AAS 25-113

WHAT IF STAR TRACKERS WERE NAVIGATION CAMERAS?

Paolo Panicucci[*], Claudia Balossi[†], Fabio Ornati[‡], Felice Piccolo[§], Andrea Pizzetti[¶], Francesco Topputo[‖], and Francesco Capolupo[**]

Navigation cameras and star trackers are widely used sensors in current space missions. Although both rely on similar hardware components and process images to extract information, they serve distinct purposes. Navigation cameras support vision-based navigation by estimating a vehicle's state relative to the observed environment, while star trackers are designed exclusively for precise inertial attitude estimation. This work explores the feasibility of repurposing a high-TRL Leonardo star tracker for navigation tasks. Vision-based navigation algorithms were adapted to account for the hardware, optical, and software constraints of the star tracker, including detector readout and image blurring. Feasibility was evaluated through numerical simulations, processor-in-the-loop, and hardware-in-the-loop testing, assessing both navigation performance and computational load.

INTRODUCTION

Star trackers are commonly used to determine the inertial attitude of a spacecraft. They achieve this by detecting stars, matching them to an onboard catalog, and tracking these matched stars over time. Similarly, other sensors are used to determine the spacecraft state (e.g., position and velocity) over time, a process commonly referred to as navigation.[1] In the last decades, Vision-Based Navigation (VBN) has become a preferred navigation solution due to its cost-effectiveness and high accuracy. Specifically, VBN uses navigation cameras to derive spacecraft state information from observations of the surrounding scene, such as celestial bodies. In deep-space missions, the concurrent need to perform autonomous attitude determination and navigation typically requires at least two types of imaging sensors: star trackers and navigation cameras, each dedicated to a specific task. However, despite their functional differences, star trackers and navigation cameras share similar hardware components (e.g., optics, detectors, and processing units) that could potentially be utilized for both attitude estimation and navigation.

Star trackers are optimized to provide rapid and accurate attitude measurements to the Attitude Determination & Control Subsystem. To precisely measure the stars' location in the field of view, star tracker images are slightly defocused and aberrated. Their optics are also designed to detect very faint stars, and collect as much light as possible from them. In contrast, navigation cameras are designed to observe bright objects (e.g., asteroids, moons, or planets) and have optical designs optimized for minimal distortions, low aberrations, and high telecentricity. From a software standpoint, star trackers prioritize fast attitude determination, often by analyzing partial image data (e.g., regions of interest). Conversely, navigation cameras require more complex software, involving exposure time control and full-frame acquisitions. Due to these differences, star trackers have traditionally been considered unsuitable for accurate navigation tasks. Nevertheless, leveraging star trackers for navigation could reduce mission costs while benefiting spacecraft power, size, and mass budgets. This paper presents findings from the ESA-funded Star Tracker Autonomous Relative Navigation (STAR Nav) project, led by Politecnico di Milano's DART Lab in collaboration with Leonardo SpA. The project investigates to what extent a high-TRL star tracker can be used to navigate in mid-range and close-range scenarios, without any need for dedicated software or hardware modifications.

[*]Assistant Professor, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan, paolo.panicucci@polimi.it
[†]PhD Student, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan, claudia.balossi@polimi.it
[‡]PhD Student, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan, fabio.ornati@polimi.it
[§]PhD Student, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan, felice.piccolo@polimi.it
[¶]PhD Student, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan,andrea.pizzetti@polimi.it
[‖]Full Professor, DAER, Politecnico di Milano, Via La Masa 34, 20156, Milan, francesco.topputo@polimi.i
[**]GNC Systems Engineer, ESTEC, ESA, francesco.capolupo@esa.int

Focal length [mm]	Field of View (FoV) [deg]
50.7	10

Table 1: Main characteristics of the star tracker optical head

To evaluate navigation capabilities of the selected star tracker, three VBN algorithms (two limb-based and one feature-based) were adapted to align with the specific characteristics of Leonardo's AA-STR MK II star tracker. These algorithms underwent a dedicated validation plan, engineered to validate each algorithm incrementally. First, all three algorithms were tested using synthetic images generated with a simulated model of the star tracker. Second, the limb-based VBN algorithms were embedded into the star tracker's microcontroller to assess CPU load, computational performance, and software integration. This phase emulated the eventual porting of the algorithm to the spacecraft onboard computer. Finally, the most efficient limb-based VBN algorithm was validated using hardware-in-the-loop testing in a dedicated optical facility with the actual star tracker unit.
The paper is structured as follows. Section "The star tracker" describes the star tracker used in this study. Section "Overview of VBN algorithms and validation process" provides an overview of the algorithms developed for the STAR Nav study and outlines the validation plan. Section "Image rendering" describes the rendering pipeline developed to simulate star tracker images. Section "The pixel-accuracy limb-based navigation algorithm" details the VBN algorithm developed for mid-range navigation and its performance using synthetic images. Section "Processor-in-the-loop Performance" presents results for processor-in-the-loop testing. Section "Hardware-in-the-loop testing preparation" describes the hardware-in-the-loop testbench used in this study, Section "Hardware-in-the-loop results" discusses the hardware-in-the-loop test results. Section "Conclusions and Lessons learned" summarizes the main outcomes and lessons learned from the activity.

THE STAR TRACKER

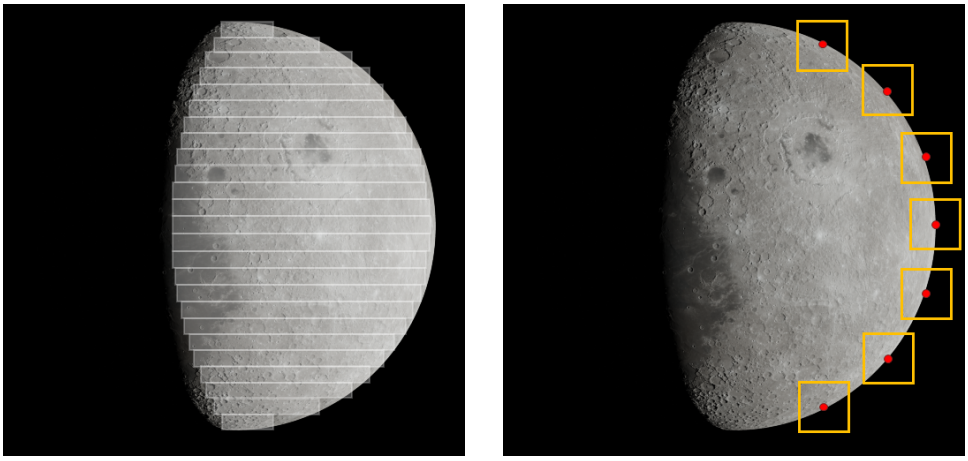
The AA-STR MK II star tracker, part of Leonardo's product line, was selected as the baseline unit for this study.[2] It integrates an optical head and a processing unit within a single mechanical enclosure. The optical head, whose properties are reported in Table 1, is intentionally aberrated and defocused to enhance star centroid determination. The detector is based on the HAS2 sensor, a 1024×1024-pixel APS with a pixel pitch of 18 μm. The processing unit is a space-graded micro-controller with strong memory constraint ($<$ 2MB), requiring careful management of algorithm memory usage. Additionally, the star tracker includes a dedicated ASIC gate array for detector scanning and pixel preprocessing, minimizing memory usage per acquisition. The gate array can also be used to directly feed electrical stimuli to the microcontroller, simulating star tracker acquisitions. This functionality is leveraged during the validation process to perform processor-in-the-loop tests.
In addition to these characteristics, it is important to highlight the key differences between the selected star tracker and a navigation camera. First, the star tracker exposure time t_{exp} is proportional to the number of pixels read by the gate array:

$$t_{\mathrm{exp}} = 8 + 0.17\ N_{\mathrm{pix}} \tag{1}$$

where N_{pix} is the number of read pixels and t_{exp} is expressed in milliseconds. This relationship sets a lower limit of 200 milliseconds for the minimum exposure time required to read the entire image. Second, the star tracker is a hard real-time system. Third, due to bandwidth limitations of the gate array, the star tracker cannot acquire full-frame images. Instead, it supports two specific acquisition modes, illustrated in Figures 1a and 1b:

1. The *full image compressed mode* acquires a list of pixels - hereafter called segments - above certain thresholds.

2. The *windows mode* allows for the selective reading of the detector by defining small windows (patches) within the image. Each window provides pixel content as digital numbers. Up to 16 windows can be defined.



(a) Full Image Compressed Mode. Segment are reported in white.

(b) Window mode. Windows and their centers are reported in orange and red respectively.

Figure 1: Visual representation of the two acqusition modes.

The star tracker can alternate between these two modes during its operation. Finally, the star tracker is, by design, highly sensitive to light, which can result in image saturation when acquiring images of a bright celestial objects, even at low exposure times (e.g., tens of milliseconds).
These unique features and limitations significantly influenced the design of the navigation algorithms. Consequently, state-of-the-art vision-based navigation algorithms were tailored to accommodate the specific characteristics of Leonardo's star tracker.

OVERVIEW OF VBN ALGORITHMS AND VALIDATION PROCESS

The STAR Nav project focuses on developing navigation algorithms for exploration missions, with particular emphasis on their integration into the star tracker hardware. Two scenarios were considered during the project:

1. Mid-range navigation at the Moon exploiting the Moon's limb as inspired by the LUMIO mission[3]
2. Asteroid fly-by vision-based navigation leveraging tracked surface features as inspired by the HERA mission[4]

To address these scenarios, three algorithms were developed:

1. A limb-based navigation algorithm with subpixel accuracy, designed for cislunar applications and alternating segments and windows readings to find the accurate location of the Moon limb. This algorithm has been designed to work on well-exposed images. It first identifies the limb in the image with a segment reading. Then, this information is used to open windows in proximity of the extracted limb and refine the limb position using the pixel intensities in the opened windows.
2. A limb-based navigation algorithm with pixel accuracy, designed for cislunar applications and processing segment readings to identify the Moon limb. This algorithm has been designed to work with both saturated images (as the ones produced by the star tracker) and well-exposed ones, by operating the star tracker in segment mode only. Moreover, this algorithm guarantees low computational load and memory allocation as it does not include any subpixel-accuracy limb refinement.
3. A terrain relative navigation (visual odometry) algorithm for asteroid flyby that processes windows readings to extract and track features on the asteroid surface among windows. The algorithm has been designed to work with well-exposed images and operating in windows mode.

