14th Symposium on Advanced Space Technologies in Robotics and Automation

20 - 22 June 2017, Sheltema, Leiden, the Netherlands

ADVANCEMENTS OF VISION BASED AUTONOMOUS PLANETARY LANDING SYSTEMS AND PRELIMINARY TESTING BY MEANS OF A NEW DEDICATED EXPERIMENTAL FACILITY

Marco Ciarambino, Paolo Lunghi, Luca Losi, Michèle Lavagna

Politecnico di Milano, Dept. of Aerospace Science and Technologies, Italy

Outline

- The Autonomous Landing Problem
- Autonomous Guidance, Navigation & Control chain
- Autonomous Guidance
- Hazard Detection & Target selection
- Vision Based Navigation
- Experimental Landing Facility
- Conclusions and future developments

The Autonomous Landing Problem

To guide a lander spacecraft to a given target on the planet's surface with an accuracy of fewer than several hundred meters.

- In last decades, several improvements were achieved in automatic landing precision.
- Still **large uncertainties** at touchdown.
- No control on horizontal position:
  - Velocity and altitude are controlled to ensure a soft landing.
  - **Horizontal position** at touchdown largely depends on **uncertainties at entry/deorbiting point**.
- **Long** and **complex target selection** process.
- **The absolute landing site dispersion ellipse must be fit in a safe area**.

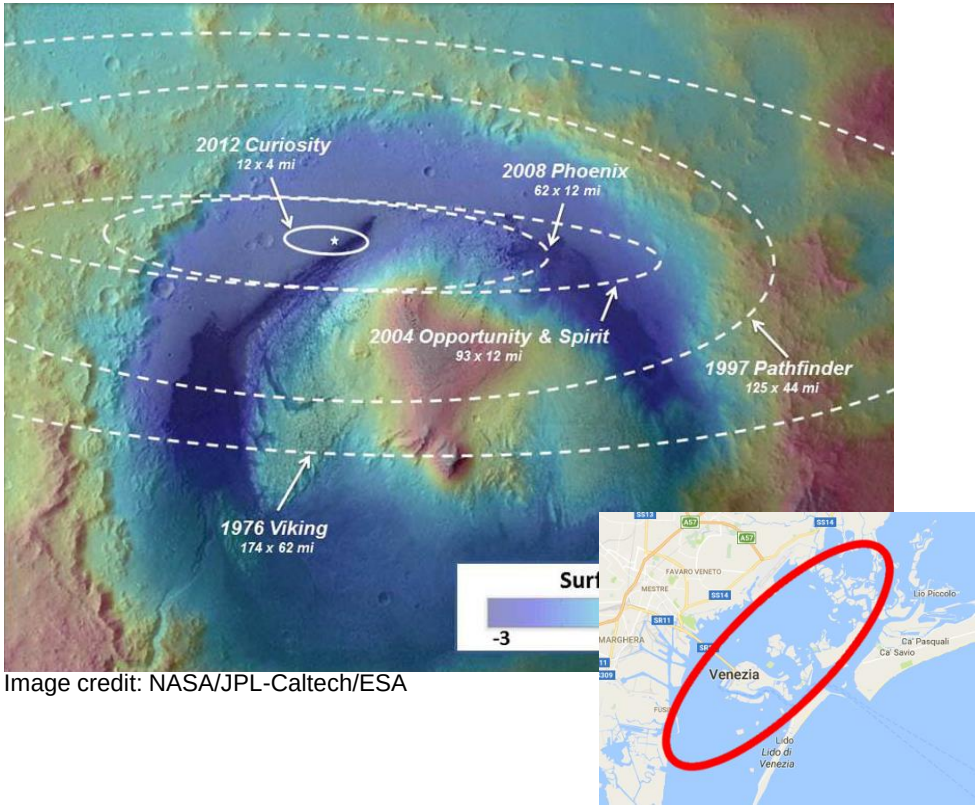
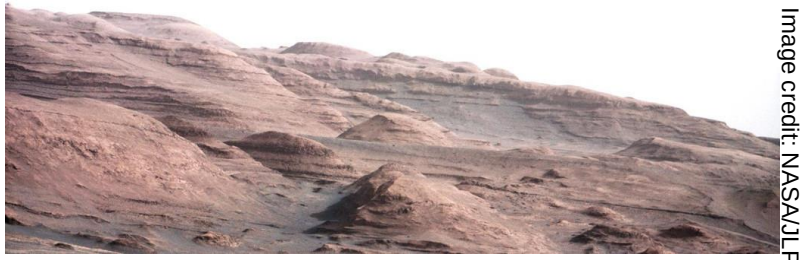


Image credit: NASA/JPL-Caltech/ESA

New Requirements for Future Landing Systems

- Most **scientifically relevant places** are often located near unsafe areas.
- Planet or NEA surface may be **not known in detail** before landing.
- **Short duration** of landing phase, w.r.t. large **telecommunications delay**, because of interplanetary distances.
- Impossibility to counteract **unexpected events**.



