75nd International Astronautical Congress (IAC), 14-18 October 2024, Milan, Italy.

IAC-24,B4,IP,93,x85058

Autonomous Vision-Based Navigation for Deep-Space CubeSats: Algorithm Development and Hardware Validation

Eleonora Andreis[1], Paolo Panicucci[2], Fabio Ornati[3], Davide Perico[4], Francesco Topputo[5]

The space industry is undergoing a significant transformation, with CubeSats emerging as cost-effective and standardized miniaturized spacecraft that reduce mission costs and development times. As the number of interplanetary CubeSat missions increases, traditional ground-based navigation methods are becoming unsustainable, necessitating the development of autonomous solutions. In response, the ERC-funded EXTREMA project seeks to enable self-driving spacecraft by challenging the current paradigm of deep-space navigation. This work aims to advance the technology readiness level (TRL) of the EXTREMA's optical navigation algorithm for CubeSats, which autonomously localizes spacecraft in interplanetary space using deep-space images of unresolved celestial bodies. The algorithm is progressively validated through hardware-in-the-loop testing, incorporating optical facilities and miniaturized processors to ensure robustness and suitability for CubeSat applications.

1 Introduction

CubeSats play a central role within the new space economy. The flourishing growth experienced so far in Earth orbits will affect deep space soon. CubeSats allow a democratization of space exploration and open new exploitation frontiers to minor players, thanks to their lower production and launching costs compared to standard probes. One step further in the mission cost reduction consists of cutting down the expenses due to ground-based GNC. Moreover, being detached from ground control avoids the problem of ground station saturation, allowing the exploitation of a greater number of probes in space.

Driven by these factors, several works have been focused on the implementation of autonomous vision-based navigation (VBN) algorithm applicable to deep-space scenarios [1, 2, 3, 4]. This work presents the results of the validation process aimed at advancing the Technology Readiness Level (TRL) of the navigation algorithm proposed in [1] study by evaluating its performance through laboratory simulations. This validation includes the integration of Commercial Off-The-Shelf (COTS) components into the simulation loop to ensure the algorithm meets the performance requirements of miniaturized hardware.

While the primary focus is on CubeSat applications, the impact of this research extends to larger spacecraft as well. If the proposed navigation approach proves effective with low-cost, low-performance payloads, it can be adapted for use with larger spacecraft equipped with higher-performance hardware.

2 Problem Statement

In deep space, a spacecraft can estimate its position by acquiring optical information from the observations of unresolved celestial bodies, e.g., planets and asteroids. The LoS to the beacons are used to feed the on-board navigation method, from which the probe state is estimated. When two LoSes directions associated with different beacons are acquired simultaneously, the kinematic celestial triangulation problem can be solved [5, 2]. Considering the two-beacon observation scenario, the solution of the triangulation problem exists as long as when the probe and beacons are not in conjunction or opposition [6].

The method presented so far solves only for the spacecraft position at the same epoch of measurement acquisition. Dynamic methods, e.g., Kalman filtering, are however more accurate and can estimate the full state vector, including the spacecraft velocity. For this reason, they will be preferred in the development of the algorithm. Moreover, the following conservative assumptions are made to specialize the problem to the case of interplanetary CubeSats:

[1]GNC Engineer, Deutsches Zentrum für Luft- und Raumfahrt, GNC Department, Bremen, Germany. eleonora.andreis@dlr.de

[2]Assistant Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, paolo.panicucci@polimi.it

[3]PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, fabio.ornati@polimi.it

[4]PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, davide.perico@polimi.it

[5]Full Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, francesco.topputo@polimi.it

1. Only one miniaturized optical sensor is adopted on board. Thus, almost all the time only one beacon at a time is tracked.
2. Because of the limited performances of the sensor, minor bodies are not detected [7], and therefore only the planets are used to get the measurements.

2.1 Planets Selection

When estimating the probe position, it is beneficial to track those planets that will yield the highest accuracy possible in state determination. This is known as the optimal beacons selection strategy. The approach followed for the selection of the most suitable pair of navigation planets is the one in [6], which is implemented directly into the navigation filter. It works by taking as input the estimated position of the spacecraft and planets ephemerides, and it returns the optimal pair of planets to be tracked. At first, the planets observability is assessed by checking their Solar Exclusion Angle SEA, the apparent magnitude, and Object-to-Pixel Ratio (OPR). Only the planets whose SEA and apparent magnitude pass the threshold values imposed by the optical sensor, and whose OPR<1 can be considered available for tracking.

3 Deep-Space Image Generation

VBN algorithms rely on physically accurate optical images for their development and validation. However, there is a lack of publicly available mission databases, and those that do exist are specific to particular sensors and missions, limited to the operational orbit covered by the spacecraft. This limitation is particularly relevant for far-range navigation.

As a result, it is clear that relying solely on real mission images to evaluate the functionality of VBN algorithms is not feasible. Therefore, the use of high-fidelity image simulators becomes essential. Given the absence of a publicly available rendering tool suitable for deep-space applications, a new high-fidelity rendering engine has been developed, with the initial version introduced in [8].

The updated version of the image simulator is detailed in [9]. It operates in two modes: software-simulation mode and HIL-simulation mode. The software-simulation mode is used when the generated synthetic image is directly input into the navigation filter. In contrast, the hardware mode is used when optical test benches are integrated into the simulation loop.

Both simulation modes share the same rendering approach regarding geometrical modeling. However, they differ in the selection of noise sources to include in the image simulator and in the conversion of irradiance values into pixel digital counts.

For the software-simulation mode, since the image is used directly to correct the estimated state, it must be as realistic as possible. Therefore, it includes all types of noise, both external and internal. Once the geometrical position of the objects is calculated, internal and external effects that alter the location of their related centroids are applied, such as light time, light aberration, optical distortion, and object smearing. Subsequently, the overall radiometric content of the image is determined by adding the various contributing elements. Thus, the standard deviation of the total detector noise is added to the number of photoelectrons of the blurred observed objects, to which the PSF has been applied, along with environmental artifacts.

At this stage, the electron signal is converted into pixel digital counts. Therefore, the output of the software-simulation mode is an image as it would be captured by the modeled sensor in deep space, with intensity expressed in pixel digital counts according to the camera bit depth.

