69th International Astronautical Congress, (IAC), Bremen, Germany, 1-5 October 2018.

IAC-18,D1,3,2,x48350

Ground testing of vision-based GNC systems by means of a new experimental facility

Paolo Lunghi[a]*, Luca Losi[a], Vincenzo Pesce[a], and Michèle Lavagna[a]

[a]*Department of Aerospace Science and Technology, Politecnico di Milano, via La Masa 34, 20156 Milano, Italy*, paolo.lunghi@polimi.it, luca.losi@polimi.it, vincenzo.pesce@polimi.it, michelle.lavagna@polimi.it
* Corresponding Author

Abstract

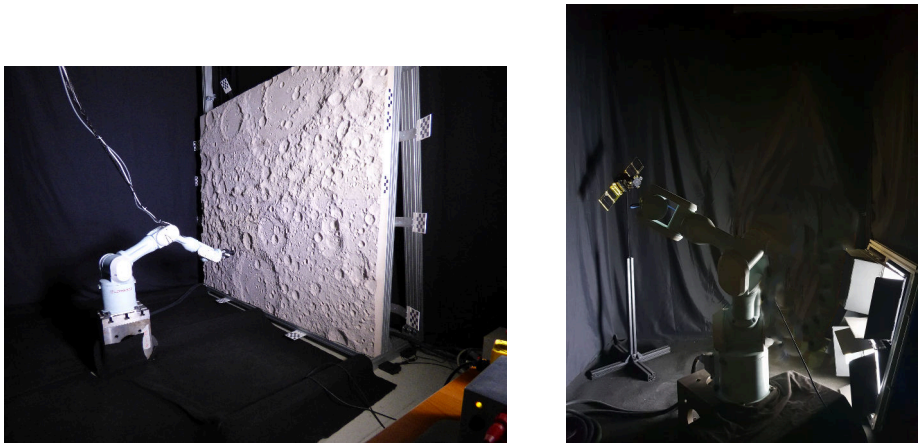
A new experimental facility for the testing of autonomous vision-based GNC algorithms, recently developed at Politecnico di Milano, Aerospace Science and Technology Department (DAER) is presented. The system core is a 7DoF robotic arm that moves a navigation camera reproducing the spacecraft dynamics around a target. Different targets are provided: a lunar terrain diorama for planetary landing simulation; a satellite mock-up for proximity GNC for rendezvous, docking and on-orbit servicing; an asteroid model to simulate proximity flight around small celestial bodies. The system can follow predefined trajectories, operating as a simple dataset generator; software-in-the-loop simulations are possible by coupling GNC with the spacecraft dynamics simulation. Different scale factors can be selected, simulating different phases of landing or close approach maneuvers. The system is designed to be upgradeable to perform real time hardware-in-the-loop simulations, with also the possibility to include additional sensors (eg. IMU, laser altimeter). The suite of autonomous GNC algorithms under study at DAER, and their preliminary experimental tests in the new facility are shown and discussed.
Keywords: *Vision-Based Autonomous GNC, Testing facility, Relative navigation, planetary landing*

1. Introduction

Spacecraft autonomy represent a vital technology for the enhancement and success of future space missions. A field in which a high degree of autonomy would give the major benefits is Guidance, Navigation and Control: possibility to land on unknown surfaces without prior landing spot knowledge, to perform divert maneuvers and autonomous navigation around different targets would augment significantly operative possibilities for space exploration, on-orbit servicing, formation flight and active debris removal. Within this context, Autonomous GNC chain exploiting one or multiple camera as main sensor is a promising technology, but the high robustness required for the aforementioned applications implies to perform extensively test and validation campaigns. This is when analog facilities come into play: they offer the possibility to simulate different spacecraft maneuvers in a scaled environment, supplying repeatable and controllable datasets reliable enough to increase the Technology Readiness Level (TRL) of the AGNC algorithms up to 6.

A new experimental facility dedicated to vision-based AGNC algorithm validation fully developed at Politecnico di Milano, Aerospace Science and Technology Department (DAER) is here presented. The system is composed by a 7DoF robotic arm that moves a navigation camera reproducing the spacecraft dynamics around a target. Different targets are provided: a lunar terrain diorama for planetary landing simulation; a satellite mock-up for proximity GNC for rendezvous, docking and on-orbit servicing; an asteroid model to simulate proximity flight around small celestial bodies. Realistic, controllable illumination is provided by means of an adjustable small beam spotlight, while a dedicated dimming system excludes undesired light, reflections, and diffuse illumination.

The system is designed for flexibility: it can follow predefined trajectories, operating as a simple dataset generator; software-in-the-loop simulations are also possible by coupling AGNC with the spacecraft dynamics simulation. Different scale factors can be selected, simulating different phases of landing or close approach maneuvers. The system is upgradable to perform real time hardware-in-the-loop simulations, with also the possibility to include additional sensors (eg. IMU, laser altimeter). To obtain a proper validation of the algorithms, the ground truth of the adopted trajectory shall be known with an accuracy at least one order of magnitude better than the one achieved by those ones: in practice, the actual required value depends on the scale factor used in the simulation. To reach this objective, the shape of the adopted target is reconstructed through dense matching techniques: several images are taken from positions all around the target, relevant features are extracted and matched across different frames to identify the camera pose in each view. This information is exploited to build a dense point cloud



(a) Optical navigation for landing. (b) In-orbit relative navigation.

Fig. 1: Experimental facility running test under different configurations.

model with a sub-millimeter accuracy; laser measures are then taken at selected points to validate the results, obtaining a reliable low cost alternative to more expensive systems. Two different algorithms, developed at DAER, have been tested: a feature-tracking relative navigation system, and a hazard detection system based on artificial neural networks. Preliminary results are shown and discussed.

2. The facility

Since the scarce availability of complete real landing imagery datasets, vision-based algorithms development relies widely on synthetic images. However, despite being a powerful tool for preliminary validation, the high computational cost required for rendering makes difficult close loop testing and hardware-in-the-loop (HIL) impossible. To validate the approach proposed, experiments are therefore necessary. Moreover, the whole navigation system performance can be assessed only connecting the composing parts together, to verify mutual influences. This is the reason which has led to development of analog facilities: they can supply repeatable and controllable data, and allow for HIL testing activities which can bring the TRL of the AGNC algorithms up to 6.
The PoliMI-DAER's facility is composed by a robotic arm carrying a suite of sensors to simulate lander dynamics, control and test computers, an illumination system with dedicated controlled environment, and different camera targets: a machine milled planetary mock-up, a 3D-printed scaled spacecraft and a 3D-printed scaled model of asteroid Itokawa. The setup is shown in Fig. 1a. The goal is to reproduce landing or relative in-orbit maneuvers over a scaled but realistic environment. The system is designed to verify either hardware and software breadboards up to TRL 4, with the possibility to update the system in the future to carry out also real-time hardware-in-the-loop simulations to qualify GNC technologies up to TRL 5.

2.1 Robotic arm

The robotic arm represents the core of the analog facility and allows to properly simulate the spacecraft motion and dynamics for the different mission scenarios considered. A 7-DoF Mitsubishi PA10-7c arm is adopted for this task: it is servoactuated and sensorized with brushless resolvers, able to carry on its tip a maximum load of $10\,\mathrm{kg}$. The robot's hardware and software open architecture allows to control and modify any aspect of its behaviour as well as to include new sensor information to the control loop. The arm is shown in Figure 2, while an overview of the technical specifications is given in Table 1. The robot operative envelope has 1m radius

Fig. 2: Mitsubishi PA10-7c.

