E. Andreis, H. Krüger, S. Woicke, P. Panicucci, F. Topputo
Hardware-In-the-loop Validation of Autonomous Interplanetary Navigation Algorithms for Interplanetary Cruises with the Optical Star Stimulator
in: AIAA Scitech 2024 Forum, AIAA, 2024, ISBN: 9781624107115, p. 1-22, AIAA 2024-0776
[AIAA Scitech 2024 Forum, Orlando, FL, USA, 8-12 Jan. 2024]
doi:10.2514/6.2024-0776

Hardware-In-the-Loop Validation of Autonomous Navigation Algorithms for Interplanetary Cruises with the Optical Star Stimulator

Eleonora Andreis*
Politecnico di Milano, Milan, 20156, Italy

Hans Krüger† and Svenja Woicke‡
DLR, German Aerospace Center, 28359, Bremen, Germany

Paolo Panicucci§ and Francesco Topputo¶
Politecnico di Milano, Milan, 20156, Italy

Thanks to the rising focus on deep-space exploration and exploitation, the demand for sustainable and efficient navigation approaches has become crucial. Standard ground-based radiometric tracking, while accurate, is expensive and resource-intensive, posing long-term sustainability challenges. Therefore, enhancing spacecraft autonomy is crucial to avoid ground station saturation. Autonomous onboard guidance, navigation, and control (GNC) offer cost reduction and expand interplanetary exploration opportunities. Among various navigation alternatives, vision-based navigation (VBN) stands out for its cost-effectiveness, ground independence, and applicability to different spacecraft classes. Ground testing campaigns are crucial to ensure accurate and robust vision-based navigation algorithms for interplanetary missions. However, obtaining real interplanetary sky-field images for validation is challenging due to limited successful missions and datasets. To overcome these limitations, high-fidelity rendering engines and hardware-in-the-loop (HIL) simulations are necessary to generate image datasets for testing. This work presents the development of a procedure for on-ground testing and validation of autonomous navigation algorithms for interplanetary cruises using Jena Optronik's Optical Sky Stimulator (OSI) at the DLR GNC Department in Bremen. The proposed paper includes preparation activities, calibration and compensation procedures, and final hardware-in-the-loop simulations.

I. Introduction

As we approach a new era of interplanetary exploration and utilization, it is becoming increasingly important to adopt sustainable and efficient navigation methods. Although traditional ground-based radiometric tracking is accurate and reliable, it is an expensive and resource-intensive paradigm that cannot be sustained in the long term due to the dependence on limited resources such as ground stations and dedicated teams [1]. Therefore, improving the autonomy level of spacecraft is a critical objective for future deep-space missions to prevent ground station saturation. Additionally, autonomous guidance, navigation, and control (GNC) solutions onboard spacecraft can reduce the cost of space entry, thereby expanding deep-space exploration and utilization to new and private players. Among the various navigation alternatives for spacecraft, including autonomous X-ray pulsar-based [2], semi-autonomous radio-based navigation [3], and autonomous vision-based navigation (VBN), the latter is often preferred due to its reduced implementation cost, full ground independence, and applicability to different spacecraft classes [4]. In addition, VBN is an approach compatible with all mission phases toward celestial bodies: cruise [5–8], mid-range [9–12], and

*Ph.D. Student, Politecnico di Milano, Via La Masa, 34, 20156, Milan, Italy, eleonora.andreis@polimi.it, AIAA Member

†Head of Navigation Group of GNC Systems Department, Guidance, Navigation and Control Systems Department, Institute for Space Systems, DLR, German Aerospace Center, 28359, Bremen, Germany, Hans.Krueger@dlr.de

‡GNC Engineer, Guidance, Navigation and Control Systems Department, Institute for Space Systems, DLR, German Aerospace Center, 28359, Bremen, Germany, Svenja.Woicke@dlr.de

§Assistant Professor, Politecnico di Milano, Via La Masa, 34, 20156, Milan, Italy, paolo.panicucci@polimi.it

¶Full Professor, Politecnico di Milano, Via La Masa, 34, 20156, Milan, Italy, francesco.topputo@polimi.it, AIAA Senior Member close proximity [13], including landing [14, 15]. VBN allows probes to determine their state by observing celestial bodies on images taken by optical sensors and has already been adopted by various missions [16]. However, VBN solutions for interplanetary exploration have only undergone onboard testing, beginning with the Deep-Space 1 (DS1) mission in 1998 [5]. Nevertheless, in recent years, there has been growing interest in VBN algorithms for interplanetary exploration, particularly for CubeSat missions [6].

To ensure that the VBN algorithms are both accurate and robust enough to be deployed onboard a spacecraft during an interplanetary mission, it is essential to conduct testing campaigns in a representative environment [17]. These campaigns aim to analyze the performance of the navigation algorithm when real mission conditions and hardware characteristics are introduced in the simulation. However, real mission databases are scarce in number and they are constrained to the operational orbit covered by the spacecraft. This limitation is particularly relevant in the context of far-range navigation since the majority of available images have been acquired for the science phase in proximity to planets, moons, and asteroids. Consequently, these images cannot be effectively employed to feed deep-space VBN filters due to the resolved nature of celestial bodies. Moreover, the few available deep-space images acquired during a cruise rarely contain information about planets or other celestial bodies. These images are typically taken for different purposes, such as on-orbit camera calibration, making them unsuitable for optical navigation. Other optical sensors intensively adopted during a cruise are star trackers. Nevertheless, they do not usually provide the acquired image as output. Hence, it is clear that exploiting real mission images to entirely assess the functionality of VBN algorithms is unfeasible. If high-fidelity rendering engines [18–21] can partially overcome this problem, the errors from external and internal sources are still applied with an approximated model. Therefore, to obtain images with a more realistic error representation, the exploitation of real hardware becomes necessary inside the image generation step. At this aim, optical facilities that include a real optical sensor can be exploited [17, 22–24].

In this framework, the goal of this work is to develop a procedure to test and validate autonomous navigation algorithms for interplanetary cruises through the exploitation of the Optical Sky Stimulator (OSI) by Jena Optronik present at the DLR GNC Department in Bremen[*]. In particular, this paper presents a detailed pipeline that includes preparation activities, optical facility calibration, compensation procedures to take care of the warping introduced by the optical facility, and final hardware-in-the-loop (HIL) simulations to analyze and validate the performances of the optical navigation algorithm.

This document is subdivided as follows: Sec. II defines the preliminary tasks to be performed for hardware selection and simulation setup; Sec. III presents a static analysis of the hardware performances, where some initial considerations related to the chosen hardware are given; Sec. IV describes the calibration procedure adopted to model the distortion introduced by the OSI and the upstream compensation pipeline to correct them; Sec. VI presents the approach adopted to render sky-field images on the OSI screen, and Sec. VII introduces the autonomous VBN algorithm and describes how static and dynamic HIL simulations are performed.

II. Hardware Setup

The optical facility adopted to validate the VBN algorithms is Jena Optronik's OSI. It consists of an optical head (OH), which comprehends a high-resolution micro-display, on which sky-field images are rendered, and a collimating lens to place the observed objects at infinity. Note that the term *optical system* will be adopted to address the camera–lens system, whereas the term *optical facility* is used to address the optical system–OSI assembly.

In this section, the preliminary tasks needed to prepare the simulation framework are defined. In particular, they are subdivided into hardware selection, where the optical system is chosen, and mechanical alignment, where a procedure to align manually the screen and the camera reference frame is described. For the hardware selection, two compatibility criteria have been followed:

1) Ensuring that the lens image circle is greater than the image sensor to avoid mechanical vignetting
2) Ensuring the match between the OSI and the optical system FoVs

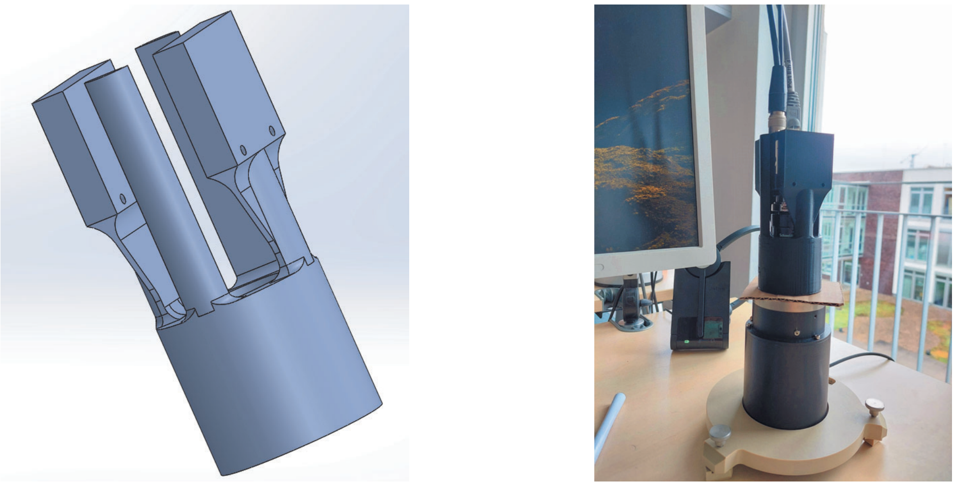
From the datasheet, OSI has a 20° circular FoV. To select the most suited optical system, the angular diagonal FoV ($\mathrm{AFoV_D}$) is evaluated for all the matches camera–lens available at the DLR. The results are reported in Table 1. The $\mathrm{AFoV_D}$ is not evaluated for the pairs camera–lens that do not respect the first compatibility criterion.

[*] `https://www.dlr.de/irs/en/desktopdefault.aspx/tabid-11074/`

Table 1 AFoV$_D$ **for the available matches camera–lens**

	GC1350	GC1380H	GE1900	GT2050NIR
Xenoplan 1.4/17	25.80°	35.20°	-	-
Cinegon 1.9/10	42.96°	57.82°	79.38°	79.38°
Cinegon 1.8/4.8	82.98°	106.34°	-	-
Stilar 2.8/8	51.93°	69.26°	93.38°	93.38°
LM8HC	54.92°	73°	97.82°	97.82°

The combination GC1350–Xenoplan 1.4/17 is selected as its corresponding AFoV$_D$ is the closest to the OSI one. However, a mechanical vignetting will be still present as the optical system AFoV$_D$ is greater than 20°. To prevent this effect, a lens with a longer focal length can be adopted to reduce the AFoV$_D$ of the optical system. In the absence of that, a region of interest (ROI) in the captured image can be defined during the post-processing. Once the optical system is selected, it must be placed on top of the OSI optical head. Yet, the originally available baffle mount has been developed to hold the Jena Optronik's AstroAPS star tracker. Therefore, a new adaptor is modeled to allow the exploitation of the chosen optical system. Fig. 1a shows the CAD model of the adaptor, whereas Fig. 1b shows the complete hardware configuration.