In contrast, for the Hardware-In-the-Loop simulation mode, the simulator is used to conduct HIL simulations involving optical test benches such as OSI and RETINA described in Sec. 5. Specifically, the synthetic deep-space scenes generated by the image simulator are projected onto the micro-display of the optical facility and captured by its camera. The output of this rendering mode is a raw image matrix whose intensity represents the illuminance of the displayed stars, celestial bodies, and environmental artifacts, which must be converted into digital values before being projected on the optical facility screen. This procedure is detailed in [10] and [11, 12] for OSI and RETINA, respectively.

To correctly stimulate the facility camera, various measures need to be implemented within the image simulator. First, the image displayed on the screen must match both the screen resolution and the camera FoV, and only external errors should be introduced, as internal ones are already present due to the use of a real physical camera. Furthermore, after applying the geometrical model, the rendered scene must undergo warping to compensate for distortion and errors introduced by the facility. Finally, the conversion of illuminance values to digital numbers is applied outside the simulator.

4 Navigation Algorithm

The navigation algorithm, whose validation with HIL simulation is carried out in this work, is extensively described in [1]. Only the major details are reported in this section. First, the image processing (IP) pipeline ex-

ploited to extract relevant information from deep-space images is detailed. Then, the filter components are outlined.

4.1 Image Processing

In deep space, the projection of the planet position in the 2D camera reference frame $\mathbb{C}$, i.e., ${}^{\mathbb{C}}\boldsymbol{r}_{\text{pl}}$, or its associated LoS direction, is the only information available to support state estimation. Therefore, the goal of the IP procedure is to recognize the planet projections in the image among the centroids available. The procedure goes through three steps: 1) The probe attitude is determined, 2) the light-aberration correction is applied to bright star centroids, and 3) the planets are identified.

4.1.1 Attitude Determination

As first step, the probe determines its attitude. To this aim, Niblack's thresholding method [13] is adopted to remove the background noise to patches of the image centered on bright pixels and delimited by squared windows with a one-pixel margin 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 window [14].
Then, the registration problem is solved to find the correct matching between the observed star asterism (i.e., stellar pattern) and the cataloged stars in the inertial frame. This last step is performed differently according to whether the planet is acquired for the first time or not. In the former case, the selected lost-in-space (LIS) strategy is the pyramid algorithm detailed in [15].
When the spacecraft is not in LIS mode, it has a rough estimate of its orientation. Therefore, a recursive registration method can be applied.
When stars are identified, the probe attitude is determined by solving Wahba's problem [16] between the stars LoS directions in the camera and inertial reference frame exploiting the Singular Value Decomposition (SVD) method [16]. Moreover, the robustness of the solution to Wahba's problem is increased thanks to the adoption of a RANdom-SAmple Consensus (RANSAC) procedure [17, 18]. The RANSAC algorithm aims to detect the bright objects that have been misidentified by the star identification, which can thus lead to a wrong attitude determination.
When the recursive attitude determination fails, the spacecraft orientation at the following image acquisition is determined again with the LIS method. Conversely, when the LIS algorithm succeeds in the determination of the probe orientation, in the following image acquisition the recursive attitude determination algorithm is adopted.

4.1.2 Light-Aberration Correction

After the first attitude determination, the centroids of the stars are corrected for the light-aberration effect, and the probe attitude is recomputed by taking into account the corrected stars LoS directions. The procedure adopted is described in [5, 1].

4.1.3 Beacon Identification

At this step, the planet must be identified in the image and its projection ${}^{\mathbb{C}}\boldsymbol{r}_{\text{pl}}$ must be extracted. The identification is performed through the evaluation of the mean and covariance matrix associated with the planet location. This latter defines the Gaussian probability of finding the planet in that portion of the image. At first, the expected location of the observed planet in the image plane is evaluated as:

$$ {}^{\mathbb{C}}_{\hbar}\boldsymbol{r}_{\text{pl}_0} = \boldsymbol{K}_{\text{cam}}\boldsymbol{A}_{\text{corr}}({}^{\mathcal{I}}\boldsymbol{r}_{\text{pl}} - {}^{\mathcal{I}}\boldsymbol{r}) \tag{1} $$

where ${}^{\mathcal{I}}\boldsymbol{r}$ is the predicted probe position in the inertial reference frame $\mathcal{I}$, ${}^{\mathcal{I}}\boldsymbol{r}_{\text{pl}}$ is the planet ephemeris, $\boldsymbol{K}_{\text{cam}}$ is the camera intrinsic matrix, and $\boldsymbol{A}_{\text{corr}}$ is the rotation matrix from the inertial reference frame $\mathcal{N}$ to the camera reference frame $\mathcal{C}$ corrected for the stars light aberration. If ${}^{\mathbb{C}}\boldsymbol{r}_{\text{pl}_0}$ falls within the boundaries of the image, its associated 3σ covariance ellipse is computed. The latter depends on the uncertainties of the spacecraft pose and planet position and is centered in ${}^{\mathbb{C}}\boldsymbol{r}_{\text{pl}_0}$. The ellipse of ${}^{\mathbb{C}}\boldsymbol{r}_{\text{pl}_0}$ represents the area of the image where the planet is most likely to be found within a 3σ probability. The spike contained in the 3σ ellipse is identified as the planet location ${}^{\mathbb{C}}\boldsymbol{r}_{\text{pl}}$. If multiple spikes are located within this ellipse, the closest one to the expected planet position is identified as the planet, as it is most likely to be the true planet location.
The covariance matrix of the beacon location $\boldsymbol{P}$ due to the spacecraft pose (i.e., probe attitude and position) and beacon position uncertainty is computed as

$$ \boldsymbol{P} = \boldsymbol{G}\boldsymbol{S}\boldsymbol{G}^{\top} \tag{2} $$

$\boldsymbol{G}$ is the matrix describing the mapping between ${}^{\mathbb{C}}\boldsymbol{r}_{\text{pl}}$ and the spacecraft pose and beacon position. $\boldsymbol{S}$ is the covariance matrix of the probe pose and beacon position. A complete description of $\boldsymbol{G}$ is reported in [1].

4.2 Orbit Determination

Firstly, the dynamic and measurement models adopted in the VBN filter are detailed. Successively, the chosen filtering scheme is shown.