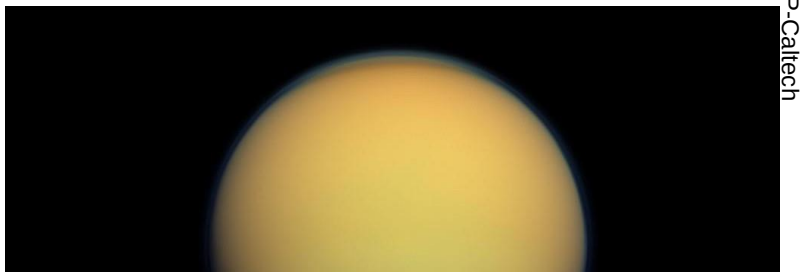
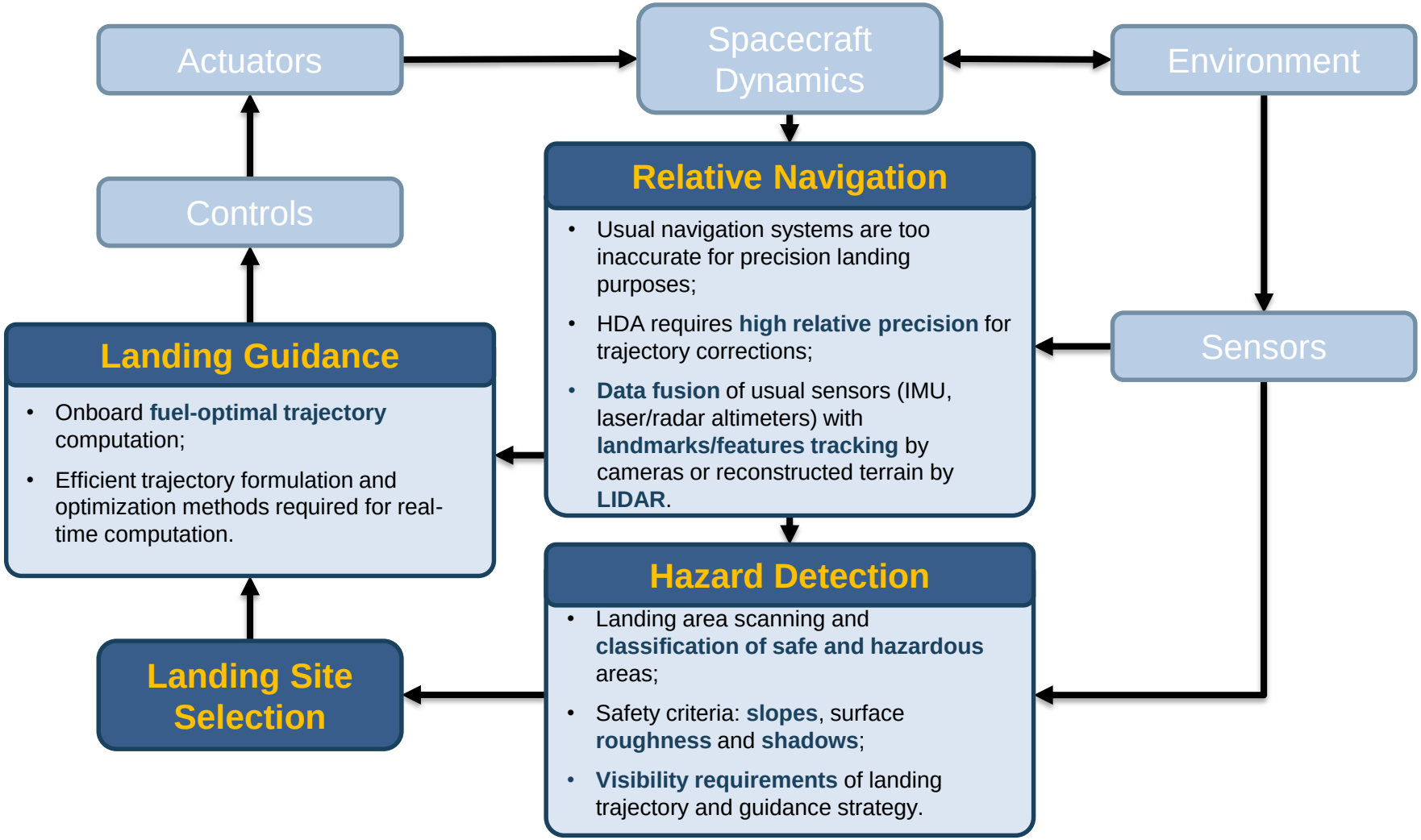


Image credit: NASA/JLP-Caltech

A high level of **precision** and **autonomy**, in landing or in the approach to uncooperative objects, will be required by the **next space systems generation**.

I	II	III
Scan the landing area to detect hazardous terrain items.	Select a **new target**, depending on safety and mission requirements.	Compute a feasible trajectory and perform a **divert maneuver**.

The Autonomous Landing GNC Chain



Formulation

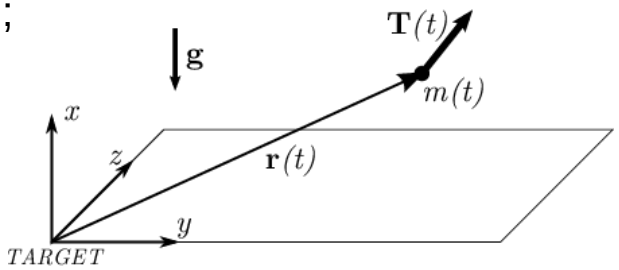
- Planet radius ≫ Altitude: **Flat ground**; **Constant gravity** field;
- **3DoF+Mass** dynamics, **throttleable** thrust:

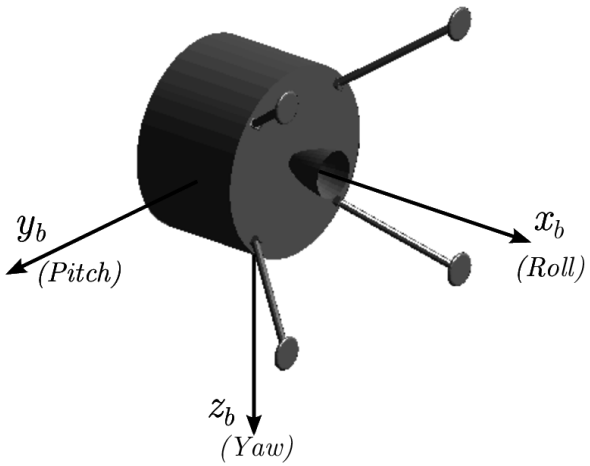
$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \dfrac{\mathbf{T}}{m} + \mathbf{g} \\ \dot{m} = -\dfrac{T}{I_{sp} g_0} \end{cases}$$

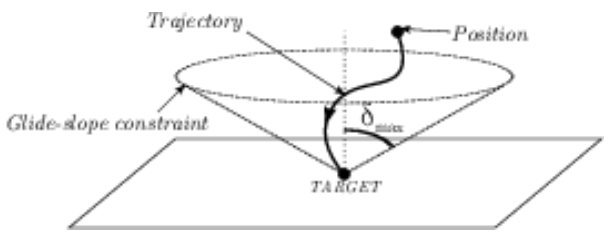
- Thrust vector tightened to spacecraft body: control acceleration depends only on **attitude** and **thrust magnitude**

$$\ddot{\mathbf{r}}(t) = a\big(T(t), \mathbf{e}(t)\big)$$

- Acceleration profile as **polynomials**: $\ddot{\mathbf{r}} = P_n(t)$, <u>$n =$ minimum order required to satisfy boundary constraints</u>.
- **Inverse dynamics** allow to obtain a control profile function of time-of-flight t_{tof} and initial thrust magnitude T_0.
- **Optimization problem**: find the free parameters $\mathbf{x} = [t_{tof}, T_0]$ that minimizes the cost function $f(\mathbf{x}) = m(t_{tof}) - m(0)$, subject to **box constraints** $\mathbf{x}_{L/U}$ and **path constraints** $\mathbf{c}(\mathbf{x})_{L/U}$







Differential Algebra Based Optimization

- Origin: attempt to solve analytical problem trough an **algebraic approach**;
- Any quantity is represented not by its value at a specified point, but with its **Taylor expansion** about that point up to an **arbitrary order**;
  - **Single variables**: $x = x_0 \rightarrow [x] = x_0 + \delta x$
  - **Functions**: $f(x, y) \rightarrow [f] = \mathcal{P}_f(\delta x, \delta y)$
- A DA object carries **more information** rather than its mere value;

