Experimental Results of a Terrain Relative Navigation Algorithm using a Simulated Lunar Scenario

Luigi Ansalone[a,1,*], Eleonora Grava[b,2], Fabio Curti[b,3]

[a] *ASI, Italian Space Agency, Technology Division, Via del Politecnico snc, Rome 00133, Italy*

[b] *ARCA Lab, Department of Astronautical Electrical and Energy Engineering, Sapienza University of Rome, Via Salaria 851, Rome 00138, Italy*

Abstract

The paper delals with the problem of the navigation of a lunar lander based on the Terrain Relative Navigation approach. An algorithm is developed and tested on a scaled simulated lunar scenario over which a tri-axial moving frame has been built to reproduce the landing trajectories. At the tip of the tri-axial moving frame a long-range and a short-range infrared distance sensors are mounted in order to measure the altitude. The calibration of the distance sensors is of crucial importance to obtain good measurements. For this purpose, the sensors are calibrated by optimizing a nonlinear transfer function and a bias function using a least squares method. As a consequence, the covariance of the sensors is approximated with a second order function of the distance. The two sensors have two different operation ranges that overlap in a small region. A switch strategy is developed in order to obtain the best performances in the overlapping range. As a result, a single error model function of the distance is found after the evaluation of the switch strategy. Because of different environmental factors, like the temperature, a bias drift is evaluated for both the sensors and properly taken into account in the algorithm. In order to have information of the surface and to use it in the navigation algorithm, a Digital Elevation Model of the simulated

[*] Corresponding author

[1] Postdoctoral Fellow, Italian Space Agency

[2] MSc Astronautical Engineer, Department of Astronautical Electrical and Energy Engineering, Sapienza University of Rome

[3] Associate Professor, Department of Astronautical Electrical and Energy Engineering, Sapienza University of Rome

lunar surface has been carried out. The navigation algorithm is designed as an Extended Kalman Filter which uses the altitude measurements, the Digital Elevation Model and the accelerations measurements coming from the moving frame. The objective of the navigation algorithm is to estimate the position of the simulated space vehicle during the landing from an altitude of 3 km. Because the algorithm needs to be updated during the landing, a crater peak detector is conceived in order to reset the navigation filter with a new state vector and new state covariance. Experimental results of the navigation algorithm are presented in the paper.

Keywords: Terrain Relative Navigation Algorithm, Lunar Lander, Cartesian Robot, Simulated Lunar Scenario

1. Introduction

In the space branch there is an high interest in the autonomous exploration missions which include the problem of the autonomous landing of a space vehicle. The present work is focused on the study of the lunar landing by exploiting a Simulated Lunar Scenario that has been built to reproduce a scaled lunar surface which includes a tri-axial moving frame (a Cartesian Robot) in order to simulate the landing trajectories [1]. At the tip of the tri-axial moving frame a long-range and a short-range infrared distance sensors are mounted in order to measure the altitude of the space vehicle from the surface.

Space agencies are deeply involved in developing projects regarding autonomous navigation, involving navigation cameras [13] or LIDARs [14]. The simulation of the final phase of the landing trajectory is of crucial importance, therefore software [16] and hardware [15] facilities have been used to test the algorithms. Optical Navigation seems to be one of the most promising techniques, see [17], [18], [19]; in this complex field, the concept of Optical Flow deserves a primary role [20] [23].

The altimetry technique is also largely used in space programs, such as the study of the planets' morphology or the study of the topography, especially of the Moon to answer on questions concerning lunar origin and evolution [2].

This technique is also used to capture topographic data of the Martian soil, required in preparation for the eventual deployment of rover vehicles and establishment of Martian bases [2].

The present study is based on a sort of Terrain Relative Navigation (TRN) method. The principle of this method lies on the localization of the space vehicle through the observations of known characteristics of a map. This technique is used when GPS is not available, such as for cruise missile navigation or space robot localization [3]. TRN has been used and tested also with helicopters [21] and Autonomous Underwater Vehicles (AUVs) [22].

TRN systems have the purpose to assist a space vehicle during the landing phase to execute a precise and safe landing. The systems use active range sensors to collect the altitude values of the surface under the lander and compare it with a Digital Elevation Model (DEM) database [4]. For this purpose a DEM of the simulated lunar surface has been carried out using the infrared (IR) sensors in order to have the database of the flying area. A comparison between real time data obtained from the distance sensors and the database [5] gives a position information that is an input of the navigation algorithm. A "crater peak detector" strategy is developed in order to update the navigation algorithm. This part of the algorithm computes the local inclination of the terrain to find the highest inclination points in the database; when such points are reached and recognized, the navigation filter is reinitialized.

The landing trajectory imposed to the lander is obtained from the acceleration phases of the Next-Moon mission [7]. The selection of the landing site, evaluated according to [11], has been carried out to maximize the scientific return of the mission.

The experimental facility presented in this paper has also been used to test optical navigation algorithms [24].

2. Experimental setup

The experimental facility is composed by a tri-axial moving frame (a Cartesian Robot) and a a simulated lunar scenario. Two IR distance sensors are mounted on the tip of the cartesian robot in order to measure the altitude. Because currently the tip is not able to change its orientation, in the following we do not consider the simulation of the lander attitude.

Cartesian Robot - The cartesian robot is actuated by stepping motors (laid on the principal axes of the robot) with holding bipolar torque of 820 N·cm. The frames of the robot are in aluminium. Due to the maximum linear speed of the robot, especially along z axis that is of 2 cm/s, the simulation time is 4 times larger than the real landing time.

Simulated Lunar Scenario - The cartesian robot is installed on a reproduction of the lunar equatorial zone located at the Mare Serenitatis (23 North - 14 deg East) with sizes 3 m x 4 m that corresponds to a dimension scale of 1:2000, see Fig. 1. The lunar soil simulant is made by sifted basalt powder, while the craters are made by calk using molds with sizes and shapes according to [12].

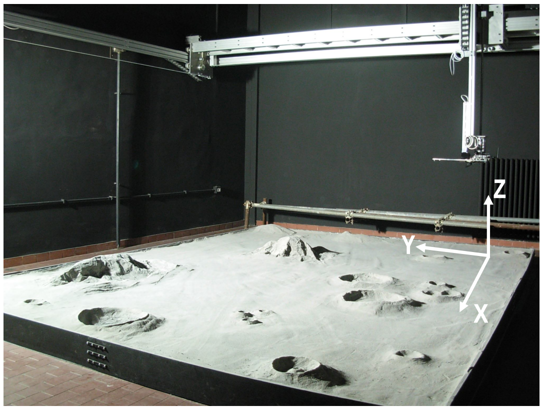

Figure 1: The experimental setup and the scenario.

IR Distance Sensors - Two IR distance sensors are used:

- a Long Range Sensor (LRS) with range from 150 cm to 20 cm (SHARP model *GP2Y0A02YK0F*), simulating the scaled range from 3000 m to 400 m;
- a Short Range Sensor (SRS) with range from 30 cm to 4 cm (SHARP model *GP2D120XJ00F*), simulating the scaled range from 600 m to 16 m;

for a total range from 150 cm to 4 cm with an overlapping zone between 30 cm and 20 cm. The IR sensors are equipped with an adapter and an interface to communicate with the main computer with wireless connection.

3. Calibration of IR distance sensors

3.1. Measurements and Noise Models

The aim of this section is to find the suitable *transfer function*, that relates the output of the IR sensors with the distance. In addition, the error function between the measurements and the true values of the distance is found.

The calibration of the IR distance sensors is obtained using a least squares method. The measurement data are acquired on the lunar scenario (the ground is a grey surface) for each IR sensor. The sensors are ratiometric: this means that their outputs are proportional to the supply voltage. The sensors are connected to an analog-to-digital converter that yields a *SensorValue*, which represents a digital value with a resolution of 12 bit (varying from 0 to 4095). The measured distance is a function of the inverse of the output voltage, see the datasheet, so it is a function of the inverse of the *SensorValue*.

The steps for the sensors' calibration are:

- find the *transfer function*
- obtain a zero mean error
- find the *error function*.

