Aerospace Science and Technology 155 (2024) 109700

Contents lists available at ScienceDirect

Aerospace Science and Technology

journal homepage: www.elsevier.com/locate/aescte

Closed-loop AI-aided image-based GNC for autonomous inspection of uncooperative space objects

Andrea Brandonisio [*,1], Michele Bechini [1], Gaia Letizia Civardi, Lorenzo Capra, Michèle Lavagna

Politecnico di Milano, Dept. of Aerospace Science & Technology, Via La Masa, 34, Milan, 20156, Italy

ARTICLE INFO

Communicated by Christian Circi

Keywords:
Autonomous GNC
Image-based navigation
Deep reinforcement learning
Relative dynamics
On-orbit servicing

ABSTRACT

Autonomy is increasingly crucial in space missions due to several factors driving the exploration and utilization of space. In the meanwhile, Artificial Intelligence methods begin to play a crucial role in addressing the challenges associated with and enhancing autonomy in space missions. The proposed work develops a closed-loop simulator for proximity operations scenarios, particularly for the inspection of an unknown and uncooperative target object, with a fully AI-based image processing and GNC chain. This tool is based on four main blocks: image generation, image processing, navigation filter, and guidance and control blocks. All of them have been separately tested and tuned to ensure the correct interface and compatibility in the close-loop architecture. Afterwards, the overall architecture is deployed in an extensive Montecarlo testing campaign to verify and validate the performance of the proposed IP-GNC loop.

1. Introduction

Nowadays enhanced autonomy is the research driver of most of the leading space agencies, as spacecraft independence would allow for reliable, cost-effective, lower-risk services, and much more flexibility in mission planning. Moreover, in the fast-developing field of space exploration and satellite deployment, the concept of IOS (In-Orbit Servicing) has emerged as one of the primary solutions to extend the lifespan of a mission, enhance the capabilities and ensure the sustainability of spacecraft, especially those orbiting around the Earth. IOS can include a wide selection of activities of great interest such as maintenance, refuelling, debris mitigation, upgrade, repair, assembly, relocation, orbit modification and non-contact support [1,2]. Therefore, a high level of autonomy while carrying out these activities would lead to greater benefits. That is why, at the heart of this groundbreaking field lies a crucial aspect: the spacecraft's GNC (guidance, navigation and control) system autonomy. Indeed, autonomy is a benchmark which ensures precise navigation, rendezvous, docking, or the execution of maintenance operations, which require cutting-edge technologies, algorithms, and sensors that increase the capability of these vehicles to adapt, respond, and perform delicate operations with unparalleled precision in a challenging environment as the space one. Guidance and control play an integral role in this autonomy level: the first requires the use of advanced algorithms to determine optimal trajectories in very different types of space scenarios exploiting the information coming from integrated systems made up of a multitude of sensors, such as star trackers, GPS, and LiDAR, that enable precise positioning and orientation, facilitating safe proximity operations and challenging manoeuvres around other space bodies or objects. Control autonomy, instead, grants spacecraft the ability to make real-time decisions and adjustments, ensuring stability, collision avoidance, and successful execution of complex tasks. This autonomous control may be strongly empowered by AI-driven software, adaptive control algorithms, and fault-tolerant systems that enable spacecraft to respond promptly to unforeseen situations, guaranteeing mission success while minimizing possible risks. Subsequently, in the constant evolution of in-orbit servicing, the focus on spacecraft guidance and control autonomy emerges as a fundamental aspect which strongly drives the success and efficiency of missions.

In recent decades, the increasing interest in artificial intelligence and machine learning has heavily influenced this growing hunger for spacecraft autonomy. Indeed, the number of works on machine learning techniques applied to space-related problems is continuously growing, especially in the context of guidance, navigation and control [3–5], affecting almost all the most important aspects of the GNC in a great

[*] Corresponding author.
E-mail addresses: andrea.brandonisio@polimi.it (A. Brandonisio), michele.bechini@polimi.it (M. Bechini), gaialetizia.civardi@polimi.it (G.L. Civardi), lorenzo.capra@polimi.it (L. Capra), michelle.lavagna@polimi.it (M. Lavagna).

[1] Equal contribution.

https://doi.org/10.1016/j.ast.2024.109700
Received 7 May 2024; Received in revised form 17 October 2024; Accepted 26 October 2024
Available online 29 October 2024

variety of scenarios ranging from feasibility studies to on-board applications [6,7]. The potential of machine learning, mainly resulting from its adaptability and diversity, allows it to be tailored for every need, offering different solutions (supervised, unsupervised and reinforcement learning) and declinations (i.e. linear, convolutional, recurrent neural networks, etc.).

The initial stage of an autonomous GNC system involves acquiring measurements, specifically the relative pose, which is predominantly derived from monocular images due to the advantages of monocular cameras over other sensors [8]. Consequently, in recent years, many vision-based algorithms utilizing monocular frames have been developed for various applications, including spacecraft relative navigation. The growing interest in vision-based solutions has been accompanied by advancements in image generation methodologies, characterized by higher accuracy and photorealism, supported by enhanced computational capabilities. Nonetheless, access to actual spaceborne images remains limited, as does the availability of validated synthetic spaceborne image datasets. Furthermore, the existing datasets are typically designed for benchmarking pose initialization algorithms and do not provide the temporal sequences of frames required for testing complete navigation pipelines. Notably, the integration of autonomous guidance and control algorithms presents an additional challenge: the need to generate images that align with the relative dynamics of the system, which are influenced by guidance and control actions based on the state estimates obtained from processing the image generated in the previous iteration within a closed loop. Thus, an in-loop image generation algorithm is required to ensure the generated images are representative and consistent with the system's actual dynamics.

Concerning the vision algorithms, the outcomes of ESA satellite pose estimation challenges held in 2019 (SPEC2019) [9] and 2021 (SPEC2021) [10] demonstrated that relying on Convolutional Neural Networks (CNNs) to perform target detection and features regression from monocular images combined with Perspective-*n*-Point (PnP) problem solvers (such as EPnP [11]) is the most promising approach for autonomous relative pose (i.e., relative position and attitude) estimation when dealing with known uncooperative targets, over conventional machine vision-based approaches. Despite that, the top-performing architectures [9,10,12] that participated in SPEC2019 and SPEC2021 aimed at achieving high accuracy, disregarding the overall computational complexity, relying on heavyweight CNNs [13] and domain discriminator modules [10], that may result in computational overheads not compatible with currently available space-grade on-board computers. The possibility of relying on lightweight CNNs developed for low inference time on mobile devices was initially addressed in [14] by proposing an architecture inspired by [15] where the target detection step and the line segment detection were entrusted to a YOLOv5 and a modified M-LSD [16], respectively. The architecture in [14] proved that top-level performances in relative pose estimation and high-level domain gap bridging capabilities can be achieved by relying on lightweight CNNs. Despite that, the need for processing a high amount of 2D-to-3D matches stands as the main bottleneck for the architecture in [14], being the most time-consuming step of the overall pipeline and making it suitable only for the initialisation and not for the tracking of the relative state. Leveraging the advancements in Human Pose Estimation (HPE), which consists of retrieving the 2D locations of pre-selected landmarks corresponding to joints of the human skeleton, a novel architecture leveraging the YOLOv8-pose model as the only CNN within the pipeline to perform both target detection and keypoint regression in a single inference was introduced in [17]. This novel architecture leverages a check on the actual need for a second inference (where the image cropped to the ROI extracted during the first inference is processed to retrieve the 2D landmarks) based on the outputs of the first inference, enhancing the efficiency of the overall pipeline. Further, relying on the confidence score associated with each keypoint, the optimal subset of 2D-to-3D matches to be fed to the EPnP solver is adjusted for each image processed, discarding the low-confidence keypoints and improving the overall robustness and reliability. The YOLOv8-pose-based architecture was proven in [17] to achieve top-level performances both on synthetic SPEED images [18] and mock-up frames from SPEED+ [19] (the benchmark datasets introduced during SPEC2019 and SPEC2021), thanks to the custom augmentation techniques used during the training phase [17], with maximum running time on CPU of ≈ 131 ms, featuring also demonstrated generalisation capabilities to complex targets such as Envisat.

At the same time, Deep Reinforcement Learning (DRL), as a subdiscipline of machine learning, is gaining much interest, especially for autonomous guidance and control problems, thanks to its peculiarity of having agents capable of learning how solving tasks (like navigation or planning) within a potentially changing environment, ensuring the power of adapting the mathematical model to a practically endless number of space-related applications. Several scenarios have already been investigated: spacecraft hovering or orbiting around small bodies, planetary landing and close-proximity operations [20–23], which oppose similar results to more classical approaches [24–26]. According to [27], in the current state-of-the-art, DRL is a powerful tool when dealing with decision-making problems and its perfect compatibility with ANN as function approximators allows it to improve the generalising capabilities of the resulting policy, and to solve more and more complex problems characterised by high-dimensionality and continuous state and action spaces [28].

Being inspired by the growing trend of AI applications in space-related problems, in this paper, we propose an innovative AI-based GNC architecture, combining an image-based navigation tool with an autonomous DRL-based guidance and control (GC) agent in a simulation environment with rendering-in-the-loop to ensure that the effects of the adaptive control actions are correctly taken into account in AI-based image processing (IP) at each iteration. The navigation block interfaces a pose estimation pipeline, based on the YOLOv8-pose-based architecture [17], and a relative navigation filter. The former estimates the chaser-target relative pose for each processed frame that is then passed to the navigation filter which outputs the overall estimation of the chaser-target relative state. Despite the high accuracy of the YOLOv8-based pose estimation, the inclusion of filtering algorithms within the navigation chain is expected to enhance both the accuracy and the robustness of the system, as highlighted in [29]. From a high level perspective, we propose a *loosely-coupled* architecture, in which the relative pose is computed by a pose solver prior to the navigation filter. As pointed out in [30], a loosely coupled approach is preferable for uncooperative tumbling targets, because the fast relative dynamics could negatively affect feature tracking and return highly unreliable measurements to the filter. Similarly to [31], this work adopts two distinct filters: one for the relative rotation and one for the relative translation. The proposed navigation architecture relies on a linear $H\infty$ filter for the translational dynamics, and Invariant Extended Kalman Filter (IEKF) for the rotational part. The output of the navigation block is a refined relative pose, which is then used by an autonomous guidance and control agent [32,33], trained via Deep Reinforcement Learning to plan the chaser's relative trajectory around an uncooperative space object (i.e. the target) to optimise its inspection. Within this work, the following contributions want to be assessed:

- feasibility study of a closed-loop AI and vision-based GNC architecture, also in sight of a future processor (PIL) or hardware-in-the-loop (HIL) test campaign;
- assessment of real input data (i.e. IP and NAV block output) compatibility for the guidance and control autonomous agent;
- relaxation of some assumptions (e.g. attitude control) on the previously tested guidance and control agent.

In the following sections, the GNC architecture is described in detail, and its performance is investigated in a TANGO relative dynamics scenario. Specifically, the dynamical models are briefly presented in Sec. 2;

Table 1
TANGO classical orbital elements.

a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
7133	0.0045	98.28	0	0	0

Sec. 3 describes the GNC algorithm architecture, defying all the used blocks, from the image generation tool to the navigation filter. The GC agent is defined and characterised in detail in Sec. 4. Afterwards, in Sec. 5, the performance results are presented and analysed in terms of image processing, filter errors, and guidance and control return.

2. Dynamical models

This section introduces the dynamical model employed to describe the motion of both the chaser and the target. The relative dynamics between the two objects is obtained from the differential absolute state of the two retrieved using a high-fidelity dynamical model for each object. The TANGO orbit is defined by the classical orbital elements reported in Table 1.

Considering the specifics of the region of space in which the scenario takes place and the time window of the investigated case, the main perturbation added to the nominal two-body problem dynamics is the Earth's oblateness effect J_2. The set of equations describing the absolute dynamics of the two spacecraft is reported in Eq. (1).

$$\ddot{\mathbf{r}} = -\mu \frac{\mathbf{r}}{\|\mathbf{r}\|} + \mathbf{p} \tag{1}$$

The perturbing acceleration component $\mathbf{p}$ in eq. (1), due to the J_2 zonal harmonics effect, takes the explicit form in Eq. (2):

$$\mathbf{p} = \frac{3}{2}\frac{J_2 \mu R^2}{r^4}\left[\frac{x}{r}\left(5\frac{z^2}{r^2}-1\right)\mathbf{i} + \frac{y}{r}\left(5\frac{z^2}{r^2}-1\right)\mathbf{j} + \frac{z}{r}\left(5\frac{z^2}{r^2}-1\right)\mathbf{k}\right] \tag{2}$$

where $J_2 = 0.00108263$, μ is the Earth's geocentric gravitational constant, and R is the Earth's radius.

