Post-Print

When citing this work, cite the original published paper.

Permanent link to this version
http://hdl.handle.net/11311/1080211

Autonomous Relative Navigation Around Uncooperative Spacecraft Based on a Single Camera

Vincenzo Pesce*
Politecnico di Milano, via La Masa 34, Milano, Italy

Roberto Opromolla†
University of Naples, Piazzale Tecchio 80, Napoli, Italy

Salvatore Sarno‡
University of Campania - L. Vanvitelli, Via Roma 29, Aversa, Italy

Michèle Lavagna§
Politecnico di Milano, via La Masa 34, Milano, Italy

Michele Grassi¶
University of Naples, Piazzale Tecchio 80, Napoli, Italy

The interest of the space community toward missions like On-Orbit Servicing of functional satellite to extend their operative life, or Active Debris Removal to reduce the risk of collision among artificial objects in the most crowded orbital belts, is significantly increasing for both economical and safety aspects. These activities present significant technical challenges and, thus, can be enabled only by increasing the level of autonomy and robustness of space systems in terms of guidance, navigation and control functionalities. Clearly this goal requires the design and development of ad-hoc technologies and algorithms. In this framework, this paper presents an original architecture for relative navigation based on a single passive camera able to fully reconstruct the relative state between a chaser spacecraft and a non-cooperative, known target. The proposed architecture is loosely coupled, meaning that pose determination and full relative state estimation are entrusted to separate, but rigidly interconnected processing blocks. Innovative aspects are relevant to both the pose determination algorithms and the filtering scheme. Preliminary performance assessment is carried out by means of numerical simulations considering multiple realistic target/chaser relative dynamics and target geometries. Results allow demonstrating robustness against measurement error sources caused possibly by image processing as well as fast rotational dynamics.

---

*PhD Candidate, Department of Aerospace Science and Technology, `vincenzo.pesce@polimi.it`
†Post-Doctoral Researcher, Department of Industrial Engineering (DII), `roberto.opromolla@unina.it`
‡PhD Candidate, Department of Engineering `salvatore.sarno@unicampania.it`
§Associate Professor, Department of Aerospace Science and Technology, `michelle.lavagna@polimi.it`
¶Full Professor, Department of Industrial Engineering (DII), `michele.grassi@unina.it`

I. Introduction

The necessity to significantly enhance the level of autonomy of spacecraft to enable next-generation space missions has been recently acknowledged by major institutions like NASA [1]. In fact, autonomy allows to cope with the inability to rely on commands from ground control stations (due to communications latencies and black-outs), but it can also provide increased mission frequency, robustness, and reliability [2, 3]. In this framework, this paper investigates the problem of autonomous relative navigation between an active satellite (chaser) and another space object (target) orbiting in close-proximity. This activity is of interest to a variety of applications, namely Formation Flying [4], On Orbit Servicing (OOS) of functional satellites [5] or space station [6], active capture and removal of space debris (ADR) [7, 8]. Indeed, in such mission scenarios, the onboard processing unit of the chaser must be able to autonomously estimate its relative state ensuring both high accuracy and update-rate, thus being able to satisfy control requirements and minimize collision risks. Specific attention is addressed to the case of uncooperative targets which are particularly difficult to be approached. Consequently, advanced, ad-hoc, technological and algorithmic solutions shall be envisaged. With regards to the technological aspects, Electro-Optical (EO) sensors have been identified as the best option for relative navigation purposes when close-proximity maneuvers (e.g., rendezvous and docking) towards uncooperative targets are required [9]. In this respect, either active Light Detection and Ranging (LIDAR) systems or passive monocular and stereo cameras can be used. The selection of the navigation sensor must consider the resources available on board in terms of mass, electrical and processing power, on one side, the mission scenario and the costs to be sustained for design and development of the satellite system, on the other side [10]. This work is focused on the use of monocular cameras. Indeed, they provide advantages with respect to both LIDARs (lower hardware complexity, cost, weight and power consumption, possibility to be simultaneously used for supervised applications) and stereovision systems (much larger operational range, not limited by the size of the platform). Clearly, the adoption of monocular cameras shall involve the identification of strategies to cope with their operational drawbacks, like the sensitivity to adverse illumination conditions [11] (e.g., saturation under direct Sun illumination, or absence of light during eclipse), and the inability to provide direct three-dimensional (3D) measurements about the target. The problem of estimating the state of an uncooperative target using passive cameras can be tackled by exploiting either a loosely-coupled or a tightly-coupled relative navigation architecture. The tightly-coupled approach consists in directly processing natural features (e.g., corners [12], edges [13], or more complex descriptors [14, 15]), extracted from the acquired images, within a filtering scheme (e.g., a Kalman filter). Specifically, the Line-of-Sight (LOS) or the 3D position vector of each feature shall be included in the state vector [16, 17]. Consequently, the computational effort increases with the number of detected features. Moreover, tightly-coupled architectures are not robust in case of fast relative dynamics or if the target has complex geometry, since the capability to adequately track features can be compromised. On the other hand, it is important to underline that these approaches are the best option when dealing with uncooperative unknown targets, for which also information on the inertia parameters can be recovered [18, 19]. However, for the scenarios of interest to

this work, i.e., OOS and ADR, the uncooperative target is a known object. Hence, at least basic information about its geometry are assumed to be available and can be exploited within the developed relative navigation architecture. Under such condition, loosely-coupled architectures are typically preferred. This term is used to indicate the fact that the relative navigation architecture is composed of two separate, consecutive blocks. First, the pose determination block analyzes the acquired images to provide an independent estimate of the target/chaser relative position and attitude. Then, this pose estimate is used within the measurement equation of the Kalman Filter. In this paper, a loosely-coupled architecture for relative navigation based on monocular images as well as on the knowledge of a simplified target model is presented. Specifically, original contributions are provided with respect to the state of the art in terms of both pose determination and filtering solutions. With regards to the pose determination block, several monocular techniques have been recently proposed [9], and their performance has been tested using synthetic data [20], real images from past missions [21, 22] or experimental tests [23]. These studies have shown that main open challenges are related to the pose initialization, i.e., the initial estimation of the pose parameters carried out when no prior information is available about the target pose [9]. Specifically, the necessity to limit the computational load as well as to enhance robustness against variability of pose conditions has been identified. To this aim, a customized implementation of the RANdom SAmple Consensus (RANSAC) algorithm which exploits the Principle Component Analysis (PCA) and the knowledge of a simplified target model, is proposed in this paper. On the other hand, pose tracking, i.e., the problem of updating the pose parameters based on new measurements, is carried out exploiting the SoftPOSIT algorithm [24].

Once a pose estimate is available, this measurement is typically filtered to improve the state estimation. The EKF (Extended Kalman Filter) is the standard approach for space applications. This solution has been used also for several relative navigation techniques with different sensors architectures [18, 25–27]. Two different strategies exist for relative navigation filters architectures. Specifically, a single filter for both translation and rotation can be exploited or the two dynamics can be decoupled. The first approach is necessary when the measurements are a combination of relative translational position and angular velocity [18, 27] (i.e. measurements are directly the feature points in a tightly-coupled configuration). However, due to the non-linear nature of the rotational dynamics, the choice of the filter is limited to sub-optimal nonlinear approaches such as EKF or Unscented Kalman Filter (UKF). This implies a higher computational time (with respect to linear filters) and a degradation of the performance with non-nominal tuning and non-Gaussian measurement noise. Instead, if the measurements are directly the relative position and relative attitude, the two filters can be decoupled. In our case, the nature of the problem and the classical limited duration of proximity operations suggest to decouple the relative translational and rotational dynamics by neglecting external disturbances and spacecraft flexibility. Another important point to underline is that, typically, the determination of the relative position is usually more robust with respect to relative attitude estimation [9], which is more prone to errors and ambiguities. For this reason, a decoupled strategy in which the translational and rotational filters work independently, guarantees the control

of the proximity phase also in the case of poor relative attitude estimation. The proposed architecture exploits a linear H-∞ Filter for the translational motion and a 2nd Order Non-linear Filter on the Special Orthogonal group ($SO(3)$) for the rotational part. The H-∞ Filter guarantees robustness by minimizing the ∞-*norm* of the estimation error and it does not make any assumptions about the statistics of the process and measurement noise [28]. The implemented 2nd Order Non-linear Filter is based on a novel formulation on the $SO(3)$. This choice is justified by the fact that filters exploiting a dynamics directly expressed on the $SO(3)$ have shown better performance than filters designed using other attitude parameterizations [29–32].

The contribution of this paper and the main innovative aspects are summarized below:

- an original loosely-coupled architecture for relative navigation toward a non-cooperative, known target, based on a single passive camera, is proposed.
- Innovative aspects are relevant to both pose determination algorithms and filtering scheme.
- The critical problem of feature-model matching during the initial acquisition of the target pose is entrusted to an original strategy which exploits the PCA to classify the extracted features based on their distribution on the image plane.
- A decoupled structure of the navigation filter to guarantee close proximity phase control is proposed.
- A robust filter for the translational filter and an original formulation on Lie group for the rotational one are presented.