The calibration process is presented using the measurements from the LRS. The same calibration process is applied to the SRS, whose only the final results are presented. The calibration data are obtained using the full operative range of the LRS, 20-150 cm . The tip of the cartesian robot moves with a vertical constant velocity of 1 cm/s, and the acquired data are fitted to the true values using a least squares method. In order to find the suitable *transfer function* F, polynomial and exponential functions of the acquired data are used as in Table 1. Defining $x_{data} = 1/SensorValue$, it is possible to select F as what gives the minimum mean quadratic error $\overline{\epsilon^2}$, expressed as:

$$\overline{\epsilon^2} = \frac{\sum_{k=1}^{n_{points}} (F(x_{data_k}) - x_{true_k})^2}{n_{points}} \tag{1}$$

where n_{points} is the number of the acquired data and x_{true} is the true value of the altitude, commanded to the cartesian robot. In our case $n_{points} = 26010$, that corresponds to ten full range tests with a 20 Hz sensor frequency. For each function in Table 1, the best values of the constants C_i are found through the technique based on a *"Trust Region Approach"* [6]. The best choice is the function with the minimum $\overline{\epsilon^2}$, and, as a consequence, the *transfer function*

$Function$	$\overline{\epsilon^2}[cm^2]$
$F_1 = C_1 + C_2 x_{data}$	6.7565
$F_2 = C_1 + C_2 x_{data} + C_3 x_{data}^2$	4.6705
$F_3 = C_1 + C_2 x_{data} + C_3 x_{data}^2 + C_4 x_{data}^3$	4.3959
$F_4 = C_1 + C_2 x_{data} + C_3 x_{data}^2 + C_4 x_{data}^3 + C_5 x_{data}^4$	4.6704
$F_5 = C_1 + C_2 e^{x_{data}}$	6.9264
$F_6 = C_1 + C_2 e^{C_3 x_{data}}$	8.4824

Table 1: Transfer functions and related mean quadratic errors.

that we use is the cubic function F_3, with the constants' values shown in Eq.(2). Fig. 2 shows the behaviour of F_3 (measurements) and the true values as functions of the distance.

$$F_{LRS} = 5.300 \cdot 10^{-2} + 1.060 \cdot 10^4 x_{data} + 2.441 \cdot 10^5 x_{data}^2 - 1.634 \cdot 10^7 x_{data}^3 \quad (2)$$

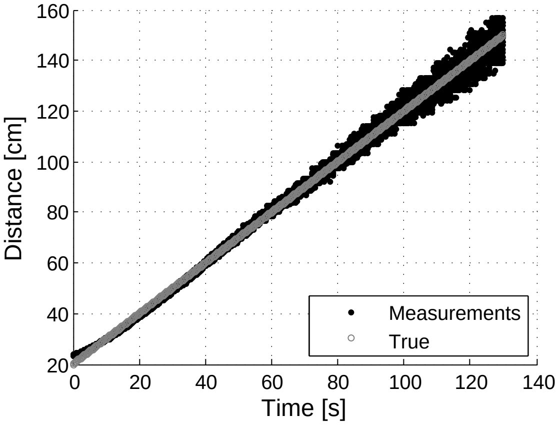

Figure 2: The values obtained with the cubic transfer function in comparison with the true data for the LRS' calibration.

Now we need to find the *error function* between F_3 and the true values in order to have zero mean errors. In Fig. 3 the black points show the errors as a function of the distance, while the grey line is the interpolation with a polynomial function that fits at the best the errors in a least squares sense. An iterative method has been applied in order to find the solution of the interpolation. The method considers polynomials with increasing order that stops when the percent variation of the mean square value between two succesive iterations is below 0.1%. As a result, the *error function* $\tilde{e}$ is found as a polynomial of 8^{th} degree, see the grey line in Fig. 3, where $\tilde{e} = \sum_i \lambda_i d^i$, d is the distance, and the coefficients λ_i are given in Table 2.

λ_0	λ_1	λ_2	λ_3	λ_4
20.442	-1.282	$2.486 \cdot 10^{-2}$	$-2.758 \cdot 10^{-4}$	$7.280 \cdot 10^{-6}$
λ_5	λ_6	λ_7	λ_8	
$-1.705 \cdot 10^{-7}$	$1.916 \cdot 10^{-9}$	$-1.004 \cdot 10^{-11}$	$2.000 \cdot 10^{-14}$	

Table 2: λ coefficients of the polynomial $\tilde{e}$ that fits the error of the LRS.

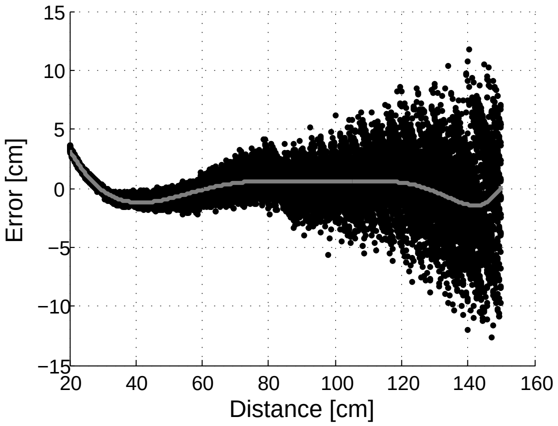

Figure 3: LRS error (black points) and the interpolation of the error (grey line)

Subtracting $\tilde{e}$ by the measurements, it is possible to obtain a zero mean error that is useful for the navigation filter. Experimental tests have demonstrated that the sensors show an error that is a quadratic function of the distance. This deviation is reported in Fig. 4 (black points) where it is possible to notice that it has a behaviour similar to a parabola. A second order function that yields the deviation at any altitude is obtained with an interpolation at three distances, marked with star points in Fig. 4: at the extremes of the range and at an intermediate value: 20 cm, 150 cm and 90 cm. The data to be interpolated correspond to the 3σ deviation at the three selected distances, see Table 3. The tests at these distances have been conducted using 1001 samples acquired at 20 Hz, with the cartesian robot at rest. The resulting parabola is described by the following equation:

$$3\sigma_{LRS}(d) = -0.209 + 2.883 \cdot 10^{-2}d + 3.249 \cdot 10^{-4}d^2 \tag{3}$$

LRS	3σ $[cm]$
$d_1 = 20$ cm	0.497
$d_2 = 90$ cm	5.017
$d_3 = 150$ cm	11.426

Table 3: 3σ values for LRS at the three selected distances.

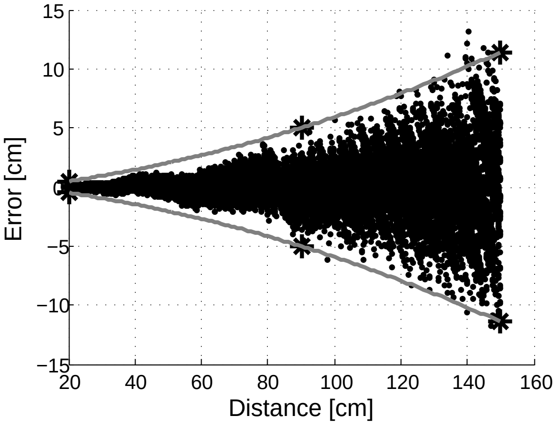

Figure 4: LRS zero mean error (black points) and the 2^{th} order function of the deviation (grey line).

The calibration procedure for the SRS sensor is the same for the LRS sensor. As a result, the *transfer function* is:

$$F_{SRS} = -0.791 + 2.682 \cdot 10^3 x_{data} + 1.558 \cdot 10^4 x_{data}^2 - 3.080 \cdot 10^6 x_{data}^3 \tag{4}$$

while the *error function* is a polynomial of 11^{th} degree, with the coefficients given in Table 4.

The data to be interpolated to find the second order function of the 3σ deviation are shown in Table 5, while the expression of the quadratic function of the error is:

$$3\sigma_{SRS}(d) = 0.385 - 2.047 \cdot 10^{-2}d + 1.392 \cdot 10^{-3}d^2 \tag{5}$$

λ_0	λ_1	λ_2	λ_3	λ_4
19.360	-17.937	7.142	-1.601	0.223
λ_5	λ_6	λ_7	λ_8	λ_9
$-2.024 \cdot 10^{-2}$	$1.221 \cdot 10^{-3}$	$-4.905 \cdot 10^{-5}$	$1.291 \cdot 10^{-6}$	$-2.123 \cdot 10^{-8}$
λ_{10}	λ_{11}			
$1.964 \cdot 10^{-10}$	$-7.727 \cdot 10^{-13}$			

Table 4: λ coefficients of the polynomial $\tilde{e}$ that fits the error of the SRS.

SRS	3σ $[cm]$
$d_1 = 4\ cm$	0.325
$d_2 = 17\ cm$	0.439
$d_3 = 30\ cm$	1.024

Table 5: 3σ values for SRS at the three indicated distances.

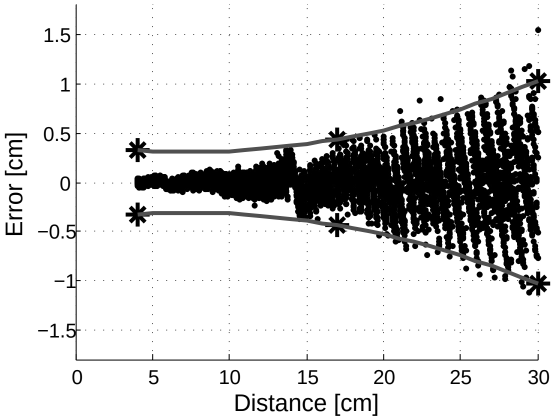

Figure 5: SRS zero mean error (black points) and the 2^{th} order function of the deviation (grey line).

The results are showed in Fig. 5.

