Design and testing of star tracker algorithms for autonomous optical line-of-sight deep-space navigation

Casini, Stefano; Cervone, Angelo; Monna, Bert; Visser, Pieter

DOI
10.1364/AO.494586

Publication date
2023

Document Version
Final published version

Published in
Applied Optics

Citation (APA)
Casini, S., Cervone, A., Monna, B., & Visser, P. (2023). Design and testing of star tracker algorithms for autonomous optical line-of-sight deep-space navigation. *Applied Optics*, *62*(22), 5896-5909. https://doi.org/10.1364/AO.494586

applied optics

Design and testing of star tracker algorithms for autonomous optical line-of-sight deep-space navigation

STEFANO CASINI,[1,*] ANGELO CERVONE,[1] BERT MONNA,[2] AND PIETER VISSER[1]

[1] Delft University of Technology, Kluyverweg 1, Delft 2629 HS, The Netherlands
[2] Phosphoenix, Meibergdreef 5, Amsterdam 1105 AZ, The Netherlands
*s.casini@tudelft.nl

Received 3 May 2023; revised 30 June 2023; accepted 3 July 2023; posted 5 July 2023; published 24 July 2023

This paper aims to investigate the capabilities of exploiting optical line-of-sight navigation using star trackers. First, a synthetic image simulator is developed to generate realistic images, which is later exploited to test the star tracker's performance. Then, generic considerations regarding attitude estimation are drawn, highlighting how the camera's characteristics influence the accuracy of the estimation. The full attitude estimation chain is designed and analyzed in order to maximize the performance in a deep-space cruising scenario. After that, the focus is shifted to the actual planet-centroiding algorithm, with particular emphasis on the illumination compensation routine, which is shown to be fundamental to achieving the required navigation accuracy. The influence of the center of the planet within the singular pixel is investigated, showing how this uncontrollable parameter can lower performance. Finally, the complete algorithm chain is tested with the synthetic image simulator in a wide range of scenarios. The final promising results show that with the selected hardware, even in the higher noise condition, it is possible to achieve a direction's azimuth and elevation angle error in the order of 1–2 arc sec for Venus, and below 1 arc sec for Jupiter, for a spacecraft placed at 1 AU from the Sun. These values finally allow for a positioning error below 1000 km, which is in line with the current non-autonomous navigation state-of-the-art.

https://doi.org/10.1364/AO.494586

1. INTRODUCTION

The exploration of deep-space has become one of the most challenging and exciting frontiers in modern science and technology. The ability to navigate and control spacecraft in deep-space environments is crucial for the success of these missions, and the development of precise and accurate attitude estimation and autonomous navigation systems is a key aspect of this challenge. Star trackers have emerged as a promising technology for attitude estimation and autonomous navigation in deep-space due to their high precision and accuracy.

A star tracker is a device that captures images of the star field and uses algorithms to determine the orientation of the spacecraft with respect to an inertial reference frame. The high precision and accuracy of star trackers make them ideal for navigation in deep-space, where GPS signals are not available. In fact, together with attitude, observing visible bodies (e.g., planets, moons, asteroids) with a star tracker allows us to also estimate the spacecraft's state (position and velocity) in a heliocentric reference frame. This technique can be referred to as optical line-of-sight (LoS) navigation, and it is based on feeding a navigation filter with the directions (in terms of azimuth and elevation) of observable bodies, mainly planets; these directions are matched within the filter with the actual positions of the object, which are stored onboard.

The objective of this paper is to review the current state-of-the-art in star tracker algorithms for attitude determination and to design a complete estimation chain that maximizes the performance in deep-space cruising scenarios. The paper focuses on the performance of these algorithms in various deep-space missions. Together with attitude, also the planet-centroiding algorithm is developed, posing particular attention to the illumination compensation (IC) algorithm. Attitude estimation and planet centroiding are intrinsically related in LoS optical navigation because measurements shall enter the navigation filter in the form of azimuth and elevation,

$$az = \arctan \frac{y_{\mathrm{los}}}{x_{\mathrm{los}}}, \tag{1}$$

$$el = \arcsin z_{\mathrm{los}}, \tag{2}$$

where $x_{\mathrm{los}}, y_{\mathrm{los}}, z_{\mathrm{los}}$ are the components of the unit vector pointing from the spacecraft to the observed object. This vector should be computed in a heliocentric frame because within the filter it is matched with the actual LoS, which can be easily computed in the same frame. So, planet centroiding computes the LoS vector in a camera frame, and this information will be converted to a heliocentric frame to be used within the filter. This is done with the rotation matrix computed by solving the attitude estimation,

$$\vec{r}^{\,h}_{\mathrm{los}} = R^h_b \vec{r}^{\,b}_{\mathrm{los}}, \tag{3}$$

where $\vec{r}^{\,h}_{\mathrm{los}}$ and $\vec{r}^{\,b}_{\mathrm{los}}$ are, respectively, the LoS vector in a heliocentric and body (or camera) frame, while R^h_b is the rotation matrix between them, obtained by solving the attitude estimation. Achieving precise azimuth and elevation estimation is of utmost importance in evaluating the effectiveness of LoS navigation. According to findings presented in [1], an error of 1 arc sec in azimuth and elevation lead to positioning errors ranging from 100 km to 400 km, depending on the celestial bodies being observed. Furthermore, this error worsens significantly to approximately 1000 km when the measurement error increases to 10 arc sec. To achieve positioning errors below 100 km, it is necessary to reduce the azimuth and elevation error to 0.1 arc sec. Hence, the significance of robust and performant attitude estimation and planet's centroiding algorithms cannot be overstated for LoS navigation as they directly determine the attainable positional precision.

Then, the algorithms are tested on a developed space image simulator (SIS), implementing the characteristics of specific hardware, and the performance is highlighted.

This work is intended to fill the gap in the literature, as preliminary analysis has shown promising performance for the application of autonomous LoS navigation to real deep-space cruising mission scenarios. In [2], the technique is investigated for its application to the currently under-development mission M-ARGO. The same authors in [3] developed an analytical performance index that quantifies whether an observation geometry is or is not appropriate to achieve good results. In [4], the technique has been tested on a Raspberry Pi to evaluate the computing performance. In [1], the intrinsic properties of the method have been investigated, posing attention to the influence of measurement error on the position and velocity estimation error, showing that—to reach a positioning accuracy in the order of a few hundred meters—the LoS measurement error should not be larger than few arc seconds. However, all of these studies considered out-of-the-loop the actual image processing and observation scenario, whose importance is shown in this paper. In fact, these analyses assumed a constant standard deviation error on the measurement generation, based on star tracker characteristics, which in reality is not the case. First, each planet has a different centroiding estimation error, which is based on its observation characteristics. Second, even the same planet presents different estimation errors as a function of the actual observation scenario. So, this paper is intended to pose the basis to perform image-in-the-loop navigation analysis to further prove the feasibility of exploiting LoS navigation in real missions.

The paper is organized as follows: In Section 2, the SIS is presented, focusing on the needed improvement for LoS navigation analysis. Moreover in Section 2, also the selected hardware (sensor and lens) that is later used for performance analysis is presented. The decision of presenting these two pieces of work at the beginning of the paper is because they are exploited in the definition of appropriate algorithms, even before the complete testing. In Section 3, the complete attitude estimation chain is evaluated and designed. In Section 4, the planet-centroiding algorithm, with IC, is developed. In Section 5, the SIS is used to generate a wide range of observation scenarios for Venus and Jupiter in order to test the full LoS extraction chain. The analysis has been limited to two representative planets allowing proper testing. Finally, in the last section, conclusions are drawn, with particular emphasis on the further steps of the road map for completely deep-space autonomous spacecraft.

2. SPACE IMAGE SIMULATOR

The numerical results presented later in the text are obtained with the help of a SIS that has been developed for this application. While the simulation of stars has been sufficiently consolidated, the simulation of planet images is not. This is especially the case when the planet disks are particularly small (e.g., the case of LoS optical navigation) and when blur is applied (e.g., the case of using star trackers).

A. Star Simulation

The baseline to simulate stars in the field-of-view (FoV) is inspired by [5]. The intensity of the stars (e.g., the number of readout electrons from the sensor) is calculated as a function of the sensor and lens system characteristics and of the visual magnitude of the star. The number of emitted photons is also a function of the given wavelength, and the conversion from photons to electrons is again a function of it since the quantum efficiency (QE) is wavelength-dependent. However, in this simulator, this is left out because the increased complexity would not add much to the analysis. The central V-band wavelength is selected ($\lambda = 555.6$ nm). The number of photons (later converted to electrons) is calculated for a reference star (α-Lyrae, with apparent magnitude 0.03 in the central V-band), and then each other star's photons number is computed by scaling as a function of the magnitude. Going briefly to it, starting with the Planck–Einstein equation, it is possible to express the number of electrons read by the sensor for α-Lyrae as

$$F_{e-} = 3.44 \cdot 10^{-8} \mathrm{QE}\, \lambda\, T_{\mathrm{lens}}\, \pi\, R^2_{\mathrm{lens}} \frac{\mathrm{BW}}{c\,h} \tau, \tag{4}$$

where the constant term is the emitting flux for the star at the chosen λ, QE defines the percentage of photons successfully converted into electrons, T_{lens} is the lens transmissivity that defines the percentage of photons passing through the lens, R_{lens} is the lens aperture radius, BW is the bandwidth, c is the speed of light, h the Planck's constant, and τ is the exposure time.