This activity lies in the framework of the project VINAG, funded by the Italian Space Agency, aimed at developing a highly integrated unit (including vision, inertial and GNSS systems) for autonomous absolute and relative navigation of spacecraft. The paper is organized as follows. The proposed relative navigation architecture is presented in Section II. The pose determination algorithms and the filtering scheme composing this architecture are described in detail in Section III and IV, respectively. Finally, Section V presents the simulation environment and the related results.

II. Monocular Relative Navigation Architecture

A block diagram describing the proposed architecture for relative navigation of a chaser satellite with respect to an uncooperative target is presented in Figure 1.

This architecture is loosely-coupled since it is composed of two separate blocks. The pose estimator processes the images acquired by a monocular camera to compute the target/chaser position and attitude parameters, which are then used as input for the navigation filter. Both the acquisition and tracking functions of the pose estimator exploit a simplified model of the target which is built off-line thanks to the knowledge of the target geometry (e.g., a CAD model). The navigation filter is also divided in two components responsible of the relative rotational and translational dynamics, respectively. A link between the two blocks is ensured by the fact that the updated target pose provided by the filter is used to initialize the tracking function of the pose estimator. Concerning the mathematical notation adopted in the
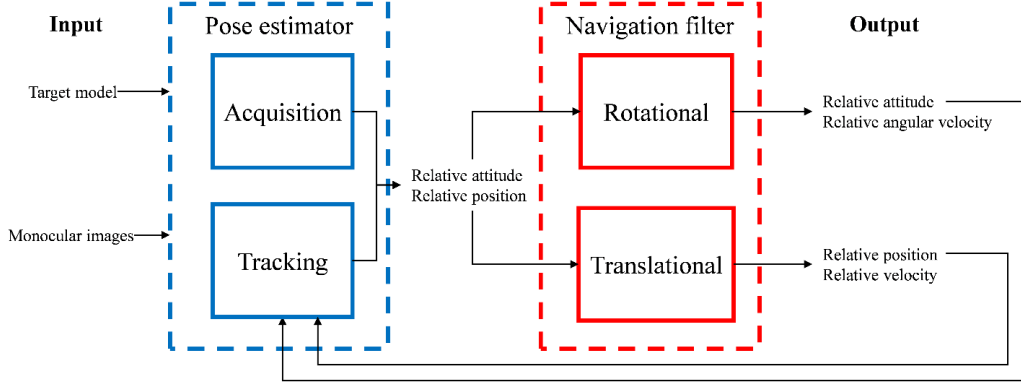
Fig. 1   Relative Navigation Architecture: Block Diagram

following sections to describe these blocks, italic style is used for any symbol, but vectors and matrices are highlighted in bold.

III. Pose Estimator

1. Coordinate Systems

In this paragraph, the coordinate systems used in the following sections are introduced. In particular, two spacecraft orbiting the Earth are considered, a *chaser* and a *target* respectively. The reference frames used in this formulation are: an Earth-centered, inertial reference frame, $\mathcal{I}$; a local-vertical, local-horizontal (LVLH) reference frame fixed to the chaser spacecraft center of mass, C with $\hat{\mathbf{x}}$ being a unit vector directed radially outward, $\hat{\mathbf{z}}$ normal to the spacecraft orbital plane, and $\hat{\mathbf{y}}$ completes the triad; a Cartesian body-fixed reference frame attached to the target spacecraft center of mass, $\mathcal{T}$. Without losing generality, C is assumed to be coincident with its body-fixed reference frame, as well as with the camera reference frame.

2. Pose Estimation Algorithm

The output of this block, i.e., the target/chaser pose, is parameterized by a rotation matrix ($\boldsymbol{R}_{\mathcal{T}C}$) which represents the attitude of the Camera Reference Frame (C) with respect to the Target body Reference Frame ($\mathcal{T}$), and a position vector ($\boldsymbol{t}$) directed from C to $\mathcal{T}$ (expressed in C). Both the acquisition and tracking functions of the pose estimator are entrusted to feature-based algorithms. Specifically, they allow estimating $\boldsymbol{R}_{\mathcal{T}C}$ and $\boldsymbol{t}$ by trying to optimize the matches between natural features extracted from the acquired images, and the target model. In this work, the attention is focused on point features, such as corners (which can be detected using standard image processing algorithms [12]). Consequently, the target model, generated off-line (i.e., on ground before mission starts) from the knowledge of the target geometry, will be a dataset of M 3D landmarks, i.e., the position vectors of the real corners of the target in $\mathcal{T}$ (generally speaking the landmarks can be 3D points of the target object which are more likely to be identified by image processing algorithms).

Clearly, the accuracy level attainable by this kind of approaches (typically known as model-based [33]) in the estimation of the pose parameters could be affected by the level of detail in the model. Typically, this effect will be negligible if the 3D landmarks' positions are taken from a CAD model which is highly-consistent to the target real geometry. Instead, performance could be negatively affected if the target has been damaged during its operative life, and thus it could be necessary to update the model on board [9]. A critical issue for both acquisition and tracking is the identification of correct correspondences between the extracted feature points (p) and the model's landmarks (P). Indeed, the problems of image-model matching and pose estimation are coupled and can be solved using iterative approaches. As regards the tracking step, the evolution of the pose parameters is followed by implementing the SoftPOSIT algorithm [24]. Specifically, the relative position and attitude parameters are determined by optimizing in a least-squares sense, a cost-function that is derived following the POSIT (Pose from Orthography and Scaling with Iterations) approach [34]. This method returns the pose by approximating the true perspective projection with a scaled orthographic projection but requires the assignment between image and model corners to be known. Therefore, a correspondence matrix is considered as an additional decisional variable, and it is built iteratively through Softassign-based routines [35]. As long as the optimization proceeds, the correspondence matrix tends to a zero-one matrix which univocally discriminates image-model associations, while the relative pose is refined to correct the initial assumption of scaled projection. SoftPOSIT requires an initial guess, which is set as the solution computed at the acquisition step. Hence, an original approach has been developed to derive the initial relative state of the target. The coupled problem of feature-matching and pose-estimation is entrusted to a Hypothesize-and-Test approach. First, a set of image-model correspondences (typically indicated as consensus set, S) is selected. These matches are used to compute a tentative pose solving the Perspective-n-Point (PnP) problem [36]. The size of the initial consensus set (n) is determined by the selected PnP solver. The estimated values of t and $R_{\mathcal{TC}}$ are then used to project all the 3D landmarks on the focal plane (thus generating a virtual image). At this point, the re-projected feature points are compared to the ones detected in the original image to try to obtain an enlarged consensus set (S^*). If the hypothesized correspondences are not confirmed (which means that the corresponding pose solution is not correct), the procedure shall be restarted selecting a new initial consensus set. On the other hand, if the number of matched feature points reaches a limit value (n_{lim}), the matching process is ended, and the pose parameters are refined based on the matches in S^*. The proposed Hypothesize-and-Test approach, summarized by the pseudo-code in Algorithm 1, is now analyzed step by step.

Algorithm 1 Hypothesize-and-Test approach

    Step 0: Image acquisition and processing

        Corner Detection

    Step 1: Hypothesize

        Select the initial consensus set S ($size(S) = n < n_{lim}$)

    Step 2: Test

        Compute t and $\boldsymbol{R}_{\mathcal{T}C}$ (PnP solver)

        Corner reprojection

        Check *correspondences*

        if *correspondences*=1 (correct initial matches) **then**

            Creation of S^*

        else

            Go back to Step 1

        end if

    Step 3: Pose refinement

        if size $S^* < n_{lim}$ **then**

            Go back to Step 2

        else

            Compute t and $\boldsymbol{R}_{\mathcal{T}C}$ (PnP solver)

        end if

    Step 4: Runtime check

        Evaluate *runtime*

        if *runtime* $\geq t_{lim}$ **then**

            Start again from Step 0

        end if

First, image acquisition and processing are carried out to extract a set of N_p 2D feature points (**Step 0**). Then, S must be generated assigning at least n matches between the N_p 2D feature points and the M 3D landmarks (**Step 1**). To this aim, a RANSAC-based approach [36] aided by PCA [37] is applied. Indeed, the purely random nature of RANSAC-based image-model matching strategy is not efficient and may produce an unacceptable runtime. So, the proposed method consists in building S considering only a subset (T) of the M 3D landmarks, i.e. those belonging to spacecraft components which are clearly separated with respect to the target center of mass (e.g., solar arrays, telescopes, antennas), and the corresponding 2D feature points in the image. The elements of T are selected off-line, depending

on the target geometry. On the other hand, the corresponding 2D feature points are identified exploiting information obtained applying the PCA. Indeed, if the target is fully visible in the camera FOV (this condition will always occur at the beginning of the close-proximity maneuver, when pose acquisition is required), the target *appendices* are very likely to be imaged far from the target centroid on the focal plane. PCA is a technique used to analyze multidimensional datasets. Specifically, it allows determining their principal directions by analyzing eigenvectors and eigenvalues of the associated covariance matrix. For instance, if N_p feature points are extracted in the image, the covariance matrix (Q) can be computed as follows:

$$Q = \frac{1}{N_p} \begin{bmatrix} \sum_{i=1}^{N_p}(u_i - u_c)^2 & \sum_{i=1}^{N_p}(u_i - u_c)(v_i - v_c) \\ \sum_{i=1}^{N_p}(u_i - u_c)(v_i - v_c) & \sum_{i=1}^{N_p}(v_i - v_c)^2 \end{bmatrix} \tag{1}$$

where (u_i, v_i) and $(u_c \; v_c)$ are the focal-plane coordinates of the ith extracted feature point and image centroid, respectively. The eigenvectors of Q identify two directions on the image plane ($\boldsymbol{d_1}$ and $\boldsymbol{d_2}$). Hence, the extracted 2D feature points can be classified in two datasets (B_1 and B_2) depending on their distances from the PCA axes (D_1 and D_2), computed as follows:

$$\begin{aligned} D_{1,2} &= |\boldsymbol{d_1} \cdot \boldsymbol{p_i}| \\ D_{2,1} &= |\boldsymbol{d_2} \cdot \boldsymbol{p_i}| \end{aligned} \qquad i = 1, ..., N_p \tag{2}$$