The sensors have a common working range between 20 and 30 cm, and as a consequence, the sensor with the minimum deviation in this range has been chosen. The LRS has a slightly smaller deviation in the entire common range (see Fig. 6), and it has been chosen as the reference sensor for that range. The SRS has been therefore re-calibrated within the new range 4-20 cm, being 20 cm the switch distance. The error of the two sensors in the overlapping range are very similar, so a single 3σ deviation formula has been obtained by interpolating the deviation at the extreme distances and at the switch distance with a second order function, see Fig. 7.

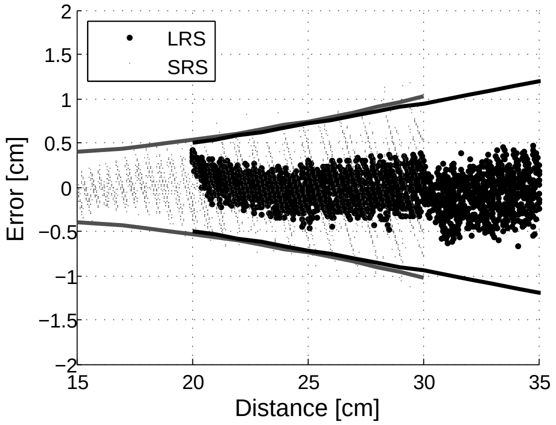

Figure 6: Common working range: the error and the second order function of SRS (grey) and LRS (black).

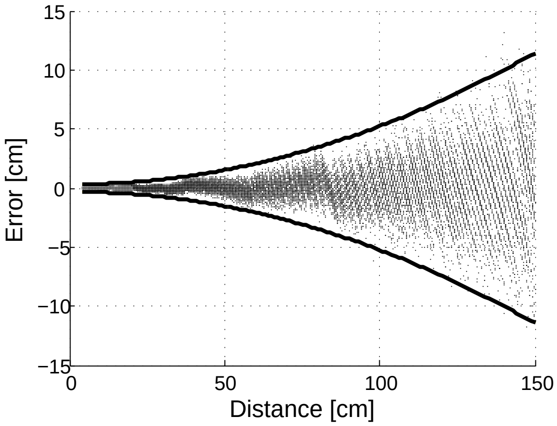

Figure 7: The second order function obtained using the two IR distance sensors as a single sensor.

The Eq. (6) represents a very useful equation to obtain the covariance matrix R for the Kalman filter:

$$3\sigma_{distance}(d) = 0.3001 - (4.2728 \cdot e^{-5})d + (4.9475 \cdot e^{-4})d^2 \qquad (6)$$

Several tests have shown that the sensors' measurements were affected by environmental conditions, especially the temperature. As a consequence, periodical tests were conducted in different days at different times to evaluate the environmental effects. As an example, Fig. 8 shows the *SensorValue* for the LRS at 150 cm: the points in the figure are the result of 1001 samples at the fixed altitude. Fig. 8 shows the mean values (black circles) and the three sigma errors (grey circles). The black line is the interpolation of the mean values to show the behaviour of the mean value. The noon for every day is reported on the x axis (the test started at 17:00 of the first day). In Fig. 9 the black circles represent the three sigma errors, while the grey line is the interpolation. As above, the noon for every day is reported on the x axis.

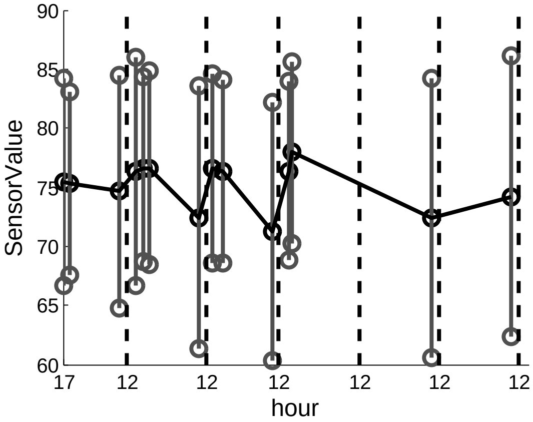

Figure 8: The effect of environmental conditions on LRS at 150 cm: mean values and related 3σ.

The collected data have been used to find a linear relation that represents a bias term in the covariance matrix R into the Kalman filter:

- Long Range Sensor's bias term

$$\sigma_{biasLRS}(d) = 0.2340 + 0.0736d \tag{7}$$

- Short Range Sensor's bias term

$$\sigma_{biasSRS}(d) = 0.1642 + 0.0036d \tag{8}$$

In this case there is an equation for each sensor, and this involves a discontinuity when the sensors switch at 20 cm. To consider additional disturbances effects on the sensor, it is introduced another term is included in the covariance matrix of the EKF. This term takes into account:

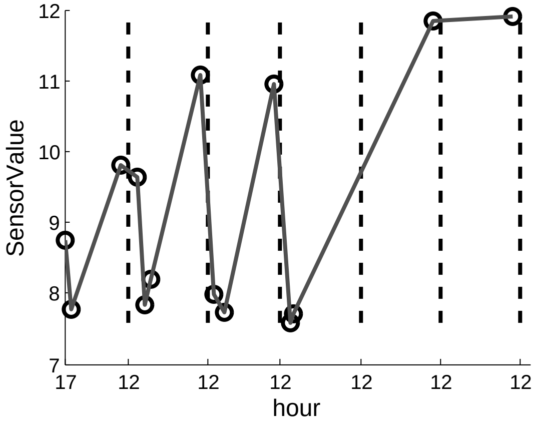

Figure 9: The effect of environmental conditions on the standard deviation for LRS at 150 cm .

- the vibration of the Cartesian manipulator;
- the positioning error of the Cartesian manipulator;
- the clock synchronization.

This term, named σ_{noise} and expressed in cm, has been estimated through several tests conducted on the Cartesian manipulator and it has been evaluated to be:

$$\sigma_{noise} = 0.2 \tag{9}$$

The covariance matrix R is the quadratic sum of three terms, two common for both the IR distance sensors and one dependant on which sensor is active:

- R for the LRS is:

$$R = \sigma^2_{distance} + \sigma^2_{biasLRS} + \sigma^2_{noise} \tag{10}$$

- R for the SRS is:

$$R = \sigma^2_{distance} + \sigma^2_{biasSRS} + \sigma^2_{noise} \tag{11}$$

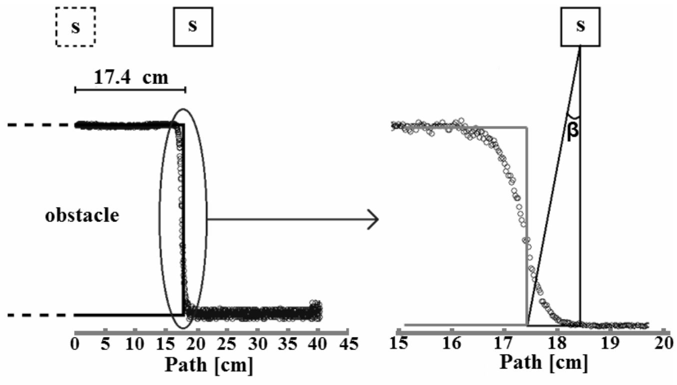

Figure 10: Experimental setup to study the sensor's FOV and footprint.

3.2. Sensors Field Of View Model

The Field Of View model of LRS and SRS has to be found in order to characterize the sensors.

In Fig. 10 the sensor is posed at a known distance from a cube-shaped obstacle with known sizes, and it is moved from the start point towards right with a velocity of 0.5 cm/s. The measurements are acquired at the sampling rate of 20 Hz. When the sensor FOV is crossing the upper obstacle corner, the distance measurement starts to increase, revealing the presence of the edge of the obstacle. When the FOV is crossing the lower obstacle corner, the measurement becomes constant, as shown in the right side of Fig. 10; at this point the angle β of the sensor, representing the half FOV, can be computed:

$$\beta = 0.9874^\circ \tag{12}$$

The footprint of the IR distance sensors is a circle, the size of this circle is a function of the altitude and it is given by the angle β [9]. To find the footprint equation of the sensor we have conducted ten measurements at different altitudes above the obstacle. Beginning at the known distance of 24.9 cm and decreasing it of 0.5 cm after each measurement, we have found the following footprint equation:

$$\phi(d) = 0.0345d + 0.0039 \tag{13}$$

where $\phi(d)$ is the estimated diameter of the footprint.

4. Digital Elevation Model of Simulated Lunar Surface

The DEM is realized with the IR distance sensors to obtain a model of the ground, see the picture in Fig. 11. It is used to choose the suitable landing sites and to obtain the slope map. The DEM and the slope map are stored in the database for the Kalman Filter.

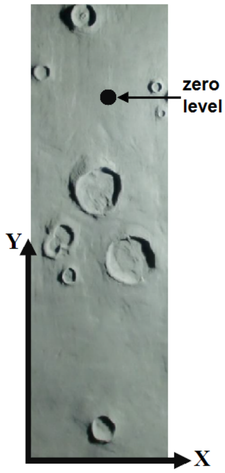

Figure 11: The reproduction of the Moon with the reference frame adopted and the zone considered as zero level.

The point named *zero level* in Fig. 11 is an arbitrary landmark chosen to be the zero reference for the altitude measurements.