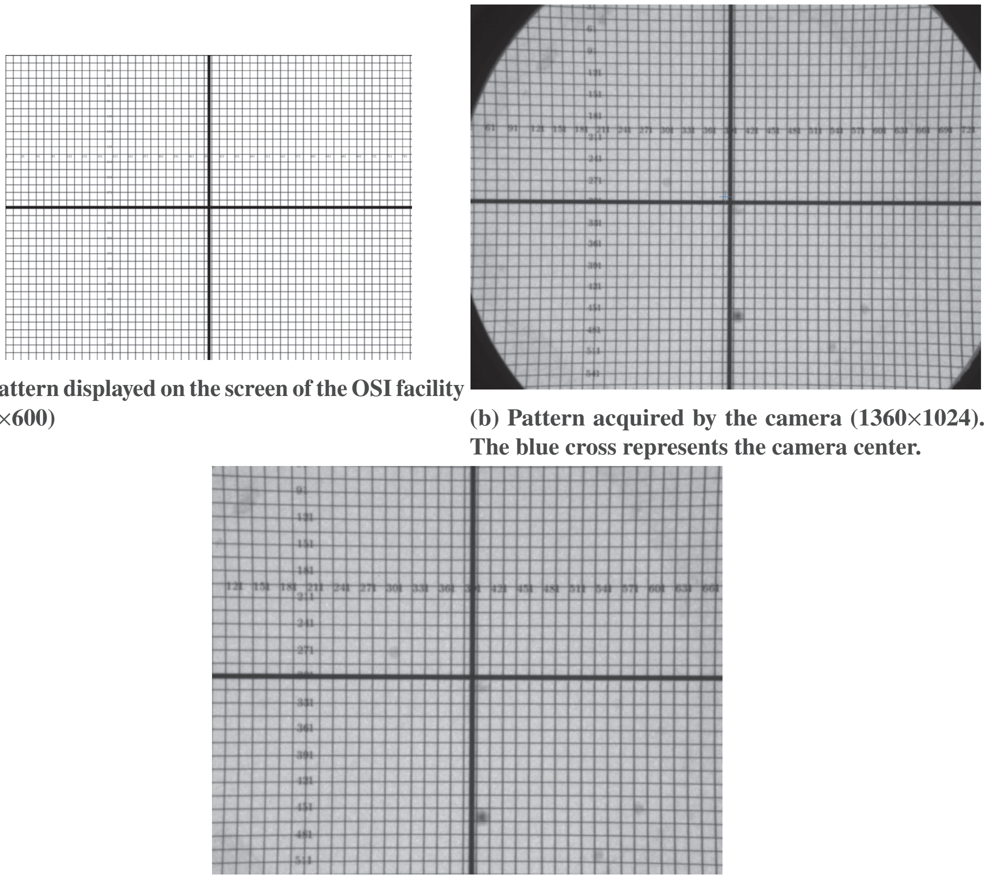


(a) CAD model of the adaptor **(b) Complete configuration**

Fig. 1 Hardware setup

Once the optical facility, i.e., OSI and the optical system, are in place, a manual alignment of the camera and screen reference frames is performed. Fig. 2a is rendered in full screen on the OSI display. The image acquired by the camera is shown in Fig. 2b. The size of the camera image is 1360×1024. The mechanical vignetting present in the image validates the assumptions previously made.

By observing the camera image, a manual alignment is performed to center and align the camera reference frame with the screen reference frame as precisely as possible. The remaining misalignment will be later estimated by the calibration. At this point, a ROI of the acquired image is selected to cut the black borders. The enumeration of the screen pixels along the screen horizontal and vertical directions in the acquired image is adopted to determine correctly which portion of the screen is contained inside the ROI selected. This is of paramount importance for the next steps of the simulation since the image rendered on the screen will be constrained inside this region. In this case, the selected camera ROI is an 1100×880 window, which results in a screen ROI of 580×465. Therefore, the camera FoV is shrunk to 16°× 13°. Fig. 2c shows the selected ROI. Note that the ratio between the resolution of the camera and the screen ROI is 1.89 in both vertical and horizontal directions.



(a) Pattern displayed on the screen of the OSI facility (800×600)

(b) Pattern acquired by the camera (1360×1024). The blue cross represents the camera center.

(c) ROI selected

Fig. 2 Pattern adopted for the mechanical alignment

III. Static Analysis of the Hardware Performances

By analyzing the selected hardware, some preliminary considerations can be made regarding the performance of the simulation. In particular, the OSI optical head consists of a micro-display (800× 600 pixels, $d_{\text{pix}_\text{s}} = 15\mu\text{m}$), collimating optics ($f_{\text{coll}} = 30$ mm), and a control box. From these information, the screen pixel angular dimension (δ_{screen}) as a function of the camera boresight offset (α_{offset}) is found as [17]

$$\delta_{\text{pix}_\text{s}} = \tan^{-1}\left(\tan\alpha_{\text{offset}} + \frac{d_{\text{pix}_\text{s}}}{2f_{\text{coll}}}\right) - \tan^{-1}\left(\tan\alpha_{\text{offset}} - \frac{d_{\text{pix}_\text{s}}}{2f_{\text{coll}}}\right) \tag{1}$$

This value represents the minimum angular displacement between two adjacent features on the screen. If the separation has a lower value, the two feature positions will fall inside the same pixel, and they will approximate to coincide with the pixel center. Indeed, the maximum error resulting in position approximation is $\pm\delta_{\text{pix}_\text{s}}/2$, and it is uniformly distributed throughout the pixel. In other terms, $\delta_{\text{pix}_\text{s}}$ represents the apparent angular size of the screen pixel.

For medium-small FoV (small value of α_{offset}), as in this case, Eq. 1 can be approximated to [25]

$$\delta_{\text{pix}_\text{s}} = 2\tan^{-1}\frac{d_{\text{pix}_\text{s}}}{2f_{\text{coll}}} \approx 103 \text{ arcsec} \tag{2}$$

In this case, at the boresight, $\delta_{\text{pix}_\text{s}}$ is 103 arcsec along the screen pixel horizontal and vertical directions, and about 145 arcsec along the screen pixel diagonal direction. Fig. 3 represents the minimum angular displacement curves as a function of the off-boresight angle. Note that the variation of the values is limited. The apparent angular size of the camera pixel can be computed as well as

$$\delta_{\text{pix}_\text{c}} = 2\tan^{-1}\frac{d_{\text{pix}_\text{c}}}{2f} \approx 54 \text{ arcsec} \tag{3}$$

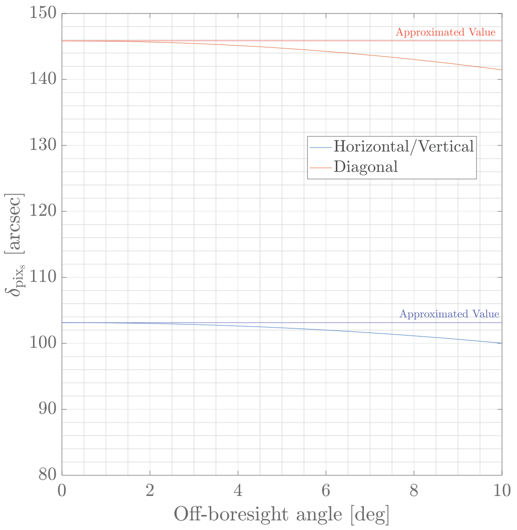


Fig. 3 Minimum angular displacement between two adjacent screen pixels as a function of the off-boresight angle

Note that the ratio between the screen and the camera pixel apparent size is 1.9:1 in both the horizontal and vertical directions, confirming what is found in the mechanical alignment procedure in Sec. II. This means that 1 pixel of the screen corresponds to 4 pixels of the camera, i.e. 2 along X and 2 along Y. On one side, the obtained ratio is not ideal for representing a continuous world with a screen and a camera, which would require instead a 1:4 ratio [17]. Yet, on the other side, the defocus obtained at the camera level mimics the defocusing procedure adopted by modern star trackers to get sub-pixel accuracy. Thus, this hardware setup can still represent a valid testing environment.

IV. Calibration Procedure

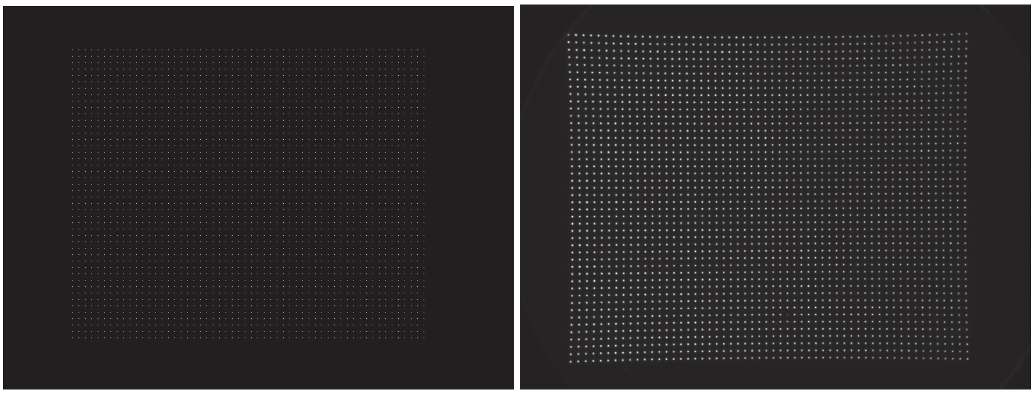
Before exploiting the OSI facility for HIL simulations, it is necessary to compensate for the effects introduced by the optical facility that would not be present in orbit. The alignment between the OSI and the camera reference frame is manually performed with the procedure explained in Sec. II. Thus, a residual misalignment will be always present. In addition, the light emitted by the OSI miniaturized display is warped by the collimator and camera lenses. Therefore, the image captured by the camera is distorted with respect to the one rendered on the display, and it is not more representative of the real sky. It is of paramount importance to elaborate a procedure to evaluate the distortion model that represents the effect of the optical facility on the screen points, and, eventually, to compensate for this effect. The first step of the procedure is defined calibration, whereas, the latter is referred to as compensation [17]. The optical system is calibrated by comparing the images rendered on the OSI screen with the ones delivered by the camera. In this work, the camera is not calibrated independently but this should be object to future investigations. This last step would decouple the optical errors due to the camera lens from the optical and mechanical errors due to the OSI optical head and its misalignment with the camera. Being the camera independently calibrated, the following correction procedure would not compensate for the camera optical errors. This strategy allows a more realistic representation of the mission scenario since the camera optical effects, which are due to the camera lens, would also appear on real sky images acquired by the same optical sensor during a deep-space cruise.