In star trackers, lens defocusing is done on purpose in order to increase the accuracy of the star centroiding. As stars are at infinite distances, they can be considered point-source light emitters, which implies that their light may be focused on only 1 pixel. This of course limits the accuracy to the resolution of the pixel. It is common practice to blur the light information over several surrounding pixels, and with this approach, the centroiding accuracy can reach a subpixel level. The simulator implements a Gaussian blurring function as

$$I_{e-}(u) = \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{PSF}}} e^{-\frac{u-u_c}{2\sigma^2_{\mathrm{PSF}}}}, \tag{5}$$

where $I_{e-}(u)$ is the distribution function of the electrons as a function of the u-component of the 2D sensor frame (same equation for the v-component), u_c is the u-component star center, and σ_{PSF} is the blurring radius. By integrating this function in both components for each pixel, it is possible to calculate the total fraction of the electrons hitting a particular pixel as a function of the actual center of the stars and the blurring radius. Once the number of electrons for each pixel has been calculated, the conversion to digital number can be approximated as

$$\mathrm{DN} = n_{e-/\mathrm{pixel}} \frac{\mathrm{BitDepth}}{Q_{\mathrm{sat}}}, \tag{6}$$

where $n_{e-/\mathrm{pixel}}$ is the number of electrons per pixel, while Q_{sat} is the pixel saturation capacity, which quantifies the maximum number of electrons that can be read out by the singular pixel. This conversion is based on the assumption that the maximum DN is associated with saturation. This is often not the case, as the maximum DN is usually associated with a slightly lower value. However to simplify the analysis, and as no CMOS datasheet reports this detail, it has been assumed like this.

The star 2D center location is another important parameter that needs to be computed. By assuming a pinhole camera model, the 3D unit vector can be converted into 2D sensor coordinates as

$$u = \frac{u_{\max}}{2} \frac{y}{x} \left(\arctan \frac{a_h}{2} + 1 \right), \tag{7}$$

$$v = \frac{v_{\max}}{2} \frac{z}{x} \left(\arctan \frac{a_v}{2} + 1 \right), \tag{8}$$

where $u_{\max}$ and $v_{\max}$ are the horizontal and vertical pixel resolution, a_h and a_v are the horizontal and vertical FoV, and x, y, z are the three components of the star direction unit vector. The SIS may be further improved by including eventual distortions; however, for this analysis, a complete calibration has been considered in order to focus on the important basic aspects.

B. Planet Simulation

Once the star background has been simulated, attention can be devoted to representing accurately the planet. The approach presented in [6] can be exploited and extended for this purpose. The method is a purely geometric approach, which computes if a certain pixel is lighted by a certain surface point of the planet. This method works very accurately when the apparent angular size of the planet is sufficiently large that the pixel resolution can describe its shape. However, this approach does not provide directly a fraction of light per pixel, and for this reason, when the planet's apparent size is comparable to the size of the pixel, this method does not provide reliable information on how the light is spread over the pixels. For this reason, for the simulation, each pixel is divided into a grid of subpixels. The number of subpixels can vary as a function of the apparent size of the planet. Figure 1 below shows the apparent size in arc sec of the planetary disks having an observer placed at a 1 AU distance from the Sun. This distance has been chosen because this study is particularly interesting for near-Earth asteroid (NEA) exploration, which is defined as those asteroids whose perihelion is below 1.3 AU.

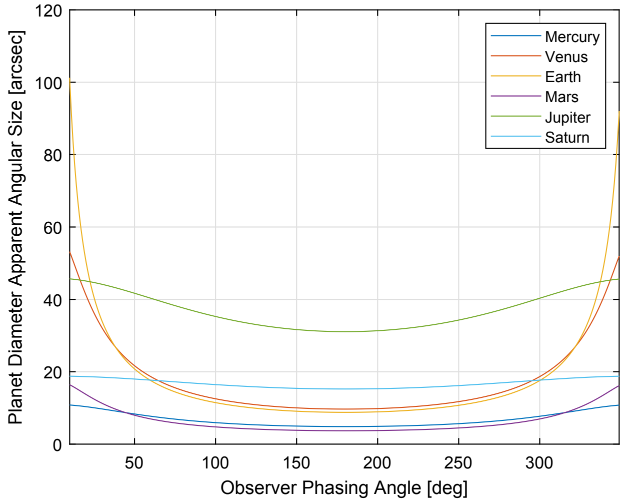
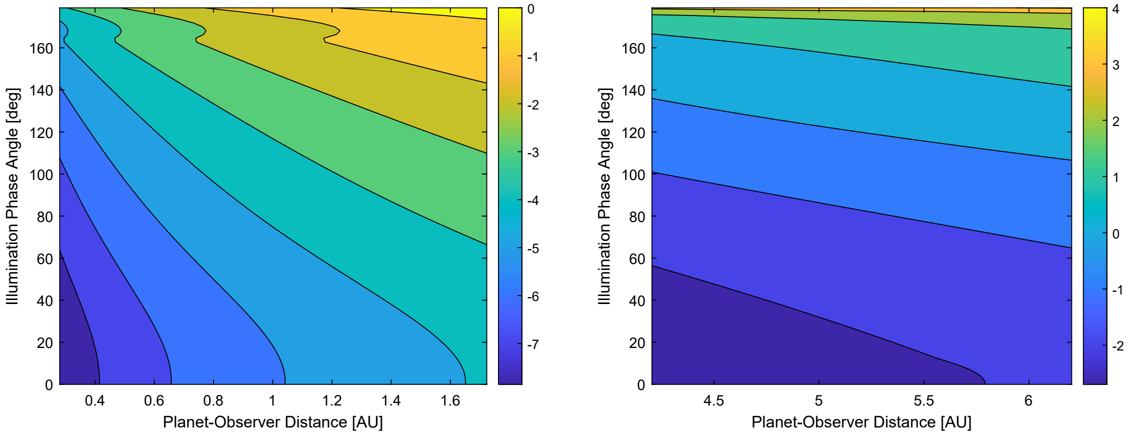


Fig. 1. Planet's apparent diameter for an observer placed at 1 AU distance from the Sun. The angle between spacecraft and planet position vectors is defined as the observer phasing angle.

The subpixel grid will have a dimension that guarantees representation of the shape of the planet with sufficiently good accuracy. However, a large size of the grid impacts the computational length quite significantly. So, a compromise between the accuracy and computational load suggested choosing a grid of 200 × 200 for Venus. As will be shown in the next subsection, the selected hardware has a pixel angular size of 67 arc sec. Since Fig. 1 shows that the minimum observable radius for Venus is approximately 15 arc sec, this selected grid allows us to represent the shape of the planet with at least 45 subpixels, which is reasonable in terms of accuracy. The apparent size of Jupiter is always above 30 arc sec, meaning that a subgrid of 100 × 100 is sufficient to always describe the shape of the planet. With the subpixel grid, it is possible to simulate the illumination condition with sufficiently good accuracy and finally to blur the image. Once the image has been blurred, the subpixels are summed up in order to compute the percentage of total electrons that hits each pixel. This is still not sufficient to calculate the intensity of the pixel. To do so, it is possible to calculate the apparent visual magnitude of the planet following the approach reported in [7], and then, similarly to the process used for stars, this value is converted first into the number of electrons and finally to DN. The basic equation to approximate the apparent magnitude of a planet as a function of the illumination condition is [7]

$$V = 5\log_{10}(rd) + V_1(0) + c_1\alpha + c_2\alpha^2 + \ldots, \tag{9}$$

where r is the planet's distance from the Sun, d is the observer distance, and α is the illumination phase angle. Each planet is characterized by its own coefficients mostly based on Earth on ground observation. As reported in [8], the sum of the terms dependent on the phase angle can be referred to as phase law. Venus apparent magnitude can be computed as



(a)

(b)

Fig. 2. (a) Venus apparent magnitude as a function of observer distance and α. (b) Jupiter apparent magnitude as a function of observer distance and α.

$$V = 5\log_{10}(rd) - 4.384 - 1.044 \cdot 10^{-3}\alpha + 3.687 \cdot 10^{-4}\alpha^2 - 2.814 \cdot 10^{-6}\alpha^3 + 9.938 \cdot 10^{-9}\alpha^4, \tag{10}$$

$$V = 5\log_{10}(rd) + 236.05828 - 2.8191\alpha + 8.39034 \cdot 10^{-3}\alpha^2, \tag{11}$$

where the first equation refers to illumination phase angles below 160°, while the second for larger values.

Similarly, Jupiter's apparent magnitude can be approximated as

$$V = 5\log_{10}(rd) - 9.395 - 3.7 \cdot 10^{-4}\alpha + 6.16 \cdot 10^{-4}\alpha^2, \tag{12}$$

$$V = 5\log_{10}(rd) - 9.428 - 2.5\log 10\left(1.0 - 1.507\frac{\alpha}{180} - 0.363\left(\frac{\alpha}{180}\right)^2 - 0.062\left(\frac{\alpha}{180}\right)^3 + 2.809\left(\frac{\alpha}{180}\right)^4 - 1.876\left(\frac{\alpha}{180}\right)^5\right), \tag{13}$$

where the first equation applies to the illumination phase angles below 12°. Figures 2(a) and 2(b) show the apparent magnitudes of Venus and Jupiter as a function of the illumination phase angle and of the distance planet-observer. The range of distances has been chosen for each of the two planets considering an observer placed at a distance of 1 AU from the Sun. It is important to remark that the star tracker is usually associated with a Sun-exclusion angle, so part of these phase angle-distance combinations is not actually observable. This depends especially on the characteristics of the hardware, and it usually applies on a larger scale to inner planets. The discontinuity in the Venus plot is related to the discontinuity in the equation. This phenomenon is associated with forward scattering by liquid droplets in Venus' atmosphere [7]. Moreover, it has to be remarked that the larger the illumination phase angle is, the lower the precision of the magnitude value. While for Venus it can be considered pretty accurate because a large portion of illumination conditions can be experienced from Earth, the same cannot be said for Jupiter, whose formula is based on observations of Cassini spacecraft and can be considered reliable up to 130° [7]. However, NEA missions would not encounter similar illumination conditions.

The problem of detectability of planets or minor bodies such as asteroids is well treated in [8], and it is a problem intrinsically related to both apparent magnitude and hardware characteristics.