Robotic arm	Technical specifications
Mitsubishi PA10-7c	Dedicated controller
	7 Degrees of Freedom
	Real time ready

Table 1: Robotic arm technical specification overview.

approximately and allows for spacecraft motion simulation under different scaled trajectories. While for the landing simulations a single motion is carried by the arm, for in-orbit relative navigation simulations the arm is controlled to reproduce the superimposed motion of both the chaser and target spacecraft. At its end effector, the robot carries the sensor suite, that is made up by a navigation camera and is upgradable with other sensors; such as range finder simulating a laser altimeter or Inertia Measurements Unit (IMU).

2.2 Navigation camera

The navigation camera is the main sensor of the AGNC chain under development. It has to be representative of a possible space hardware. Being of interest the navigation around targets from the final approach phase, a wide angle camera shall be used, with high resolution not needed and fixed focus. A Chameleon 3 camera produced by PointGrey has been adopted for the facility. Its characteristics are shown in Table 2; camera is fully tunable and, being specifications better than the requirements imposed, a configuration with gray-scale images, 1024x1024 resolution and lower frame rate is adopted for the simulations. Moreover, camera is fully programmable from computer with dedicated C++ libraries, a fundamental characteristic to build an interface with the developed algorithms.

Technical specification	
Resolution	1280x1024
Frame rate	149
megapixels	1.3
Chroma	color/gray-scale
ADC	10 Bit
Sensor format	1/2
Focal length	6 mm
Field of view	43.5 ^{o}

Table 2: Navigation camera technical specification.

2.3 Illumination system

Ensuring the proper illumination and environmental conditions is a fundamental task to obtain realistic images with the camera and consequently to proper validate the AGNC algorithm under test. Diffuse light shall be avoided, being the operational conditions under investigation all non-atmospheric. To this aim, a dedicated dark room as been built around the facility exploiting black courtains and dark floor cover which prevent from light reflection. A dedicated lighting shall then simulate the sun illumination: a dedicated LED array with narrow beam angle and 5700K light temperature is therefore exploited, whose main characteristics are expounded in Table 3

Technical specifications	
High CRI spotlight	
Light temperature	5700 K
Beam angle	60^o
LED array dimension	1024x1024

Table 3: Illumination technical specifications.

2.4 Planetary mock-up

A planetary mock-up for landing simulations has been fully realized at the PoliMI-DAER laboratories representing a portion of the Lunar surface from its far side. Urethane foam has been chosen as material for the diorama due to its surface finish that already yields the correct optical properties and because of its great workability. The overall dimension of the diorama, 2400 x 2000 mm, was selected to exploit the full envelope of the robotic arm and a maximum field of view of 60^o for the landing camera. The model is divided in 8 tiles, each measuring 1200 x 500 mm. The Digital Elevation Model (DTM) has been selected from the GLD-100 NASA LROC dataset, in order to have mixed terrain features, from plains to rough slopes. The model was enriched with low scale detail, non visible due to the limited DEM resolution, by the addiction of small craters, boulders and fractal noise [1]. Selected scale factor has driven the facility manufacturing requirements. It is assumed that the hazard detection routine from the AGNC chain starts at a maximum altitude of 2000 m. Given the robotic arm operative envelope of 1 m of radius, a maximum scale factor of 2000:1 has been considered. The target accuracy for a navigation algorithm at touchdown is in the order of 10 m, which corresponds to 5 mm in the scaled environment. To have a resolution of the terrain at least one order of magnitude greater than the landing accuracy, a resolution of at least 0.5 mm is therefore required. The scale factor can be then adapted to simulate closer range maneuvers with higher details, due to the characteristic fractal structure of the Moon surface.

2.5 Satellite mock-up

A scaled satellite mock-up has been realized in the house at PoliMI-DAER premises for in-orbit relative navigation around uncooperative target simulations. Since the main driver for the design is the model appearance, a reference spacecraft had to be chosen. In particular, it had to be representative of a wide class of spacecraft without presenting too many details that would add complexity during the mock-up assembly. The choice fell on the Solar and Heliospheric Observatory (SOHO) spacecraft. The model has been 3D printed with a Bits from Bytes BFB-3000 available at Polimi DAER Laboratory. PLA material has been used to build the mock-up. The model dimensions are maximized to reduce the scaling of the experiment. In this way, the accuracy to validate the algorithms can be reached. A spray-plaster of a gray color was applied twice on the model and we sanded it down with sandpaper with different grit size. A spray acrylic black paint, usually used for modeling, was applied twice on the surface to reproduce the black surface of the spacecraft. The details were adjusted by a thin brush. Gold aluminum foils were used to reproduce the classical thermal protection of spacecraft, made of Multi-Layer Insulation (MLI). Finally, we used some leftover of real spacecraft solar cells from Azurespace. The final spacecraft mock-up is shown in Fig. 3.

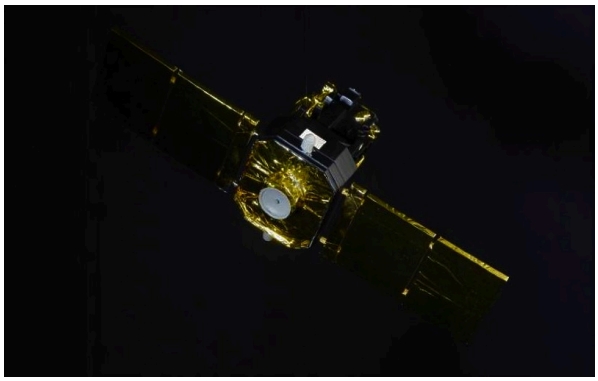


Fig. 3: Satellite Mock-up

2.6 Asteroid mock-up

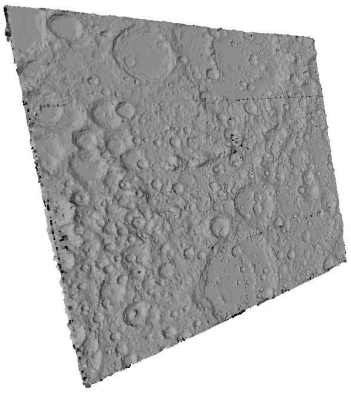
In a similar way, a 3D-model of the asteroid Itokawa has been printed. Also in this case, a spray-plaster of a gray color was applied on the model. A spray acrylic grey paint was applied to the surface to enhance the optical characteristics of the asteroid. A scale of $1:2300$ was used to comply with the printer's bed constraint.