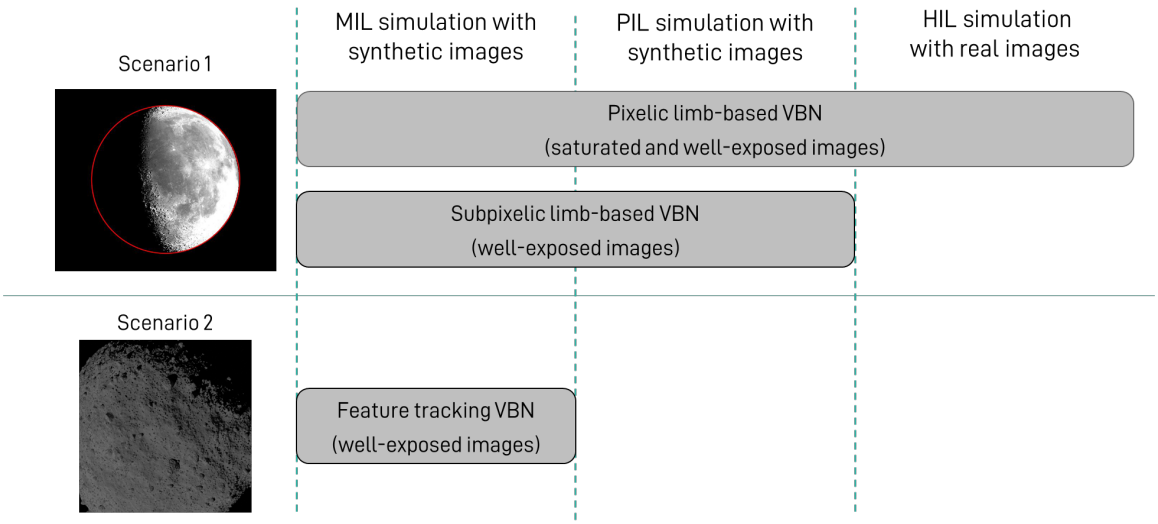


Figure 2: An overview of the validation plan undergone by the three algorithms developed in the study.

Further details on the design of these algorithms can be found in Reference 5 for the lunar scenario and Reference 6 for the asteroid scenario. These references also report the numerical performance of the algorithms on preliminary synthetic images. Figure 2 provides an overview of the algorithms and their corresponding validation steps.

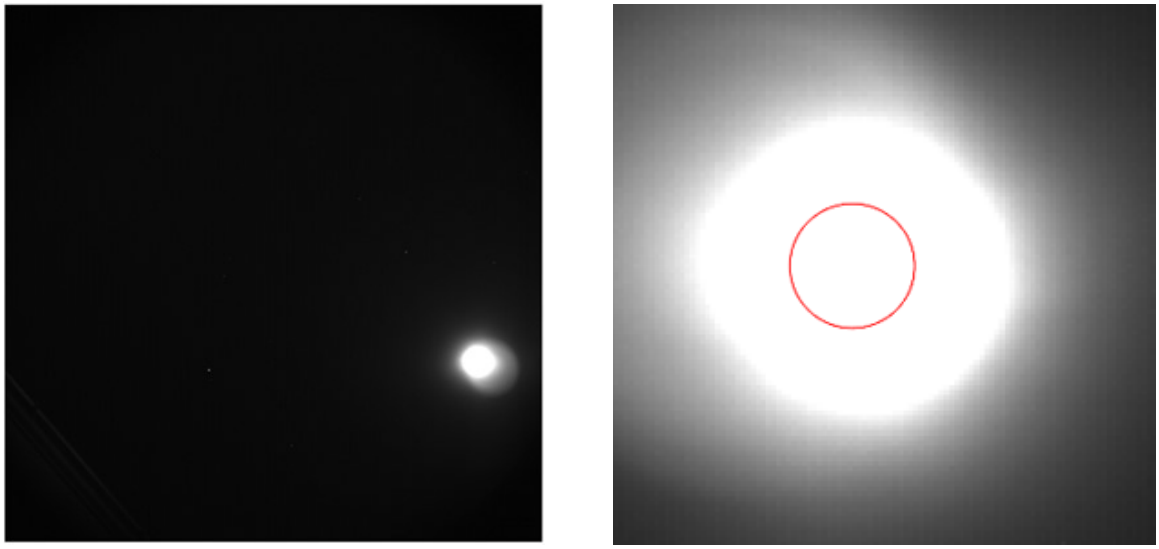
The validation plan is designed to be of incremental complexity to pinpoint possible issues at an early phase of the design. First, model-in-the-loop (MIL) simulations are performed to assess the algorithm performance with synthetic images rendered with a dedicated rendering tool. These simulations provide insights into algorithm performance in a fully controlled environment. Second, the algorithm undergoes processor-in-the-loop (PIL) testing with the same synthetic images. This PIL test verifies the algorithms' compatibility with the star tracker processor, measures any latency in the VBN processing chain when executed on actual hardware, and validates the integration of VBN software with the star tracker application software. Finally, the algorithm is tested in the hardware-in-the-loop (HIL) optical bench TinyV3RSE,[7] which uses a high-resolution screen to optically stimulate the star tracker unit. The VBN software embedded in the processor then processes the acquired data, enabling a full-system validation with camera and processor effects included.

The three algorithms underwent different validation steps due to time constraints in the project execution. The pixel-accuracy limb-based algorithm has been validated up to HIL simulation, ultimately reaching TRL 5. The subpixel-accuracy limb-based algorithm has been tested up to PIL simulation, whereas the feature-based algorithm has been tested up to MIL simulations. Given its higher level of maturity, the remainder of this paper focuses on the pixel-accuracy limb-based algorithm.

IMAGE RENDERING

Image rendering has been a key aspect of the study, given the need to reproduce star tracker acquisitions under different illumination conditions. Several factors are essential for generating realistic star tracker images:

1. The star tracker has a large entrance pupil, which introduces distortions to be modeled.
2. The star tracker is intentionally aberrated and defocused.
3. Designed for observing faint objects, the star tracker is subject to severe saturation and nonlinear detector effects (e.g., blooming, smearing, crosstalk) when observing bright objects. This was confirmed by an open-sky test conducted by Leonardo. The acquired image, shown in Figure 3a, highlights severe



(a) Full image

(b) Zoom of the full image with the expected Moon dimension in red

Figure 3: Opensky test executed with the star tracker (Credits: Leonardo SpA).

saturation occurring for an exposure time of 200 ms. A zoomed detail in Figure 3b shows the expected Moon dimensions in red, emphasizing the saturation and the spread of electrons to adjacent pixels.

A schematic overview of the rendering pipeline developed for the study is shown in Figure 4. The core of the pipeline is based on CORTO,[8] a collection of Python scripts for programmatic rendering with Cycles path-tracing. CORTO has been used in multiple ESA CubeSat missions (e.g., LUMIO[3] and Milani[9]) and and has been geometrically validated by comparisons with images from real missions such as Rosetta, OSIRIS-REx, and Orion. However, CORTO is not radiometrically consistent, implying that it cannot render an image accounting for the physical parameters (e.g., exposure time, pupil size, quantum efficiency) that determine the amount of light collected by the sensor. To overcome this limitation, a dedicated radiometric calibration procedure of CORTO was implemented. A radiometrically consistent rendering procedure, called ABRAM,[10] was developed to align radiometric quantities in Blender's rendering outputs with real-world radiometric quantities. ABRAM has been validated using real images from the SMART-1 mission's AMIE camera, as demonstrated in Figures 5a and 5b. Further validation details are available in 10. Using ABRAM, CORTO

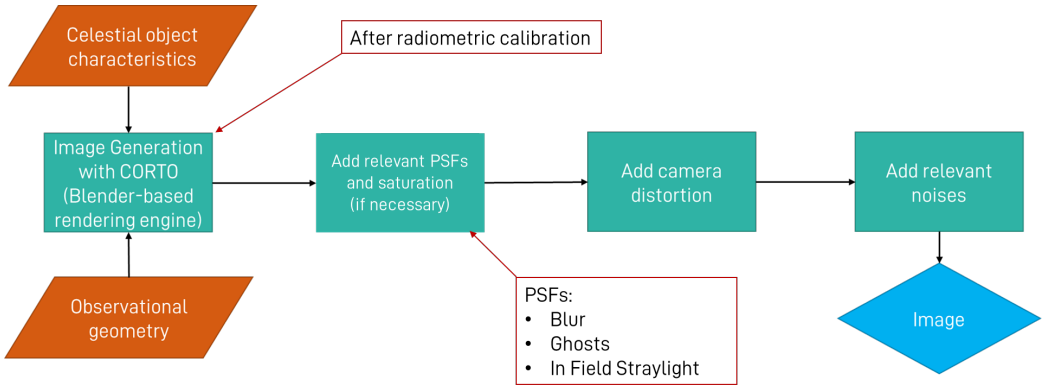


Figure 4: The rendering pipeline.

(a) Image from the AMIE camera

(b) Rendered image

Figure 5: Comparison between a real image and rendered one from the physical-based rendering engine.

(a) Example of well-exposed rendered image

(b) Example of saturated rendered image

Figure 6: Comparison between well-exposed and saturated images rendered with the proposed pipeline.

was tuned to generate radiometrically consistent images. This, together with the use of detailed lunar Digital Elevation Models, enables the generation of accurate synthetic images of the Moon that correctly represent the geometry of the scene and the photon counts collected by the detector.

Images are generated based on the LUMIO trajectory,[11] incorporating the AA STR MK-II characteristics (e.g., focal length, image resolution, and pixel pitch). Moreover, the Moon's position in the image and the Moon phase angle were varied randomly, to test the algorithm robustness. After the generation of these images, point spread function (PSF) has been applied to the image to model the optical head. This includes a PSF modeling aberrations and defocusing, and a second one capturing the infield stray light and ghosts. Then, the pixels exceeding the full well capacity are saturated, before applying distortion and detector noises

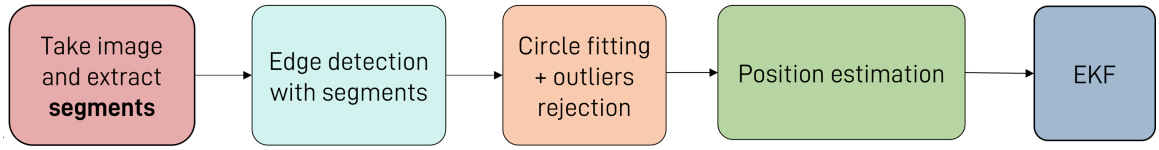


Figure 7: The VBN chain designed for the Algorithm 1.