Regarding the attitude of the two spacecraft, the standard Euler equations in Eq. (3) are used, and the orientation is expressed via quaternion representation, as in Eq. (4).

$$\begin{aligned}\dot{\omega}_x &= \frac{I_{yy}-I_{zz}}{I_{xx}}\omega_y\omega_z + \frac{M_x}{I_{xx}}\\ \dot{\omega}_y &= \frac{I_{zz}-I_{xx}}{I_{yy}}\omega_z\omega_x + \frac{M_y}{I_{yy}}\\ \dot{\omega}_z &= \frac{I_{xx}-I_{yy}}{I_{zz}}\omega_x\omega_y + \frac{M_z}{I_{zz}}\end{aligned} \tag{3}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega\mathbf{q} \tag{4}$$

In the equations above, $\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}$ are the angular velocity vector and its derivative, respectively; I_{xx}, I_{yy}, I_{zz} are the principal inertial components, M is the vector of external torque, $\mathbf{q}, \dot{\mathbf{q}}$ are the quaternion vector and its derivative respectively, and Ω is the skew-symmetric matrix associated to $\boldsymbol{\omega}$.

3. GNC architecture description

This paper proposes a fully AI and image-based GNC algorithm working in closed loop thanks to the image generation step performed at each iteration. The testing environment is based on the following five different blocks:

- **Dynamics.** This block defines the chaser-target relative environment: the relative position and velocity are computed from the absolute position and velocity of both target and chaser integrated in the Earth-Centered Inertial (ECI) reference frame, relying on the J2-perturbed dynamics already defined in Eq. (1). Concerning the relative attitude dynamics, the Euler equations are integrated starting from the initial conditions for each satellite; then, the relative attitude is retrieved from the absolute orientation variables in the ECI reference frame.
- **Image Generation.** The image generation (IG) block generates a synthetic frame with a high level of photorealism, if compared to actual spaceborne images, for each GNC iteration, which is then fed to the IP block. The IG block comprises a synthetic image generation core based on POV-Ray [34] (i.e., a simplified version of the image generation tool described in [35]) and a high-fidelity noise generator developed in [17]. Namely, given the outputs of the Dynamics block, the rendering tool generates a noiseless image that is then processed through the noise generator (which entails the actual sensor characteristics and settings), obtaining the noised frame as output that serves as input to the IP step.
- **Image Processing.** The image processing (IP) block relies on the YOLOv8-pose-based architecture introduced in [17] to process each image generated by the IG block. The outputs of the IP block are the relative target position (r) and attitude (q) with respect to the chaser camera frame. Notice that, without any loss of generality, here it is assumed that the camera reference frame corresponds to the chaser body reference frame.
- **Navigation.** The navigation (NAV) block is based on a loosely coupled architecture involving a relative translational filter and a relative rotational filter. The relative position and velocity estimation is entrusted to a H-∞ filter, which is a robust linear estimator. The filter exploits a linearized relative motion model The outputs of the H-∞ filter are the relative position and velocity in the chaser LVLH reference frame. Regarding the relative rotational filter, the Invariant Extended Kalman Filter (IEKF) is exploited here. The input of the filter is the relative chaser-target angular velocity inferred from the chaser's absolute navigation, the earth-based observation of the target's rotational state, and the relative quaternion retrieved from the IP block. The output of the IEKF is a refined relative quaternion.
- **Guidance** and **Control.** The Guidance and Control (GC) is the last block of the pipeline and is based on [36]. It takes as input the relative chaser-target position and velocity in the target LVLH reference frame and the relative attitude of the target in the chaser camera frame. Although these quantities are not directly available from the navigation or the IP blocks, the outputs of these blocks can be manipulated to get the correct inputs for the GC block. The GC block outputs the control action of the chaser (shaped as an acceleration vector) that affects both the dynamics and navigation blocks.

Fig. 1 shows the architecture pipeline described above. The relative chaser-target position and velocity needed by the GC block always come from the H-∞ filter; on the contrary, the relative attitude may be derived directly from the real dynamic with the addition of Gaussian noise, from the IP block, or the rotational navigation filter. This choice is due to the fact that the guidance algorithm is sensitive to the input attitude. In this way, it is possible to test its robustness against a realistic input attitude (coming from the filter) with respect to a simpler input (real dynamics with white noise). In the TANGO case study definition investigated in the following sections, these different options will represent some of the steps performed to increase the complexity of the simulation during the testing analyses.

The environment simulation is Python-based, and all the blocks involved need an initialization process that shapes them in the correct environmental conditions based on `.yaml` input files. Each of the blocks has its evaluation metrics. The following subsections detail the IG, IP, and NAV blocks of the pipeline proposed. The dynamics model has already been outlined in Sec. 2, while, due to its relevance, the GC block is thoroughly outlined in Sec. 4, mainly focusing on its evaluation metrics and parameters.

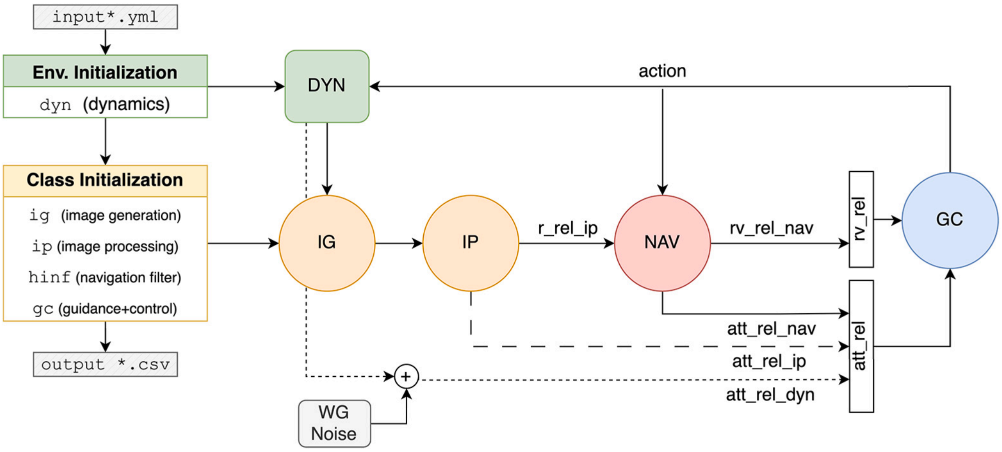
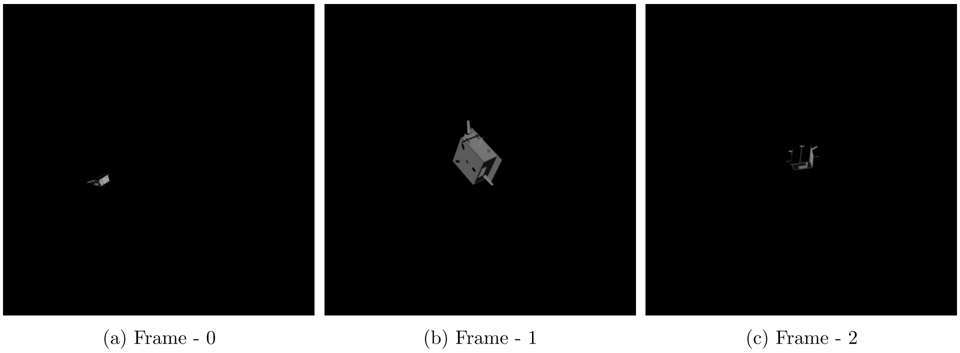


Fig. 1. AI-based GNC pipeline architecture scheme.


(a) Frame - 0

(b) Frame - 1

(c) Frame - 2

Fig. 2. Samples of TANGO rendered images taken from a testing simulation of the GNC pipeline.

3.1. Image generation

The image generation tool adopted within this work is based on POV-Ray [34], an open-source software already successfully adopted to render spaceborne scenes comprising both natural and artificial targets [37–39,35]. The POV-ray-based tool for spaceborne image generation originally developed in [35] and adopted for the generation of the dataset named MINIMA [40] has been adapted to generate a single image per iteration, instead of a full dataset, and included within the pipeline outlined in Fig. 1. Namely, the pipelines that define the input file fed to the rendering engine have been optimized without affecting the image generation process. Consequently, the validations of the POV-Ray-based tool, the simplified TANGO model, and the generated synthetic images outlined in [35] are still valid and not reported here for brevity. Fig. 2 shows some examples of the rendered TANGO images taken in different moments of one of the testing simulations. Although the rendering tool is already capable of handling the case of celestial bodies in the background (e.g., the Earth) [35], and despite the performances of the IP step leveraged within this work pointed out an extremely low sensitivity to this disturbance [14], all the images are rendered with a perfectly black background (simulating the deep space and disregarding any background celestial bodies) to avoid long rendering times that entails a heavy increment of the overall simulation time. Further, the guidance reward function does not include terms related to the Earth or its effects on the quality of the images acquired. Please refer to [35] for a detailed description of the image generation process and the modeling assumptions used within the image generation tool adopted within this work.

Each rendered noiseless image is further processed to include the characteristic noise levels of the assumed camera model before being fed to the IP block. The adopted noise generator is the high-fidelity VIS sensor model introduced in [17]. Namely, the VIS sensor model includes the main noise sources that characterize CCD and CMOS sensors, leveraging the distributions and models introduced in [41,42]. Notably, the VIS sensor model also retains the fixed pattern noises (e.g., the pixel response non-uniformity and the dark signal non-uniformity), modeling them as prescribed by the EMVA1288 standard [43]. The noise generator estimates the average photons received by the detector by applying the quantum efficiency and the ADC gain of the selected camera to the noiseless images. Once the average photons per pixel are available, the noise sources are applied consistently with the camera model characteristics and the EMVA1288 standard before returning the noised image already expressed in digital numbers. The high-fidelity noise generator has been compared in [44] against frames acquired with an actual CMOS sensor (i.e., the Teledyne FLIR CM3-U3-13Y3C), proving the high level of representativeness of each noise component and the final noised images given as output. Namely, the comparison between synthetic and actual noises has been performed by assessing qualitatively through visual inspection the fidelity of the dark and bias frames, using histogram comparison, and numerically by evaluating the mean value and the standard deviation of the frames, achieving an average variation of about +4% between the actual and synthetic mean noise value [44]. A detailed discussion of the VIS sensor noise generator adopted within this work and its comparison against actual images can be retrieved in [44]. Please notice that the FLIR CM3-U3-13Y3C CMOS camera is the one available in the PoliMi-DAER facility for simulating proximity operations [45], hence it has been selected as the default camera model for the image generation block, forecasting future testing activities of the proposed pipeline with hardware-in-the-loop. The interested reader is referred to [17] for more details on the high-fidelity VIS sensor model and its comparison against actual frames. Once that the noised image is retrieved, it is fed to the IP block. Table 2 summarises the inputs and the outputs of the IG block. Notice that all the inputs and the intermediate results of this block (including the ground truth relative pose) are

Table 2
Input/Output of IG block.

Input	Chaser and target position in ECI reference frame, chaser and target body orientation in ECI reference frame, Sun position in ECI reference frame, camera parameters.
Output	Noised image of the target as seen from the chaser, compliant with the selected camera model and the actual dynamics.

saved in a dedicated file for the post-processing of the performances registered.

3.2. AI-based space object pose estimation

The IP block consists of an AI-aided keypoint-based pose estimation algorithm that leverages a single, multi-tasking, and efficient CNN that performs both target detection and keypoint regression in a single inference. Namely, the pipeline leverages the YOLOv8s-pose[2] model due to its efficiency and top-level performances on benchmark datasets for human pose estimation tasks. The relative pose estimation pipeline leveraging the YOLOv8-pose has been introduced in [17]. The analyses in [17] point out that the architecture is capable of handling the relative pose estimation tasks for several target geometries (TANGO included) by providing accurate estimates thanks to the outlier rejection scheme based on the confidence scores associated with the retrieved landmarks. Namely, in line with [46], a minimal subset of keypoints is defined by selecting a constant number of keypoints, N_{kp}, with the highest confidence scores. Subsequently, all remaining detected keypoints with an associated confidence score higher than the threshold $\tau_{kp,conf}$ are appended to this minimal subset. It is worth noting that this outlier rejection scheme helps prevent the inclusion of low-confidence keypoints (e.g., those corresponding to components outside the field of view or keypoints erroneously detected by the CNN) in the 2D-to-3D matches fed to the EPnP solver, thereby reducing errors in the relative pose estimated by the EPnP solver. Before optimizing the EPnP output using the Levenberg-Marquardt (LM) minimization scheme, the relative position estimated by EPnP, t_{EPnP}, is compared with a coarse estimate, t_{coarse}, derived by combining the region of interest (ROI) and keypoints, to detect potential outliers, as suggested in [46]. If an outlier is detected and the ROI sides do not partially overlap with the image borders (indicating that the target may be partially out of the field of view), the coarse relative position, t_{coarse}, is used as the initial estimate for LM optimization, in conjunction with the relative attitude provided by EPnP. Otherwise, the LM optimization takes the relative pose provided by EPnP as its input. The result of the LM optimization provides the final relative pose estimate delivered by the IP architecture. Also, the pipeline can partially bridge the domain gap between synthetic and mock-up frames due to the custom augmentations included in the training phase. Further, the check on the need for two inferences (based on the ratio between the ROI and the image diagonals that shall be lower than a threshold value k_{diag} to trigger the second inference) improves the efficiency, making it possible to retrieve a pose estimate after an overall mean computational time on CPU of about 60 ms and 131 ms for the case of a single and a double inference, respectively. The interested reader is referred to [17] for a detailed discussion of the YOLOv8-pose-based pose estimation algorithm and the assessment of its performances on benchmark datasets.