The calibration procedure applied in this work consists of the following:

- Rendering multiple grid patterns on the OSI screen
- Acquiring the rendered patterns with the camera
- Evaluating the centroids of the camera and screen images
- Determining the calibration coefficients

A. Grid Patterns Rendering and Acquisition

Once the screen ROI is obtained (Sec. II), images containing highlighted pixels are rendered on the screen and acquired by the camera. At this aim, a set of grid patterns is generated by shifting the positions of the dots rigidly in the horizontal and vertical directions. It is important to cover evenly the desired screen region to allow the calibration to capture the distortion most accurately. Particular attention has to be paid to the corner points whose positions are the ones most subjected to deformation. Figs. 4a and 4b represent one of the grids displayed on the OSI screen and acquired by the camera, respectively. As has been previously mentioned, the bright dots in the acquired images are bigger in angular size and they appear as defocused with respect to their screen counterpart. Moreover, the pincushion distortion introduced by the OSI collimating optics is well visible.



(a) Grid displayed on the OSI screen **(b) Grid acquired by the camera**

Fig. 4 Grid patterns

B. Centroids Computation

Once the images are acquired, the bright dots centroids are extracted from the camera and the screen images. First, Niblack's thresholding method [26] is adopted to remove the background noise to connected portions of the image delimited by squared windows with a margin of one pixel on each side. Hence, the centroid of the object is computed by applying an intensity-weighted center of gravity algorithm considering the pixels inside the associated squared windows [27]. An additional check is performed inside the procedure concerning the camera centroids evaluation. If a window has a dimension of only one pixel, the bright dot inside is rejected. This is done to avoid acquiring bright pixels due to camera noise.

C. Calibration Coefficients Determination

To represent the deformation introduced by the optical facility, the direct ($\mathcal{D}$) and inverse ($\mathcal{D}^{-1}$) distortion models shall be defined. The direct distortion model describes the warping applied by the optical assembly to the screen points. Whereas, the inverse model defines the deformation that is necessary to apply to compensate for the effect introduced by the optical assembly. The distortion flow is shown in Fig. 5. Instead, a representation of the warping applied from the direct and inverse deformation to the screen points is shown in Figs. 6a and 6b, respectively. The inverse distortion model applies to the screen points a barrel deformation, which is opposite to the pincushion one introduced by the hardware.

To compensate for the effect introduced by the optical assembly, the inverse distortion model is computed. In this work, this latter is represented by two polynomial equations of the 7th order that map how the angular coordinates of the i-th camera point (α_i, β_i) are warped into its associated pixel coordinates (x_i, y_i) in the 2D screen reference frame $\mathbb{S}$ [22]:

$$ {}^{\mathbb{S}}x_i = \sum_{k=0}^{7}\sum_{j=0}^{7} a_{k,j}\alpha_i^k\beta_i^j \qquad {}^{\mathbb{S}}y_i = \sum_{k=0}^{7}\sum_{j=0}^{7} b_{k,j}\alpha_i^k\ \beta_i^j \tag{4} $$

where $a_{k,j}$ and $b_{k,j}$ are the unknown coefficients of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of the two polynomial equations. To retrieve them, the centroids of the N grid points extracted from the camera and screen images are exploited in

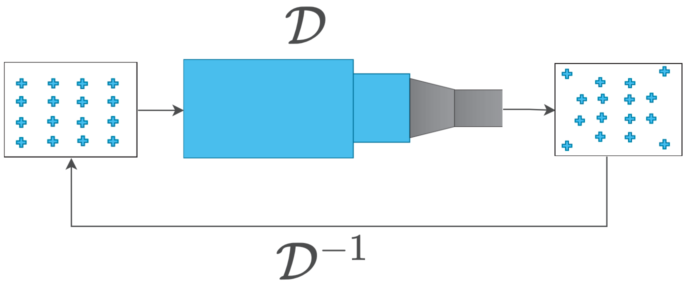
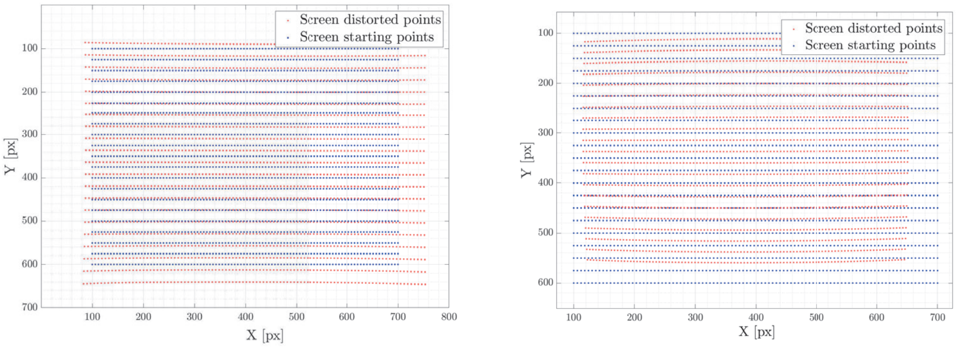


Fig. 5 Distortion flow



(a) Direct distortion effect

(b) Inverse distortion effect

Fig. 6 Graphical representation of the distortion effects

the least square method such as:

$$\boldsymbol{a} = (\boldsymbol{M}^\top \boldsymbol{M})^{-1}\boldsymbol{M}^\top\, {}^{\mathbb{S}}\boldsymbol{x}_{\text{sc}} \qquad \boldsymbol{b} = (\boldsymbol{M}^\top \boldsymbol{M})^{-1}\boldsymbol{M}^\top\, {}^{\mathbb{S}}\boldsymbol{y}_{\text{sc}} \tag{5}$$

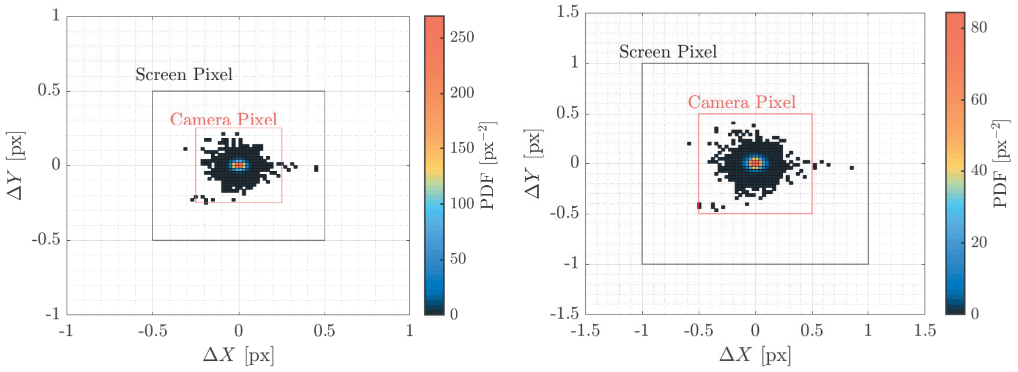
where ${}^{\mathbb{S}}\boldsymbol{x}_{\text{sc}}, {}^{\mathbb{S}}\boldsymbol{y}_{\text{sc}}$ are the vectors contained the x and y coordinates of the N screen grid points and $\boldsymbol{M}$ is a matrix $N \times 64$ defined exploiting the angular coordinates of the N camera grid points as:

$$\boldsymbol{M} = \begin{bmatrix} \dots \\ \sum_{k=0}^{7}\sum_{j=0}^{7} \alpha_i^k \beta_i^j \\ \dots \end{bmatrix} \qquad \text{for i= 1, ..., N} \tag{6}$$

Once the coefficient vectors of the inverse distortion model are found, they can be exploited to compensate for the distortion introduced by the optical facility. The pixel coordinates of the compensated camera points in the 2D screen reference frame are evaluated as

$$ {}^{\mathbb{S}}\hat{\boldsymbol{x}}_{\text{cam}} = \boldsymbol{a}^\top \boldsymbol{M}^\top \qquad {}^{\mathbb{S}}\hat{\boldsymbol{y}}_{\text{cam}} = \boldsymbol{b}^\top \boldsymbol{M}^\top \tag{7}$$

The errors between the position of the screen grid points and the compensated camera grid points in the 2D screen and camera reference frame are represented in Fig. 7.



(a) Screen reference frame

(b) Camera reference frame

Fig. 7 Error distribution of the grid points position obtained from the inverse calibration procedure

V. Compensation Procedure

Once the distortion model is defined, the compensation procedure, whose goal is to compensate for the warping effect introduced by the optical assembly, is started. The compensation can be applied at the screen level, thus, before the image is acquired by the camera (upstream compensation), or at the camera level, thus, after the image is taken (downstream compensation). The upstream compensation is usually preferred over the downstream since the latter requires performing an additional step during the image post-processing which would not be present in an algorithm applied to an actual mission scenario. Instead, the upstream compensation does not require a modification of the image processing procedure of the autonomous navigation algorithm, allowing true test-as-you-fly verification of the optical sensor. The downstream compensation flow is shown in Fig. 8. At first, a different set of grid images is rendered on the OSI screen and acquired by the camera. Afterward, the centroids of the camera bright points are exploited to evaluate the matrix $\boldsymbol{M}$ of Eq. 6, and, eventually, the compensated camera points in the 2D screen camera frame are found by applying Eq. 7. The error distribution between the nominal screen points and the resulting camera points is shown in Figs. 9a and 9b in the 2D screen and camera reference frame, respectively. The upstream compensation flow is graphically represented in Fig. 10. Also in this case a new set of grids is projected on the OSI screen, acquired by the camera, and the associated centroids are extracted from the camera and screen images. Then, the compensation is performed by applying Eq. 7 after having computed the matrix $\boldsymbol{M}$ for the screen grid points. The resulting compensated screen points $({}^{\mathbb{S}}\hat{\boldsymbol{x}}_{\text{scr}}, {}^{\mathbb{S}}\hat{\boldsymbol{y}}_{\text{scr}})$ are then approximated with the closest pixel center. Fig. 11 represents the error distribution between the starting, non-compensated, screen points and the ones extracted from the camera images in the 2D screen and camera reference frame, respectively. The error is uniformly distributed inside the screen pixel with a variation of ±53 arcsec along both directions, which verifies the preliminary assumptions made in Sec. III.

