IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 13, 2019, accepted December 10, 2019, date of publication December 13, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959438

Uncooperative Spacecraft Relative Navigation With LIDAR-Based Unscented Kalman Filter

ROBERTO OPROMOLLA AND ALESSIA NOCERINO
Department of Industrial Engineering, University of Naples "Federico II", 80125 Naples, Italy
Corresponding author: Roberto Opromolla (roberto.opromolla@unina.it)

ABSTRACT Autonomous relative navigation is a critical functionality which needs to be developed to enable safe maneuvers of a servicing spacecraft (chaser) in close-proximity with respect to an uncooperative space target, in the frame of future On-Orbit Servicing or Active Debris Removal missions. Due to the uncooperative nature of the target, in these scenarios, relative navigation is carried out exploiting active or passive Electro-Optical sensors mounted on board the chaser. The focus here is placed on active systems, e.g., LIDARs. In this paper, an original loosely-coupled relative navigation architecture which integrates pose determination algorithms designed to process raw LIDAR data (i.e., 3D point clouds) within a Kalman filtering scheme is presented. Pose determination algorithms play a twofold role being used to initialize the filter state and covariance as well as in the update phase of the Kalman filter. The proposed filtering scheme is an Unscented Kalman Filter designed to use, as measurements for the update phase, relative position, attitude and angular velocity estimates. Performance assessment is carried out within a simulation environment realistically reproducing the operation of a scanning LIDAR and the relative motion between two spacecraft during a target monitoring maneuver. The numerical simulation campaign demonstrates robustness of the proposed approach even when dealing with challenging conditions (e.g., low range measurement accuracy, low update rate and high point-cloud sparseness) determined by the LIDAR noise level and operational parameters.

INDEX TERMS Active debris removal, LIDAR, on-orbit servicing, pose determination, relative navigation, target monitoring, uncooperative spacecraft.

I. INTRODUCTION

The space community has recently paid a growing attention to the sustainability of space activities threatened by the presence of space debris, especially in the most crowded orbital regions, such as Low Earth Orbit and Geostationary Earth Orbit [1], [2]. Consequently, the concept of Active Debris Removal (ADR) [3], has been introduced as one of the solutions (together with passive mitigation measures) to prevent the triggering of the so-called Kessler Syndrome [4]. While ADR aims at actively removing large defunct satellites from their orbits to eliminate the risk of collisions with other objects, thus preserving the future of space activities, the idea of providing services (e.g., refueling, repair, maintenance) to active spacecraft to extend their operative life, which is related to the concept of On-Orbit Servicing (OOS) [5], plays a strategic role for space agencies and industries due to the potentially associated economic benefits [6].

To safely conduct future ADR and OOS missions, the need to enhance the level of autonomy of spacecraft, especially regarding their Guidance, Navigation and Control (GNC) functions, has been widely recognized, as it can be seen, for instance, looking at the NASA Technology Roadmap [7]. Indeed, ADR and OOS require the servicing spacecraft (chaser) to perform high-risk maneuvers (such as rendezvous and docking) in close-proximity with respect to a space target, without being able to rely on commands from ground control stations (due to the associated communications delays). Also, the target is typically uncooperative, i.e., uncontrolled, not able to actively communicate with the chaser, and not equipped with easily recognizable artificial markers, though, in many cases, it is a known object meaning that at least a simplified model of its geometry is available.

This paper focuses on the relative navigation task. Indeed, the development of robust and reliable solutions to estimate the target-chaser relative state (including position, velocity, attitude and angular velocity) in real-time and with high accuracy is not only crucial to meet strict safety criteria [8],

The associate editor coordinating the review of this manuscript and approving it for publication was Ming Xu.