(i.e., photon shot noise, PRNU, DSNU, and readout).
Examples of well-exposed and saturated images are reported in Figures 6a and 6b. It is worth noting that the developed rendering pipeline incorporates a modeling of blooming. However, due to the lack of real saturated datasets for reference. Therefore, there is the need to validate images in a HIL setup as reported in Section "Validation of rendering pipeline and implications".

THE PIXEL-ACCURACY LIMB-BASED NAVIGATION ALGORITHM

This section provides an overview of the pixel-accuracy limb-based navigation algorithm developed for this study. For a detailed explanation of the algorithm and a comprehensive review of numerical results, readers are encouraged to consult Reference 5. The algorithm operates as follows (see Figure 7):

1. Segments are read from the star tracker and fed to the VBN algorithm for processing.
2. The algorithm identifies the limb location based on the input segments and the Sun's direction. By dividing the image plane into four sectors, the algorithm determines the points exposed to sunlight for each line. For instance, if sunlight originates from the right, the limb point corresponds to the start of the first segment. Similar logic is applied for sunlight from the left, top, or bottom of the image. Additional details are provided in Figure 6 of Reference 5.
3. Once the edges are computed, an outlier rejection step based on RANSAC is performed. The RANSAC fits a circle among the selected points and rejects the ones not fulfilling the circle equation. It is worth noting that the Moon projection in the image is not a circle, but this approximation simplifies the model fitting and the error function to be used in the outlier rejection procedure. Numerical investigation has shown that this approximation yields good conic fitting performances, while limiting calculations due to ellipse fitting.
4. Edge inliers are used to compute the relative position between the star tracker and the Moon's center in the star tracker reference frame. This calculation utilizes the Christian-Robinson algorithm.[12]
5. The position measurement from the algorithm is input into an Extended Kalman Filter (EKF) to estimate the spacecraft's full state in a J2000 reference frame centered on the Earth-Moon barycenter. The EKF's dynamic model incorporates the gravitational attractions of the Moon, Earth, and Sun. The equations of motion are integrated using a fourth-order Runge-Kutta scheme, and the state transition matrix is computed by integrating the variational equations alongside the motion equations. Further details on filter design are available in Reference 3.

The VBN chain was tested on synthetic images to evaluate its performance in MIL simulations. Results from the pixel-accuracy VBN algorithm's limb measurement on a saturated dataset are illustrated in Figure 8. Saturation effects lead to a limb position error of approximately 10 pixels. This is because the PFSs spread rays widely in the field of view and the high exposure time leads to saturate multiple pixels in the neighbor of the limb. Saturation causes a loss of information about the limb's precise location in the image, resulting in a significant bias in limb measurement.
The bias partially propagates to the navigation solution, as the EKF cannot reduce the filter covariance below thousands of kilometers. Limb-based navigation algorithms can typically achieve accuracies within hundreds

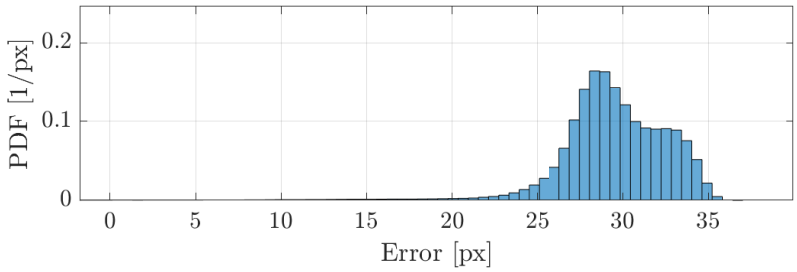
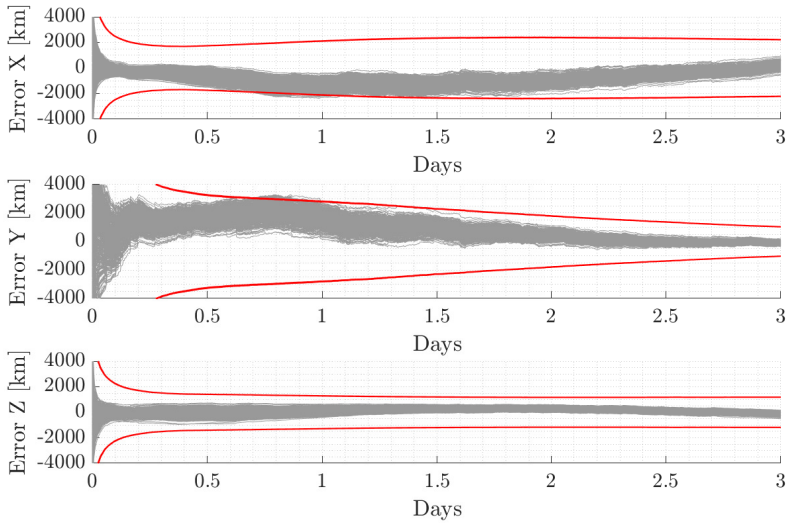
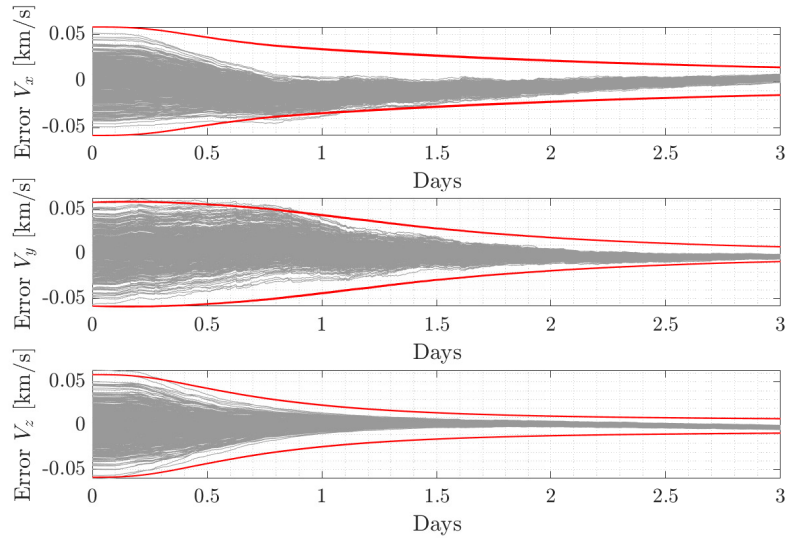


Figure 8: Probability Density Function (PDF) of the edge locations' error with respect to the true limb for MIL simulations on the saturated dataset.

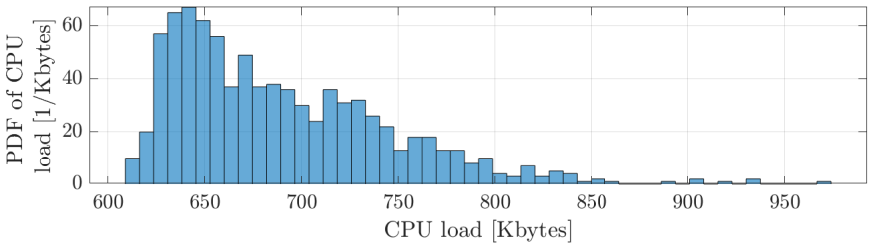


(a) The position error in the inertial reference frame.

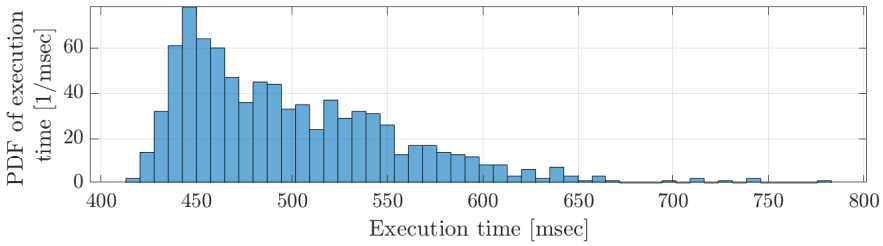


(b) The velocity error in the inertial reference frame.

Figure 9: Results for MIL simulations on the saturated dataset.



(a) The computational load.



(b) The execution time.

Figure 10: Results for PIL simulations on the saturated dataset.

of kilometers when the limb is correctly determined.[3,11] Despite the reduced accuracy in this application, the EKF partially compensates for the image processing bias by incorporating a Gauss-Markov process into the measurement model. Consequently, navigation errors are limited to under 2000 km in position and less than 50 m/s in velocity. Given the probe's orbit between 40,000 and 80,000 km and its velocity oscillating between 0.93 and 1.35 km/s, these errors correspond to a relative accuracy of approximately 5% in both position and velocity. For more detailed discussions on MIL simulations, refer to Reference 5.

PROCESSOR-IN-THE-LOOP PERFORMANCE

Following the assessment of numerical performance, PIL simulations were conducted. PIL testing provides a realistic evaluation of software execution, including CPU load, execution time, and the verification of interfaces between the VBN software and the star tracker software. Additionally, these tests confirmed the equivalence between the embedded software and its prototype implementation in MATLAB.

The PIL test relies on electrically stimulating the star tracker by directly injecting rendered images into its gate array, bypassing both the optics and the detector.

This is accomplished using Electrical Ground Support Equipment (EGSE). Synthetic images are converted into electrical stimuli and fed to the Electrical Stimuli Generator (ESG). Concurrently, the star tracker is managed via telecommands sent by the AASTR Unit Tester (AST), which also collects telemetry data from the star tracker. During the test, the ESG provides electrical stimuli to the star tracker software, which in turn forwards the processed data to the VBN software integrated into the star tracker application. The VBN software, auto-coded from its MATLAB version, underwent adjustments to align with the microprocessor's resource constraints. First, variable types were carefully selected to ensure the required accuracy during execution. Second, the maximum number of extracted limb points was limited to 600, reducing matrix allocation demands in the code. These modifications enable to limit the memory usage on the microprocessor to fulfill the memory requirement.