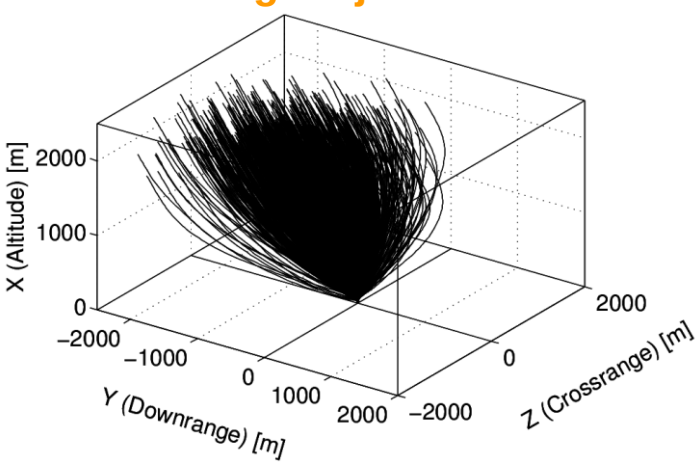
Algebra of Real Numbers → Algebra of Taylor Polynomials

- All the **standard mathematical operators** are defined **between DA objects** as well as between floating point numbers;
- Plus: **derivation**, **integration** and **map inversion** are simple operations between Taylor coefficients in the DA domain.
- Cost function is expanded as DA variable: it includes **sensitivity to optimization variables** without additional computational burden, exploited to efficiently find the optimum.

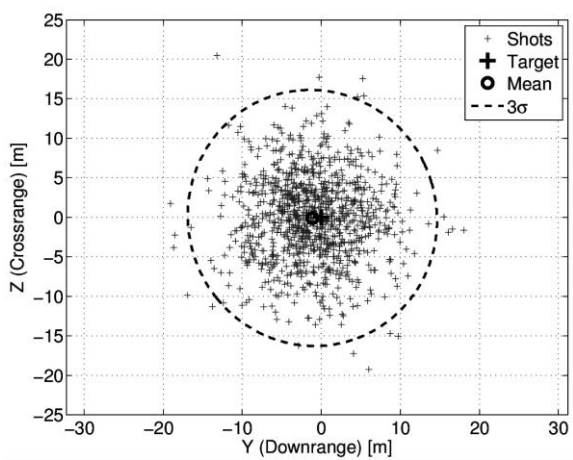
Performances

- **Lunar** soft landing **approach phase** (2000÷30 m altitude with HA maneuver);
- **Montecarlo** simulation:
  - 6 DoF dynamics;
  - Actuators models
  - Sensors errors models;
  - Trajectory update every 5 seconds;
- Large dispersion on **initial position** representative of a **large scale hazard avoidance**:
  - r_0 = [2000,−1062,0] ± [130,600,600] m (1σ)
- **Initial conditions** uncertainties:
  - Velocity, Attitude, Fuel Mass
- **Model** Uncertainties:
  - Specific Impulse, Available Thrust, Inertial Properties, Gravity
- Small t_{tof} (~ 10^2 s for a 2000 m initial altitude) maintains **low dispersion**;
- **Precision** better than **16 m** (3σ) with a **diversion capability** better than **2300 m**.

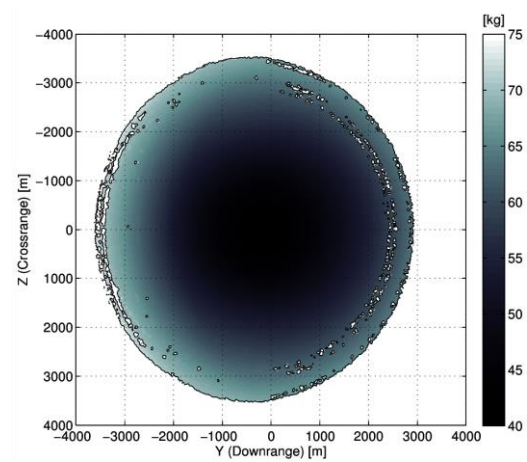
Landing Trajectories



Landing Precision



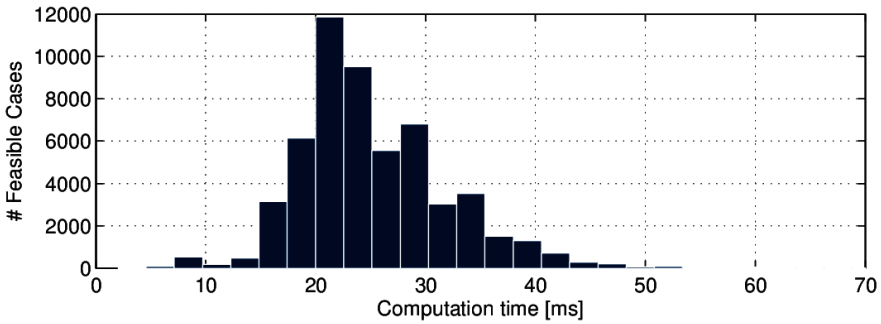
Attainable Area

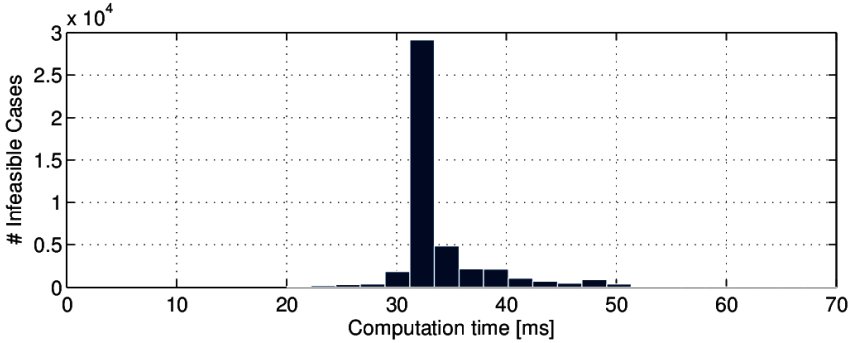


Computation Time

- Estimation of computation time obtained by attainable area simulation;
- Simulation run on a PC on a Intel® Core™ i7-2630QM CPU (2 GHz)

- **Feasible cases** – variable iteration number (~55k cases)
  - **Mean time**: 25.2 ms (3σ < 46.7 ms)
  - **Mean total iteration #**: 12 (17÷45)
- **Infeasible cases** – stable iteration number (~45k cases)
  - **Mean time**: 33.9 ms (3σ < 46.5 ms)
  - Max iteration limit: 30
  - **Mean total iteration #**: 29.5 (17÷30)