C. Hardware Selection

The SIS needs the characteristics of the sensor and the lenses as input, which define the range of performance of the algorithms later described in the paper. There is a wide range of CMOS sensors available on the market; however, the datasheet is not always complete enough for simulations following the approach described above, especially because of the background noise description. Moreover, space applications require stricter requirements in terms of testing and validation, especially connected to radiation. For this reason, for this analysis and application, the attention has been focused on the TELEDYNE EV76C660 [9] as it is a CMOS sensor developed and tested for space applications. Moreover, its datasheet is particularly detailed and offers noise performance in two different scenarios, which allows us to see how the noise impacts the accuracy of the algorithm. The details of the sensor are reported in Table 1.

Some remarks are necessary. QE has been assumed constant to 0.8. The QE profile as a function of the wavelength is

Table 1. TELEDYNE EV76C660 Main Characteristics

Resolution	1280 × 1024 [pixels]
Pixel Size	5.3 × 5.3 [µm]
Bit depth	10 [bits]
Qsat	8.4 [ke-]
Dark signal at 25C	31 [LSB/s]
Dark signal at 65C	600 [LSB/s]
QE at 400–700 nm	0.6–0.8

reported in the datasheet. For $\lambda > 450$ nm, QE is above 70%, while for $\lambda = 500$ nm, it is around 80%. Regarding noise, the dark signal is usually associated with a combination of factors, some of which cannot be completely modeled. As the majority of dark current is dependent on the thermally induced electrons, this process can be associated with a Poisson distribution. Again, this is only an assumption, and a full characterization of the sensor when it comes to real testing is required. Moreover, the datasheet reports also dark signal non-uniformity (DSNU), which characterizes a different response of every single pixel, related to various aspects such as manufacturing. For example, some rows or columns may be brighter than others. However, this is a complicated value to be translated in a simulation as the mean value reported in a datasheet does not really offer information on how the non-uniformity is distributed across the sensor. For this reason, in this analysis, it is neglected, but if hardware-in-the-loop testing is possible, it will be important to characterize it. Sensors typically have a variety of exposure times, and determining the best value requires some analysis. When dealing with highly dynamic conditions, the setting of exposure time becomes crucial. It must ensure that light information does not spread across numerous pixels, while still providing a signal that can be detected above the noise. In [10], exposure time is optimized to minimize errors in star centroiding under highly dynamic conditions. In more static situations, such as optical navigation during deep-space travel, the exposure time setting is somewhat less critical but can still be optimized. To maximize performance, both the distribution of dark noise and the saturation limit must be taken into account. A standard value of 100 ms was assumed in this analysis as no detailed DSNU distribution was available.

The chosen lens is the MVL16M23 [11] as it is compatible with a sensor with form factor 1/1.8 in. and because it allows us to have a vertical FoV of approximately 20°, which is compliant with the application, as the final goal is to implement these algorithms for deep-space missions. Usually, these are associated with cruising in the same plane as the ecliptic. Implementing a passive navigation strategy is based on avoiding or limiting the reorientation of the satellite to observe visible bodies. So, in order to maximize the observability time length, the camera's boresight direction should lay in the ecliptic plane. Then, a vertical FoV of at least 20° ensures that Venus' elevation is never outside the FoV. In fact, Venus' orbit presents an inclination of 3.39°. Figure 3 shows the evolution of Venus' elevation with respect to the ecliptic, considering an observer coincident with the Earth. Moreover, and this is clear, the larger the FoV is, the larger the number of stars that can be detected, but this will be discussed in the next section.

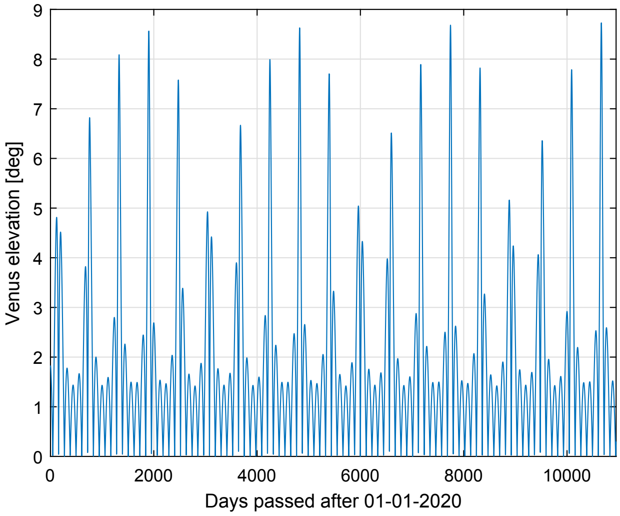


Fig. 3. Venus' elevation angle as a function of time.

Table 2. MVL16MR Characteristics

Focal length	16 [mm]
Aperture diameter	$f/1.4$ [mm]
Transmissivity at 350–750 nm	0.8

The characteristics of the lens system are reported in Table 2. The transmissivity value is averaged in the considered bandwidth.

3. ATTITUDE ESTIMATION CHAIN DESIGN

The design of star tracker algorithms for attitude estimation is a topic that has been widely investigated in recent years. The complete attitude estimation chain can be divided into fundamental intermediate steps: star detection, centroiding, star identification, and rotation matrix estimation. During the star detection process, the 2D image is scanned to look for bright pixels that can be associated with the presence of a star. It is common practice to associate with a pixel the presence of a star when its DN is larger than the mean DN of the dark background plus 5 times its standard deviation [12]. In addition, to reduce the number of operations, a pixel is saved as a candidate star if its DN is larger than its neighboring pixels' DN. However, this is a limitation for those stars that can saturate more than 1 pixel, but they are so few that they can be treated separately. For all the candidate stars, the centroid needs to be computed. Centroiding is the process of analyzing the intensities of pixels belonging to a candidate star to define as accurately as possible the location of the center within the brightest pixel. Once a list of candidate stars' centers has been defined, 2D coordinates shall be associated with an ID of a star. This process is usually done by matching some angular characteristics of star patterns with a precomputed matching catalog, stored onboard. Finally, the rotation matrix between the inertial and body reference frames can be computed by solving the well-known Wahba's problem (WP) [13], which is meant to minimize the following index:

$$J(R) = \frac{1}{2}\sum_{k=1}^{N} ||w_k - Rv_k||^2, \qquad \textbf{(14)}$$

where N is the number of measurements, w_k represents the inertial reference vectors, v_k represents the measured directions in the body frame, and R is the rotation matrix from the body to the inertial frame, which characterizes the attitude of the spacecraft.

In the following subsections, each intermediate step is analyzed in order to define the most appropriate algorithm as a function of the mission scenario. In fact, the deep-space cruising navigation scenario does not require necessarily high-speed algorithms, which are on the other hand needed for attitude determination systems of satellites devoted to Earth observation

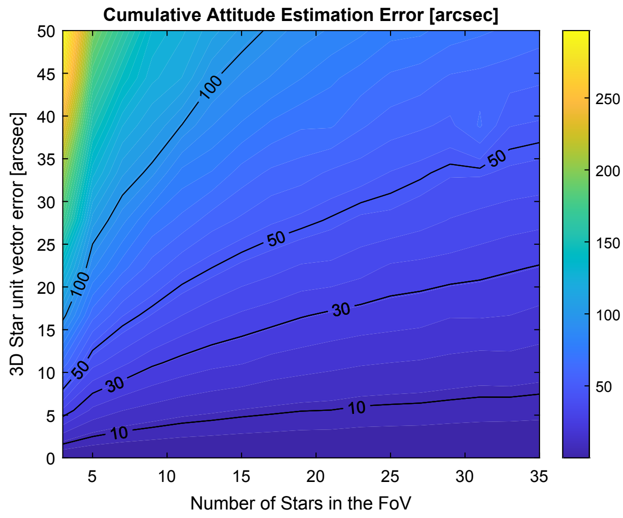


Fig. 4. Cumulative Attitude Estimation error as a function of the number of stars and 3D unit vector error.

or communication. So speed performance can be partially sacrificed if this implies higher attitude estimation performance, which directly reflects in better navigation performance. As shown in [1], the LoS measurement error is the trigger parameter for navigation accuracy. For this reason, for the algorithm selection process, improved accuracy will be the final objective.

A. Centroiding Algorithm Selection

Before diving into the definition of the most appropriate algorithm for computing the center of a star, it is interesting to analyze how different parameters impact the accuracy of the attitude estimation. First, it is possible to relate two important parameters: the accuracy of the 3D unit vector of a star and the number of stars in the FoV. Figure 4 shows the behavior of the cumulative attitude estimation error, defined as the sum of roll, pitch, and yaw angle errors, as a function of the number of stars in the FoV and the error on the 3D unit vector. It can be highlighted that, to obtain a performant attitude estimation system (e.g., well below 50 arc sec), the number of stars should be at least 15, and the angular error is possibly below 5 arc sec.

Concerning the number of stars in the FoV, this is highly dependent on both the actual FoV and the limiting magnitude defined by the hardware characteristics. Figure 5 shows the number of stars in the FoV for an observer placed on the ecliptic with the camera's boresight in the same plane, as a function of the azimuth of the boresight direction, for the vertical FoV of the camera system considered in this paper.

For example, with a limiting magnitude of 6, the minimum number of stars in the FoV is around 15, which is compliant with our previous requirement.

Finally, also the pixel size plays a role, as shown in Fig. 6. Clearly, the lower the pixel size is, the lower the error in the 3D unit vector with the same 2D error in pixel units. As the plot shows for a 5 micron pixel size, it would be optimal to have a centroiding error below 0.05 pixels.