As the VBN algorithm processes an image every 5 minutes during nominal operations, the PIL test has been performed in an accelerated framework processing one image every 1.5 seconds. This accelerated framework allowed for a small Monte Carlo simulation with 20 samples. In each sample, the spacecraft's initial conditions and algorithm parameters (e.g., attitude knowledge and Sun direction) were perturbed to reflect the typical performance characteristics of CubeSat missions. The results in terms of computational

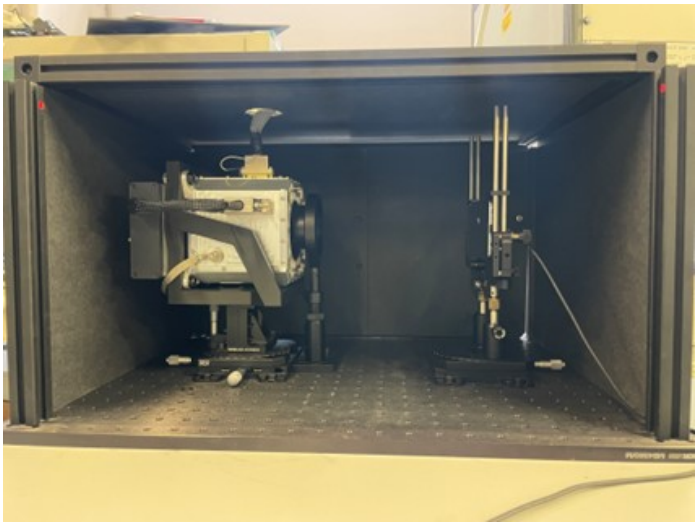


Figure 11: Lateral view of the TinyV3RSE facility with an engineering model of the star tracker.

Angular position in the FoV [deg]	RMS radius on the detector [mm]	RMS radius on the detector [px]
0	8.88	0.49
3.67	12.89	0.72
7.32	26.74	1.49
9.61	38.42	2.13

Table 2: Optical degradation due to the collimator placement between the star tracker and the screen.

load and execution time for these simulations are reported in Figures 10a and 10b. The VBN algorithm needs less than 650 milliseconds to execute the whole VBN chain, using less than 900 kB of memory. These results demonstrate the algorithm's efficiency and its ability to meet the stringent resource constraints typical of star tracker hardware.

HARDWARE-IN-THE-LOOP TESTING PREPARATION

The final step in the validation plan involves Hardware-in-the-Loop (HIL) testing with the star tracker optically stimulated and the processor in the loop. These tests evaluate the algorithm's robustness against errors introduced by the optical system. A non-exhaustive list of such errors includes camera calibration inaccuracies, distortion estimation errors, chromatic and achromatic aberrations, variability in focus across the field of view (FoV), and detector-induced noise. Additionally, this validation phase demonstrates the feasibility of running the IP and VBN algorithm chains on high-TRL components, raising the algorithm's TRL to 5. This system-level integration test showcases that the VBN algorithms can operate effectively on limited-performance hardware and confirms that star trackers can provide valuable information for navigation. Furthermore, it is worth noting that HIL acquisition offers an ideal platform to validate the saturated image rendering pipeline. Since saturated images are not compared against any pre-existing dataset, this test allows the assumptions underpinning saturated image rendering to be tested directly as the star tracker acquires images within the HIL facility.

For the HIL tests to be meaningful, the testbench shall reproduce faithfully the images taken on orbit in terms of geometry (i.e., observational point of view) and radiometry (i.e., the equivalence of image pixels' content). This section outlines the efforts undertaken to prepare the star tracker for the HIL facility, validate the facility itself, and confirm the accuracy of the rendering pipeline.

The HIL tests are conducted by placing the star tracker in a controlled dark environment and stimulating its

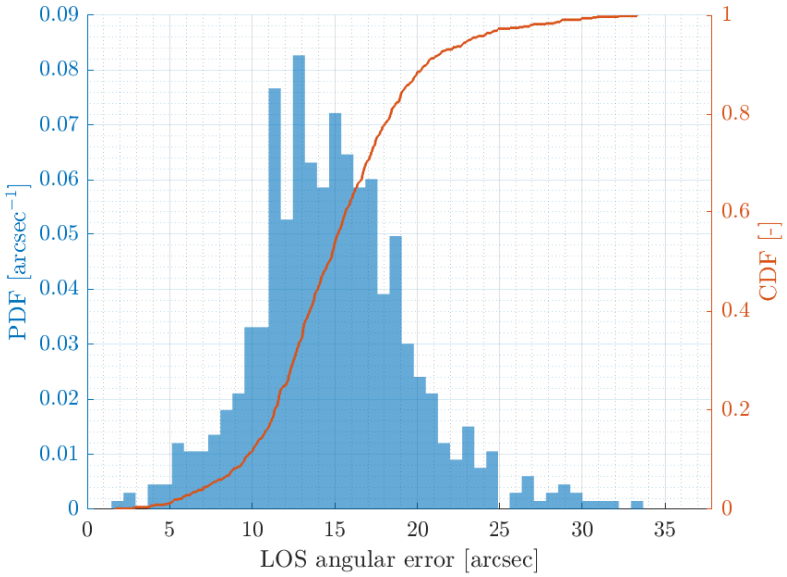


Figure 12: Probably and Cumulative Density Functions of the geometric calibration residuals.

detector using a high-resolution screen. To simulate the collimated light conditions encountered in orbit, a collimator is positioned between the star tracker and the screen. Historically, such HIL setups have been used to test star trackers and attitude determination algorithms.[13,14] More recently, their application has been extended to validate VBN algorithms.[15–17]

In this work, the TinyV3RSE optical testbed[7,18] is used to stimulate the star tracker. The TinyV3RSE testbed consists of three primary assemblies mounted on an optical breadboard and enclosed in a black box to minimize external light pollution. These assemblies are:

1. The star tracker assembly, composed of the mechanical interface with the star tracker and its mechanical support enabling vertical translation, pitch, and yaw mechanical adjustments.

2. The high-resolution screen assembly, composed of the screen and its support, whose orientation is set to ensure that the screen and the optical plane of the camera are parallel. This component is not modified with respect to the previous design of the facility.

3. The collimator assembly, composed of the collimating lens and its support. The collimator support includes an optical holder that can rotate, change in elevation, and be finely adjusted laterally and transversely.

TinyV3RSE customization

It is worth noting that TinyV3RSE was customized to interface with the star tracker. A dedicated mechanical design was implemented to accommodate the star tracker within the facility while ensuring compliance with the facility's internal spatial allocations. The star tracker support was manufactured at DART Lab in PETG with a 100% fill factor to achieve the required rigidity and resistance to bending. Additionally, the optical lens support was carefully selected to prevent vignetting at the star tracker's entrance pupil. The design was tested in a CAD environment prior to production and was subsequently assembled at DART Lab, as illustrated in 11.

The collimator was selected based on the governing optical stimulator equation:[7]

$$\tan\left(\frac{\mathrm{FoV}_{\mathrm{STR}}}{2}\right) \leq \frac{h_{\mathrm{s}}}{2f_{\mathrm{coll}}} \tag{2}$$

where $\mathrm{FoV}_{\mathrm{STR}}$ is the star tracker FoV, h_{s} is the screen height, and f_{coll} is the collimator focal length. Specifically, $h_{\mathrm{s}} = 65$ mm and $FoV_{\mathrm{STR}} = 10$ degrees, $f_{\mathrm{coll}} \leq 180.07$ mm. Based on these requirements, the chosen collimator is an off-the-shelf achromatic doublet lens with a focal length of 180 mm. This lens is a cemented doublet designed to minimize chromatic aberrations in the visible spectrum as light passes through it. Despite the optimized lens selection, a dedicated analysis of the lens-induced aberrations was conducted using Zemax optical simulation software to evaluate the potential degradation of the star tracker's performance due to the collimator. The results of this analysis are presented in Table 2 where the RMS radius represents the root-mean-square radius of all rays emitted from a single location on the screen and impacting the detector, forming a spot diagram. The analysis indicated that the induced optical degradation is negligible, particularly given the defocus inherent to the star tracker design. Therefore, the selected collimator effectively meets the requirements without significantly impacting the star tracker's imaging quality.

Geometric Calibration

To ensure proper stimulation of the star tracker, geometric calibration is required. This calibration addresses geometrical errors caused by optical component misalignments and distortions. The objective is to estimate and compensate for these errors before testing, ensuring accurate stimulation during tests. Specifically, the geometric calibration process involves minimizing the discrepancy between the desired point locations in the star tracker's image and the uncalibrated observed points. This process yields a polynomial fit that represents the facility's distortion model. With this mapping, the distortions can be compensated prior to image stimulation.[7]

To derive the calibration polynomial, a series of points are displayed on the screen and captured by the star tracker. By minimizing the error between the acquired and desired patterns, the geometrical calibration polynomial is computed. More details about the methods and the mathematical derivation can be found in References 7 and 19. This calibration enables the displayed images to be pre-warped, compensating for distortions and misalignments, and resulting in geometrically consistent stimulation. Using this methodology, points displayed on the screen stimulate the camera with an angular error of less than 30 arc-seconds. Figure 12 illustrates the angular error between the desired and obtained lines of sight.