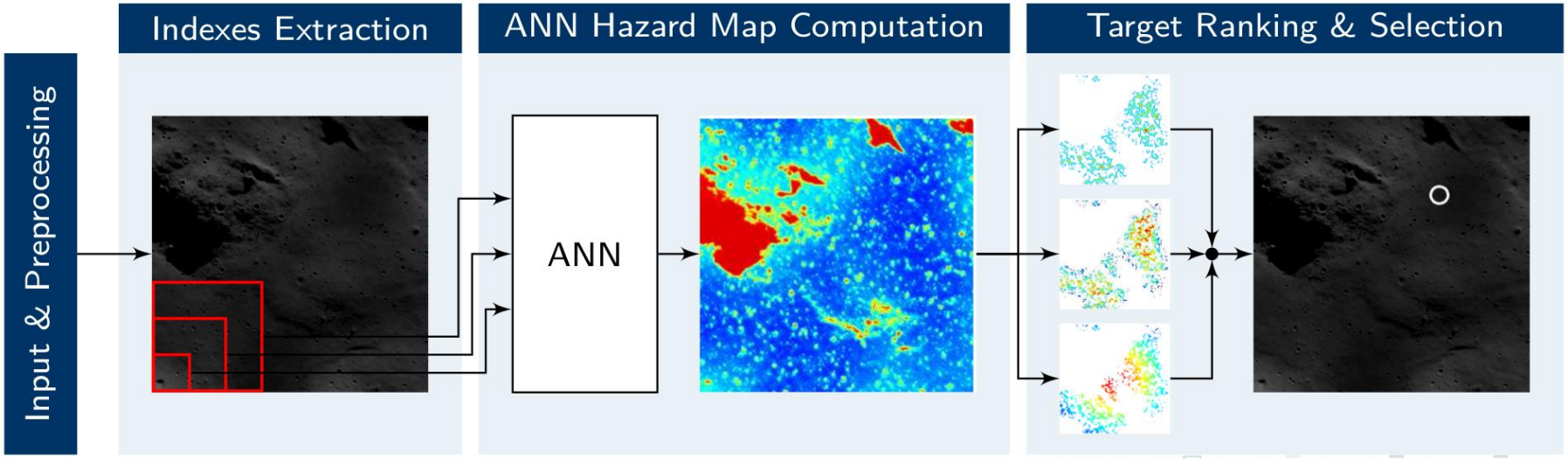




Computation time compatible with on-board computation.

System Architecture

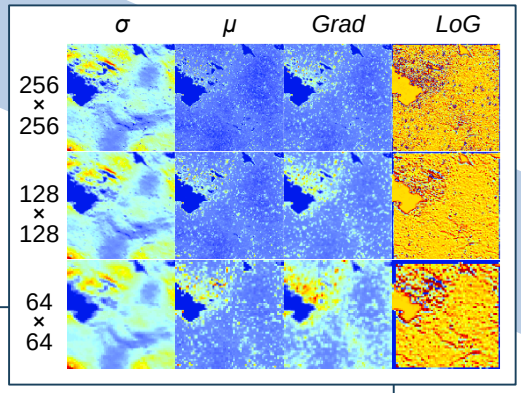
- **Safety criteria**:
  - **Sensors visibility** (shadows). Unobservable areas must be classified a priori as unsafe.
  - **Max surface roughness**. Max local obstacles dimension handleable by the system.
  - **Max slope**. Maximum slope angle to avoid capsizing.
  - **Min area**. Determined by lander footprint and expected GNC error at touchdown.
- **ARTIFICIAL NEURAL NETWORKS** are capable to operate also in **conditions not explicitly considered during the project phase**;
- Safety is expressed by a **hazard index:** $z \in [0 = \text{absolutely safe}, 1 = \text{absolutely hazardous}]$;
- Each image position is associated with a value of z, shown as a **hazard map**.



Hazard Map Computation

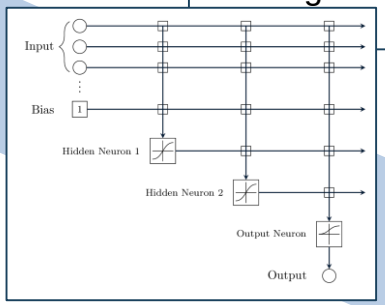
Input & Preprocessing

- Single image, 8 bit grayscale, 1024×1024 px
- Perspective image correction for inclined views
- 3×3 Median filtering (remove image noise)



Cascade Neural Network

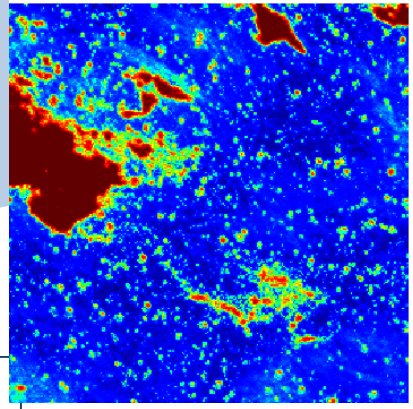
- Each hidden layer is made up by one single neuron;
- The number of hidden neurons is part of the training;
- This allows to obtain a **nearly optimal architecture**.
- Trained with a set of 100 artificial images.



Indexes Extraction

Elementary information is extracted from the original frame at **3 different scales**:

- **Mean** (*μ*, 0-th order information)
- **Standard Deviation** (σ)
- **Local image gradient** (1-st order information)
- Local **Laplacian of Gaussian** (2-nd order information)
- The **Sun Elevation Angle.**



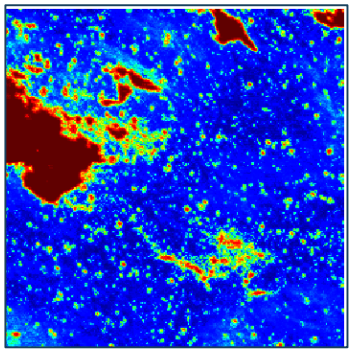
Hazard Map

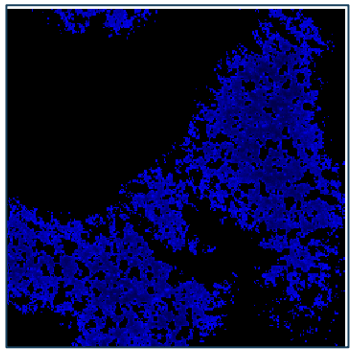
Target Ranking and Selection

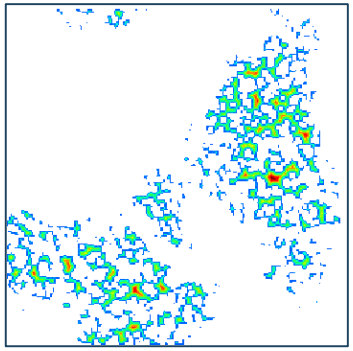
Landing Site (LS) criteria:

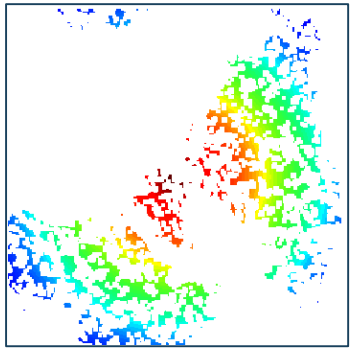
- **Hazard index** (lower is better);
- **LS area** (higher is better);
- **Distance from nominal LS** (lower is better).