To tailor the architecture to the case of TANGO adopted for the analyses here presented, the YOLOv8s-pose has been trained on the MINIMA dataset [35], using the available ROI annotations [47] and labeling the 2D landmarks for each image from the 3D wireframe model and the already available relative pose annotations [40]. Please notice that the visibility score associated with each keypoint for each frame has been defined by using the Möller–Trumbore ray-triangle intersection algorithm [48] as in [49]. The 3D keypoints selected for TANGO are those already exploited in [46], corresponding to the tip of the three main appendages, the four corners of the upper solar array, and the four corners of the lower plate of the main body. MINIMA has been split into 27000 training images, 3000 validation images, and 3000 testing images. The training images have been augmented during the training phase by using the default augmentations of the YOLOv8 and the high-fidelity VIS sensor noise augmentation due to its proven capability of enhancing the performances of the YOLOv8s-pose also on synthetic images [17]. The training has been performed using a NVIDIA RTX™ A6000 GPU using the stochastic gradient descent method with a learning rate of 0.01 linearly decaying to 0.0001, a momentum of 0.937, weight decay of 5×10^{-4}, and a minibatch size of 64 images, using the default losses of the YOLOv8-pose with relative gain set to 7.5 and 18 for the object and keypoint losses, respectively. The training lasts 900 epochs since no improvements have been noticed in training longer. The criteria adopted to evaluate the performances of the YOLOv8s-pose are the Intersection-over-Union (IoU) and the average precision (AP) for the object detection performances, while the Object Keypoint Similarity Index (OKS) and the keypoint AP are considered as the keypoint detection performance metrics. The details and definitions of these standard metrics are not discussed here for brevity, but the interested reader can find a detailed discussion in [50]. The trained model scored a mean AP_{50}^{95} of 97.4% and a mean IoU of 96.11% for the object detection task, while for the keypoint regression task, it achieved a mean AP_{50}^{95} of 99.1% and a mean OKS of 97.99% on MINIMA images. The architecture hyperparameters (i.e., those related to the check for the need for the second inference and to the keypoint outlier rejection scheme) have been optimized through the 3D grid search using the MINIMA test set, following the same procedure outlined in [17] for the case of SPEED images, resulting in a diagonal ratio threshold of $k_{diag} = 0.45$, a keypoints minimum set of $N_{kp} = 4$, and a minimum confidence threshold value for keypoints to be added to the minimum set of $\tau_{kp,conf} = 0.80$. Fig. 3 provides examples of detected ROI and keypoints extracted from sample images using the trained YOLOv8s-pose model. Notably, the extracted ROI is highly accurate and does not crop out any portion of the target. Concerning the keypoints and the related accuracy mapped in the colormap in Fig. 3, both the 2D locations and the confidences are high for landmarks within the camera FOV while, for keypoints out of the FOV, the 2D location is wrong and, notably, the associated confidence is extremely low and well split from the confidence of keypoints that are within the FOV, allowing to detect these outliers and discard them from the subset fed to the EPnP as detailed above, resulting in highly accurate pose estimates even in that case (provided that enough keypoint are within the FOV). Table 3 resumes the inputs and the outputs of the IP block. Similarly to the case of the IG block, all the outputs of the IP block are saved in a dedicated file for the post-processing.

3.3. Navigation filter

The loosely coupled architecture for the relative navigation block relies on two distinct filters, i.e., one for the relative rotational dynamics and another for translational dynamics, to refine the IP relative pose estimates and to make the whole GNC chain more robust to outliers or faulty relative state measurements. The inputs/outputs of the block are summed up in Table 4.

3.3.1. Relative translation

The standard Kalman filter (KF) is an optimal estimator for linear systems with zero-mean Gaussian process and measurement noise [51].

[2] https://github.com/ultralytics/ultralytics.

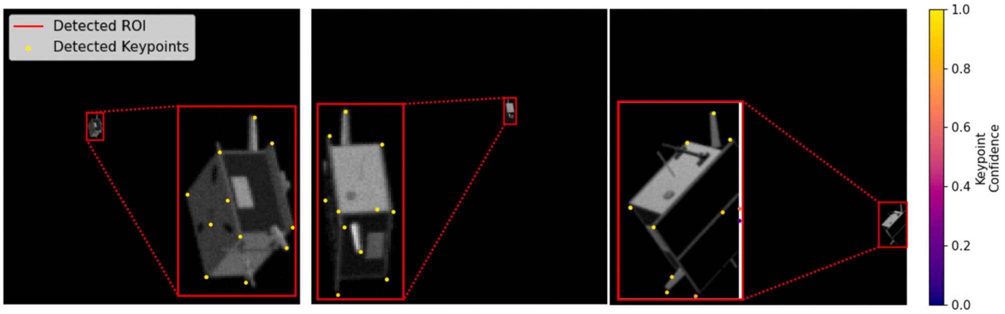


Fig. 3. Detected ROI, keypoints, and confidence score using the trained YOLOv8s-pose model on sample images. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

Table 3
Input/Output of IP block.

Input	Noised image of the target as seen from the chaser, compliant with the selected camera model and the actual dynamics.
Output	Relative chaser-target position and target orientation in camera reference frame.

However, when dealing with the outputs of an image processing chain that may contain nonlinearities and outliers, the assumptions of the Kalman filter may not be met. A robust counterpart of the KF is the H-∞ filter, which minimizes the ∞-norm of the estimation error without relying on restrictive assumptions regarding the statistics of the process and measurement noise. This approach was already proven to be successful in a similar scenario in [52]. Let's consider a linear time-invariant system:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= F\boldsymbol{x}_k + G\boldsymbol{w}_k \\ \boldsymbol{y}_k &= H\boldsymbol{x}_k + \boldsymbol{v}_k \end{aligned} \tag{5}$$

with $\boldsymbol{x}_k$ being the state vector, $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ the process and measurement noise respectively with associated covariance matrices Q and R. The peculiarity of the H-∞ filter is the computation of the gain matrix K_k, which is chosen such that $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_\infty < \frac{1}{\gamma}$, where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ represent the true and predicted state, respectively and γ is the tuning parameter. The complete formulation of the H-∞ filter is reported in Algorithm 1.

Algorithm 1 H_∞ filter.

```
Initialization:
$\hat{\mathbf{x}}(t_0) = \mathbf{x}(t_0)$
$P(t_0) = P_0$
Propagation:
$\hat{\mathbf{x}}_k^- = F_{k-1}\hat{\mathbf{x}}_{k-1}^+ + G_{k-1}\mathbf{u}_{k-1}$
$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1}$
Filtering:
$K_k = P_k^-[I - \gamma P_k + H_k^T R_k^{-1} H_k P_k]^{-1} H_k^T R_k^{-1}$
$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\mathbf{y}_k - H_k\hat{\mathbf{x}}_k^-)$
$P_k^+ = (I - K_k H_k)P_k^-(I - K_k H_k)^T$
```

Notice that the H-∞ filter adopted here exploits the first-order J2 linear model formulated in [53]. The relative state vector is defined as:

$$\varrho = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{6}$$

The linearized dynamical model can be written as:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = A(t)\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + B(t)\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{7}$$

where the matrices $A(t)$ and $B(t)$ are defined as in Eq. (8) and Eq. (9), respectively.

$$A = \begin{bmatrix} 0 & 2h^2/r^2 & 0 \\ -\frac{2h^2}{r^2} & 0 & -\frac{2k_{J2}s_{2i}s_\theta}{r^3 h} \\ 0 & \frac{2k_{J2}s_{2i}s_\theta}{r^3 h} & 0 \end{bmatrix} \tag{8}$$

$$B = \begin{bmatrix} \frac{2\mu}{r^3} + \frac{h^2}{r^4} + \frac{4k_{J2}(1-3s_i^2 s_\theta^2)}{r^5} & -\frac{2\dot{r}h}{r^3} + \frac{3k_{J2}s_{2\theta}s_i^2}{r^5} & \frac{5k_{j2}s_{2i}s_\theta}{r^5} \\ \frac{2\dot{r}h}{r^3} + \frac{5k_{J2}s_{2\theta}s_i^2}{r^5} & -\frac{2\mu}{r^3} + \frac{h^2}{r^4} - \frac{k_{J2}(1+2s_i^2 - 7s_i^2 s_\theta^2)}{r^5} & \frac{3k_{J2}\dot{r}s_{2i}c_\theta}{r^5} \\ \frac{5k_{J2}s_{2i}s_\theta}{r^5} & \frac{3k_{J2}\dot{r}s_{2i}s_\theta}{hr^4} & -\frac{\mu}{r^3} - \frac{k_{J2}(3-2s_i^2-5s_i^2 s_\theta^2)}{r^5} \end{bmatrix} \tag{9}$$

State Noise Compensation (SNC) is adopted to compensate for unmodeled accelerations, following the analytical approach presented in [54], in which the process noise covariance is evaluated in the RTN frame. This approach is used to mitigate the effect of having such a simple linear model, as opposed to the nonlinear dynamics characterising the relative motion between the chaser and the target spacecraft. Please notice that the filter relies on the knowledge of the chaser's absolute state vector, which is a reasonable assumption if considering missions with at least one GPS receiver. In this way it is possible to transform the relative position vector given as output by the IP block from the chaser camera reference frame to the chaser LVLH reference frame required by the filter. A fluctuation of the chaser true anomaly can indeed affect the overall estimation accuracy. For this reason, the true anomaly of the chaser spacecraft is corrupted by means of an additive white gaussian noise. Specifically, the noise level is described by a Gaussian distribution with standard deviation of $\sigma_{pos} = 10^{-2} km$ and $\sigma_{vel} = 10^{-4} km/s$, respectively.

3.3.2. Relative rotation

The attitude of a rigid body naturally evolves on the Special Orthogonal Group *SO(3)*, i.e., the set of 3×3 orthonormal matrices. The Lie algebra of *SO(3)* is represented by the set of 3×3 skew-symmetric matrices, and it is generally denoted as *so(3)*. The relative attitude kinematics between two rigid bodies can be written as:

$$\dot{R} = R(\boldsymbol{\omega})_\times, \quad R(0) = R_0 \tag{10}$$

where $R(t) \in SO(3)$ is the relative rotation matrix which represents the attitude of the target object with respect to the chaser spacecraft expressed in the chaser body frame, and $\boldsymbol{\omega}$ is the relative angular velocity expressed in the chaser frame.

The relative attitude filter within the NAV relies on the recently proposed Invariant Extended Kalman Filter (IEKF), which is a generalization of the classical Multiplicative Extended Kalman Filter (MEKF). The overall formulation of the IEKF is reported in Algorithm 2.

Table 4
Input/Output of NAV block.

Input	Relative chaser-target position in chaser LVLH reference frame, relative chaser-target attitude in chaser body frame and relative angular velocity.
Output	Relative chaser-target position and velocity in chaser LVLH reference frame and relative chaser-target attitude in chaser body frame.

Algorithm 2 IEKF filter.

```
Initialization:
$\hat{R}(t_0) = R(t_0)$
$P(t_0) = P_0$
Propagation:
$\hat{R}_k^- = \hat{R}_{k-1}^+ exp((\omega_{k-1})_\times \Delta t)$
$P_k^- = P_{k-1}^+ + M_c \Delta t$
where:
$M_c = \hat{R}_k \Sigma_u \hat{R}_k^T$
Filtering:
$H = \left[ (A(\boldsymbol{q})\bar{\boldsymbol{r}}_1)_\times, \quad (A(\boldsymbol{q})\bar{\boldsymbol{r}}_2)_\times \right]^T$
$K_k = P_k^- H^T (H_k P_k^- H_k^T + \hat{R}_k \Sigma_y \hat{R}_k^T)^{-1}$
$\hat{\boldsymbol{y}} = \left[ \hat{R}_k^- \bar{\boldsymbol{r}}_1, \quad \hat{R}_k^- \bar{\boldsymbol{r}}_2 \right]^T \qquad \boldsymbol{y} = \left[ A(\boldsymbol{q})\bar{\boldsymbol{r}}_1, \quad A(\boldsymbol{q})\bar{\boldsymbol{r}}_2 \right]^T$
$R_k^+ = exp((\hat{\boldsymbol{y}} - \boldsymbol{y})_\times) R_k^-$
$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T$
```

In which $A(\boldsymbol{q})$ is the attitude matrix extracted from the IP estimate, while $\bar{\boldsymbol{r}}_1$ and $\bar{\boldsymbol{r}}_2$ are two mutually orthogonal reference directions. The matrices Σ_u and Σ_y represent the process noise and the measurement noise covariance, respectively. It is important to remark that the relative angular velocity is directly computed from the reference dynamics, adding a white Gaussian noise with σ equal to 5% of its magnitude before feeding it to the filter.