The DEM is obtained moving the cartesian robot at constant velocity of 2 cm/s along the y direction with IR measurements' update rate of 10 Hz. This means that the grid along this direction is spaced of 0.2 cm. The grid spacing in the x direction is of 1 cm. Because the acquired DEM is not equispaced, the data have been interpolated with bi-dimensional splines in order to have an equispaced grid. After the interpolation a median filter has been applied to minimize the sensor noise, see Fig.12.

From the DEM and the slope map, a safe touchdown point can be determined based on two parameters:

- the highest acceptable slope (α), chosen equal to 10°;
- the side of the squared area (l) that contains the touchdown point, chosen equal to 10 cm (corresponding to 200 m in the actual lunar scale).

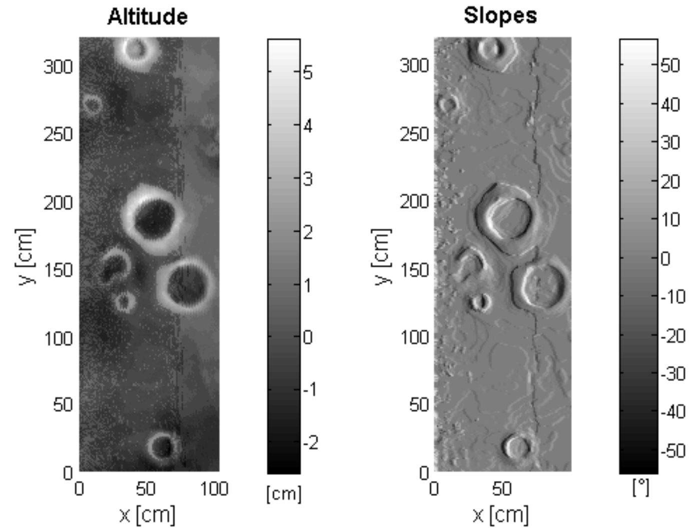

Figure 12: DEM altitude (on the left side) and slopes (on the right side) after the interpolation of data and the application of the median filter.

The results are reported in Fig. 13. These values have been chosen according to the requirements of typical case of lander missions [11] [7].

The values of this two parameters could be changed to see how they affect the reachable landing sites, see Fig. 14. The α angle has been changed from 5° to 20°, while the length l between 5 and 20 cm. The choice of α and l can be suitable obtained through a trade-off between the variation of this values.

The safe landing site is selected a crater in order to fulfil the Next - Moon mission's requirements [7], see Fig. 15.

The DEM is used in real time to compare the terrain altitude data. Tests have demonstrated that during the landing phase the IR sensors'data were useless up to an altitude of 75 cm (1500 m in scale). It means that the comparison between the real time data and the DEM archived data can be done only below this altitude. This effect is admisible with respect to Next - Moon mission: as a matter of fact this is the altitude allowable for the second retargeting [7].

5. Navigation Algorithm

This section faces on the navigation of a space vehicle during the last phase of landing. In particular we consider only a in-plane trajectory with

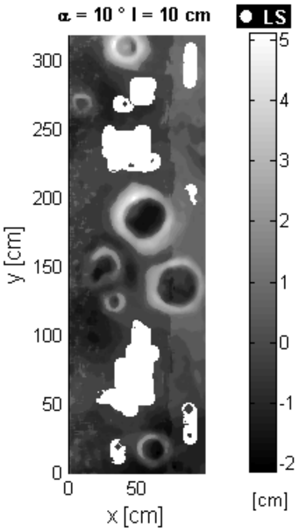

Figure 13: Safe Landing sites (white points) obtained with $\alpha = 10°$ and $l = 10\ cm$.

a given thrust profile for the engines. The navigation algorithm is based on a discrete Extended Kalman Filter (EKF) which uses as inputs the distance measurements, the acceleration measurements and the DEM. The equation of the dynamics is the equation of a body in a constant gravitational field with engines capable of applying a sequence of thrusts for fixed times in horizontal and vertical directions. With these assumptions the state variable of the system is $s = [z\ \dot{z}\ y\ \dot{y}]^T$, where y and z are respectively the along track and the vertical direction of the space vehicle, having the following equation of the dynamics:

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_z \\ a_y \end{bmatrix} + \begin{bmatrix} 0 \\ -g_M \\ 0 \\ 0 \end{bmatrix} \tag{14}$$

where g_M=1.622 m/s^2 for the Moon, and a_z, a_y are the accelerations in the along track and vertical directions acting on the lander. The accelerations are measured on-board by the lander. In our experimental tests, the accelerations are measured througt the cartesian robot which imposes the dynamics of the descending path in the simulated lunar scenario.

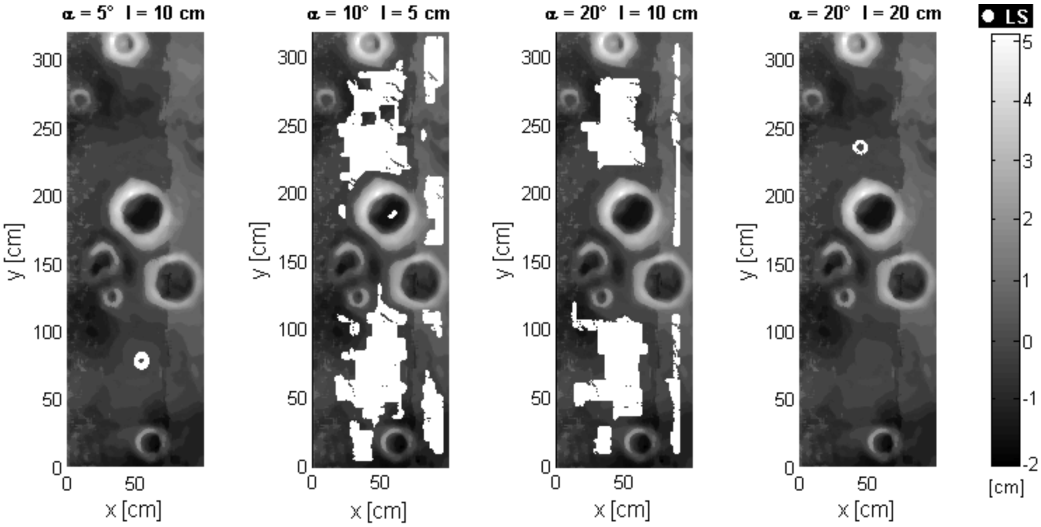

Figure 14: Safe Landing sites (white points) obtained varying α and l.

5.1. Discrete Model

In order to implement the navigation filter, the discrete models of the state transitions and the measurements are written at the sampling time $\tau = 0.1$ s. The expressions are:

$$s_{k+1} = A \cdot s_k + B \cdot u_k + G_k + w_k \tag{15}$$

$$m_k = z_k - DEM(y_k) + v_k \tag{16}$$

where:

- A is the transition matrix:

$$A = \begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

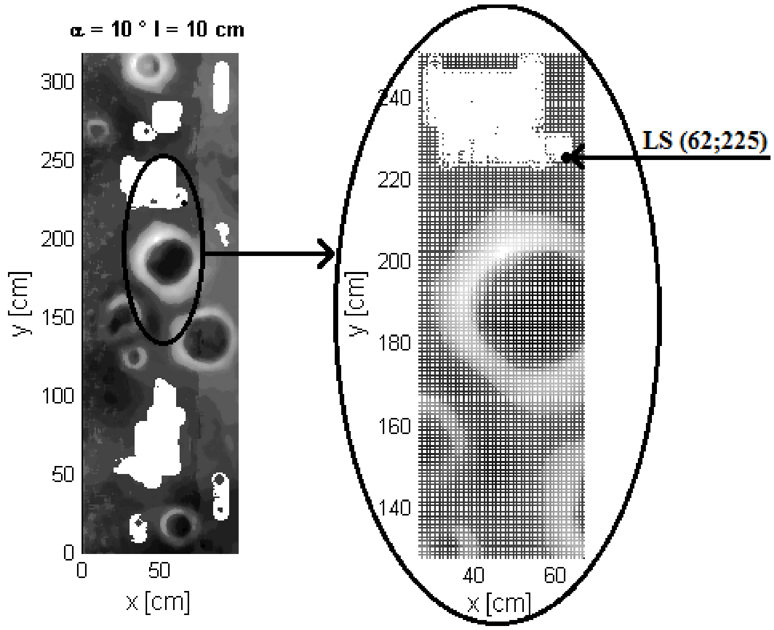

Figure 15: The chosen safe Landing Site(the black point in the white region).

- B is the matrix of distribution of the accelerations:

$$B = \begin{bmatrix} 0.5 \cdot \tau^2 & 0 \\ \tau & 0 \\ 0 & 0.5 \cdot \tau^2 \\ 0 & \tau \end{bmatrix} \tag{18}$$

- u_k is the vector of the acceleration measurements at time k:

$$u_k = \begin{bmatrix} a_z \\ a_y \end{bmatrix} \tag{19}$$

- G_k is the vector of the gravitational acceleration:

$$G_k = \begin{bmatrix} -0.5 \cdot g_M \cdot \tau^2 \\ -g_M \cdot \tau \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

In the case of the experimental tests we use a space scale of 1:2000 and a time scale of 4:1. The last results in scaling the simulated lunar gravitational acceleration to $g_M = 0.0051\ cm/s^2$.