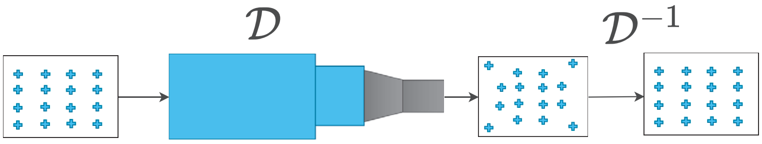
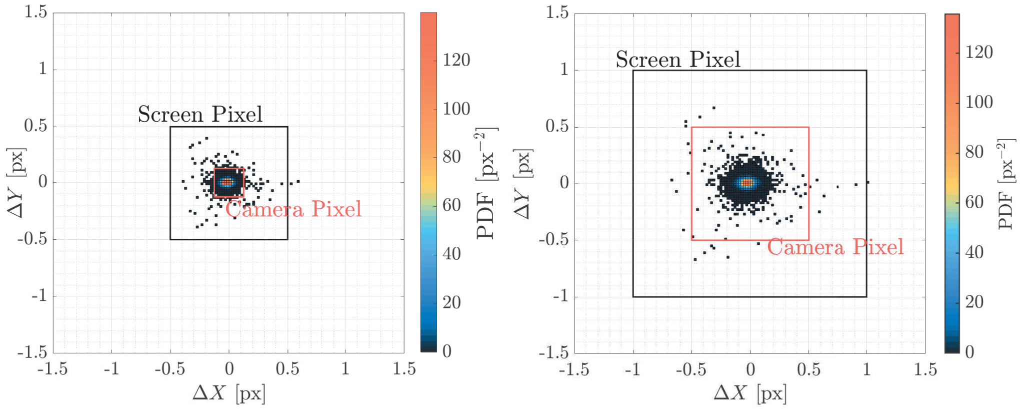


Fig. 8 Downstream compensation flow

To lower the centroiding error and reach sub-pixel accuracy, the light emitted by one pixel of the screen can be spread over more pixels. This approach consists of switching other three pixels near the original one by following the scheme reported in Fig. 12a with $(\Delta X, \Delta Y)$ the distance between the compensated screen centroid and its closest pixel center, i.e., $(X_* - X_c, Y_* - Y_c)$ labeled in Fig. 12b.

The digital count value associated with each one of the four pixels is proportional to the portion of the area of the square with edge $r = 0.5$ and center in the compensated screen centroid (see Fig. 12b) that is



(a) Screen reference frame

(b) Camera reference frame

Fig. 9 Error distribution of the grid points position during the downstream compensation

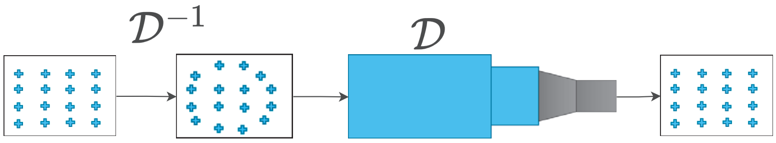
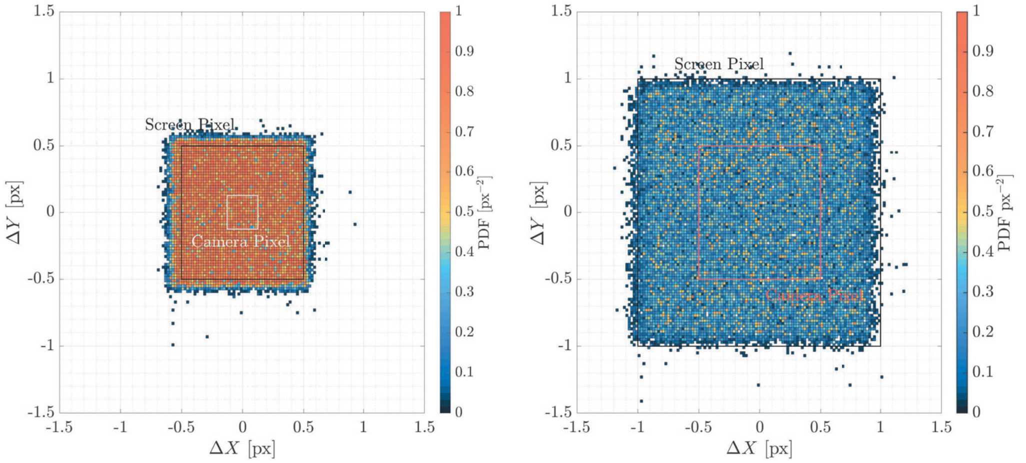


Fig. 10 Upstream compensation flow



(a) Screen reference frame

(b) Camera reference frame

Fig. 11 Error distribution of the grid points position during the upstream compensation

contained inside the pixel itself, such as:

$$DC_i = 255\frac{A_i}{r^2} \tag{8}$$

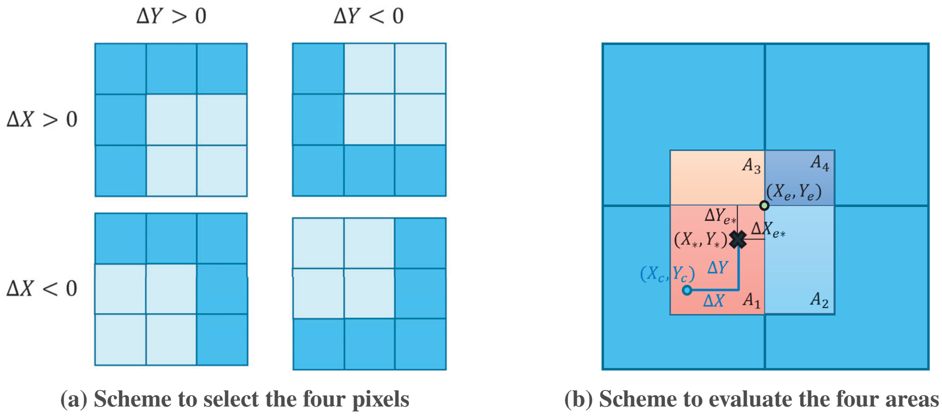
By referring to Fig. 12b, the four portions of the square area are evaluated as:

$$A_1 = (r + \Delta X_{e*})(r + \Delta Y_{e*}) \tag{9}$$

$$A_2 = (r - \Delta X_{e*})(r + \Delta Y_{e*}) \tag{10}$$

$$A_3 = (r + \Delta X_{e*})(r - \Delta Y_{e*}) \tag{11}$$

$$A_4 = (r - \Delta X_{e*})(r - \Delta Y_{e*}) \tag{12}$$



(a) Scheme to select the four pixels **(b) Scheme to evaluate the four areas**

Fig. 12 Definition of the procedure to get sub-pixel accuracy

where

$$\Delta X_{e*} = ||X_e - X_*|| \qquad \Delta Y_{e*} = ||Y_e - Y_*|| \tag{13}$$

and the point (X_e, Y_e) is defined following the logic in Table 2.

	$\Delta Y > 0$	$\Delta Y < 0$
$\Delta X > 0$	$X_e = X_c + 0.5$ $Y_e = Y_c + 0.5$	$X_e = X_c + 0.5$ $Y_e = Y_c - 0.5$
$\Delta X < 0$	$X_e = X_c - 0.5$ $Y_e = Y_c + 0.5$	$X_e = X_c - 0.5$ $Y_e = Y_c - 0.5$

Table 2 Definition of the point (X_e, Y_e)

Eventually, the effect on the 2D error distribution resulting from the application of the point spread procedure is shown in Fig. 13. Note that the error now is almost confined inside the camera pixel.

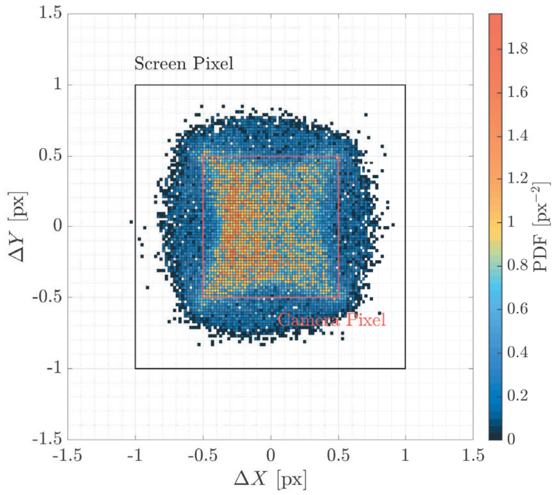


Fig. 13 Error distribution of the grid points position in the 2D camera frame after the application of the sub-pixel procedure

VI. Sky-field Rendering Engine

To validate the autonomous optical navigation algorithm through HIL simulations, deep-space images need to be rendered on the OSI screen. At this aim, high-fidelity rendering engines can be exploited to generate sky-field images representative of the analyzed mission scenario. In this work, an extended version of the deep-space rendering engine described in [18] is adopted. In general, the simulator can operate in two modes: software-simulation mode and hardware-in-the-loop-simulation mode. In the former, the rendering engine incorporates the camera model, which considers factors like the camera Point Spread Function, the conversion from photons to electrons, the sensor response, and internal error sources, such as shot noise, amplifier noise, reset noise, pattern noise, and discretization noise. The rendering engine can also simulate streaks caused by celestial objects in the image due to rapid spacecraft slew, as well as external noise sources like light time, light aberration, and the impact of cosmic rays. The output of the rendering engine in software-simulation mode is an image as it would be taken by the modeled sensor in deep space and whose intensity is expressed in pixel digital count according to the camera bit depth.

On the contrary, the second mode foresees the rendering of deep-space scenes to project them on the screen of an optical facility. To be representative of a real deep-sky scene taken with the optical sensor employed in the HIL simulation, the image projected on the optical facility screen must have the resolution of the screen and the FoV of the camera. Moreover, the rendered image is warped through the application of the upstream compensation by taking into account the calibration coefficients obtained during the calibration of the optical assembly. Additionally, the procedure detailed in Sec. V can be applied to obtain sub-pixel accuracy during the centroids extraction step. For what concerns the noises, only the effects created by the external environment are modeled since internal errors are already embedded in the simulation due to the physical presence of the camera. The output of this mode is a raw image matrix to be projected on the optical facility screen whose intensity is representative of the illuminance of the displayed stars and celestial bodies. Note that the bright objects illuminance is evaluated through the following similarity equation by knowing that the illuminance value of a star with $m = -14.2$ is 1 lm/m^2 [†]:

$$E = 10^{(-(m+14.2)/2.5)} \tag{14}$$

where m is the absolute magnitude of the bright object [23].

