. 1-4 can be neglected. By subtracting the first and the third it is finally possible to obtain the third dimension, i.e. the optical depth Z,

$$Z = \frac{f\,b}{u_L - u_R} = \frac{f\,b}{d}$$

Eq. 1-5: Optical depth Z

which evaluation depends on the focal length, on the relative geometry between the cameras and on the parameter *d* called *pixel disparity*: the closer the scene point P, the higher the disparity.

Figure 1-4 shows an example of pixel disparity (*disparity map*) for the sample scenario depicted in Figure 1-3Figure 1-3, where there are two objects at different distances from the stereo viewer.

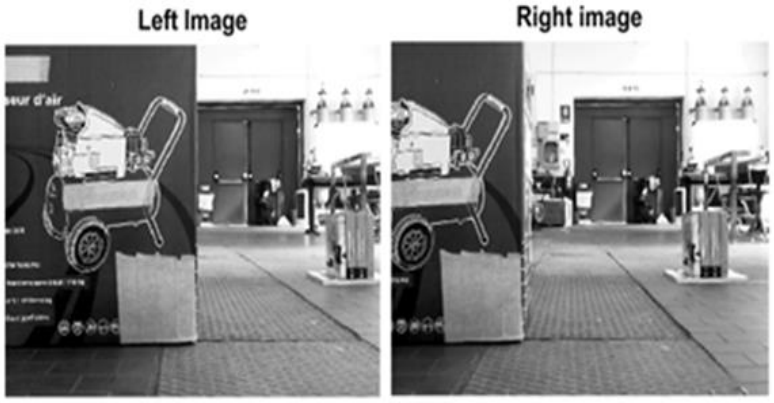


Figure 1-3: A sample of stereo images

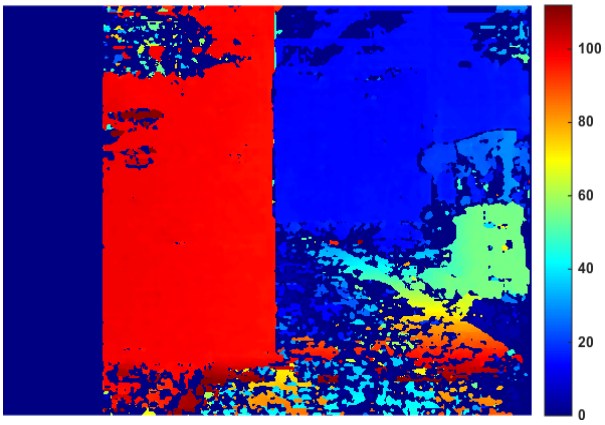


Figure 1-4: Disparity map for the scenario depicted in Figure 1-3

According to Eq. 1-5, it can be seen how the foreground box is characterized by a greater pixel disparity than the background objects. However, the raw disparity map shown in Figure 4 does not allow to detect the obstacles easily. In fact, there are many sparse color spots, for example those corresponding to the floor, which do not identify an existing obstacle. Therefore, a cleaning algorithm should be implemented to filter out this noise and to identify the true obstacles. This process could require a high computational time because the disparity map should be analyzed pixel by pixel. So, another approach – based on the object recognition techniques - has been chosen in this work in order to save the computational time.

1.3 The software architecture of the stereo-based navigation system

The architecture of the navigation software developed for obstacles detection and localization consists of three blocks:

- stereo image acquisition
- stereo image processing
- obstacles detection and localization in the (X, Z) motion plane

The resolution of the Eq. 1-5 requires the knowledge of the viewer's baseline and of the cameras focal length. The former is a geometric parameter and it can be chosen by the user while the latter is an intrinsic characteristic and it must be determined through a calibration procedure. The Zhang's method [6] has been adopted for the calibration of the stereo-viewer. This method allows to determine the intrinsic parameters of both cameras and their relative orientation by solving the perspective law for a set of points called markers. These markers belong to a predefined 2D calibration object and their (X, Y, Z) coordinates are known. The most used calibration object is a rectangular chessboard with squared cells of known dimensions (Figure 1-5).

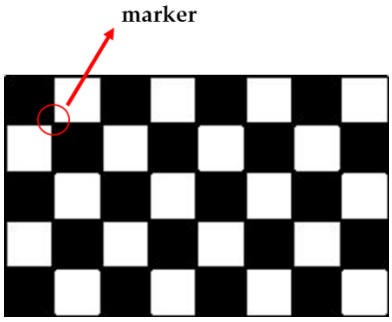


Figure 1-5: The calibrator object and the sample marker

The markers are identified in the vertices of each cell. By writing Eq. 1-2 in a reference frame rigid to the chessboard, the cameras orientation ($\underline{\underline{R}}$) and position ($\underline{t}$) with respect to this frame must be taken into account:

$$P_i = \lambda \underline{\underline{K}}\ \underline{\underline{R}}^T (P - \underline{t}) = \lambda \underline{\underline{K}} \left[\underline{\underline{R}}^T \mid -\underline{\underline{R}}^T \underline{t} \right] \begin{Bmatrix} P \\ 1 \end{Bmatrix} = \lambda \underline{\underline{K}} \left[\underline{\underline{R}}^T \mid -\underline{\underline{R}}^T \underline{t} \right] P_h = \underline{\underline{H}}\ P_h$$

Eq. 1-6: definition of the homography matrix

$\underline{\underline{H}}$ is a rectangular matrix (3x4) and is called *homography*. It is possible to make $\underline{\underline{H}}$ square by taking into account those markers having the same Z coordinate which can be set to zero without loss of generality. In this way, the previous equation reduces to the following:

$$P_i = \begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = \lambda \underline{\underline{K}} \left[r_1 \; r_2 \; r_3 \; -\underline{\underline{R}}^T \underline{t} \right] \begin{Bmatrix} X \\ Y \\ 0 \\ 1 \end{Bmatrix} = \lambda \underline{\underline{K}} \left[r_1 \; r_2 \; \underline{t}' \right] \begin{Bmatrix} X \\ Y \\ 1 \end{Bmatrix} = \left[h_1 \; h_2 \; h_3 \right] \begin{Bmatrix} X \\ Y \\ 1 \end{Bmatrix} = \underline{\underline{H}} \, P_h$$

The Zhang's method allows to determine the stereo parameters by calculating the homography $\underline{\underline{H}}$ through the knowledge of a set of (P_i, P_h) points pairs. In order to obtain an accurate calibration of the three axes of each camera it needs to acquire chessboard's images from different points of view. The results of Zhang's calibration procedure are the following:

$$\underline{\underline{K}}_L = \begin{bmatrix} 605.9893 \pm 1.1312 & 0 & 331.9116 \pm 0.5466 \\ 0 & 605.2356 \pm 1.1355 & 238.8710 \pm 0.6377 \\ 0 & 0 & 1 \end{bmatrix} \text{ pixel}$$

$$\underline{\underline{K}}_R = \begin{bmatrix} 603.0425 \pm 1.0914 & 0 & 316.9373 \pm 0.7559 \\ 0 & 602.4318 \pm 1.0880 & 231.9409 \pm 0.5975 \\ 0 & 0 & 1 \end{bmatrix} \text{ pixel}$$

$${}^L\underline{\underline{R}}_R = \begin{bmatrix} 0.9992 & 0.0066 & -0.0397 \\ -0.0066 & 1 & 0.0015 \\ 0.0397 & -0.0015 & 0.9992 \end{bmatrix}$$

$$\underline{t} = \begin{bmatrix} 148.5633 & 4.7982 & 4.6790 \end{bmatrix} \text{ mm}$$

The above results show that the principal points (i.e. the center of projection) of the webcams do not correspond with the center of the respective CCD sensor. The values of the rotation matrix and the translation vector are almost similar to the expected ones. In fact, the stereo viewer has been built with a baseline of 15 cm while the two cameras are aligned along the X axis.

1.3.1 Stereo image acquisition

The built in Matlab® USB webcam package allows the user to define the webcam object and to set its properties like the image resolution and the value of the focus. For the stereo viewer used, the standard resolution 640x480 and the autofocus mode have been chosen. The acquisition of the stereo image generates two 3D matrices which dimensions are 480x640x3. The third dimension refers to the number of colors used to define the single image i.e. the RGB three-color. Each element of the image matrix is defined as an 8-bit integer, so it assumes a value in the [0-255] range. As the processing phase requires a greyscale image, the following conversion is needed:

$$I = 0.2989\ R + 0.5870\ G + 0.1140\ B$$

1.3.2 Stereo image processing

The resolution of the triangulation problem Eq. 1-5 assumes a previous knowledge of the corresponding points' coordinates (P_L, P_R). The identification and matching of these points' pairs represents the main and most computationally expensive step of the whole navigation algorithm, since it needs a significant stereo image processing. In recent years, thanks to the constant improvement in computing power and software skills, several matching techniques have been developed with the goal to reduce the required computational time and to improve the robustness of the algorithms.

The typical process can be divided into two different phases:

- research, extraction and characterization of particular points, called features, in both images
- comparison of left and right extracted features and matching of the corresponding pairs

A feature can be defined as a small region of the image characterized by one or more repeatable properties. There are different types of features depending on the researched properties:

- *edge*: it is a point of the image characterized by a discontinuity in the brightness gradient. This feature allows to identify the edges of a scene's object
- *corner*: it is a point identified by the intersection of two edges where the gradient has a significant curvature. This feature allows to identify the corner of a scene's object
- *blob*: it is a region of the image characterized by peculiar properties (e.g. brightness) with respect to the surrounding area.

Every kind of feature can be extracted by means of different detection algorithms, while each extracted point needs to be coupled to a descriptor vector with these properties:

- *distinguishability*: the descriptor has to make the generic feature distinguishable from the others. The level of distinguishability depends on the dimension of the descriptor
- *robustness*: the descriptor must be detectable in any lighting conditions and regardless of photometric distortions or noise

The choice of the descriptor depends on the goal of the image processing. For the purpose of obstacles identification, it needs a combination of detector-descriptor which does not require high computational time. After an accurate analysis of the existent extraction algorithms, the SURF (*Speeded Up Robust Feature*) and BRISK (*Binary Robust Scale Invariant*) detectors have been chosen.

SURF [15] is a blob detector with an associated descriptor of 64 elements. The detection algorithm is a speeded-up version of the classic SIFT detector. The generic feature is extracted by analyzing the image in the scale space and by applying sequential box filters of variable dimension. The features are then localized in the point of the image where the determinant of the Hessian matrix H(x,y) is maximum. The descriptor vector is than determined by calculating the Haar wavelet [19] response in a sampling pattern centered in the feature.

BRISK [16] is a corner detector with an associated binary descriptor of 512 bit. The generic feature is identified as the brightest point in a sampling circular area of N pixels while the descriptor vector is calculated by computing the brightness gradient of each of the N(N-1)/2 pairs of sampling points.

Once left and right features have been extracted, their descriptors are compared in order to determine the corresponding points pairs. The matching criterion consists in seeking for the two descriptors for which their relative distance is minimum. This distance corresponds to the Euclidean norm for SURF case and to the Hamming distance for the BRISK one. The latter distance is intended as the minimum number of substitutions needed to make two binary strings equal. The matching process seems to require a very high computational time because each left feature should be compared with all right ones. This may be true for object recognition processes where the algorithm has to identify a particular object by comparing its acquired image with a database of images. Instead, in the case of stereovision applied for obstacles detection, the computational time can be reduced by taking into account the theory of the epipolar geometry. In fact, it states that there exists a geometric constrain between the left and right projection of the scene point P. As a consequence, the space where a matching feature has to be researched reduces to a portion of the image.

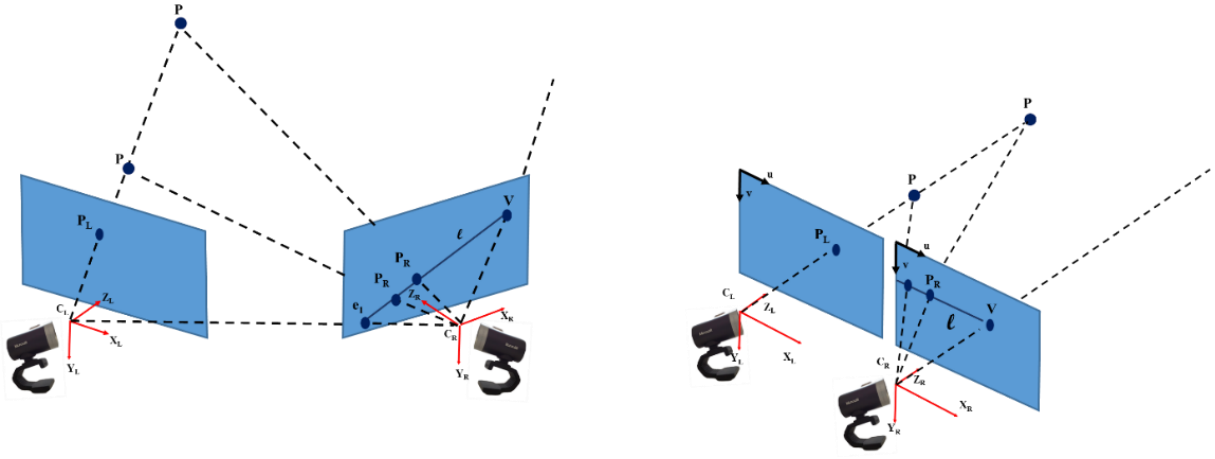


Figure 1-6: Epipolar geometry for real and ideal stereo viewer (in the ideal setup the epipolar point is at infinity because the sheaf of epipolar lines is non-regular. lines are in fact parallel).

As shown in Figure 1-6, for a fixed P_L the corresponding P_R lies on the line *l*, called epipolar, whose inclination depends on the geometry of the viewer. The position of P_R varies depending on the depth of the scene point P. In the case of a real stereo viewer, the epipolar line is delimited by the *epipole point* e_1 and the *vanishing point V*. The first is the center of the regular sheaf of epipolar lines and corresponds to the projection of C_L on the right image plane while the second is the projection of a scene point P with infinite depth. For the ideal stereo viewer used, the epipolar lines are horizontal therefore two matching features will have roughly the same v-coordinate. The searching process can be limited to a subset of right image rows thus obtaining a time savings.

Figure 1-8 and Figure 1-9 show the results of the processing phase for the sample space-like scenario in Figure 1-7. The irregular shape together with the color non-homogeneity of the close object entail that more SURF features than BRISK have been detected. By comparing the results obtained with features extraction to those of disparity map, it can be note that the former does not require a post processing noise filtering because there are few outliers which can be cut out by mean of a selection operation. On the contrary, the obtained disparity map shows the obstacle shape very well but a lot of noisy values too. A post processing filtering is therefore mandatory in order to make the map easily exploitable.

Figure 1-7 Sampled stereo image (a) and disparity map (b)

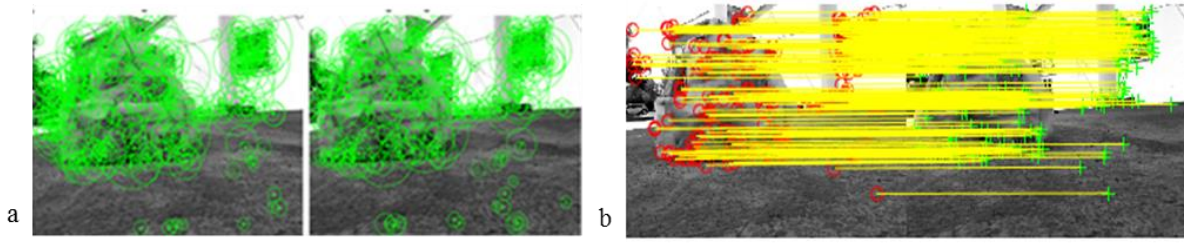


Figure 1-8 Left and right extracted (a) and matched (b) SURF features

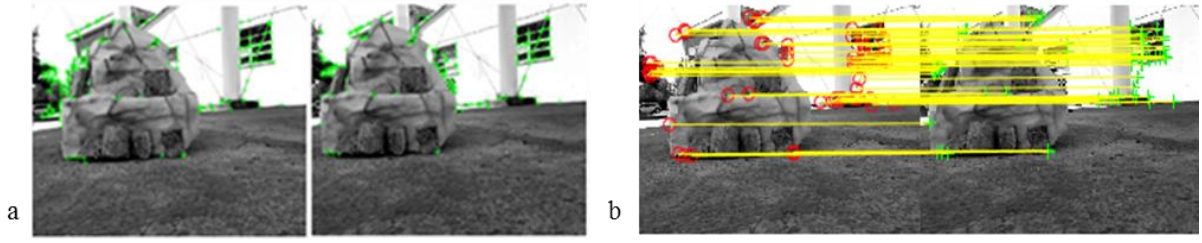


Figure 1-9 Left and right extracted (a) and matched (b) BRISK features

Regarding the computational effort, the results are summarized in Table 1-1 and Table 1-2. The onboard computer is a quadcore PC with a 2.4 GHz CPU and 8 GB RAM.

	Extracted (L/R)	Matched
BRISK	236/200	65
SURF	526/496	216

Table 1-1. Extracted and matched features

	Detection	Matching	Total
BRISK	238	79	317
SURF	141	79	220

Table 1-2. Computational time of detection and matching process (in ms)

1.3.3 Obstacles Detection and Localization

Once the viewer has been calibrated and the correspondences have been found, the triangulation equation Eq. 1-5 can be solved. As the mission is supposed in a 2D space, the triangulation results are reported in the (X,Z) rover reference frame. This frame has the same axes of the camera reference (but it is applied in the vehicle center of gravity. In order to select only the most interesting points among the set of triangulated ones, a threshold value is imposed on the optical depth Z. All the points with $Z > 2m$ have been cut out thus saving only the foreground features. The criterion adopted for obstacles

identification is based on the evaluation of the features density. The (X,Z) plane is divided into square cells for each of which the density of contained features is calculated. A threshold value d_{TH} is then established. All the cells with a density greater than d_{TH} is marked as occupied by an obstacle. The threshold value depends on the discretization step: the higher it is and the lower the threshold has to be. Figure 1-10 shows the result of selection and identification process: background features have been cut out together with the low density ones.