- m_k is the altitude measurement at the time k. Because we have distance sensors (as the radar altimeter in the real case), the altitude measurement results as the difference between z and the DEM at the along track position y_k as stated in Eq.(16).
- w_k and v_k are random variables at the time k representing the process noise and the measurement noise respectively. These noises are assumed to be statistical indipendent, with zero-mean and Gaussian distribution[4]:

$$w_k \simeq N(0, Q) \tag{21}$$

$$v_k \simeq N(0, R) \tag{22}$$

5.2. Extended Kalman Filter

The navigation filter is based on the discrete *"predictor-corrector"* Extended Kalman Filter (EKF) approach. Let s_k^{pre} and m_k^{pre} be the predicted state and altitude estimates. The altitude estimate is given by:

$$\hat{m}_k = \hat{z}_k - DEM(\hat{y}_k) \tag{23}$$

that is related to the DEM as a function of $\hat{y_k}$. The resulting residual is the difference between the altitude and the altitude estimate:

$$m_k - \hat{m}_k = (z_k - \hat{z}_k) - DEM(y_k) + DEM(\hat{y}_k) + v_k \tag{24}$$

As a first approximation, neglecting the high order terms, we can write:

$$DEM(y_k) = DEM(\hat{y}_k) + J(\hat{y}_k) \cdot (y_k - \hat{y}_k) \tag{25}$$

where

[4] A Gaussian distribution is indicated with $N(\mu, \sigma^2)$ where μ is the mean of the distribution and σ^2 is the variance

$$J(\hat{y}_k) = \frac{DEM(\hat{y}_{k+1}) - DEM(\hat{y}_k)}{\Delta \hat{y}_k} \tag{26}$$

is the Jacobian of and $\Delta \hat{y}_k = \hat{y}_{k+1} - \hat{y}_k$ is a constant because the grid of the DEM is equispaced (see section 4). As a consequence, we can express the altitude residuals as:

$$m_k - \hat{m}_k = \begin{bmatrix} 1 & 0 & -J(\hat{y}_k) & 0 \end{bmatrix} \begin{bmatrix} z_k - \hat{z}_k \\ \dot{z}_k - \dot{\hat{z}}_k \\ y_k - \hat{y}_k \\ \dot{y}_k - \dot{\hat{y}}_k \end{bmatrix} = H_k \left(s_k - \hat{s}_k \right) \tag{27}$$

The covariance matrix Q is given by the sum of two terms:

- Q_p, related to the process noise, that throught several tests to tune the EKF is:

$$Q_p = \begin{bmatrix} 10^{-5} & 0 & 0 & 0 \\ 0 & 10^{-5} & 0 & 0 \\ 0 & 0 & 10^{-7} & 0 \\ 0 & 0 & 0 & 10^{-7} \end{bmatrix} \tag{28}$$

- Q_{acc}, related to the acceleration noise, that is added only when the accelerometer measures an acceleration, that is when the thrusters are activated.

This means that the matrix covariance Q is varying according to the thrusting profile for the descending path. As a result, the covariance matrix is given by the following equation:

$$Q = Q_p + Q_{acc} = Q_p + \delta \cdot B \cdot R_{acc} \cdot B^T \tag{29}$$

where

- δ is the on/off term, equal to 1 when the thrusters are on and to 0 when the thrusters are off. This solution is related to the activation of the accelerometer measurements: the accelerometer measurements are included and give an error only when the thrusters are on.

- R_{acc} is a 2x2 diagonal matrix having the acceleration noise (with a standard deviation equal to 0.0293 cm/s^2) on the diagonal. The value of the standard deviation noise is of a commercial accelerometer (*1044 PhidgetSpatial 3/3/3*) in the space and time scale of the experimental setup (see section 2) and increased ten times to get worse conditions.

The prediction and updating discrete-time equations are [10]:

- *predictor*: time update

  - predict state estimate

$$s_{k+1}^{pre} = A_k \cdot s_k + B_k \cdot u_k + G_k \tag{30}$$

  - predict covariance estimate

$$P_{k+1}^{pre} = A_k \cdot P_k \cdot A_k^T + Q_k \tag{31}$$

- *corrector*: measurement update

  - Kalman gain

$$K_k = P_k^{pre} \cdot H_k^T \cdot (H_k \cdot P_k^{pre} \cdot H_k^T + R_k)^{-1} \tag{32}$$

  - update estimate with the measurement m_k

$$s_k = s_k^{pre} + K_k(m_k - H_k \cdot s_{k+1}^{pre}) \tag{33}$$

  - update the error covariance

$$P_k = (I - K_k \cdot H_k) \cdot P_k^{pre} \tag{34}$$

The equation of the measurement (Eq. (16)) contains the information from the DEM, considering that it is a function of the coordinate y.

The *a priori* estimate error covariance P is given by a three sigma error of 100 m on the position vector components and of 3 m/s on the velocity vector components [7]. These values are scaled according to the space scale (1:2000) and the time scale (4:1) adopted for the experimental setup (see section 2).

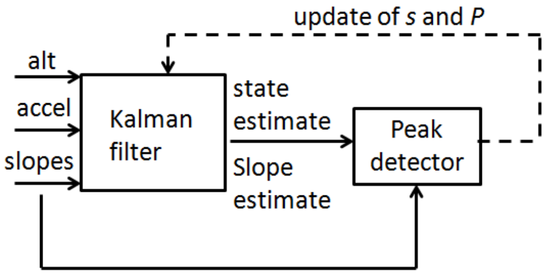

Figure 16: The peak detector scheme.

5.3. Peak Detector

As mentioned in section 5.2, the DEM archived data allow to update the position and the matrix covariance of Kalman filter through an algorithm developed for this pourpose, named ”peak detector”. This algorithm has as input the output of the Extended Kalman Filter, the estimated state and the slope, and has stored the archived data from DEM (see scheme in Fig. 16).

The comparison between the archived slopes and the computed slopes in real time starts at the altitude of 75 cm (corresponding to 1.5 km in real scale, according to the second retargeting of Next - Moon mission [7]). This comparison is obtained in the range $[y_k^{stim} - 3\sigma(y_k^{stim}), y_k^{stim} + 3\sigma(y_k^{stim})]$, where the apex indicates the value estimated by the EKF. When the peak detector identifies an equivalence between the archieved slopes and the real time slopes, it updates the coordinate y (the along track direction of the vector state s), with the value corresponding to the position of that slope in the database, and the element of the covariance matrix referred to y. This value is related to the error of the DEM, evaluated to be 0.2 cm, according to our accuracy in obtaining the DEM.

The slopes are computed at an i-th step at which the Kalman filter is updated. This step corresponds to an interpolation range: this means that the filter will start to analyze the output of the IR sensor after i values. In this way the filter will determine the regression line that is the interpolation between the values *k-(i-1)* and k. This linear regression of i values minimizes the possibility of an uncorrect correlation due to the errors of the altimeter with few samples. At this point it is possible to determine the slope of this regression line to compare with the archived slope obtained from DEM, at the same step i. This comparison permit to determine the position of the lander along the direction y.

6. Results

The landing trajectory, for the experimental results, is obtained by repeating the following two phases:

- free fall due to the lunar gravitational acceleration;
- ignition of the thrusters to decrease the lander velocity.

The sequence of the accelerations along y and z directions is showed in Fig. 17 and the resulting trajectory is showed in Fig. 18.

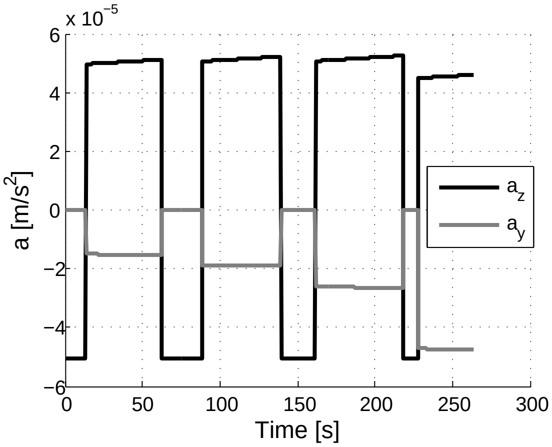

Figure 17: The acceleration components imposed to the Cartesian manipulator for the descent phase.

According to the acceleration profile, the velocity components along y and z directions are showed in Fig. 19.

Fig. 20 shows the symbolism that is adopted to define the altitude profile h as the difference between the altitude of the sensor with respect to the surface z and the value detected by the distance sensor d (Eq. (35)):

$$h = z - d \tag{35}$$

Fig. 21 shows the values of altitude (left side) and slopes (right side) obtained from the database, related to the flying zone. Fig. **??** shows the values of altitude (left side) and slopes (right side) obtained from the IR sensors during the landing phase.