Specifically, B_1 and B_2 are defined as:

$$B_1 = \{\boldsymbol{p_i} : D_{1,i} > D_{th}\}, \quad B_2 = \{\boldsymbol{p_i} : D_{2,i} > D_{th}\} \tag{3}$$

where D_{th} is a distance-threshold on the image plane. It is defined as the mean Euclidean distance of the features from the image centroid. An example of the results of the off-line classification of the 3D landmarks, and PCA-driven on-line classifications of the feature points is shown in Figure 2.

At this point, S is generated by randomly selecting n matches between T and B (**Step 1**), where B (i) is the random choice between B_1 and B_2 if both the subsets contain more than n elements, (ii) coincides with B_1 if B_2 contains less than n elements, or (iii) coincides with B_2 if B_1 contains less than n elements. Clearly, if both B_1 and B_2 contain less than n feature points, it is highly probable that the camera is observing the target from a not favorable point of view (i.e., the principal directions of the target are not clearly imaged on the focal plane, as it would happen if the camera is pointed in a direction almost aligned to the one identified by target appendixes). If this occurs, the algorithm shall return to **Step 0**. A new image will be thus acquired to benefit, in terms of relative observation geometry, from the evolution of the relative rotational dynamics. This choice allows increasing the robustness of the proposed acquisition strategy since
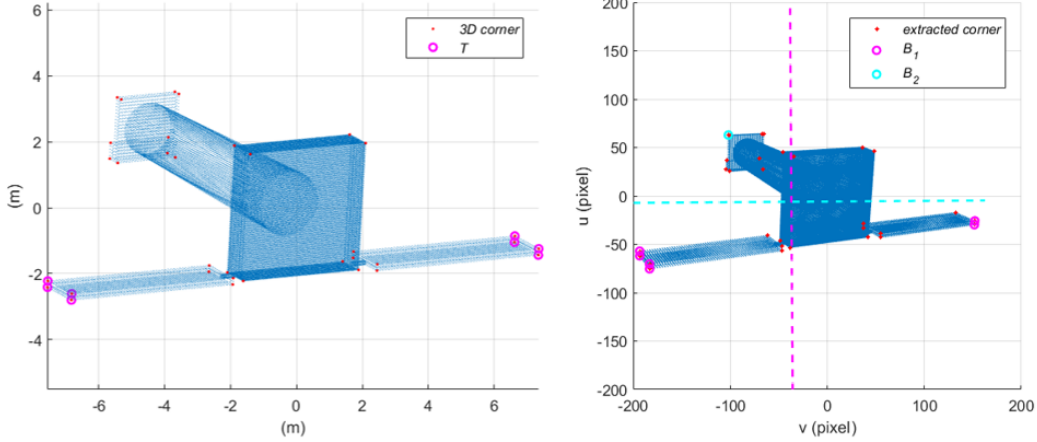
Fig. 2 Off-line feature point classification for the simplified target model (right). PCA-driven feature point classification for a simulated image (left). The target is the XMM-satellite.

not favorable observation geometries (which are likely to produce wrong matches, and, consequently, inaccurate pose estimates) are discarded. As soon as the initial consensus set is created, a tentative value for the pose parameters is evaluated and the assumed correspondences are verified to try enlarging S (**Step 2**). Concerning the PnP solver, the Efficient Perspective-n-Points (EPnP) method [38] has been selected, since it provides a closed-form solution based on a minimum of $n = 4$ image-model matches. For this reason, the EPnP solver is highly efficient (which is important when several runs are requested as in RANSAC-based iterative approaches [20]). This pose estimate is then used to verify the image-model correspondences. First, the 3D feature points are re-projected on the focal plane (p^r), using the classical equation of the perspective projection as shown hereunder

$$p^r = K(R_{\mathcal{T}C}P + t) \tag{4}$$

where K is intrinsic camera calibration matrix [39]. Clearly, the intrinsic camera parameters, i.e., focal length, skew factor, principal point coordinates and image distortion coefficient, are computed offline through an ad-hoc calibration procedure [40]. Then, a feature-matching algorithm is applied to verify the correspondences and, if possible, enlarge the consensus set. Specifically, each of the re-projected feature points is associated to the closest image feature according to the Euclidean metric. Hence, the obtained (p, p^r) couples are declared as confirmed matches if their Euclidean distance is below a threshold (D_m). The value of D_m is computed dynamically as the minimum detectable distance between two different bi-dimensional feature points. Indeed, it is reasonable to define this acceptance threshold according to the separation between the satellites: the farther the target from the camera, the smaller the distance between extracted feature points and the harder the image-model association. On the other hand, if the acquisition scenario is assumed to be same, the closer the target, the higher the separation among the extracted feature points, hence the D_m value. If one of the elements of S is not declared as a confirmed match, a new initial consensus set must be generated. Instead, if

9

additional matches are confirmed, they are added to S thus creating S^*. This process is iterated until the size of S^* is lower than an acceptance threshold (n_{lim}), selected as the closest integer to $0.6N_p$. This means that the iteration process is ended when 60% of the detected feature points are associated to the target model. As soon as this condition is satisfied, the pose solution is refined exploiting the optimized EPnP routine (Efficient Gauss Newton Optimization) proposed in [38]. Finally, a check on the algorithm runtime is carried out (**Step 4**). Specifically, if it exceeds a threshold (t_{lim}), typically of the order of a few seconds, the acquisition process is restarted acquiring a new image. This latter functionality is necessary to avoid that the target/chaser pose evolves too quickly during the runtime (which can occur if the relative rotational dynamics is particularly fast). This latter functionality is necessary to avoid that the initial pose solution falls outside the field of convergence of the tracking algorithm, which can occur if the relative rotational dynamics is so fast that the pose evolves too quickly during the runtime. The reasoning for this strategy is that the runtime tends to increase when the relative observation geometry between the target and the camera is not favorable (e.g., the principal direction of the target is not visible in the FOV). Indeed, if the target is observed from relatively large distance (as when the initial acquisition must be carried out), it is highly likely that the relative observation geometry will rapidly evolve toward a more favorable condition (e.g., the target *principal* direction is fully visible in the FOV) for which the runtime check (Step 4) is satisfied. The issues related to the occurrence of not favorable observation geometries can also be counteracted by exploiting the knowledge of the target to design relative trajectories which satisfy both safety and relative navigation requirements (i.e., the target observation geometry is optimized) [41]. Indeed, if the run-time threshold is exceeded, the initial pose solution could fall outside the field of convergence of the subsequent tracking algorithm.

IV. Navigation Filter

In this section the navigation filter used to process the measurements available from the pose determination block is described. The output of the tracking algorithm is the relative position vector between the target and chaser center of mass and the relative rotation matrix between the target and the chaser spacecraft. These quantities are fed to the navigation filter. As already mentioned, the translational and rotational dynamics are treated separately. Hence, the adopted filtering strategy exploits two separated filters, one for the relative translation and another for the relative rotation.