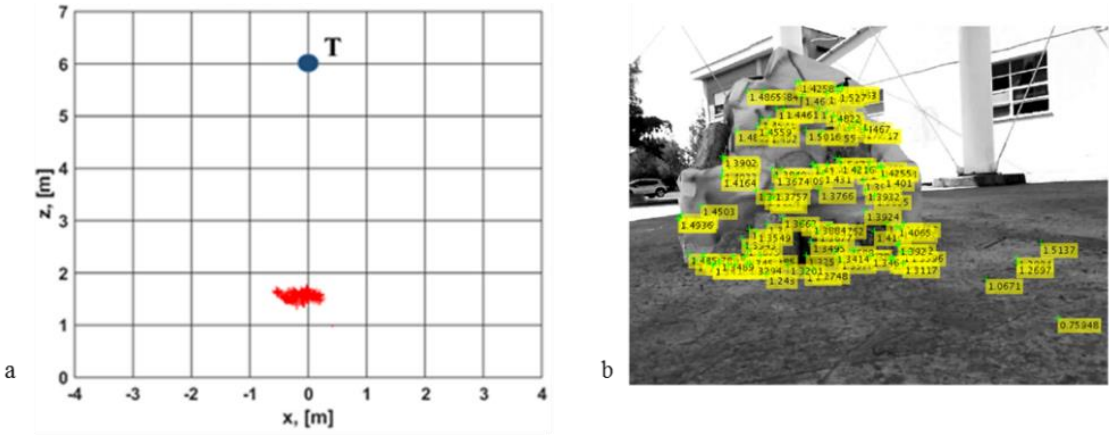


Figure 1-10: Obstacle localization in 50cm step discretized (X,Z) plane (left) and depth marks in camera image (right)

1.4 Stereo Vision applied to robotic manipulators

The goal to identify the information about depth is important in every scenario, including for sure the operational domain of a robotic manipulator. The application of stereovision to this specific field has been already considered as mature [7] for terrestrial, secluded environments, as the one typical to pick-and-place machines. In the industrial field the interest for stereoscopic vision deals with the limited costs and the reasonable complexity of the system. In fact, off-the-shelf imaging devices and easily designed software with standard hardware can successfully attain good performance.
A totally different assumption should be done with respect to space applications, as manipulation of captured/grasped spacecraft for in- situ servicing. These applications, while deeply in need of a 3-D vision system, have nevertheless two important, limiting characteristics:

- the scenario is not repetitive at all;
- the light conditions change (even suddenly) in time, due to occlusions occurring during the manoeuvre.

These two characteristics deeply affect the peculiar advantage of the stereo-vision systems as implemented in manufacturing industry today. However, the comparable active systems capable to intrinsically solve the 3-D issue, as LIDAR or time of flight cameras, turn out to be extremely expensive. Moreover, they can have significant limitations in terms of power/mass/volume requirements or in terms of performance, especially as far as it concerns the maximum attainable distance. As a result, stereoscopic vision, as a passive technique where much effort is left to the software/computation side (i.e. to an extremely fast advancing area) has a large interest also for space applications. Even more, it is maybe the only suitable solution for small platforms (Figure 1-11).

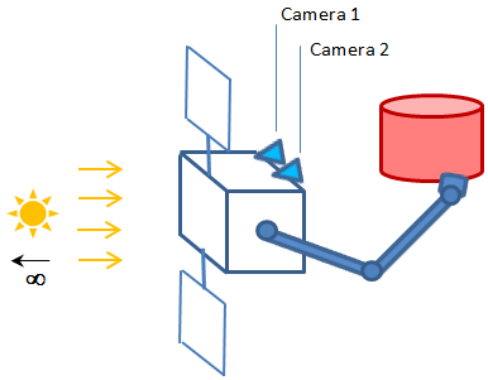


Figure 1-11: A sketch of a possible stereo-vision system accommodated on a small satellite help in-orbit servicing. The Sun recalls about sudden switches from shadow to full light conditions which affect image analysis.

The identification of the different features in the pictures captured by the two cameras to provide meaningful pairs (u_L, u_R) and (v_L, v_R) in Eq. 1-4 becomes a consuming task (these techniques will be discussed in the following of the thesis). In fact, the two recalled limits associated with the generic applications in space require a careful analysis of the two images, making use of advanced artificial intelligence tools. To ease such a process, it can be noticed that the positioning of the robotic arms' elements in the

scenario can be actually computed by using markers purposely located at the joints (Figure 1-12, [8], [9][10]).

Once the arms' configuration and the other features of interest, as the target in Figure 1-12 sketch, have been identified, the computation of the relevant coordinates can be carried out by means of the process expressed by Eqs.(1)-(6). Likely, the final goal for these applications is the evaluation of the commands to be given to the arms in order to execute the required operations. An interesting, complete example for such a task has been reported, even if in simpler laboratory scenario, in [9][10]. The idea is to draw on the images, by means of a geometrical approach, a reference frame which will provide the relative coordinates of the target with respect to the manipulator elements. This drawing obviously takes into account the fact that depth can be computed by stereoscopic technique only by comparing the images from two cameras. Then, the computed relative state of the elements allow to command the torques to be provided by the motors at the arms' joints.

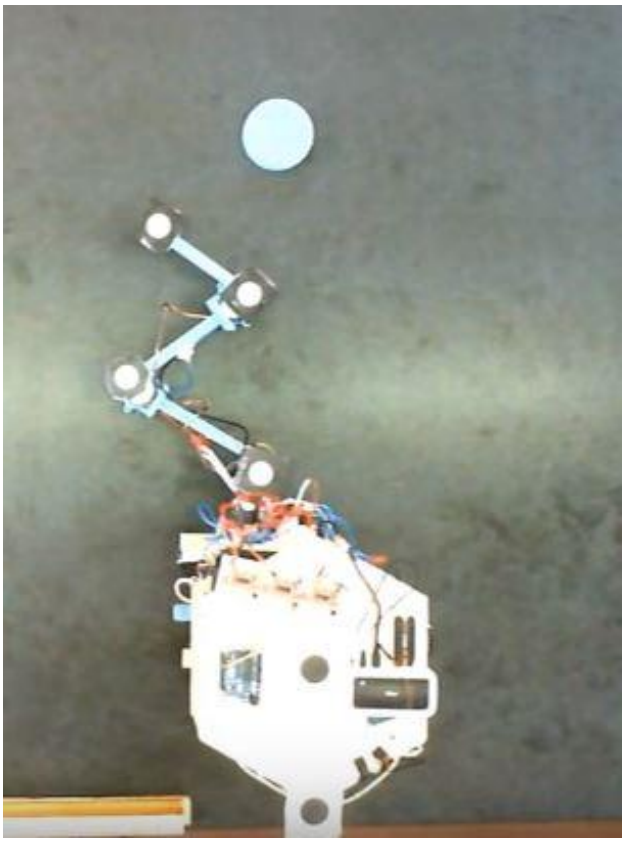


Figure 1-12: Positioning of space manipulators in a webcam image

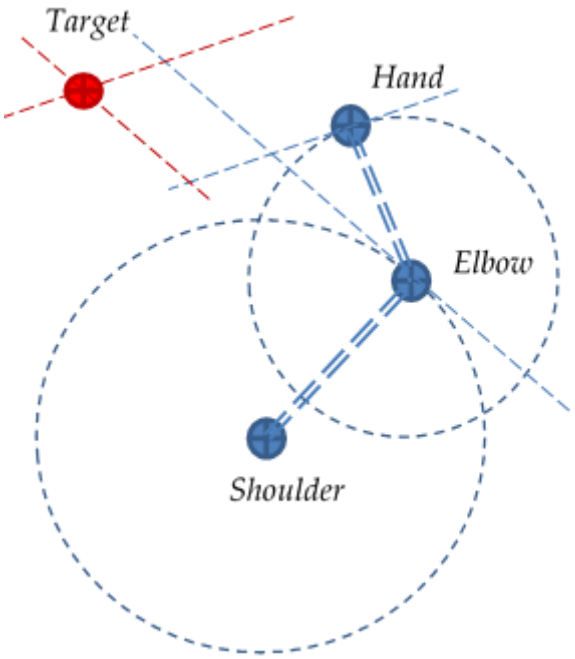


Figure 1-13: A sketch of the geometrical approach to draw a reference system to enable the computation of the distance to the target and the relevant commands for the manipulator. The drawing is performed in each single image to later reconstruct the 3D scene by comparison between two cameras [9][10]. Links of the manipulator are represented as dashed lines as they are actually projections, lacking the third dimension, and therefore not in scale with the real elements

Aside from the arrangement depicted in Figure 1-11, there is the opportunity to exploit the capability to accommodate the cameras on the arms of the manipulator. This solution has been recently proposed in a test to accurately compute the kinematics of the arms [11], practically offering a back-up of the encoders. Such a configuration, in spite of the possible need for an iteration of the calibration process, can offer distinctive advantages, as it allows to better capture close scenarios, being also more robust to occlusions. Basically, this arrangement generates a shift from the classical *hand-in-the-eye* configuration to the more flexible *eye-in-the-hand* one [12], that turns out to be convenient in some specific operation. Furthermore, standing the typical length of the links of the space robotic arms, this arrangement can offer a significant increase in the baseline between a camera located on the bus and the other located at the end effector. In such a case, previous relation Eq. 1-3 should be modified as

$$\begin{cases} Z\,P_L = \underline{\underline{K_L}}\,P \\ Z\,P_R = \underline{\underline{K_R}}\left(P\text{-}C^{*T}_R\right) \end{cases}$$

where the intrinsic matrices of the camera (K) become different as different could be their focal length, and the vector C* can easily have all elements different from zero. In fact, the angle β in Fig. 8 is not constrained to be equal to 90° as in typical applications (and in human vision) where the optical axis is orthogonal to the baseline.

According to Eq. 1-5, the increase in baseline grants a high disparity for the same focal length and optical depth, or – in other words - an extension of the maximum suitable range for a given resolution of the camera sensor. Such an advantage ca be extremely important when dealing with proximity navigation [13], i.e. in the preliminary phases of the rendezvous leading to in-orbit servicing manoeuvres (Figure 1-14).

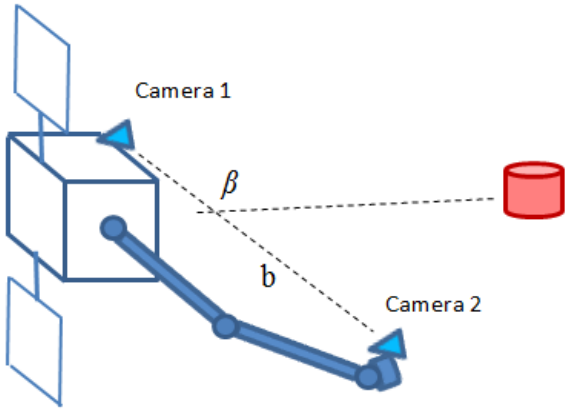


Figure 1-14: A sketch of the geometrical approach to enable the computation of the distance to the target and the relevant commands for the manipulator

Finally, we can claim that computer vision techniques are expected to be the workhorse of future autonomous space missions. Among them, stereoscopic vision is considered a very interesting option, as requirements easily fit the volume, mass, power limitations of typical space vehicles. An outline of the stereoscopic vision technique has been presented, with likewise applications in the fields of the space manipulators and of the planetary exploration. Specifically, the case of a rover has been carried out in detail, including the design of the vision-based navigation system, its implementation on board a small terrestrial autonomous vehicle, and the test campaign.

Chapter 2.

A Vehicle Prototype and its Guidance System

2.1 Introduction

Planetary exploration through mobile laboratory platforms has always been one of the most important topic of space engineering. The first examples are the American lunar vehicles and the Russian remotely control Lunokhod 1-2. Since the early 90s the space agencies focus on the exploration of Mars. NASA was the first to land a platform on the Red planet with the Sojourner (1996, Mars Path Finder). Till today other three rovers have landed on Mars: Spirit and Opportunity (2004, Mars Exploration Rover) and Curiosity (2011, Mars Science Laboratory). These platforms are the unique examples of autonomous mobile robots designed for planetary exploration. Autonomy has been accomplished by integrating a panoramic binocular vision system (NavCams). NavCams1 software is able to identify and localize close obstacles by processing the acquired stereo image. This navigation system has been improved in Curiosity, where other four stereo pairs (Hazard avoidance cams) has been added in order to obtain a wider vision of the surrounding environment. Future Mars platforms like NASA's Mars 2020 and ESA's ExoMars 2018 are going to integrate a further improvement of the stereo-based navigation system. The reasons of the great success of the stereoscopic vision are multiple. At first, stereo vision is an attractive technology for rover navigation because is passive, i.e. sunlight provides all the energy needed for daylight operations. The second reason is that only a small amount of power is required as for the cameras electronics as to obtain knowledge about the environment. In addition, by installing enough cameras or cameras with a wide field of view, no moving parts are required to capture the 360° surrounding environment. Having fewer motors reduces the number of components that could fail. Thanks to the constant improvements in computer vision techniques, hardware performances and optical sensors quality, future space missions

are likely to exploit stereo-based systems for applications aside from the rover navigation. Typical examples are the proximity maneuvers, like rendez-vous or docking, but also the more general in-orbit servicing like autonomous robotics arm grasping and manipulation of objects with unknown shapes.

This chapter deals with the application of the stereoscopic vision to rover navigation in order to make the platform totally autonomous. The design and development phases are explained in details together with the results obtained during a test session. In particular, Section 2.2 introduces the rover RAGNO used in his work: its starting and current state of the art is described in details. Section 2.3 is focused on the guidance strategy needed to plan a safe path leading the rover toward the final goal position. Section 2.4 shows the results obtained during the test campaign while the last Section 2.5 summarizes the main points of this work and suggests future applications of both autonomous vehicles and stereovision.

2.2 The rover RAGNO

RAGNO [28], standing for *Rover for Autonomous Ground Observation and Navigation*, is a four wheels small rover originally designed in 2011 at the Guidance and Navigation Lab, Sapienza Università di Roma in order to familiarize with robotics platforms, multibody systems and remote control. RAGNO can be defined as a modular vehicle because different subsystems, for example a robotic arm, can be allocated onboard according to the mission needs.

In the present project RAGNO has been equipped with a stereo viewer and a sensors platform containing a triaxial gyroscope and a triaxial magnetometer. This hardware integration allows to convert the rover from a remote-control system (Figure 2-1a) to an autonomous one (Figure 2-1b). The reading of sensors measures and the control of the angular velocity of the four wheels are realized by two Arduino shields. All the data are then sent to the guidance and navigation Matlab software through a serial communication.

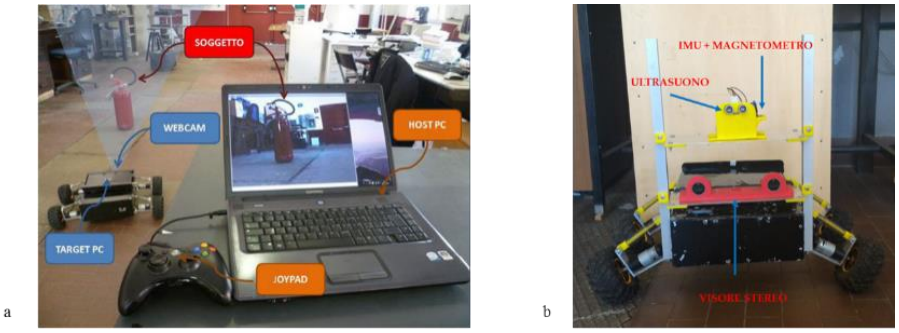


Figure 2-1: Initial (a) and final (b) RAGNO state of the art

RAGNO

The rover RAGNO (Rover for Autonomous Guidance Navigation and Observation) developed at the *Guidance and Navigation Lab of the University of Rome, "La Sapienza"*, is a friendly multi-task mobile platform. RAGNO has been used in different studies concerning the remote control (Figure 2-2, left), the multibody dynamics when equipped with a robotic arm (Figure 2-2, right) and the operational autonomy through stereo-based navigation [14], which is the focus of this section.

Figure 2-2: Applications of the RAGNO rover

For such an application, with the goal to reach a target while avoiding the obstacles in the scene, RAGNO is equipped with a commercial stereo viewer, a three axes magnetometer, a gyroscope and a central processing unit (Figure 2-3). The stereo viewer is essential to understand a 3D scenario and autonomously detect and localize the obstacles to define an optimum and safe path towards the target. Notice that the stereo viewer requires a calibration [6] before the mission begins to attain the desired accuracy. The gyro and the magnetometer are instead needed for the control section to reconstruct RAGNO true path by means of a Kálmán filter implementation. The comparison

between the desired and the Kálmán estimated paths generates the commands for the four-wheels motors to correct tracking errors by applying a speed control law.

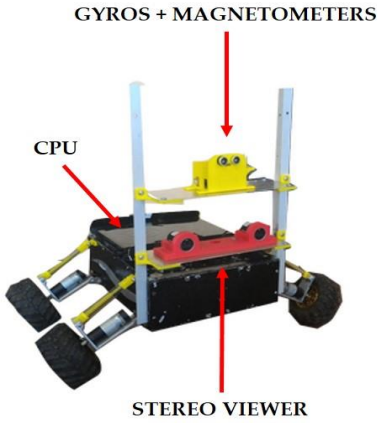


Figure 2-3: The RAGNO rover as equipped for the project

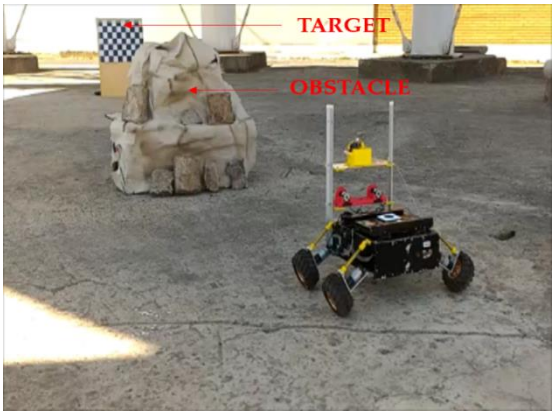


Figure 2-4: Sample outdoor scenario

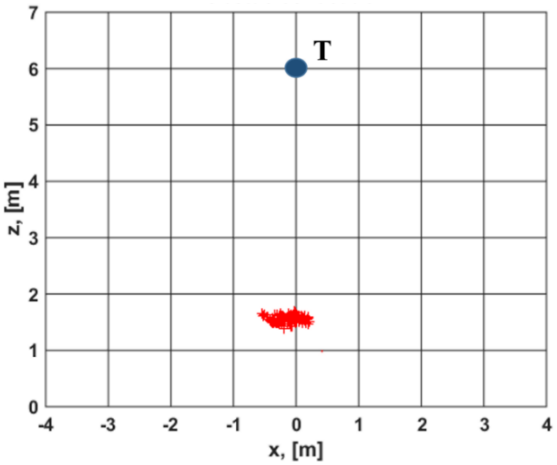


Figure 2-5: The location of the obstacle resulting from the use of the stereo vision system is depicted in a map to help the identification of a convenient path from the current position (0,0) to the target T.