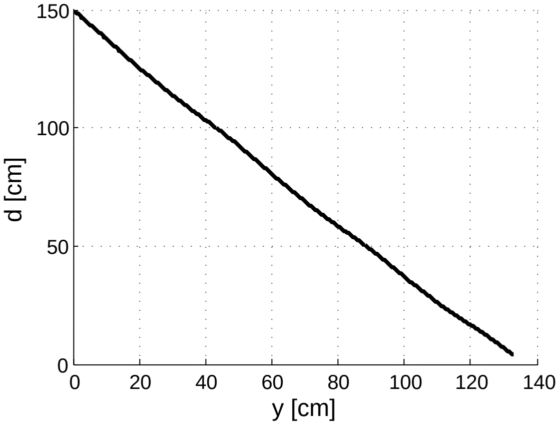

Figure 18: The landing trajectory imposed to the Cartesian manipulator.

From Fig. 22 it is possible to notice that the error related to the altitude of the IR distance sensor allows the usee of the same sensors only under the altitude of 75 cm (corresponding to 1.5 km in real scale). As a matter of fact over this altitude the noise of the sensors does not allow the evaluation of the slopes and, as a consequence, the comparison between the archived data and the real time data.

In Fig. 23 it is possible to see the comparison between the landing trajectory and the archived DEM: the dash grey line represents the limit of the use of the IR distance sensors:

The results of the estimation process using the Extended Kalman Filter for the reference landing trajecory are showed in the figures from Fig. 24 to Fig. 30. In these figures, the values related to the actual state are indicated with the subscript r, while the values related to the Kalman filter estimation are indicated with the subscript e. Futhermore the values indicated with σ in the figures' legend are obtained from the covariance matrix P of the EKF computed at each step.

The comparison between actual altitude of the cartesian robot during the landing phase (z_r) and the altitude estimated with the EKF (z_e) is given in Fig. 24.
It is possible to notice that both the trends are into the range $z_e \pm 3\sigma_{z_e}$. Furthermore it is pointed out the switch from the LRS to the SRS. On the right side of Fig. 24 there is enlarged the switch event to show that the estimation process remains in the 3σ range in both cases. The sensors have

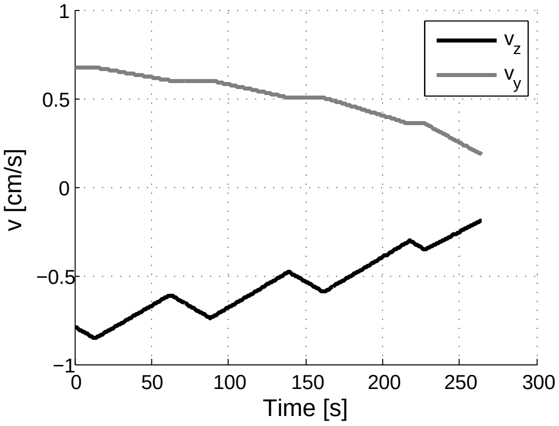

Figure 19: The velocity components imposed to the Cartesian manipulator for the descent phase.

to switch at an altitude equal to 20 cm (see section 3): in the figure it is possible to notice that the switch introduces a discontinuity in the estimation of the altitude. This phenomenon is due to the sensors' errors which involves a difference between the altitudes revealed by the LRS and the SRS. However the estimated altitude remains into the range $z_e \pm 3\sigma_{z_e}$. This is more clear in Fig. 25 where it is showed the error related to the altitude on the left side, with a zoom on the switch on the right side, given by Eq. (36):

$$e_z = z_r - z_e \tag{36}$$

Fig. 26 shows the comparison between the actual position along the y direction (y_r) and the position estimated by the EKF y_e. In the figure is pointed out when the EKF is updated by the peak detector (see section 5). To underline the effect of the peak detector on the right side of Fig. 26, there is a zoom of the update event.

In Fig. 27 it is drawn the error related to the position along the y direction on the left side, with a zoom on the update on the right side, given by :

$$e_y = y_r - y_e \tag{37}$$

Observing Fig. 27 the error in the along the track direction is diverging until the update event, while it converges to zero after the update.

Fig. 28 shows the DEM in the database (DEM_{load}) and the DEM obtained in real time (DEM_{rt}). This figure confirms the useless of the IR

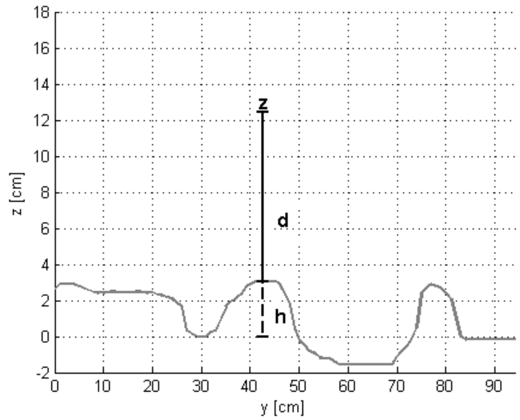

Figure 20: An example about the symbolism adopted.

distance sensors up to a distance of 75 cm, corresponding to 65 cm along the y direction. Furthermore, it is underlined the effect of the peak detector algorithm through the update of the y direction (on the right side of Fig. 28). As a matter of fact, it is possible to notice that the profile obtained in real time is more or less the same of that in database, but with a delay in the y direction: revealing a match between the slope in real time and in the database (clearly shown in right figure) the peak detector updates the y value, causing a shift of the real time data (towards right in this case).

Fig. 29 and Fig. 30 show the errors related to the velocities along z (Eq. (38)) and y (Eq. (39)) directions respectively, given by the difference between the true velocity (v_r) and the estimated velocity (v_e):

$$e_{v_z} = v_{z_r} - v_{z_e} \tag{38}$$

$$e_{v_y} = v_{y_r} - v_{y_e} \tag{39}$$

Both the figures show that the errors are constant and small in spite of the absence of velocity sensors. It is worth noting that the error related to the velocity v_z is constant and close to zero, while the error related to v_y has the same behaviour of the position error. As a matter of fact this error is initially divergent and it is convergent when the peak-detector updates the EKF.

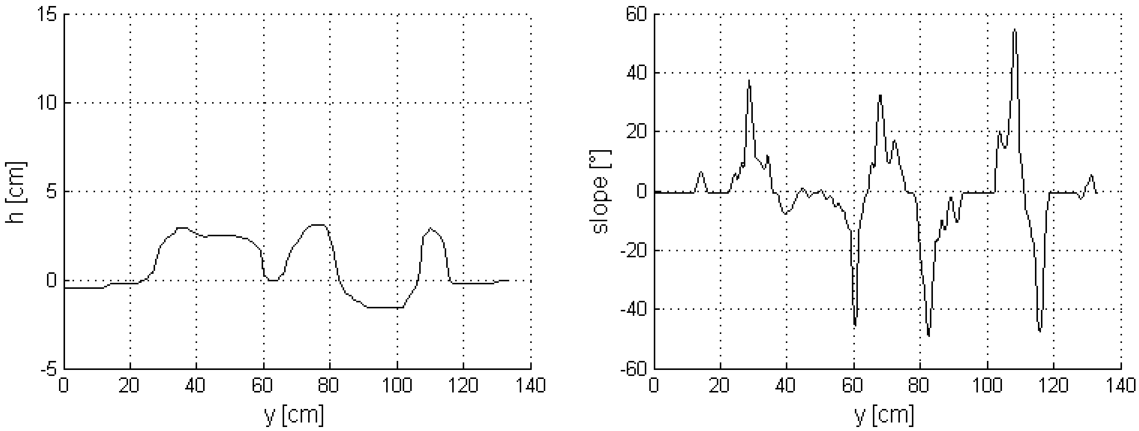

Figure 21: The altitude profile on the left side and the slopes on the right side obtained from the archived DEM.

7. Conclusion

A Terrain Relative Navigation algorithm for a lunar lander has been developed using the information data of DEM and distance sensors' measurements. A calibration procedure of the distance sensors has been set up in order to enhance the accuracy of the state estimate. The algorithm has been tested using a test-bed reproducing a scaled site of Moon and able to simulate a descending path through a cartesian robot. The navigation algorithm is based on the *predictor-corrector* Extended Kalman Filter, with the capability of recognizing the variation of altitudes of a crater rim. This feature, named *peak-detector*, allows to better estimate the along track position coordinate of the descending trajectory, that is not measured, through the comparison with the stored DEM data. The result is a navigation filter with a varying covariance matrix which uses the DEM as measurements. The experimental tests have been implemented for an in-plane trajectory. A future work will concern to apply the algorithm for trajectories that will include out of plane manuvers.

8. Bibliography

[1] Curti, F., Parisse, M., Ansalone, L., et al. *Multi-Purpose Experimental Test-bed for Space and Planetary Exploration*, 2nd IAA Conference on Dynamics and Control of Space Systems, March 24-26 2014, Rome, Italy, Paper IAA-AAS-DyCoSS2-14 -12-01.