3. Facility Calibration procedure

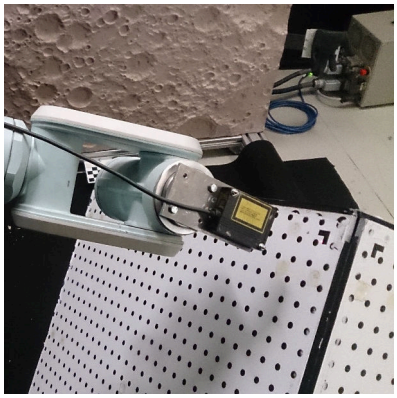
To effectively test navigation algorithms on an analog target models, the facility shall be able to reconstruct the camera trajectory with a precision at least one order of magnitude better than the expected precision of the navigation algorithm under test. Then, the actual required value depends on the actual scale factor adopted for the specific simulation. Considering for example a the Lunar landing scenario, being the desired accuracy in the development of navigation algorithms up to $10\,\mathrm{m}$ 3σ, which corresponds to $5\,\mathrm{mm}$ at the target scale factor 1:2000, the target accuracy in calibration shall be better than $0.5\,\mathrm{mm}$. The calibration process consists in 3 different tasks:

Target model reconstruction. The actual target model is generally different from the numerical model, due to imperfections during the production process. Dense matching method has been selected to perform the shape reconstruction: several photos from different angles are taken; than, structure from motion algorithms are used to obtain the camera pose for each image. Finally, dense cloud point models are obtained by triangulation of optical features between different frames. For the Lunar terrain mock-up, reconstruction is performed for each constitutive tile, and models are eventually assembled by means of Iterative Closest Point (ICP) algorithms, generating the final model for the whole diorama. The final accuracy is estimated by the features reprojection error, which for the Lunar diorama resulted well below the 0.5 mm threshold. An image of the whole reconstructed diorama assembly is shown in Fig. 2a.

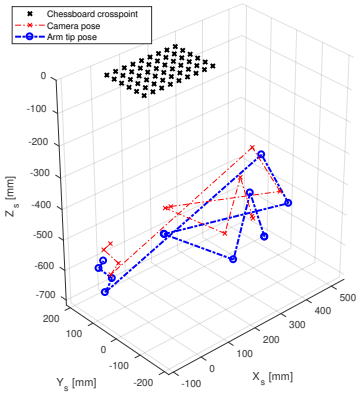
Camera vs Robotic arm calibration. Vision-based navigation algorithms determine the position of the camera optical reference frame with respect to the surrounding environment. In general, this frame is not coincident nor aligned with the camera body and its required to establish its position with respect to the robotic arm's tip. This is achieved by the acquisition of an image set depicting an object of known geometry from different distances and angles. The calibration target is dotted with a know pattern of visual markers, allowing the identification of the camera intrinsic parameters and distortion, as well as the camera extrinsic parameters, which contain information on the camera relative pose for each frame. The absolute position of the robotic arm with respect to the calibration target is determined through measures by a laser range finder mounted on the arm's tip. The high quality of the measures by both the laser range finder and the robotic arm gives a final accuracy better than $\pm 0.1\,\mathrm{mm}$ 3σ The comparison of the tip poses along the calibration path with respect to the camera poses in the acquired images allows to identify the rigid rototransla-



(a) Diorama digital reconstruction by means of Dense Matching.



(b) Robotic arm positioning with respect to calibration target.



(c) Path comparison between optical frame and robotic arm's tip.

Fig. 4: Facility calibration process.

tion between the end effector and the optical reference frame. The process is depicted in Fig. 2b and 2c.

Robotic arm vs target calibration. The last step of the facility calibration is the determination of the robotic arm pose with respect to the target model. This is obtained again with the laser range finder: a low resolution point cloud of the model is computed by measuring multiple points on the target; the rototranslation between the robotic arm and the model is then computed by aligning the calibrated dense model with the sparse point cloud obtained by laser measures. The residuals after the alignment give also an independent estimation of the accuracy of the calibration.

3.1 Performances

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo

facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

4. Developed algorithms

Following, the autonomous GNC suite under development at PoliMi-DAER, is briefly presented. Three main blocks compose the system: autonomous guidance, hazard detection, and vision-based navigation. Whenever available, preliminary experimental results are presented as well.

4.1 Autonomous Guidance

Autonomous guidance constitutes the link between relative navigation and hazard detection in a full HDA GNC chain. Once a safe landing site is selected, the system must compute a new feasible trajectory toward the new target. A fuel-optimal solution is sought to maximize the attainable landing area in subsequent target updates, that may be required as soon as smaller terrain features become observable. Here the basic structure of the algorithm and some test results are summarized: for a detailed description, see [2] and [3].

A planetary landing is characterized by fast dynamics. The expected time of flight of the approach phase in which HDA tasks take place is in the order of $1\,\mathrm{min}$, and the mass is supposed to significantly change during the maneuver. In this case, distances, for both downrange and altitude, are small compared to the planet's radius; thus, the assumption of a constant gravity field with flat ground is appropriate. Aerodynamic forces are neglected: the effects of the possible presence of atmosphere (especially for low densities, as in the case of Mars) could be omitted due to the relative low velocity (on the order of $100\,\mathrm{m\,s^{-1}}$) and the associated forces can be treated as disturbances [4]. The translational dynamics of the spacecraft are expressed in a ground reference system as:

$$\dot{\mathbf{r}} = \mathbf{v} \qquad \dot{\mathbf{v}} = \frac{\mathbf{T}}{m} + \mathbf{g} \qquad \dot{m} = -\frac{T}{I_{\mathrm{sp}} g_0} \tag{1}$$

where $\mathbf{r} = [x, y, z]^T$, x is the altitude, y is the downrange direction and z is the cross-range; $\mathbf{g}$ is the constant acceleration of gravity vector of the planet, I_{sp} the specific impulse of the main engine, and g_0 the standard gravity acceleration on Earth. The thrust net magnitude is indicated with $T = \|\mathbf{T}\|$.

The thrust vector acts as the control variable. The mass equation is linked to the control acceleration by the thrust-to-mass ratio $\mathbf{P}$:

$$\mathbf{P} = \mathbf{T}/m = \dot{\mathbf{v}} - \mathbf{g} \tag{2}$$

Then, the mass equation in system (1) can be rewritten as a first order linear ordinary differential equation:

$$\dot{m} = -\frac{P}{I_{\mathrm{sp}} g_0} m \tag{3}$$

where $P = \|\mathbf{P}\|$. The states $\mathbf{r}_0$, $\mathbf{v}_0$ and m_0 at the initial time t_0 are supposed to be known. At the end of the maneuver, at time t_{f}, the final states $\mathbf{r}_{\mathrm{f}}$ and $\mathbf{v}_{\mathrm{f}}$ are constrained to assume fixed values. Then, the optimal guidance problem is to find a control profile $\mathbf{T}(t)$ to bring the system from the initial to the target final states, compatibly with all the constraints imposed by the actual system architecture. For sake of simplicity is considered $t_0 = 0$. The main thruster is assumed to be tightly connected to the spacecraft body. Then, the thrust vector depends only on the attitude of the spacecraft, and on the thrust magnitude T. At the beginning of the maneuver, the attitude is assumed to be known. Then, the initial acceleration is function only of the initial thrust magnitude. At the end of the maneuver, the lander is required to be aligned with the local vertical on the Target Landing Site: in case of flat surface, this condition reduces to impose null horizontal acceleration. A total of 17 boundary constraints are then available for position, velocity and acceleration components: 6 on initial states, 3 on initial acceleration (function of initial thrust magnitude), 6 on target final states and 2 on the final acceleration due to

the final attitude requirements.

$$\begin{aligned} \mathbf{r}(0) &= \mathbf{r}_0 & \mathbf{r}(t_\mathrm{f}) &= \mathbf{r}_\mathrm{f} \\ \mathbf{v}(0) &= \mathbf{v}_0 & \mathbf{v}(t_\mathrm{f}) &= \mathbf{v}_\mathrm{f} \\ \dot{\mathbf{v}}(0) &= \mathbf{f}(T_0) & \dot{\mathbf{v}}(t_\mathrm{f}) &= [\mathit{free}, 0, 0]^T \end{aligned} \tag{4}$$