A. Relation between pixel illuminance and pixel digital count

To have a perfect matching between the real star radiometric signal and the signal emitted by the screen pixel in which the star is projected, an accurate radiometric calibration of the facility would be needed. Indeed, it is not possible to predict in advance at the rendering level which is the digital count of each screen pixel since these values depend on the screen radiometric setting. To radiometrically calibrate the facility, a powermeter needs to be placed in correspondence with the camera entrance pupil to measure the irradiance received by the camera when pixels with different digital count values are switched on [28]. In the absence of a powermeter, the procedure described in [23] can be adopted. This consists of a pipeline that can be applied to evaluate the pixel digital counts by computing the maximum representable illuminance level of the images projected on the screen, over which the pixel is assumed to be saturated. To define this value, in the absence of technical specification, as the maximum luminance of the screen, the following approach is employed. Firstly, a deep-sky image is generated by the high-fidelity rendering engine in the software mode considering the optical sensor characteristics. The image obtained is considered as the true representation of reality. Therefore, it is adopted to tune the maximum representable illuminance value of the images to be projected on the screen. In particular, the tuning is performed by comparing the true image and the image output of the optical assembly obtained by varying the maximum representable illuminance value until a correspondence is found. At this point, the pixels with an illuminance greater or equal to the maximum representable illuminance level found are switched on with the maximum digital count. Instead, the other illuminance levels are linked to a digital count value linearly.

In this work, the maximum representable illuminance value selected is the one associated with an absolute magnitude equal to 5.5. The validity of the selection is confirmed by the fact that usually, a miniaturized camera has a magnitude threshold limit of around 7/8.

[†] https://stjarnhimlen.se/comp/radfaq.html#7, Last access on October 16, 2023.

VII. Validation of the Autonomous Navigation Algorithm through HIL Simulations

A. Autonomous Vision-Based Navigation Algorithm for Interplanetary Cruises

During an interplanetary cruise, the probe can determine its position by observing the movements of celestial bodies such as planets and asteroids, which appear as unresolved bright dots in images taken by optical sensors. Since the celestial bodies light falls within a single pixel of the image, the only exploitable information to estimate autonomously the probe state onboard are the LoS directions. If the LoS directions of two or more reference beacons are simultaneously available, they can be adopted to statically triangulate the probe position [8, 29]. However, if only one reference beacon is visible, a dynamic estimation method is required. In particular, this work investigates the validation of an autonomous VBN algorithm applied to CubeSats, which brings the enforcement of some constraints that make the navigation problem even more challenging than for standard probes. In general, CubeSats differ from traditional spacecraft due to their lower production costs, achieved also by utilizing commercial-off-the-shelf components, and their smaller size. Consequently, the optical sensor, such as a star tracker or camera, on the CubeSat must be compact and cost-effective, resulting in limited light detection capabilities. Typically, a miniaturized optical sensor has a threshold magnitude of 7, allowing only brighter celestial bodies like planets to be exploited for navigation, while asteroids are too faint to be observed [30]. Given that observations are limited to the positions of the planets and that these latter are greatly sparse in space, it is unlikely to have more than one visible celestial body in an image. Therefore, the probe to localize itself has to track the first planet for a certain amount of time, and then perform a slew maneuver to point to the successive planet to observe. Thus, to determine the CubeSat state, a dynamical estimator is needed. In this work, a non-dimensionalized EKF fed by asynchronous planet measurements is selected for the development of the optical autonomous navigation strategy thanks to its better numerical stability and computational performance analyzed in [7]. Furthermore, another crucial aspect to take into account in deep-space navigation consists of the shift induced by light effects, such as light-time and light aberration, on the position of celestial bodies in the images [7, 31]. When estimating the probe state, it is important to correct the displacements caused by these factors to avoid biases. The main aspects of the VBN algorithm tested through HIL simulations are detailed in the following paragraphs.

1. Dynamics and Measurement Models

The adopted dynamics model of the algorithm is an extension of the one detailed in [7]. The state vector $\boldsymbol{x}$ is defined as

$$\boldsymbol{x}(t) = [\boldsymbol{r}(t), \boldsymbol{v}(t), \boldsymbol{\eta}(t)]^\top \tag{15}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ are the inertial probe position and velocity, respectively, and $\boldsymbol{\eta}$ is a vector of Gauss–Markow (GM) processes accounting for unmodeled terms: a 3-dimensional residual accelerations $\boldsymbol{\eta}_{\text{R}}$ and the stochastic component of the Solar Radiation Pressure (SRP) $\boldsymbol{\eta}_{\text{SRP}}$; that is, $\boldsymbol{\eta} = [\boldsymbol{\eta}_{\text{R}}, \boldsymbol{\eta}_{\text{SRP}}]^\top$. The process is modeled using the following equation of motion

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t) + \boldsymbol{w} \tag{16}$$

where $\boldsymbol{f}$ is the vector field embedding the deterministic part, while $\boldsymbol{w}$ is the process white noise:

$$\dot{\boldsymbol{x}}(t) = \underbrace{\begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{a}_{\text{Sun}} + \boldsymbol{a}_{\text{SRP}} + \boldsymbol{a}_{\text{pl}_\text{i}} \\ -\xi \boldsymbol{\eta}_{\text{R}} \\ -\xi \boldsymbol{\eta}_{\text{SRP}} \end{bmatrix}}_{\boldsymbol{f}} + \underbrace{\begin{bmatrix} \boldsymbol{0}_{3x1} \\ \boldsymbol{\eta}_{\text{R}} + \boldsymbol{\eta}_{\text{SRP}} \\ \boldsymbol{w}_{\text{R}} \\ \boldsymbol{w}_{\text{SRP}} \end{bmatrix}}_{\boldsymbol{w}} \tag{17}$$

and

$$\boldsymbol{a}_{\text{Sun}} = -\mu_{\text{Sun}} \frac{\boldsymbol{r}}{r^3} \tag{18}$$

$$\boldsymbol{a}_{\text{SRP}} = C_{\text{R}} \frac{P_0 R_0^2}{c} \frac{A_{\text{s}}}{m_{\text{s}}} \frac{\boldsymbol{r}}{r^3} \tag{19}$$

$$\boldsymbol{a}_{\text{pl}_\text{i}} = \mu_i \Big(\frac{\boldsymbol{r}_{\text{pl}_\text{i}} - \boldsymbol{r}}{||\boldsymbol{r}_{\text{pl}_\text{i}} - \boldsymbol{r}||^3} - \frac{\boldsymbol{r}_{\text{pl}_\text{i}}}{||\boldsymbol{r}_{\text{pl}_\text{i}}||^3} \Big) \tag{20}$$

The terms that describe the SRP are [32]: C_R the coefficient of reflection, P_0 the solar power, R_0 the Sun radius, A_s the cross-section area of the probe, and m_s its mass. The third-body perturbation of the Earth-Moon barycenter, Mars, and Jupiter is included. In the Langevin equations that govern the GM processes the coefficient ξ defines the reciprocal of the correlation time, while $\boldsymbol{w}_\mathrm{R}$ and $\boldsymbol{w}_\mathrm{SRP}$ are the process noises of the GM parameters with σ_R and σ_SRP standard deviations, respectively [33]. The process noise covariance matrix is $\boldsymbol{Q}$:

$$\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{0}_{3x3}, \boldsymbol{Q}_a, \boldsymbol{Q}_\mathrm{R}, \boldsymbol{Q}_\mathrm{SRP}) \tag{21}$$

with $\boldsymbol{Q}_\mathrm{R} = \sigma_\mathrm{R}^2 \boldsymbol{I}_{3x3}$, $\boldsymbol{Q}_\mathrm{SRP} = \sigma_\mathrm{SRP}^2 \boldsymbol{I}_{3x3}$, and $\boldsymbol{Q}_\mathrm{a} = (\boldsymbol{Q}_\mathrm{R} + \boldsymbol{Q}_\mathrm{SRP})/(2\xi)$.

Instead, the measurement model selected for the VBN algorithm gives a representation of the planet position projection affected by the light-time and light-aberration effects in pixel coordinates in the 2D camera reference frame [7]:

$$^{\mathbb{C}}_{h}\boldsymbol{r}_\mathrm{pl} = \boldsymbol{h}(\boldsymbol{x}_k) = \boldsymbol{K}_\mathrm{cam}\,\boldsymbol{A}\,\boldsymbol{l}^\mathrm{aberr}_\mathrm{pl/sc} \tag{22}$$

where

$$\boldsymbol{l}^\mathrm{aberr}_\mathrm{pl/sc} = \boldsymbol{l}_\mathrm{pl/sc} + \boldsymbol{l}_\mathrm{pl/sc} \times \left(\boldsymbol{\beta}_\mathrm{sc} \times \boldsymbol{l}_\mathrm{pl/sc}\right) \tag{23}$$

with $\boldsymbol{\beta}_\mathrm{sc} = \dfrac{\boldsymbol{v}}{c}$, and $\boldsymbol{l}_\mathrm{pl/sc}$ defined as

$$\boldsymbol{l}_\mathrm{pl/sc} = \frac{\left(\boldsymbol{r}_\mathrm{pl}(t-\Delta t) - \boldsymbol{r}(t)\right)^\top \left(\boldsymbol{r}_\mathrm{pl}(t-\Delta t) - \boldsymbol{r}(t)\right)}{\left\|\left(\boldsymbol{r}_\mathrm{pl}(t-\Delta t) - \boldsymbol{r}(t)\right)^\top \left(\boldsymbol{r}_\mathrm{pl}(t-\Delta t) - \boldsymbol{r}(t)\right)\right\|} \tag{24}$$

is equal to the planet LoS direction from the spacecraft position at time t, i.e., the epoch at which the light is received by the spacecraft, to the planet position at time τ, i.e., the epoch at which the light is emitted by the planet, where $\Delta t = t - \tau$. Eventually, the measurement error covariance matrix is:

$$\boldsymbol{R} = \begin{bmatrix} \sigma^2_{\mathrm{err}_\mathrm{X}} & 0 \\ 0 & \sigma^2_{\mathrm{err}_\mathrm{Y}} \end{bmatrix} \tag{25}$$

where $\sigma_{\mathrm{err}_\mathrm{i}}$ is the 1-σ standard deviation of the measurement error along the i-th directions of the 2D camera reference frame $\mathbb{C}$.

2. Image Processing Pipeline for the Measurement Acquisition

The observable used to correct the probe state estimation in the navigation filter is the projection of the planet position $^{\mathbb{C}}\boldsymbol{r}_\mathrm{pl}$ onto $\mathbb{C}$. To extract this information from the deep-space image, an image processing (IP) algorithm specifically designed for interplanetary cruises is employed [34, 35]. The main objective of this procedure is to identify the centroid in the image that corresponds to the planet projection and to obtain information about the planet location in the image. The IP pipeline follows a three-step process: 1) Determining the probe attitude, 2) correcting the light aberration of the stars centroids, and 3) identifying the planets in the image. The first step is performed differently if the planet is acquired for the first time or not. In the former situation, a lost-in-space (LIS) algorithm is applied to determine the spacecraft attitude without any previous knowledge of the probe pose. In particular, the pyramid star identification algorithm [36] bulked up by a RANSAC procedure to reject outliers is exploited [37]. When the planet is not observed for the first time, a recursive attitude determination method can be applied to recover the probe orientation since the information about the estimated attitude at the previous step is available. A RANSAC algorithm is always applied at the end of the procedure to reject possible false matches. When the recursive attitude determination fails, the spacecraft orientation at the following image acquisition will be determined again with the LIS method. Vice versa, when the LIS algorithm succeeds in determining the probe orientation, the recursive attitude determination algorithm will be adopted in the following image acquisition.