4. AI-based guidance and control

The Guidance and Control block is based on the work developed in [55], and further investigated in [33,32]. In this aforementioned framework, an autonomous guidance and control agent is proposed to plan trajectories around uncooperative and unknown space objects to reconstruct the overall shape of the object itself. The autonomous problem is formulated as an active SLAM (Simultaneous Localization and Mapping) problem, which is phrased as a Partially Observable Markov Decision Process (POMDP) and solved by the use of Deep Reinforcement Learning (DRL). The PPO (Proximal Policy Optimization) algorithm is used, as developed in [56], to optimize the autonomous agent. In this DRL broad domain, three main models need to be defined: state and action spaces, and reward function. The state space defines the input state of the GC block, and, therefore, as already mentioned it is composed of the relative chaser-target pose in terms of position, velocity, and attitude. The action space, instead, defines the output of the GC block, i.e. the action the autonomous agent decides to select at each time step of the simulation. In this paper, the action space is defined as a discrete space, which includes a pre-defined and limited set of actions, as introduced in [55]. The action is directly formulated as a vector in the chaser LVLH reference frame and afterwards fed to the dynamics and navigation block to close the simulation and GNC loop. Lastly, the reward function drives the agent policy optimization, aiming at maximizing it through positive and negative scores, which should incentivize a specific agent's behaviour. This reward is based on three main scores: distance, time and map. The *distance* score defines the region of space in which the spacecraft must remain around the object to avoid escaping or collision; the *time* score, instead, limits the time window by when the spacecraft should collect all the images to reconstruct the shape, whose quality and completeness is established by the *map* score. The map is retrieved by the photo that the spacecraft is assumed to take each 20 s, whose quality is computed based on the incidence angles between the Sun and camera, and the faces of the object's mesh. To conclude the definition of all the elements needed by PPO to optimize the autonomous guidance problem, it is necessary to mention the neural network architecture used to approximate PPO policy and value functions. Different options have been extensively studied in [55,32]: in this paper, an LSTM-based (Long-Short Term Memory) recurrent network is selected for both policy and value networks, as defined in Table 5. This general framework has been designed to be flexible to scenarios characterised by distinct target objects, space environments (even if specifically based on relative dynamics) and illumination conditions.

Table 5
Policy (Actor) and Value (Critic) Networks Architecture definition.

		Policy Network		Value Network	
Layer	Type	Elements	Activation	Elements	Activation
Input	Linear	12	-	12	-
Hidden - 1 (h1)	LSTM	24	-	24	-
Hidden - 2 (h2)	LSTM	24	-	24	-
Hidden - 3 (h3)	Linear	100	tanh	100	ReLU
Hidden - 4 (h4)	Linear	100	tanh	100	ReLU
Output	Linear	7	softmax	1	linear

Nevertheless, a preliminary training and testing phase is essential to tune the Guidance and Control agent to suit at best in this new GNC architecture. At first, the training is constructed to teach the agent to be compatible with the new and unknown environment; in particular, three main new characteristics have been updated starting from the baseline case scenario described in [32]:

- **Target object**: the case study that will be treated in Sec. 5 is characterized by the TANGO object (from PRISMA mission), on which the image generation tool is based; TANGO is shaped with the mesh defined in Fig. 4.
- **Camera model**: is based on the Chameleon 3 CM3-U3-13Y3C, already exploited in [45], on which the IG model is developed. The Chameleon 3 is characterized by a field-of-view of 44.54°x44.54°, a size of 1024 x 1024 pixels, and a focal length of 6 mm.
- **Trajectory range**: is limited by a minimum of 1.5 m and a maximum of 35 m. This constraint is driven by the IP block performance, whose estimation drastically decreases when overcoming about 35 m of relative distance between the Chameleon 3 and TANGO.

The PPO algorithm hyper-parameters setting is summarized in Table 6.

The agent's training process and testing activity are evaluated based on different metrics, such as episode score, map level and time. In Fig. 5 the resulting trends for the training simulation are depicted. It can be observed that the agent can achieve a high level both in terms of reward scoring and map level, almost reaching 90%, in line with the training behaviour in the baseline case [33,32].

It is important to underline that the training process from which the agent that will be used for the GNC pipeline is generated, is built on two strong assumptions:

- **Perfect input state**: the state space defined for the training process is not affected by any noise or uncertainty, meaning that the input state for the policy and value networks is equal to the

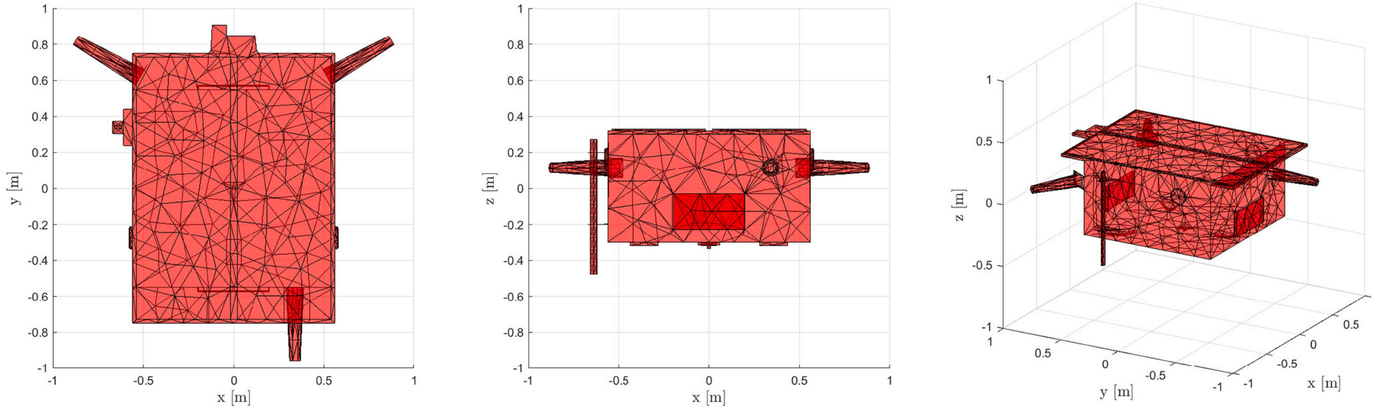


Fig. 4. TANGO (from PRISMA mission) mesh: symmetrical and cubical shape with a maximum length of 1 m, 174 nodes, 344 faces.

Table 6
Set of PPO algorithm's hyper-parameters used for training and testing the TANGO case study agent.

PPO Hyperparameters	
Reward Discount Factor γ	0.99
Terminal Reward Discount Factor λ	0.95
Clipping Factor ϵ	0.1
Entropy Factor s_2	0.02
Optimizer	ADAM
Optimization Step	each 10 episodes
Optimization Batch	32
Epochs	5
Simulation length	18000 episodes (training)
Policy Network	
Initialization	-
Learning Rate	1e-4
Final Activation Function	Sigmoid
Output Distribution	Categorical
Output Action	Sampling (training) argmax (testing)
Value Network	
Initialization	-
Learning Rate	1e-4

Table 7
Average testing results for TANGO agent.

Average Score	Average Map	Average Time	Average Thrusting
406.4	86.6%	6.6 h	22.6%

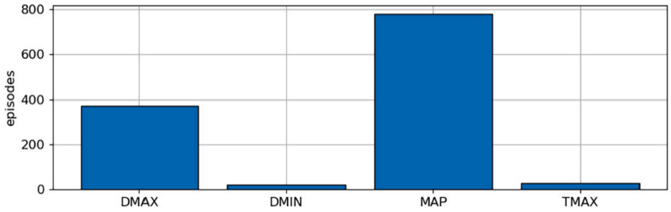


Fig. 6. TANGO case study agent's performance in testing simulation: ending conditions statistics.

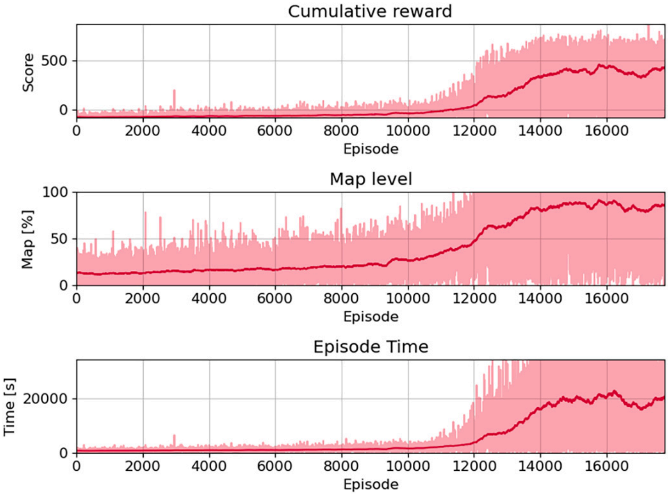


Fig. 5. TANGO case study training of the GC agent: metrics (reward score, map level, episode time) evolution along the simulation. The thick red line represents the average metrics values during the simulation.

real relative pose and attitude coming out from the dynamics of the environment. This approximation may influence negatively the performance of the agent when interfaced with the navigation chain of the GNC algorithm which intrinsically is affected by errors.

- **Fixed camera direction**: this particular approximation fixes the camera direction towards the target object. It is done to simplify the dynamic model of the environment and, consequently, the training process to uncouple the orbital and attitude manoeuvres.

Once the training phase is concluded, the outcoming policy network has been tested to understand the effective performance in nominal conditions: same random initial conditions, acceleration size, trajectory limit range and perfectly known input state; the results are summarized in Table 7. In particular, the average termination conditions have been analysed to define the expectation of the agent in the GNC pipeline; the results are shown in Fig. 6 for a 1200-episode simulation's length.

Before analysing the performance of the GNC algorithm, wide sensitivity analyses were performed on the trained network to assess the robustness of the agent and to foresee some of the needs that GC will incur in the interaction with a different environment, which will not always be coherent with the characteristics on which it has been trained. In particular, the sensitivity is investigated for random initial conditions with higher discretization ranges, different acceleration (action) sizes, different integration steps, and noisy input based on the error level of the IP algorithm.

5. TANGO fly-around case study

This section describes the testing campaign done to analyse the performance of the AI and image-based GNC pipeline. The algorithm schematic and architecture have already been presented in Fig. 1. The baseline simulation is based on some main assumptions that must be considered in the overall pipeline due to the input constraints of the different blocks:

- The **initial conditions** are the same random initial conditions on which the GC agent has been trained.
- The **Earth** is never in the **background** of the image for two main reasons: the guidance reward model was not built considering Earth illumination and the IG block would have been excessively slow in rendering the images to maintain a feasible simulation time.
- The guidance and control agent inserted in the GNC pipeline is trained on a **fixed target pointing** scenario, which does not require any attitude controller at first.
- The **trajectory** is **limited** to remain between 1.5-35 m from the target to avoid collisions or over-reducing the target size in pixels, leading to unfeasible target detection and relative pose estimation steps within the IP block.

It is worth mentioning that the analyses in [17] related to the robustness of the IP step pointed out that the presence of backgrounds has negligible effects on the estimated relative pose accuracy. These assumptions lead to the definition of some tests on the GNC pipeline, focusing on the different blocks. The first concerns the input of the chaser-target relative orientation, which could have been derived in three different ways: the first way derives it from the real quaternion value (computed by the dynamics block) with the addition of a proper level of noise, the second is based on the IP estimation, and the third exploits a rotational navigation filter to deal with this information. The second aspect is related to attitude control, which has been added as the last step of the plan, removing the assumption of fixed target pointing. Therefore, starting from these different features of each of the blocks involved in the algorithm, a testing workflow has been followed to increase the complexity of the simulations step by step, without rushing directly to the most general case. The adopted procedure is summed up in Table 8. A new level of complexity has been added at each new simulation (in *bold* in Table 8): change of the attitude input information for the guidance and control block, the increment of the noise within the IG block, and the addition of the attitude control for target pointing.

Each of the tests listed in Table 8 run on 5 episodes simulation length due to heavy computational time, with the same stopping conditions defined for the GC agent's training: exceeding of the trajectory limits (range between 1.5-35 m) or the time window (set to maximum 4 hours of trajectory length), and the achievement of map coverage (if reaches 100%). In the table all the testing process' steps are presented, even if in the following blocks' analysis only the results on the worst case scenario conditions (step ID:4 to ID:7) are treated and commented.

GC attitude input As just introduced, the attitude information needed by the GC agent can be derived directly by the dynamics, the IP block or by the rotational filter. In the first case, the estimation is less realistic because only noise (added following the model used in [57]) on the real quaternion is being considered. In the second and third cases, the input is more proper since it comes from the IP or NAV estimation. Nevertheless, for the IP-based formulation, another assumption must be considered: indeed the IP block outputs the chaser-target relative orientation in the chaser camera reference frame, while the GC agent needs it in the LVLH one. Therefore, to rotate the quaternion from the body to LVLH frame, the knowledge of the absolute position of the target object must be assumed, which can be still computed knowing the estimation of the relative position and the chaser absolute one.

Chaser attitude control Once the fixed target pointing assumption has been removed, a PD control has been activated in the dynamics block, computing the target pointing with the quaternion estimator algorithm (QUEST) [58]. In this way, it is also possible to introduce the chaser attitude estimation error within the navigation chain and to assess its influence on the relative pose estimation. The proportional and derivative gains are not fine-tuned on the chaser to leave a small level of uncertainty on the attitude control to analyse the robustness of the GC. Moreover, the coupling between the trajectory and attitude control is not taken into account: on the one hand to not increase too much the complexity, on the other because the GC agent is not trained for this kind of scenario. Nevertheless, even if this test explores a lot of states never experienced by the agent, it will be able to act properly.

Table 8
Planning of the testing simulation to assess the feasibility of the AI and Image-based GNC pipeline. In **bold** the new features that are added at each simulation step to increase the complexity of the system.