By applying Eq. 1-5 to matched points' pairs it is possible to obtain the 2D map (of the plane containing the rover motion) depicted in Figure 2-5. In order to plan a safe path toward the target position T, a guidance law needs to be implemented. A suitable solution is represented by a graph-based guidance law [14], which allows the rover to reach the target moving on the shortest path. However, tracking errors easily occur while following this desired optimal path. By means of the additional sensors (gyro and magnetometer) it is possible to feed with relevant measurements a Kálmán filter built on a simple dynamic model, and to obtain an accurate estimate of the actual path. At this stage, the control section commands the wheel motors to ensure the tracking of the correct path.

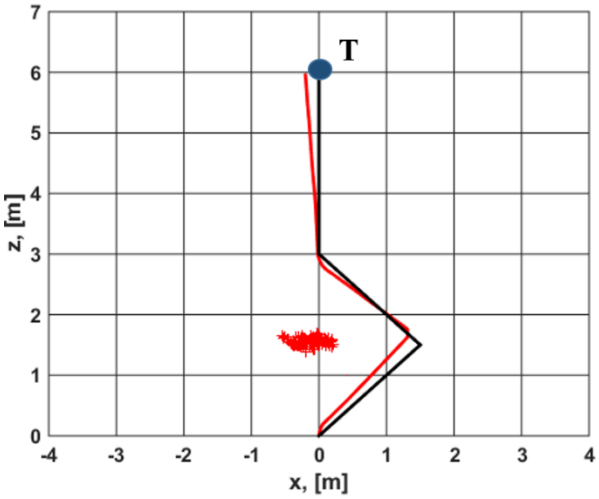


Figure 2-6: Planned and Kálmán estimated path

The findings from path planning and tracking tasks are shown in Figure 2-6 above: the black line indicates the desired path while the red one the estimate of the actual trajectory. The guidance algorithm requires about 400 ms to compute the shortest path, while the whole guidance and navigation algorithm time requests is in the order of 1 s, a value deemed satisfactory for a space rover moving at limited speed in a static environment.

2.3 The guidance system

The output of the navigation system, i.e. the coordinates of the localized obstacles, becomes one of the inputs of the guidance system. The purpose of the designed guidance algorithm is to plan a safe path that brings the rover from its starting position toward the target. Different guidance strategies can be found in literature [27]. One of the most used solutions in space-related problems like rendez-vous, descent and landing is based on the Lyapunov's stability theory. This approach provides to define a custom function, the artificial potential V, which describes the interaction between the body and the surrounding environment. The motion of the RAGNO platform can be compared to a sample positive charged particle in an electric field generated by other particles: the sample particle is attracted by the negative (the target) and repulsed by the positives (the obstacles). In the electric analogy, the total action exerted by the charges is the Coulomb's force and it corresponds to the gradient of the potential function V. For the purpose of rover path planning, as the problem is kinematic, the following potential function of the platform velocity has been elaborated:

$$V = \begin{cases} V_{att} + V_{rep} = \frac{1}{2}\varepsilon\rho_t^2 + \sum_{j=1}^{\#obs}\frac{1}{2}\eta\left(\frac{1}{\rho_{obs,j}} - \frac{1}{\rho_0}\right)^2\rho_t^2 & \rightarrow \ \rho_{obs,j} \le \rho_0 \\ V_{att} = \frac{1}{2}\varepsilon\rho_t^2 & \rightarrow \ \rho_{obs,j} > \rho_0 \end{cases}$$

Eq. 2-1: Expression of the potential function

The potential is a non linear function of rover distance from the obstacles ($\rho_{obs,j}$) and the target (ρ_t), and has a global minimum at the goal position. The j-th repulsive action vanishes when rover-obstacle distance is larger than the ρ_0 threshold; ε and η parameters can be modified in order to tune the magnitude of the two components. The integration of the gradient ∇V allows to obtain the safe trajectory that the rover has to follow.

Figure 2-7a shows the results of a simulation of a simple scenario. The red obstacle is included in the safety blue area while the three curves refer to different values of η and

ε. The legend reports the computational time needed for each simulation. It can be noted that the greater is η/ε, the stronger is the repulsive component of potential V, so that resulting trajectory is longer and safer.

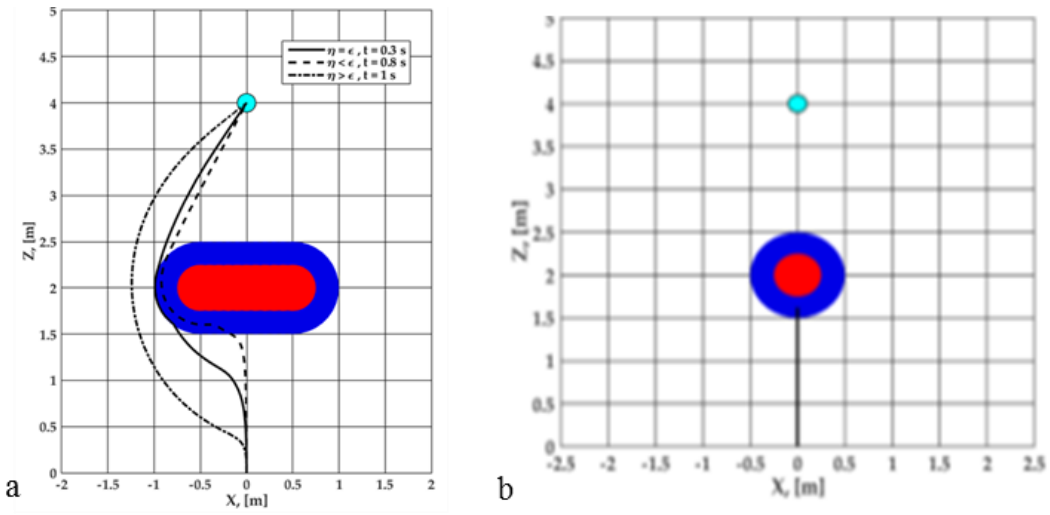


Figure 2-7: Trajectories obtain with potential function (left) and example of local minimum problem (right)

This guidance strategy allows the user to select the preferred function V, while necessarily including the drawback if possible local minima ending up as traps. Indeed, depending on the position of the obstacles, the integration of the gradient function can generate an oscillating solution around wrong minima points (an example is shown in Figure 2-7b).

An alternative solution to the potential guidance has been proposed in the frame of this project [28]. It is a customized version of the A-star search algorithm that is widely used in path finding process. Once (X,Z) plane has been discretized in square cells, the path can be seen as sequence of straight lines that connect the starting to the goal position passing through the centers (nodes) of selected and adjacent cells. The line that links two adjacent nodes identifies a graph. The generic displacement is determined by minimizing the following cost function:

$$J = d_N(n_0, n_{opt}) + d_E(n_{opt}, n_{tgt})$$

Eq. 2-2: The cost function J

where $d_N(n_0, n_{opt})$ denotes the nodal distance between the starting node n_0 and the candidate optimal node n_{opt} while $d_E(n_{opt}, n_{tgt})$ is the Euclidean distance between the candidate optimal node and the target one n_{tgt}. The algorithm defines two lists of nodes. The first one, named OPEN, is the list within which the optimal node is sought, including all the nodes reachable by the current candidate. Once the optimal node has been extracted by minimizing Eq. 2-2, the list is not deleted but it is updated with the adjacent nodes of the next candidate. The extracted node is deleted by the OPEN and it is inserted into the second list, called CLOSED. The algorithm ends when the target node appears in the OPEN list. The constant update of this list implies that more than a path is assembled during the optimization process. Figure 2-8.a shows this concept clearly. As the target (green) is aligned with the rover starting cell (cyan), the shortest path initially identified by the algorithm is the straight line between rover and target (light blue). After extracting the node #39, the presence of obstacle cells (red) erases the current path and the algorithm establishes node #16 as the optimal one. The OPEN list is updated with new nodes (grey) until the target node #94 appears. Figure 2-8b shows the set of complete paths identified by the algorithm, all of them optimal because they have the same length. In this case, an additional parameter, i.e. the number of rotations that RAGNO has to actuate in order to follow an optimal trajectory, is taken into account. The optimal path is now defined as the one with the minimum length and the minimum number of rotations required and, as a result, the light brown trajectory is extracted.

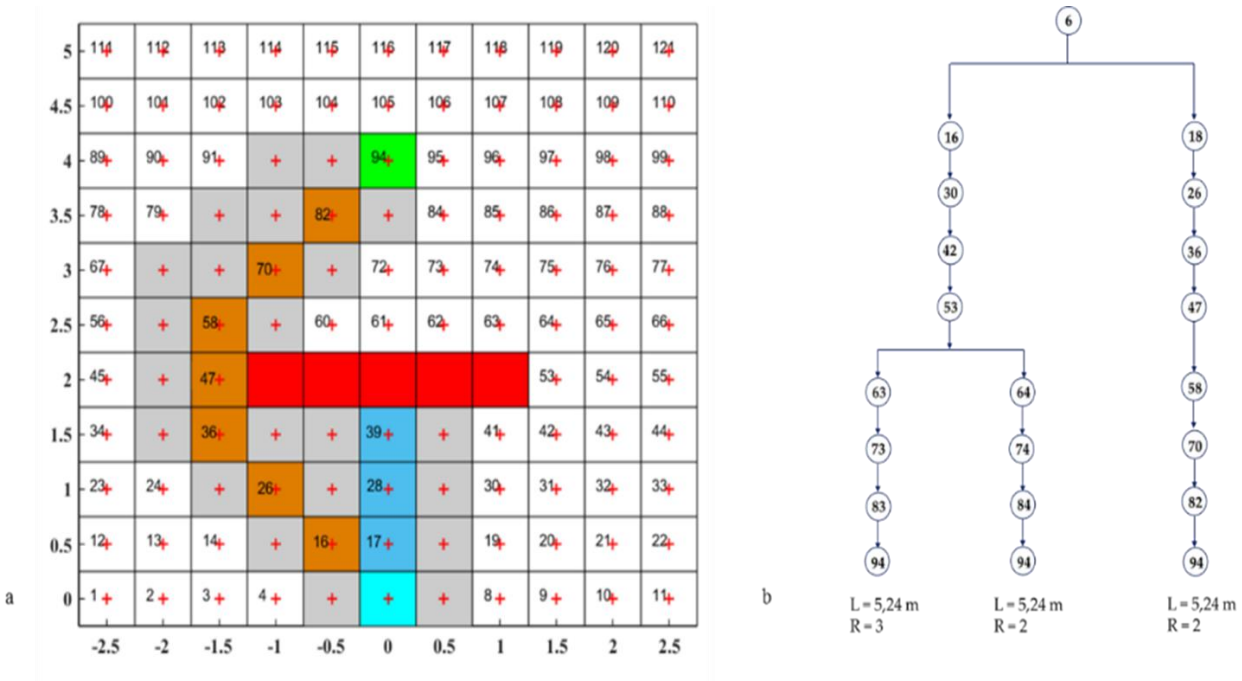


Figure 2-8: A-star path planned (a) and paths tree generated during the research (b)

The computational time of this algorithm is about 300 ms. This time is affected by the complexity of the scenario. In fact, the greater are the obstacles cells and the lower are the cells to be examined thus obtaining a useful time savings. The opposite behaviour is shown by the potential strategy where the presence of more than an obstacle affects as the repulsive potential (there are more repulsive terms) as the integration process. Another advantage of the graph-based strategy is that local minima disappear as the minimization of the cost function Eq. 2-2 does not require an integration operation. In addition, the A-star trajectory is easier to follow than the one computed by artificial potential approach because the rover has to actuate only two movements i.e. a pivoting or a small forward displacement. The required control effort is smaller and more suitable for RAGNO onboard actuation system.

2.4 Test sessions

Performance of the designed guidance and navigation system in terms of autonomy and robustness has been evaluated during outdoor test sessions. Two scenarios have been taken into account: the first (Figure 2-9) is given by a single obstacle in front of the

rover while the second (Figure 2-10) presents a hidden obstacle behind the foreground rock, thus resulting not visible in the first stereo image acquisition.

In order to verify the effective tracking of the planned path the current trajectory must be reconstructed by mean of an estimation process. In order to estimate the position of RAGNO step by step, both hardware and software integration is needed. The hardware includes the following onboard sensors:

- *wheels incremental encoders*, measuring the rotation angle of the wheels. If the sample rate is high, the angular velocity can be obtained by numeric derivative of two successive measures. As a consequence, the linear velocity and the yaw rate can be calculated too
- *triaxial gyroscope*, measuring the angular velocity about three body axes (yaw, pitch, roll). In this case the yaw rate is the most significative measure because it describes the rotational dynamics about Y-axis of RAGNO. A change in yaw rate implies a change in heading angle. Gyro measures are affected by bias error so a calibration process has been carried out in order to filter it out. Calibrated measures allow to calculate the yaw angle evolution through a numeric integration of two successive samples
- *triaxial magnetometer*, measuring the total magnetic field i.e. the sum of environmental and platform generated field. Since the magnitude decreases as r^{-3} where r is the distance from the source the latter term can be filtered out by installing the sensor far from the field source (i.e. electric motors and onboard computer). In the case of the performed test sessions, Earth field is altered by different kinds of time variant perturbations as the test site is located near to a high-speed railway, and a low confidence has been indeed assigned to the magnetometer. In addition, the raw magnetometer measurements are affected by bias and scale factor errors, so that a calibration procedure is mandatory.

The acquired measures constitute the inputs of the implemented estimation algorithm (Eq. 2-3). The linear Kàlmàn filter has been chosen since it provides real time state estimation in low computational time.

$$ {}^{e}\underline{X}_k = {}^{p}\underline{X}_k + \underline{\underline{K}}_k (\underline{z}_k - \underline{\underline{H}}\,{}^{p}\underline{X}_k) = {}^{p}\begin{Bmatrix} \psi \\ \dot{\psi} \\ V \end{Bmatrix}_k + \underline{\underline{K}}_k \left(\begin{Bmatrix} \psi_{MAGN} \\ \dot{\psi}_{GYRO} \\ \omega_{ENC}^{sx} \\ \omega_{ENC}^{dx} \end{Bmatrix}_k - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{L_{semiaxis}}{R_{wheel}} & \frac{1}{R_{wheel}} \\ 0 & -\frac{L_{semiaxis}}{R_{wheel}} & \frac{1}{R_{wheel}} \end{bmatrix} {}^{p}\begin{Bmatrix} \psi \\ \dot{\psi} \\ V \end{Bmatrix}_k \right) $$

Eq. 2-3: Estimation algorithm formula

The current state ${}^{e}\underline{X}_k$, consisting in yaw angle, angular rate and linear velocity, is estimated by summing the predicted state, i.e. the solution of the platform dynamics model, and the innovation term i.e. the difference between the current ($\underline{z}_k$) and expected ($\underline{\underline{H}}\,{}^{p}\underline{X}_k$) measures. The innovation term is weighed with the Kàlmàn gain matrix $\underline{\underline{K}}$. Since accelerations are not taken into account, the dynamics model is reduced to a simple kinematics one:

$$ \dot{\underline{X}} = \underline{\underline{A}}\underline{X} + \underline{\underline{B}}\underline{u} = 0_{3x3} \begin{Bmatrix} \psi \\ \dot{\psi} \\ V \end{Bmatrix} + \begin{bmatrix} \frac{R_{wheel}}{2L_{semiaxis}} & -\frac{R_{wheel}}{2L_{semiaxis}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \omega_{wheel,sx} \\ \omega_{wheel,dx} \end{Bmatrix} $$

Eq. 2-4: Kinematic model of the rover RAGNO

In order to obtain the (X,Z) position coordinates and the heading angle ψ, the following relations must be computed:

$$ \begin{Bmatrix} X_k \\ Z_k \\ \psi_k \end{Bmatrix} = \begin{Bmatrix} X_{k-1} + {}^{e}V_k \sin\left({}^{e}\psi_k\right) dt \\ Z_{k-1} + {}^{e}V_k \cos\left({}^{e}\psi_k\right) dt \\ {}^{e}\psi_k \end{Bmatrix} $$

Eq. 2-5: Kinematic state of the rover RAGNO

The results of Kàlmàn filtering are shown in Figure 2-9 and Figure 2-10 referring to successfully performed test. The areas represented by blue cells have been introduced all around the obstacle (after having identified its location) to add a safety zone.