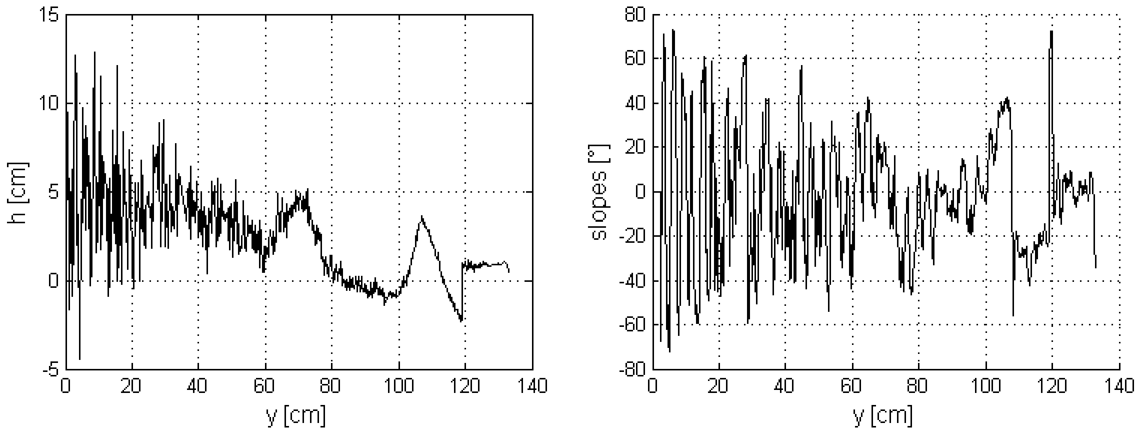

Figure 22: The altitude profile on the left side and the slopes on the right side obtained in real time during the landing phase from the IR distance sensor.

[2] Bufton, J. L., *Laser Altimetry Measurements from Aircraft and Spacecraft*, Proceedings of the IEE, vol. 77, no. 3, March 1989.

[3] Carreno, S., Wilson, P., Ridao, P., Petillot, Y., *A Survey on Terrain Based Navigation for AUVs*, OCEANS 2010, September 2010.

[4] Temel, ., Unaldi, N., Ince, F., *Novel Terrain Relative Lunar Positioning System Using Lunar Digital Elevation Maps*, Recent Advances in Space Technologies, June 2009.

[5] Johnson, A. E., Montgomery, J. F., *Overview of Terrain Relative Navigation Approaches fo Precise Lunar Landing*, Aerospace Conference, 2008 IEEE Volume, Issue, 1-8 March 2008.

[6] Coleman, T.F. and Y. Li, *An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds,” SIAM Journal on Optimization*, Vol. 6, 1996.

[7] Melloni, S., Mammarella, M., Gil-Fernndez, J., Colmenarejo, P., *GNC solution for lunar pin-point and soft landing*, GLUC-2010.1.5.5721 Moon robotic exploration, May 2010.

[8] ESA, NEXT Team, *Next Lunar Lander with In-Situ Science and Mobility: Phase A mission Study. Lunar Enviroment Specification*, NEXT-LL-LES-ESA(HME)-0001, 2008.

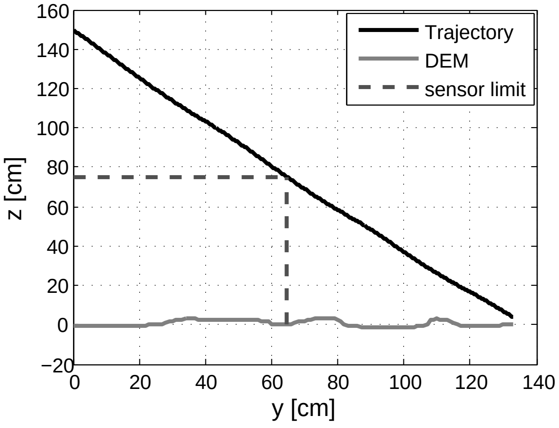

Figure 23: The sensor limit due to the noise of the IR distance sensors.

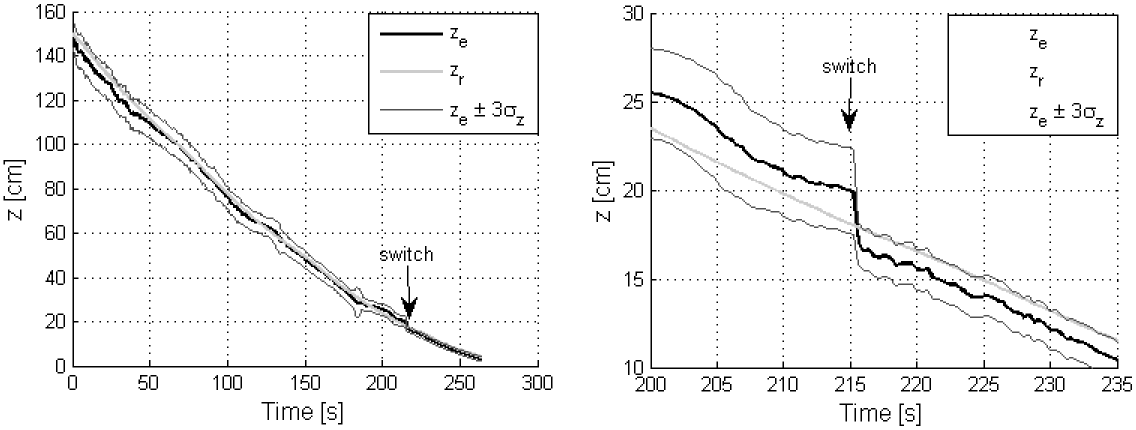

Figure 24: Landing phase: comparison between the true altitude (z_r) and the estimated altitude (z_e) on the left and the zoom on the switch on right side.

[9] Malheiros, P., Goncalves, J., Costa, P., *Towards a more accurate infrared distance sensor model*, INESC Porto Manufacturing Systems Engineering Unit, Faculty of Engineering of the University of Porto.

[10] Welch, G., Bishop, G., *An Introduction to the Kalman Filter*, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, September 1997.

[11] De Rosa, D., Bussey, B., Cahill, J.T., et al., *Characterisation of Potential Landing Sites for the European Space Agencys Lunar Lander Project*,

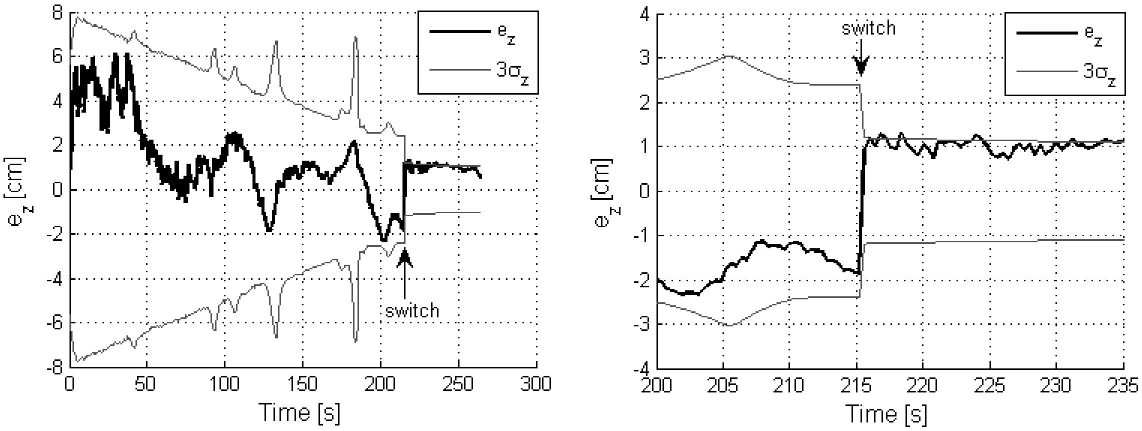

Figure 25: Landing phase: the error related to the altitude on the left side and the zoom on the switch on the rigth side.

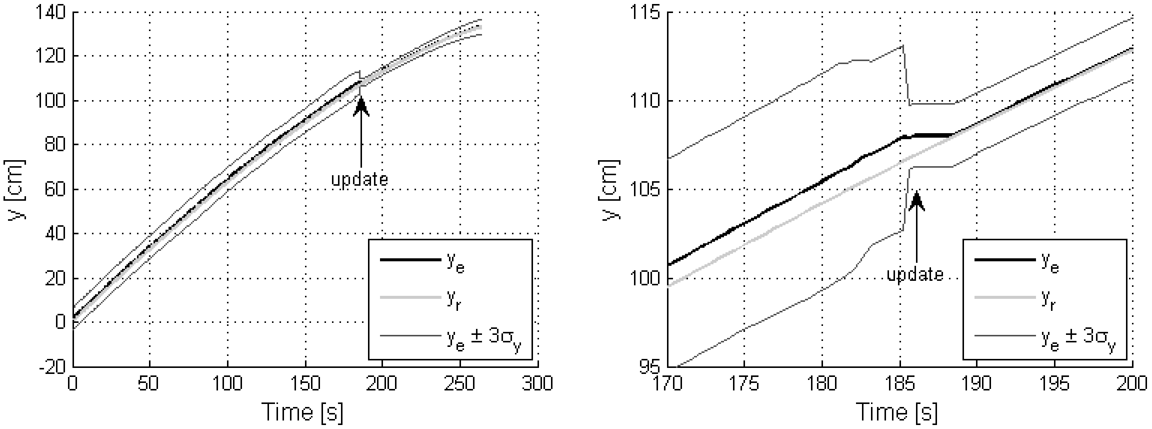

Figure 26: Landing phase: comparison between the true position along y direction (y_r) and the estimated (y_e)

Planetary and Space Science, Volume 74, Issue 1, December 2012, Pages 224-246.