4.2.1 Dynamics Model

The process state $\boldsymbol{x}$ is defined as

$$\boldsymbol{x}(t) = [\boldsymbol{r}(t), \boldsymbol{v}(t), \boldsymbol{\eta}(t)]^\top \tag{3}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ are the inertial probe position and velocity, respectively, and $\boldsymbol{\eta}$ is a vector of Gauss–Markov (GM) processes accounting for unmodeled terms: a 3-dimensional residual accelerations $\boldsymbol{\eta}_{\mathrm{R}}$ and the stochastic component of the Solar Radiation Pressure (SRP) $\boldsymbol{\eta}_{\mathrm{SRP}}$; that is, $\boldsymbol{\eta} = [\boldsymbol{\eta}_{\mathrm{R}}, \boldsymbol{\eta}_{\mathrm{SRP}}]^\top$ [19]. The process is modeled using the following equation of motion

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t) + \boldsymbol{w} \tag{4}$$

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

$$\dot{\boldsymbol{x}}(t) = \underbrace{\begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{a}_{\mathrm{Sun}} + \boldsymbol{a}_{\mathrm{SRP}} + \sum_i \boldsymbol{a}_{\mathrm{pl}_i} \\ -\xi_{\mathrm{R}}\boldsymbol{\eta}_{\mathrm{R}} \\ -\xi_{\mathrm{SRP}}\boldsymbol{\eta}_{\mathrm{SRP}} \end{bmatrix}}_{\boldsymbol{f}} + \underbrace{\begin{bmatrix} \boldsymbol{0}_{3x1} \\ \boldsymbol{\eta}_{\mathrm{R}} + \boldsymbol{\eta}_{\mathrm{SRP}} \\ \boldsymbol{w}_{\mathrm{R}} \\ \boldsymbol{w}_{\mathrm{SRP}} \end{bmatrix}}_{\boldsymbol{w}} \tag{5}$$

where $\boldsymbol{a}_{\mathrm{Sun}}$ is the two-body gravitational acceleration, $\boldsymbol{a}_{\mathrm{SRP}}$ is the perturbation given by the SRP, and $\sum_i \boldsymbol{a}_{\mathrm{pl}_i}$ is the i-esimal third-body perturbation. In the Langevin equations that govern the GM processes the coefficients ξ_{R} and ξ_{SRP} define the reciprocal of the correlation times, 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. The covariance of these two process noises can be defined as

$$\mathrm{E}[\boldsymbol{w}_{\mathrm{R}}\boldsymbol{w}_{\mathrm{R}}^\top] = \sigma_{\mathrm{R}}^2\boldsymbol{I}_{3x3} = \boldsymbol{Q}_{\mathrm{R}} \tag{6}$$

$$\mathrm{E}[\boldsymbol{w}_{\mathrm{SRP}}\boldsymbol{w}_{\mathrm{SRP}}^\top] = \sigma_{\mathrm{SRP}}^2\boldsymbol{I}_{3x3} = \boldsymbol{Q}_{\mathrm{SRP}} \tag{7}$$

$$\tag{8}$$

The complete process noise covariance matrix is $\boldsymbol{Q}$:

$$\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{0}_{3x3}, \boldsymbol{0}_{3x3}, \boldsymbol{Q}_{\mathrm{R}}, \boldsymbol{Q}_{\mathrm{SRP}}) \tag{9}$$

4.2.2 Measurement Model

The adopted measurement model expresses the observations in pixel coordinates in the camera plane. In addition, it embeds the light effects and their dependencies with respect to the planet and spacecraft state. Therefore, the navigation filter takes into account these effects during the mean and covariance update.

Once the time delay Δt is computed by solving the second-order linear equation reported in [1], the planet LoS can be expressed as the unit vector for the spacecraft position at time t, the image acquisition time, to the planet position at time τ, when the light was emitted by the planet. Thus:

$$\boldsymbol{l}_{\mathrm{pl/sc}} = \frac{(\boldsymbol{r}_{\mathrm{pl}}(t-\Delta t) - \boldsymbol{r}(t))^\top (\boldsymbol{r}_{\mathrm{pl}}(t-\Delta t) - \boldsymbol{r}(t))}{\left|\left|(\boldsymbol{r}_{\mathrm{pl}}(t-\Delta t) - \boldsymbol{r}(t))^\top (\boldsymbol{r}_{\mathrm{pl}}(t-\Delta t) - \boldsymbol{r}(t))\right|\right|} \tag{10}$$

This unit vector is warped by relativistic light aberration as the spacecraft is not fixed with respect to the inertial reference frame. At first order, this effect can be expressed as follows [20]:

$$\boldsymbol{l}_{\mathrm{pl/sc}}^{\mathrm{aberr}} = \boldsymbol{l}_{\mathrm{pl/sc}} + \boldsymbol{l}_{\mathrm{pl/sc}} \times \left(\boldsymbol{\beta}_{\mathrm{sc}} \times \boldsymbol{l}_{\mathrm{pl/sc}}\right) \tag{11}$$

where $\boldsymbol{\beta}_{\mathrm{sc}} = \dfrac{\boldsymbol{v}}{c}$.

Finally, the warped line of sight is projected in the camera:

$${}^{\mathbb{C}}_{h}\boldsymbol{r}_{\mathrm{pl}} = \boldsymbol{K}_{\mathrm{cam}}\,\boldsymbol{A}_{\mathrm{corr}}\boldsymbol{l}_{\mathrm{pl/sc}}^{\mathrm{aberr}} \tag{12}$$

$${}^{\mathbb{C}}\boldsymbol{r}_{\mathrm{pl}} = \frac{1}{{}^{\mathbb{C}}_{h}\boldsymbol{r}_{\mathrm{pl},(3)}} \begin{pmatrix} {}^{\mathbb{C}}_{h}\boldsymbol{r}_{\mathrm{pl},(1)} \\ {}^{\mathbb{C}}_{h}\boldsymbol{r}_{\mathrm{pl},(2)} \end{pmatrix} \tag{13}$$

where ${}^{\mathbb{C}}_{h}\boldsymbol{r}_{\mathrm{pl}}$ is the projection of the planet line of sight in the image plane in homogeneous coordinates, ${}^{\mathbb{C}}\boldsymbol{r}_{\mathrm{pl}}$ is the same vector but in non-homogeneous coordinates, and ${}^{\mathbb{C}}_{h}\boldsymbol{r}_{\mathrm{pl},(i)}$ is the i-esimal coordinate of vector ${}^{\mathbb{C}}_{h}\boldsymbol{r}_{\mathrm{pl}}$.

4.2.3 Selected Filtering Strategy for the Vision-Based Navigation Algorithm

A non-dimensionalized EKF is selected as the most appropriate filtering approach for the development of a VBN algorithm for CubeSat applications. The selection has been performed in [21], where the behavior of five different EKFs has been analyzed in terms of estimator numerical stability and computational performance. Indeed, it is important to remark that the autonomous VBN algorithm has to be deployed on a miniaturized processor characterized by limited computation capabilities comparable to the one of a Raspberry Pi.