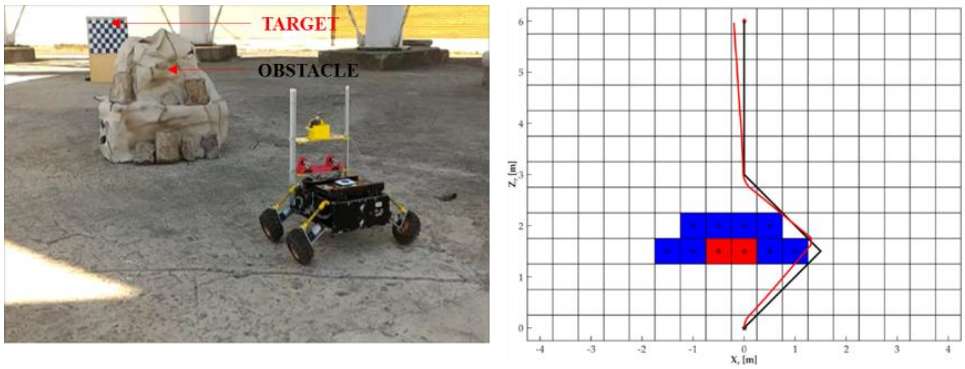


Figure 2-9:Planned (black) and estimated (red) trajectories of the first test scenario

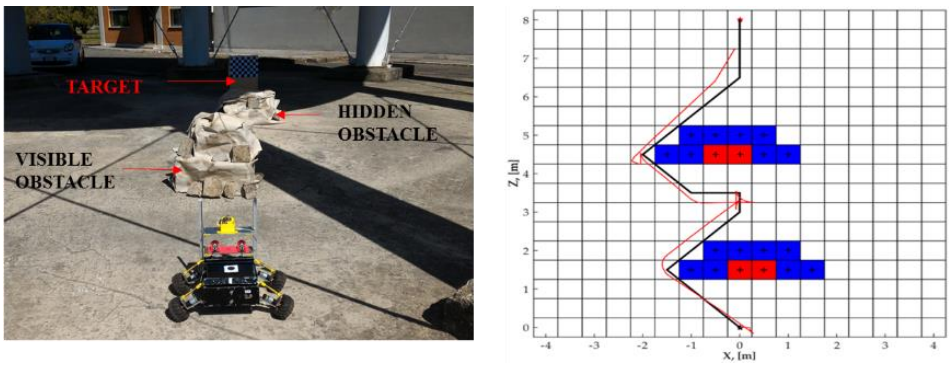


Figure 2-10: Planned (black) and estimated (red) trajectories of the second test scenario

In the last scenario, the hidden obstacle does not appear in the first stereo acquisition. The stereo image processing can not be performed when RAGNO is driving toward the goal because it needs consistent computational time. This means that it is necessary to integrate an auxiliary sensor which is able to detect a close object during the travel of the platform. An ultrasound sensor satisfies this need. As shown Figure 2-11 when the sensor detects a close object, it sends a stop bit to the onboard computer which orders the rover to arrest. The whole guidance and navigation process restarts and a new safe path is planned. An example of successful test session has been uploaded to the YouTube "GN Lab" channel (https://www.youtube.com/watch?v=XOt2iRUeDag).

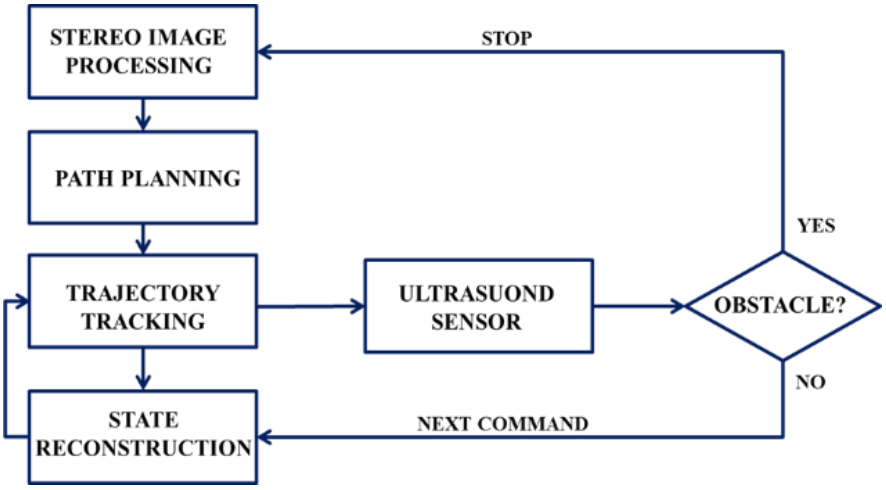


Figure 2-11: Mission block scheme

2.5 Conclusions and further developments

The theme of autonomous robots is becoming more and more prevailing both in space and civil sector due to the wide range of applications. This growth is supported by the constant improvements in computer hardware, sensors technology and computer software. In addition, the recent developments in the field of computer vision and artificial intelligent allow to integrate and implement human-like systems and logics. In this paper an example of this application has been presented: the RAGNO, low-cost prototype of an autonomous vehicle has been proofed to be able to accomplish a transfer mission while avoiding the obstacles in the surrounding environment. The stereoscopic vision together with the processing algorithm are inspired by the human visual system and brain's perception method. The advantages and drawbacks of this navigation system have been discussed in details.

Low-cost, simple and reliable autonomous rovers can change planetary exploration scenario. Till now, missions have been characterized by rovers working uncooperatively in specifically assigned and limited areas. The availability of low-cost autonomous platforms allows to afford missions involving the use of a swarm of cooperative rovers targeting – at the same time - the same celestial bodies. The logic of cooperation is inspired by nature swarm (birds, fishes or ants) and it is based on the fact that each member shares a set of information that might help the others in their mission.

The advantages of this exploration approach are clear: parallel missions collect, without overlapping, a greater amount of coordinated data; at the same time, versatility and robustness to failure are largely increased.

Chapter 3.

Examples of Possible Missions and Applications

3.1 Introduction

Automated vehicles have significantly improved their performance and are nowadays able to easily move in difficult scenario. These characteristics are shared by extreme, state-of-the-art vehicles as the rovers built for planetary exploration [30] as well to systems for surveillance, as the aerial drones adopted by several air forces [31]. Even more important, automated vehicles are now available at a limited operational cost. This is the reason to consider them as the ideal platforms for systems devoted to continuous monitoring, matching current needs for the assessment of environmental conditions in large areas. Possible applications include the characterization of hazard sites and distressed areas, as well as the analysis of the pollution in urban areas. In these applications the build-up of a network of static sensors can be expensive due to the number of devices requested to provide a significant resolution. Instead fleets of mobile platforms can easily collect a large set of data with the required coverage in space and time and the resolution associated with specific analysis [32]. These fleets are intrinsically versatile, being able to be deployed at need in different scenarios through minimal adaptations. The advances in sensors and computational power allow to navigate even quite complicated scenes, while the miniaturization of the sensors ensure that collected data will be significant as measured and elaborated by small labs. Not to say that the autonomy saves for the cost of human crews, with only supervisory tasks left to the control center, and a 24/7 operational availability.

In a recent paper [33], it has been proposed to use wheeled autonomous vehicles to monitor atmospheric pollutants. These rovers allow for long-time observations over relatively large areas, provided that the mobile platforms could perform the relevant guidance, navigation and control in a robust and reliable way. Global Positioning

System (GPS) and an Inertial Measurement Unit (IMU) were used to determine the platform position, while a stereovision system was implemented in order to detect and avoid possible obstacles. The capability to inspect an even wider area to detect possible sources of dangerous substances can be increased by including an aerial drone in the architecture. The drone itself can be autonomous or remotely controlled by a human operator. Its favorable point of view can be exploited to communicate the coordinates of hidden points of interest to the rover, which actually hosts the sensing device.

The rover RAGNO, designed and built at the Guidance and Navigation Lab of the Scuola di Ingegneria Aerospaziale in Sapienza Università di Roma, and already described in the previous chapter, is the autonomous terrestrial vehicle used for the tests.

The gas sensing device accommodated onboard he rover has been developed within the PRACTICE (Planning Re-thinked Ageing Cities Through Innovative Cellular Environments) project, in the frame of a cooperation between Sapienza Università di Rome and the Royal Institute of Technology of Stockholm (KTH).

The sensor adopted to characterize atmospheric pollution derives from a prototype designed to monitor indoor air quality, indeed called HOPES (Home Pollution Embedded System). Equipped with light and sound alarms to warn about significant risky measurements, can be also linked to the web as per the Internet of Things approach. While the sensor's conception and manufacturing is clearly out of this thesis work, and not even a contribution of the author, some more details about useful to correctly understand the application have been reported in the following.

Sensor development

The improved model developed for the current project uses *Figaro* sensors [34] due to their low dimensions and their capability of measuring different gases. The sensor is capable to measure gases as toluene (C_7H_8), ethanol (CH_3CH_2OH), ammonia (NH_3), methane (CH_4), isobutane (C_4H_{10}), propane ($CH_3CH_2CH_3$), carbon monoxide (CO),

methyl-mercaptan (CH_3SH) hydrogen sulfide(SH_3) and trimethylamine $N(CH_3)_3$ as well as dangerous pollutants PM 10 and PM 2.5. To limit the cost of the hardware a common ARM instruction processor (ATMega328) has been used, instead of a SAM one, and the measured accuracy was reduced to 10-bit due to the integrated ADCs. The communication moved to a wireless solution and lights, sounds and a web dashboard replaced the display for the output feedback, reducing too the cost of the actual system by a 30% factor. A picture of the current HOPES prototype is shown in Figure 3-1.

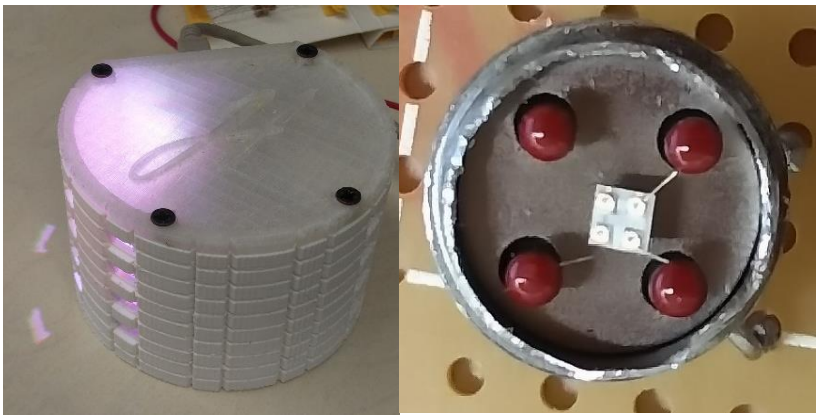


Figure 3-1: Current *HOPES* prototype and sensor detail

Working principle

The working principle of this class of sensors is based on the variation of the electric resistance of their active sensing layer, which is exposed to the targeted gases. Ideally, chemical reactions of the gases with the sensor layer are fully reversible processed. The metal-oxide gas sensors' design is relatively inexpensive and indeed adopted in a wide range of applications. Depending on the specific metal oxide substrate and on the gases targeted for detection, the temperature of the sensing layer varies in the range between 300°C and 900°C. The required temperature is attained thanks to an electrical heater located right under the sensor element. Common sensor material includes a number of semiconducting metal oxides, such as SnO_2, ZnO, Fe_2O_3, Al_2O_3, Ga_2O_3, and In_2O_3. Some new materials as V_2O_5, WO_3 and $Cr_{2-x}Ti_xO_{3+z}$ have been also attempted. A picture of the sensor with the active layer is reported in Figure 3-1.

Looking at the sensor equivalent measuring circuit (Figure 3-2), the circuit voltage (V_c) is applied across the sensing element, which resistance is indicated as R_s. The temperature of the elements rises thanks to the heater, attaining the active statefor the reaction. R_s changes as the reaction proceeds, indicating the presence of the target gas. The sensor signal is measured indirectly as a change in voltage across R_L, a load resistor connected in series to build a voltage divider. Indeed, the sensor resistance (R_s) is obtained by the formula:

$$R_S = \frac{V_C - V_{out}}{V_{out}} R_L$$

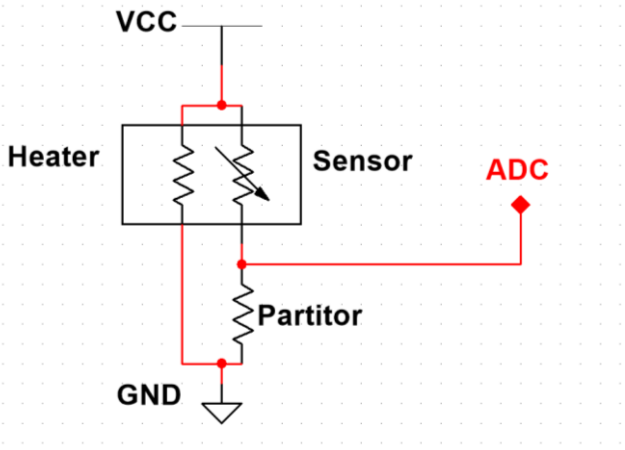


Figure 3-2: *HOPES* equivalent circuit

Concerning the accuracy, the resistance R_s varies according with temperature (T) and humidity (H) values (Figure 3-3). The warm up time is about 30 minutes and a dead time of 5 minutes between measurements is considered to avoid false readings and to allow the sensor to reach a new steady condition. In order to minimize statistical errors, every measure is computed from about 64 readings, oversampling by 3 bits the ADC resolution and giving a total value of 13-bit precision on the measurements.

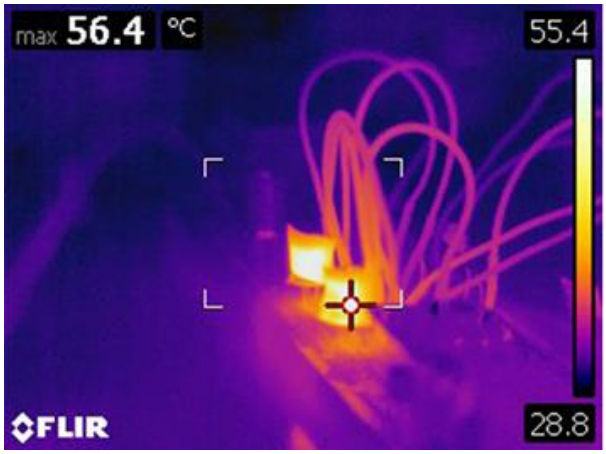


Figure 3-3: Sensor temperature analysis by means of a thermal camera

The system is also equipped with a Particulate Matter Sensor able to identify particulate of 1 to 2 μm in a 3-10 μm particles' set. To infer these measurements, an infrared light emitting diode (IR LED) and a photodiode are optically arranged in the device, with the latter detecting the IR LED light reflected by dust particles in air.

In the present case, the main asset is given by rover's autonomy and robustness in computing and tracking the paths required for the scanning of the pollution status over a large area. To this aim RAGNO has been equipped for this project with a GPS receiver, three axes gyros, accelerometers and magnetometers and a stereoscopic vision system, shown in Figure 3-4, which is instrumental to detect possible obstacles along the path [29].

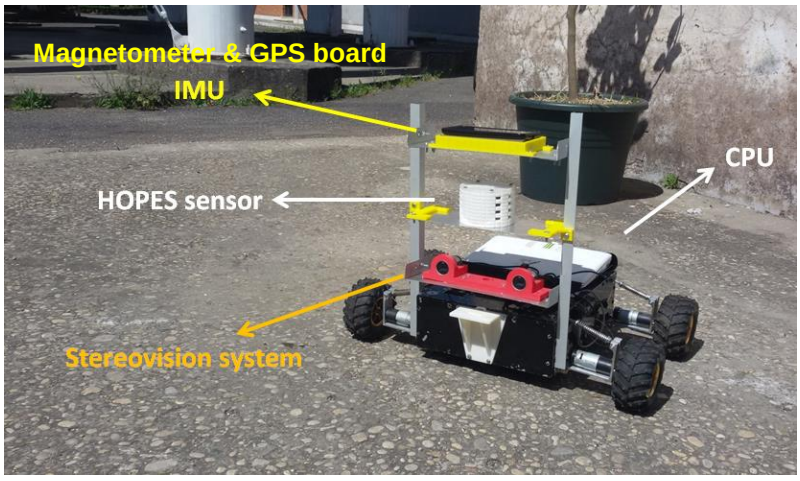


Figure 3-4:The rover equipped with the HOPES sensor

In fact, while the path can be planned relying on the computed position and on preloaded maps, the presence of unforeseen obstacles would easily jeopardize the

mission. This is the elective area of application for the stereoscopic vision technique presented in chapter 1 that could be now implemented onboard RAGNO.

3.2 Rover-Drone Coordination

3.2.1 Related works

The cooperation between ground and aerial unmanned vehicles is one of the most relevant research field of today robotics. In fact, the use of heterogeneous multi-robot systems allows to accomplish different kinds of missions in an unstructured environment, by exploiting both the advantages of an aerial view of the scenario and the capabilities of heavier, better equipped multi-task ground vehicles. On the other side, the optimal behavior of the swarm should be identified by considering the different dynamics – with different time scales – proper to the platforms, a significant issue with respect to the operations of homogeneous swarms.

The coordination and cooperation of the heterogeneous swarm require a robust navigation system and a significant control effort depending on the desired level of autonomy. Many examples of swarm architectures and applications can be found in literature. In these works, the presence of a single UAV - or a formation of - allows to support the ground vehicles (UGV) in safe path planning and navigation phases within a GPS-denied scenario characterized by the possible presence of obstacles. In all the analyzed works, the common solution adopted includes the integration of a vision system in the UAVs.

Garzon et al. [38] deal with the problem of the exploration of a wide and unknown area using a cooperative UAV-UGV system. The UAV is equipped with a single camera and an image processing software which is able to detect both the obstacles and the UGV. Once the safe path has been planned, it is communicated to the UGV. Krishna et al. [39] propose a similar swarm architecture underlining the advantages, in terms of UGV power consumption, obtained by executing the navigation and path planning phases on the UAV onboard computer. This approach allows to reduce the need of various

sensors on board the UGV for autonomous navigation. A different architecture for this hybrid swarm has been presented by Kim et al [40]. This work focuses principally on the navigation issue of a UGV in a GPS-denied and unstructured environment. The presence of a formation of two autonomous UAVs, both equipped with a monocular camera, allows to realize a stereovision system with a movable baseline. Considering the theory of the stereovision (i.e. the epipolar geometry), this system can generate disparity maps of the ground, detecting obstacles. Differently from the previous works, in this case, a specific effort is needed for the control of UAVs formation. A similar approach has been described by Oriolo et al. [41], looking at the case of an aerial segment consisting in a single drone while the ground part is a formation of identical UGVs. This study deals with the control effort that the UAV must fulfill to keep the ground formation inside the camera field of view to reconstruct the relative pose of each UGVs.

3.2.2 RAGNO-UAV proposed architecture

One of the main limitations of the rover is the impossibility to detect targets of possible interest that are hidden by ground obstacles. Rover's range of investigation is by far increased if its operations are supported by an aerial drone. Following the approach in [14], the proposed architecture considers a prototype drone (Figure 3-5) working as a remote vision sensor. Drone's task will be to detect by visual inspection the target, to geolocalize it, and to communicate its coordinates to the rover. As described in Section 2.2, the RAGNO rover presents high level of motion autonomy - thanks to the onboard integration of a guidance and stereovision based navigation system - that it will be capable to define its path once target coordinates are known.