The second step of the IP consists of the correction of the shift caused by the light aberration on the stars centroids. The procedure applied to correct the effect applies to planets in [31]. Eventually, the probe orientation is recomputed with the corrected stars centroids.

The final step is the identification of the planets in the image and the extraction of their position projections, which will be used to correct the state estimation by the filter. The recognition is performed through the evaluation of the statistical momenta associated with the planet position projection, which defines the Gaussian probability to find the planet in that portion of the image, i.e., its expected position and the associated uncertainty ellipse. The no-stellar object contained in the 3σ ellipse is identified as the planet position projection ${}^{\mathbb{C}}\mathbf{r}_{\text{pl}}$. If multiple no-stellar objects are located within this ellipse, the object closest to the expected planet position is considered to be the planet itself, as it is most likely the true projected position of the planet.

If the IP algorithm fails, two procedures are implemented depending on the failure type: 1) when the attitude is not determined or the planet measurement is not extracted from the image, the state vector and its error covariance matrix are simply propagated until the next step; 2) when the planet is wrongly determined, a course outlier detection method is applied to reject false positives [38].

3. Filtering Scheme

The filtering scheme adopted by the non-dimensionalized EKF is described in Table 3.

Table 3 Filtering Strategy

System State Space	$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), t) + \boldsymbol{w}$ $\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{\nu}_k$ $\dot{\boldsymbol{P}} = \boldsymbol{F}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F}^\top + \boldsymbol{Q}$	
Propagation Block	$\boldsymbol{x}_{p_k} = \boldsymbol{x}_{c_{k-1}} + \int_{t_{k-1}}^{t_k} \boldsymbol{f}(\boldsymbol{x}(t), t)\mathrm{d}t$ $\boldsymbol{P}_{p_k} = \boldsymbol{P}_{c_{k-1}} + \int_{t_{k-1}}^{t_k} \dot{\boldsymbol{P}}\mathrm{d}t$	$\boldsymbol{x}_{c_0} = E[\boldsymbol{x}_0]$ $\boldsymbol{P}_{c_0} = E[\boldsymbol{x}_0\boldsymbol{x}_0^\top]$
Correction Block	$\boldsymbol{K}_k = \boldsymbol{P}_{p_k}\boldsymbol{H}_k^\top(\boldsymbol{H}_k\boldsymbol{P}_{p_k}\boldsymbol{H}_k^\top + \boldsymbol{R}_k)^{-1}$ $\boldsymbol{x}_{c_k} = \boldsymbol{x}_{p_k} + \boldsymbol{K}_k[\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_{p_k})]$ $\boldsymbol{P}_{c_k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{p_k}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^\top + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^\top$	

Here, $\boldsymbol{x}_{p_k}$ is the predicted state vector with error covariance matrix $\boldsymbol{P}_{p_k}$ at epoch k, $\boldsymbol{K}_k$ the Kalman gain, $\boldsymbol{x}_{c_k}$ the corrected state vector with error covariance matrix $\boldsymbol{P}_{c_k}$, $\boldsymbol{F}$ the Jacobian of the dynamics model equation, $\boldsymbol{h}$ the measurement model equation with Jacobian $\boldsymbol{H}_k$, $\boldsymbol{\nu}_k$ the measurement white noise, and $\boldsymbol{y}_k$ the external measurement vector.

B. Static Hardware-In-the-Loop Simulations

Firstly, static simulations are executed to analyze the performance of the IP algorithm, which is the step most challenging in the HIL simulation. A Monte Carlo run is performed for 1000 images generated with random spacecraft pose values to test the IP performance. The IP algorithm has a success rate in attitude determination greater than 98.5% when the LIS procedure is employed, which is coherent with the results reported in the state-of-the-art [36]. Furthermore, this result shows that the calibration procedure has been properly performed since the errors in the centroids position do not substantially worsen the performance of the attitude determination step. In this simulation, the attitude is not determined for less than 1% of the cases, and it is wrongly determined for 0.5%. When an attitude value is found, a total of 1293 planets are detected in the images. Among these, 5 are false positives ($< 0.5\%$). This happens always when the planet is close to the image border or it is too faint to be detected. No false negatives are found. Figure 14 shows the Probability Density Function (PDF) of the errors obtained during the evaluation of the planet centroids. The errors are below the screen pixel size, validating the procedure adopted to get sub-pixel accuracy. In addition, Fig. 15 shows the histograms of the attitude errors along the pointing axis and the orthogonal axes to the pointing direction, respectively.

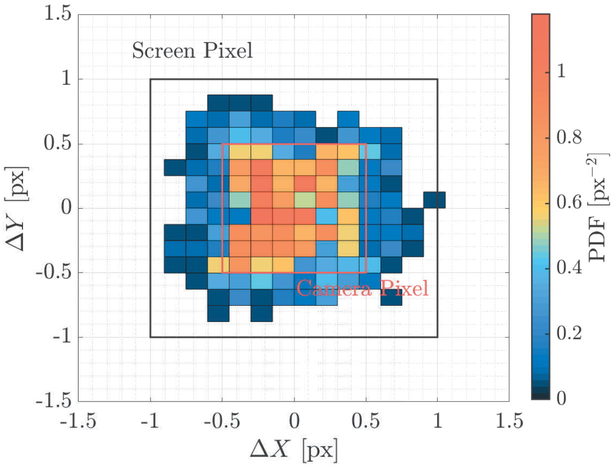


Fig. 14 Planet centroid error

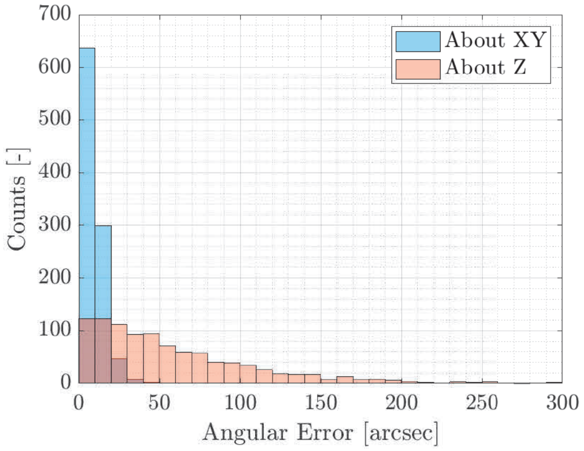


Fig. 15 Attitude determination error

To understand how much the calibration error affects the attitude determination performance, the same Monte Carlo campaign is performed again on a new set of images generated with the rendering engine in software mode by considering the same probe poses and camera characteristics. Therefore, no calibration error is present since no optical assembly is involved in the image generation. In 99.5% of the cases, the attitude is determined correctly, in the 0.4% the attitude is not determined, and in the 0.1% the attitude is wrongly determined (error $> 1°$). Over the 1313 planets found when the attitude is determined, 2 are false positive (0.15%). As could be foreseen, the success rate in attitude determination is higher when no calibration errors that shift the position of the stars centroids are introduced. Nevertheless, it is important to underline that even the performances obtained with the hardware-in-the-loop are comparable to the ones evidenced by the state of the art.

Eventually, the static simulations are repeated to test the performance of the recursive attitude determination algorithm. The success of the algorithm rises to 100% when an attitude value perturbed of 20 arcsecs with respect to the nominal one is given in input to the IP algorithm.

C. Dynamic Hardware-In-the-Loop Simulations

Once the performance of the IP pipeline is verified, the navigation algorithm is tested on a high-fidelity interplanetary trajectory leg between Earth–Mars. At the beginning of each navigation leg, the optimal selection approach detailed in [39] is adopted to determine which couple of planets is best to track to obtain the highest filter performances. Starting from the initial time t_0, the spacecraft tracks the first planet of the optimal pair for 60 minutes. Then, it performs a slew maneuver of 30 minutes to point to the second planet to be observed. Eventually, the state is propagated for 10 days, where no external observations are acquired. This navigation cycle graphically shown in Fig. 16 is repeated until the end of the analyzed trajectory leg.

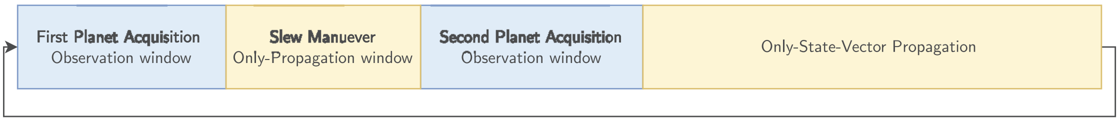


Fig. 16 Navigation concept of operations

Whenever a planet is observed, a deep-space image corresponding to the mission scenario is generated, rendered on the OSI screen, and acquired by the camera. Fig. 17 shows the workflow followed at each time step of the simulation when a planet is observed. The state vector is propagated from the epoch t_k to t_{k+1}. Then, the estimation is corrected with the external observation, i.e., the planet position projection ${}^{\mathbb{C}}\boldsymbol{r}_{\text{pl}}$, extracted from the image captured by the camera placed on the top of OSI.

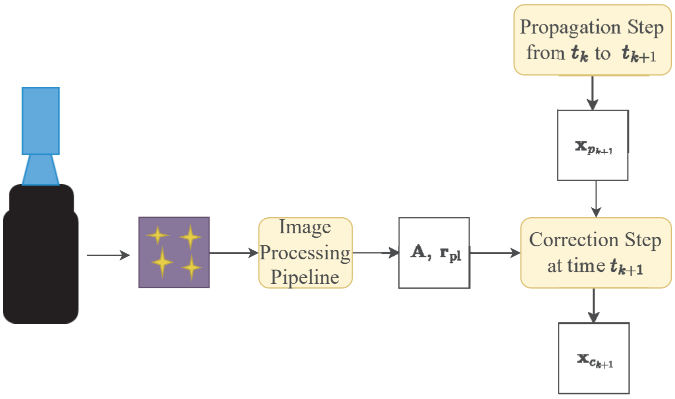


Fig. 17 HIL simulation workflow

The filter performances are studied through a Monte Carlo analysis conducted on 100 samples. The initial uncertainties of the state are reported in Table 4.