ID	Initial Cond.	Traj. Range	GC input	IG noise
1	GC	1.5-35 m	rv: NAV q: DYN	3 dB
2	GC	1.5-35 m	rv: NAV **q: DYN + noise**	3 dB
3	GC	1.5-35 m	rv: NAV q: DYN + noise	**6 dB**
4	GC	1.5-35 m	rv: NAV q: DYN + noise	**12 dB**
5	GC	1.5-35 m	rv: NAV **q: IP + noise**	12 dB
6	**GC + att. cont.**	1.5-35 m	rv: NAV q: IP + noise	12 dB
7	GC + att. cont.	1.5-35 m	rv: NAV **q: NAV**	12 dB

5.1. *IP analysis*

By taking the simulations outline in Table 8, it can be noticed that the IP block is mainly tested by increasing the noise added to the input images processed to retrieve the relative pose. Namely, the synthetic images are noised by applying the VIS sensor noise [17] (including also the fixed pattern noises), using the parameters of the Chameleon 3 sensor. Keeping constant the sensor characteristics, the noise level can be adjusted by tuning the AGC Gain. Namely, the noise has been increased by raising the ADC Gain from 3 dB to 12 dB, i.e., reducing the estimated Signal to Noise Ratio (SNR) from about 37 dB (3 dB gain) to about 33 dB (12 dB gain). The maximum noise level of 12 dB of ADC Gain has been selected since it is one of the highest noise levels that allows retrieving reliable outputs from the IP step without leading to high errors within the training distance range, as assessed in [17]. In the following, the analyses will be detailed only for the worst cases (i.e., the highest noise level) for brevity. It is pointed out that the noise levels simulated are specific for the camera model adopted and that the IP pipeline shall be re-evaluated and, if needed, re-tuned to be compliant with other camera models leveraging the sensor parameters and settings defined for the specific mission to generate a representative noise through the noise generator adopted within this work, due to the high variability of the VIS sensor noise and SNR as a function of the sensor itself, its settings, and the imaging conditions.

The assessment of the robustness of the IP step has been performed by collecting the outputs for all the episodes simulated, resulting in about 100000 samples retrieved for both the constrained target pointing (ID 4 – 5 in Table 8) and controlled pointing (ID 6 – 7 in Table 8) conditions. It is worth remarking that the images are randomised between each episode due to the adaptive guidance involved in the simulation loop, resulting in an ideal scenario to prove the robustness and reliability of the YOLOv8s-pose-based pose estimation algorithm. The performances have been evaluated by retrieving for each frame the normalized relative distance error e_t and the relative quaternion error e_q as in Eq. (11) and Eq. (12), respectively, as defined in [15].

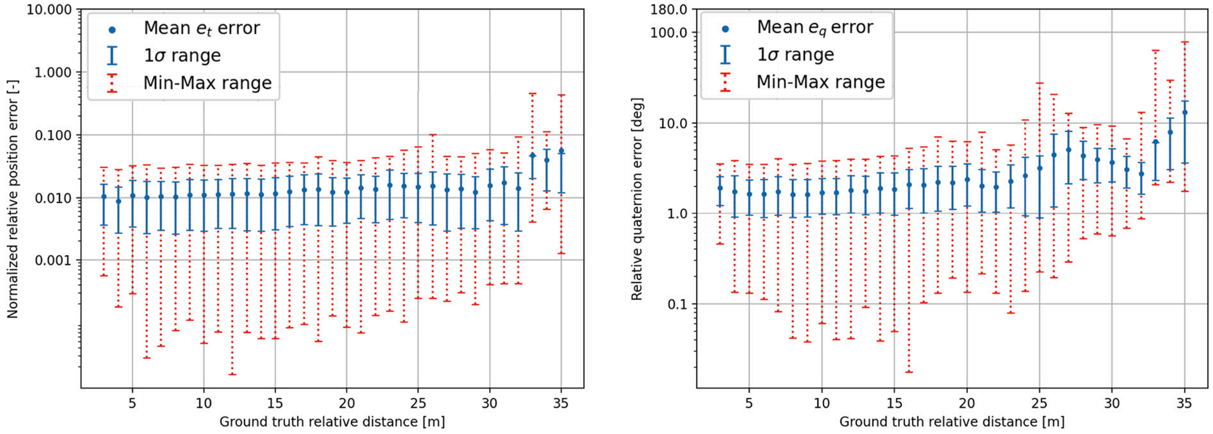


Fig. 7. YOLOv8s-pose-based IP robustness analysis in target-pointing scenario.

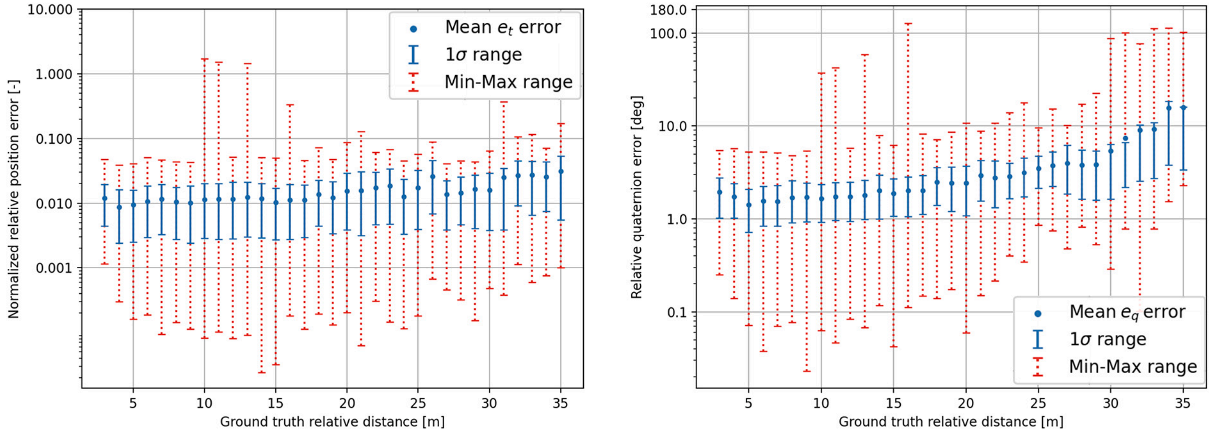


Fig. 8. YOLOv8s-pose-based IP robustness analysis in non-target-pointing scenario.

$$e_t = \frac{E_t}{||\mathbf{t}_C||} = \frac{||\mathbf{t}_C - \hat{\mathbf{t}}_C||}{||\mathbf{t}_C||} \tag{11}$$

$$e_q = 2 \cdot \arccos|\mathbf{q} \cdot \hat{\mathbf{q}}| \tag{12}$$

Where $\mathbf{t}_C$ and $\mathbf{q}$ are the ground truth pose vectors, while $\hat{\mathbf{t}}_C$ and $\hat{\mathbf{q}}$ are the estimated ones. Figs. 7 – 8 show the evaluation metrics (in logarithmic scale) and their range of variability as a function of the relative ground truth distance. Namely, the samples have been organized in batches by discretizing the ground truth distance in steps of 1 m. The mean value of each error metric is computed for each batch. The vertical lines in Figs. 7 – 8 give the range of values within the 1σ range (i.e., the values within the 16-84 percentiles), being the central trend of the distribution, and the range between the minimum and maximum error within each batch, to provide the full range of variability of the error measured.

Regarding the errors in the non-target-pointing scenario shown in Fig. 8 (ID 6 – 7 in Table 8), it can be noticed that there are four peaks of maximum error between 10 m and 16 m of relative distance for both the normalized position and the quaternion error. By looking at the generated images, it has been noticed that those peaks are due to a weak attitude control action that makes the target exit the FOV. The high errors correspond to the transition from inside to outside of the FOV and vice versa. During these transitions, there are images for which only a minimal portion of the target (e.g., from 1 to 3 keypoints) is within the FOV and, in those images, the accuracy of the keypoint regression became poor, and the errors in the retrieved relative pose increase. Notably, these high errors can be classified as outliers since they do not affect the central trend of the distributions. Overall, the normalized relative position error is below 5% (both the mean value and the 16-84 percentiles) for both the scenarios, except for distances higher than 33 m in the target pointing scenario (Fig. 7), where the error slightly increases (still limited to a maximum of less than 6%) due to images captured in low illumination conditions. The relative attitude error mean value and the upper bound of the 16-84 percentiles are below 10 degrees up to a maximum relative distance of 31 m. For ground truth relative distances higher than 31 m, the maximum error mean and upper bound value of the 1σ range is 11 degrees. Notably, the angular error is below 5 degrees up to 25 m of relative distance for both the scenarios. By comparing the plots in Figs. 7 – 8, it can be noticed that they are similar (except for the previously mentioned outliers due to the target exiting the FOV), proving that the IP architecture is robust with respect to those cases in which the target is within the FOV but not centered in the image. In conclusion, the analysis proved the robustness of the YOLOv8s-pose architecture, providing the error ranges analyzed over a wide range of images and scenarios using a medium-high level of noise in the processed images. It is worth mentioning that the cases analysed are the worst cases in terms of image noise, hence, the performances are expected to be even better in case the noise level in the input frames is lower (e.g., for the test cases referred as ID 1 – 3 in Table 8). For the sake of completeness, the comparison between the ground truth and estimated trajectory of the target with respect to the camera in a non-target-pointing condition for a single episode has been reported in Fig. 9 together with the normalized relative position error for each component. As expected from the analyses in [17], the errors along the camera boresight axis (i.e., the z-axis) are about one order of magnitude higher than those along the other two components. This behaviour is well-known and has already been noticed in other relative pose estimation algorithms that rely on keypoints (e.g., [46,12]), and it originates since errors in the estimation of the keypoints coordinate mostly affect the z-coordinates, causing a sort of “zoom-in” and “zoom-out” effect [17]. The x and y coordinates are less affected by the errors in the keypoints coordinates since, to have a relevant effect

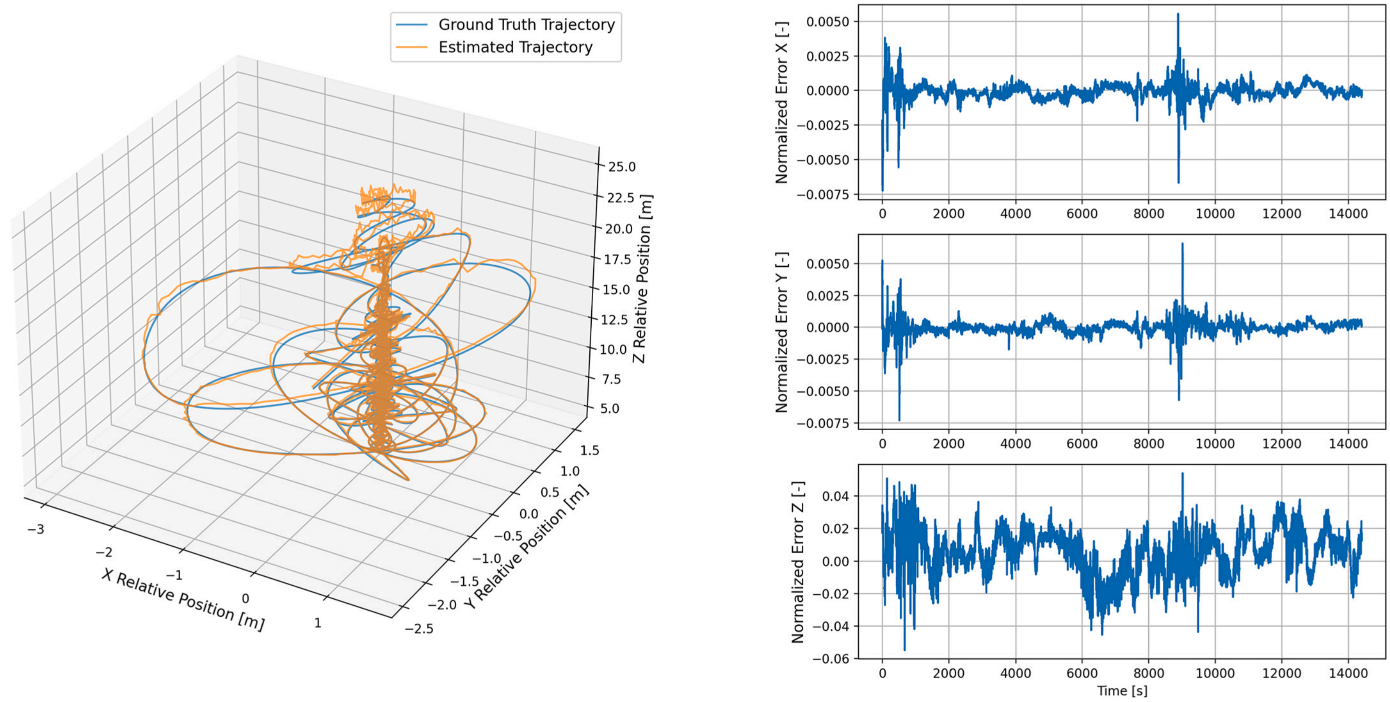


Fig. 9. Comparison of ground truth and estimated relative target trajectory with respect to the camera.

along these directions, all the errors in the keypoint coordinates should be in the same direction, which is highly not probable.