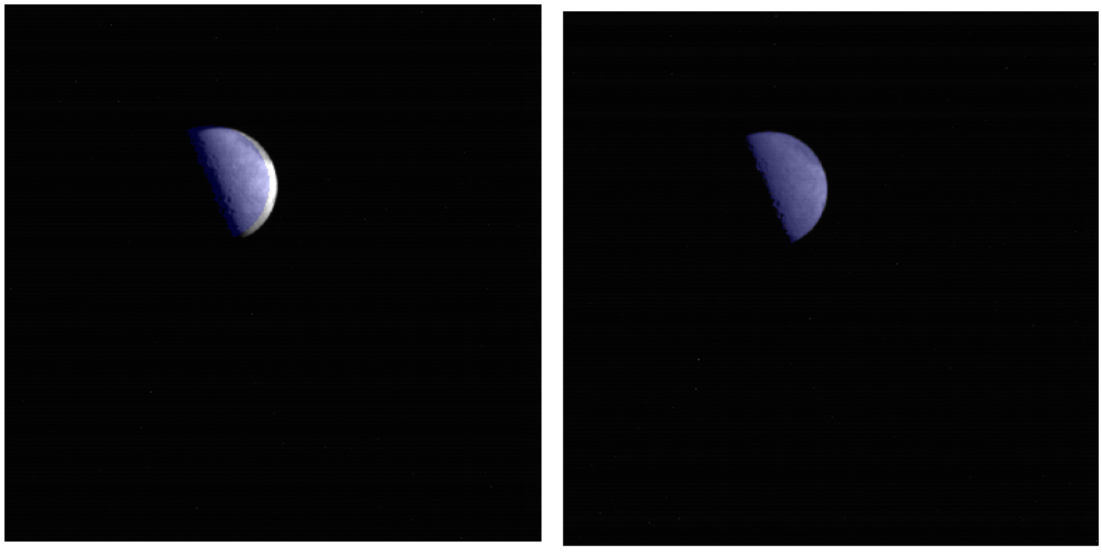
It is worth mentioning that the proposed geometrical calibration technique uses pointwise features (e.g., pixel-accurate dots) to estimate the calibration polynomial. It is crucial to validate this approach for extended objects, such as the Moon. A test was conducted to evaluate the calibration's effectiveness when the Moon is displayed on the screen. Results, shown in Figures 13a and 13b, demonstrate that visible misalignments are corrected to pixel-level accuracy, and distortions induced by the collimator are effectively removed.

In summary, the geometric calibration ensures that the star tracker receives geometrically accurate stimuli, both for pointwise and extended objects. The results confirm that the calibration method is effective, achieving high accuracy and eliminating distortions.

Radiometric Calibration

To accurately reproduce star tracker images, it is essential not only to replicate the observational point of view but also to ensure that the correct amount of light is collected by the detector. This is achieved through a radiometric calibration procedure. Radiometric calibration establishes the relationship between the screen's digital count (DC) and the quantity of electrons generated on the detector by the emitted light. Using this mapping, a desired radiant flux at the camera can be matched with the appropriate illumination level on the screen. The procedure follows these steps:

1. The screen's spectral emittance is distributed across the red, green, and blue channels and depends on the screen DC. This emittance is measured using a power meter, which maps the screen spectral emittance to the screen DC.



(a) Before calibration (b) After calibration

Figure 13: Comparison of the obtained results before and after calibration of an image of the Moon. The blue circle labels where the Moon should appear.

2. The radiant flux received by the detector when the screen stimulates the camera is the convolution of the camera's spectral response and the screen's emittance. This quantity is referred to as the effective radiant flux density.

3. Using the radiometric rendering engine developed for this study, the spectral emittance of the Moon is computed based on its physical properties (e.g., albedo, morphology, illumination). The rendering engine calculates this emittance given the geometrical and radiometric parameters of the scene

4. The radiant flux density generated by the Moon is the convolution of the camera's spectral response and the Moon's spectral emittance.

5. To achieve equivalent radiometric stimulation, the Moon's radiant flux density is set equal to the effective radiant flux density. This establishes a direct mapping between the screen DC and the desired radiant flux emitted by the Moon.

Further mathematical details and derivations of the radiometric calibration can be found in Reference 20. An example of the resulting mapping is shown in Figure 14.
A dedicated test is designed to validate the proposed radiometric calibration. On the one side, the radiometric consistent rendering engine ABRAM is used to generate radiometrically consistent images of a Lambertian sphere using the star tracker parameters. These images are compared to real acquisitions from the star tracker after radiometric calibration. The output of these validation tests is reported in Figures 15 and 16 for different maximum screen illumination. Indeed, as described by Equation 1, the star tracker's exposure time directly affects the number of pixels read, which determines the maximum screen DC required to avoid saturation. Absence of saturation is critical for trustworthy image comparison, as saturation compromises image fidelity. For validation purposes, a test was also conducted where saturation was present. Here, a star tracker acquisition from observing the Moon under open-sky conditions served as ground truth. This image is compared against a star tracker acquisition reproducing the same open-sky scene in the facility after radiometric and geometric calibration. The fidelity between the two images, particularly in highly nonlinear effects such as blooming and crosstalk, demonstrates the robustness of the radiometric calibration.

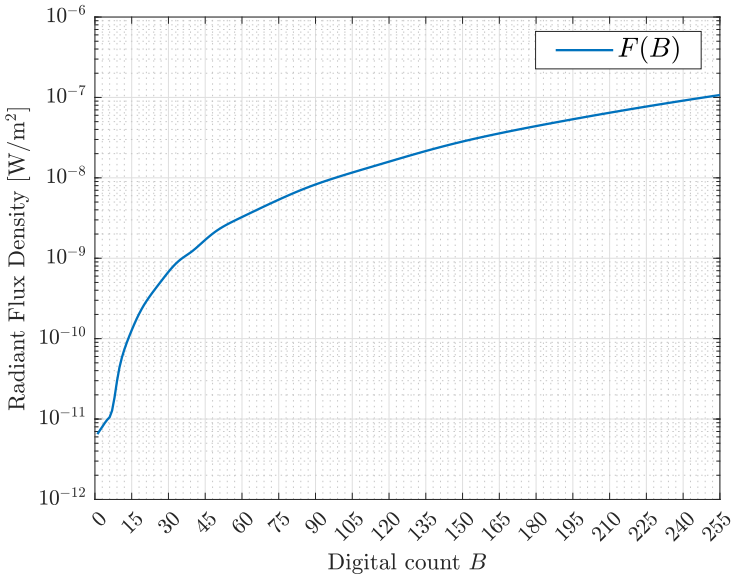
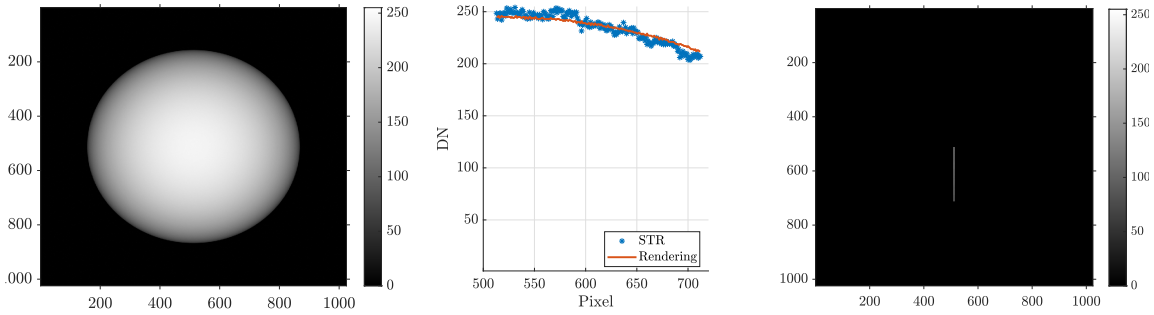
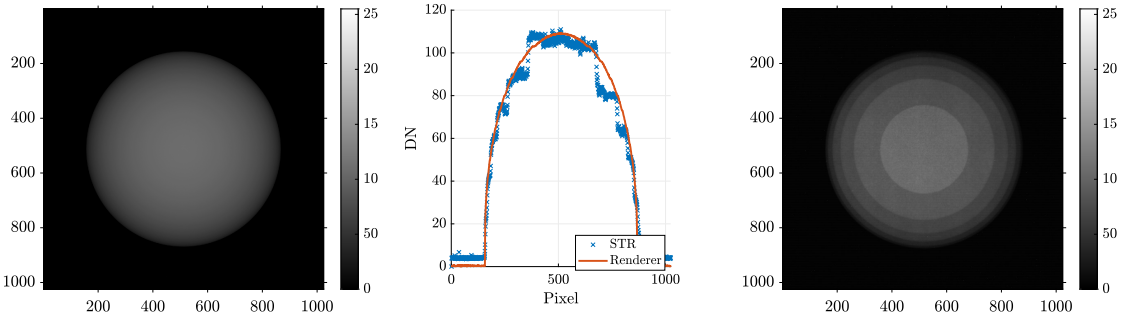


Figure 14: The mapping between the radiant flux emitted by the Moon and the screen DC B.



(a) Theoretical image rendered (b) Comparison (c) Star tracker acquisition

Figure 15: Comparison between the theoretically expected image and the one acquired in the facility after the radiometric and geometric calibration. The maximum screen DC is set to 52 to avoid star tracker saturation.



(a) Theoretical image rendered (b) Comparison (c) Star tracker acquisition

Figure 16: Comparison between the theoretically expected image and the one acquired in the facility after the radiometric and geometric calibration. The maximum screen DC is set to 13 to enable the acquisition of a full-frame image without saturation. The step behavior of the image acquired in the facility is related to the discretization of the screen intensity level.

(a) Real image (Credits: Leonardo SpA). (b) Image acquired in the facility

Figure 17: Comparison between radiometric content of a real image from the star tracker and a reproduced one in the facility

Scaling between exposure time and albedo

After implementing the calibration procedure, a hardware limitation prevented the proper execution of the HIL test. Specifically, no commercially available screen can emit sufficient light to reproduce the effective radiant flux of the Moon. This limitation has also been observed in previous studies involving Jovian moons.[15] As proposed in Reference 15, the solution in this work involves leveraging the relationship between the Moon's albedo and the star tracker's exposure time in the radiometric budget (e.g., see Reference 21). Since these two factors are multiplied in the radiometric equation, it is possible to artificially reduce the Moon's albedo in the renderer to match the screen's maximum emission capability, while increasing the star tracker's exposure time proportionally. This ensures that their product remains constant. To validate this approach, a test was conducted in the facility. Lambertian disks with increasing albedo were displayed on the screen, and the exposure time was adjusted accordingly. The histograms of the resulting star tracker images are shown in Figure 18, demonstrating the perfect equivalence of images that adhered to the albedo-exposure-time scaling. Notably, a control sample that deliberately violated this scaling was included, and its histogram does not align with the others.
Finally, an end-to-end test was performed to validate the combined processes of geometric calibration, radiometric calibration, and albedo-exposure-time scaling. An image was generated using the radiometrically consistent renderer with an exposure time of 0.015 milliseconds and compared to an HIL acquisition with a scaling factor of 13,333. The results, shown in Figures 19a and 19b, highlight the remarkable similarity between the two images in terms of both pixel content and geometrical alignment.