Moreover, two additional procedures are implemented in the navigation filter to face the errors of the IP algorithm: 1) When observations are not acquired due to an IP failure, the state vector and its error covariance matrix are simply propagated until the next step; 2) an innovation-based outlier detection method is applied to reject false positives [22]. In particular, when the absolute value of the innovation term ($||\boldsymbol{r}_{\mathrm{pl}_k} - \boldsymbol{h}(\boldsymbol{x}_{p_k})||$) is

greater than $k\sqrt{\boldsymbol{M}_{ii}}$ with $\boldsymbol{M} = \boldsymbol{H}_k \boldsymbol{P}_{p_k} \boldsymbol{H}_k^\top + \boldsymbol{R}_k$ and $k = 3$, the innovation term is set to zero, and the filter correction step is not performed. Indeed, it is preferred to keep an old but good prediction so as not to worsen the estimation.

5 Validation Campaign

To ensure that VBN algorithms are accurate and robust enough for deployment on spacecraft during interplanetary missions, it is crucial to conduct testing in environments that closely replicate actual mission conditions [23]. These testing campaigns simulate the hardware characteristics and mission scenarios a spacecraft would encounter.

While high-fidelity rendering engines help address some of these challenges, they still rely on approximate models that may not fully capture errors from both external and internal sources. Therefore, incorporating real hardware into the image generation process is essential to achieve a more realistic error representation. Optical facilities equipped with real sensors are used to test the navigation software, providing a more accurate assessment of the algorithm performance when unmodeled errors are introduced into the images [23, 24, 25, 26].

Additionally, it is important to evaluate the compatibility of the VBN algorithm with a CubeSat processor. This involves implementing and testing the navigation algorithm on a platform that mimics a CubeSat onboard computer, demonstrating the algorithm feasibility within the constraints of miniaturized hardware.

In this section, the procedure followed for enhancing the TRL of the algorithm through the validation with integrated hardware is described.

For what concerns the optical hardware, two facilities characterized by an incremental levels of fidelity are used: Jena Optronik's OSI and DART's RETINA. The OSI testing was conducted at the GNC department of DLR in Bremen, as detailed in [10], while the RETINA validation took place at the DART LAB. The setup and calibration of RETINA is deeply described in [23, 12, 11]. The OSI facility underwent only geometric calibration before being used in the validation process, whereas RETINA was both geometrically and radiometrically calibrated. Geometric calibration ensures that objects within the field of view are observed in a manner consistent with orbital observation, while radiometric calibration ensures that the light irradiance received by the camera of the optical bench is equivalent to the one the sensor would encounter in an actual mission. Therefore, testing the VBN algorithm with RETINA represents a more advanced step in validation, as the images produced are equivalent to real ones from both a radiometric and geometric point of view.

For what concerns the processor, a Raspberry Pi has been chosen as the representative COTS On-Board-Computer (OBC) for testing due to its affordability, reliability, and compact size, which align well with the CubeSat philosophy[1]. Although these platforms currently outperform many state-of-the-art space-grade processors, an increasing number of users are testing Raspberry Pi computers in space [27, 28]. Specifically, a Raspberry Pi 4 Model B with an ARM Cortex-A72 processor has been utilized.

The deployment of the navigation filter was accomplished using MATLAB Coder[2], which automatically generates optimized C/C++ code from a MATLAB script to run on the Raspberry Pi processor. The navigation algorithm is then invoked within a C program running on the Raspberry Pi, where timing and operational frameworks are implemented. Deep-space images and input variables are stored in the Raspberry Pi primary memory and transferred to the RAM when processed by the navigation filter, as described in [29].

The validation is executed by asynchronously implementing the image acquisition and the state estimation steps. At first, deep-space images are rendered with the rendering engine, projected on the optical facility microdisplay, and acquired by the optical facility camera. Successively, the facility deep-space images are input into the navigation filter deployed on the Raspberry-Pi. The simulation loop is illustrated in Fig. 1. Specifically, the NPU is the processor unit over which the VBN filter is run.

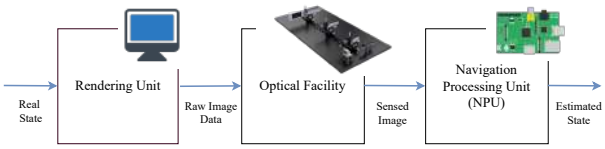

Fig. 1: Simulation loop of EXTREMA Experiment 1

5.1 *Validation with OSI and Raspberry Pi*

OSI is a device designed as a testing system for Jena-Optronik's star sensors[3]. It consists of an optical head, which comprehends a high-resolution micro-display on which deep-space images are rendered, a collimating lens to place the observed objects at infinity, and a control

[1] `https://www.raspberrypi.com/for-industry/space/`, Last Visited in August 2024

[2] `https://it.mathworks.com/products/matlab-coder.html`, Last Visited in August 2024

[3] `https://www.jena-optronik.de/products/star-sensors/ground-support-equipment.html`, Last Visited in August 2024

box. The micro-display is an OLED with resolution 800× 600 pixels and pixel size $d_{\text{pix}_s} = 15\mu$m. The collimating optics has a $f_{\text{coll}} = 30$ mm, and a 20° circular FoV. The combination GC1350–Xenoplan 1.4/17[1][2] is selected as optical system, respectively. The screen pixel angular dimension is about 100 arcsec. This value represents the minimum angular distance between two adjacent features on the screen. Figure 2 shows the final OSI configuration.

Fig. 2: OSI configuration

Fig. 3: Hardware setup for the OSI optical facility

The navigation algorithm is tested on the ballistic martian corridor leg shown in Fig. 4 [30]. The same scenario has been exploited in [1], where the performance of a desktop-simulation of the fully integrated algorithm is presented. These results are used inside this section as a benchmark of the performance when hardware is introduced inside the simulation. The dynamics of the reference true trajectory include the SRP perturbations, the main attractor acceleration, third-body accelerations due to all the planets in the Solar System, and relativistic perturbations. The dynamic model selected for the filter has a lower fidelity. The initial state uncertainties are defined in Table 1, and the navigation concept of operations is described in Sec. 2. The probe physical features adopted to model the SRP are illustrated in Table 2.