Figure 3-5: The remotely controlled drone adopted for this project

The limited performance of the drone onboard computer allows to control only the rotors angular velocity and the drone attitude, while the image processing is currently carried out by an external workstation inside the Lab building. This workstation acts as a server for the telemetry, command and control between the rover and the drone, with all the data being broadcast via internet.

In a further development, there will be an upgrade, increasing the drone's computing power and communication capabilities and thus converting the current centralized, lab-based architecture in a totally independent one. According to this future approach, the hybrid system could be considered as a communication net in which a master node (the drone) broadcasts the data to the slave nodes (one or more UGVs). Next, once the new architecture will be successfully tested, a further step could be also accomplished, by enabling a two-ways communication. At such a stage, rovers too will be able to send important updates to the drone, influencing its guidance on the basis of the data gathered on ground.

In the original, current architecture the task of identifying the target can be performed by exploiting the aerial images by means of the human intervention (if the drone is remotely controlled) or by computer vision algorithms that allows the tracking of particular features in the acquired image. Even with a drone remotely controlled, its navigation system must be able to determine its position and attitude. In fact, these data

are necessary to evaluate the target position both in a georeferenced system and with respect to the target. Since the main instrument to detect the target position is the onboard camera (a monocular vision system, in this case), the information about the drone's camera altitude (*h* in Figure 3-6) is still required to extract the measurement of the target position (*T*), the rover position (*R*) and the TR vector.

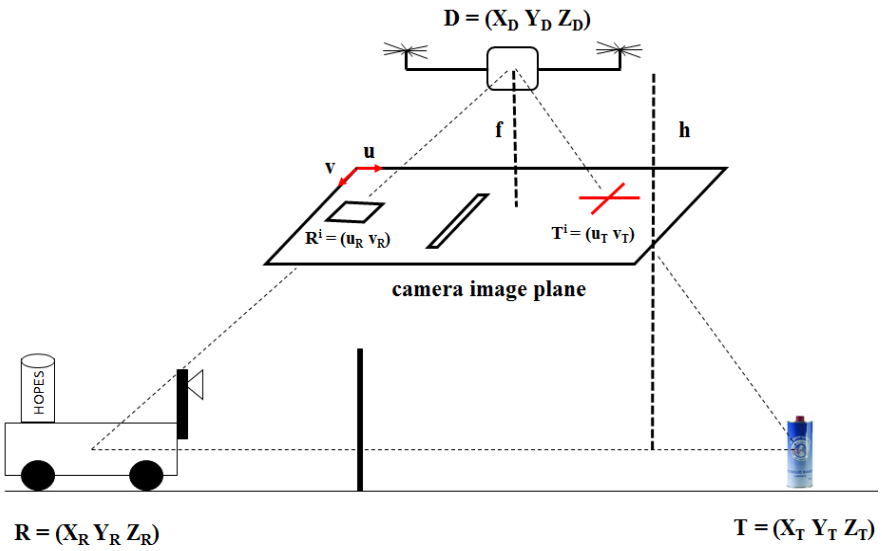


Figure 3-6:Scheme of the geo-localization of the target.

3.3 Experiment

In the preliminary experiment performed to verify the validity of the concept, a fictitious obstacle has been placed between the rover and the target to be analyzed by means of the HOPES sensor (see Figure 3-7).

The drone is tasked to capture the scenario in order to detect the target, elaborate its position and communicate it to the rover. In this preliminary work, in which the drone is not yet fully autonomous, nor equipped with enough computing power to perform the relevant navigation and image processing, previous steps are completed offline. In a real operative scenario, the coordinates of the target could be computed either as the vector sum of the inertial position of the drone plus the target/drone distance computed by image analysis, or as a relative distance with respect to the rover. The difference between the two approaches consists in the required sensors: in the first case the rover

must be equipped with a navigation system providing its inertial position (like INS or, for longer time intervals, GPS device), to exploit the inertial position of the target as communicated by the drone; in the second case, since only relative positioning is involved, the GPS receiver is not needed. Indeed, the second technique enables missions in GPS-denied environment, at the – significant - cost of the computational resources onboard the drone required to fully process the images to identify the target and the rover. The onboard batteries' capacity is also part of the trade-off, to ensure the required a satisfactory flight time allowing to accomplish a successful image processing. The choice between a greater importance given to the image analysis or, conversely, to the GPS position depends on the scenario, according to the ranges of interest, the size of the possible obstacles and several additional parameters.

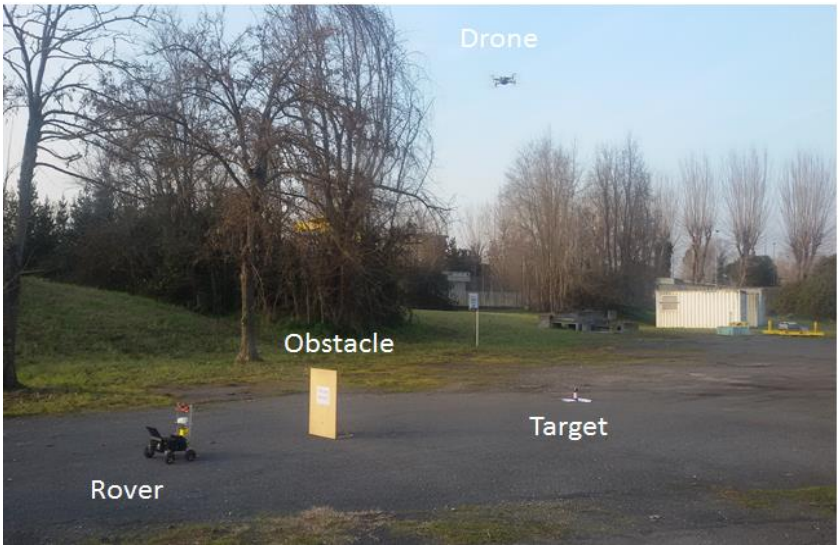


Figure 3-7:The scenario of the experiment.

Figure 3-8 shows an example of image acquired by the drone; the target is highlighted by the use of a white cross-shaped marker. In this case, a template matching strategy could be used to determine the position of the target in the image plane. Notice that aside from the possible previous knowledge of the target size, the altitude of the point of view, i.e. the 3rd coordinate not immediately exploitable from the image, can be provided by the drone navigation system. The information from the image is indeed actually translated in 3D coordinates.

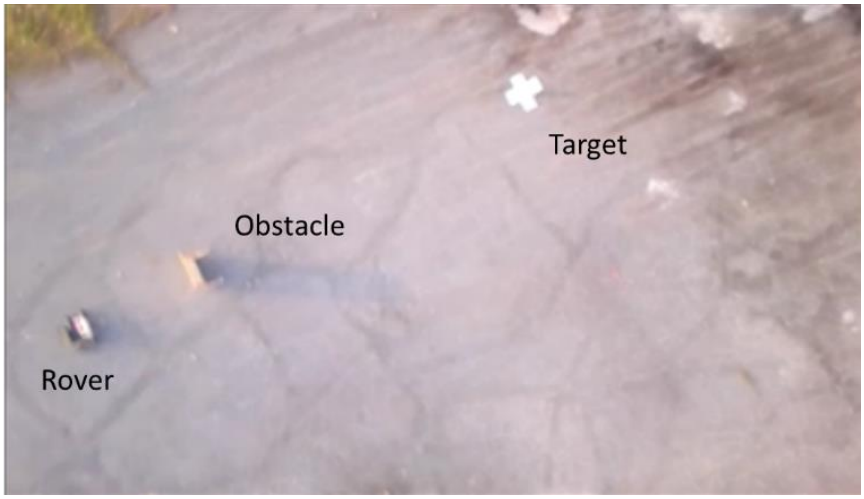


Figure 3-8:A snapshot acquired by the drone camera.

Once the rover receives the information about the target position (inertial or relative to its position) the stereovision is activated to verify the presence of possible obstacles along the path. Figure 3-9 shows the fictitious obstacle acquired by the left and right cameras.

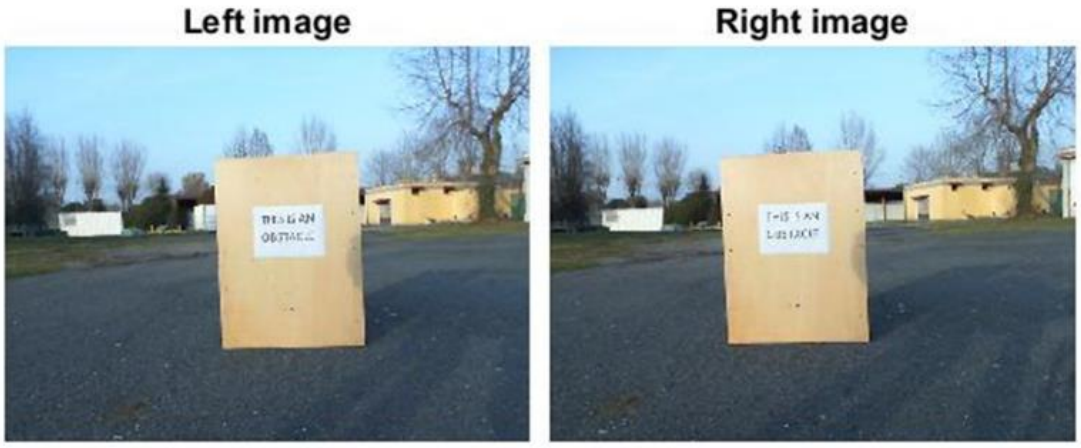


Figure 3-9:Observed obstacle

Target and obstacle positions relative to the rover camera can be evaluated, and the optimal path to reach the target can be computed. To this aim, the so-called A-star algorithm, based on the graph theory, is currently used by the rover logic. Indeed, the motion plane is divided in square cells and the shortest path is computed by optimizing a constrained cost function of the displacements. The admissible elementary displacement is such that it drives the rover from its current cell to one among the neighbours. The resulting path is illustrated in Figure 3-10. The green square represents

the target position, while the red square represents the obstacle, with the blue squares representing the safety zone around the obstacle.

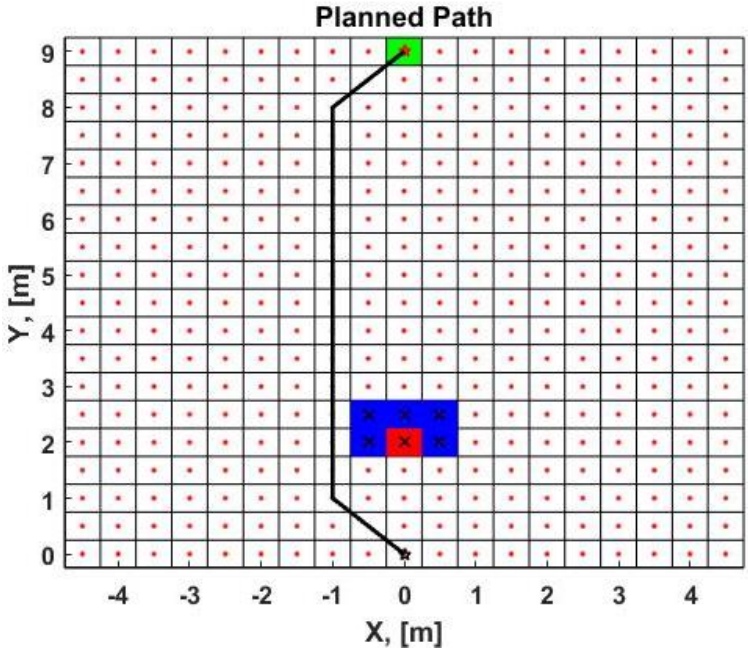


Figure 3-10: A-star planned path

As a preliminary test to assess the feasibility of the approach, a clearly identifiable polluting target - i.e. an acetone bottle - has been placed in the target site. The rover slowly approaches the target, while the HOPES sensor acquires data at the rate of 0.2 Hz, with the results reported in Figure 3-11.

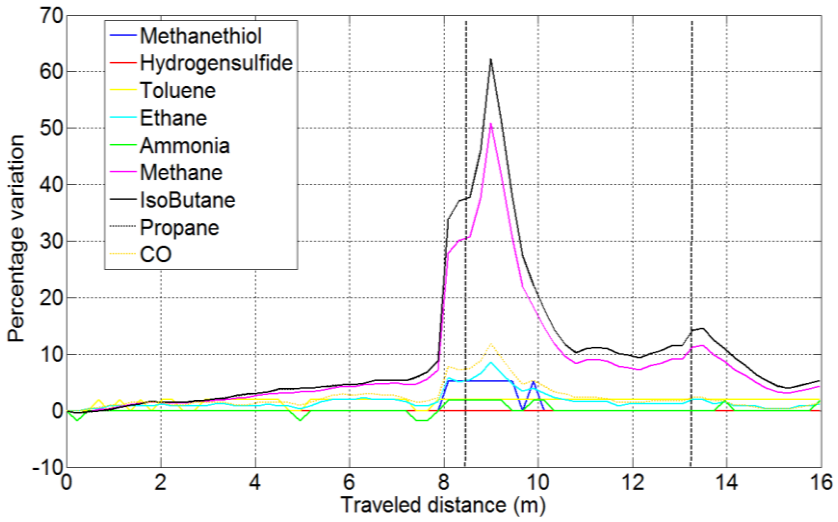


Figure 3-11: Results of the preliminary test campaign; the two peaks in the detected substances are clearly visible.

The presence of the acetone bottle at a distance of about 9 meters from the rover exploration starting point can be easily assessed thanks to the data relevant to involved polluting substances, especially methane and isobutane.

3.4 Final Remarks

The obtained results are quite promising about the capabilities of the proposed autonomous system. Once the system will be fully validated, i.e. when all the measurements and relevant processing will be performed in real time and in a reliable loop involving the drone and the rover (i.e. totally excluding the human in the loop as instead in the experiment of this paper), the approach will be extended by making use of the concept of robotic swarms. In fact, a single drone could function as a leader agent for multiple rovers, spread over a wide area. Their inter-rover links, together with the central command provided by the drone, should hugely increase the performance of the single robot architecture both in terms of a clever coverage of the area, as well as in the relevant terms of the efficiency measuring possible sources of pollution.

A possible scenario for this approach could be represented by areas in which the rover can move without the risk of collisions with high speed objects that the stereovision cameras would not be able to avoid. Dismantled industrial sites, for example, could present a number of polluting or contaminating sources that the interaction between the aerial and terrestrial robots could extensively detect (drone's role), investigate (rover + sensor role) and eventually even remove safely, thanks to ad hoc robotic manipulators onboard the rovers.

A different example could be represented by a cooperative system given by a drone formation and a swarm of rovers, all together exploring a wide area. By supposing that drones are able to accomplish targets recognition and path planning operations, and by assuming that the position of each rover is known to them, the formation can assess which rovers are closer to the detected targets and command them to follow the

planned paths. In this way, the ground swarm can be smartly managed to limit power consumption and therefore attain a longer duration for the mission.

Overall, the successful implementation of this small fleet for environmental monitoring opens logically the path to the following chapters, devoted to swarm exploring unknown scenarios.

Chapter 4.

Guidance of Fleet of Autonomous Vehicles

4.1 Introduction

Space exploration has been consistently based on autonomous systems, due to the strong constraints given by limited visibility and, in many cases, extended link time. Such a requirement for autonomy usually meant complex equipment, able to work without supervision. Miniaturization in electronics changed this scenario, allowing to reduce the volume, the mass and the power required by space vehicles, with obvious advantages in terms of launch capabilities and costs.

Miniaturization and increase in performance provide today also the case for rethinking autonomy. In fact, instead of building large (or at least as large as the launcher could support) monolithic explorers, it should be possible to deploy fleets of smaller vehicles. As a result of the analysis of many living organisms, exploration or patrolling by means of swarms of robot found since the 90's significant interest [42]. Several studies attempted to extend to space exploration scenarios the swarm behaviour [43]. In fact, having a fleet of simpler yet capable and autonomous explorers instead of a single, large, even more powerful vehicle can greatly increase the mission data return. Planetary explorers, specifically rovers, are extremely slow and – also considering the intrinsic failure risk in manoeuvring - the extent of the unknown land they can survey daily is quite limited, with severe scheduling issues [44]. Acting with several, simpler vehicles can greatly increase the amount of observed surface, allowing indeed to a quicker identification of areas of interest. Furthermore, less complex agents should be also easier to operate, and less prone to failures. At the same time, the failure of a single agent does not mean the complete failure of the exploration campaign, with the fleet being able to re-organize [45]. If the survey discovers locations requiring more powerful instruments, specialized members of the fleets of explorers could be directed to their elective playground. Notice that such a plan includes the possibility of a fleet composed

by explorers of different nature, specialized in different tasks, as already proposed by several NASA studies [46]. An example of this cooperative effort is offered by the experiments carried on at Surrey in the frame of the FASTER project [47]. This exploration scheme does require some coordination among the vehicles, depending on the specific tasks assigned to the agents and to their capabilities in terms of guidance, navigation and control [48].

This work is only considering a swarm of identical, wheeled rovers, a solution capable to provide several advantages, yet certainly not the only one possible. Specifically, attention will be focussed on the patrolling role of the swarms, with the analysis of possible coordinated guidance schemes.

The optimal guidance problem has received in time a great deal of attention from researchers. In the frame of the convenient guidance from the current position to a given target, one of the best solutions is offered by the A* method [50], already described in Chapter 2. Similar approaches can be useful also with respect to the requirement of exploring a given area, i.e. to move among assigned nodes. With a swarm of platform cooperating in the same scenario, guidance rule should be necessarily updated. A number of studies [51][52] tackle the issue of the optimal operations for a fleet of explorers.

The material reported in the following includes the findings of the study carried out at the Guidance and Navigation Lab, based on numerical simulations inspired by the real-world characteristics of a platform like the RAGNO rover described in previous chapter 2. Contents have been presented to major conferences, mainly with papers [13] and [14].

After introducing the STAR concept, focus will be given to the navigation subsystem, up to the detailed definition of a strategy designed for the drawing of a morphological map of the explored environment. The simulation environment created to analyse system performance will be also presented. Later on, the fundamental interconnection between the navigation and the communication subsystems will be considered, and the

simulation environment will be modified and improved to take into account real world characteristics affecting the navigation function performance.