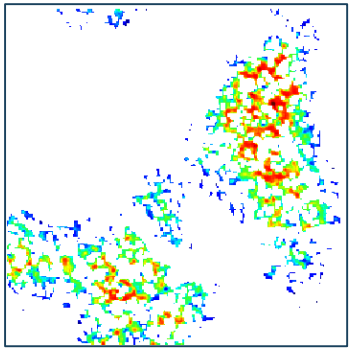
Step 1: thresholding at maximum allowed hazard index z_{max}.









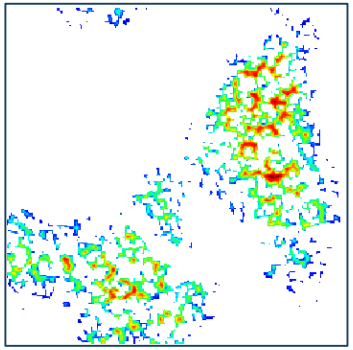


Step 2: **Ranking scores** assignment:

1. Nearest unsafe pixel distance (r_{CLSij});
2. Nominal Landing Site distance (d_{CLSij});
3. Mean hazard index (z_{CLSij}).

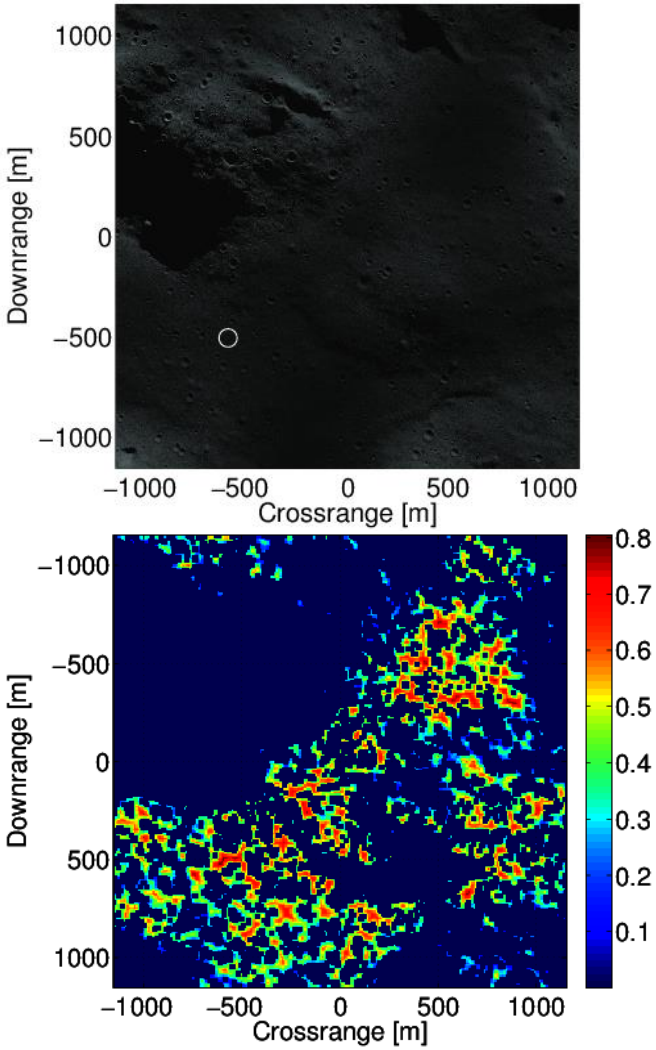
Step 3: The **Global Landing Index** I_{CLSij} computation.

- $I_{CLSij} = [r_{CLSij}\ d_{CLSij}\ z_{CLSij}] \cdot \boldsymbol{w}$
- $max(I_{CLSij})$ is assumed as new **Target Landing Site**;

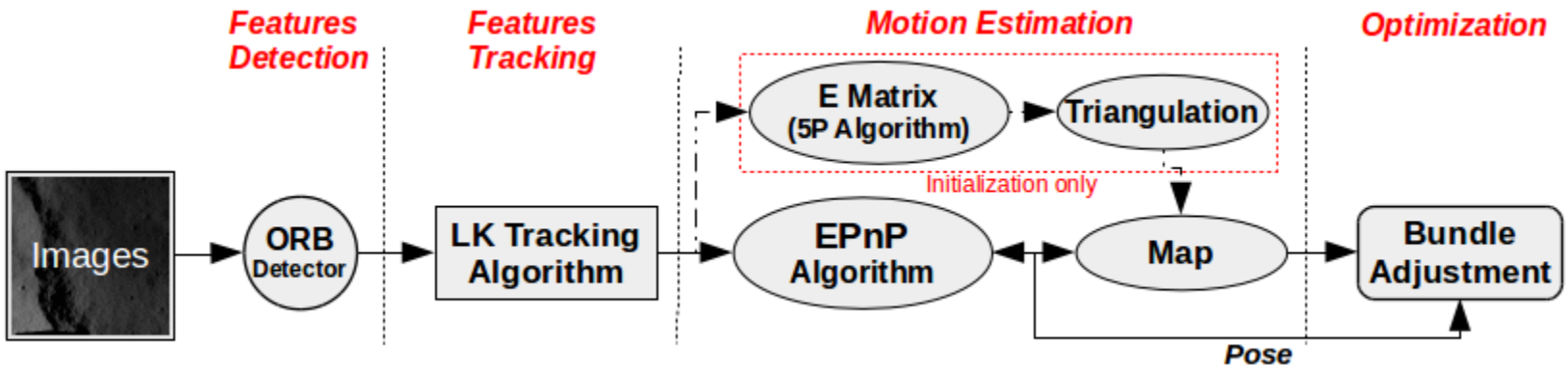


Performance

- Test set **never considered for network training**
- 8 test cases (4 landing areas at 2 sun angles)
- **Always a True Positive is selected as target**;
- Mean first **False Positive** ranking: **695** (worst case: 39)
- **Backup solution** is always possible.
- Runtime < 450 ms (on PC with code optimization margins)
- **Real-time compatible** computational performance.



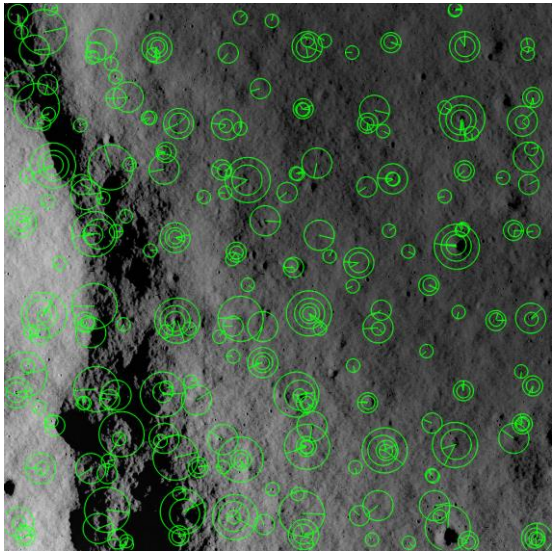
System Architecture



Vision-Based navigation reconstructs relative position and orientation (pose) of the spacecraft in real-time exploiting images from a monocamera:

- **Features detection**, salient features are detected from incoming image.
- **Tracking**, detected features are tracked across subsequent images.
- **Essential matrix** is retrieved from first two images and features triangulated to initialize a **sparse 3D map.**
- Tracked features are related to the map and the set of **2D to 3D** correspondences obtained used to solve the **P*n*P problem.**
- **Bundle Adjustment** technique is exploited for both map and relative spacecraft pose optimization.