The measurement error covariance matrix, which depends on the adopted camera and size of the bright

[1] https://www.alliedvision.com/en/camera-selector/detail/prosilica-gc/1350/, Last Visited in August 2024

[2] https://cdn.alliedvision.com/fileadmin/content/documents/products/accessories/lenses/Schneider/Data_sheet/1001957_Xenoplan_1.4-17.pdf, Last Visited in August 2024

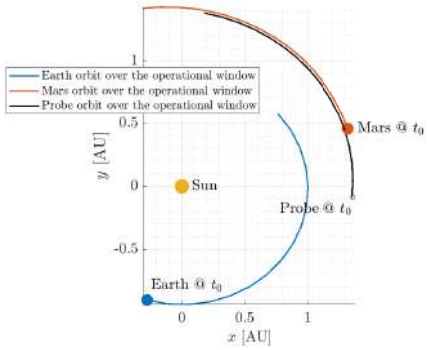

Fig. 4: Leg of the ballistic corridor towards Mars

Tab. 1: Accuracy of the state components at t_0

σ_{r} [km]	10^4
σ_{v} [km/s]	10^{-1}
$\sigma_{\eta_{\text{SRP}}}$ [km/s^2]	10^{-10}
$\sigma_{\eta_{\text{R}}}$ [km/s^2]	10^{-10}

Tab. 2: Spacecraft parameter for the Earth-Mars transfer

A_s [m^2]	m [kg]	C_R [-]
0.32	24	1.2

spots, is:

$$\boldsymbol{R} = \begin{bmatrix} 0.4^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} [\text{px}^2] \tag{14}$$

The measurement uncertainties have been found though the analysis carried out on the performance of the image processing algorithm with OSI in the loop[10].

Figures 5 and 6 show the absolute values of the position and velocity errors, respectively. At the end of the considered leg, the accuracy in position and velocity estimation is 10277 km and 2.13 m/s, respectively. The time required to run a single filter call is 0.2 s.

In [1], the position and the velocity of the spacecraft is estimated with a 3σ accuracy of about 4467 km and 0.97 m/s. The average time required to run one filter call is 0.05 s. With OSI in the simulation loop, the degradation in performance is strictly tied to the uncertainty introduced during the calibration of the optical facility and to the apparent screen pixel size. Using a more resolute screen with the same optical device would enhance the VBN algorithm performance and lower the estimation error, as the calibration error in arcsec would be minimized. This will be evident in the following section. While the navigation filter performance in this simulation is conservative compared to a real-world scenario where the same camera is used without additional cal-

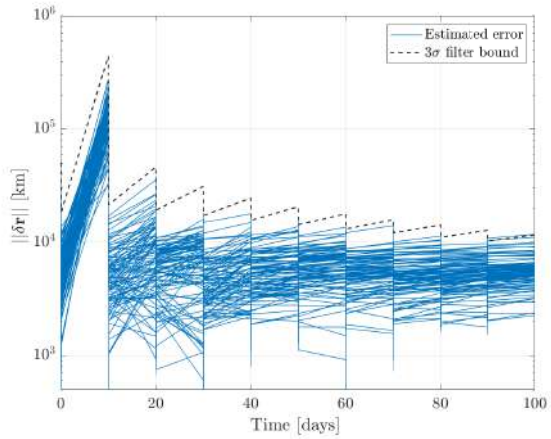

Fig. 5: Position error profiles with OSI and Raspberry Pi in the simulation loop

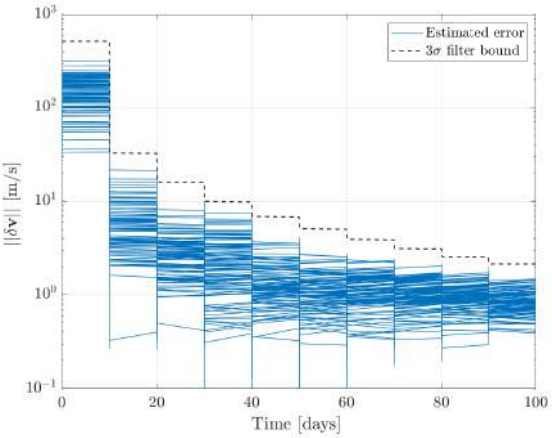

Fig. 6: Velocity error profiles with OSI and Raspberry Pi in the simulation loop

ibration errors, it underscores the filter robustness in the presence of unmodeled errors such as camera misalignment, which can offset the detected location of the celestial body from its nominal position.

5.2 Validation with RETINA and Raspberry Pi

RETINA, shown in Fig. 7, is an optical test bench conceived in the framework of the EXTREMA project, whose architecture and design process has been outlined in [31, 11]. The facility implementation was inspired by the variable-magnification optical stimulator detailed in [32]. Differently from OSI and other fixed magnification facilities present in literature [23, 26, 24, 25], where the collimator focal length is precisely chosen to ensure the matching between testing hardware and screen FoV, RETINA is featured by a multiple corrective optics that guarantees an enhanced versatility in hardware validation. By coupling RETINA, with the Raspberry Pi, the

Fig. 7: RETINA test bench

experiment of the EXTREMA Pillar 1 is performed. Experiment 1 is a building block for the ESH.
For comparison sake, the 100-sample MC run is performed on the same scenario and by exploiting the same initial state uncertainties presented in Sec. 5.1, where the algorithm performance has been tested with OSI in the loop.
In this case, the measurement error covariance matrix is:

$$\boldsymbol{R} = \begin{bmatrix} 0.45^2 & 0 \\ 0 & 0.45^2 \end{bmatrix} [\text{px}^2] \tag{15}$$

The results of the MC runs are reported in Figs. 8 and 9. The blue solid lines represents the 100-sample error profiles, whereas the black dashed lines are the 3σ filter covariance bounds. At the end of the 100 days, the position and velocity errors are computed with a 3σ accuracy of about 5491 km and 1.23 m/s, respectively, which mirrors the results obtained in [1]. The mean CPU time required to call the navigation filter on the Raspberry Pi is 0.31 s. The achieved accuracy in state estimation is higher than the one obtained by exploiting OSI images. This is due to the fact that the error introduced by the facility calibration step is about half than the one introduced by OSI, i.e, about 20 arcsec versus 50 arcsec, respectively. As consequence, the estimation error is about half the one obtained with OSI in the loop. This demonstrates that the adoption of a screen with a smaller apparent angular size, which introduces a smaller deviation on the observable positions, makes the filter achieve better performance.
In both simulations with HIL, the robustness of the VBN algorithm in handling unmodeled errors, such as camera misalignment, which can shift the detected position of the celestial body from its nominal location has been demonstrated. Moreover, the VBN algorithm has been assessed to be compatible with a miniaturized COTS processor.