4.2 STAR: Swarm of Two Autonomous Rovers

A homogeneous swarm of robots can be defined as a set of mostly simple platforms which work in the same environment with different tasks. The task depends on the payload hosted onboard each platform. As the rovers move in a common scenario, they can cooperate by sharing useful information. Therefore, the cooperation makes the exploration more efficient and safer. The data set to be sent can be arbitrarily defined depending on the overall target of the exploration campaign and on the desired/allowed level of collaboration.

The case of the Swarm of Two Autonomous Rovers (STAR) exploring an unstructured area around a main station will be specifically detailed in the following. The main aim of the work is to describe the strategy adopted for the cooperative motion of the swarm. We assume a specific, realistic scenario: the main station is a scientific base on Mars surface and the swarm has been landed far from it. The goal of the mission is therefore the building of a morphological map of the surrounding environment while the robots drive from the landing points toward the main station. Once the swarm reaches the base, there could be the chance to download all the acquired information and send them to Earth as well as exploit them to plan more specific activities.

The presence of static obstacles could jeopardize the integrity of the swarm and, therefore, the mission. For this reason, a robust guidance strategy needs to be implemented. In our model, the two rovers follow the same planned trajectory, adopting a leader-follower configuration, until an obstacle is detected on the path. At this point, the swarm switches to a bypass mode: the leader separates from the follower and they circumnavigate the obstacle following paths with opposite directions. As the circumnavigation is completed, the leader-follower configuration is restored and the swarm drives toward the goal position following the preassigned trajectory.

In the next sections, the steps of the swarm behaviour implementation as well as the adopted guidance strategy are reported in detail.

4.3 Rovers' implementation and trajectory choice

The implementation and simulations of STAR in different types of scenarios have been realized in MATLAB® 2015. At the beginning of the simulation, the user defines the desired exploration trajectory as well as the configuration of the scenario. The two rovers are identical from the hardware point of view and they host the same sensors. A vision capability – intended as a software simulation of a real stereovision system - is implemented on both the platforms. The system is characterized by a triangular field of view defined by the opening angle (α) and the maximum perceived depth (h) (Figure 4-1), with the user able to modify the two parameters. The simulated system synthetizes in software, in a very simple way, how stereovision can be used for navigation.

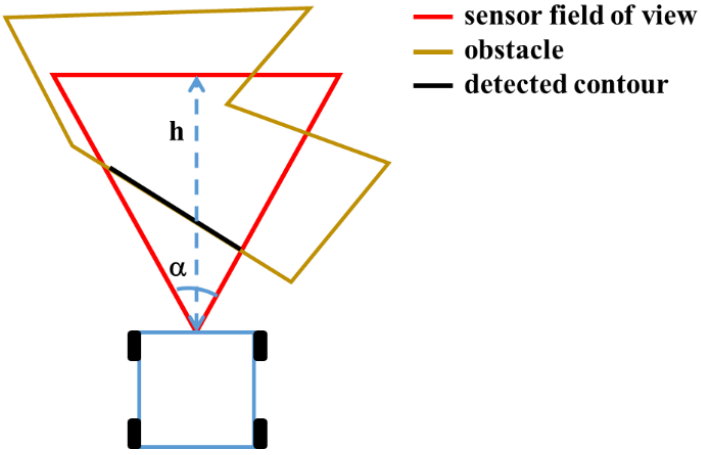


Figure 4-1: Relevant parameters for the implemented vision system

The vision system allows to detect and localize the obstacles in the scene by computing an intersection operation between the sensor's field of view and the contour of the obstacle. The process also considers the presence of possible occlusions. By looking at Figure 4-1, it is possible to notice that some occluded walls of the obstacle could pass undetected by the simulated sensor: this feature is included in the simulation, making the implemented system more realistic.

Once the sensor has been implemented, an exploration trajectory needs to be planned. We suppose that the relative position of STAR with respect to the target station is known. The path must satisfy two constraints:

- beginning and ending points must coincide with the STAR and the base station positions;
- the explored area surrounding the base station must be maximized.

A suitable trajectory which satisfies these constraints is the spiral (Figure 4-2). In fact, it allows the rovers to progressively explore the surroundings while approaching the goal position. The number of laps around the target can be set: it should take into account the sensor's characteristics (α, h) in order to avoid repeated passages in already covered areas thus minimizing the consumption too. The wider the field of view, the lesser the number of arms should be.

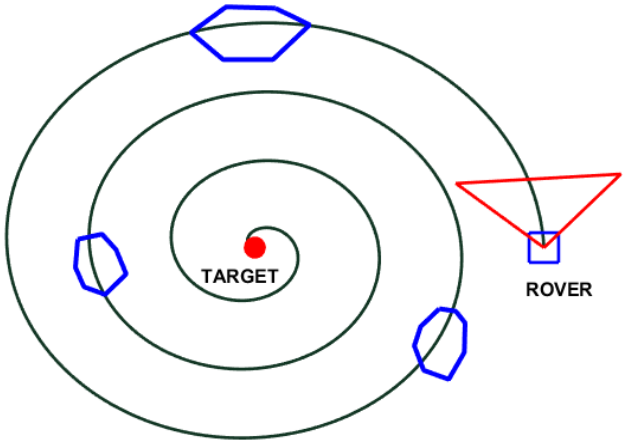


Figure 4-2: the selected spiral trajectory

4.4 The guidance strategy

The design of the guidance strategy considers the goal of the mission as well as the constraints imposed by the remote planetary environment. In the case of STAR, the target is the building of a morphological map of the explored area by exploiting the cooperation of the two robots. We suppose the unavailability of any external systems (like the GPS or similar), therefore the rovers must rely only on their onboard sensors for both navigation and communication issues. In other words, due to the expected lack

of references, preference should go to autonomous navigation techniques. Among realistic possibilities there are inertial navigation and image-based navigation.

Inertial navigation – belonging to the deduced reckoning class, will suffer from errors monotonically increasing in time. Sensors includes accelerometers and gyroscopes of different flavours. All of them are available in miniaturized forms, already having flight experience. Of course, a deduced-reckoning type system, with errors monotonically increasing in time, will ask for re-alignment from time to time; such an operation could be performed by external, supervisory agent or by recognition of relative pose with respect to landmarks.

Even more important, as related to the safety of the agent themselves, is likely to be the short-range navigation to avoid obstacles and dangerous passages. The only option to solve this issue is related to visual techniques, and especially to stereoscopic navigation as – within reasonable power budget compared to active LiDaR – it would allow to define the depth of the scenario (like in RAGNO that is assumed as a model).

This rationale justifies the architecture for the navigation subsystem of the two vehicles as discussed in previous paragraph.

Notice that while the real-time positioning can be – in a technological meaning – "easily" achieved through the acquisition and fusion of different measurements (accelerometer, gyroscopes and wheels encoders), the data sharing can be harder to realize.

In the remote space exploration, the data link between the mobile rover and the ground mission control can be achieved in two ways: directly or indirectly. The direct way implies that Earth is in the field of view of the robot's antenna. In this case, the data rate is affected by different factors like the Earth visibility time, the robot's antenna characteristics and, mainly, the robot available power. The indirect link implies that the data are sent to a relay-orbiter and then forward to the Earth mission control. This kind of communication system has three advantages:

- the orbiter passes many times over the rover therefore the information can be transmitted with a lower rate, thus saving rover power consumption.
- the orbiter has the Earth in its field of view for much longer time periods than the rover on the ground, therefore also this leg of the link improves.
- the orbiter can be equipped with larger antennas and has more power available. This allows to send much more data to Earth than a rover could do.

STAR has been thought as an Earth-independent exploration system, i.e the swarm does not send data to Earth while travelling toward the base station. The communication link is solely established between the two robots. In the implemented software, the two platforms communicate thanks to a local Wi-Fi network. The TCP/IP has been chosen as the communication protocol. The main drawback of the WiFi signal is its limited range. This implies that the robots must move close each other. For this reason, the implemented guidance strategy envisages two possible operation modes for STAR: the *leader-follower* (LF) and the *sequential obstacle circumnavigation* (SOC).

In the LF configuration, the rovers move single row. The leader explores the surrounding, looking for dangerous obstacles. If no one is detected, it sends a *free-way* bit to the follower which continues to drive along the planned spiral trajectory. In the opposite case, the leader sends a *stop* bit to the follower and STAR switches to the SOC configuration.

The selection of a safe circumnavigation path is achieved by discretizing the motion plane in square cells, whose central points are called nodes. The deviation is then planned by means of the A-star searching method: this algorithm is based on the graph theory and allows to compute the safe shortest path by optimizing a cost function under motion constraints. In fact, the rovers' admissible elementary displacement is such that it drives the generic robot from the current node to one of the eight adjacent cells. The cells which are occupied by an obstacle are marked as unfeasible.

The SOC configuration envisages that the leader platform bypasses the obstacle while the follower is at rest. At the end of the circumnavigation, the leader transmits the list

of the path's cells and of the cells marked as unfeasible (because of the presence of the obstacle) to the other rover and authorizes it to move. In this way, the follower knows which side of the obstacle has been mapped. Indeed, it plans and follows a path that allows to reconstruct the unexplored contour of the obstacle. When the two robots rendez-vous, the LF configuration is restored.

The computation of the shortest path requires the definition of a target node. The discretization of the motion plane means that the spiral trajectory can be considered as a succession of way-points nodes. When the leader detects an obstacle in a K-th cell of the discrete spiral trajectory, the corresponding node is no more considered as a way-point. Therefore, the A-star algorithm is initialized by setting the (K+1)-*th* node as the target one. Once the first guess path has been computed, the leader starts to follow it. The target node needs to be updated every time the rover reaches a new way-point and acquires a new image of the scene. In fact, three situations could happen:

1. the target node is still free and the robot has not bypassed it yet;
2. the target node is still free but the robot has already bypassed it;
3. a new part of the obstacle is detected within the current cell of the target node.

In the first case, the target node is not changed and the robot continues to follow the previous guess path. In the other ones, the target is replaced with a following node of the spiral path. Let's focus on case 2, depicted in Figure 4-3. Notice that, in the robot reference frame, the target node T is behind the rover because the angle between the RT segment and the Y-axis is negative. The implemented guidance algorithm replaces it with another node and then it computes a new deviation path.

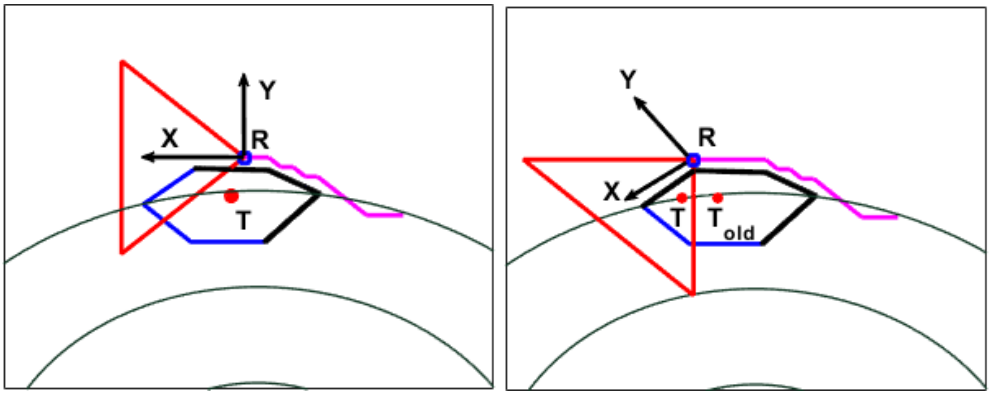


Figure 4-3: the target node updating process. In the left figure, the T node is beyond the rover R. In the right figure it has been passed by the rover, therefore it is replaced with a new T node. XY is the body reference frame.

Once the leader has bypassed the obstacle, it stops and sends a "*wake-up*" bit and a packet of information to the follower. This packet is composed by the following arrays of data:

- index of the leader current node;
- indices of the travelled cells;
- indices of the detected obstacle cells.

The reception of these information allows the follower to plan its circumnavigation path. However, the guidance strategy is different from the previous one. In fact, the leader's current node is assumed as target and it is not changed during the follower's travel. In addition, the cells contained in arrays 2 and 3 are marked as not accessible, to plan a deviation path which bypasses the obstacle from the unexplored side. The sharing of the occupied cells, instead of the detected contour of the obstacle, allows to reduce the amount of data to send. At every step of the circumnavigation, the follower sends a *wake-up* bit to the explorer. The bit changes when the follower-leader rendezvous is achieved. At that time, the LF configuration is restored. The following tables show the steps of the guidance algorithm of both the explorer and leader rovers:

Explorer rover

```
while XROVER ≠ XTARGET
  acquire a new image of the scene

  if DetectedObstacle = true
    send StopFollower bit
    define WayPointNode
    define CurrentNode

    while CurrentNode ≠ WayPointNode
      plan the CircumPath
      reach the next node of the CircumPath
      update the WayPointNode (if necessary)
      update the CurrentNode
      update the PathNodes list
      update the ObstacleNodes list
      acquire a new image of the scene
      send StopFollower bit
    end

    send [CurrentNode PathNodes ObsNodes]
    receive WakeUp bit

  else
  move along the spiral trajectory
  update XROVER
  send FreeWay bit
end
```

Table 4-1: Explorer's guidance algorithm.

Follower rover

```
while XROVER ≠ XTARGET
  define CurrentNode
  receive StopFollower bit
  acquire a new image of the scene

  if StopFollower = 1
    receive [TargetNode PathNodes ObsNodes]

    while CurrentNode ≠ TargetNode
      plan the CircumPath
      reach the next node of the ShortestPath
      update the CurrentNode
      acquire a new image of the scene
      send WakeUp bit
    end
```

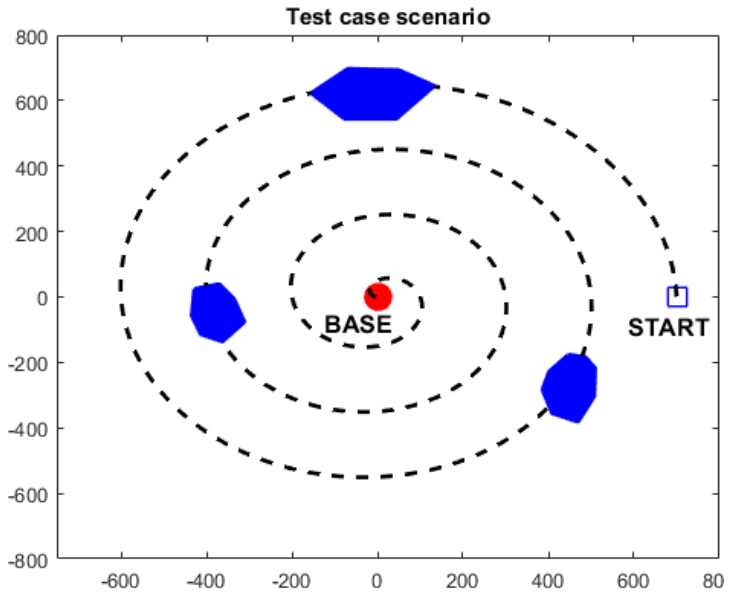
else
  move along the spiral trajectory
  update X_{ROVER}
End

Table 4-2: Follower's guidance algorithm

4.5 Simulations and results

STAR guidance algorithm has been validated by running a large amount of simulations in different scenarios. These simulations have been performed on a unique PC, equipped with a 2.4 GHz CPU and 8 GB RAM, with two sessions of Matlab 2015® running in parallel. Regarding the data sharing, the two robots' onboard computers will be linked by adopting the TCP/IP function embedded in the Matlab 2015® suite. Preliminary tests have allowed to estimate the average computational time required for the execution of a single iteration of the two programs. This interval (nearly 500 ms) has also been considered as the time rate of the data link since every iteration ends with the sending of an information.

The following figure shows the scenario and the path resulting for the two rovers from a generic simulation.



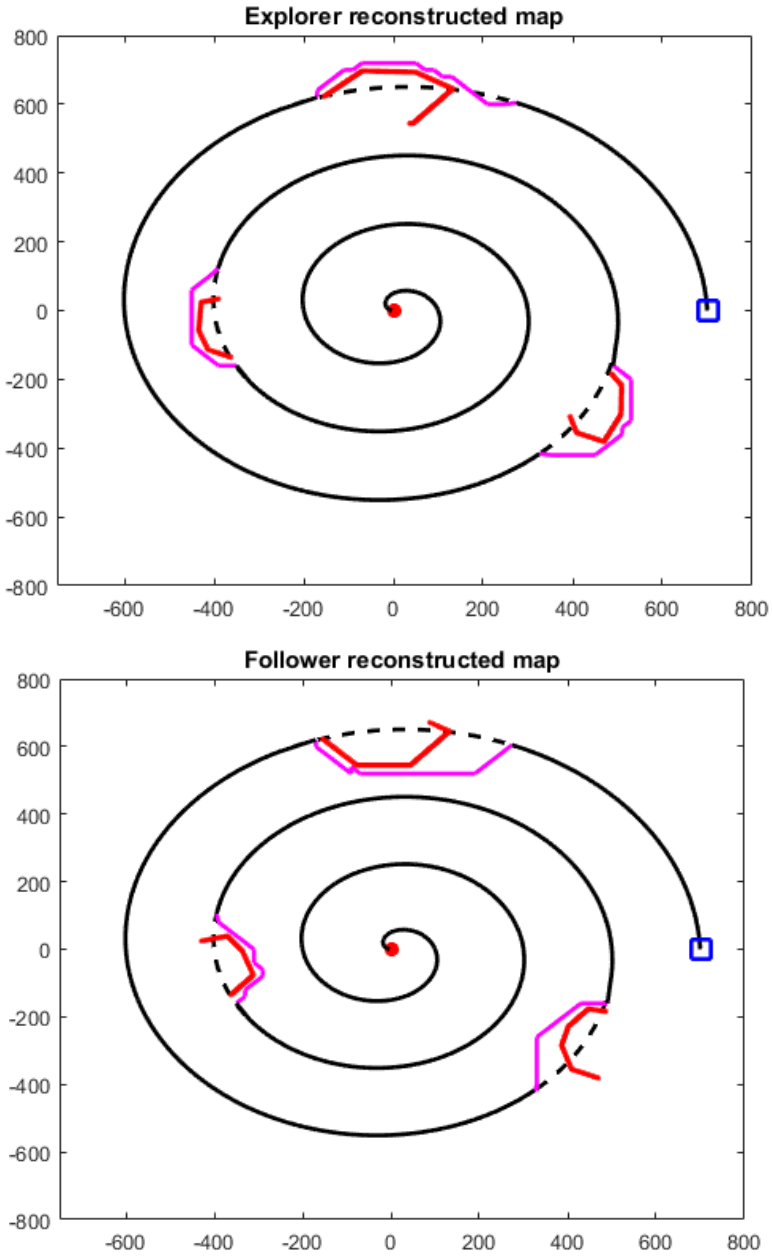


Figure 4-4: simulation of a test case scenario, with the sketches of the reconstruction given by the two agents. Units are in meters, with a=45°, h = 20 m for the sensor. The dashed path is the desired one, while the deviation trajectory is in magenta.