5.2. NAV analysis

The output of the NAV functional block is tightly related to the IP one, and the specific estimation error values depend on the episode configuration. This section reports some representative results, to analyse the behaviour of the filters and to discuss their role within the whole GNC chain. The performances have been evaluated as described in Eq. (13) and Eq. (14) for the relative position and attitude, respectively.

$$e_\rho = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \tag{13}$$

$$e_R = arccos\left(1 - \frac{tr(I - R^T\hat{R})}{2}\right) \tag{14}$$

where R and $\hat{R}$ are the ground-truth and the estimated relative rotation matrices. To cope with a time changing relative inter-satellite distance, the measurement covariance matrix of the filter has been made adaptive by means of the following relation:

$$R = r_y|\hat{\rho}| \tag{15}$$

with $r_y = 1 \times 10^{-2}$ being a fixed parameter. In this way, the measurement covariance matrix is scaled according to the estimated relative distance, without requiring further tuning during the inspection phase. With reference to 2, the two gains have been set to $\Sigma_u = 4 \times 10^{-5}$ and $\Sigma_y = 10^{-1}$. It is important to underline that these tuning settings are not the best possible settings, but they were found to offer reasonable performances is a wide operational scenario. It is also essential to remark that these settings were kept constant throughout all the simulations performed for these analyses. The results reported in Fig. 10 are related to a simulation showing nominal IP performances. It is possible to notice that the percentage of image-based outlier measurements is extremely low, and there are no failures within the IP block. It can be noticed that the filter is beneficial in both cases, since it reduces the estimation error of the IP functional block. It can also be noticed that the presence of the filter is effective in mitigating the error peaks that occur during the episode. The overall position and attitude estimation decrease by 30% and 7%, respectively. Even if the position error reduction is higher in percentage, it is important to stress the fact that its magnitude is always extremely low, always below 5% of the relative inter-satellite range.

However, to better understand the importance of the filter within the GNC chain, it is necessary to consider a test case in which the IP block fails to provide meaningful measurements, for instance when the target is no longer in the FOV. Such example is reported in Fig. 11, in which it is possible to notice that the IP stops working for 200 *s* towards the end of the simulation. This failure is due to the fact that the target is outside the camera's field of view, making the IP step to provide no measurements to the NAV block. Nevertheless, the filter is able to mitigate this shortcoming. It is important to stress the fact that the sudden increase in the attitude estimation error is partially due to the lack of measurements and partially due to the fact that the attitude controller is enforcing a high angular rate to the chaser spacecraft, thus enhancing the error propagation within the kinematics model embedded in the filter.

5.3. Testing analysis: GC

In this section, the results obtained in the test campaign described in Table 8 are analysed focusing on the performance of the guidance and control agent. Considering the trained agent studied in Sec. 4 and its relative performances summed up in Table 7, it is reasonable to expect that at least 4 on 5 episodes would succeed in reaching the 100% of map or, at least, remaining inside the trajectory limit ranges. As explained in Sec. 3 and Sec. 4, the action output of the guidance and control agent is shaped as an acceleration vector; in particular this vector has a fixed module of 0.001 m/s2. Therefore, considering the average simulation episode time and the percentage of thrust used, the ΔV budget can be easily computed by integrating the acceleration module. In general, considering a worst-case scenario in which for all the 4 h time of inspection the control action is not null, the maximum ΔV would be equal to 14.4 m/s.

The results are reported in Table 9 and show that the agent performance remains more or less constant during all the tests, proving the great robustness of the trained model. Even if the sample of episodes

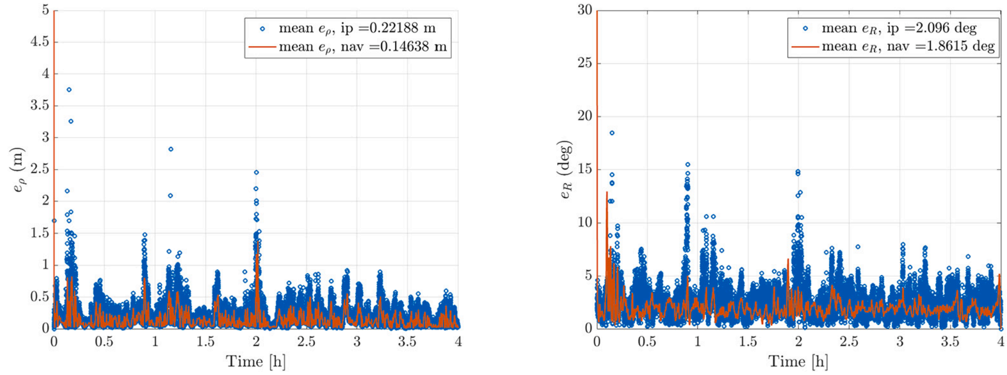


Fig. 10. Relative translation (left) and rotation (right) filter estimation errors in non-target-pointing scenario, nominal IP behaviour.

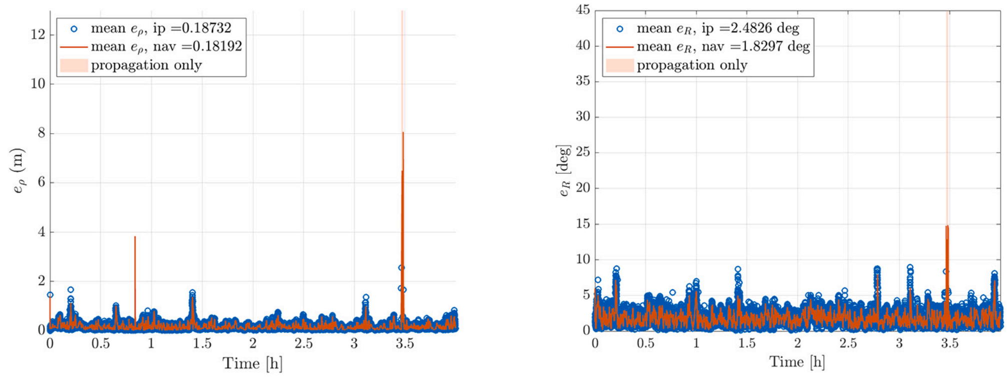


Fig. 11. Relative translation (left) and rotation (right) filter estimation errors in non-target-pointing scenario, off-nominal IP behaviour.

Table 9
GNC pipeline results.

ID	Scr.	Map	Time	Thr.	Ending Conditions			
					DMAX	*DMIN*	*MAP*	*TIME*
1	372.2	81.7%	2.7 h	48.3%	1	-	2	2
2	341.2	81.3%	2.6 h	40.7%	1	-	3	1
3	437.6	86.8%	3.6 h	33.1%	1	-	1	3
4	414.4	93.7%	3.1 h	34.5%	1	-	3	1
5	378.2	89.5%	2.7 h	28.6%	1	-	3	1
6	505.8	84.3%	3.2 h	29.4%	2	-	-	3
7	574.2	96.5%	4.0 h	53.2%	-	-	-	5

is short, just up to 5 for each simulation, the fact that the metrics levels are almost the same in all the tests supports the findings. Moreover, the fact that in some simulations the average metric is lower or higher with respect to the others is then connected to the single episode and not to the average performance. Indeed, it is easy to notice how, on average, the evaluation metrics respect the standards of the nominal ones. Considering the ending conditions, it is interesting to notice that, differently from the previous ones, only the ID:6 never gets the maximum map level. This may be caused by the additional attitude control aspect in the environment, which, not guaranteeing anymore a fixed target pointing, may slightly lower the agent's performance.

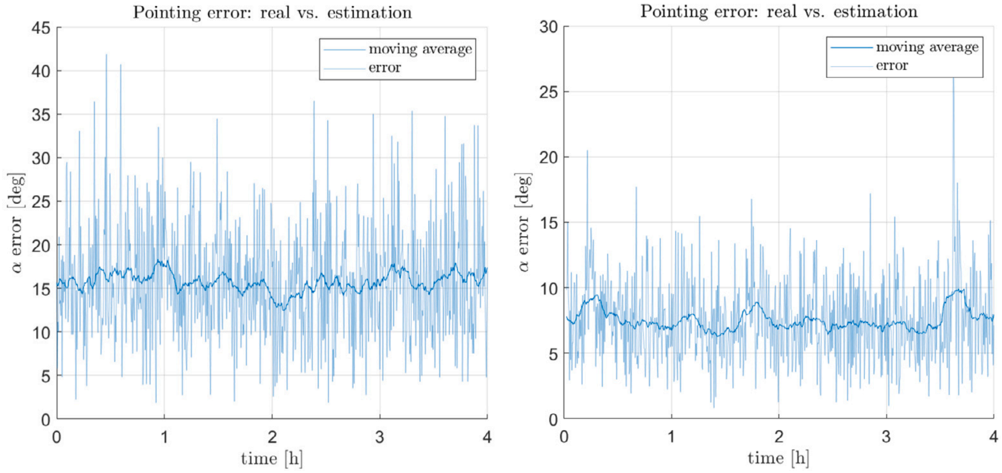
It is important to notice also the behaviour in ID:7, when the attitude information needed by the GC is retrieved by the rotational navigation filter. In particular, all episodes end exceeding the time constraint of 4 h, even if achieving a very high mapping level. This result shows that the agent benefits from the rotational filter which, indeed, reduces the average level of the pointing information input, as summed up in Table 10 and depicted in Fig. 12. It can be noticed that the output coming from the rotational navigation filter almost halves the pointing error associated to the chaser-target relative attitude, increasing then the reliability of the guidance and control block. Indeed, this behaviour is reflected in the simulation metrics as showed in Table 9 for ID:7, in which the score and map level reach the maximum results among the simulation investigated. This increased robustness, anyway, favours the survival of the spacecraft around the object instead of the final acquisition of the map, nevertheless it is, in average, higher than all other cases.

Table 10
Average pointing error in GC input throughout the different step of the GNC pipeline tests.

Pointing technique	DYN+noise	IP+noise	NAV
Average Pointing Error	10-20°	10-20°	5-10°

Two examples of results acquired throughout the test analysis are shown in the next pages. In Fig. 14, one episode of ID:5 test simulation is outlined; in Fig. 15, instead, one episode of ID:6 test. In both figures, the relative chaser-target position and velocity (highlighting when the agent takes active control on the chaser), the simulation metrics (score, map level and thrust percentage), and the trajectory around the object are plotted. From the graphs, it is easy to notice how the agent tries to keep the spacecraft at an average relative position and velocity value, actively controlling it only these two parameters start to drift away from this value.

One last analysis, shown in Fig. 13 has been carried out on the comparison of the agent behaviour when acting in the GNC environment or in the training environment defined in Sec. 4, considering fixed target pointing. In blue is depicted the relative position and velocity when the agent acts in the GNC algorithm, thus exploiting the input of the IP and NAV blocks. On the contrary, in orange, is the trajectory's variable followed by the chaser starting from the same initial conditions, but taking as input the real values as happening during the training phase. It can be noticed that, even if they clearly fly-around the target with two different trajectories, they still accomplish the same objective. This means that for the agent knowing exactly its current state is not strictly impor-



(a) Image Processing (b) Navigation Filter

Fig. 12. Error between real and estimated camera pointing: comparison between IP and rotational NAV filter performances.

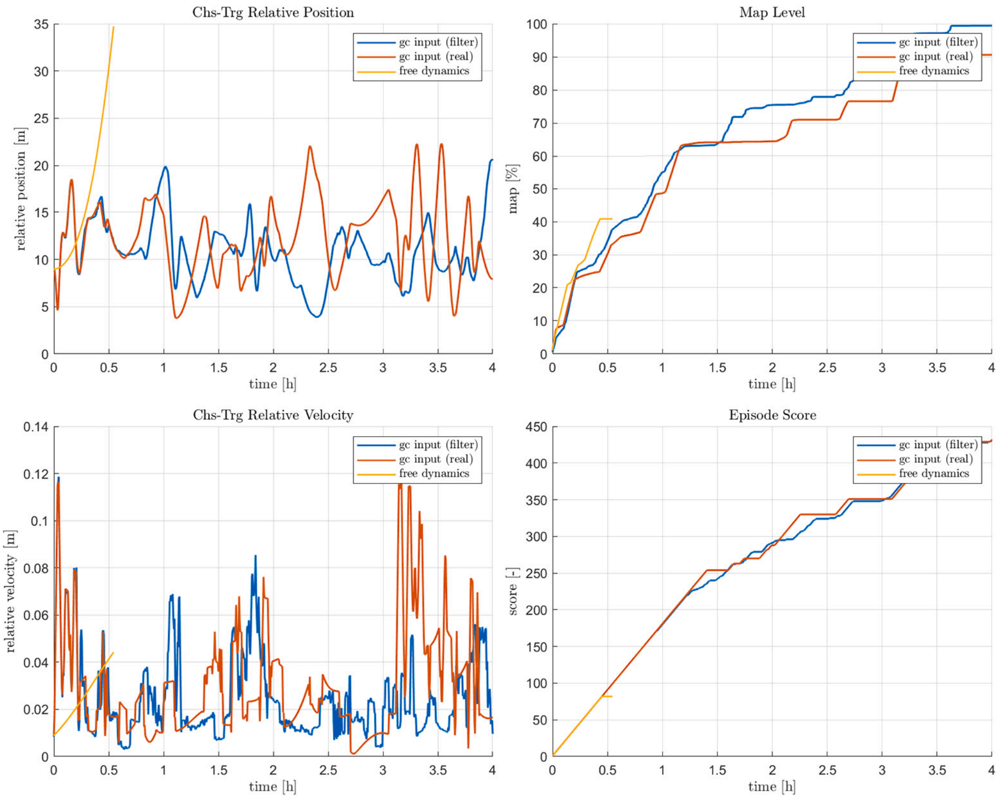


Fig. 13. Comparison of the trajectory generated by the GC agent within the GNC algorithm framework and the training environment defined in Sec. 4.

tant as long as the input represents a reasonable state condition. This very powerful feature is a direct consequence of the exploitation of a guidance and control policy approximator trained through deep reinforcement learning. For sake of clarity, the yellow line in Fig. 13 depicts the *free-dynamics* state, demonstrating again how an uncontrolled trajectory is totally unable to even approach an intermediate result. In the end, considering the overall results, the DRL trained GC agent seems to be perfectly capable of interacting with realistic IP and NAV blocks in a GNC closed-loop pipeline.