5.3 Validation with Real Sky-Field Images

To perform a step further in the assessment of the navigation performance, the IP is tested on real raw images

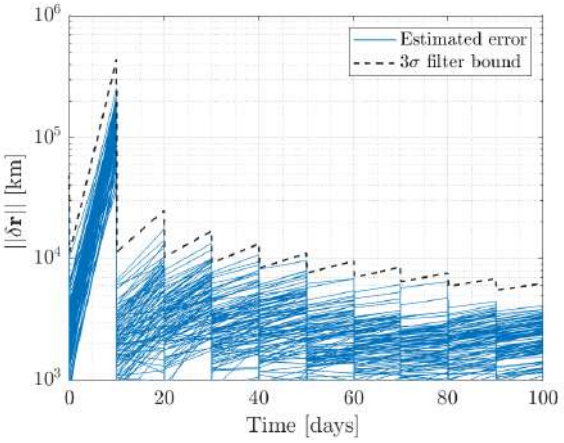

Fig. 8: Position error profiles with RETINA and Raspberry Pi in the simulation loop

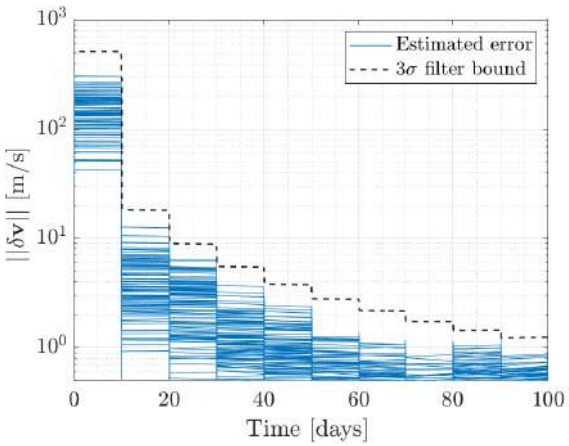

Fig. 9: Velocity error profiles with RETINA and Raspberry Pi in the simulation loop

taken from the ground. A preliminary open-sky campaign has been carried out where various targets with different exposure times have been imaged. The adopted optical device is the RETINA camera (see Sec. 5.2) coupled with two lenses of 25mm and 50mm focal length[1]. The photos were taken with an aperture of F/4.
The true orientation of the spacecraft has been evaluated through identification of stars in the image by exploiting the tool Astrometry.net [33].
The pipeline has been tested on two different portions of the sky, one with Jupiter and the other with Orion constellation.

[1] `https://www.edmundoptics.com/p/25mm-c-series-fixed-focal-length-lens/16528/`, Last Visited in August 2024

5.3.1 Orion Constellation

In this scenario, the pipeline is tested on images without planets, specifically validating the most crucial step of the IP algorithm: attitude determination. The LIS mode of the IP pipeline is applied to Figs. 10 and 11, whose main features are outlined in the respective captions. In Fig. 10, the reference stars employed to compute the true orientation are Betelgeuse (Be), Alnilam (Am), and Alnitak (Ak).
The orientation errors along the pointing axis and the orthogonal axes to the pointing direction are 234 arcsec and 78 arcsec, respectively. Better results can be obtained by calibrating the camera before the application of the IP. Indeed, in this way, the real camera calibration matrix and the optical distortion model would be known and could be used inside the IP. In Fig. 10, the crosses represent the retroprojection of the catalog for the reference stars. Conversely, the squares are the extracted centroids of the reference stars.

The same procedure is adopted for Fig. 11 to compute the true camera orientation. In this case, Orion 51 is exploited instead of Betelgeuse. The LIS mode is activated, which results in attitude determination errors of 5 arcsec and 497 arcsec along the cross-boresight and boresight directions, respectively.

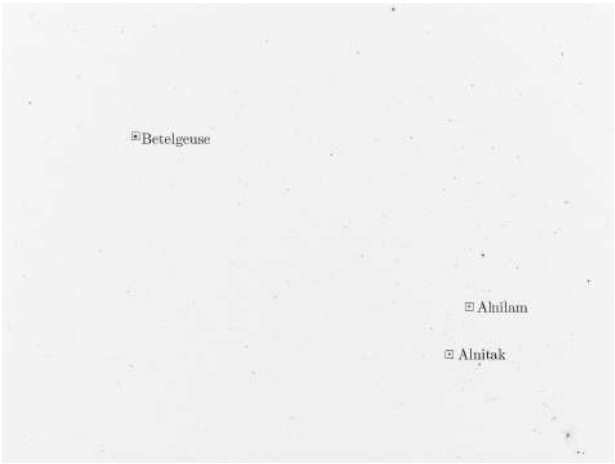

Fig. 10: Real sky-field scenario. $f = 25$ mm, FoV $= [16^\circ \times 12^\circ]$, $T_{\text{exp}} = 1$ s. Credits: Fabio Ornati

Eventually, Fig. 12 shows a close-up of Betelgeuse. The diffraction effect of the camera iris is clearly visible. This consists of an unmodeled optical behavior as it is not considered inside the image simulator. Therefore, the pipeline was not been tested in presence of it when synthetic images were adopted. Despite diffraction and other noise sources, the IP algorithm successfully determines a correct orientation, confirming its robustness.