The 3 components of the acceleration can be expressed in a polynomial form. The minimum order needed to satisfy the boundary constraints is 2 for the vertical axis, 3 for the horizontal components:

$$\dot{\mathbf{v}}(t) = \begin{bmatrix} \dot{v}_\mathrm{x} \\ \dot{v}_\mathrm{y} \\ \dot{v}_\mathrm{z} \end{bmatrix} = \begin{bmatrix} \dot{v}_{0\mathrm{x}} + c_{1\mathrm{x}}t + c_{2\mathrm{x}}t^2 \\ \dot{v}_{0\mathrm{y}} + c_{1\mathrm{y}}t + c_{2\mathrm{y}}t^2 + c_{3\mathrm{y}}t^3 \\ \dot{v}_{0\mathrm{z}} + c_{1\mathrm{z}}t + c_{2\mathrm{z}}t^2 + c_{3\mathrm{z}}t^3 \end{bmatrix} \tag{5}$$

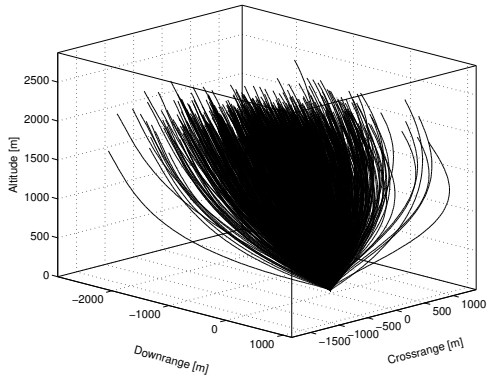
By integrating the acceleration twice and applying the boundary conditions, the trajectory is parametrized in terms of time-of-flight t_f and initial thrust magnitude T_0, that are considered as optimization variables. Once the acceleration profile is defined, the thrust-to-mass ratio can be obtained from Eq. (2) and the thrust profile is:

$$\mathbf{T} = m\mathbf{P} \tag{6}$$

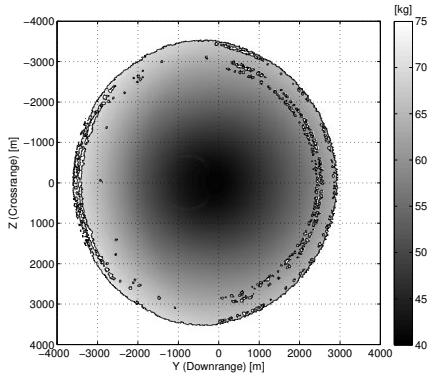
where the mass profile is obtained by solving Eq. (3). From the thrust vector a complete guidance profile, in terms of Euler angles and thrust magnitude, is easily obtained. In addiction to boundaries constraints, the system is subject also to box and path constraints. Box constraints delimit the space of the optimization variables: the initial thrust magnitude is bounded to the thrust actually available on-board, while the time-of-flight must lie between its lower and upper limit.

Path constraints need to be satisfied at every time instant during the landing: pseudospectral techniques allow to evaluate the path constraints discretely at Chebyshev-Gauss-Lobatto (CGL) points, with easy derivation and integration [5]. The angular velocity of the spacecraft is limited by the actual control torques M_{Cmax} given by the Attitude Control System (ACS). Moreover, the spacecraft is required to remain in a cone pointed at the target and defined by the maximum slope angle δ_{max}, with the dual purpose to assure that the the lander does not penetrate the ground, even in presence of bulky terrain features near the landing site, and to limit the angle of view on the target, crucial for the performances of vision-based navigation systems [6, 7].

The optimization could be solved with any non-linear programming (NLP) solver: the choice of this solver has a huge impact over the final convergence properties and computational time. A dedicated optimization algorithm based on Taylor Differential Algebra (DA) was developed. DA techniques were devised to attempt solving analytical problems through an algebraic approach [8]. Instead to be modelled as simple real numbers, quantities



(a) Large scale diversion maneuver simulation (from Ref. [3]).



(b) Attainable area and fuel consumption comparison (from Ref. [3]).

Fig. 5: Adaptive Guidance Monte Carlo Simulation.

are represented as their Taylor expansion around a nominal point. In this way, DA variables carry more information rather than their mere punctual values. Computing the objective function as a DA variable it is possible to estimate its sensitivity to the variation of the optimization variables. Then through the operation of map inversion, stationary points of the function can be found in a restricted number of iterations, using only simple algebraic computation between Taylor coefficients. A detailed discussion about the developed optimization algorithm is included in [3]. To estimate the performances of the proposed algorithm, a Monte Carlo (MC) simulation of a realistic case of lunar landing was carried out. Dispersion is applied to initial position, velocity, attitude, amount of fuel on board, specific impulse, spacecraft moment of inertia, available thrust and gravity acceleration. Figure 5a shows the trajecotry obtained in a MC simulation of a large scale diversion maneuver from a nominal altitude of $2000\,\mathrm{m}$: in all the cases, the ordered diversion was found feasible by the guidance algorithm. These results were compared with the solutions computed with a general-purpose non-linear optimization software (SNOPT): in the worst case observed, the dif-

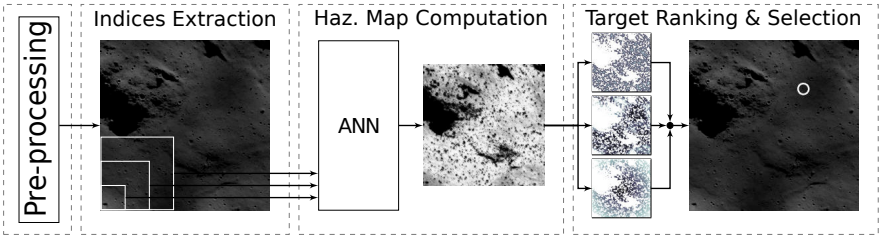


Fig. 6: Hazard detector working flow.

ference between the two solutions was less than $0.2\,\%$. A MC simulation is exploited also to assess the algorithm performances in terms of attainable landing area and fuel consumption, with a set of of 1×10^5 random diversions between $\pm 4000\,\mathrm{m}$ along both downrange and crossrange. The attainable landing area is obtained by correlating optimization results together with the coordinates of the TLSs, as shown by Figure 5b, in which only the targets found feasible are shown. The system is able to compute a feasible landing path in an approximately circular landing area of radius larger than $2300\,\mathrm{m}$ centred at the nominal landing site (at the origin of the figure), a performance better than what is required for similar scenarios [9, 10].

All the simulations were tested on a Intel® Core™ i7-2630QM CPU at $2\,\mathrm{GHz}$ of frequency. The mean computation time is $25.23\,\mathrm{ms}$ with a standard deviation (STD) of $7.16\,\mathrm{ms}$. The algorithm is very fast (and further improvements are possible in code optimization), compatible with on-board computation.

4.2 Hazard Detection

The hazard detector in development at PoliMi-DAER features Artificial Neural Networks (ANN). The capability of ANNs to autonomously correlate input and output without any prevoius knowledge of the actual rules that link the twos, also in conditions not explicitly considered during the projects phase, makes this kind of system very attractive for HDA tasks. Information extracted from a single-channel image provided by a monocular navigation camera are processed to generate a hazard map of the landing area, processed by the site selection routine to compute the new target. Figure 6 shows a logical scheme of the algorithm. Following, the hazard detection method is briefly summarized. A detailed dissertation is available in [11]. The retargeting process is divided in 4 different subphases:

Input and preprocessing. It has been considered as input a 8-bit gray-scale frame with a resolution of 1024×1024 pixels. The spacecraft is assumed to be in a near vertical attitude in this phase: small deviations from nadir pointing are corrected with perspective transformation.