6. Final remarks

This paper proposes an innovative GNC algorithm architecture which has been presented in all of its components in the previous sections. The performance of the algorithm is investigated analysing the results with respect to the different blocks: image processing, navigation filter, and guidance and control. In particular, the guidance and control block is treated in depth starting from the analysis of the training problem shaped for the TANGO environment, up to the extensive testing and sensitivity analysis campaign. Once the agent's level of robustness and flexibility has been assessed, the developed block was inserted and interfaced into the image-based navigation tool. As shown in the previous pages, the trained agent exhibits a stable and unaltered level of performance throughout all the steps of the testing simulations summed up in Table 8, although some assumptions were quite distant with respect to the environmental model in which the GC agent has been trained, i.e. controlled chaser's attitude dynamics. The policy learned is robust to all the uncertainties and model differences introduced in the new GNC

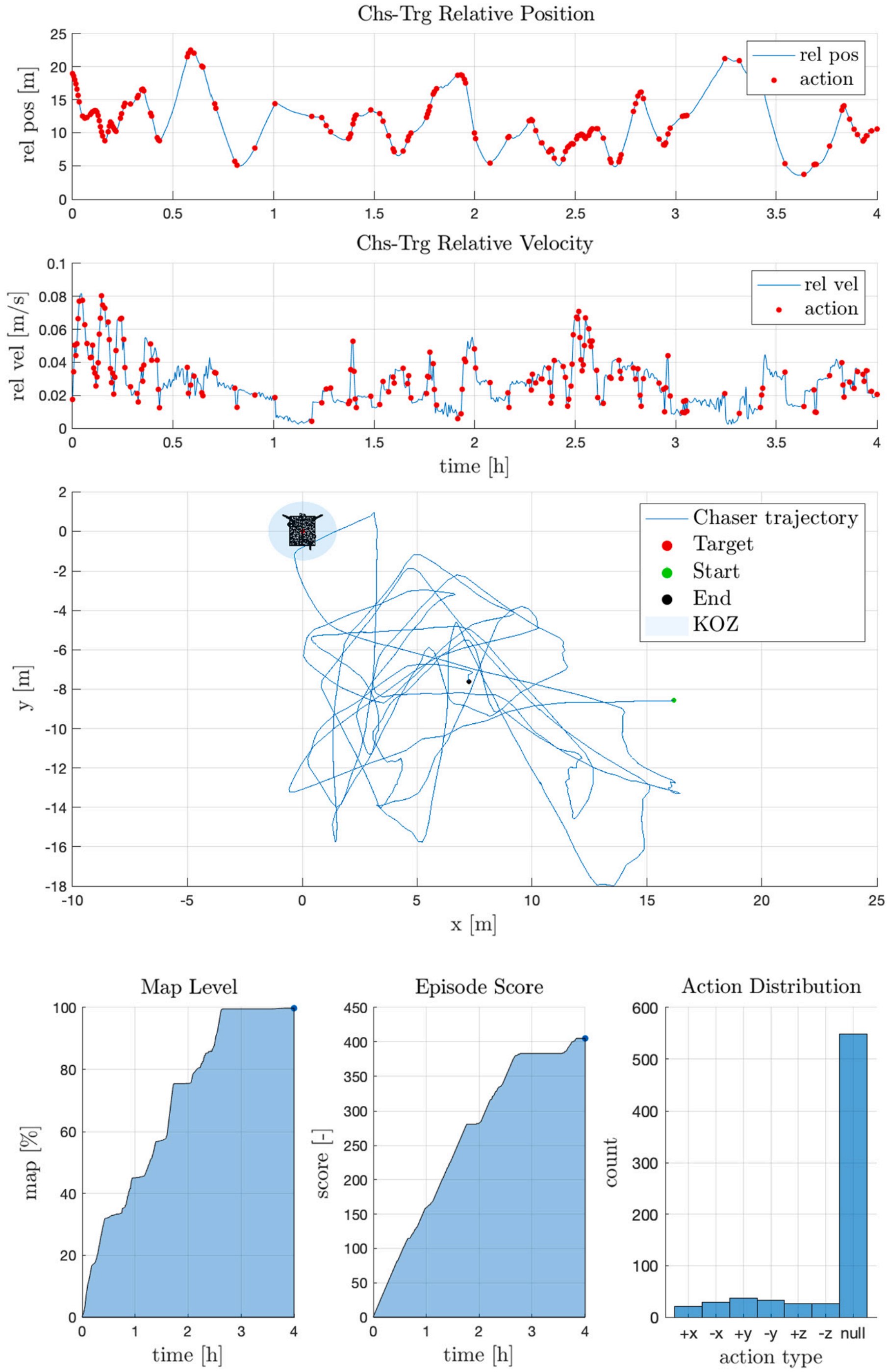


Fig. 14. Example of Chaser-Target trajectory and evaluation metrics in the *fixed* target pointing case.

framework: this is demonstrated by the trajectory path that the agent follows - or at least tries to follow - at each simulated episode. Moreover, the intrinsic estimation errors generated by the image processing and navigation filter do not affect the performances even if they are greater or comparable to more classical navigation techniques. Afterwards, both image processing and navigation filter performances have been analysed. The YOLOv8s-pose-based relative pose estimation algorithm included in the IP block provided highly accurate measures to the GNC scheme. The assessment of the performances of the IP step has been performed by evaluating the pose estimation scores measured over a wide range of scenarios (i.e., about 100000 images). The outcomes confirmed the strong robustness of the proposed approach, with a mean relative angular error below 5 degrees up to 25 m of relative distance (reaching a maximum error of about 11 degrees) and normalized mean relative position errors below 5% up to 33 m of relative distance (for the simulated VIS sensor). Notably, the error trends are similar for the perfect target pointing conditions and controlled attitude cases, proving that the pose estimation pipeline is not affected by the position of the target within the FOV. The navigation filters downstream the IP functional block enhance the robustness of the GNC chain. It has been shown that the beneficial effect of the filters is twofold: first of all they improve the accuracy of the pose estimate which is fed to the GC block, and further they enable operations even in case of faulty IP measurements, for example when the target exits the camera FOV. Lastly, the outcome of the tests highlights the fact that the output of the navigation filters is less oscillatory than the one of the IP block, which in turns lead to improved

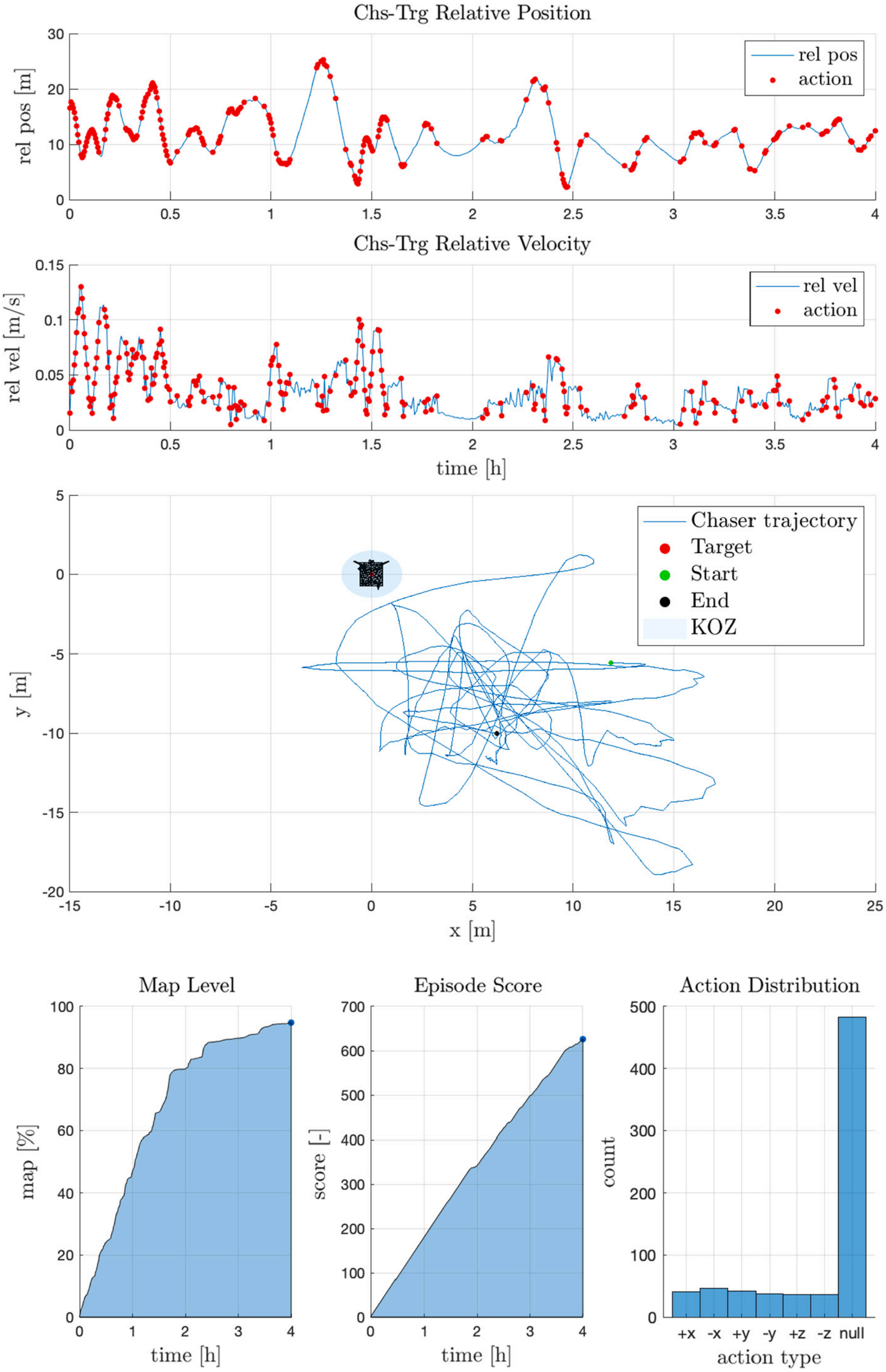


Fig. 15. Example of Chaser-Target trajectory and evaluation metrics in the *controlled* target pointing case.

mapping performances. In conclusion, even if some points still remain open, e.g. coupling compatibility between trajectory and attitude control, absolute attitude knowledge assumption, etc., the main objective of this analysis, as defined in Sec. 1, may be considered achieved with good success. Of course, these results have strengthened the awareness of having a powerful tool, that, at this point, feels the need of being tested also in a realistic or hardware-in-the-loop environment. This may be a complex path, which may also require specific facilities but, at the same time, may be greatly useful in understanding the real applicability of such algorithms. The activity may be subdivided into two parallel analyses: one aimed to study and optimize the bread-boarding of the image processing, navigation, guidance and control algorithms, the other more related to the development of a testing facility equipped with a target object mock-up, an illumination system and a VIS/TIR camera device which may simulate the chaser behaviour if placed on a controllable robotic arm. Thanks to this twofold research, it would be possible to concretely evaluate the performance of a realistic image and AI-based GNC system.

CRediT authorship contribution statement

Andrea Brandonisio: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Michele Bechini:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Gaia Letizia Civardi:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis. **Lorenzo Capra:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis. **Michèle Lavagna:** Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

In the person of Andrea Brandonisio, this publication resulted from research supported by *Fondazione Fratelli Confalonieri*, Milano.

Data availability

Data will be made available on request.

References

[1] A. Flores-Abad, O. Ma, K. Pham, S. Ulrich, A review of space robotics technologies for on-orbit servicing, Prog. Aerosp. Sci. 68 (2014) 1–26, https://doi.org/10.1016/j.paerosci.2014.03.002, https://www.sciencedirect.com/science/article/pii/S0376042114000347.

[2] M. Shan, J. Guo, E. Gill, Review and comparison of active space debris capturing and removal methods, Prog. Aerosp. Sci. 80 (January) (2016) 18–32, https://doi.org/10.1016/j.paerosci.2015.11.001.