In the example, the swarm – simply represented by two agents - has been landed 700 m far from the base station. As we can see, the two robots follow different deviation trajectories, thus allowing a more complete mapping of the explored terrain. Since the rovers share the cells occupied by the detected obstacles instead of their contour, it is necessary to merge the two maps to obtain the full one. This might be done once STAR has reached the base station.

4.6 About the Navigation-Communications Subsystems' Connection in Space Exploration Vehicles

Exploration of the solar system is constantly proceeding, with a timeline sometimes supposed to be slow but actually quite faster if considered the really short duration of the space era. Notwithstanding the recent beginnings, the vision for space exploration already changed, in agreement with the quick advances of technology and with the increasingly deeper knowledge of the space environment.

On the first side, i.e. the technological one, the huge cost of launching significant masses, the very large integration in electronics and the significant improvements in electro-mechanical miniaturization and in computation power all support the trend towards smaller explorers. On the side of expertise, the acquired awareness of the significant chance of failure in challenging space missions and their long preparation time advice for a partition of risks. The newly available capabilities and the quest for higher robustness indicate a flexible, effective mission concept where several probes (either satellites or rovers) operate together to accomplish mission's goals. The concept has been already approached in many terrestrial uses, to enable autonomous, robust behaviour in hostile environments [49][55][56]. Looking at space applications, the multi-agent architecture has been already implemented in commercial ventures aimed at terrestrial orbits with large constellations. It is however definitely advancing also in the field of exploration with the initial step of main platforms deploying orbiters (BepiColombo for Mercury) or landers (Rosetta and Philae): the overall return can significantly increase thanks to the combination of the observation from different vehicles, while single issues to one of them do not mean a complete failure.

Collective character will assume stronger relevance in future, with fleets of explorers aiming at the same target. However, the effective implementation of the multiple agent concept poses important issues, unknown to large, single bus missions. Different architectures can be envisaged, with a single leader and several slaves, several subleaders and additional slaves, or perfectly identical agents. The different solutions

would provide different levels of reliability, of robustness to partly unknown environmental conditions, of attainable results and returns in terms of missions' payload.

All of these missions are strongly depending on their communication [57] and navigation subsystems [58]. In fact, whatever the specific goal, to know the actual position of each single agent is clearly of paramount importance to in order to carry on a meaningful mission. Such a knowledge can be achieved at either local or global level, and eventually shared with other – or all of – agents, depending on the selected control architecture. In order to communicate the position, and also to allow the use of payload data, a communication system should be operational too.

Also with reference to the increasing literature in the field, this section aims to present some preliminary considerations about the two subsystems, in the attempt to maintain the analysis as general as possible with respect to the characteristics of the host platforms. In the case of planetary exploration some features of the environment will be of course important. Among the different scenario possibilities, some reference will be given to the case of Mars, that seems to be the most interesting for real applications in a close future.

The study will proceed through the analysis of the different aspects, starting with the communication side (paragraph 4.7) aimed to introduce the link characteristics. Then, paragraph 4.8 will discuss the navigation issues, detailing possible selection of the observable and the related constraints on the sensors. Of course, all of these subsystems (communication and navigation) will be primarily focussed on simplicity, economicity, reliability. Paragraph 4.9 provides some basic information about the application of estimation theory (filtering, to be more precise) and the resulting accuracy in solution. Finally, paragraph 4.10 discusses robustness issues, from the two different points of view of the performance in presence of obstacles and of the resilience with respect to agents' failures.

4.7 Communication constraints

Efficient guidance of a fleet of explorers builds on a communication network, to be conveniently provided in radio-frequency bands. In fact, optical communications could be far more complex in hardware, hampered by obstacles in the line of sight, challenged by sand storm (on Mars) and, more important, do not seem required at all due to the limited volume of data to be transferred among the explorers.

Nevertheless, communication link should be modelled in a realistic way. A first traditional concern deals with the selection of the link's frequency. In the specific case of a swarm of explorers, the maximum achievable range is certainly important. At the same time, robustness of the associated equipment and indeed the wide availability of an existing and relatively inexpensive technology (equipment would need to be procured in huge numbers) are also important. A preliminary analysis can be carried on by using the Frijs equation (also known as radar equation) traditional approach to compute the available received power at the link's end, with the evaluation of the resulting signal-to-noise ratio and its comparison with the requirements from the receiver in order to effectively process it:

$$P_{rx} = \frac{P_{tx}\, G_{tx}\, L_{ch}\, A_{rx}}{4\pi R^2}$$

$$SNR = P_{rx} - 10\log_{10} T_{sys} - 228.6 - L_{rx}$$

Eq. 4-1: Received power and signal-to-noise ratio equations

In general, once defined the characteristics of the transmitting and receiving equipment, the free space path loss appearing in Eq. 4-2 will increase with the operating frequency (Figure 4-5, and [59]):

$$FSL = 20\log_{10}\left(\frac{4\pi r}{c}\, f\right)$$

Eq. 4-2: Free space loss equation

In addition, limit on the channel capability (i.e. data rate) should not be a constraint for the specific application, as explained before. It is possible also to remark that required

bandwidth is not intended to be large, again due to the expected limited amount of data to be transmitted. Moreover, in possible critical conditions the trade-off between narrow bandwidth and a larger one requiring coding ends up in favour of the first solution, due to the limited efficiency of the coding as far as it concerns power requirements [60]. Antenna size, an important factor in space hardware and a possible advantage for shorter wavelength domain - from the approximate, general still valuable relation

$$G = \frac{4\pi A}{\lambda}$$

Eq. 4-3: The gain of the antenna

- is not a specific issues as requested gain G should be very low and more on the side of the omnidirectional broadcast to handle poorly known relative positioning of the agents. Indeed, portion of VHF or UHF bands are likely to be a suitable choice.

4.8 Planetary surface radio-propagation

Dealing with agents wandering on a planetary surface, previous consideration should be obviously adjusted to the specific environment.

A first concern deals with the long range-capability, associated with possible beyond line of sight communications. Looking at the case for Mars, these capabilities depends on the ionosphere characteristics, which are different, due to the weak magnetic field, with respect to the Earth. According to [61], the simplest structure and limited charged particles' density (two orders of magnitude less than in the terrestrial case) leads to a ionospheric layer which blocks frequencies lower than 4.5 MHz and is transparent for frequency higher than 450 MHz. Transmissions at frequency lower than ft can be managed for ionospheric bounces and indeed realize very long range links.

A second concern deals with the attenuation. As far as it concerns atmospheric attenuation, once considered the composition of the atmosphere, assumed at the planetary surface [59] as CO2 (95.3% in volume), N2 (2.7%), Ar(1.6%), O2(0.13%) and

then CO (800ppm) and H20 (300ppm), the attenuation in the case of Mars can be reported as approximately due to the only components affecting microwaves, i.e.

$$k_{atmo}(f) = k_{H_2O}(f) + k_{O_2}(f)$$

Eq. 4-4: Attenuation factors for the martian atmosphere

According to [59], the attenuation factors in dB/Km are 4 orders of magnitude lower than in the terrestrial case, and indeed could be considered as negligible. Above all, it can be remarked that, once a reasonable composition – even based on the findings form more recent missions – would be assumed, there is the possibility to compute all losses in Eq. 4-2. An additional, and specific problem of the Martian atmosphere is instead represented by storms. Also in this case, the effect can be considered as small (like 3 dB loss at Ka band, yet depending on the assumed size of the dust particles [61]), except with regard with possible optical communications that call for more in-depth analysis and could end up to be impossible at all.

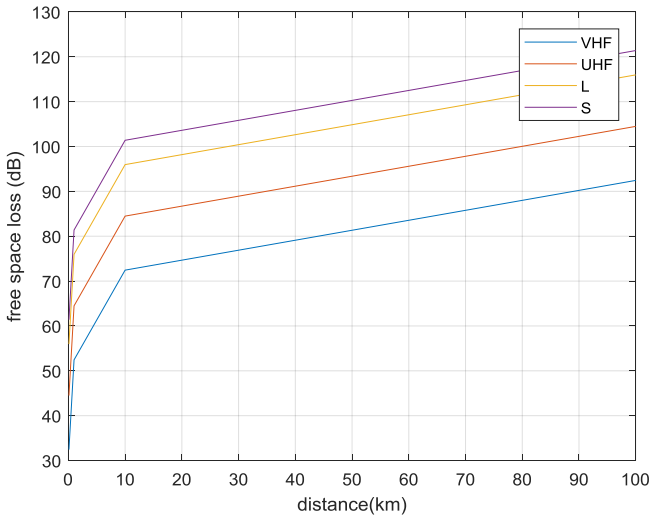


Figure 4-5: Free space loss in different bands

4.9 Navigation

As stated, the knowledge of the actual position of every single agent is obviously instrumental to define and then operate an effective exploration campaign. According to the main goal to identify a reliable solution, it seems clear that in an almost unknown – and risky - environment the solution to have full knowledge available to each single

agent should be pursued. Indeed, every single probe should be able to define its full kinematic state, and to communicate it to the rest of the fleet. Then, there is the need to combine such a knowledge with the surrounding environment, by locating the probe with respect to significant landmarks, and to possible obstacles in the motion, and to convey such a useful information in an effective way. This second step is clearly halfway to the guidance function, as it will remarkably affect the way exploration will be carried out.

Notice that – even in the mixed communication-navigation frame, the exploration behaviour is still dictated by the capabilities of the sensors onboard the agents, as the maximum viewing range will command the allowance to the obstacles, and indeed the confidence in the assessed path as well as the time to effectively complete the mission.

Overall, the architecture already discussed can be confirmed, and actually reinforced when uncertainties related to data exchange area introduced in the loop. In fact, strengthening the validity of the information to be transmitted (i.e. the locally compute kinematic state of the vehicle) will add reliability to the system increasing the confidence in the few data eventually received during intermittent, noisy links.

The overall, designed scheme for exploration already presented has been improved to consider more general scenario. The main characteristic is the implementation of an environment where different distribution of obstacles could be included. The simulation is performed by means of more MATLAB processes running at the same time and exchanging their knowledge by accessing a common database (Figure 4-6). Such a multiple access is realized through the TCP/IP technique as in previous paragraphs, yet now the availability of the data can be of course modulated to take into account the characteristics of the link and indeed far better representing the real world.

In numerical simulations, the scenarios already presented in previous paragraphs have been first considered, to accomplish a validation for this more complete, system-level approach. Then improvements to the code, or the possibility to tackle more general environments have been added, including as an example the capability to follow

different paths, more specifically, sinusoidal or square-wave like trajectories (Figure 4-7). While the final goal is clearly represented by a full hardware-based test, further developments are possible also in the simulation software. In particular, the future model will be implemented in a 3D space to make it more realistic. To this aim, it would be possible to join the Matlab® programming tool with a robot simulator software, like Gazebo or Vrep. Such a combination would allow to implement and test more complex STAR configurations, with a better graphic representation too. In addition, it would be possible to enrich the suite of sensors accommodated onboard and considered, through their correct modelling, in the code.

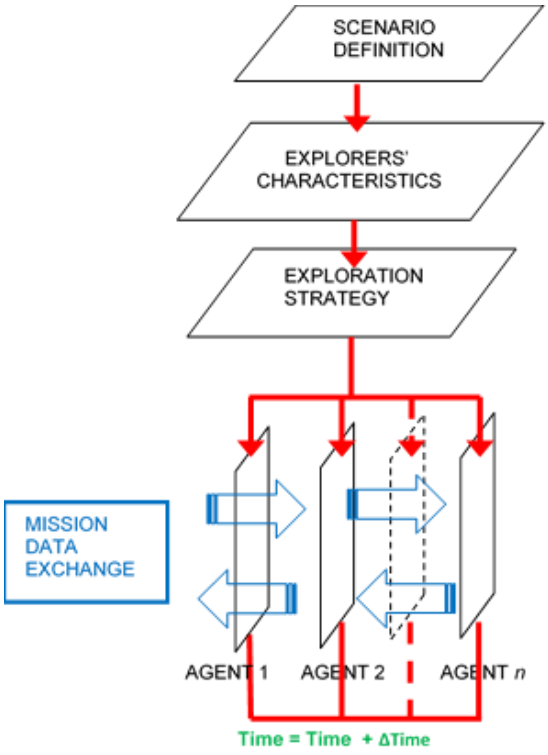


Figure 4-6:A general scheme for the simulation code implemented by means of parallel MATLAB processes.

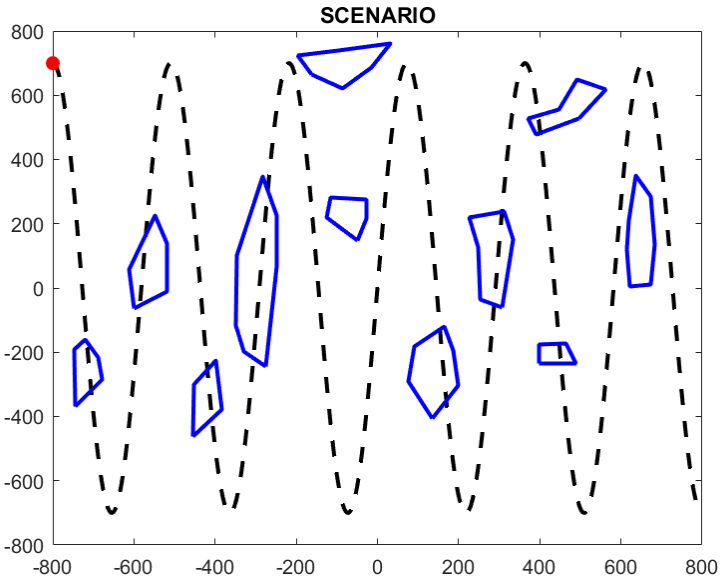


Figure 4-7: Example of a scenario spread with obstacles explored following a sinusoidal trajectory.

4.10 Estimation

The exploration success will obviously depend on the accurate navigation performed by the agents and by their capability in sharing correct findings from such a survey. In looking at more complex scenarios than the two agents' one considered before, it is likely that issues in communications and outages would significantly modify the case. In fact, the worsening in the link quality due to environmental conditions, to the increasing distance, to the proximity to obstacles or to obstacles' interposition in the line-of-sight will generally cause lack of communications from time to time. Such events could not be managed by simply waiting for the link recovery, as it could not happen at all (too large distance or obstacle) or ends up in excessively delaying the operations. The solution stays in the autonomous capabilities of the vehicles, that are capable to proceed on their own. At the same time such a capability should be conveniently related to the collective motion, so to do not vanish the benefit of a coordinated exploration. The inclusion of an estimation section to define in a probabilistic manner the evolution of the positions of other agents would solve such a problem. However, these filters need to manage the lack of measurements occurring from time to time, where outages in the line of sight or issues on the link interrupt the data flow.

A specific implementation for these filters can be obtained from classical Kalman filter loop (depicted in Figure 4-8, with a classical nomenclature) by considering that missing measurements can be ideally considered as measurements affected by infinite noise (R). Indeed, the Kalman gain can be computed as zero, and the update of the state will be clearly based only on the prediction. Some attention is required on the update of the covariance, that should be based on the predicted covariance and would therefore include the process noise (Q). Indeed, covariance will increase – as expected – as a result of every time step when lack of measurements will occur. As soon as measurements will be again available, the process to converge towards a solution will be effectively empowered.

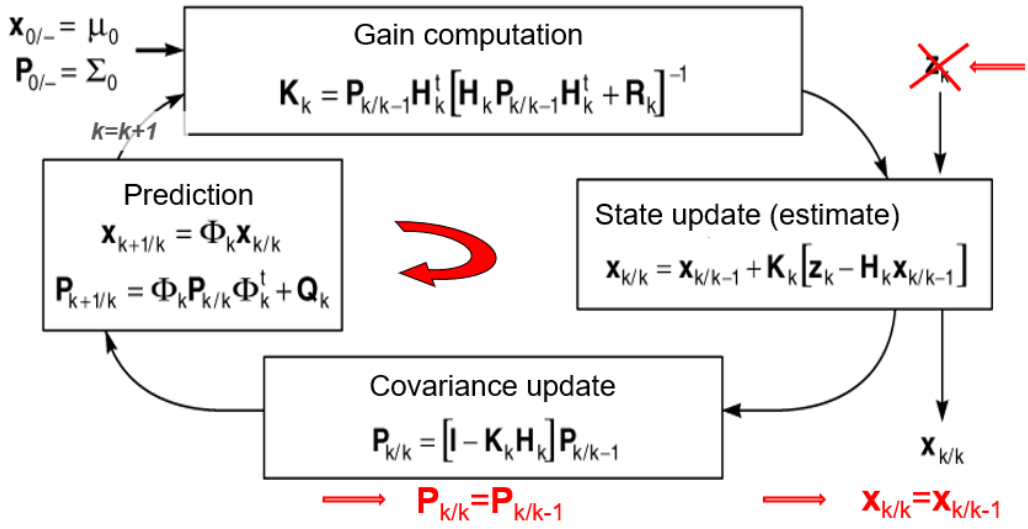


Figure 4-8: The Kalman loop, with the highlighted path in case of missing measurements.