Image processing and input assembly. Low level information is extracted from the image and passed as input vector to the ANN. Image is segmented and image processing algorithms are applied at 3 different scales to allow the ANN to grasp the scene and depth [12]. Each image segment corresponds to a pixel in the final hazard map, with an output resolution of 256×256 pixels.

Indexes extracted from the image include local mean μ and standard deviation σ, image gradient *Grad* and Laplacian of Gaussian *LoG* [13]. Finally, the Sun elevation angle is added to the input vector, to give to the network robustness to different illuminaiton conditions.

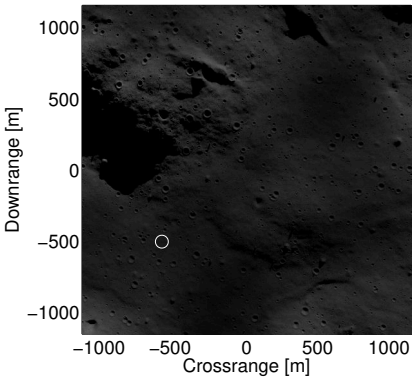
Hazard map computation. The assembled input matrix is processed by a cascade neural network. In the training of this kind of structure, the network is progressively increased adding one hidden layer at once, leading to a near-optimal configuration [14]. The output of the network represents a pixel of the hazard map, whose value spans from 0 (completely safe), to 1 (completely unsafe).

Target landing site selection. As the hazard map is available, a new target landing site is computed. Sites that do not respect minimum requirements of safety or dimension (including expected GNC errors) are immediately discarded, while the remaining candidate are ranked according to 3 criteria: minimum hazard index, maximum landing area, minimum distance from the nominal landing site (to maximize the probability to find a landing site actually reachable with the divert capabilities of the lander), as shown in Fig. 7. The influence of these parameters can be adjusted with three corresponding weights.

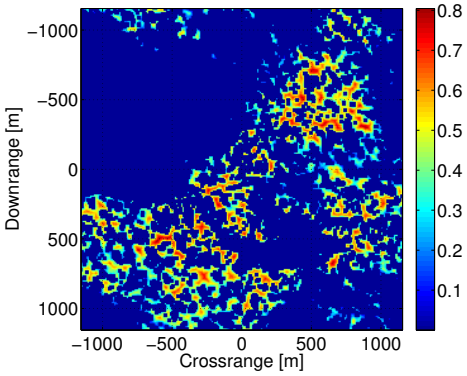
High resolution lunar DEMs (5-2 m/point) from LROC* were exploited to generate the required dataset of images to train and test the ANN. The models resolution has been increased up to 0.3 m/point adding small craters, boulders and fractal noise [15, 1, 16]. Finally, images were realistically rendered with ray-tracing soft-

*Courtesy of NASA and Arizona State University. URL: http://wms.lroc.asu.edu/lroc/rdr_product_select, last visit on: October 5, 2018.

ware[†], assuming a pinhole camera with 60° angle of view. Ground truth is computed directly from DEM data, by assigning a hazard index 1 for pixels in shadow, 0.33 to those pixels which fail the roughness or the slope criteria, 0.66 to those which fail both. The remaining pixels are classified as safe and set to 0. Finally, the ground truth solution is downsampled through Gaussian pyramids to 256×256 pixels.



(a) Original input image. The white circle marks the selected target.

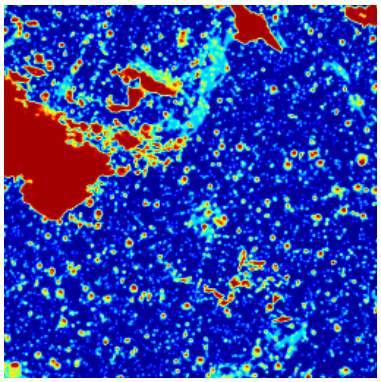


(b) Ranking of the landing sites found.

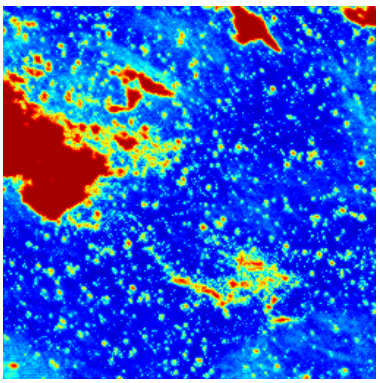
Fig. 7: The target landing site computation routine ranks all the suitable landing sites in the hazard map taking into account for each site the minimum hazard index, the maximum landing area and the minimum distance from the nominal landing site of the mission. The most scoring site represents the target.

Performances are assessed with a test set of images never exploited in the network training phase. The test set consists in 8 images of four lunar regions rendered at two different Sun inclination, 15° and 80°. A lander of 3 meters of diameter in footprint and a navigation error of 15 meters at 3σ has been considered. For the ground truth hazard calculations, terrain roughness over 0.5 meters and slopes over 15° are considered unsafe.

[†]Persistence of Vision Raytracer (Version 3.7). Retrieved from http://www.povray.org/download/



(a) Ground truth hazard map



(b) Computed hazard map

Fig. 8: Comparison between the ground truth and the hazard detector generated hazard maps.

A qualitative comparison between the ground truth and the hazard detector generated hazard map is shown in Fig 8: the terrain features are in general correctly interpreted by the network, that tends to slightly overestimate the hazard index of the lunar surface features. Landing sites classified as safe are compared to the ground truth. In particular, a landing site can be classified as *True Positive* (TP), *False Positive* (FP), *True Negative* (FN), and *False Negative* (TN), where a TN is a safe landing site correctly classified. Defining the *Safety Ratio* r_S as the fraction of true positives with respect to the total number of landing sites found, and the *Correctness Ratio* r_C as the fraction of correctly identified sites (TP) with respect to the true safe landing site in the image:

$$rs = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad r_C = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{7}$$

the probability to select an unsafe site is minimized maximizing r_S, whereas as r_C increases, the available landing area increases. The whole performance can be assessed with a unique index J expressed as:

$$J = r_S^5 R_C^{1/5} \tag{8}$$

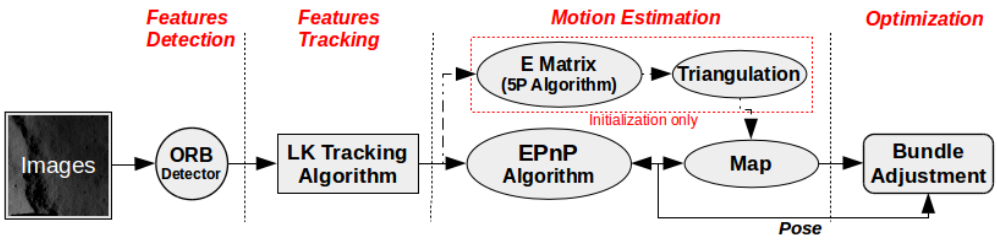


Fig. 9: Navigation System architecture. Feature detection, tracking, motion estimation and optimization, the main subsystems, are shown along with their components.

where the two exponents privilege landing sites safety during J maximization. A range of safety thresholds have been tested, and maximum J is recorded for a safety threshold of 0.17. In the ranking of target landing sites, a 0.6 weight has been selected for the landing site area, to increase robustness to navigation errors. A value of 0.3 is given to the distance with respect to the nominal landing site and the lowest of 0.1 is given to the mean hazard index of the site (since all the ranked sites are already classified as safe). With these parameters, the system:

- always selects a True Positive as Target Landing site;
- the worst case in ranking of the first False Positive is position 39, with an average on the test dataset of 695;
- an average Safety Ratio of 0.9649, meaning that over 96% of the landing sites found are actually safe.