Validation of rendering pipeline and implications

Once all the calibrations and the scaling are developed, the HIL setup is ready to stimulate the star tracker as in orbit. These procedures ensure the star tracker acquisition is precise in terms of both pixel content and observational geometry. Consequently, it becomes possible to cross-validate the saturated rendering pipeline against real acquisitions. In essence, the closest acquisition to the one that the star tracker would capture in orbit is obtained in the HIL facility. The validation results are shown in Figures 20a and 20b. Visual inspection reveals that the developed rendering pipeline does not align with the star tracker acquisition. This

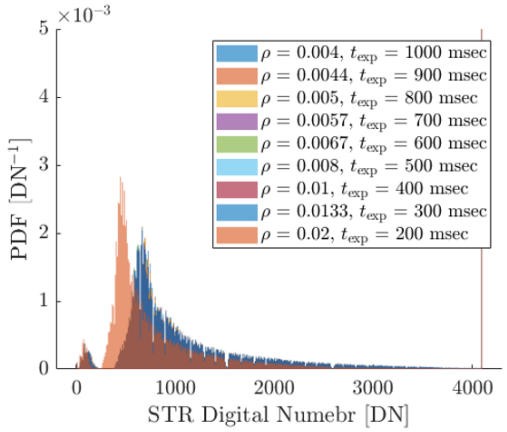


Figure 18: Comparison of image histogram when changing the albedo and the exposure time by keeping their multiplication constant. Note the only histogram not aligned with the other is the one not verifying the constancy hypothesis. It has been added as a control sample in the test.

discrepancy likely stems from inaccuracies in modeling the detector under extreme saturation conditions, such as crosstalk, smearing, and blooming.
The image acquired in the HIL testbed (Figure 20b) highlights that no useful navigation information can be derived from the star tracker under its current configuration. The inability to reduce the exposure time and avoid saturation poses a significant challenge. However, by leveraging the ability to simulate various fictitious exposure times by adjusting the Moon's albedo and the star tracker's exposure time, the study has identified several potential modifications to improve the acquisition of well-exposed images. These modifications are explored through four distinct scenarios:

1. A scenario considering no modification of the star tracker, where the full frame can be read at an exposure time of 200 milliseconds.

2. A scenario envisioning images acquired with an exposure time of 7.7 milliseconds. This value is gathered by studying the robustness of the image processing to detector saturation. As the current star tracker cannot have such a low exposure time, this test case imagines a firmware modification of the ASIC gate array to reduce the exposure time.

3. A scenario considering the lowest exposure time obtainable with a firmware modification of the star tracker (i.e., 0.18 milliseconds).

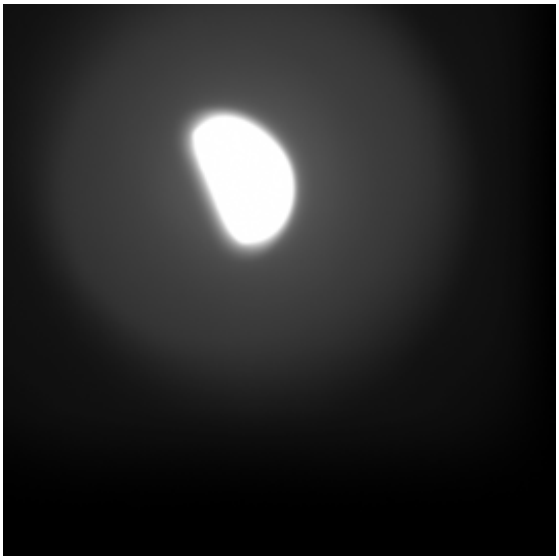
4. A scenario of a perfectly exposed Moon image, acquired by the star tracker. This implies an exposure time of 0.015 milliseconds. This exposure time can be achieved with hardware modification of the star tracker (e.g., neutral filter attenuation or pupil size reduction). This test case highlights how PSF effects have no impact on the limb measurement performance, and, therefore, that with the correct exposure time, star trackers can indeed be used as navigation sensors.

It is important to note that Previous analyses relied on synthetic images, which do not accurately represent star tracker acquisitions. To address this, the HIL simulations used a two-step approach. First, a reference dataset was generated in the HIL facility using the proposed fictitious exposure times. This dataset was used to tune the image processing filter via Monte Carlo analysis on a desktop computer. Second, single VBN chain execution was conducted in the HIL setup, ensuring that the processing chain was fully executed on the star tracker's computing unit while handling real measurements. Examples of images for each scenario are presented in Figures 21a–21d.

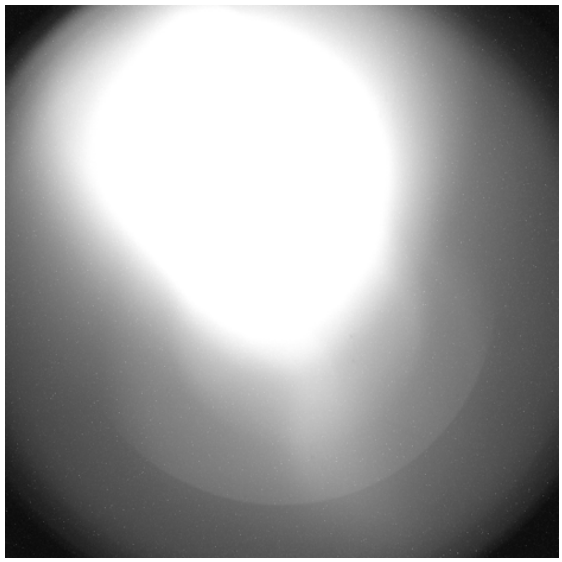
(a) Image rendered with a radiometric consistent renderer with 0.015 milliseconds of exposure time.

(b) Image acquired by the star tracked scaling the albedo of 1/13333 and the exposure time of 13333.

Figure 19: Comparison between a rendered image with a given exposure time and a reproduced one in the facility exploiting the proposed scaling.



(a) Synthetic image.



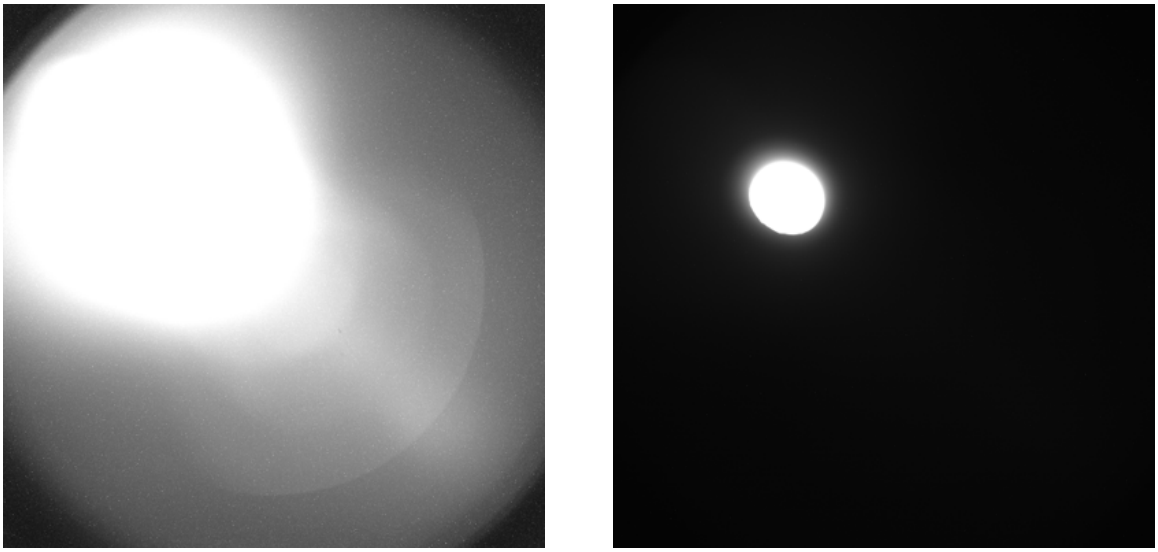
(b) Image acquired in the HIL setup.

Figure 20: Validation of synthetic image against star tracker image acquired in the HIL facility.

HARDWARE-IN-THE-LOOP RESULTS

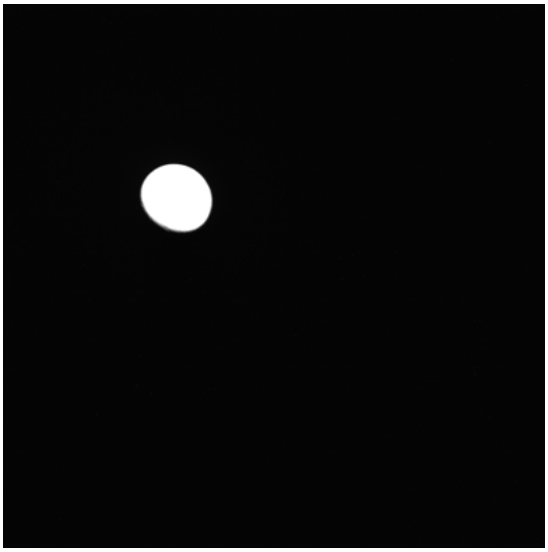
Despite the four scenarios have been thoroughly investigated, only the results for 7.7 and 0.015 milliseconds are presented for the sake of paper brevity. The main reason behind this choice is that:

1. The simulation with 200 milliseconds experiences severe degradation, implying poor VBN performance.

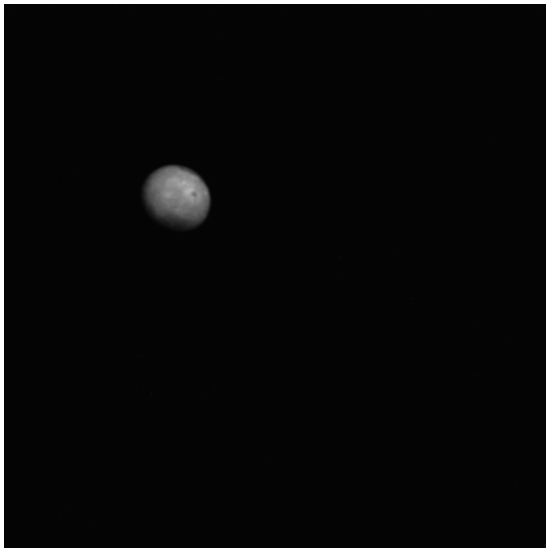


(a) 200 milliseconds

(b) 7.7 milliseconds



(c) 0.2 milliseconds



(d) 0.015 milliseconds

Figure 21: Comparison between well-exposed and saturated images rendered with the proposed pipeline.