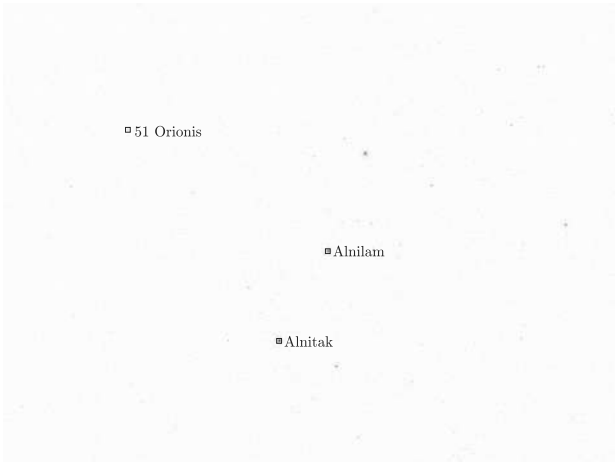

Fig. 11: Real sky-field scenario. $f = 50$ mm, FoV $= [8^\circ \times 6^\circ]$, $T_{\text{exp}} = 0.1$ s. Credits: Fabio Ornati

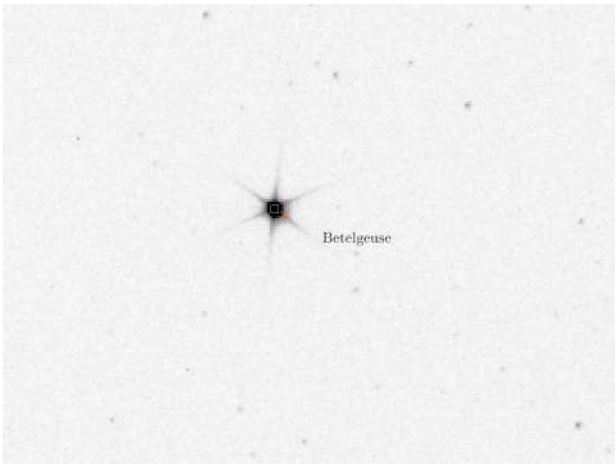

Fig. 12: Close-up on Betelgeuse. The green square represents the centroids of the image, whereas, the red cross is the retroprojection of the catalog

5.3.2 Jupiter

In this scenario, the IP is tested with Jupiter in the image. This time the recursive mode is applied to evaluate the camera orientation. Specifically, a perturbation up to 600 arcsec applied to the nominal attitude is admissible for the determination of a correct attitude solution. Figure 13 shows the results. Identified stars are marked with white squares, while the planet is denoted with a green square. Figure 14 provides a closer view of Jupiter, revealing two of its moons. Here, it is clearly visible the diffraction effect due to the camera iris. Additionally, stray lights caused by the proximity of the Moon to the camera FoV are evident in the upper image portion.

For this scenario, the LIS mode did not converge to any attitude solution due to the limited number of stars in the image, which are obscured by stray light. Moreover, the majority of these stars are located near the edges of the image and therefore heavily affected by optical distortions.

Anyway, the IP is demonstrated to be robust in the planet identification even when various unmodeled errors sources, such as stray-lights from external bodies and optical effects, are present in the image.

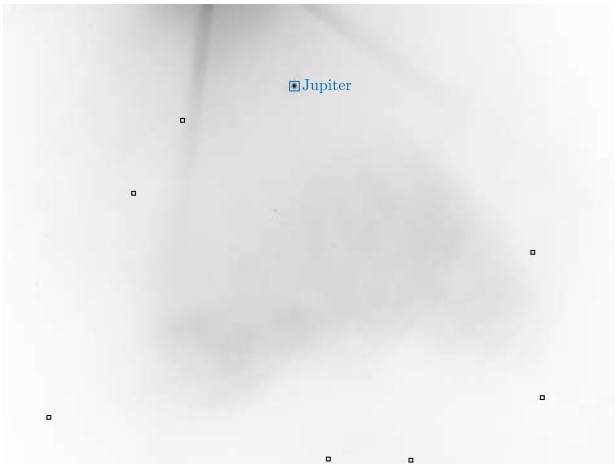

Fig. 13: Real image with identified stars and Jupiter in reverse colors. $f = 25$ mm, FoV $= [16^\circ \times 12^\circ]$, $T_{\text{exp}} = 0.5$ s. Credits: Fabio Ornati

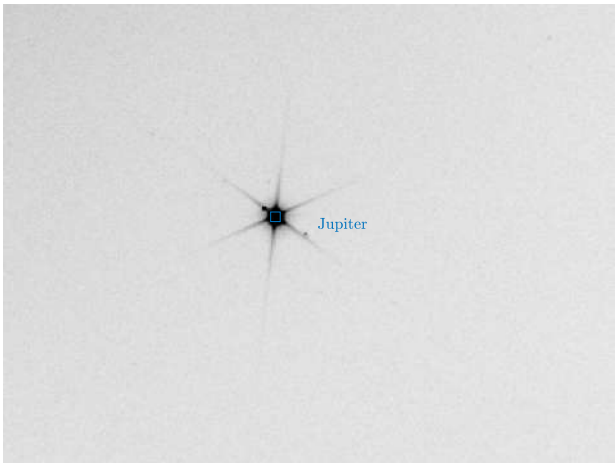

Fig. 14: Close-up on Jupiter.

6 Conclusion and Open Points

The performed HIL simulations demonstrated the robustness of the filter in managing unmodeled errors from optical devices and its compatibility with a miniaturized

COTS processor. As a result, the overall validation with representative hardware was deemed successful, allowing the navigation algorithm to achieve TRL 3.
To advance further, several activities need to be performed. Initially, space-grade processors, such as Zedboard[1] and LEON platforms[2], should be utilized to evaluate the feasibility of the navigation algorithm. A preliminary assessment using the Zedboard has already been conducted, yielding promising results. However, further testing is required to complete the validation. From the optical facility point of view, the camera adopted in the optical facility should be calibrated to decouple the deformation applied by this latter from the one due to the optical facility, making possible to compensate solely the effects due to the presence of the test bench.
Once the algorithm is deployed in a laboratory environment with space-graded hardware, it would achieve TRL 4. Subsequently, an open-sky testing campaign that analyzes the performance of the complete VBN algorithm should be conducted to further mature the software and elevate it to TRL 6. Another option to rapidly advance the TRL scale may involve utilizing images from the recent in-flight demonstration [34], although these are currently not publicly available.

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).

Bibliography

[1] Eleonora Andreis, Paolo Panicucci, and Francesco Topputo. Autonomous vision-based algorithm for interplanetary navigation. *Journal of Guidance, Control, and Dynamics*, 47(9):1792–1807, 2024.

[2] Sébastien Henry and John A Christian. Absolute triangulation algorithms for space exploration. *Journal of Guidance, Control, and Dynamics*, 46(1):21–46, 2023.

[1] `https://digilent.com/shop/zedboard-zynq-7000-arm-fpga-soc-development-board/`, Last Visited in August 2024

[2] `https://www.esa.int/Enabling_Support/Space_Engineering_Technology/LEON_the_making_of_a_microprocessor_for_space`, Last Visited in August 2024

[3] V Franzese and F Topputo. Deep-space optical navigation exploiting multiple beacons. *The Journal of the Astronautical Sciences*, 69(2):368–384, 2022.