The routine was coded in C++ and run on a AMD A10-7700K APU, with a 64 bit Ubuntu 14.04 GNU/Linux operative system. Both the methods resulted in a mean execution time well below $440\,\mathrm{ms}$. The indexes extraction stage confirmed as the task with the highest computational burden, spanning almost 50% or the total runtime. In the profiling test, the routine was forced to run on a single CPU core: however, recent improvements in space qualified hardware for massively parallel execution, such as Field Programmable Gate Arrays [17, 18], will make possible to dramtically speed-up the heaviest tasks.

4.3 Vision-Based navigation for landing

The Navigation Algorithm in development at PoliMi-DAER relies on a single grayscale camera working in the visible spectrum and is based on Visual Odometry and Simultaneous Localization and Mapping (SLAM) [19][20]. Salient features are extracted and tracked from the incoming images, while a local sparse 3D map of the landing area is built and used for navigation. Final relative position and attitude of the spacecraft are optimized with Bundle Adjustment (BA), an optimization technique widely diffused in computer vision. Here a summary of the structure of the system is presented, along with part of the results obtained from a test campaign made on synthetic images from the Lunar dataset. The current outline of the Navigation System is shown in Figure 9.

Features detection. Features are salient points on the processed image and different approaches exist to detect them [21][22][23]. Oriented FAST and Rotated BRIEF (ORB) [23] is exploited by the navigation algorithm for detection. The features extracted are extremely fast to compute, have good invariance to viewpoint and scale, and are resilient to different light conditions. To improve the features distribution over the frame, the image is segmented into 64 sectors, in which features are extracted independently. An upper bound of 300 features is set to be extracted to maintain a low computational cost.

Features tracking. ORB features extracted from the first frame are tracked on subsequent images exploiting the pyramidal Lucas-Kanade algorithm [24]. Known features are projected on subsequent frames and correspondence with new ones are searched in a bounded region. The pyramidal approach is exploited to make the algorithm more robust to track large motions. A stringent culling keypoint procedure is applied to reject wrongly tracked features or ones tracked with low accuracy, with the aim to constantly retain a modest number of features but tracked with high precision. If tracking fails or each time the number of tracked features drops below a fixed threshold, a new ORB detection is triggered and tracking restarted. Old features still tracked at the time of re-detection are merged with the new ones in order to constantly keep tracking the highest number possible.

Motion estimation. Motion estimation relies on correspondences between *3D* features of a map and *2D* tracked features, for this reason a map needs to be built and initialized for algorithm bootstrapping. Each time a new features detection is triggered, *2D* to *2D* correspon-

dences given by the tracking are exploited to implement the 5-Point algorithm [25] and to retrieve the essential matrix *E* connecting two frames. Motion is then obtained up to a scale factor applying Singular Value Decomposition (SVD) and cheirality check. Results obtained are improved by RANSAC iterations to reject presence of outliers. From the obtained motion 2D features are triangulated with the optimal method proposed in [26]. Reprojection error check is made at every step and a stringent culling policy is applied to discard wrongly triangulated points along with correspondent features. Built map is finally optimized along with the retrieved motion between the two frames applying bundle adjustment (BA). Whenever the tracking fails or is re-triggered, a new map is triangulated and merged with the existent one. Relative scale of the map and reconstructed trajectory is therefore obtained exploiting triangulation from subsequent image pairs as explained in [19].
Once the *3D* sparse map is initialized and the *2D* features are tracked, for each incoming image a set of *3D* to *2D* map to features correspondences is retrieved and used to solve the "Perspective-n-Point problem" (P*n*P). *EPnP* [27] algorithm is exploited at this step, which gives a non-iterative solution of the problem and is applicable both for planar and non planar map point configurations. The algorithm is implemented within a RANSAC routine as made for the essential matrix estimation in order to be robust to outliers and gives good estimation of the spacecraft pose.

Optimization. Motion estimation by itself is an error prone step and does not give a sufficient accuracy in trajectory reconstruction. Features extracted and tracked from the images are affected by noise, and as a consequence rotation matrix and translation vector from P*n*P solution are erroneous at a certain level. Moreover, these motion information are concatenated together to reconstruct the whole trajectory, also concatenating the errors, that grow exponentially as the trajectory develops. This results in an increasing drift in time which is classical of the VO problems. Bundle Adjustment is implemented on the navigation algorithm to counteract these effects. *g2o* library [28], specifically developed for pose graph optimization problems, is used as tool for the BA implementation. Levenrbeg Marquadt algorithm is exploited to solve the optimization problem, which is implemented in two ways:
Full BA. At the end of map initialization a full BA run is made. Optimizing both triangulated points location and estimated pose between the two frames used for initialization.
Motion only windowed BA. At the end of each motion estimation step, retrieved camera pose is optimized along with a defined number of poses retrieved from previous frames keeping fixed the observed map points.

4.3.1 Test on the experimental facility

Results obtained for two test performed on the navigation algorithm with the experimental facility are here expounded. Two trajectories with fixed attitude have been adopted as benchmark: Main Brake and Approach. Main Brake is a constant thrust horizontal brake maneuver over the Moon Planck crater, while Approach is a variable thrust approach phase maneuver representing the last part of the landing. These trajectories used for validation have two different scales: Main Brake trajectory is almost $26\,\mathrm{km}$ long and spans from $8\,\mathrm{km}$ to $2\,\mathrm{km}$ in altitude, while Approach trajectory is almost $2.8\,\mathrm{km}$ long and spans from $2.5\,\mathrm{km}$ to $0.5\,\mathrm{km}$ in altitude. Taking as reference the most critical scenario, corresponding to the Main Brake one, given the $1\,\mathrm{m}$ maximum stroke of the robotic arm, a $1:26000$ scale factor shall be used. Considering the current achieved sub-millimeter precision in the determination of the camera pose w.r.t. the diorama, this means that a trajectory scaled in that way would have a ground truth reconstruction with an uncertainty of $26\,\mathrm{m}$. Neverthless, the scaling factor constraint can be relaxed. The accuracy in MB trajectory reconstruction expected from the algorithm is in fact of about $150\,\mathrm{m}$, that is still one order of magnitude above the ground truth accuracy obtainable with the PA10 arm, thus classifying the $26\,\mathrm{m}$ uncertainty as an acceptable threshold. The same reasoning holds also for the approach trajectory, where a $1:2000$ scale factor is needed.