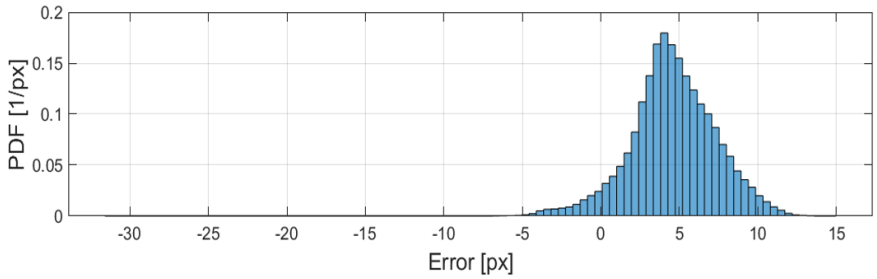
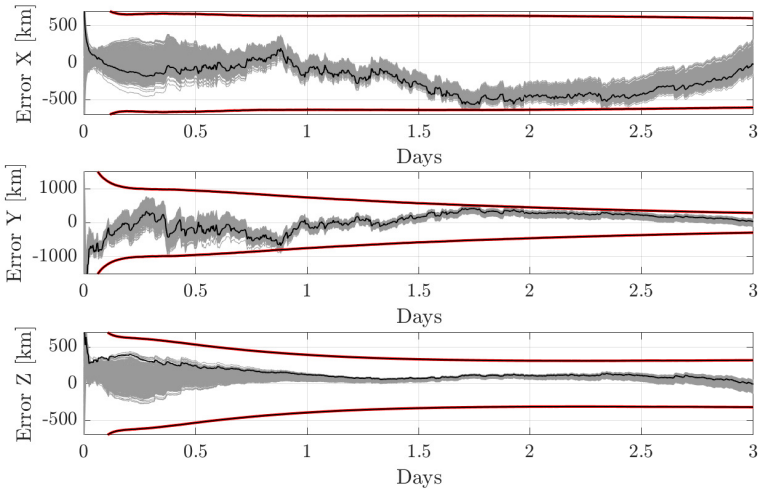
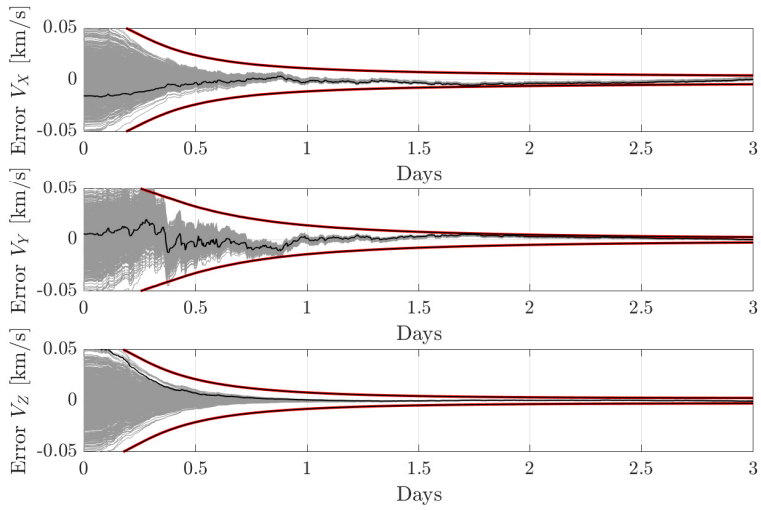


Figure 22: Probability Density Function (PDF) of the edge locations' error with respect to the true limb for HIL simulations for the 7.7-millisecond scenario.

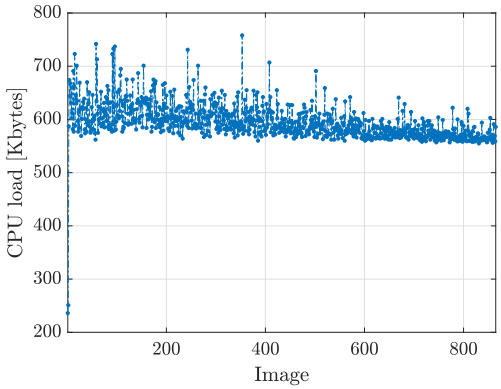


(a) The position error in the inertial reference frame.

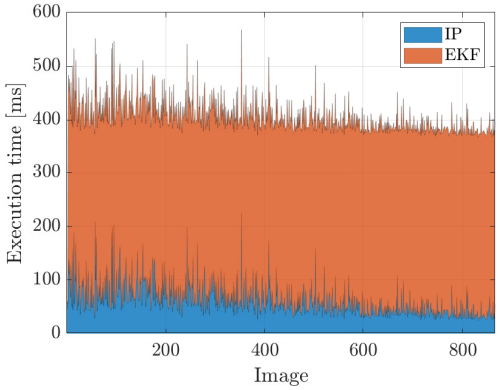


(b) The velocity error in the inertial reference frame.

Figure 23: Results for HIL simulations for the 7.7-millisecond scenario. The black line is the simulation executed on the star tracker during the HIL acquisition.



(a) CPU load.



(b) Execution time.

Figure 24: Telemetries from the star tracker for the 0.015-millisecond scenario.

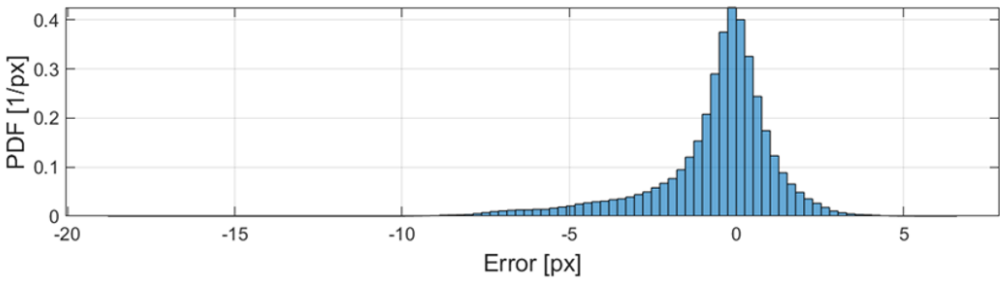
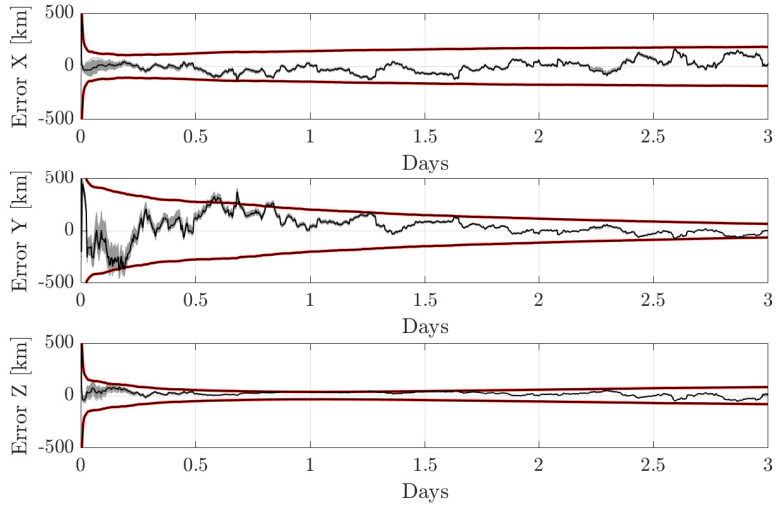
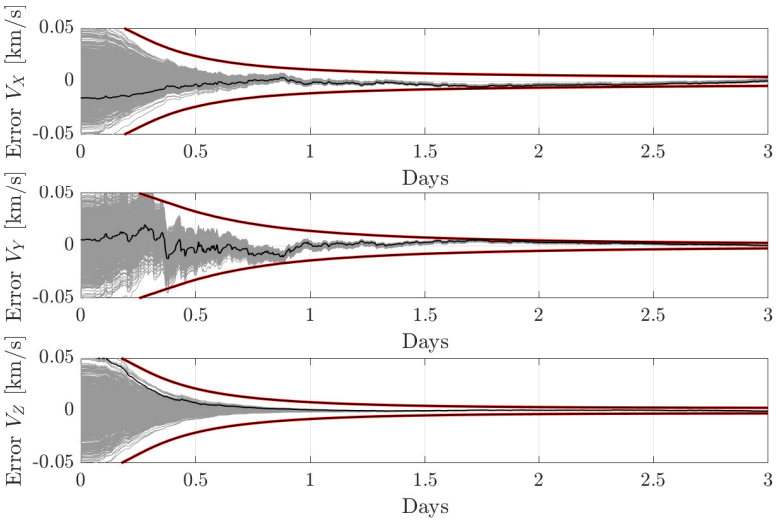


Figure 25: Probability Density Function (PDF) of the edge locations' error with respect to the true limb for HIL simulations for the 0.015-millisecond scenario.

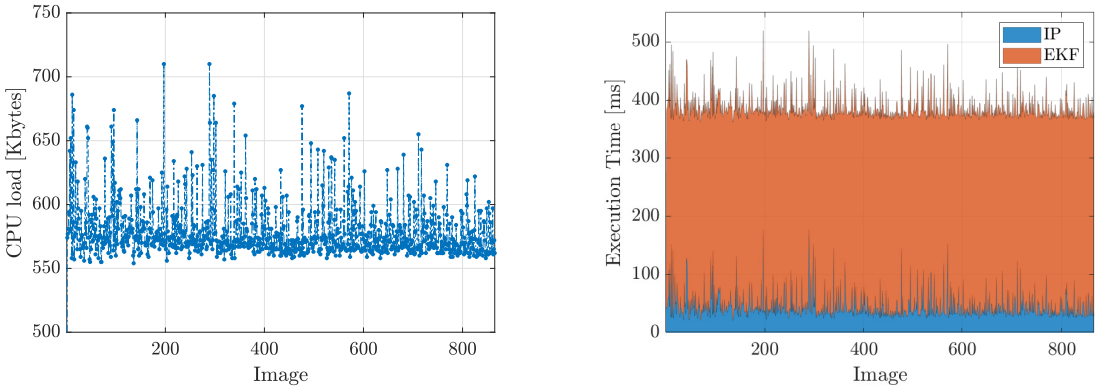


(a) The position error in the inertial reference frame.