Back in 1986 it has been proofed [62] as, in evaluating processes where measurements are missing from time to time (and in such a way that a certain probability could be associated to this lack of measurements), the traditional Kalman algorithm is not optimal anymore. However, it is possible to compute an upper and lower bound for the covariance matrix, indeed granting a confidence to the estimation process itself.

Notice that there is no specific reason to consider the dynamics and associated noises as gaussian ones, and indeed classical Kalman filter could not be the optimal solution. This is the reason why advanced or "beyond" Kalman techniques (specifically

unscented Kalman and particle filters, [63]) could be an interesting option: their applicability should be however evaluated with respect to the requested computational cost, usually far larger than classical Kalman. Significant effort simulations are needed to quantitatively analyse this problem, and these simulations - according to the research schedule - will be tackled in the next future.

Final remarks and future perspectives

The dissertation has presented the author contribution in a really promising and cutting-edge field like the one of the autonomous vehicles.

In fact, the continuous technological development, especially in the field of sensors, GPU, processing units and memories optimization, miniaturization and cost reduction, makes them attractive for a wide range of applications, from civil to military passing through the space ones.

At the same time, the requirement of high level of autonomy and safety, especially in human-involving applications, is addressing the researchers' efforts towards the development of ever more performing software algorithms in the branch of Artificial Intelligence as well as of Computer Vision.

Among the areas of interest for these technologies, the automotive one is nowadays the most attractive, also in view of its quite relevant economic size. The concept of electric self-driving cars is in fact polarizing the attention of both the most important car-makers and hi-tech companies' R&D departments. The core of the current research activities is mainly focused on the development of robust and efficient neural networks. By processing the input images coming from car's onboard cameras and sensors, these algorithms provide for an estimate of the road status (traffic, presence of pedestrians, obstacles and so on) and therefore for a guidance solution.

Tesla Motors Inc. is surely one of the most-advanced firms in the frame of self-driving cars. It is currently testing its *hardware 3.0 autopilot* involving an ASIC chip fully dedicated to the designed neural network in addition to some new features like the response to traffic lights or the autonomous driving in urban streets.

Another example of self-driving car is carried on by the Google LLC autonomous car, developed in the frame of the Waymo project. The goal of the project is the adoption of AI even to public transportation vehicles or trucks, ensuring high level of safety for passengers.

A different but quite important area of application that is worth to be mentioned (and actually can be considered the motivating and inspiring one for this work) is the space exploration, which can be considered pioneering for the autonomous vehicles. In fact, in the frame of planetary exploration, the need of motion and operative autonomy has raised since the beginning as the latency of radio signal did not allow a real time remote control of the robot. Starting from the first Martian platforms, where prototypes of vision-based guidance and navigation systems have been developed, the today's exploration robots have reached a very high level of autonomy. The Mars2020 (NASA, JPL) and the ExoMars Rosalind-Franklin (ESA, Roscosmos) rovers, both going to be launched by the end of 2020, are clear examples of the current state-of-the-art.

The Mars2020, developed in the frame of NASA Mars exploration program, is an evolved version of the Curiosity rover, still operating on the Mars ground. One of the Mars2020 improvements concerns the descent and landing system based on a method called *terrain-relative navigation*. A set of cameras will acquire ground images during the descent phase then a dedicated onboard software will compare these images with a preloaded reference map in order to compute the current position and the landing site. In case the rover is approaching a risky site, the descent software will autonomously provide for new and minor risk landing coordinates thus modifying the descent trajectory too. Another innovation that will be lead by the mission is the first prototype of an extraplanetary helicopter (*Mars Helicopter Scout*). The Scout will fly in the martian atmosphere with an autonomy of 90 s it will take off and land from the rover itself.

The Rosalind Franklin rover, developed in the frame of ExoMars program, will be the first European autonomous robot to explore the Mars planet. The rover will land on Mars within the lander Kozacok which will also cooperate with the rover in some of the planned scientific activities. As the NASA's rover, the ExoMars navigation system will be based on the stereo-vision principles, thanks to the PanCam equipped onboard. The main instrument carried on by the rover will be a special drill, which will be able to extract sample of 1cmx3cm of martian soil and then analyse it onboard.

Back to terrestrial applications, another field of interest for autonomous vehicles concerns the usage of both drones and ground robots in a swarm configuration. The range of application of such setup is very wide. An example is the DARPA OFFSET (*OFFensive Swarm-Enabled Tactics*) program, which involves an heterogeneous swarm (drones + ground vehicles) for military operations like surveillance, investigation and patrolling. Other programs are focusing on the potential use of drones for other kinds of application such as search and rescue, especially in hardly reachable environment, or delivery. In the latter case, the effort is focusing in developing high precision navigation systems and algorithms in order to guarantee the mission accomplishment even when the GPS coverage is low, like in urban canyons.

The analyses and tests carried on in the frame of this dissertation work could be considered as relevant to all the areas shortly depicted above. Within the limits of the hardware and tools available in a university lab, they have been mainly focussed to gain practical experience in the field, believed as a must for real engineering. For sure, this work helped in increasing the knowledge in a quickly evolving subject, as a preparation required for contributing in a close future to this exciting area.

References

[1] P. Corke, *Robotics, Vision and Control*, Springer, Berlin-Heidelberg, 2011.

[2] M.Larry, M. Maimone, A. Johnson et al., *Computer Vision on Mars*, Int. Journal of Computer Vision, 75:1 (2007), 67-92.

[3] G. Casonato, G. B. Palmerini, *Visual techniques applied to the ATV/ISS rendez-vous monitoring*, IEEE Aerospace Conference, Big Sky, USA, 2004.

[4] M. Sabatini, R. Monti R., P. Gasbarri, G. B. Palmerini, *Deployable space manipulator commanded by means of visual-based guidance and navigation*, Acta Astronautica 83 (2013), 27-43.

[5] T. Moons, L. Van Gool, M. Vergauwen, *3D Reconstruction from Multiple Images Part 1: Principles, Foundations and Trends in Computer Graphics and Vision*, 4:4 (2010) 287-404.

[6] Z. Zhang, *A flexible new technique for camera calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:11 (2000) 1330–1334.

[7] T. Borangiu, A. Dumitrache, *Robot Arms with 3D Vision Capabilities*, in: E. Hall (Ed.), Advances in Robot Manipulators, InTech, Rijeka-Shangai, 2010, 503-514.

[8] M. Sabatini, P. Gasbarri, G. B. Palmerini, *Coordinated control of a space manipulator tested by means of an air bearing free floating platform*, paper IAC-16-C1.1.3, 67th International Astronautical Congress, Guadalajara (Mexico), 2016.

[9] R. Szabo, A. Gontean, *Controlling a Robotic Arm in 3D with Stereo Vision*, TELFOR2013 21st Telecommunication Forum, Belgrade (Serbia), 2013, 916-919.

[10] R. Szabo, A. Gontean, *Robotic Arm Control with Stereo Vision Made in LabWindows/CVI*, 38th International Conference on Telecommunications and Signal Processing (TSP), Prague (Czech Republic), 2015.

[11] W. Jiang, Z. Wang, *Calibration of visual model for space manipulator with a hybrid LM–GA algorithm*, Mechanical Systems and Signal processing, 66-67 (2016) 399-409.

[12] M. Sabatini, R. Monti, P. Gasbarri, G. B. Palmerini, *Adaptive and robust algorithms and tests for visual-based navigation of a space robotic manipulator*, Acta Astronautica, 83(2013) 65-84.

[13] P. Gasbarri, M. Sabatini, G. B. Palmerini, *Ground tests for vision based determination and control of formation flying spacecraft trajectories*, Acta Astronautica, 102 (2014) 378-391.

[14] M. Carpentiero, *Design and test of a guidance and stereo-based navigation system for an autonomous rover,* Master thesis in Space and Astronautical Engineering, Sapienza Università di Roma, 2016

[15] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, *SURF: Speeded Up Robust Features*, ETH Zurich, Katholieke Universiteit Leuven, 2006

[16] S. Leutenegger, M. Chli, R. Siegwart. *BRISK: Binary Robust Invariant Scalable Keypoints*, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011.

[17] G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60:2 (2004), 91-110.

[18] R. Edward, T. Drummond, *Machine learning for high-speed corner detection*, European Conference on Computer Vision, 2006, 430–443.

[19] P. Porwik, A. Lisowska, *The Haar–Wavelet transform in digital image processing: its status and achievements*, Machine Graphics&Vision 13 (2004) 79-98.

[20] Maki, J, Bell, F., Herkenhoff, K. et al. "*Mars Exploration Rover Engineering Cameras*", Journal of geophysical research, vol. 108, no. e12, 2003

[21] Larry, M., Maimone, M., Johnson, A. et al., "*Computer Vision on Mars*", Int. Journal of Computer Vision, 67-92, 2007.

[22] G. Casonato, G. B. Palmerini, "*Visual techniques applied to the ATV/ISS rendez-vous monitoring*", IEEE Aerospace Conference, Big Sky, USA, 2004

[23] Fiore, A. "*Progetto e realizzazione di un rover a controllo remoto*", Msc. dissertation, Aeronautics and Astronautics Dept., Università di Roma, La Sapienza, Rome, Italy, 2012

[24] Moons, T., Van Gool, L., Vergauwen, M., "*3D Reconstruction from Multiple Images Part 1: Principles*", Foundations and Trends® in Computer Graphics and Vision, Vol. 4, No. 4, pp. 287-404

[25] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., "*SURF: Speeded Up Robust Features*", Computer Vision and Image Understanding, Vol. 110, Iss 3, pp 346–359, 2008

[26] S. Leutenegger, M. Chli, R. Siegwart. "*BRISK: Binary Robust Invariant Scalable Keypoints*", Proceedings of the IEEE International Conference, ICCV, 2011

[27] McInnes, C.R, "*Potential function methods for autonomous spacecraft guidance and control*", Proceedings of the AAS/AIAA Astrodynamics Conference, Halifax, Canada, pp. 2093-2109, 1996

[28] Carpentiero, M.," *Progetto e realizzazione di un sistema di guida e navigazione stereoscopica per un rover spaziale*", Msc. dissertation, Aeronautics and Astronautics Dept., Università di Roma, La Sapienza, Rome, Italy, 2016

[29] M.Carpentiero, M.Sabatini, G.B.Palmerini, "*Capabilities Of StereoVision Systems For Future Space Missions*", Proceedings of the 67thInternational Astronautical Congress, 2016

[30] J.A. Crisp, M. Adler, J.R. Matijevic, S.W. Squyres, R.E. Arvidson, D.M. Kass, "*Mars Exploration Rover mission*", Journal of Geophysical Research E: Planets, Vol. 108, Issue 12, 2003, pp. ROV 2-1 - ROV 2-17.

[31] P.B.Sujit, S.Saripalli, J.B. Sousa "*Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicles*", IEEE Control Systems, vol.34, No.1, pp.42-59 (2014).

[32] A. Sharma, D.D. Massey, A. Taneja, "*Horizontal Gradients of Traffic Related Air Polluttants Near a Major Highway in Agra, India*", Indian Journal of Radio & Space Physics, Vol. 38, December 2009, pp. 338-346.

[33] L. Gugliermetti, M. Sabatini, G. B. Palmerini and M. Carpentiero, "*Air quality monitoring by means of a miniaturized sensor onboard an autonomous wheeled rover*", IEEE International Smart Cities Conference (ISC2), Trento, 2016.

[34] L. Gugliermetti, B. Mattoni, F. Nardecchia, F. Bisegna, C. Galati, "*Home smart grid device for energy saves and failure monitoring*," Proceedings of the IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC), Rome, 2015, pp. 671-676.

[35] S. Leutenegger, M. Chli, R. Siegwart "*BRISK: Binary Robust Invariant Scalable Keypoints*", Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011.

[36] G. Lowe, "*Distinctive image features from scale-invariant keypoints*", International Journal of Computer Vision, Vol.60, No.2, pp.91-110, 2004.

[37] M. Sabatini, G.B. Palmerini "*Collective control of spacecraft swarms for space exploration*", Celestial Mechanics and Dynamical Astronomy, Vol.105, pp.229-244, 2009.

[38] M. Garzon, J. Valente, D. Zapata, A. Barrientos, "*An aerial–ground robotic system for navigation and obstacle mapping in large outdoor areas*", Sensors, Vol.13, pp. 1247-1267, doi:10.3390/s130101247, 2013.

[39] D. Krishna, H. Yadav, V. Salvaraj, S. Krithivasan, "*Aerial assisted path planning for terrestrial rover without complete environment map*", 2nd International Conference on Innovations in Information, Embedded and Communication systems (ICIIECS), 2015.

[40] J. H. Kim, J. W. Kwon, J. Seo, "*Multi UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV*", Electronic Letters, Vol. 50, No. 20, pp. 1431-1432, 2014.

[41] M. Cognetti, G. Oriolo, P. Peliti, L. Rosa, P. Stegagno, "*Cooperative control of a heterogeneous multi-robot system based on relative localization*", IEE/RSJ International Conference on Intelligent Robots and Systems, 2014.

[42] E. Bonabeau, M. Dorigo, G. Theraulaz, "*Swarm Intelligence*", Oxford University Press, 1999.

[43] M. Sabatini, G.B. Palmerini, "*Collective control of spacecraft swarms for space exploration*", Celestial Mechanics and Dynamical Astronomy. 163 (2010) 51–59.

[44] D.Gaines et al., "*Expressing Campaign Intent to Increase Productivity of Planetary Exploration Rovers*", PlanRob ICAPS workshop, 2017.

[45] M. Sabatini, G.B. Palmerini, P. Gasbarri, "*Control laws for defective swarming systems*", Advances in Astronautical Sciences, 153 (2015) 749-768.

[46] P.S. Schenker et al., "*The expanding venue and persistence of planetary mobile robotic exploration — new technology concepts for Mars and beyond*", Proceedings of SPIE 5267 (2003) 43-58.

[47] Y. Nevatia et al., "*Safe long-range travel for planetary rovers through forward sensing*", Proceedings of the 12th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA), Noordwijk, the Netherlands, 2013.

[48] S. Sand et al., "*Swarm Exploration and Navigation on Mars*", International Conference on Localization and GNSS (ICL-GNSS), Turin, Italy, 2013

[49] M. Carpentiero, L. Gugliermetti, M. Sabatini, G.B. Palmerini, "*A swarm of wheeled and aerial robots for environmental monitoring*", ICNSC 2017 - 14th IEEE International Conference on Networking, Sensing and Control, Calabria, Italy, 2017, 16 – 18 May.

[50] P.E. Hart, N.J. Nilsson, B. Raphael, "*A Formal Basis for the Heuristic Determination of Minimum Cost Paths*", IEEE Transactions on Systems Science and Cybernetics 4:2 (1968) 100–107.

[51] C. Trevai, N. Fujii, J. Ota, T. Arai, "*Multiple mobile robot exploration and patrol strategy using a self-organizing planner based on a reaction-diffusion equation on a graph*", J. Rob. Mechatron. 20 (2008) 24–37.

[52] A.Marino, L. Parker, G. Antonelli, F. Caccavale, "*Behavioral control for multi-robot perimeter patrol: a finite state automata approach*", IEEE Int. Conf. Rob. Autom., Kobe, Japan, 2009, May 12-17.

[53] M. Carpentiero, M. Sabatini, G.B. Palmerini, "*Swarm of autonomous rovers for cooperative planetary exploration*", paper IAC-17-D1.2.9, 68th International Astronautical Congress, Adelaide, 2017.

[54] M. Carpentiero, M. Sabatini, G.B. Palmerini, "*Communication and Navigation Architecture for Planetary Exploration Carried-On by a Swarm of Mobile Robots*", paper IAC-18-B2.1.11, 69th International Astronautical Congress, Bremen, 2018.

[55] L. Bayindir, "*A Review of Swarm Robotic Tasks*", Neurocomputing 172 (2016) 292–321.

[56] Y. Tan, Z. Zheng, "*Research Advance in Swarm Robotics*", Defence Technology 9 (2013) 18–39.

[57] F. Ducatelle, G.A. Di Caro, C. Pinciroli, F. Mondada, L. Gambardella, "*Communication assisted navigation in robotic swarms: self-organization and cooperation*", IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco (CA), USA, 2011, 25-30 September, 4981–4988.

[58] L. Silva Junior, N. Nedjah, "*Efficient Strategy for Collective Navigation Control in Swarm Robotics*", Procedia Computer Science 80 (2016) 814–823.

[59] Z. Bouhanna, C.P. Bridges, "*Relative Range Estimation using SDR for Space Traffic Management*", Proceedings of the IEEE Aerospace Conference, Big Sky (MT), USA, 2018, 12–16 October.

[60] T. Lassen, "*Long-range RF communication: Why narrowband is the de facto standard*", Texas Instruments Application Notes (2014).

[61] C. Ho, N. Golshan, A. Kliore, "*Radio Wave Propagation Handbook for Communication on and Around Mars*", JPL Publication 02-5 (2002).

[62] H.M. Faridani, "*Performance of Kalman Filter with Missing Measurements*", Automatica 22 (1986) 117–120.

[63] B. Ristic, S. Arumpalam, N. Gordon, "*Beyond Kalman Filter*", Artech House, 2004.

Ringraziamenti

Desidero ringraziare infinitamente il mio Supervisor, il Prof. Giovanni Palmerini, per avermi spronato, incoraggiato ed aiutato a raggiungere questo traguardo nonostante le mille vicissitudini, lavorative e non, che si sono presentate durante il percorso. Spero, un giorno, di essere in grado di affrontare le infinite sfide, lavorative e di vita, col suo stesso pragmatismo nonché grinta e professionalità.

Un ringraziamento particolare anche al Prof. Paolo Gasbarri, al dott. Marco Sabatini – nonostante le nostre vedute divergenti sul Capitano Daniele De Rossi – e ai miei colleghi e amici del GN-Lab grazie ai quali questo percorso è stato meno faticoso.

Grazie poi a Francesca, la mia dolce metà, che ha supportato e incoraggiato le mie scelte lavorative, invitandomi a credere sempre in me stesso e a non abbattermi mai. Sono davvero fortunato a poter condividere con te questi anni importantissimi!

Infine, ringrazio la mia famiglia e la famiglia di Francesca che, nonostante la notevole distanza che ci separa, non mancano mai di farci sentire il loro affetto. Senza di voi sarebbe tutto più complicato.