[4] Stefano Casini, Angelo Cervone, Bert Monna, and Eberhard Gill. On line-of-sight navigation for deep-space applications: A performance analysis. *Advances in Space Research*, 72(7):2994–3008, 2023.

[5] Reza Raymond Karimi and Daniele Mortari. Interplanetary autonomous navigation using visible planets. *Journal of Guidance, Control, and Dynamics*, 38(6):1151–1156, 2015.

[6] V Franzese and F Topputo. Optimal beacons selection for deep-space optical navigation. *The Journal of the Astronautical Sciences*, 67(4):1775–1792, 2020.

[7] Vittorio Franzese and Francesco Topputo. Celestial bodies far-range detection with deep-space cubesats. *Sensors*, 23(9):4544, 2023.

[8] Salvatore Andrea Bella, E. Andreis, V. Franzese, P. Panicucci, and F. Topputo. Line-of-sight extraction algorithm for deep-space autonomous navigation. In *2021 AAS/AIAA Astrodynamics Specialist Conference*, 2021.

[9] Eleonora Andreis. *Autonomous Vision-Based Navigation for Deep-Space CubeSats : Algorithm Development and Hardware Validation.* PhD thesis, Politecnico di Milano, 2024.

[10] Eleonora Andreis, Hans Krüger, Svenja Woicke, Paolo Panicucci, and Francesco Topputo. *Hardware-In-the-Loop Validation of Autonomous Interplanetary Navigation Algorithms for Interplanetary Cruises with the Optical Star Stimulator.* Orlando, FL, USA.

[11] Fabio Ornati, Paolo Panicucci, Eleonora Andreis, and Francesco Topputo. Retina: a highly-versatile optical facility for camera-in-the-loop testing of spaceborne vision-based sensors. In *46th AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, USA, 2024.

[12] Eleonora Andreis, Paolo Panicucci, Fabio Ornati, Francesco Topputo, et al. Towards validation and verification of autonomous vision-based navigation for interplanetary spacecraft. In *12th International Conference on Guidance, Navigation & Control Systems (GNC) and 9th International Conference on Astrodynamics Tools and Techniques (ICATT)*, Sopot, Poland, 2023.

[13] Laila Kazemi, John Enright, and Tom Dzamba. Improving star tracker centroiding performance in dynamic imaging conditions. In *2015 IEEE Aerospace Conference*, pages 1–8, Big Sky, MT, USA, 2015.

[14] Xiaowei Wan, Gangyi Wang, Xinguo Wei, Jian Li, and Guangjun Zhang. Star centroiding based on fast gaussian fitting for star sensors. *Sensors*, 18(9):2836, 2018.

[15] Daniele Mortari, Malak A Samaan, Christian Bruccoleri, and John L Junkins. The pyramid star identification technique. *Navigation*, 51(3):171–183, 2004.

[16] F Landis Markley and John L Crassidis. *Fundamentals of spacecraft attitude determination and control*, chapter 3 Static Attitude Determination Methods, pages 183–235. Springer, 2014.

[17] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[18] R. Hartley and A. Zisserman. *Estimation – 2D Projective Transformations*, chapter 3, pages 117–120. Cambridge University Press 2000, 2004.

[19] C. N. D'Souza J. R. Carpenter. Navigation Filter Best Practices. Technical Report 20180003657, NASA, 04 2018.

[20] Malcolm D Shuster. Stellar aberration and parallax: a tutorial. *The Journal of the Astronautical Sciences*, 51:477–494, 2003.

[21] Eleonora Andreis, Vittorio Franzese, and Francesco Topputo. Onboard orbit determination for deep-space cubesats. *Journal of guidance, control, and dynamics*, 45(8):1466–1480, 2022.

[22] Hancong Liu, Sirish Shah, and Wei Jiang. On-line outlier detection and data cleaning. *Computers & chemical engineering*, 28(9):1635–1647, 2004.

[23] Paolo Panicucci and Francesco Topputo. The tinyv3rse hardware-in-the-loop vision-based navigation facility. *Sensors*, 22(23):9333, 2022.

[24] Malak A Samaan, Stephen R Steffes, and Stephan Theil. Star tracker real-time hardware in the loop testing using optical star simulator. In *2011 Spaceflight Mechanics*, San Diego, CA, USA, 2011.

[25] Nuno Filipe, Laura Jones-Wilson, Swati Mohan, Kevin Lo, and William Jones-Wilson. Miniaturized star tracker stimulator for closed-loop testing of cubesats. *Journal of Guidance, Control, and Dynamics*, 40(12):3239–3246, 2017.

[26] Giancarlo Rufino, Domenico Accardo, Michele Grassi, Giancarmine Fasano, Alfredo Renga, and Urbano Tancredi. Real-time hardware-in-the-loop tests of star tracker algorithms. *International Journal of Aerospace Engineering*, 2013, 2013.

[27] Steven M Guertin. Raspberry pis for space guideline. Technical report, NASA, 2022.

[28] David Honess and Oliver Quinlan. Astro pi: Running your code aboard the international space station. *Acta Astronautica*, 138:43–52, 2017.

[29] Davide Perico, Gianfranco Di Domenico, Gianmario Merisio, and Francesco Topputo. Elapse: a flatsat software and processing unit for deep-space autonomous gnc systems testing. In *46th AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, USA, 2024. AAS.

[30] Gianmario Merisio, Francesco Topputo, et al. Characterization of ballistic capture corridors aiming at autonomous ballistic capture at mars. In *2021 AAS/AIAA Astrodynamics Specialist Conference*, Virtual, 2022.

[31] Eleonora Andreis, Vittorio Franzese, Francesco Topputo, et al. An overview of autonomous optical navigation for deep-space cubesats. In *72nd International Astronautical Congress*, Dubai, UAE, 2021.

[32] Connor Beierle and Simone D'Amico. Variable-magnification optical stimulator for training and validation of spaceborne vision-based navigation. *Journal of Spacecraft and Rockets*, 56(4):1060–1072, 2019.

[33] Dustin Lang, David W. Hogg, Keir Mierle, Michael Blanton, and Sam Roweis. Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal*, 139(5):1782–1800, 2010.

[34] Michael Krause, Ava Thrasher, Priyal Soni, Liam Smego, Reuben Isaac, Jennifer Nolan, Micah Pledger, E Glenn Lightsey, W Jud Ready, and John Christian. Lonestar: The lunar flashlight optical navigation experiment. *arXiv preprint arXiv:2401.12198*, 2024.