In this work, two centroiding algorithms are analyzed. The first is the standard center-of-gravity (CoG) method. The center is defined as

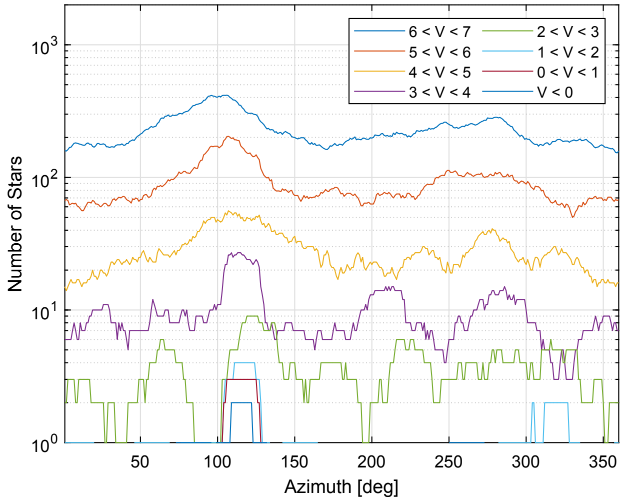


Fig. 5. Number of stars in the camera's FoV as a function of the boresight azimuth angle for different apparent visual magnitudes V.

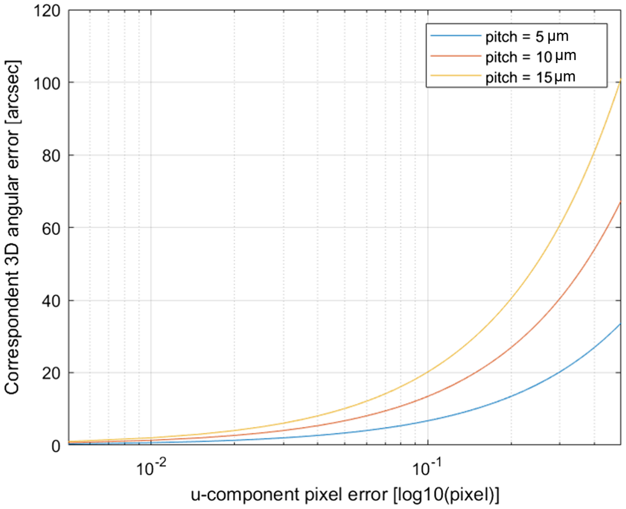


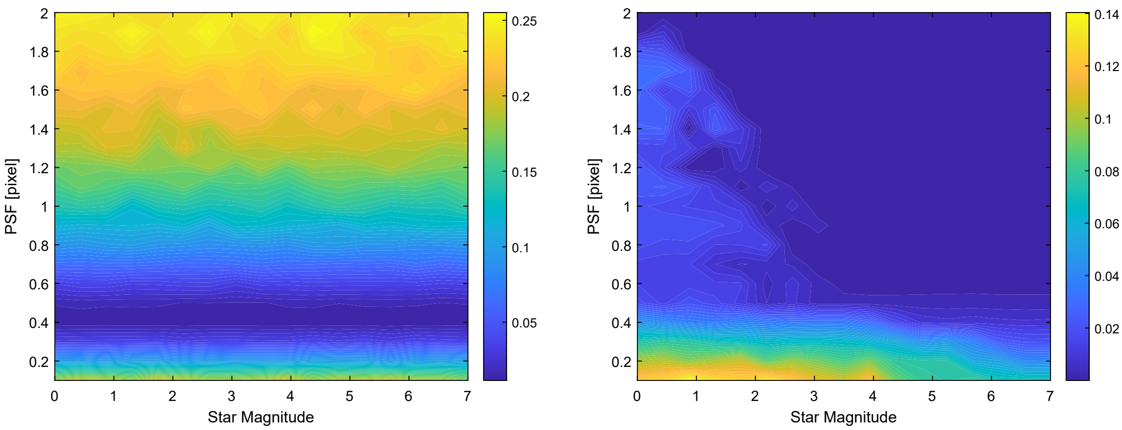
Fig. 6. 3D angular vector as a function of the 2D u-component error in pixels.

$$u_c = \frac{\sum_{k=1}^{n} \sum_{j=1}^{n} I(u_k, v_j) u_k}{\sum_{k=1}^{n} \sum_{j=1}^{n} I(u_k, v_j)}, \tag{15}$$

$$v_c = \frac{\sum_{k=1}^{n} \sum_{j=1}^{n} I(u_k, v_j) v_j}{\sum_{k=1}^{n} \sum_{j=1}^{n} I(u_k, v_j)}, \tag{16}$$

where u_c and v_c are the 2D coordinates, $I(u_k, v_j)$ is the DN of the pixel, u_k and v_j are the pixel coordinates, and n is the size of the region-of-interest (ROI). This is always the quickest method; however, it is usually associated with lower centroiding performance.

The second is a least squares fitting method (LSFM). For each component (u, v), the process is the following. First for each row or column in the ROI, a marginal is defined as



(a)

(b)

Fig. 7. (a) CoG performance as a function of star magnitude and PSF radius. (b) LSFM performance as a function of star magnitude and PSF radius.

$$I_{m,u_i} = \sum_{j=-nb}^{nb} I(u_i, v_j), \tag{17}$$

where $I(u_i, v_j)$ is again the intensity of a given pixel, while nb is the radius of the ROI, considering $I(u_0, v_0)$ as the central pixel. Then, the least squares problem is formulated as

$$S(\beta) = \sum_{i=-nb}^{nb} (I_{m,u_i} - f(u_i, \beta))^2, \tag{18}$$

where $\beta = (a, u_c, \sigma_u)$ is the vector containing the unknown parameter, including the center of the star. The function f corresponds to the Gaussian blurring and can be written as

$$f(u_i, \beta) = a e^{\frac{-(u_i - u_c)^2}{2\sigma_u^2}}. \tag{19}$$

To solve for the center of the star (u_c, v_c), the Levenberg–Marquardt algorithm can be used.

For this analysis, a 3 × 3 ROI has been selected, and as for many star magnitudes, the noise would be predominant in outer pixels, risking the accuracy to be lower. However, larger ROI (5 × 5) in low noise conditions could be an option.

To compare the two centroiding algorithms, the intrinsic properties are explored by removing the background dark current noise and testing the algorithms on the actual number of electrons read by the detector (following the approach defined in Section 2). In this way, also the quantization noise is removed from the analysis.

Figures 7(a) and 7(b) show the error as a function of the intensity of the star and the σ_{PSF} for the two methods. The CoG method has an optimal spreading radius around 0.5 pixel, while LSFM has a larger dependancy on the magnitude. Comparing the two, it is clear that the LSFM offers better centroiding performance, for a wide range of radius, so it is the selected algorithm for the chain. Moreover, for the rest of the analysis, the blurring radius is set to 1 pixel.

B. Star Identification

Star identification is the process of associating an ID to each spotted star in the image so that later on the unit vectors can be compared to their corresponding inertial ones to estimate the rotation matrix. Through the years, this has been sufficiently investigated and consolidated. There is a wide range of identification algorithms, ranging from brightness based, to neural networks, up to the most common ones based on geometric features matching. Surveys of methods can be found in [14,15], comparing the advantages and disadvantages of various algorithms depending on the scenarios where the spacecraft is operating. An example is given by the algorithm proposed in [16], which is characterized by a large number of false stars, so robustness against falsely detected stars is needed. In general, together with robustness, speed and matching catalog size are the parameters that will be investigated to select the most appropriate algorithm. In this application, as remarked in Subsection 3.A, speed is not a priority for the cruising *per se*, but catalog size and robustness are. A triangle-based matching algorithm has been developed because of its robustness and limited catalog size, which is fundamental in this application considering that planetary ephemerides will be stored as well onboard. Another option would have been a Polestar algorithm [17], or one of its derived methods; however, the increase in robustness is not worth a sufficiently larger catalog (as both pattern and geometric characteristics catalogs will be built). Moreover, even if speed is not the main trigger for the selection, the Polestar in the original paper has been proved to be 6 times slower than a triangle matching method based on the original Liebe's algorithm [18]. The triangle matching method exploited in this work is an extension of the algorithm presented in [19]. The method is based on defining three geometrical quantities characterizing a triplet of stars (Fig. 8).

The central star defines an angle γ with its two neighoubouring stars. Then the angular distances between the central star and the two neighoboors can be used. However, despite in the original Liebe's method the 2D projections being used, in [19] it has been found that defining 3D quantities can help the distribution of the catalog to be uniform and being more precise. So Fig. 8 shows the 3D formulation of the star triplets. First, $\sin(\alpha)$ and $\sin(\beta)$ can be defined as

$$\sin(\alpha) = ||\vec{u}_a \times \vec{u}_c||, \tag{20}$$

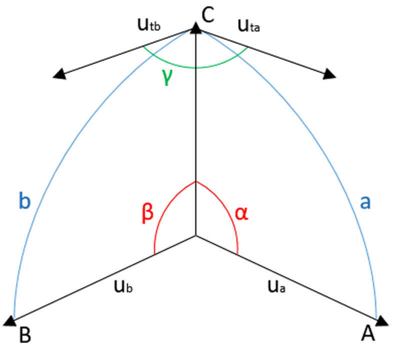


Fig. 8. Stars triplet geometrical features.

$$\sin(\beta) = ||\vec{u}_b \times \vec{u}_c||. \tag{21}$$

Then, the definition of γ is slightly more challenging, and it starts by defining

$$\vec{y}_{ta} = \vec{u}_a - \vec{u}_c(\vec{u}_a \cdot \vec{u}_c), \tag{22}$$

$$\vec{y}_{tb} = \vec{u}_b - \vec{u}_c(\vec{u}_b \cdot \vec{u}_c), \tag{23}$$

and finally

$$\gamma = \text{atan2}(||\vec{u}_{ta} \times \vec{u}_{tb}||, \vec{u}_{ta} \cdot \vec{u}_{tb}), \tag{24}$$

where atan2 in coding language is a variant of the arctan function, whose value is contained between $-\pi$ and $+\pi$. To build the onboard catalog, only stars with minimum brightness 6 have been considered. If magnitude 7 was considered as the limit, the catalog would have been trice larger; however, in the magnitude range 6–7, usually stars are not always recognizable with a larger noise. Moreover, as shown in Fig. 5, there are always at least 15 stars with a magnitude below 6 in the FoV. The other parameter to be set is the number of neighbor stars with which a pattern is computed and recorded in the catalog. Again a compromise for the length of the catalog is to have 12 close stars, which leads to 66 triplets for each star, which leads to a catalog length of 300,894 patterns.