A. Translation Filter

In this subsection, the H-∞ Filter adopted for the translational motion estimation is presented. For linear systems with process and measurement noise represented by a zero mean Gaussian distribution, the Kalman filter [42] represents the optimal estimator. In fact, if the standard deviations of the process and measurement noise are known, the Kalman filter is the minimum variance estimator. However, if these assumptions are not satisfied or if the tuning is off-nominal, a

filter that minimizes the worst-case estimation error rather than the variance of the estimation error could outperform the Kalman filter. This kind of filter is called H-∞ filter or also minimax filter. It minimizes the ∞-norm of the estimation error and it does not make any assumptions about the statistics of the process and measurement noise [28]. The choice of such filter is driven by the fact that, dealing with optical measurement, the assumptions of the Kalman Filter are usually not satisfied. Moreover, both process and measurement noise are usually influenced by the position of the spacecraft and by the illumination conditions respectively. This may drive, in some operative phases, to noise levels different from the ones adopted during the tuning process. All these considerations lead to a robust approach rather than an optimal one. Let's consider a linear time-invariant system:

$$\mathbf{x_{k+1}} = \mathbf{F_k}\,\mathbf{x_k} + \mathbf{w_k}$$
$$\mathbf{y_k} = \mathbf{H_k}\,\mathbf{x_k} + \mathbf{v_k}$$

(5)

with $\mathbf{x_k}$ the state vector, $\mathbf{w_k}$ and the $\mathbf{v_k}$ the process and measurement noise respectively with associated covariance matrix $\mathbf{Q_k}$ and $\mathbf{R_k}$, and $\mathbf{y_k}$ the measurement output. $\mathbf{F_k}$ and $\mathbf{H_k}$ are the state-transition and observation model matrices respectively. Similarly to the Kalman Filter, the correction equation can be defined as:

$$\mathbf{\hat{x}_{k+1}} = \mathbf{F_k}\,\mathbf{\hat{x}_k} + \mathbf{F_k}\,\mathbf{K_k}\,(\mathbf{y_k} - \mathbf{H_k}\,\mathbf{\hat{x}_k})$$

(6)

However, the filter gain $\mathbf{K_k}$ has not the same expression of the Kalman Filter. In particular $\mathbf{K_k}$ has to be chosen such that $\|T_{ew}\|_\infty < \dfrac{1}{\theta}$, where T_{ew} represents the difference between the predicted and real state and θ is a tuning parameter. To find the value of $\mathbf{K_k}$ it is possible to explicit a Riccati-like equation:

$$\mathbf{P_{K+1}} = \mathbf{F_k}\mathbf{P_k}[\mathbf{I} - \theta\mathbf{P_k} + \mathbf{H_k}^\mathbf{T}\mathbf{R_k}^{-1}\mathbf{H_k}\mathbf{P_k}]^{-1}\mathbf{F^T} + \mathbf{Q_k}$$

(7)

where $\mathbf{P_k}$ is the state covariance matrix.

The expression of the gain is:

$$\mathbf{K_k} = \mathbf{P_k}[\mathbf{I} - \theta\mathbf{P_k} + \mathbf{H_k}^\mathbf{T}\mathbf{R_k}^{-1}\mathbf{H_k}\mathbf{P_k}]^{-1}\mathbf{H_k}^\mathbf{T}\mathbf{R_k}^{-1}$$

(8)

The complete *H-∞* formulation is reported in Table 1.

The possibility to use a linear filtering is justified by the fact that the relative translational dynamics can be described by a set of linear equations. The relative translational motion between two objects in space can be described in different ways. The most general approach, exploiting nonlinear differential equations implies the use of nonlinear filtering techniques, such as the Extended Kalman Filter, that do not guarantee the optimality of the solution. Linearized formulations for relative dynamics exist but they are valid under some restrictive assumptions. In particular, in 1960, Clohessy and Wiltshire [43] derived the equations of motion for one satellite relative to another. This relative motion

11

Table 1 The H-∞ Filter

Initialization:
$$\hat{\mathbf{x}}(\mathbf{t_0}) = \hat{\mathbf{x}}_0, \quad \mathbf{P}(\mathbf{t_0}) = \mathbf{P_0},$$

Filtering:
$$\mathbf{K_k} = \mathbf{P_k}[\mathbf{I} - \theta\mathbf{P_k} + \mathbf{H_k^T R_k^{-1} H_k P_k}]^{-1}\mathbf{H_k^T R_k^{-1}}$$ $$\hat{\mathbf{x}}_{\mathbf{k+1}} = \mathbf{F_k}\hat{\mathbf{x}}_\mathbf{k} + \mathbf{F_k K_k}(\mathbf{y_k} - \mathbf{H_k}\hat{\mathbf{x}}_\mathbf{k})$$ $$\mathbf{P_{k+1}} = \mathbf{F_k P_k}[\mathbf{I} - \theta\mathbf{P_k} + \mathbf{H_k^T R_k^{-1} H_k P_k}]^{-1}\mathbf{F_k^T} + \mathbf{Q_k}$$

model is based on the assumptions of nearly circular orbits, small target-chaser distance compared to the orbit radius, and spherical Earth. In the following, the Yamanaka and Ankersen model [44] is considered. Such a linearized formulation accounts for arbitrary elliptical orbits and leads to a state transition matrix that is advantageous when implementing filtering techniques. In fact, an expression for the $\mathbf{F_k}$ matrix of Equation 5 is directly derived. The details of the implementation can be found in [44].

B. Rotation Filter

For the rotation part, a second-order minimum energy filter on the Lie group is implemented. Minimum energy filtering was introduced by Mortensen [45], and has been specialized to attitude estimation on the Special Orthogonal Group SO(3) by Zamani et al. [46]. In [47], the authors showed that such minimum energy filter on SO(3), thanks to the Lie group structure of SO(3), outperforms the industry standard Multiplicative Extended Kalman Filter for absolute attitude estimation. A further development of the presented minimum energy filter on $SO(3)$ was introduced by Saccon *et al* [48]. They present an explicit formula for a second-order optimal nonlinear filter. With a proper choice of an affine connection, they showed that the filter has the familiar form of a gradient estimator along with a perturbed Riccati-type matrix differential equation which describes the evolution of the filter gain. A slightly different formulation of the second-order minimum energy filter is hereby proposed, as in [32]. First, the filter in [48] is adapted to relative state estimation. Despite the kinematic equations do not change in a relative framework, the dynamics must include both the target and leader contributions. In our formulation the derivative of the angular acceleration is forced to zero to avoid the explicit dependence on the inertia matrix of the target spacecraft. This is somehow similar to what it is usually done for α-β-γ filters [28]. Despite the possibility of achieving slightly worse performance that the one provided by the filter scheme in [48], the proposed formulation is more promising due to its capability to deal also with unknown or partially unknown objects. In particular, the rotational dynamics exploited by the filter is:

$$\dot{\boldsymbol{R}}_{\mathcal{TC}} = \boldsymbol{R}_{\mathcal{TC}}\left(\omega(t)\right)_\times, \tag{9}$$

$$\dot{\omega} = \boldsymbol{B}\boldsymbol{\delta}. \tag{10}$$

with $(\boldsymbol{R}_{\mathcal{T}C}, \boldsymbol{\omega}) \in SO(3) \times \mathbb{R}^3$, $\boldsymbol{\omega}$ representing the relative angular velocity and $\boldsymbol{B} \in \mathbb{R}^{3x3}$ indicating a coefficient matrix that allows for different weightings of the components of the unknown process noise $\boldsymbol{\delta}$.

Equations 9 and 10 are directly expressed in relative terms and in the chaser frame C, without an explicit dependence on the inertia matrix of the target spacecraft. The formulation of the proposed relative filter is summarized in Table 2.

V. Simulation Environment and Results

In this section, the numerical simulation environment to validate the proposed relative navigation algorithm is described. The measurement generation process is illustrated and the simulation scenarios are presented. Finally, an overview of the results for different orbital conditions, target spacecraft and noise level is provided.

A. Numerical Simulation Environment Description

The proposed relative navigation architecture is preliminarily validated through numerical simulations, carried out in MATLAB environment. In particular, the relative translational and rotational orbital dynamics between two spacecraft are reproduced by using non-linear dynamical models that are detailed in the next paragraphs III..1, V.A.1. An unperturbed, unforced motion is considered mainly because of the short duration of vision-based relative proximity operations and of the limited chaser-target separation. This is a common assumption for the validation of similar vision-based relative navigation algorithms [18, 27, 49]. The generated relative trajectory is used as a reference to evaluate the relative navigation algorithm performance, but also to provide input measurements to the pose determination block (as detailed in paragraph V.A.2).

1. Relative Dynamics

The relative translational orbital dynamics is described according to the model presented in [50]. Hence, the free-evolving eccentric trajectory of the target in the LVLH reference frame of the chaser is simulated by integrating the non-linear differential equation in Equation 11.

$$
\begin{aligned}
\ddot{x} - 2\dot{\theta}_C \dot{y} - \ddot{\theta}_C y - \dot{\theta}_C^2 x &= -\frac{\mu(r_C + x)}{[(r_C + x)^2 + y^2 + z^2]^{\frac{3}{2}}} + \frac{\mu}{r_C^2} \\
\ddot{y} + 2\dot{\theta}_C \dot{x} + \ddot{\theta}_C x - \dot{\theta}_C^2 y &= -\frac{\mu y}{[(r_C + x)^2 + y^2 + z^2]^{\frac{3}{2}}} \\
\ddot{z} &= -\frac{\mu z}{[(r_C + x)^2 + y^2 + z^2]^{\frac{3}{2}}}
\end{aligned}
\tag{11}
$$

where $\rho = [x, y, z]$ is the relative position vector, θ_C is the chaser true anomaly, r_C is the magnitude of the chaser spacecraft position vector in the inertial frame $\mathcal{I}$ and μ is the Earth's gravitational constant.