Features Detection & Tracking



ORB detector is used for features extraction from incoming images:

- Fast to compute.
- Good scale and viewpoint invariance.
- Robust to change in light condition.
- Image is segmented in *64* sectors in which features are extracted independently.
- Around *300* features extracted to keep low computational burden.

Pyramidal Lucas-Kanade algorithm is exploited to track extracted ORB features on subsequent images:

- Known features are projected on successive image and searched in a bounded region.
- Pyramidal approach makes algorith robust to large motions.
- Stringent keypoint culling policy is applied to reject wrongly tracked features.

➢ Each time tracked features number drops below *70* a new detection is triggered and tracking restarted.

Motion Estimation & Optimization

First two frames are exploited to initialize a new **sparse map**:

- ***E matrix*** is retrieved with 5-Points algorithm along with motion.
- Tracked features are **triangulated** and a 3D sparse map is initialized.
- A stringent culling policy is applied to reject wrongly tracked features.

After initialization at each step 3D to 2D features correspondences are exploited to retrieve relative spacecraft pose with **EPnP algorithm**:

- Fast non-iterative solution to the P*n*P problem.
- Works both for planar and non-planar 3D point clouds configurations.
- Implemented in **RANSAC** routine to reject outliers.

Bundle Adjustment implemented as hyper-graph:

- Full BA run is made after map intialization.
- Motion Only BA is made at the end of each motion estimation step.

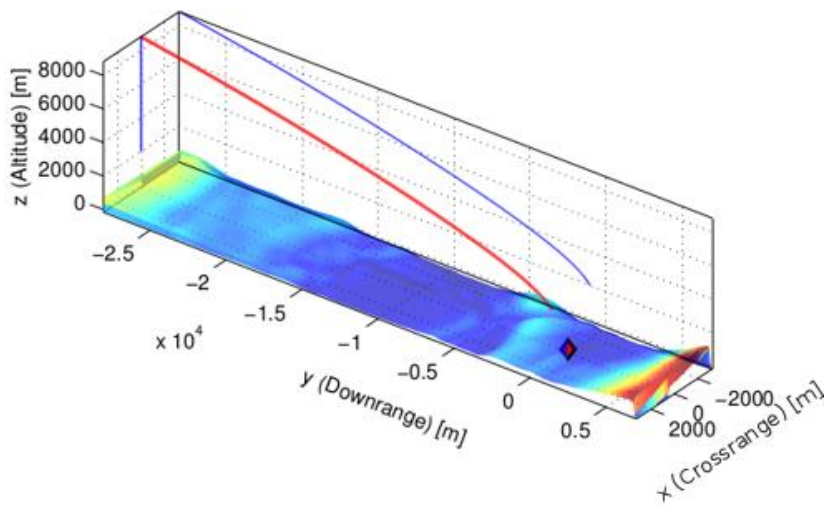
➢ Each time tracking is restarted a new map is triangulated and merged with the existent one

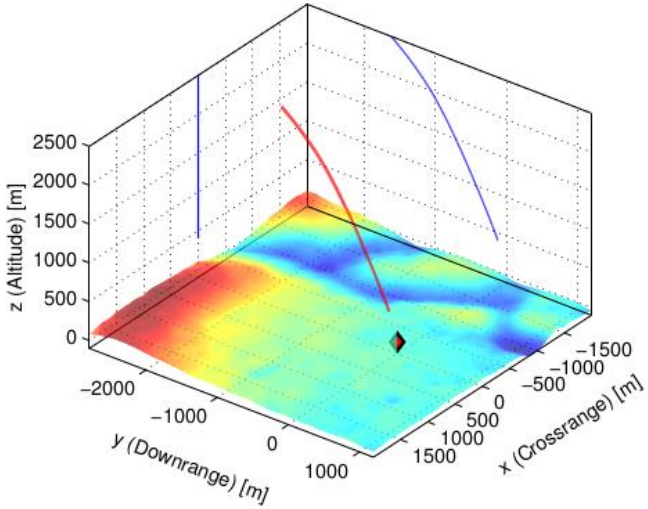
Simulations

Navigation system has been tested on **synthetic image** sequences**.**

- **Lunar dataset:** Set of two **Lunar landing trajectories** with different spacecraft attittude sequences, developed @ PoliMI. Images are obtained from Lunar DEMs from LROC dataset; with fractal noise and details added to increase resolution.

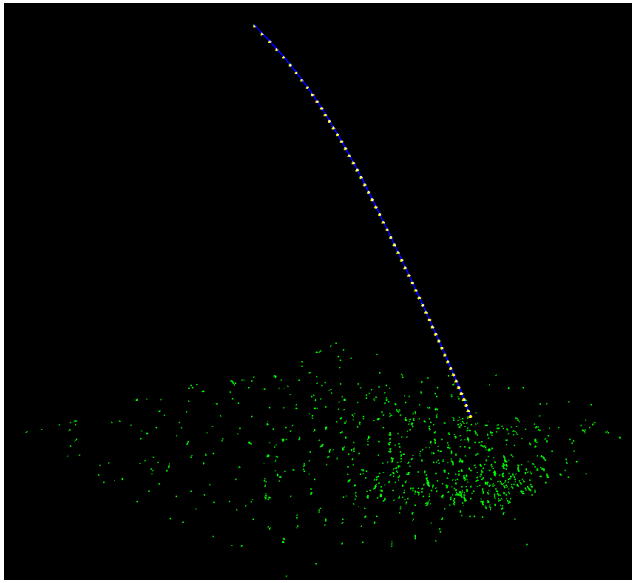
Sequence	Characteristics
Main Brake	***Large scale*** *scenario, translational motion, observed scene appears as flat.*
Approach	***Smaller scale*** *scenario, motion almost vertical with zooming effect on the images.*