For identification, each triplet of stars can be arranged in six different combinations. All the combinations are compared to the catalog, and for those, the IDs of the stars composing the triplet that has the better match are recorded. To match a pattern γ, $\sin(\alpha)$, and $\sin(\beta)$ should be within a threshold. Then, if multiple patterns are matched, the one with the lowest root mean square (RMS) is recorded as a candidate. Finally, for each candidate star, the assigned ID is the one with more hits, if they are at least three.

C. Wahba's Problem Solver

Once a sufficient amount of stars has been identified, the process of comparing the 3D unit vectors of those stars with their heliocentric ones is used to find the rotation matrix and, therefore, to estimate the attitude. To solve the WP, several algorithms have been presented: Davenport-Q, Quest, singular-value decomposition (SVD), and many more [20]. The SVD method is consolidated as it gives the best possible accuracy, together with numerous other advantages such as its stability and the possibility of providing the expected accuracy.

First, the matrix N is defined as

$$N = BR^T = \sum_{k=1}^{n} v_k w_k^t, \tag{25}$$

which can be decomposed with an SVD as

$$N = U\Sigma V^T, \tag{26}$$

and finally the optimal rotation matrix can be found as

$$R_{\text{opt}} = UMV^T, \tag{27}$$

where $\text{M} = [1, 0, 0; 0, 1, 0; 0, 0, \det(\text{UV})]$.

4. PLANET CENTROIDING

Planet centroiding is the process of determining the center of the planet imaged on the sensor. As shown in the previous section, planets are significantly brighter than the average star, which means that, with the same exposure time, the pixels can saturate more easily. So, counterintuitively, the blurring is helping also planets, despite their discrete apparent size, as the central pixels will be often saturated. For planet centroiding, again two macro-categories of algorithms are possible. A basic CoG algorithm is the easiest to be implemented, while a fitting method, similar to the one implemented for star centroiding, is slightly more difficult. This is because, for point-source light information, a simple relation can be established between the pixel intensity and the center. For a discrete element like a planet, this function is a combination of two functions: the shape function and the blurring function. To have an accurate solution, in the fitting process, the planet should be discretized in a sufficient number of point-source emitters, which makes the algorithm computationally expensive. On the other hand, a CoG method, in combination with the blurring, offers sufficient performance when the full disk is visible, while the error deteriorates in harsh illumination conditions.

To solve this issue, two approaches are investigated: a "brute force" and an analytical one. The brute force method is the least elegant, as it is based on associating to the observation of some correction tables, precomputed based on hardware characteristics that are entered with the illumination phase angle. This method has some limits as the number of scenarios stored onboard is limited and it requires interpolating numbers. Of course, despite being the least elegant, it may be computationally cheaper, and if the hardware is properly characterized, it can provide very good accuracy. Figure 9 shows the correction table for Venus. An option can be to report the correction for three typical distances and then interpolate depending on the actual mission scenario.

On the other hand, it has been possible to derive an analytical formulation for the correction shift as a function of the illumination condition. Looking at a sphere in the 3D frame (Fig. 10), it is possible to write the equation of the terminator line as a function of the illumination angle. This can be done by intersecting the equation of the sphere and the plane perpendicular to the illumination vector. If then this is projected in the observation plane, the equation of the terminator lines is

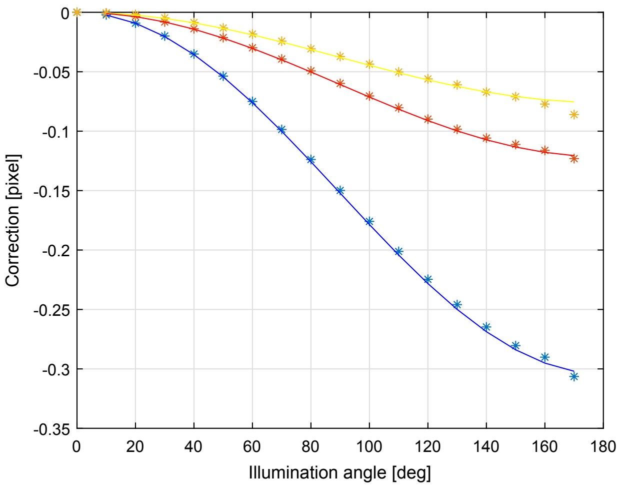


Fig. 9. Venus centroiding correction, simulated results as points, and analytical results as lines. Small distance in blue, medium distance in red, and large distance in yellow.

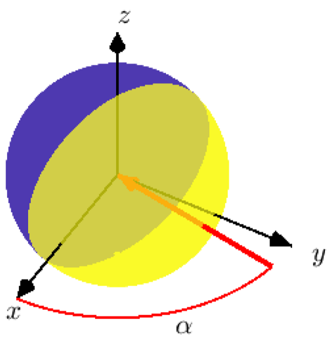


Fig. 10. Visualization of the illumination angle.

$$f_s(x) = \sqrt{R^2 - x^2}, \tag{28}$$

$$f_{is}(x) = \sqrt{R^2 - x^2(1 + \tan\theta^2)}, \tag{29}$$

where R is the sphere radius and θ is the illumination angle. With these equations, it is possible to calculate analytically the center of gravity of the planet as

$$u_{pc} = \frac{\int u f_s \pm \int u f_i s}{\int f_s \pm \int f_i s}. \tag{30}$$

By integrating this equation and simplifying, we reach the following formulation:

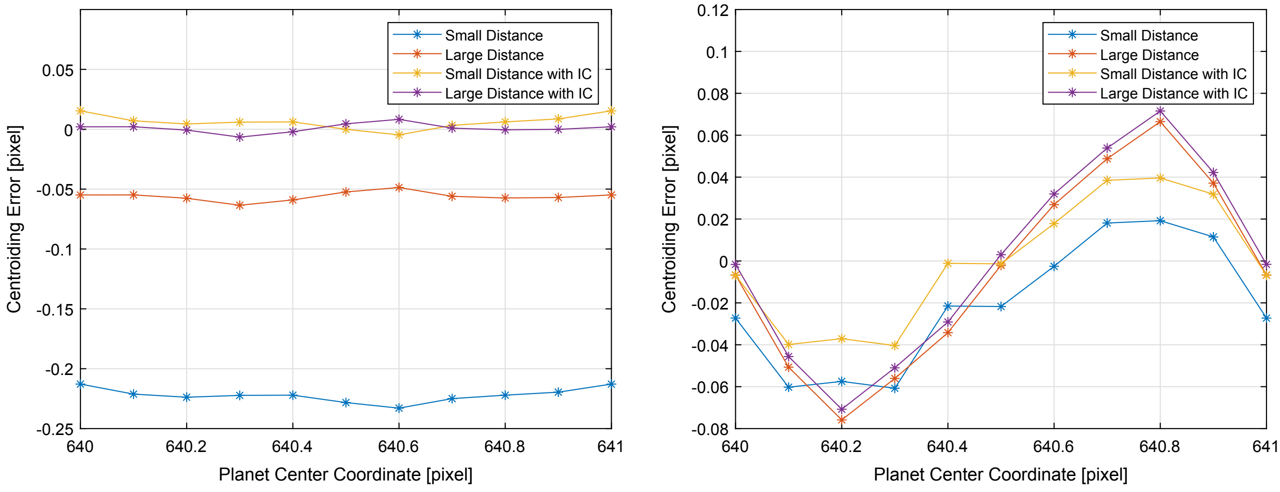
$$u_{\text{shift}} = \frac{+4}{-3} R \frac{\sqrt{c}}{\pi c} \frac{c-1}{\sqrt{c} \pm 1}, \tag{31}$$

with

$$c = \left(1 + \tan\theta^2\right), \tag{32}$$

which gives as output the shift that has to be applied to the computed center of gravity as a function of the apparent radius of the planet and the illumination phase angle. Figure 9 shows the computed shift for the observation of Venus. As can be seen, it matches the behavior of the correction table.

However, before diving into the numerical results, it is important to highlight some aspects. Due to the pixel quantization and pixel saturation, the centroiding algorithm and the IC algorithm are not the only actors playing a role. With the same observation condition, the actual planet's center location within the single pixel influences the distribution of light among the pixels, which in combination with pixel quantization and saturation creates a variable behavior of the centroiding error. As an example, Figs. 11(a) and 11(b) show the behavior of the centroiding error as a function of the pixel center in four observation conditions of Venus. The plots show how the shift in the center's u-component within pixel 640, which is central in the selected sensor, generates a variable centroiding error. Figure 11(a) shows this behavior for small and large distances of the observer for an illumination phase angle of 60°. The oscillation is in the order of 0.03 pixels, but it is important to notice that, despite the oscillation, the IC algorithm keeps reducing the computation error. On the other hand, Fig. 11(b) shows the same for a poorer



(a) (b)

Fig. 11. Venus centroiding error as a function of the center location for illumination phase angle 60° (left). Venus centroiding error as a function of the center location for illumination phase angle 150° (right).

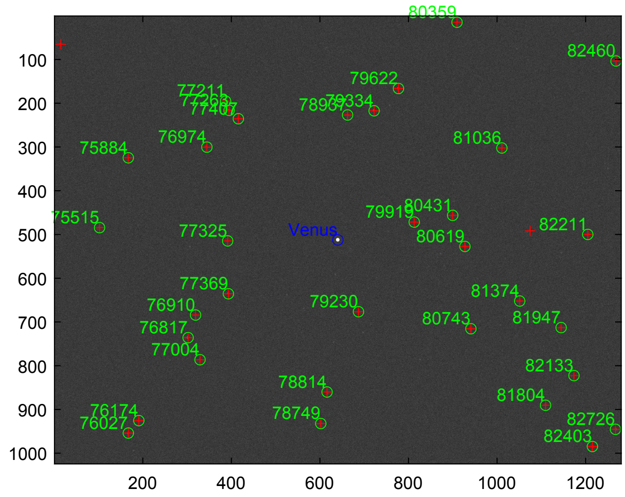


Fig. 12. Venus observation complete chain testing with SIS.

illumination condition with phase angle 150°. In this case, the error is larger and is in the order of 0.08 pixels. In this case, the IC algorithm is only partially able to contain the error in some location; nevertheless, the overall performance is still slightly improved.