On the other hand, the relative rotational dynamics is propagated using a combination of the Euler equation for the

Table 2 The 2nd order minimum energy filter on $SO(3)$

Initialization:	

$$\hat{R}_{\mathcal{T}C}(t_0) = \hat{R}_{\mathcal{T}C0}, \; \hat{\omega}(\mathbf{t_0}) = \hat{\omega}_0, \; K(t_0) = K_0,$$

Filtering:

$$\dot{\hat{R}}_{\mathcal{T}C} = \hat{R}_{\mathcal{T}C} \left(\hat{\omega}(t) + K_{11}r^R + K_{12}r^\omega \right)_\times,$$

$$\dot{\hat{\omega}} = K_{21}r^R + K_{22}r^\omega,$$

$$\dot{K}(t) = -\alpha K + AK + KA^T - KEK + BR^{-1}B^T - WK - KW^T,$$

where

$$r_t = \begin{bmatrix} r^R \\ r^\omega \end{bmatrix} = \begin{bmatrix} -u_1(\hat{r}_1 \times r_1) - u_2(\hat{r}_2 \times r_2) \\ \mathbf{0} \end{bmatrix},$$

$$u_i = \frac{b^2}{d_i^2},$$

$$\hat{r}_i = \hat{R}_{\mathcal{T}C}^T \bar{r}_i, \quad r_i = \hat{R}_{\mathcal{T}C}^T \bar{r}_i + d_i \epsilon,$$

$$A = \begin{bmatrix} -\hat{\omega}_\times & \mathbf{I} \\ 0 & 0 \end{bmatrix},$$

$$E = \begin{bmatrix} \sum_{i=1}^{2} u_i \left((\hat{r}_i)_\times (r_i)_\times + (r_i)_\times (\hat{r}_i)_\times \right)/2 & 0 \\ 0 & 0 \end{bmatrix},$$

$$BR^{-1}B^T = \begin{bmatrix} 0_{3\times3} & 0 \\ 0 & BR^{-1}B^T \end{bmatrix}$$

$$W = \begin{bmatrix} \frac{1}{2} \left(K_{11}r^R + +K_{12}r^\omega \right)_\times & 0 \\ 0 & 0 \end{bmatrix}.$$

chaser and the target [18, 51] as in Equation 12.

$$I_c \dot{\omega} = I_c R_{\mathcal{TC}} I_t^{-1} \left[-\omega_t|_{\mathcal{T}} \times I_t \omega_t|_{\mathcal{T}} - I_c \omega_c \times \omega_c - [-\omega_c \times I_c \omega_c] \right], \tag{12}$$

where ω and ω_c are, respectively, the relative and chaser angular velocity expressed in the chaser frame C and $\omega_t|_{\mathcal{T}}$ is the target angular velocity expressed in the target frame $\mathcal{T}$. $\mathbf{I_c}$ and $\mathbf{I_t}$ are the inertia matrices of chaser and target spacecraft respectively.

2. Measurement Generation

In order to preliminary assess the performance of the proposed approaches for monocular pose determination, simulated sets of 2D point features (e.g., corners) are generated. Specifically, 3D landmarks composing the target model are projected on the image according to the true relative attitude and position parameters. Hence, the position of each detected feature point on the image plane is modified to account for potential errors introduced by the image processing. Specifically, a Gaussian white noise is added to the pixel coordinates of each point feature (whose standard deviation (σ_{pix}) is expressed in terms of a certain number of pixel). Moreover, a number of outliers (n_{out}) is randomly located in the region of the image plane occupied by the target (according to a uniform distribution) to take into account either false alarms from the image processing algorithms or potential deviation of the simplified model from the actual target geometry.

B. Simulation Scenario

In this section, the scenarios considered to validate the proposed algorithm, are presented, both in terms of target and orbit characteristics.

1. Target

XMM-Newton   The first selected target is the ESA X-ray space observatory, XMM-Newton. This is an example of very large spacecraft for which servicing or removal operations may be necessary. The inertia matrix of the object has been computed considering available information about its size, shape and weight [52, 53], and assuming a simplified geometry (i.e., the real geometry of the target is modeled as a combination of elementary volume elements, such as cuboids, cones, cylinders etc.), and uniform mass distribution. Clearly, the estimated values do not represent with high accuracy the real inertia of the considered target. This is due to the lack of information about spacecraft components
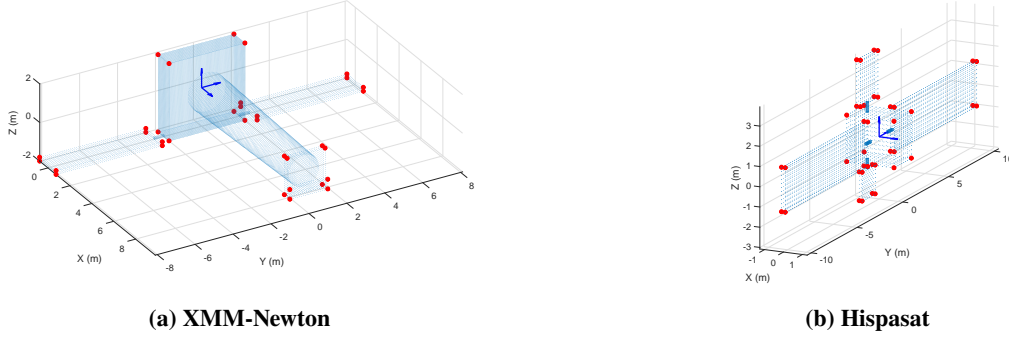
(a) XMM-Newton　　　　　　　　　　　　　　(b) Hispasat

Fig. 3　Targets Point Cloud and Reference Frame

available in the open literature. The resulting inertia matrix:

$$\mathbf{I_{XMM}} = \begin{bmatrix} 7052 & -820 & -3991 \\ -820 & 129129 & 0 \\ -3991 & 0 & 131196 \end{bmatrix} kgm^2 \tag{13}$$

is computed considering the body fixed reference frame as in Figure 3a.

Hispasat　　The second target is one of the satellites of the Hispasat family. It represents the next generation of SmallGEO platforms for telecommunication. This lower mass class of telecom satellites is the ideal target for future on-orbit servicing missions. Similarly to XMM, the inertia matrix is computed by exploiting the information in [54], and assuming the reference frame as in Figure 3b:

$$\mathbf{I_{hispa}} = \begin{bmatrix} 6396 & 0 & -46 \\ 0 & 3703 & 0 \\ -46 & 0 & 5918 \end{bmatrix} kgm^2 \tag{14}$$

2. Orbital Scenario

Absolute Orbital Parameters　　Testing the filter and its dynamics over different possible absolute orbits is important to verify its robustness. For this reason, three different orbital regimes are considered: Low-Earth-Orbit (LEO), Highly-Elliptical-Orbit (HEO) and Geostationary-Earth-Orbit (GEO). These alternatives represent the most important classes of Earth orbits with a variety of altitudes and eccentricities. Three real satellites are considered, Envisat (LEO), XMM (HEO), Hispasat (GEO). The associated classic parameters are reportes in Table 3.

Table 3　Scenarios Orbital Parameters

	LEO	HEO	GEO
a [km]	7143.1	66931.6	42164.1
e [-]	$1.4*10^{-4}$	0.808	$1.1*10^{-4}$
i [deg]	98.2	69.9	0.005
ω [deg]	85.9	96	41.6
Ω [deg]	79.2	352.5	274.9

Relative Motion   Relative translational trajectories and rotational motion are defined depending on the absolute reference orbit of the chaser spacecraft and on the relative angular velocity. Five different scenarios are considered, as summarized in Table 4.

Table 4   Cases Definition

	Case 1 - LEO Low	Case 2 - LEO High	Case 3 - LEO Mix	Case 4 - HEO	Case 5 - GEO
Chaser Orbit	LEO	LEO	LEO	HEO	GEO
Target	XMM	XMM	XMM	XMM	Hispasat
ρ_0 [m]	[0 30 0]	[0 30 0]	[0 30 0]	[0 30 0]	[0 40 0]
$\dot{\rho}_0$ [m/s]	$[0\ 1\ 0]*10^{-4}$	$[0\ 1\ 0]*10^{-4}$	$[0\ 1\ 0]*10^{-4}$	$[3\ -1\ 0]*10^{-3}$	$[0\ -1\ 0]*10^{-4}$
$\omega_{t0}\vert_{\mathcal{T}}$ [deg/s]	[0.1 0 0]	[1 0 0]	[0.5 -0.3 0.2]	[0.1 0 0]	[0.1 0 0]

where ρ_0 and $\dot{\rho}_0$ are the initial relative position and translational velocity expressed in the chaser frame $\mathcal{C}$ and $\omega_{t0}\vert_{\mathcal{T}}$ is the initial target angular velocity expressed in the target reference frame $\mathcal{T}$. These quantities are the initial conditions for Equations 11, 12. For sake of completeness, it is worth nothing that no control is assumed on the chaser spacecraft orientation and that it is always aligned with its LVLH reference frame. This entails that the relative translational velocity has to be kept very small in order to avoid a rapid drift of the target spacecraft. For this reason, the initial conditions for all the different cases are not representative of any particular operative scenario but they are selected to guarantee a limited evolution of the relative position, in a chaser-target configuration, for the complete duration of the simulation. Different cases of relative angular velocity are considered to test its effect on performance of the proposed architecture.