(b) The velocity error in the inertial reference frame.

Figure 26: Results for HIL simulations for the 0.015-millisecond scenario. The black line is the simulation executed on the star tracker during the HIL acquisition.



(a) CPU load.

(b) Execution time.

Figure 27: Telemetries from the star tracker for the 0.015-millisecond scenario.

2. The navigation performance of 0.2 and 0.015 milliseconds are extremely similar because the limb is determined at pixel accuracy in both scenarios.

Scenario with 7.7 milliseconds

The first scenario considers a fictitious exposure time of 7.7 milliseconds. An example of the resulting image is shown in Figure 21b. This exposure time represents the threshold at which the image processing algorithm remains robust to saturation, as identified in a separate analysis not detailed here for brevity. In Figure 21b, saturation is evident near the illuminated limb of the Moon. This saturation introduces a bias of approximately 5 pixels in the edge detection process, as illustrated in Figure 22. This bias arises because saturation causes the detected limb location to deviate from its true position, resulting in a loss of information that cannot be recovered or compensated. The edge detection results are subsequently processed by the EKF, which estimates the spacecraft's state, as depicted in Figures 23a and 23b. Although the position estimate exhibits some bias, the EKF partially mitigates this through the Gauss-Markov process, which compensates for the systematic error. The EKF demonstrates good convergence, with the navigation solution remaining bounded within the filter's covariance estimates. The final spacecraft state estimates achieve less than 1% error across all components. Figures 24a and 24b show the memory usage and algorithm execution time for this scenario. The execution time is split between image processing (blue) and EKF computation (orange), with the combined total remaining below 500 milliseconds. Memory allocation and the computational time are low, showing the efficient performance of the VBN solution.

Scenario with 0.015 milliseconds

The second scenario evaluates a fictitious exposure time of 0.015 milliseconds. At this exposure time, the Moon is well-exposed, and the image does not exhibit any saturation effects, as shown in Figure 21d. This ensures accurate limb detection in the image, leading to significantly improved results. As shown in Figure 25, the absence of saturation results in no bias in the edge position error distribution. The reconstructed limb achieves pixel-level accuracy, consistent with the expectations of the developed algorithm. Further refinement to subpixel accuracy could be achieved by integrating a more advanced edge refinement procedure, such as the subpixel-accuracy limb-based VBN algorithm from Reference.[5] These improvements are directly reflected in the EKF's performance, as demonstrated in Figures 25a and 25b. The VBN solution exhibits no bias, and the filter achieves tighter covariance bounds, converging with an error below 0.5% in all the state components. Figures 27a and 27b highlight the low resource requirements of this scenario. Both memory usage and execution time remain minimal, confirming the lightweight nature of the VBN solution

CONCLUSIONS AND LESSONS LEARNED

This study explored the potential use of Leonardo's AA-STR MK II star tracker for navigation in mid- and close-range scenarios. It revealed that the current configuration of the star tracker is unsuitable for navigation due to severe image saturation. The selected star tracker was originally designed to observe faint stars, whereas celestial objects commonly used for navigation, such as the Moon or asteroids, are significantly brighter. Consequently, images from the star tracker exhibit high levels of blooming, saturation, and ghosting. This prevents the identification of critical features, such as the Moon's limb or asteroid surface details, resulting in poor navigation algorithm performance.

The primary limitation lies in the combination of the star tracker's fast optics and its minimum exposure time, which produce severely overexposed images. However, this work developed a HIL setup that ensures the geometric and radiometric consistency of the optical stimulation. This setup allowed critical issues with the star tracker's navigation capabilities to be identified and enabled the simulation of fictitious shorter exposure times to assess navigation performance under improved conditions. As a result, the pixel-accuracy, limb-based navigation algorithm achieved TRL 5 through a rigorous validation process, including MIL, PIL, and HIL simulations in progressively complex environments. Notably, contrary to initial expectations, the study showed that aberrations and defocusing caused by the star tracker's optics did not present major challenges for the extraction of a useful navigation information. While they slightly reduced the VBN algorithm's performance, they did not compromise its overall applicability.
This study highlighted several crucial insights for designing and validating VBN algorithms. First, accounting for hardware (i.e. sensor and processor) performance is fundamental during all phases of the algorithm design. Second, optical sensors are influenced by both geometric and radiometric factors, making it essential to address these during algorithm development and testing. For example, the high saturation observed in the star tracker must be understood, accurately modeled, and reproduced. Third, HIL simulations are indispensable for identifying effects that are too complex to model with available rendering engines. These insights underscore the importance of integrating hardware-aware considerations, comprehensive geometric and radiometric calibration, and robust HIL testing into the development pipeline. Such practices are essential for the effective design, validation, and deployment of VBN algorithms, enabling the early identification and resolution of potential issues.

ACKNOWLEDGMENTS

This work has been conducted under ESA Contract No. 4000139932/22/NL/CRS within the General Support Technology Programme (GSTP) through the support of the national delegation of Italy (ASI). P.P., F.O. and F.T are sponsored by 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). The authors would like to acknowledge the support received by Simone Becucci, Luca Cappabianca, Damiano Macchi, Andrea Sica and Marcella Belcari in the STAR Nav study.

REFERENCES

[1] J. L. Crassidis, "What Is Navigation?," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 5, 2022, pp. 792–794.

[2] M. Morresi, R. Bettarini, D. Procopio, and G. Berrighi, "AA-STR MKII: DEVELOPING A NEXT GENERATION AUTONOMOUS STAR SENSOR," *Proceedings of the 44th Annual American Astronautical Society Guidance, Navigation, and Control Conference, 2022*, Springer, 2024, pp. 233–234.

[3] P. Panicucci, F. Piccolo, A. Rizza, G. Merisio, F. Topputo, and R. Walker, "Vision-Based Navigation for the LUMIO CubeSat Mission," *46th AAS Guidance, Navigation and Control Conference*, 2024, pp. 1–20.

[4] A. Pellacani, M. Graziano, M. Fittock, J. Gil, and I. Carnelli, "HERA vision based GNC and autonomy," *Proceedings of the 8th European Conference for Aeronautics and Space Sciences. Madrid, Spain, 1-4 july 2019*, 2019.

[5] C. Balossi, F. Piccolo, P. Panicucci, M. Pugliatti, F. Topputo, F. Capolupo, *et al.*, "Moon Limb-Based Autonomous Optical Navigation Using Star Trackers," *46th AAS Guidance, Navigation and Control Conference*, 2024, pp. 1–19.

[6] F. Piccolo, C. Balossi, P. Panicucci, M. Pugliatti, F. Topputo, F. Capolupo, *et al.*, "Resource-Constrained Vision-Based Relative Navigation About Small Bodies," *46th AAS Guidance, Navigation and Control Conference*, 2024, pp. 1–18.

[7] P. Panicucci and F. Topputo, "The TinyV3RSE Hardware-in-the-Loop Vision-Based Navigation Facility," *Sensors*, Vol. 22, No. 23, 2022, p. 9333.

[8] M. Pugliatti, C. Buonagura, and F. Topputo, "CORTO: The Celestial Object Rendering TOol at DART Lab," *Sensors*, Vol. 23, No. 23, 2023, p. 9595.
[9] M. Pugliatti, F. Piccolo, A. Rizza, V. Franzese, and F. Topputo, "The vision-based guidance, navigation, and control system of Hera's Milani Cubesat," *Acta Astronautica*, Vol. 210, 2023, pp. 14–28.
[10] A. Pizzetti, P. Panicucci, and F. Topputo, "A Radiometric Consistent Rendering Procedure for Planets and Moons," *4th Space Imaging Workshop*, 2024.
[11] P. Panicucci, F. Piccolo, S. Borgia, A. Rizza, V. Franzese, F. Topputo, *et al.*, "Current status of the lumio autonomous optical navigation experiment," *12th International Conference on Guidance, Navigation & Control Systems (GNC) and 9th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2023, pp. 1–15.
[12] J. A. Christian and S. B. Robinson, "Noniterative horizon-based optical navigation by cholesky factorization," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 12, 2016, pp. 2757–2765.
[13] G. Rufino and A. Moccia, "Laboratory test system for performance evaluation of advanced star sensors," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 200–208.
[14] M. A. Samaan, S. R. Steffes, and S. Theil, "Star tracker real-time hardware in the loop testing using optical star simulator," *Spaceflight Mechanics*, Vol. 140, 2011.
[15] P. Regnier and J. Grzymisch, "JUICENA Guidance NAvigation & Control Sub-System: overall in-flight performance and Vision-Based Navigation on-ground validation," *46th Annual AAS Guidance, Navigation and Control Conference*, 2024.
[16] C. Beierle and S. D'Amico, "Variable-magnification optical stimulator for training and validation of spaceborne vision-based navigation," *Journal of Spacecraft and Rockets*, Vol. 56, No. 4, 2019, pp. 1060–1072.
[17] F. Ornati, P. Panicucci, E. Andreis, and F. Topputo, "RETINA: a highly-versatile optical facility for camera-in-the-loop testing of spaceborne Vision-Based Sensors," *46th Annual AAS Guidance, Navigation and Control Conference*, 2024.
[18] M. Pugliatti, V. Franzese, P. Panicucci, and F. Topputo, "TINYV3RSE: The DART Vision-Based Navigation Test-bench," *AIAA Scitech 2022 Forum*, 2022, p. 1193.
[19] P. Panicucci, F. Ornati, and F. Topputo, "RETINA: a hardware-in-the-loop optical facility with reduced optical aberrations," *arXiv preprint arXiv:2407.02172*, 2024.
[20] F. Ornati, P. Panicucci, A. Pizzetti, and F. Topputo, "A Radiometric Calibration Procedure for Optical Hardware-In-the-Loop Stimulators," *4th Space Imaging Workshop*, 2024.
[21] T. P. Lester, M. L. McCall, and J. Tatum, "Theory of planetary photometry," *Journal of the Royal Astronomical Society of Canada, vol. 73, Oct. 1979, p. 233-257.*, Vol. 73, 1979, pp. 233–257.