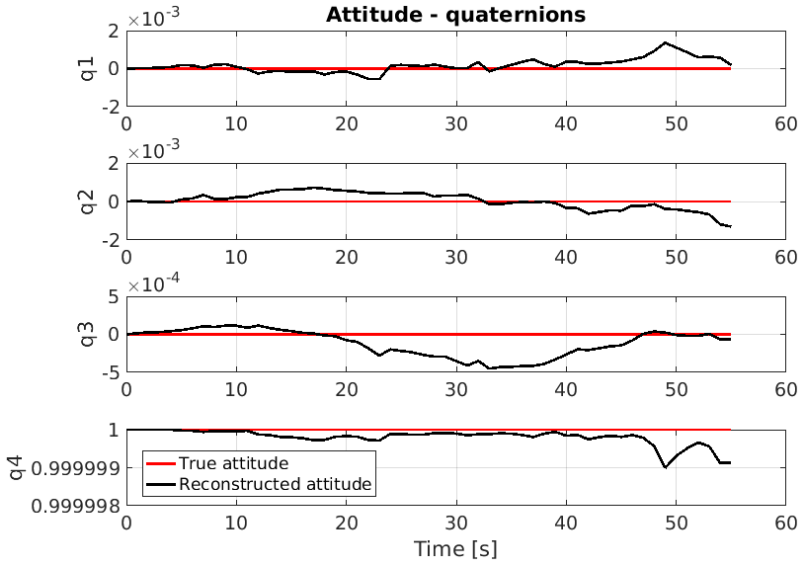


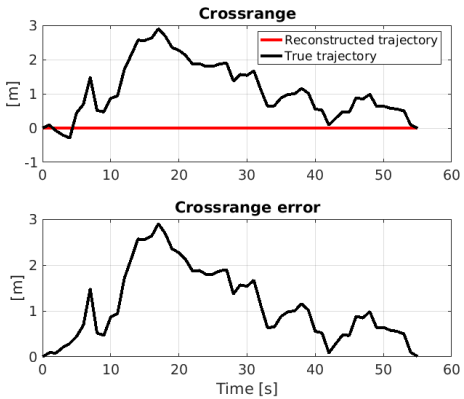


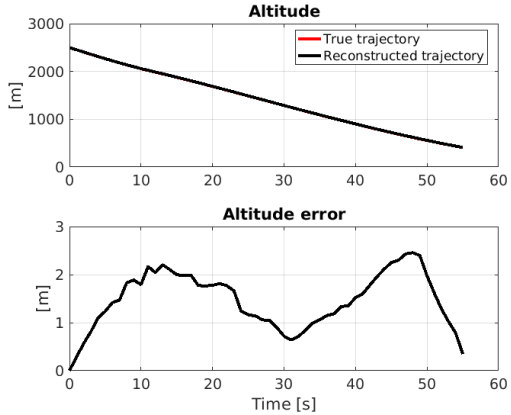
Vision Based Navigation
Performance

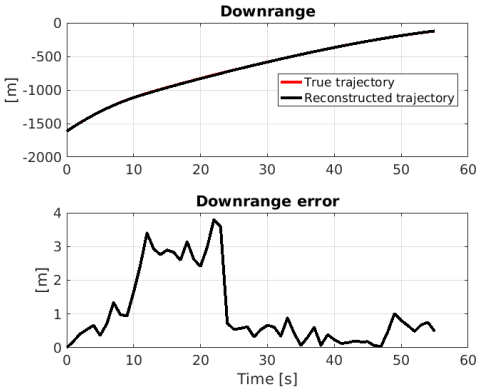
- Reconstructed Lunar **Approach** trajectory for the **downward pointing** sequence:







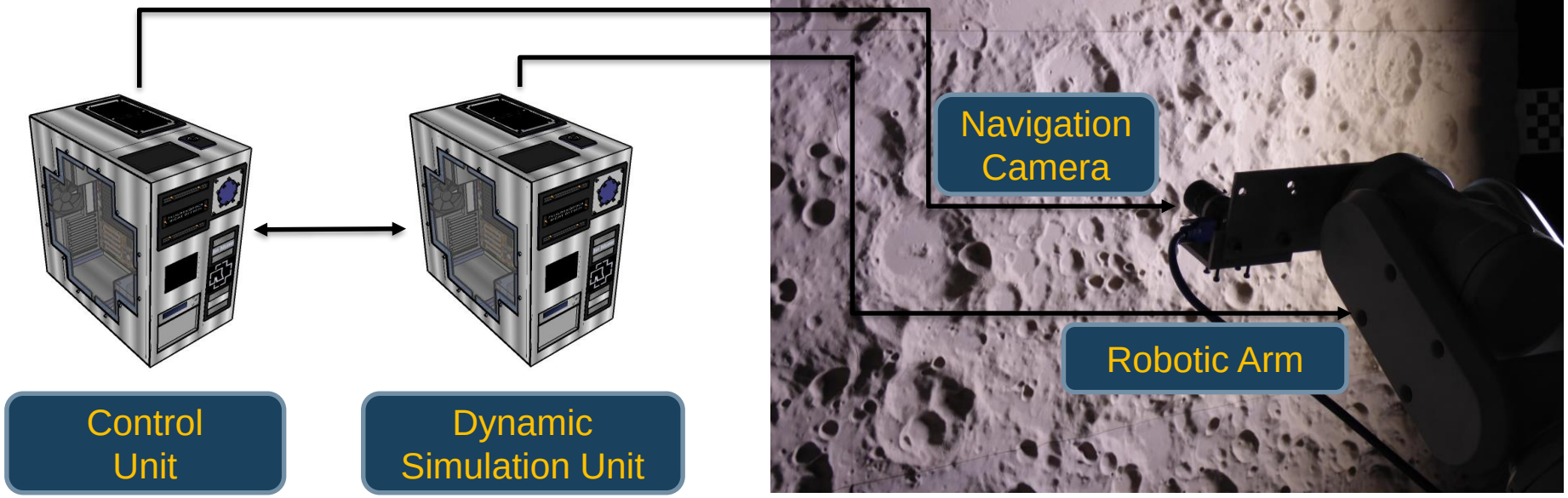




Overview

An hardware-in-the-loop experimental facility is under setup at PoliMi-DAER premises to increase the TRL of the presented algorithms up to 5. The facility features:

- A **robotic arm** carrying the suite of sensors to simulate lander dynamics.
- A 3D **planetary mock up**.
- An **illumination** system.
- A Control and test computer.