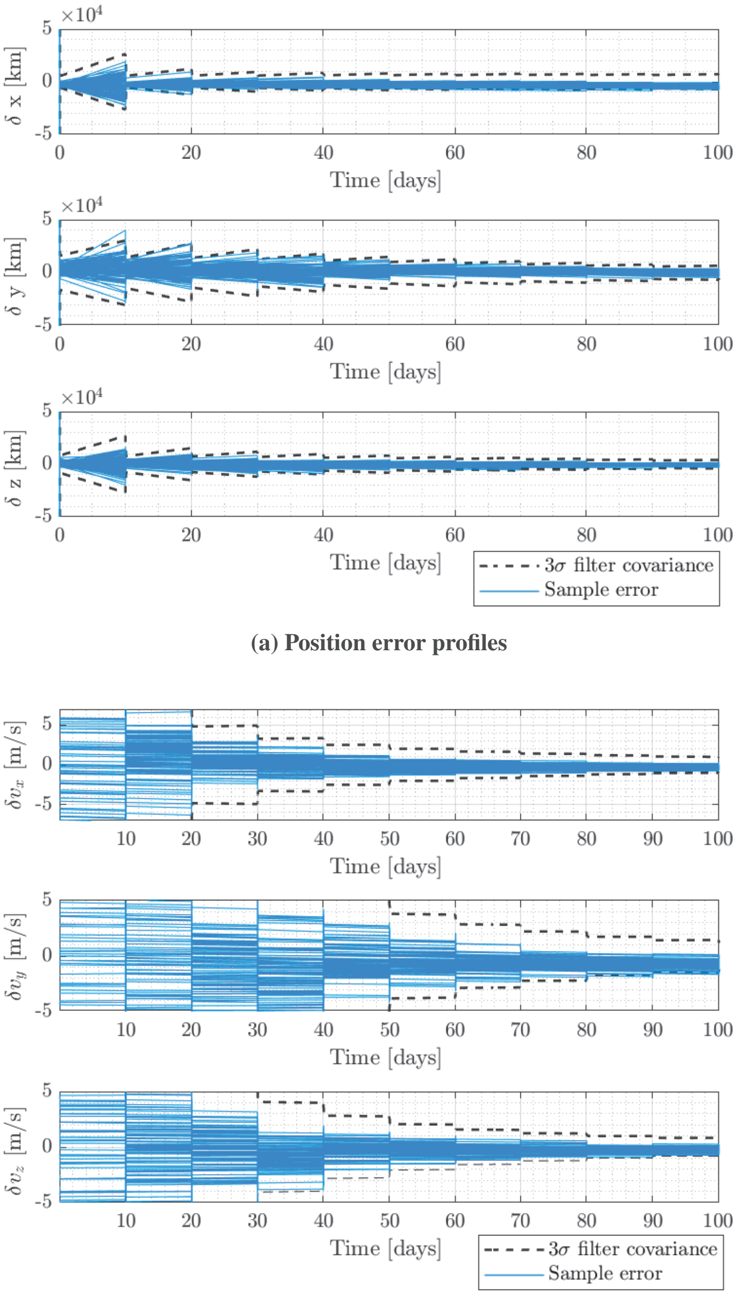
Table 4 Accuracy of the state components at t_0

σ_{r} [km]	σ_{v} [km/s]	σ_{SRP} [km/s^2]	σ_{R} [km/s^2]
10^5	10^{-1}	10^{-10}	10^{-10}

Whereas, the measurement error covariance matrix $\boldsymbol{R}$ has been accurately defined before the dynamic simulation by evaluating the standard deviation of the planet centroids error in the images adopted by the first Monte Carlo sample. Therefore, the obtained $\boldsymbol{R}$ is:

$$\boldsymbol{R} = \begin{bmatrix} 0.37^2 & 0 \\ 0 & 0.45^2 \end{bmatrix} \text{px}^2 \tag{26}$$

The resulting position and velocity error profiles are shown in Figs. 18. Here, the black dashed line represents the 3σ filter covariance bound, whereas the blue line represents the error profile of a sample. At the end of the trajectory leg, the filter estimates the spacecraft position and velocity with a 3σ accuracy of 8600 km and 1.6 m/s, respectively. The degradation in performance is strictly tied to the features measurement uncertainty in the facility images, surpassing three times the magnitude observed in their



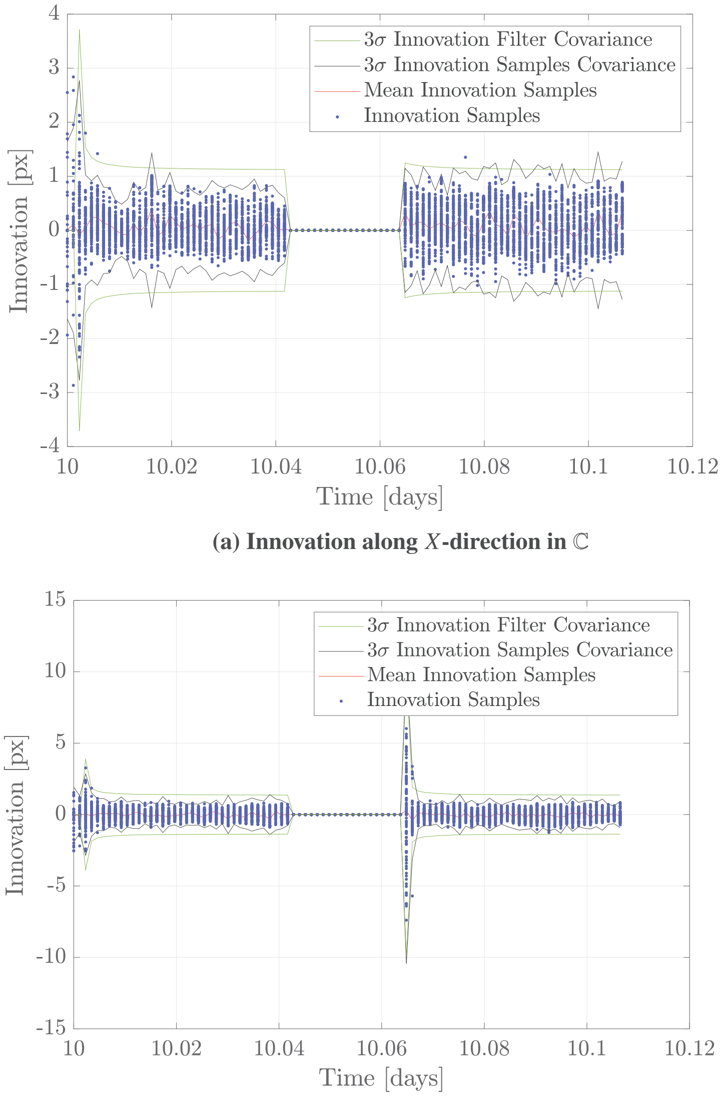
(a) Position error profiles

(b) Velocity error profiles

Fig. 18 Estimated state error profiles with related 3σ bounds.

synthetic counterparts. This increase in uncertainty is, here, specifically driven by the size of the OSI screen iFoV, which is greater than the camera one. Instead, in a real operational scenario, the measurement uncertainty would be influenced by the camera iFoV. Consequently, being this latter smaller in size, the navigation algorithm would compute the spacecraft state with heightened precision. The navigation filter performance obtained in this simulation is conservative compared to what would be achieved in a real scenario where the same camera is used. Instead, the obtained performance could be attributed to a scenario in which a camera with lower resolution is employed, and its iFoV is comparable to that of the OSI screen.

Eventually, to verify that the filter is well-tuned, the innovation profiles are computed [40]. The innovation, i.e. the difference between the observation extracted at epoch t_k and the measurement model evaluated at epoch t_k, must be a zero-mean stochastic process with a covariance equal to $\boldsymbol{H}_k \boldsymbol{P}_{p_k} \boldsymbol{H}_k^\top + \boldsymbol{R}_k$. If the filter sample innovations respect these criteria, the parameters of the filter are well-tuned. Figures 19a and 19b show a closeup of the innovation profiles over the second observation window along the X-direction and Y-direction of the 2D camera reference frame $\mathbb{C}$. In particular, the green line represents the 3σ innovation filter covariance profile, the dashed black line the 3σ innovation sample covariance profile, and the red line the mean of the innovation samples. It is possible to notice that the criteria are almost respected, ensuring the proper tuning of the filter.



(a) Innovation along X-direction in $\mathbb{C}$

(b) Innovation along Y-direction in $\mathbb{C}$

Fig. 19 Innovation Profiles over the Second Observation Window

VIII. Conclusion

The goal of this work consisted of the development of a procedure for the HIL validation of an optical autonomous navigation algorithm for interplanetary cruises through the exploitation of Jena Optronik's OSI. At this aim, a calibration and compensation pipeline is presented to correct the effects introduced by the presence of the hardware. Furthermore, the sky-field rendering engine adopted to display images on the OSI screen is described. Eventually, the autonomous navigation algorithm is validated through static and dynamic simulations to assess its performance with an actual camera in the loop. At the end of the Earth-Mars trajectory, the filter estimates the spacecraft position and velocity with an accuracy of about 8600 km and 1.6 m/s respectively, which are values greater than the ones found when the rendering engine in software mode generated synthetic deep-space images.

Future analysis should better refine the calibration procedure by decoupling the camera calibration from the OSI one. Moreover, to increase the simulation fidelity, a more rigorous radiometric calibration of the facility has to be performed with the adoption of a powermeter. Eventually, to validate the suitability of the algorithm to be run on a CubeSat onboard processor, the VBN filter should be deployed on a miniaturized processor whose computational capabilities are representative of an onboard computer.

Acknowledgments

This research is part of EXTREMA, a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 864697).

References

[1] Lichten, S. M., Abraham, D. S., Arroyo, B., Asmar, S. W., Bell, J., and Edwards, C. D., "Allocation of Deep Space Network Ground System Tracking and Communications Assets During the 2020-2021 Timeframe of the "Mars Armada"," *2018 SpaceOps Conference*, 2018. https://doi.org/10.2514/6.2018-2502.

[2] Wang, Y., Zheng, W., Sun, S., and Li, L., "X-ray Pulsar-Based Navigation Using Time-Differenced Measurement," *Aerospace Science and Technology*, Vol. 36, 2014, pp. 27–35. https://doi.org/https://doi.org/10.1016/j.ast.2014.03.007.

[3] Border, J. S., and Thornton, C. L., *Future Directions in Radiometric Tracking*, John Wiley & Sons, Ltd, 2000, Chap. 3, pp. 9–46.