These plots are not meant to fully characterize all the possible combinations of behavior but are meant to explain that the actual center of the planet within the pixel might play a role and influence the accuracy of the centroiding as oscillations within 0.05 pixel might occur. This value may seem small, but actually it corresponds to few arc sec, posing a limitation in cutting down below to 0.1 arc sec accuracy. This is, however, an "uncontrollable" parameter with state-of-the-art technologies for CubeSats, as it would require an extremely high pointing accuracy.

5. COMPLETE TESTING

This section is intended to show the results of testing the full LoS extraction chain with the SIS. To do so, the observer has been considered coincident with Earth's position on the first day of each month between 2024 and 2028. As the center within the pixel has been explored in the previous section, the center of the planet has been considered perfectly aligned with the boresight direction of the sensor. The results are presented as either the mean value or standard deviation for a Monte Carlo simulation with 40 trials. First, the results of Venus observation are presented, and then Jupiter. In both cases, high noise and low noise scenarios have been considered, giving priority to the low case. Figure 12 shows the results of a full chain testing for Venus observation.

A. Venus Observation

For the larger noise case, the attitude estimation results are presented in Fig. 13. As was expected, the roll error is significantly larger than the pitch-yaw ones, which are very close to each other. In general, the peaks of the roll error are associated with an average lower number of observed stars.

Regarding the planet centroiding, results are shown in Figs. 14(a) and 14(b). Figure 14(a) shows the mean centroiding error at each observation scenario. This shows how in a large majority of scenarios the IC algorithm works perfectly, while in extremely poor observation conditions it induces a small bias in the centroiding. Figure 14(b) shows that, despite the bias of the IC algorithm, most of the observation scenarios have a σ in the order of 0.002 pixels, while in a few cases, it is 6 times larger. These observation scenarios are, however, associated with an illumination phase angle very close to 180°, which in realistic mission scenarios will not be observed because the Sun-exclusion angle cannot be guaranteed.

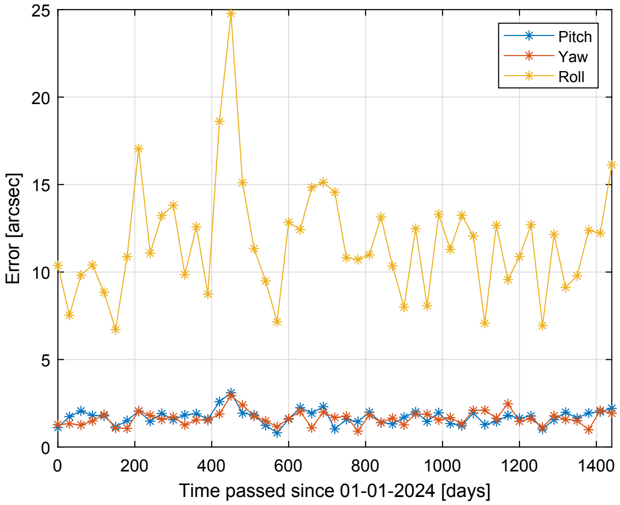
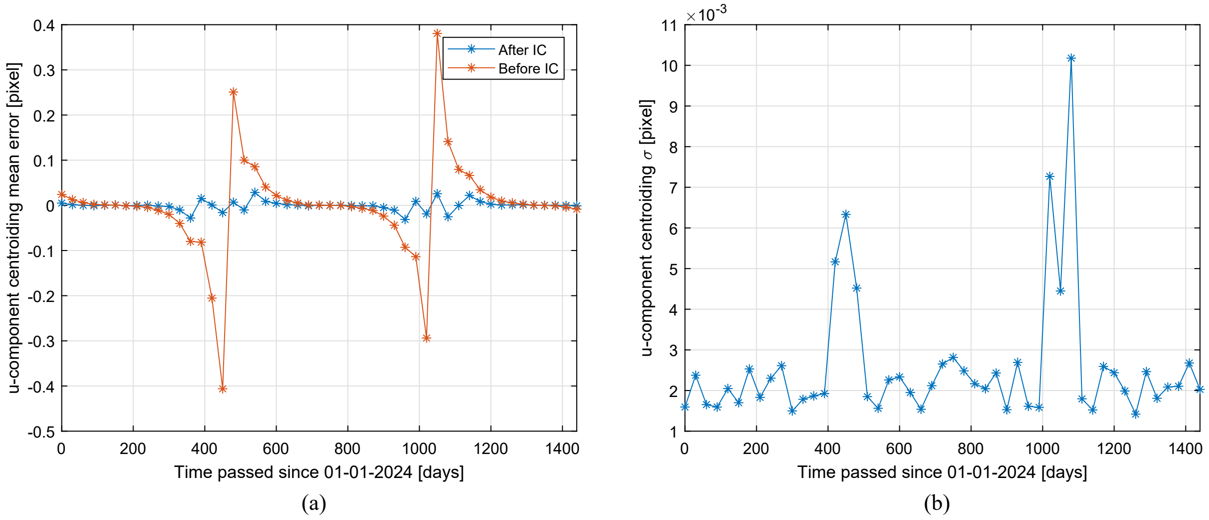


Fig. 13. Attitude estimation standard deviation as a function of the observation time (Venus high noise case).

Remarking that the main goal of this algorithm is to provide the navigation filter with the direction of the object in a heliocentric frame, it is possible to compute the difference between the azimuth and elevation and the computed one. This is because the actual navigation algorithm runs optimally with the measurement fed as azimuth and elevation computed in a heliocentric frame. This allows us also to observe the actual link between attitude estimation and the planet's centroiding. So, Figs. 15(a) and 15(b) show, respectively, the mean and standard deviation of both. As can be noticed, the mean value has a bias in the proximity of poorly illuminated conditions, and the σ have peaks corresponding to the pick of the u-component error. The additional oscillations are then related to the combination of the planet's centroiding and attitude estimation. This again testifies that, for a navigation analysis, the two cannot be completely separated, and assuming a constant σ for azimuth and elevation measurements, it is a limiting approach. Moreover, it is important to remark how, in the vast majority of the cases, both azimuth and elevation have a mean value error between −1 and 1 arc sec, while the standard deviation is between 1 and 2 arc sec. This, according to the results presented in [1], shows that this method is appropriate to achieve position estimation error well below 1000 km.

For the low noise case, as could be expected, the attitude estimation is improved. This is linked both to a better centroiding performance and also to a slightly larger average number of identified stars, as shown in Fig. 16.

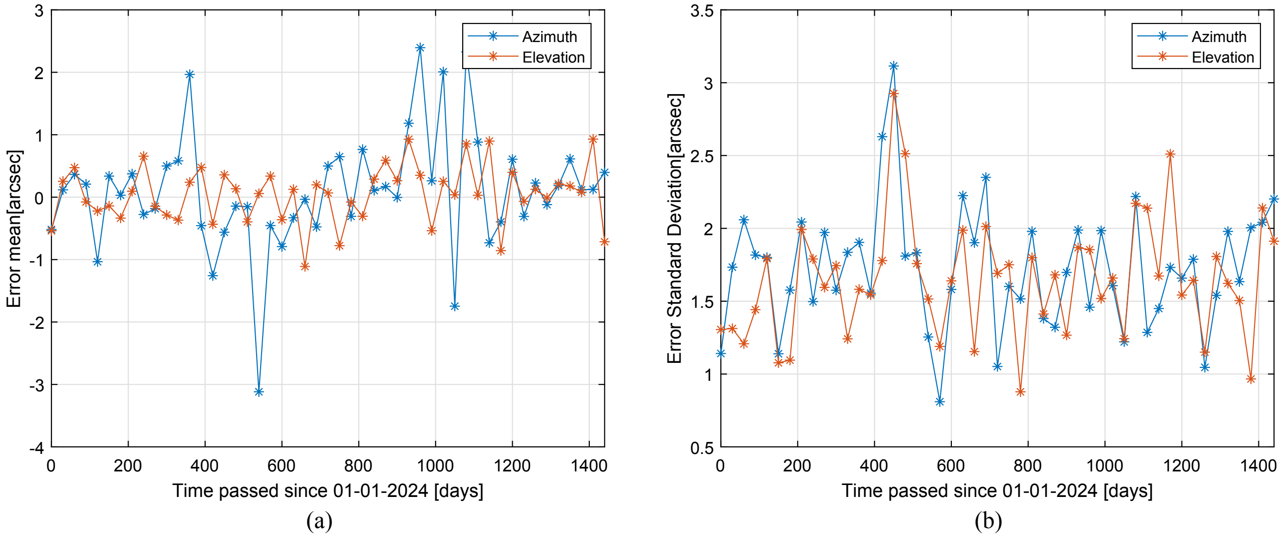
These differences result in a lower roll, pitch, and yaw lower estimation error, as depicted in Fig. 17.



(a)

(b)

Fig. 14. Venus centroiding (a) error mean value and (b) error standard deviation.



(a)

(b)

Fig. 15. Venus azimuth and elevation (a) error mean values and (b) standard deviation.