3. Noise Level

As previously explained in paragraph V.A.2, the noise is added while generating the simulated images according to a Gaussian distribution with standard deviation equal to σ_{pix}. An additional source of error is represented by the presence

of n_{out} outliers. Another relevant feature concerns the precise determination of the chase true anomaly, that is used to propagate the relative translational dynamics equations (see 11). In real applications, this value is computed by means of the spacecraft position and velocity coming from the absolute navigation filter. In literature, similar works [18, 27, 49] neglect the possible noise associated to this quantity. However, a noisy fluctuation of the chaser true anomaly can strongly affect the filter robustness and stability. For this reason, the true anomaly of the chaser spacecraft is corrupted by noise in this work. In particular, the associated noise level is described by a Gaussian distribution with standard deviation respectively $\sigma_{pos} = 10^{-2}km$ and $\sigma_{vel} = 10^{-4}km/s$.

C. Results

In this section, the results of the algorithm validation are presented. As in Figure 1, the pose estimator is divided in two blocks: *Acquisition* and *Tracking*. First, an analysis of the *Acquisition* process is carried out. In particular, the robustness to different relative angular velocities is tested. This analysis is critical since the speed of the relative rotational motion strongly affects the performance during *Acquisition*. With this study, statistical quantities of the acquired pose and acquisition time are obtained (i.e. mean and standard deviation of the distribution). These statistical parameters are then used to generate random initial conditions to test the overall architecture proposed in this manuscript (including the pose tracking algorithm and the filtering techniques) over a relatively long time interval. Before presenting the results, the definitions of the estimation errors, used for performance assessment, are introduced.

The relative position error is defined as:

$$e_\rho = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \tag{15}$$

where $\hat{x}, \hat{y}, \hat{z}$ are the position components estimates. Similarly, the relative attitude error is computed as in [55]:

$$e_R = acos\left(1 - \frac{tr(\mathbf{I} - \boldsymbol{R}_{\mathcal{T}Ci}{}^T \hat{\boldsymbol{R}}_{\mathcal{T}Ci})}{2}\right) \tag{16}$$

with $\hat{\boldsymbol{R}}_{\mathcal{T}C}$ being the estimated rotation matrix.

Acquisition Results The content of this section is focused on the performance achievable by the proposed acquisition algorithm.A deep analysis of the *Acquisition* process is presented in this paragraph. A statistical analysis of the results obtained over 100 runs is performed considering three scenarios: Case 1 - LEO Low, Case 2 - LEO High, Case 3 - LEO Mix (see V.B.2). For each of the analyzed cases, the noise level associated to the result of the image processing step is simulated as a Gaussian white noise on the location of the detected point features with standard deviation $\sigma_{pix} = 1$ as in [22]. As for the number of outliers, it is fixed to $n_{out} = 1$. The algorithm is run until the initial acquisition of the pose parameters is declared. Its performance is evaluated in terms of pose estimation error processing time. In

this regard, it is worth outlining that the value of the elapsed time is not intended as an absolute measurement of the algorithm computational efficiency (since the simulations are run in MATLAB environment). However, the results can help to assess the effect on the processing burden of the increase in the target/chaser relative rotational dynamics. Figures 4a, 4b, 4c show the position, rotation error and runtime at the end of the *Acquisition*. Figure 4a and 4b show that a difference in the initial relative angular velocity does not drastically affect the error in the acquired pose. In fact, despite the error of Case 3 is often higher than in the other two cases, the order of magnitude is comparable. However, a significant difference is present in Figure 4c. In fact, as expected, a higher relative angular velocity implies a difficulty in the acquisition convergence and, therefore, a higher time to complete the acquisition. This is certainly an aspect to consider while approaching fast-spinning objects.
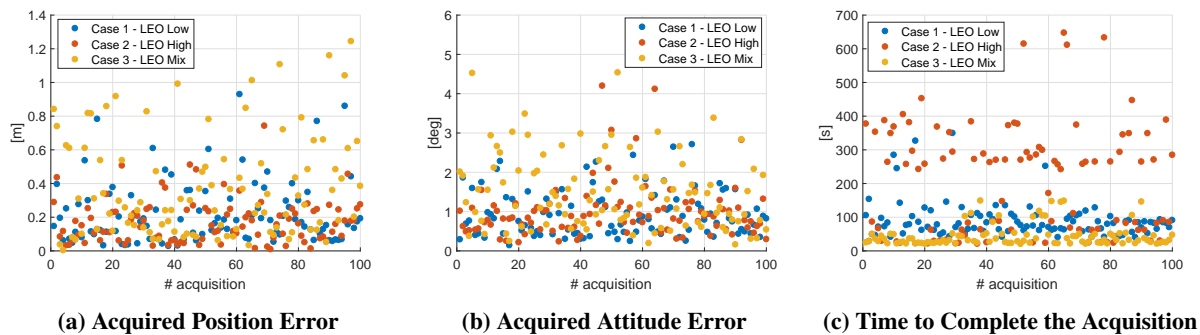

| (a) Acquired Position Error | (b) Acquired Attitude Error | (c) Time to Complete the Acquisition |

Fig. 4 Acquisition Results

For a more quantitative description of the results, the main statistics of the obtained distributions are reported in Table 5.

Table 5   Acquisition Statistical Results

	Case 1 - LEO Low	**Case 2 - LEO High**	**Case 3 - LEO Mix**
e_ρ - Mean [m]	0.2196	0.1747	0.3767
e_ρ - STD [m]	0.1879	0.1314	0.3003
e_R - Mean [deg]	0.9502	1.007	1.474
e_R - STD [deg]	0.5939	0.692	0.9339
Time - Mean [s]	91.48	206.6	44.97
Time - STD [s]	54.02	160.7	32.91

It is worth underlying that similar results are obtained also considering the Hispasat target geometry.

Tracking and Filter Results   The statistics in Table 5, obtained with the *Acquisition* analysis, are used to generate the random input for the *Tracking* and subsequently the filtering step. Please note that for Case 4 - HEO and Case 5 - GEO
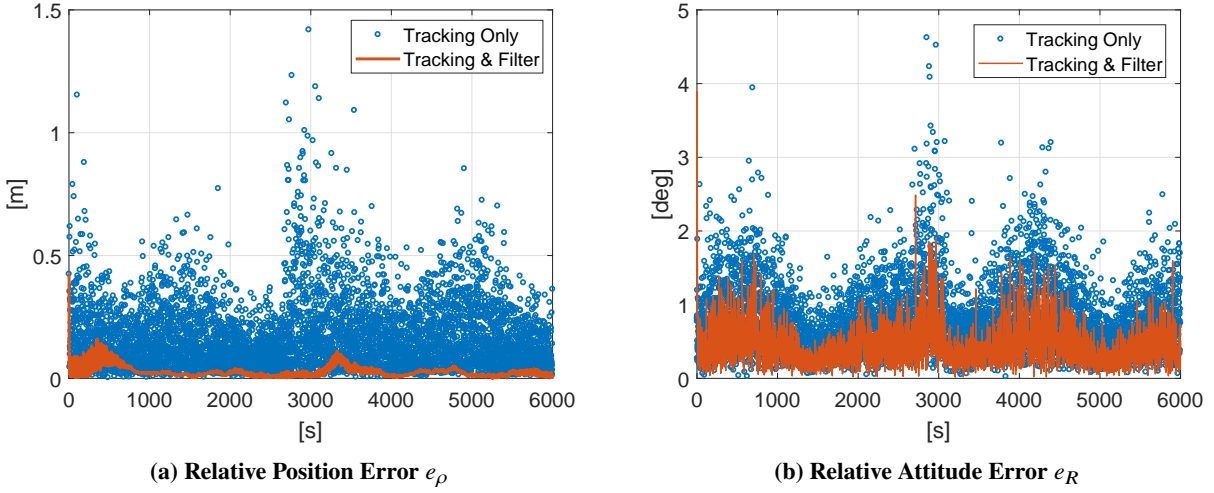
(a) Relative Position Error e_ρ (b) Relative Attitude Error e_R

Fig. 5 Estimation Errors with and without Filter

the statistics of the Case 1 - LEO are used, having the three cases the same initial relative angular velocity.

The first presented analysis wants to show the beneficial effect of having a filter downstream of the vision-based pose determination algorithm. In Figure 5 the relative position and attitude estimation errors of the *Tracking* process only are shown, together with the results from the approach combining filter and monocular navigation. The presented simulation is run on the Case 1 - LEO Low scenario with a noise of $\sigma_{pix} = 2$ and $n_{out} = 1$. In this case, a 1Hz frame rate is considered.

The filter benefits are especially evident for the position error. In fact, the estimation error with the filter in the loop is reduced by more than one order of magnitude in Figure 5a. In Figure 5b, the improvement is still evident but less remarkable. Indeed, the overall estimation error is reduced by roughly the 20%. In Figures 5a and 5b a fluctuation in the estimation error is observed. This effect can be due to the relative rotational dynamics that causes a change in the observation conditions of the target. Periodically, the camera observes the target from an unfavorable point of view; this affects the accuracy of the pose determination and, consequently, of the navigation filter estimates. The robustness of the proposed approach is further proved by means of a statistical analysis. A set of 50 runs, considering the first 30 minutes, are performed for all the five different test scenarios. It is worth underlying that, for each simulation, the initial conditions of the tracking are randomly selected from the statistics derived from the analysis on the *Acquisition* step. Moreover, the noise associated to the point features extracted from the simulated images is randomly picked from a zero-mean Gaussian distribution with standard deviation of $\sigma_{pix} = 2$. A number of outliers of $n_{out} = 1$ is used. Figure 6 shows mean relative position and attitude errors, computed according to Equations15,16 and averaged for each time step over the 50 runs for the three different LEO cases.