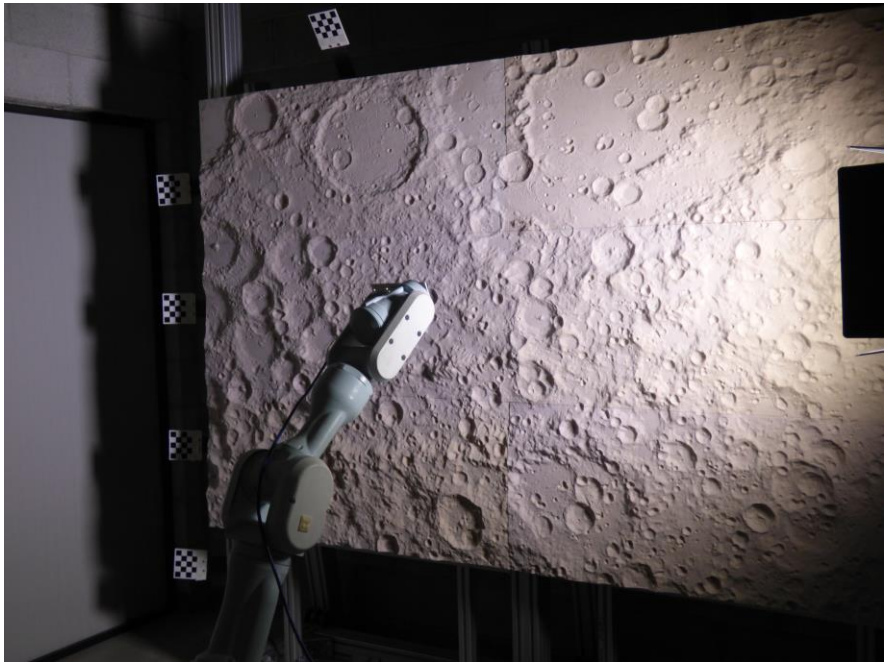


Components

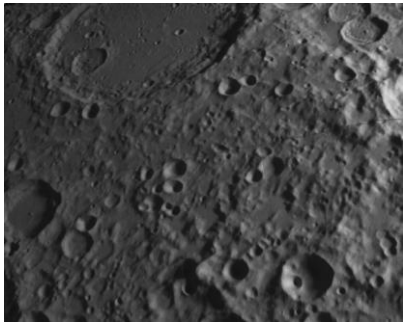
Robotic Arm

Mitsubishi PA10-7c

Technical specifications
Dedicated controller
7 Degree of Freedom
Real time ready



Navigation Camera

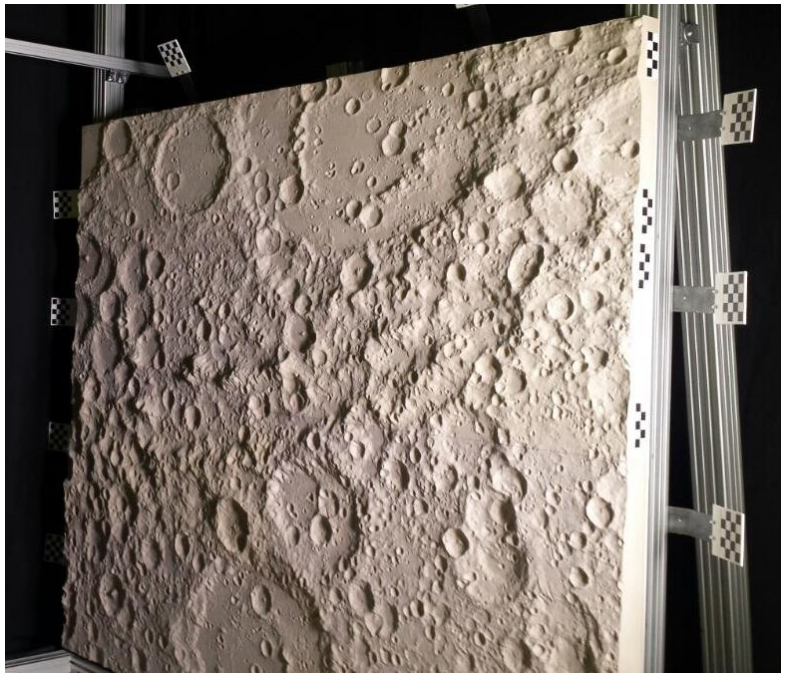


Technical specifications	
Resolution	*1280X1024*
Framerate	*149*
Megapixels	*1.3*
Chroma	*Color/Grayscale*
ADC	*10 bit*
Sensor format	½
Focal length	6 mm
Field of view	43.5°

Components

Lunar Terrain Mockup

- 2400 x 2000 mm **Lunar terrain diorama**
- Based on Large Scale Real Low-Res DEM (2m/px) from NASA LROC NAC data;
- Refined up to 0.25 m/px by addiction of: statistical random Craters Deposition, Boulders Deposition, Fractal noise.
- Carved in **Urethane Foam** with numerical control machine;
- **Optically calibrated** with Structure from Motion Technique



Lightning System

Technical specifications	
Light temperature	*5600 K*
Array dimensions	*1024X1024*
Beam angle	*60°*

- Illumination conditions as close as possible to real lunar surface.
- Exclusion of external light sources to provide full control.

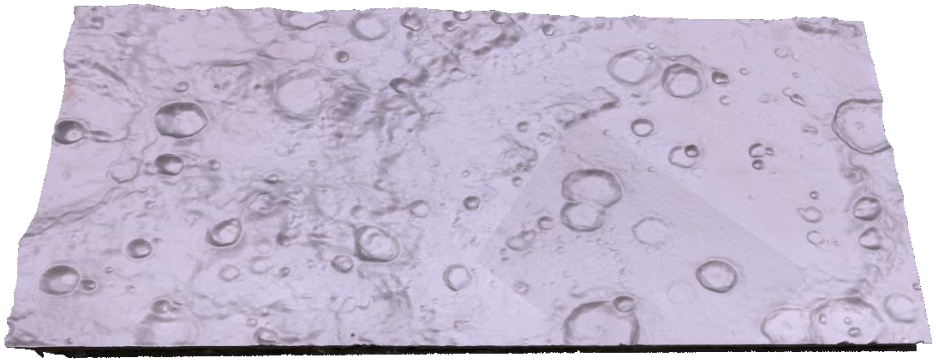
Calibration

Actual milled diorama is different from the numerical model due to imperfections during the production process.

- **True camera trajectory** relative to terrain must be reconstructed with one order of magnitude better accuracy than the navigation algorithms expected to be tested.
- Being the **desired accuracy** *10 m*, target accuracy on diorama calibration shall be better than ***0.5 mm.***

Dense matching method have been selected for shape reconstruction:

- Several photos from different angles for each tile.
- Structure from motion and dense reconstruction methods.
- Tiles models assembled with Iterative Closest Point algorithms.



➢ *Example of calibrated tile*

Conclusions & Future developments

A suite of Vision-Based tools and algorithms for autonomous landing on planets and small bodies is under development at PoliMi – DAER:

• An **Hazard Detector** based on artificial neural networks and a semi-analytical **Adaptive Guidance** algorithm are completed and ready.

• A **Vision Based Navigation** algorithm derived from Visual Odometry and Simultaneous Localization and Mapping techniques is functional and under further development.

• An **Experimental facility** dedicated to validation and testing of optical navigation algorithms is under setup at PoliMi premises and will be fully operative by the end of summer 2017.

Future developments will focus on data fusion with filtering techniques and hardware implementation of the algorithms for further validation on the landing facility.

14th Symposium on Advanced Space Technologies in Robotics and Automation

20 - 22 June 2017, Sheltema, Leiden, the Netherlands

ADVANCEMENTS OF VISION BASED AUTONOMOUS PLANETARY LANDING SYSTEMS AND PRELIMINARY TESTING BY MEANS OF A NEW DEDICATED EXPERIMENTAL FACILITY

Marco Ciarambino, Paolo Lunghi, Luca Losi, Michèle Lavagna

Politecnico di Milano, Dept. of Aerospace Science and Technologies, Italy