[3] D. Izzo, M. Märtens, B. Pan, A survey on artificial intelligence trends in spacecraft guidance dynamics and control, Astrodynamics 3 (2019) 287–299.

[4] M. Tipaldi, R. Iervolino, P.R. Massenio, Reinforcement learning in spacecraft control applications: advances, prospects, and challenges, Annu. Rev. Control 54 (2022) 1–23, https://doi.org/10.1016/j.arcontrol.2022.07.004, https://www.sciencedirect.com/science/article/pii/S136757882200089X.

[5] K. Hovell, S. Ulrich, Deep reinforcement learning for spacecraft proximity operations guidance, J. Spacecr. Rockets 58 (2) (2021) 254–264, https://doi.org/10.2514/1.A34838.

[6] S. Silvestrini, L.P. Cassinis, R. Hinz, D. Gonzalez-Arjona, M. Tipaldi, P. Visconti, F. Corradino, V. Pesce, A. Colagrossi, Chapter fifteen - modern spacecraft gnc, in: V. Pesce, A. Colagrossi, S. Silvestrini (Eds.), Modern Spacecraft Guidance, Navigation, and Control, Elsevier, 2023, pp. 819–981.

[7] D. Izzo, G. Meoni, P. Gómez, D. Dold, A. Zoechbauer, Selected trends in artificial intelligence for space applications, arXiv:2212.06662, 2022.

[8] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations, Prog. Aerosp. Sci. 93 (2017) 53–72, https://doi.org/10.1016/j.paerosci.2017.07.001.

[9] M. Kisantal, S. Sharma, T.H. Park, D. Izzo, M. Märtens, S. D'Amico, Satellite pose estimation challenge: dataset, competition design, and results, IEEE Trans. Aerosp. Electron. Syst. 56 (5) (2020) 4083–4098.

[10] T.H. Park, M. Märtens, M. Jawaid, Z. Wang, B. Chen, T.-J. Chin, D. Izzo, S. D'Amico, Satellite pose estimation competition 2021: results and analyses, Acta Astronaut. 204 (2023) 640–665, https://doi.org/10.1016/j.actaastro.2023.01.002, https://www.sciencedirect.com/science/article/pii/S0094576523000048.

[11] V. Lepetit, F. Moreno-Noguer, P. Fua, Epnp: an accurate o(n) solution to the pnp problem, Int. J. Comput. Vis. 81 (2) (2009) 155–166, https://doi.org/10.1007/s11263-008-0152-6.

[12] B. Chen, J. Cao, A. Parra, T.-J. Chin, Satellite pose estimation with deep landmark regression and nonlinear pose refinement, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 1–9.

[13] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudillière, D. Aouada, A survey on deep learning-based monocular spacecraft pose estimation: current state, limitations and prospects, Acta Astronaut. 212 (2023) 339–360, https://doi.org/10.1016/j.actaastro.2023.08.001, https://www.sciencedirect.com/science/article/pii/S0094576523003995.

[14] M. Bechini, G. Gu, P. Lunghi, M. Lavagna, Robust spacecraft relative pose estimation via cnn-aided line segments detection in monocular images, Acta Astronaut. 215 (2024) 20–43, https://doi.org/10.1016/j.actaastro.2023.11.049.

[15] S. Sharma, J. Ventura, S. D'Amico, Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous, J. Spacecr. Rockets 55 (6) (2018) 1414–1429, https://doi.org/10.2514/1.A34124.

[16] G. Gu, B.S. Ko, S.H. Go, S. Lee, J. Lee, M. Shin, Towards real-time and light-weight line segment detection, CoRR, arXiv:2106.00186, 2021.

[17] M. Bechini, Monocular vision for uncooperative targets through AI-based methods and sensors fusion, Ph.D. thesis, Politecnico di Milano, 2024, https://hdl.handle.net/10589/216793.

[18] M. Kisantal, S. Sharma, T.H. Park, D. Izzo, M. Märtens, S. D'Amico, Spacecraft Pose Estimation Dataset (SPEED), Zenodo, 2019, https://doi.org/10.5281/zenodo.6327546.

[19] T.H. Park, M. Märtens, G. Lecuyer, D. Izzo, S. D'Amico, Next Generation Spacecraft Pose Estimation Dataset (SPEED+), Zenodo, Oct. 2021, https://doi.org/10.25740/wv398fc4383.

[20] B. Gaudet, R. Linares, R. Furfaro, Adaptive guidance and integrated navigation with reinforcement meta-learning, Acta Astronaut. 169 (2020) 180–190, https://doi.org/10.1016/j.actaastro.2020.01.007, https://www.sciencedirect.com/science/article/pii/S0094576520300072.

[21] B. Gaudet, R. Linares, R. Furfaro, Six degree-of-freedom body-fixed hovering over unmapped asteroids via lidar altimetry and reinforcement meta-learning, Acta Astronaut. 172 (2020) 90–99, https://doi.org/10.1016/j.actaastro.2020.03.026, https://www.sciencedirect.com/science/article/pii/S0094576520301545.

[22] B. Gaudet, R. Linares, R. Furfaro, Deep reinforcement learning for six degree-of-freedom planetary landing, Adv. Space Res. 65 (7) (2020) 1723–1741, https://doi.org/10.1016/j.asr.2019.12.030.

[23] A. Scorsoglio, R. Furfaro, R. Linares, M. Massari, Relative motion guidance for near-rectilinear lunar orbits with path constraints via actor-critic reinforcement learning, Adv. Space Res. 71 (1) (2023) 316–335, https://doi.org/10.1016/j.asr.2022.08.002.

[24] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, Design of relative trajectories for in orbit proximity operations, Acta Astronaut. 145 (2018) 342–356, https://doi.org/10.1016/j.actaastro.2018.01.062, https://www.sciencedirect.com/science/article/pii/S0094576517315679.

[25] F. Capolupo, P. Labourdette, Receding-horizon trajectory planning algorithm for passively safe on-orbit inspection missions, J. Guid. Control Dyn. 42 (5) (2019) 1023–1032, https://doi.org/10.2514/1.G003736.

[26] G. Borelli, G. Gaias, Y. Nakajima, C. Colombo, V. Capuano, F. Saggiomo, G. Leccese, S. Natalucci, Mission analysis and guidance and control for the speye inspection cubesat, Acta Astronaut. 220 (2024) 75–87, https://doi.org/10.1016/j.actaastro.2024.04.025, https://www.sciencedirect.com/science/article/pii/S0094576524002248.

[27] S. Silvestrini, M. Lavagna, Deep learning and artificial neural networks for spacecraft dynamics, navigation and control, Drones 6 (10) (2022) 270.

[28] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, 2nd edition, MIT Press, Cambridge, Massachusetts, USA, 2018.

[29] L. Pasqualetto Cassinis, R. Fonod, E. Gill, Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft, Prog. Aerosp. Sci. 110 (2019) 100548, https://doi.org/10.1016/j.paerosci.2019.05.008, https://www.sciencedirect.com/science/article/pii/S0376042119300302.

[30] L. Pasqualetto Cassinis, R. Fonod, E. Gill, I. Ahrns, J. Gil-Fernández, Evaluation of tightly- and loosely-coupled approaches in cnn-based pose estimation systems for uncooperative spacecraft, Acta Astronaut. 182 (2021) 189–202, https://doi.org/10.1016/j.actaastro.2021.01.035.

[31] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, M. Grassi, Autonomous relative navigation around uncooperative spacecraft based on a single camera, Aerosp. Sci. Technol. 84 (2019) 1070–1080, https://doi.org/10.1016/j.ast.2018.11.042, https://www.sciencedirect.com/science/article/pii/S1270963818317346.

[32] A. Brandonisio, L. Capra, M. Lavagna, Deep reinforcement learning spacecraft guidance with state uncertainty for autonomous shape reconstruction of uncooperative target, Adv. Space Res. (2023), https://doi.org/10.1016/j.asr.2023.07.007.

[33] L. Capra, A. Brandonisio, M. Lavagna, Network architecture and action space analysis for deep reinforcement learning towards spacecraft autonomous guidance, Adv. Space Res. 71 (9) (2023) 3787–3802, https://doi.org/10.1016/j.asr.2022.11.048.

[34] T. Plachetka, POV Ray: persistence of vision parallel raytracer, in: Proc. of Spring Conf. on Computer Graphics, Budmerice, Slovakia, vol. 123, 1998, p. 129.

[35] M. Bechini, M. Lavagna, P. Lunghi, Dataset generation and validation for spacecraft pose estimation via monocular images processing, Acta Astronaut. 204 (2023) 358–369, https://doi.org/10.1016/j.actaastro.2023.01.012, https://www.sciencedirect.com/science/article/pii/S0094576523000127.

[36] A. Brandonisio, Ai-based guidance for spacecraft proximity operations around uncooperative targets, Ph.D. thesis, Politecnico di Milano, 2024.

[37] P. Lunghi, M. Ciarambino, M. Lavagna, A multilayer perceptron hazard detector for vision-based autonomous planetary landing, Adv. Space Res. 58 (1) (2016) 131–144, https://doi.org/10.1016/j.asr.2016.04.012, https://www.sciencedirect.com/science/article/pii/S0273117716301284.

[38] J. Guarneri, Autonomous optical navigation and attitude estimation system tested on artificially generated images, 7 2020.

[39] A. Rivolta, P. Lunghi, M. Lavagna, Gnc & robotics for on orbit servicing with simulated vision in the loop, Acta Astronaut. 162 (2019) 327–335, https://doi.org/10.1016/j.actaastro.2019.06.005, https://www.sciencedirect.com/science/article/pii/S0094576518308968.

[40] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Dataset for Monocular Pose Estimation, Zenodo, 2022, https://doi.org/10.5281/zenodo.6499007.

[41] M. Konnik, J. Welsh, High-level numerical simulations of noise in ccd and cmos photosensors: review and tutorial, arXiv preprint, arXiv:1412.4031, 2014.

[42] K. Wei, Y. Fu, Y. Zheng, J. Yang, Physics-based noise modeling for extreme low-light photography, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2022) 8520–8537, https://doi.org/10.1109/TPAMI.2021.3103114.

[43] E.M.V. Association, Emva 1288 standard: standard for characterization of image sensors and cameras, in: EMVA, Tech. Rep. Release 4.0 Linear, 2021.

[44] M. Bechini, L. Bianchi, M. Lavagna, Towards domain gap bridging via synthetic vis sensor model, in: 75^{th} International Astronautical Congress, 2024.

[45] M. Piccinin, S. Silvestrini, G. Zanotti, A. Brandonisio, P. Lunghi, M. Lavagna, Argos: calibrated facility for image based relative navigation technologies on ground verification and testing, in: 72^{nd} International Astronautical Congress, 2021.

[46] M. Piazza, M. Maestrini, P. Di Lizia, Monocular relative pose estimation pipeline for uncooperative resident space objects, J. Aerosp. Inform. Syst. 19 (9) (2022) 613–632, https://doi.org/10.2514/1.I011064.

[47] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Dataset for Region of Interest Estimation and Semantic Segmentation, Zenodo, 2022, https://doi.org/10.5281/zenodo.6507864.

[48] T. Möller, B. Trumbore, Fast, minimum storage ray-triangle intersection, J. Graph. Tools 2 (1) (1997) 21–28, https://doi.org/10.1080/10867651.1997.10487468.

[49] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Wireframe Dataset Model for Line Segments Detection, Zenodo, 2022, https://doi.org/10.5281/zenodo.6372849.

[50] E. Samkari, M. Arif, M. Alghamdi, M.A. Al Ghamdi, Human pose estimation using deep learning: a systematic literature review, Mach. Learn. Knowl. Extr. 5 (4) (2023) 1612–1659, https://doi.org/10.3390/make5040081.

[51] D. Simon, Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches, John Wiley & Sons, 2006.

[52] G.L. Civardi, M. Lavagna, Filtering techniques assessment towards pose estimation enhancement for image-based proximity navigation with uncooperative space objects, in: Aerospace Europe Conference 2023-Joint 10th EUCASS-9th CEAS Conference, 2023, pp. 1–14.

[53] D. Wang, B. Wu, E.K. Poh, D. Wang, B. Wu, E.K. Poh, Dynamic models of satellite relative motion around an oblate earth, in: Satellite Formation Flying: Relative Dynamics, Formation Design, Fuel Optimal Maneuvers and Formation Maintenance, 2017, pp. 9–41.

[54] N. Stacey, S. D'Amico, Analytical process noise covariance modeling for absolute and relative orbits, Acta Astronaut. 194 (2022) 33–47.

[55] A. Brandonisio, M. Lavagna, D. Guzzetti, Reinforcement learning for uncooperative space objects smart imaging path-planning, J. Astronaut. Sci. 68 (4) (2021) 1145–1169, https://doi.org/10.1007/s40295-021-00288-7.

[56] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv:1707.06347, 2017.

[57] B. Razgus, E. Mooij, D. Choukroun, Relative navigation in asteroid missions using dual quaternion filtering, J. Guid. Control Dyn. 40 (9) (2017) 2151–2166, https://doi.org/10.2514/1.G002805.

[58] J.L. Crassidis, F.L. Markley, Y. Cheng, Survey of nonlinear attitude estimation methods, J. Guid. Control Dyn. 30 (1) (2007) 12–28, https://doi.org/10.2514/1.22452.