Approach with downward pointing camera. Whole reconstructed trajectory along with sparse 3D map for the approach trajectory with downward pointing camera is shown in Figure 10. Figure 11 show the camera

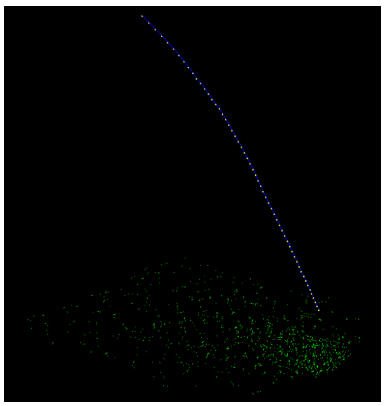
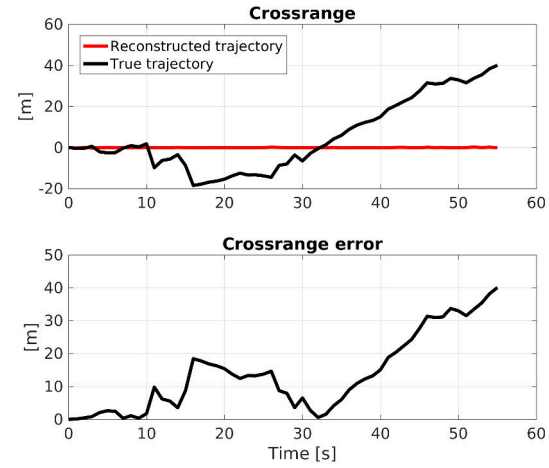
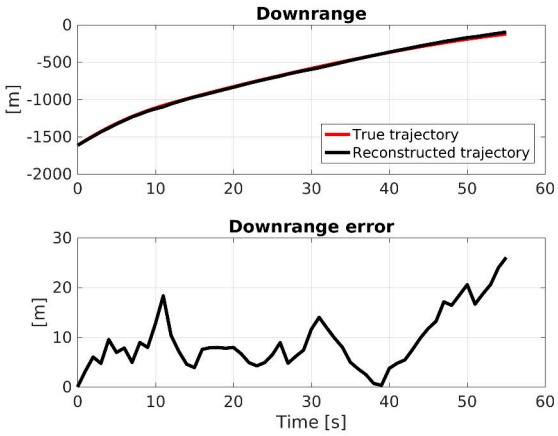


Fig. 10: Approach trajectory with downward pointing camera, reconstructed trajectory and 3D sparse map.

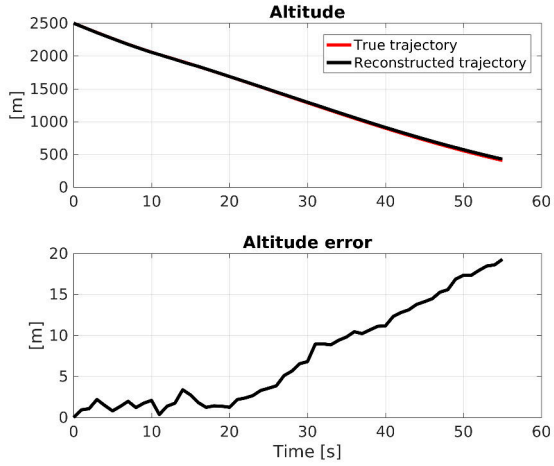
path behavior in time for each reference axis along with absolute error, while Figure 11d shows the attitude expressed in terms of quaternion. RMSE for each of the



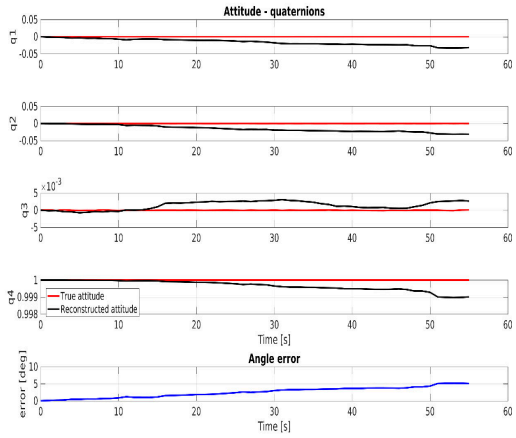
(a) Crossrange trajectory.



(b) Downrange trajectory.



(c) Altitude trajectory.



(d) Attitude quaternions.

Fig. 11: Reconstructed Approach trajectory in time compared against ground truth and error module.

aforementioned parameters is then given in Table 4. The algorithm is able to reconstruct the whole trajectory with almost optimal accuracy. Attitutde is retrieved with a low error, for what concerns position instead, maximum error is accumulated along crossrange direction but still within optimal values.

RMSE	
Crossrange	17.94 [m]
Downrange	10.89 [m]
Altitude	9.53 [m]
Angle error	2.94 [deg]

Table 4: Root Mean Square Error on each of the reference axis. Approach trajectory with downward pointing camera.

Main brake with downward pointing camera. Reconstructed Main Brake trajectory with downward pointing camera along with 3D sparse built map is shown in Figure 12. Figure 13 give the camera path and quaternion behavior in time along with absolute positioning error and angle error respectively. RMSE on each reference axis and for the angle error are shown in Table 5. Once again the navigation algorithm has been able to properly reconstruct the whole trajectory. With respect to the previous scenario, errors in position are higher, but this is mainly due to the largely increased scale of the scene observed. Results can therefore be considered satisfactory, with the attitude that is still reconstructed with optimal accuracy.

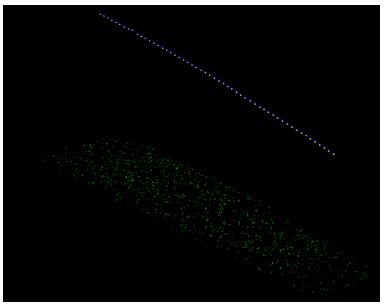


Fig. 12: Main Brake trajectory with downward pointing camera, reconstructed trajectory and 3D sparse map.

RMSE	
Crossrange	106.52 [m]
Downrange	156.69 [m]
Altitude	170.77 [m]
Angle error	1.52 [deg]

Table 5: Root Mean Square Error on each of the reference axis. Approach trajectory with downward pointing camera.

4.4 Vision-Based relative navigation with non-cooperative satellite

The Vision-Based algorithm for relative navigation with uncooperative space objects, consists in a Visual Odometry like routine [19][20] and works detecting the target spacecraft features from the incoming gray-scale images given by the mono-camera. Then, at each step, these features are matched to an available on-board map of the target (constituted by a mesh of *3D* pointswith their corresponding descriptor) and, in this way, a set of *3D* to *2D* correspondences is built. From the set of correspondences the so called Perspective-*n*-Point problem (P*n*P) is solved within a RANSAC routine set to delete incoming outliers. This process produces a first estimate of the relative satellite pose (position and orientation) Then, Bundle Adjustment (BA) [29], is applied to the map and *2D* features to optimize the obtained pose. This first estimate of the pose of the observed uncooperative object is fed to the navigation filter. A decoupled architecture is used for the navigation filter, separating the relative translational and rotational dynamics. An H-∞ filter is implemented for the translational part, since we can exploit a linear model that accounts for orbits eccentricity. For the rotational dynamics, the non-linear equations constrain to use a non-linear filter such as the Extended Kalman Filter. However, the use of an alternative formulation, based on Lie-Groups, is implemented. This formulation preserve the natural symmetry of the physical system without any ambiguity or singularity. The experimental validation of this algorithm is currently ongoing at the previously described facility. However, numerical simulations have shown promising results.

5. Conclusions

An analog experimental facility based on a robotic arm has been fully developed at PoliMI-DAER premises and will be further enhanced. The facility offers the possibility to simulate different space mission scenarios: planetary landing, relative in-orbit navigation with uncooperative objects and navigation around small bodies. Validation of vision-bsed AGNC algorithm up to TRL4 is currently possbile, with the aim to reach TRL6 level with future developments. A suite of tools and algorithms in development has been presented. An hazard detector based on a single camera and artificial neural networks and an efficient semi-analytical adaptive guidance algorithm as well as the navigation are in an advanced state. The research team scope is to complete an entire Adaptive Guidance, Navigation and Control chain with a single camera integrating the three subsystems aforementioned, to then carry out a full validation and testing activity with HIL exploiting the analog facility.