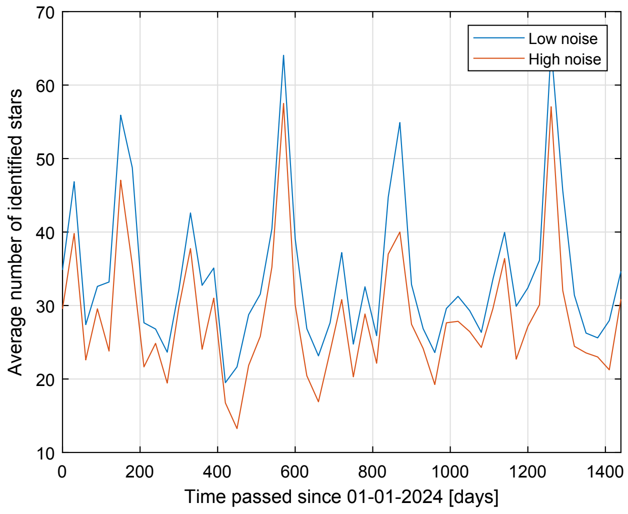


Fig. 16. Average number of identified stars as a function of the observation time for Venus observation test case.

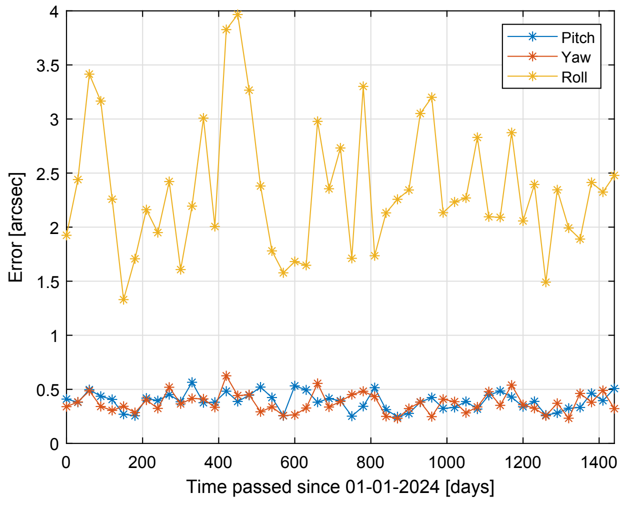


Fig. 17. Attitude estimation standard deviation as a function of the observation time (Venus low noise case).

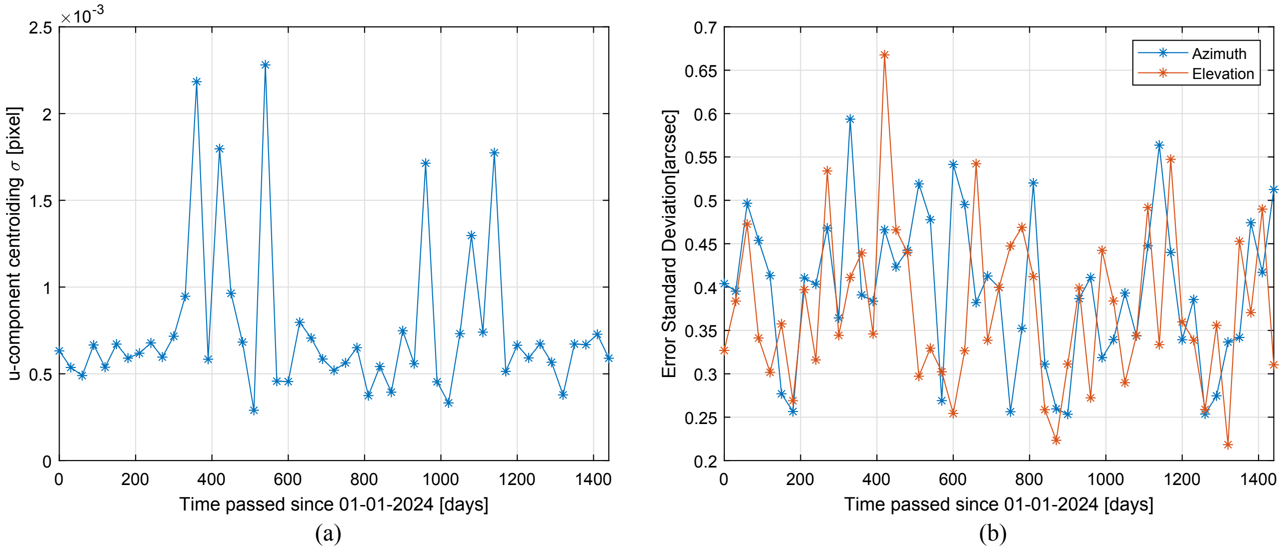


Fig. 18. Venus centroiding error standard deviation (low noise) (a). Azimuth and elevation error standard deviation (b).

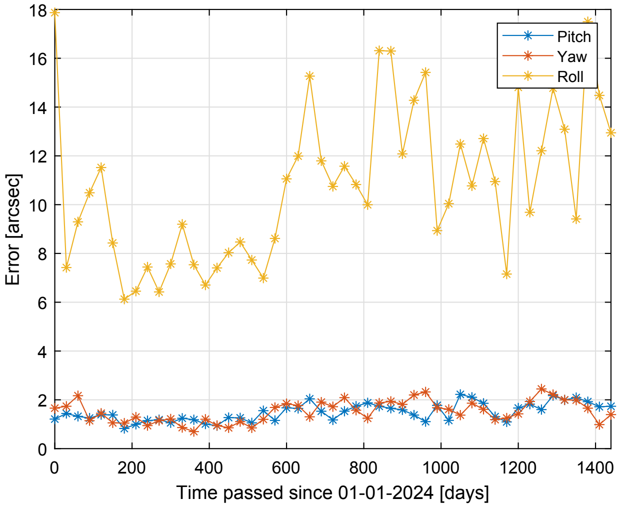


Fig. 19. Attitude estimation standard deviation as a function of the observation time (Jupiter high noise case).

These reflect also in the computation of the centroiding and then the azimuth and elevation of the body direction. As has been described before, in correspondence with poor illumination conditions, there are some centroiding biases, which are almost independent of the noise, and in fact, in this low noise example, they are comparable with the high noise case. On the other hand, the σ is significantly lower, and this is important as, with the exception of a very poor observation scenario, the low noise case produces extremely accurate measurements to feed the navigation filter. Figures 18(a) and 18(b) show the centroiding as well as the azimuth and elevation standard deviation results, respectively.

B. Jupiter Observation

The same approach has been used to exploit Jupiter observation. The attitude estimation error for the high noise case is shown in Fig. 19. Similar values to Venus observation are obviously obtained. It is worth noticing that the first half of the simulation has a higher number of stars in the FoV, which reflects in better accuracy.

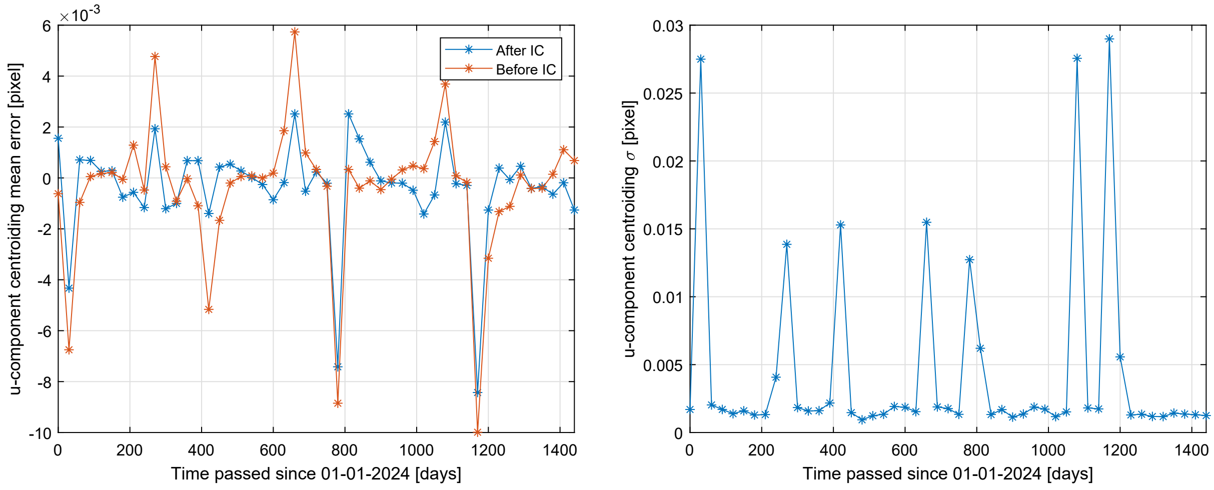
Jupiter's centroiding has better performances because, with an observer placed at 1 AU from the Sun, the observation does not experience particularly poor illumination conditions. The results are presented in Figs. 20(a) and 20(b).

Finally, the computation of azimuth and elevation is again accurate, and results are presented in Figs. 21(a) and 21(b).

Again, similarly to Venus for the lower noise case, performances are improved. Respectively, attitude estimation error, centroiding standard deviation and measurements, standard deviation results are shown in Figs. 22, 23(a), and 23(b).