[12] *Lunar Sourcebook, a Users Guide to the Moon*, Cambridge University Press, Heiken, D. Vaniman, B.M French, 1991

[13] Frapard, B., Mancuso, S., *Vision Navigation for European Landers and the NPAL project*, 6th International ESA Conference on Guidance, Navigation and Control Systems, Loutraki Greece, October 2005.

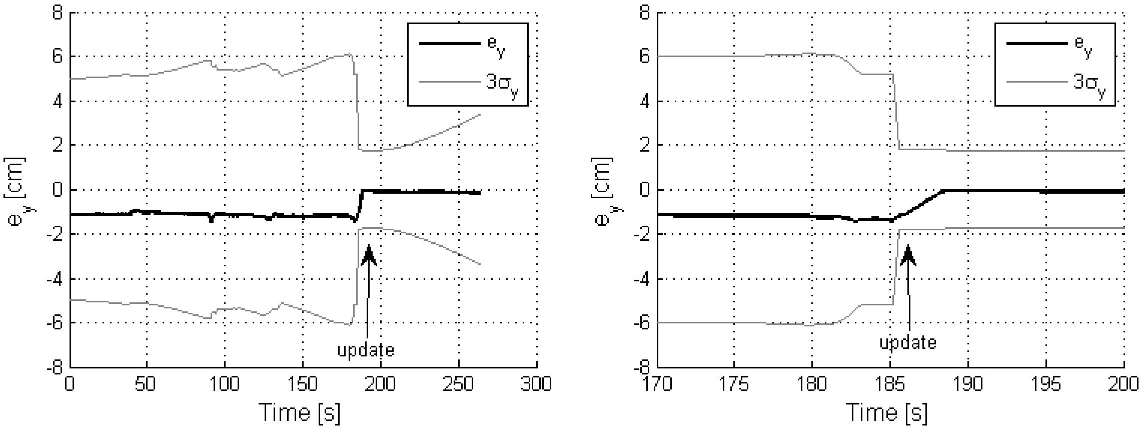

Figure 27: Landing phase: the error related to the along track direction.

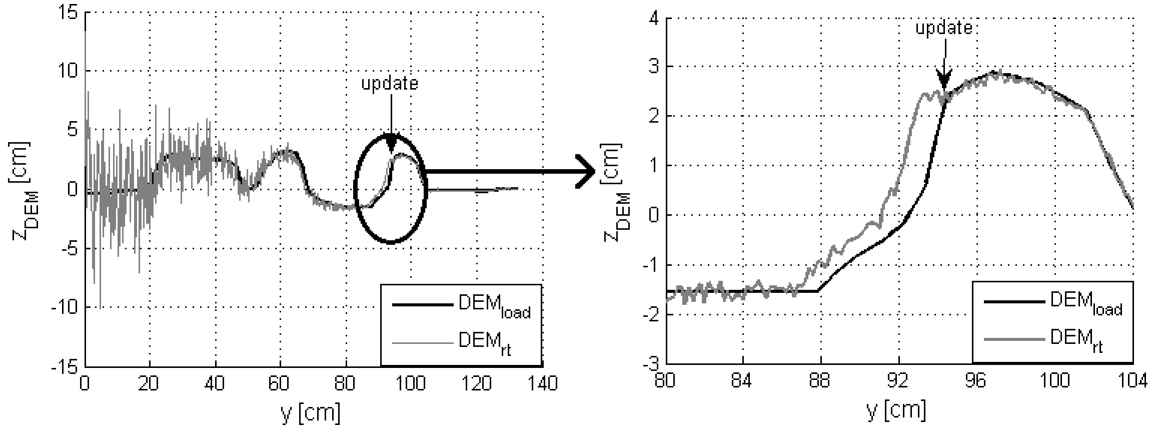

Figure 28: Landing phase: the comparison between the DEM in database (DEM_{load}) and DEM in real time (DEM_{rt}) on the left side and a zoom on the update event.

[14] Striepe, S.A., Epp C.D., Robertson, E.A., *Autonomous Precision Landing and Hazard Avoidance Technology (ALHAT) Project Status as of May 2010*, International Planetary Probe Workshop 2010, Barcelona Spain, June 2010.

[15] Kruger, H., Theil, S., *TRON - Hardware-in-the-loop test facility for lunar descent and landing optical navigation*, 18th IFAC Symposium on Automatic Control in Aerospace, Nara Japan, 2010.

[16] Parkes, S.M., Martin, I., Dunstan, M., Matthews, D., *Planet Surface*

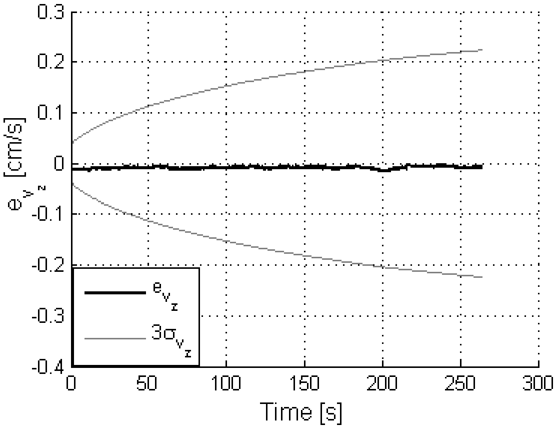

Figure 29: Landing phase: the error related to the velocity along vertical direction z.

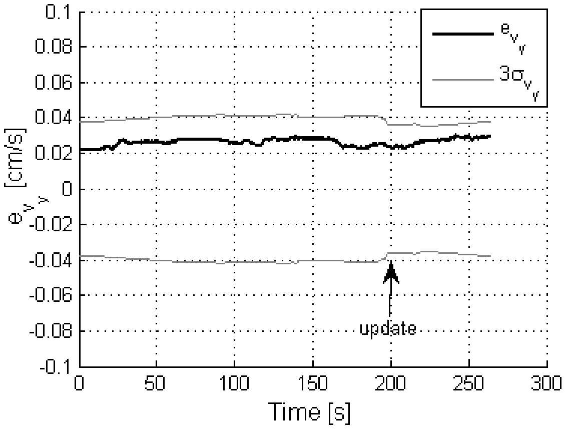

Figure 30: Landing phase: the error related to the velocity along track direction y.

Simulation with PANGU, 8th International Conference in Space Operation, May 2004.

[17] Verveld, M.J., *Relative Optical Navigation for a Lunar Lander Mission*, Advances in Aerospace Guidance, Navigation and Control, 2013, pp 661-679.

[18] Johnson, A.E., Ansar, A., Matthies, L.H. et al., *A general approach to terrain relative navigation for planetary landing*, Collection of Technical Papers - 2007 AIAA InfoTech at Aerospace Conference.2007.

[19] Delaune, J., Le Besnerais, G., Sanfourche, M., et al., *Optical Terrain Navigation for Pinpoint Landing: Image Scale and Position-Guided Landmark Matching*, Proceedings of the 35th Annual Guidance and Control Conference.

[20] Meng, D., Yun-Feng, C., Qing-xian, W., Zhen, Z., *Image Processing in Optical Guidance for Autonomous Landing of Lunar Probe*, Studies in Computational Intelligence Volume 192, 2009, pp 1-10.

[21] White, M., Criss, T., Adams, D., *APLNav Terrain Relative Navigation Helicopter Field Testing*, AIAA Guidance, Navigation, and Control Conference, 10 - 13 August 2009, Chicago, Illinois.

[22] Meduna, D.K., Rock, S.M., McEwen, R.S., *Closed-loop terrain relative navigation for AUVs with non-inertial grade navigation sensors*, Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES, Monterey, CA.

[23] Sabiron, G., Chavent, P., Burlion, L., et al., *Toward an Autonomous Lunar Landing Based on Low-speed Optic Flow Sensors*, EuroGNC, 2nd CEAS Specialist Conference on Guidance, Navigation and Control, 2013, 993-1011.

[24] Moroni, C., Ansalone, L., Curti, F., *Analysis And Experimentation Of An Optical-flow-based Navigation Algorithm For A Lander*, 2^{nd} IAA Conference on Dynamics and Control of Space Systems, March 24-26 2014, Rome, Italy, Paper IAA-AAS-DyCoSS2-14 -02-06.