References

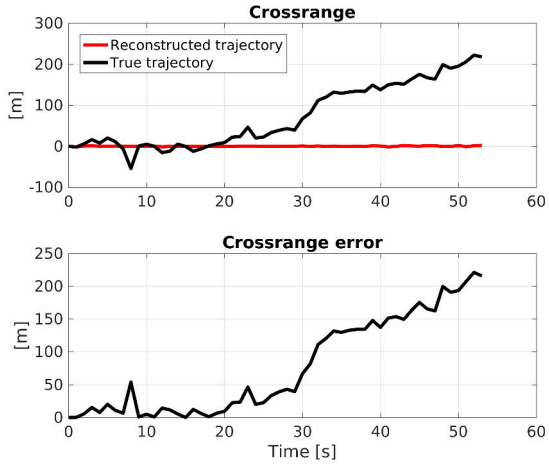
[1] P. Lunghi, M. Ciarambino, and M. Lavagna, "A multilayer perceptron hazard detector for vision-based autonomous planetary landing," AAS/AIAA Astrodynamics Specialist Conference 2015, (Vail, CO), aug 2015.

[2] P. Lunghi, M. Lavagna, and R. Armellin, "A Semi-Analytical Guidance Algorithm for Autonomous Landing," *Advances in Space Research*, vol. 55, no. 11, pp. 2719–2738, 2015.

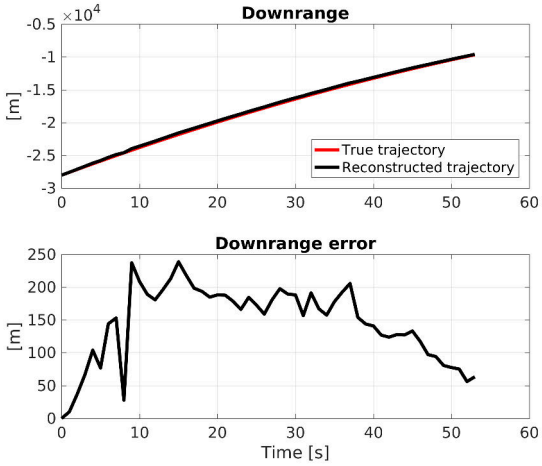
[3] P. Lunghi, R. Armellin, P. Di Lizia, and M. Lavagna, "Semi-analytical adaptive guidance computation based on differential algebra for autonomous planetary landing," 26th AAS/AIAA Space Flight Mechanics Meeting, (Napa, CA), Aug. 2016.

[4] B. Açikmeşe and S. R. Ploen, "Convex programming approach to powered descent guidance for Mars landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.

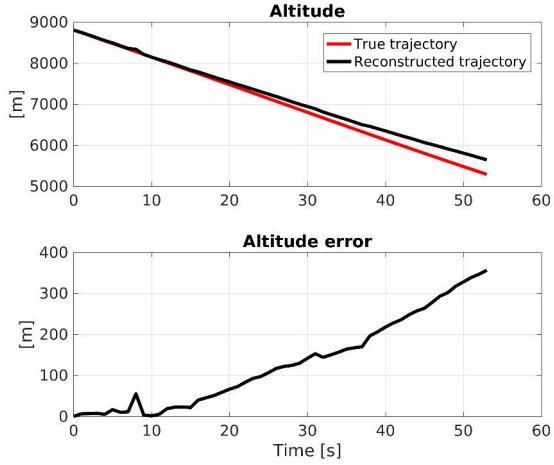
[5] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*. New York: Springer, 1988.



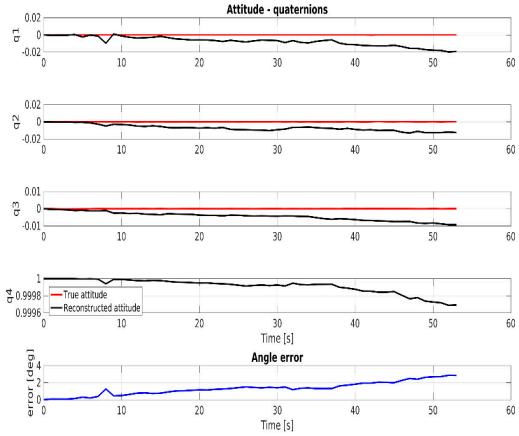
(a) Crossrange trajectory.



(b) Downrange trajectory.



(c) Altitude trajectory.



(d) Attitude quaternions.

Fig. 13: Reconstructed Main Brake trajectory in time compared against ground truth and error module.

[6] G. Flandin, B. Polle, N. Despré, J. Lheritier, N. Perrimon, and P. Blanc-Paques, "Maturing vision based navigation solutions to space exploration," AIAAGNC, (Toronto, Ontario Canada), Aug. 2010. AIAA Paper 2010-7601.

[7] J. Riedel, A. Vaughan, R. A. Werner, W. Tseng-Chan, S. Nolet, D. Myers, N. Mastrodemos, A. Lee, C. Grasso, T. Ely, and D. Bayard, "Optical navigation plan and strategy for the lunar lander Altair; OpNav for lunar and other crewed and robotic exploration applications," AIAAGNC, (Toronto, Ontario Canada), Aug. 2010.

[8] M. Berz, *Differential Algebraic Techniques. Entry in Handbook of Accelerator Physics and Engineering*. New York: World Scientific, 1999.

[9] J. Delaune, D. De Rosa, and S. Hobbs, "Guidance and control system design for lunar descent and landing," AIAAGNC, (Toronto, Ontario Canada), 2010.

[10] M. C. Johnson, "A parameterized approach to the design of lunar lander attitude controllers," AIAAGNC, (Keystone, CO), 2006.

[11] P. Lunghi, M. Ciarambino, and M. Lavagna, "A multilayer perceptron hazard detector for vision-based autonomous planetary landing," *Advances in Space Research*, vol. 58, no. 1, pp. 131–144, 2016.

[12] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in Neural Information Processing Systems*, pp. 1161–1168, 2005.

[13] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*. Newnes, 2002.

[14] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems 2* (D. Touretzky, ed.), pp. 524–532, Morgan-Kaufmann, 1990.

[15] U. Shankar, W.-J. S., T. Criss, and D. Adams, "Lunar terrain surface modeling for the alhat program," in *IEEE Aerospace Conference*, pp. 1–10, March 2008.

[16] F. Hörz, R. Grieve, G. Heiken, P. Spudis, and A. Binder, *Lunar Surface Processes*. Cambridge University Press, 1991.

[17] G. Capuano, M. Severi, E. D. Sala, R. Ascolese, C. Facchinetti, and F. Longo, "Compact and high-performance equipment for vision-based navigation," in *63rd International Astronautical Congress (IAC)*, 2012.

[18] M. Dunstan and K. Hornbostel, "Image processing chip for relative navigation for lunar landing," in *9th International ESA Conference on Guidance, Navigation, and Control Systems (GNC 2014)*, 2014.

[19] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.

[20] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 2, pp. 78–90, 2012.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.

[24] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.

[25] D. Nistér, "An efficient solution to the five-point relative pose problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 756–770, 2004.

[26] R. I. Hartley and P. Sturm, "Triangulation," *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[27] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o(n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.

[28] G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*, 2011.

[29] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Vision algorithms: theory and practice*, pp. 298–372, Springer, 1999.