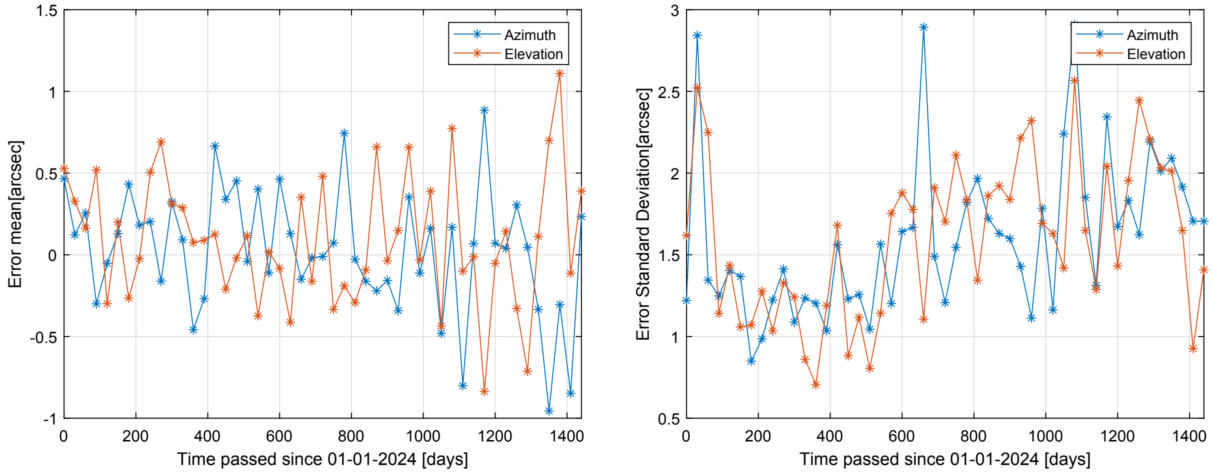
6. CONCLUSION AND FURTHER DEVELOPMENT

This paper presents a complete overview of the simulation, design, and performance evaluation of star tracker algorithms for autonomous LoS navigation. This work can be considered a necessary piece of the road map toward the implementation of this navigation technique in real missions, especially in miniaturized spacecraft where autonomy during all mission phases is fundamental. First, the SIS theory available in literature has been improved in order to target the simulation of small disk planetary objects, which is relevant for LoS navigation in deep-space. Then, the current state-of-the-art attitude determination algorithms have been investigated and selected in order to design an appropriate attitude estimation chain, which maximizes the performance in deep-space cruising scenarios. Then, two approaches for planet centroiding and IC have been proposed. Finally, the complete image processing chain has been evaluated in different relevant scenarios for deep-space, in particular for the observation of Venus and Jupiter. The observer has been placed at 1 AU distance from the Sun to represent NEA missions; however, different distances in other scenarios may slightly affect the error estimation. Results have shown that promising performance can be expected by the LoS



(a) (b)

Fig. 20. (a) Jupiter centroiding error mean value. (b) Centroiding error standard deviation.



(a) (b)

Fig. 21. (a) Jupiter Azimuth and Elevation error mean values. (b) Azimuth and elevation error standard deviation.

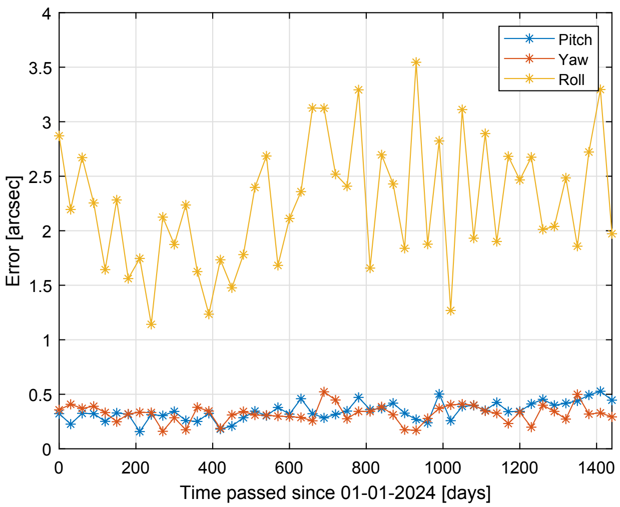
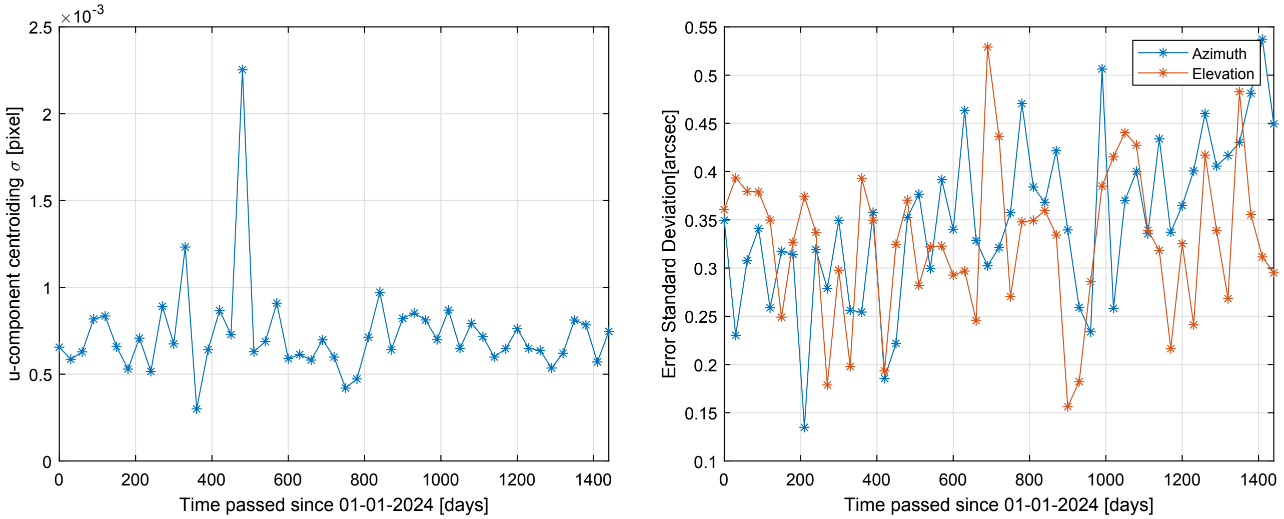


Fig. 22. Attitude estimation standard deviation as a function of the observation time (Jupiter low noise case).

extraction in terms of azimuth and elevation as, in the majority of the illumination observation scenarios, the measurement computation is in the range of 1–2 arc sec for the high noise case and reduces below 1 arc sec in a low noise case. However, it is important to remark that the actual planetary locations within the single pixel, in combination with hardware characteristics such as saturation limit and quantization, do play a role, and they can slightly lower the performance. Moreover, this study has assumed a perfectly calibrated camera, and further analysis needs to be performed to derive the influence of lens distortion when they are not properly taken into account.

In the future, validating the algorithms on actual space images will be needed to finally prove the applicability of this technique in deep-space missions. To conclude, this work can be considered the first stepping stone for a complete navigation analysis, which includes in-the-loop the generation of synthetic images and proves that the reachable accuracy of azimuth and elevation is compliant with the mission navigation requirements for deep-space applications as measurement error in the order of few arc sec would allow positioning error well below 1000 km [1].



(a)

(b)

Fig. 23. Jupiter centroiding error standard deviation (low noise) (a). Azimuth and elevation error standard deviation.

Funding. Horizon 2020 Framework Programme (Grant no. 813644).

Acknowledgment. Support was provided by AAC Hyperion (former Hyperion Technologies B.V.), where author Stefano Casini was affiliated during the Stardust-R frame.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors according to the TU Delft Data Management Plan.

REFERENCES

1. S. Casini, A. Cervone, B. Monna, and E. Gill, "On line-of-sight navigation for deep-space applications: a performance analysis," Adv. Space Res. (2022).
2. V. Franzese, F. Topputo, F. Ankersen, and R. Walker, "Deep-space optical navigation for M-ARGO mission," J. Astronaut. Sci. **68**, 1034–1055 (2021).
3. V. Franzese and F. Topputo, "Optimal beacons selection for deep-space optical navigation," J. Astronaut. Sci. **67**, 1775–1792 (2020).
4. E. Andreis, V. Franzese, and F. Topputo, "Onboard orbit determination for deep-space CubeSats," J. Guid. Control Dyn. **45**, 1466–1480 (2022).
5. M. Marin and H. Bang, "Design and simulation of a high-speed star tracker for direct optical feedback control in ADCS," Sensors **20**, 2388 (2020).
6. J. A. Christian, "Optical navigation for a spacecraft in a planetary system," Ph.D. thesis (The University of Texas at Austin, 2010).
7. A. Mallama and J. Hilton, "Computing apparent planetary magnitudes for the astronomical almanac," Astron. Comput. **25**, 10–24 (2018).
8. V. Franzese and F. Topputo, "Celestial bodies far-range detection with deep-space CubeSats," Sensors **23**, 4544 (2023).
9. Teledyne, "2D CMOS image sensors," 2023, https://imaging.teledyne-e2v.com/products/2d-cmos-image-sensors/ruby-1-3m-ev76c660-ev76c661/.
10. X. Wei, W. Tan, J. Li, and G. Zhang, "Exposure time optimization for highly dynamic star trackers," Sensors **14**, 4914–4931 (2014).
11. Thorlabs, "Mvl16m23," 2023, https://www.thorlabs.com/thorproduct.cfm?partnumber=MVL16M23.
12. C. Liebe, "Accuracy performance of star trackers—a tutorial," IEEE Trans. Aerosp. Electron. Syst. **38**, 587–599 (2002).
13. G. Wahba, "A least squares estimate of spacecraft attitude," SIAM Rev. **7**, 409 (1965).
14. B. B. Spratling and D. Mortari, "A survey on star identification algorithms," Algorithms **2**, 93–107 (2009).
15. D. Rijlaarsdam, H. Yous, J. Byrne, D. Oddenino, G. Furano, and D. Moloney, "A survey of lost-in-space star identification algorithms since 2009," Sensors **20**, 2579 (2020).
16. V. Schiattarella, D. Spiller, and F. Curti, "A novel star identification technique robust to high presence of false objects: the multi-poles algorithm," Adv. Space Res. **59**, 2133–2147 (2017).
17. E. Silani and M. Lovera, "Star identification algorithms: novel approach and comparison study," IEEE Trans. Aerosp. Electron. Syst. **42**, 1275–1288 (2006).
18. T. Schwarz, "Prototyping of a star tracker for pico-satellites," Ph.D. thesis (Luleá University of Technology, 2015).
19. G. H. Visser, "Development and verification of a star tracker algorithm chain," Ph.D. thesis (Delft University of Technology, 2017).
20. D. Mortari and F. L. Markley, "Quaternion attitude estimation using vector observations," J. Astronaut. Sci. **48**, 359–380 (2000).