As expected, a higher relative angular velocity implies a higher error and a slower convergence, for both the LEO Mix and LEO High cases, visible in Figure 6b. This is justified by the fact that the *Tracking* process is more problematic

(a) Average Relative Position Error e_ρ

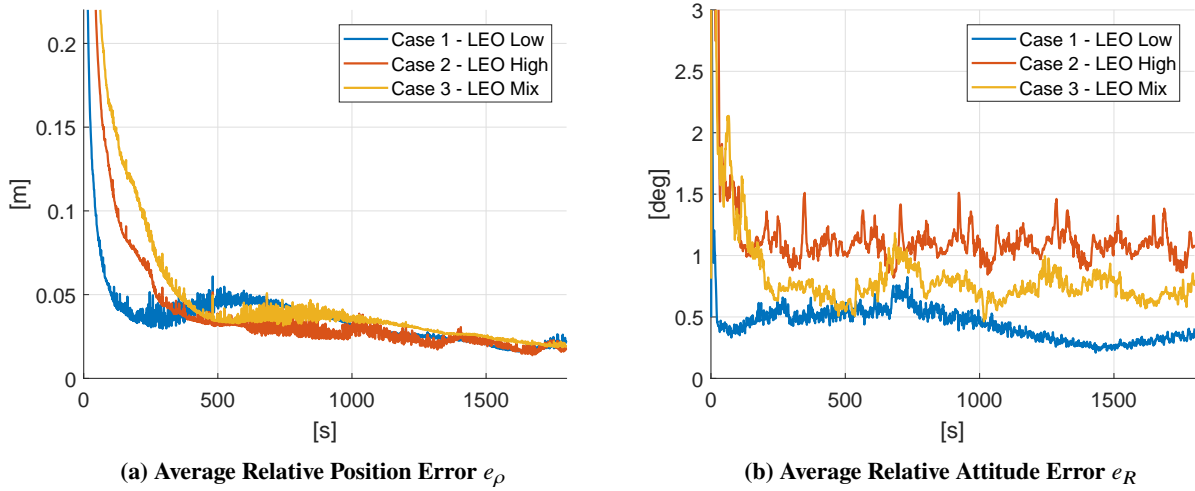(b) Average Relative Attitude Error e_R

Fig. 6   Average Estimation Errors - 50 runs

while experiencing higher angular velocity. Moreover, recalling Equation 10, the rotational filter is built assuming the relative angular acceleration equal zero and considering only white noise components. This particular formulation strongly affects the attitude estimation in the case of higher relative angular velocities. In these cases, this problem could be solved by exploiting alternative techniques, e.g. multiple model filters, to the detriment of a higher computational burden. Concerning the relative position error, the first three LEO cases converge to almost the same value with a slightly different convergence time. This is in line with the expected results being the translational motion equal in all the three cases. The different convergence rate is due to the coupling between the rotation matrix and position vector while solving the SoftPOSIT optimization. For a more quantitative analysis, in Table 6 the steady state Root Mean Square Error (RMSE) for all the cases, considering the last 1000 seconds, are reported.

Table 6   Tracking & Filter RMSE Results

	RMSE - Position [m]	**RMSE** - Attitude [deg]
Case 1 - LEO Low	0.0109	0.3259
Case 2 - LEO High	0.0225	1.0834
Case 3 - LEO Mix	0.0269	0.7205
Case 4 - HEO	0.0673	0.6711
Case 5 - GEO	0.0264	0.4183

The Case 4 - HEO is substantially different from the previous ones. In fact, the high value of the chaser orbit eccentricity implies a higher error between the linearized relative dynamics equations, used in the filter, and the reality. This is evidently remarkable from Table 6 where the error of the HEO case is much higher with respect to all the others.

21

However, this does not have any strong influence in the relative attitude estimate error that is comparable to the other test cases. Finally, despite the different target, the GEO case RMSEs are very similar to LEO cases.

Noise Sensitivity Analysis   In order to evaluate the robustness of the proposed method, a sensitivity analysis over σ_{pix} is performed. The orbital scenario of Case 1 is propagated for 1800 seconds with a noise standard deviation ranging from 2 to 6 pixels. The tuning of the filter is kept equal for all the different test cases. Given the estimation error in time for the different simulations, the Root Mean Square Error (RMSE) from 50 seconds to the end is computed and the change in the steady state error is thus evaluated. Table 7 reports the RMSE for different noise levels, for both relative position and attitude estimation errors.

Table 7   RMSE for Different Noise Levels

$\sigma_{\mathbf{pix}}$	**RMSE** - Position [m]	**RMSE** - Attitude [deg]
2	0.0312	0.4221
3	0.0599	0.6671
4	0.0735	0.9430
5	0.11	1.1323
6	0.1455	1.4902

As expected, the errors increase as long as the noise level increases. However, both the positional and angular accuracy are acceptable even considering very high levels in the measurement noise (in reference [22] the maximum value considered for σ_{pix} for numerical simulation is equal to 3 pixels).

Outliers Sensitivity Analysis   A similar sensitivity analysis is also carried out to evaluate robustness against increased number of outliers. Also in this case the simulations are run on the Case 1 scenario with fixed initial conditions and variable number of outliers n_{out}. The noise level is kept at σ_{pix} = 2. Analogously, the Root Mean Square Error (RMSE) is computed from 50 seconds to the end of the simulation. Table 7 reports the RMSE for different numbers of outliers, for both relative position and attitude estimation errors.

Table 8   RMSE for Different Number of Outliers

n_{out}	RMSE - Position [m]	RMSE - Attitude [deg]
1	0.0312	0.4221
2	0.0341	0.4314
3	0.0363	0.4371
4	0.0444	0.4479
5	0.0479	0.4648

Table 8 shows that despite a very slow increase in the estimation errors, the proposed algorithm is able to cope also with a relatively large number of outliers (it works also if the outliers are the 10% of the extracted features).

VI. Conclusion

An original approach for relative navigation between two uncooperative but known spacecraft has been presented. The proposed algorithm relies on a loosely coupled approach is adopted, involving a vision-based pose determination technique and a navigation filter. This paper addresses the problem of both pose acquisition and tracking. The acquisition phase is managed by an innovative methodology conceived to compute the target initial state when no a-priori information is available about this satellite, except that concerning its geometry. A RANSAC & PCA based procedure is proposed to solve the 2D-3D feature matching problem. Hence, the spacecraft initial pose is derived through the EPnP solver. Once the target state is acquired, it is tracked over time by filtering the monocular pose determined through SoftPOSIT. The navigation filter is conceived to have a decoupled structure. A robust H-∞ filtering strategy is exploited for the translational motion. On the other hand, an original formulation on Lie groups is developed for relative attitude estimation. Preliminary numerical validation and performance assessment are carried considering different simulation scenarios. Realistic target/chaser relative dynamics, target geometries and vision-based measurement are reproduced. Simulation results show the capability of the proposed pose acquisition algorithm to provide an accurate initialization for the tracking step (position and attitude errors lower than 0.5m and 2.5° in most of the cases). The sensitivity analysis on the acquisition step highlights a dependence between relative angular velocity and acquisition time. Specifically, higher relative angular velocities imply longer time to initialize the pose parameters. Furthermore, the pose tracking functionality and the navigation filter are validated through statistical simulations, considering different orbital scenarios. Satisfactory results are obtained for all the presented cases for both position and attitude estimation. In fact, steady state relative position and attitude RMSE are lower than 3cm (except in the HEO case) and 1° respectively. Finally, sensitivity analyses are performed to demonstrate the algorithm robustness against measurement noise and error sources. The next step of our research activity will be aimed at experimentally evaluating performance of the proposed relative navigation approach within a dedicated laboratory setup. In this respect, one important aspect to analyze is the effect on

pose determination accuracy of the level of detail of the off-line model with respect to the actual geometry of the target.

References

[1] Ambrose, R., Nesnas, I., Chandler, F., Allen, B., Fong, T., Matthies, L., and Mueller, R., "NASA technology roadmaps: TA 4: Robotics and autonomous systems," Tech. rep., Technical report, NASA, 2015.

[2] Starek, J. A., Açıkmeşe, B., Nesnas, I. A., and Pavone, M., "Spacecraft autonomy challenges for next-generation space missions," *Advances in Control System Technology for Aerospace Applications*, Springer, 2016, pp. 1–48.

[3] Di Mauro, G., Lawn, M., and Bevilacqua, R., "Survey on Guidance Navigation and Control Requirements for Spacecraft Formation-Flying Missions," *Journal of Guidance, Control, and Dynamics*, 2017, pp. 1–22.

[4] Wang, X., Qin, W., Bai, Y., and Cui, N., "A novel decentralized relative navigation algorithm for spacecraft formation flying," *Aerospace Science and Technology*, Vol. 48, 2016, pp. 28–36.

[5] Alepuz, J. P., Emami, M. R., and Pomares, J., "Direct image-based visual servoing of free-floating space manipulators," *Aerospace Science and Technology*, Vol. 55, 2016, pp. 1–9.