[4] Turan, E., Speretta, S., and Gill, E., "Autonomous Navigation for Deep-Space Small Satellites: Scientific and Technological Advances," *Acta Astronautica*, Vol. 193, 2022, pp. 56–74. https://doi.org/https://doi.org/10.1016/j.actaastro.2021.12.030.

[5] Bhaskaran, S., Riedel, J., Synnott, S., and Wang, T., "The Deep Space 1 Autonomous Navigation System-A Post-Flight Analysis," *Astrodynamics Specialist Conference*, Denver, CO, USA, 2000. https://doi.org/https://doi.org/10.2514/6.2000-3935.

[6] Franzese, V., Topputo, F., Ankersen, F., and Walker, R., "Deep-Space Optical Navigation for M-ARGO Mission," *The Journal of the Astronautical Sciences*, Vol. 68, No. 4, 2021, pp. 1034–1055. https://doi.org/https://doi.org/10.1007/s40295-021-00286-9.

[7] Andreis, E., Franzese, V., and Topputo, F., "Onboard Orbit Determination for Deep-Space CubeSats," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 8, 2022, pp. 1466–1480. https://doi.org/10.2514/1.G006294.

[8] Henry, S., and Christian, J. A., "Absolute Triangulation Algorithms for Space Exploration," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 1, 2023, pp. 21–46. https://doi.org/https://doi.org/10.2514/1.G006989.

[9] Merisio, G., and Topputo, F., "Present-Day Model of Lunar Meteoroids and Their Impact Flashes for LUMIO Mission," *Icarus*, Vol. 389, 2023, p. 115180. https://doi.org/https://doi.org/10.1016/j.icarus.2022.115180.

[10] Panicucci, P., Lebreton, J., Brochard, R., Zenou, E., and Delpech, M., "Shadow-Robust Silhouette Reconstruction for Small-Body Applications," *Journal of Spacecraft and Rockets*, Vol. 60, No. 3, 2023, pp. 812–828. https://doi.org/https://doi.org/10.2514/1.A35444.

[11] Panicucci, P., Lebreton, J., Brochard, R., Zenou, E., and Delpech, M., "Vision-based estimation of small body rotational state," *Acta Astronautica*, Vol. 213, 2023, pp. 177–196. https://doi.org/https://doi.org/10.1016/j.actaastro.2023.08.046.

[12] Pugliatti, M., Franzese, V., and Topputo, F., "Data-Driven Image Processing for Onboard Optical Navigation Around a Binary Asteroid," *Journal of Spacecraft and Rockets*, Vol. 59, No. 3, 2022, pp. 943–959. https://doi.org/https://doi.org/10.2514/1.A35213.

[13] McCarthy, L. K., Adam, C. D., Leonard, J. M., Antresian, P. G., Nelson, D., Sahr, E., Pelgrift, J., Lessac-Chenen, E. J., Geeraert, J., and Lauretta, D., "OSIRIS-REx Landmark Optical Navigation Performance During Orbital and Close Proximity Operations at Asteroid Bennu," *AIAA SCITECH 2022 Forum*, San Diego, CA & Virtual, January 2022. https://doi.org/10.2514/6.2022-2520.

[14] Maass, B., Woicke, S., Oliveira, W. M., Razgus, B., and Krüger, H., "Crater Navigation System for Autonomous Precision Landing on the Moon," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 8, 2020, pp. 1414–1431. https://doi.org/https://doi.org/10.2514/1.G004850.

[15] Leroy, B., Medioni, G., Johnson, E., and Matthies, L., "Crater detection for autonomous landing on asteroids," *Image and Vision Computing*, Vol. 19, No. 11, 2001, pp. 787–792. https://doi.org/https://doi.org/10.1016/S0262-8856(00)00111-6.

[16] Lauretta, D., Balram-Knutson, S., Beshore, E., Boynton, W., Drouet d'Aubigny, C., DellaGiustina, D., Enos, H., Golish, D., Hergenrother, C., Howell, E., et al., "OSIRIS-REx: Sample Return from Asteroid (101955) Bennu," *Space Science Reviews*, Vol. 212, 2017, pp. 925–984. https://doi.org/10.1007/s11214-017-0405-1.

[17] Panicucci, P., and Topputo, F., "The TinyV3RSE Hardware-in-the-Loop Vision-Based Navigation Facility," *Sensors*, Vol. 22, No. 23, 2022, p. 9333. https://doi.org/https://doi.org/10.3390/s22239333.

[18] Bella, S. A., Andreis, E., Franzese, V., Panicucci, P., and Topputo, F., “Line-of-Sight Extraction Algorithm for Deep-Space Autonomous Navigation,” *2021 AAS/AIAA Astrodynamics Specialist Conference*, Virtual, 2021. URL https://re.public.polimi.it/retrieve/e0c31c11-a820-4599-e053-1705fe0aef77/BELLS01-21.pdf.

[19] Rowell, N., Parkes, S., Dunstan, M., and Dubois-Matra, O., “PANGU: Virtual spacecraft image generation,” *5th Int. Conf. on Astrodynamics Tools and Techniques, ICATT*, 2012.

[20] Brochard, R., Lebreton, J., Robin, C., Kanani, K., Jonniaux, G., Masson, A., Despré, N., and Berjaoui, A., “Scientific image rendering for space scenes with the SurRender software,” *arXiv preprint arXiv:1810.01423*, 2018. https://doi.org/10.48550/arXiv.1810.01423.

[21] Pugliatti, M., Buonagura, C., and Topputo, F., “CORTO: The Celestial Object Rendering TOol at DART Lab,” *Sensors*, Vol. 23, No. 23, 2023. https://doi.org/10.3390/s23239595, URL https://www.mdpi.com/1424-8220/23/23/9595.

[22] Samaan, M. A., Steffes, S. R., and Theil, S., “Star tracker real-time hardware in the loop testing using optical star simulator,” *Spaceflight Mechanics*, Vol. 140, 2011.

[23] Filipe, N., Jones-Wilson, L., Mohan, S., Lo, K., and Jones-Wilson, W., “Miniaturized star tracker stimulator for closed-loop testing of cubesats,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, 2017, pp. 3239–3246. https://doi.org/https://doi.org/10.2514/1.G002794.

[24] Pugliatti, M., Franzese, V., Panicucci, P., and Topputo, F., “TINYV3RSE: The DART Vision-Based Navigation Test-bench,” *AIAA Scitech 2022 Forum*, San Diego, CA & Virtual, January 2022, p. 1193. https://doi.org/https://doi.org/10.2514/6.2022-1193.

[25] Rufino, G., Accardo, D., Grassi, M., Fasano, G., Renga, A., and Tancredi, U., “Real-time hardware-in-the-loop tests of star tracker algorithms,” *International Journal of Aerospace Engineering*, Vol. 2013, 2013. https://doi.org/https://doi.org/10.1155/2013/505720.

[26] Kazemi, L., Enright, J., and Dzamba, T., “Improving Star Tracker Centroiding Performance in Dynamic Imaging Conditions,” *2015 IEEE Aerospace Conference*, IEEE, Big Sky, MT, USA, 2015, pp. 1–8. https://doi.org/10.1109/AERO.2015.7119226.

[27] Wan, X., Wang, G., Wei, X., Li, J., and Zhang, G., “Star centroiding based on fast Gaussian fitting for star sensors,” *Sensors*, Vol. 18, No. 9, 2018, p. 2836. https://doi.org/10.3390/s18092836.

[28] Andreis, E., Panicucci, P., Ornati, F., Topputo, F., et al., “Towards Validation and Verification of Autonomous Vision-Based Navigation for Interplanetary Spacecraft,” *12th International Conference on Guidance, Navigation & Control Systems (GNC) and 9th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2023, pp. 1–14.

[29] Franzese, V., and Topputo, F., “Deep-Space Optical Navigation Exploiting Multiple Beacons,” *The Journal of the Astronautical Sciences*, Vol. 69, No. 2, 2022, pp. 368–384. https://doi.org/https://doi.org/10.1007/s40295-022-00303-5.

[30] Franzese, V., and Topputo, F., “Celestial Bodies Far-Range Detection with Deep-Space CubeSats,” *Sensors*, Vol. 23, No. 9, 2023, p. 4544. https://doi.org/https://doi.org/10.3390/s23094544.

[31] Raymond Karimi, R., and Mortari, D., “Interplanetary Autonomous Navigation Using Visible Planets,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 6, 2015, pp. 1151–1156. https://doi.org/https://doi.org/10.2514/1.G000575.

[32] Jean, I., Ng, A., and Misra, A. K., “Impact of solar radiation pressure modeling on orbital dynamics in the vicinity of binary asteroids,” *Acta Astronautica*, Vol. 165, 2019, pp. 167–183. https://doi.org/https://doi.org/10.1016/j.actaastro.2019.09.003.

[33] Carpenter, J. R., and D’Souza, C. N., “Navigation Filter Best Practices,” Tech. Rep. 20180003657, NASA, 04 2018.

[34] Andreis, E., Panicucci, P., Franzese, V., and Topputo, F., “A Robust Image Processing Pipeline for Planets Line-Of-Sight Extraction for Deep-Space Autonomous Cubesats Navigation,” *44th AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, USA, 2022. URL https://re.public.polimi.it/retrieve/handle/11311/1199156/695619/ANDRE01-22.pdf.

[35] Andreis, E., Panicucci, P., and Topputo, F., “An Image Processing Pipeline for Autonomous Deep-Space Optical Navigation,” *arXiv preprint arXiv:2302.06918*, 2023. https://doi.org/doi.org/10.48550/arXiv.2302.06918.

[36] Mortari, D., Samaan, M. A., Bruccoleri, C., and Junkins, J. L., “The pyramid star identification technique,” *Navigation*, Vol. 51, No. 3, 2004, pp. 171–183. https://doi.org/https://doi.org/10.1002/j.2161-4296.2004.tb00349.x.

[37] Hartley, R., and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press 2000, 2004, Chap. 3 Estimation – 2D Projective Transformations, pp. 117–120.

[38] Liu, H., Shah, S., and Jiang, W., “On-line outlier detection and data cleaning,” *Computers & chemical engineering*, Vol. 28, No. 9, 2004, pp. 1635–1647. https://doi.org/https://doi.org/10.1016/j.compchemeng.2004.01.009.

[39] Franzese, V., and Topputo, F., “Optimal Beacons Selection for Deep-Space Optical Navigation,” *The Journal of the Astronautical Sciences*, Vol. 67, No. 4, 2020, pp. 1775–1792. https://doi.org/https://doi.org/10.1007/s40295-020-00242-z.

[40] Simon, D., *Additional topics in Kalman filtering*, John Wiley & Sons, Ltd, 2006, Chap. 10, pp. 297–329. https://doi.org/10.1002/0470045345.ch6.