[6] Pinard, D., Reynaud, S., Delpy, P., and Strandmoe, S. E., "Accurate and autonomous navigation for the ATV," *Aerospace Science and Technology*, Vol. 11, No. 6, 2007, pp. 490–498.

[7] Zhang, L., Zhang, S., Yang, H., Cai, H., and Qian, S., "Relative attitude and position estimation for a tumbling spacecraft," *Aerospace Science and Technology*, Vol. 42, 2015, pp. 97–105.

[8] Felicetti, L., and Emami, M. R., "Image-based attitude maneuvers for space debris tracking," *Aerospace Science and Technology*, Vol. 76, 2018, pp. 58–71.

[9] Opromolla, R., Fasano, G., Rufino, G., and Grassi, M., "A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations," *Progress in Aerospace Sciences*, Vol. 93, 2017, pp. 53–72.

[10] Clerc, X., and Retat, I., "Astrium vision on space debris removal," *Proceeding of the 63rd International Astronautical Congress (IAC 2012), Napoli, Italy*, Vol. 15, 2012.

[11] Volpe, R., Palmerini, G. B., and Circi, C., "Preliminary analysis of visual navigation performance in close formation flying," *Aerospace Conference, 2017 IEEE*, IEEE, 2017, pp. 1–12.

[12] Shi, J., et al., "Good features to track," *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, IEEE, 1994, pp. 593–600.

[13] Canny, J., "A computational approach to edge detection," *Readings in Computer Vision*, Elsevier, 1987, pp. 184–203.

[14] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, Vol. 60, No. 2, 2004, pp. 91–110.

[15] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: An efficient alternative to SIFT or SURF," *Computer Vision (ICCV), 2011 IEEE international conference on*, IEEE, 2011, pp. 2564–2571.

[16] Augenstein, S., and Rock, S. M., "Improved frame-to-frame pose tracking during vision-only SLAM/SFM with a tumbling target," *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 3131–3138.

[17] Volpe, R., Sabatini, M., and Palmerini, G. B., "Pose and Shape Reconstruction of a Noncooperative Spacecraft Using Camera and Range Measurements," *International Journal of Aerospace Engineering*, Vol. 2017, 2017.

[18] Pesce, V., Lavagna, M., and Bevilacqua, R., "Stereovision-based pose and inertia estimation of unknown and uncooperative space objects," *Advances in Space Research*, Vol. 59, No. 1, 2017, pp. 236–251.

[19] Yuan, J., Hou, X., Sun, C., and Cheng, Y., "Fault-tolerant pose and inertial parameters estimation of an uncooperative spacecraft based on dual vector quaternions," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2018, p. 0954410017751766.

[20] Sharma, S., et al., "Comparative assessment of techniques for initial pose estimation using monocular vision," *Acta Astronautica*, Vol. 123, 2016, pp. 435–445.

[21] D'Amico, S., Benn, M., and Jørgensen, J. L., "Pose estimation of an uncooperative spacecraft from actual space imagery," *International Journal of Space Science and Engineering 5*, Vol. 2, No. 2, 2014, pp. 171–189.

[22] Sharma, S., Ventura, J., and D'Amico, S., "Robust Model-Based Monocular Pose Initialization for Noncooperative Spacecraft Rendezvous," *Journal of Spacecraft and Rockets*, 2018, pp. 1–16.

[23] Naasz, B. J., Burns, R. D., Queen, S. Z., Van Eepoel, J., Hannah, J., and Skelton, E., "The HST SM4 Relative Navigation Sensor System: Overview and Preliminary Testing Results from the Flight Robotics Lab," *The Journal of the Astronautical Sciences*, Vol. 57, No. 1-2, 2009, pp. 457–483.

[24] David, P., Dementhon, D., Duraiswami, R., and Samet, H., "SoftPOSIT: Simultaneous pose and correspondence determination," *International Journal of Computer Vision*, Vol. 59, No. 3, 2004, pp. 259–284.

[25] Rhodes, A., Kim, E., Christian, J. A., and Evans, T., "LIDAR-based relative navigation of non-cooperative objects using point Cloud Descriptors," *AIAA/AAS Astrodynamics Specialist Conference*, 2016, p. 5517.

[26] Lichter, M. D., and Dubowsky, S., "State, shape, and parameter estimation of space objects from range images," *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, Vol. 3, IEEE, ????, pp. 2974–2979.

[27] Segal, S., Carmi, A., and Gurfil, P., "Stereovision-based estimation of relative dynamics between noncooperative satellites: Theory and experiments," *IEEE Transactions on Control Systems Technology*, Vol. 22, No. 2, 2014, pp. 568–584.

[28] Simon, D., *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, John Wiley & Sons, 2006.

[29] Berkane, S., and Tayebi, A., "On the design of synergistic potential functions on SO (3)," *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, IEEE, 2015, pp. 270–275.

[30] Choukroun, D., Weiss, H., Bar-Itzhack, I., and Oshman, Y., "Direction cosine matrix estimation from vector observations using a matrix Kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 46, No. 1, 2010.

[31] Lageman, C., Trumpf, J., and Mahony, R., "Gradient-like observers for invariant dynamics on a Lie group," *IEEE Transactions on Automatic Control*, Vol. 55, No. 2, 2010, pp. 367–377.

[32] Pesce, V., Haydar, M. F., Lavagna, M., and Lovera, M., "Comparison of Filtering Techniques For Relative Attitude Estimation of Uncooperative Space Objects," *Aerospace Science and Technology*, 2018.

[33] Doignon, C., "An introduction to model-based pose estimation and 3-D tracking techniques," *Scene Reconstruction Pose Estimation and Tracking*, InTech, 2007.

[34] Dementhon, D. F., and Davis, L. S., "Model-based object pose in 25 lines of code," *International journal of computer vision*, Vol. 15, No. 1, 1995, pp. 123–141.

[35] Gold, S., and Rangarajan, A., "A graduated assignment algorithm for graph matching," *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 18, No. 4, 1996, pp. 377–388.

[36] Fischler, M. A., and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395.

[37] Wold, S., Esbensen, K., and Geladi, P., "Principal component analysis," *Chemometrics and intelligent laboratory systems*, Vol. 2, No. 1-3, 1987, pp. 37–52.

[38] Lepetit, V., Moreno-Noguer, F., and Fua, P., "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, Vol. 81, No. 2, 2009, pp. 155–166.

[39] Heikkila, J., and Silven, O., "A four-step camera calibration procedure with implicit image correction," *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, IEEE, 1997, pp. 1106–1112.

[40] Bouguet, J.-Y., "Camera calibration toolbox for matlab," *http://www.vision.caltech.edu/bouguetj/calib_doc/index. html*, 2004.

[41] Opromolla, R., Fasano, G., Rufino, G., and Grassi, M., "Design of relative trajectories for in orbit proximity operations," *Acta Astronautica*, Vol. 145, 2018, pp. 342–356.

[42] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45.

[43] Clohessy, W., "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, 2012.

[44] Yamanaka, K., and Ankersen, F., "New state transition matrix for relative motion on an arbitrary elliptical orbit," *Journal of guidance, control, and dynamics*, Vol. 25, No. 1, 2002, pp. 60–66.

[45] Mortensen, R., "Maximum-likelihood recursive nonlinear filtering," *Journal of Optimization Theory and Applications*, Vol. 2, No. 6, 1968, pp. 386–394.

[46] Zamani, M., Trumpf, J., and Mahony, R., "Minimum-energy filtering for attitude estimation," *IEEE Transactions on Automatic Control*, Vol. 58, No. 11, 2013, pp. 2917–2921.

[47] Zamani, M., Trumpf, J., and Mahony, R., "Nonlinear attitude filtering: a comparison study," *arXiv preprint arXiv:1502.03990*, 2015.

[48] Saccon, A., Trumpf, J., Mahony, R., and Aguiar, A. P., "Second-order-optimal minimum-energy filters on lie groups," *IEEE Transactions on Automatic Control*, Vol. 61, No. 10, 2016, pp. 2906–2919.

[49] Tweddle, B. E., and Saenz-Otero, A., "Relative computer vision-based navigation for small inspection spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 5, 2014, pp. 969–978.

[50] Gurfil, P., "Relative motion between elliptic orbits: generalized boundedness conditions and optimal formationkeeping," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 761–767.

[51] Segal, S., and Gurfil, P., "Effect of kinematic rotation-translation coupling on relative spacecraft translational dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 1045–1050.

[52] Barré, H., Nye, H., and Janin, G., "An overview of the XMM observatory system." *ESA Bulletin*, Vol. 100, 1999, pp. 15–20.

[53] Van Katwijk, K., Van Der Laan, T., and Stramaccioni, D., "Mechanical and Thermal Design of XMM," *ESA Bulletin*, Vol. 100, 1999, pp. 44–49.

[54] SpaceFlight101, "Hispasat 30W-6," , 2017. URL `https://spaceflight101.com/falcon-9-hispasat-30w-6/hispasat-30w-6/`.

[55] Markley, F. L., and Mortari, D., "Quaternion attitude estimation using vector observations." *Journal of the Astronautical Sciences*, Vol. 48, No. 2, 2000, pp. 359–380.