but it can also allow relaxing control requirements (and, consequently, mission costs). Due to the uncooperative nature of the target, relative navigation must be entrusted to active or passive Electro-Optical (EO) sensors [9]. The attention is here focused on the possibility to use an active Light Detection and Ranging system (LIDAR) as main relative navigation sensor. Indeed, despite the challenges related to their higher hardware complexity, and required power and mass allocation, LIDARs can provide direct 3D measurements about the scene in the form of 3D point clouds, unlike passive monocular cameras, at farther operative range than passive stereovision system [10]. Also, they are less sensitive than passive sensors to the high-variability of illumination conditions typical of the space environment [10]. Overall, LIDAR measurements can be processed to extract 6-Degree-of-Freedom (DOF) information about the target-chaser relative translation and rotational dynamics in a wider range of operating conditions, which may be a key factor especially for high-risk ADR missions conceived for the removal of large debris (such as large dead satellites or rocket bodies). However, it is worth mentioning that the relative navigation task in the context of ADR and OOS missions are expected to be entrusted to redundant sensor configurations including both LIDARs and cameras, like in the case of the recent RemoveDebris mission [11] and of the Restore-L mission foreseen for launch in 2023 [12]. In this framework, this paper proposes an innovative relative navigation architecture conceived to estimate with high accuracy the target-chaser relative position, velocity, attitude and angular velocity by processing raw LIDAR measurements. The proposed approach combines and integrates state-of-the-art pose determination algorithms [13] with an original filtering scheme. The performance of the proposed architecture is evaluated within a numerical simulation environment in which the target-chaser relative dynamics and the operation of a scanning LIDAR are realistically reproduced. The goal of the numerical simulation campaign is to deeply characterize the relative state estimation accuracy, as well as its robustness against variability of sensor performance (in terms of noise level, angular resolution and measurement update rate), target absolute dynamics and target observability conditions.

The rest of the paper is organized as follows. In section II, a detailed survey of literature works addressing the relative navigation task using EO sensors is provided, also to highlight the innovative contributions of this work. Section III describes in detail the relative navigation architecture. Section IV presents the simulation environment and scenario including the selected test case, and it collects the results of the numerical simulations campaign. Finally, Section V draws some conclusions and it provides indication about future activities.

II. LITERATURE SURVEY

Most research efforts regarding the challenges of relative navigation for a chaser maneuvering in close-proximity of an uncooperative spacecraft have been dedicated to the pose determination task, which only addresses the estimation of relative position and attitude parameters [9], [14]. Instead, an overview of literature works addressing the full relative state estimation problem is provided in the following. Rather than focusing on the typology of adopted EO sensor, the literature works are classified based on how the raw sensor data (e.g., 2D images and 3D point clouds) are integrated within the filtering scheme. In this respect, a distinction can be made between tightly-coupled and loosely-coupled approaches [15]. In the tightly-coupled case, the raw sensor data are processed to detect a set of features, i.e., landmarks associated to the geometric structure of the target, whose position is included within the state vector of the filter. In the loosely-coupled case, raw sensor data are processed by a separate pose determination block to obtain relative position and attitude estimates which are used as measurements by the Kalman Filter.

Some examples of tightly-coupled schemes can be found in [16]–[18]. Pesce et al. proposed a tightly-coupled relative navigation algorithm exploiting simulated 3D features from a stereo-vision system as measurements [16]. In this work, an Extended Kalman Filter (EKF) is compared against an Iterated EKF (IEKF) in terms of relative state estimation performance. Volpe et al. proposed an Unscented Kalman Filter (UKF) for relative state estimation and shape reconstruction of an uncooperative space target exploiting simulated 2D features from a monocular camera and the target-chaser range from a distance sensor (e.g., a laser range finder) as measurements [17]. Finally, Wang et al. presented an adaptive UKF designed to refine the estimate of the moment of inertia ratios of the target [18], using again simulated 3D features from a stereo-vision system as measurements. Overall, the use of tightly-coupled architectures has advantages when dealing with uncooperative, unknown targets, which can be found in some ADR scenarios (if the target is seriously damaged, e.g., due to onboard explosions or hypervelocity impacts) and in space exploration missions (if the target is a fully unknown comet or asteroid). Indeed, they allow getting information about the target geometry by estimating either the target shape or inertia properties (e.g., by adding the moment of inertia ratios to the state vector of the navigation filter, although the inertia matrix estimation problem is not fully observable). Main drawbacks of such architectures are the computational effort (which increases with the number of detected features), and the need to develop feature detection and tracking algorithms robust enough to deal with the recursive appearance/disappearance of features without generating outliers (which is particularly challenging in the case of fast relative dynamics).

When the target geometry is known, as in most OOS and ADR scenarios, loosely-coupled architectures are typically preferred [19]–[23]. One important advantage is the possibility to run the pose determination block at lower frequency than the filter prediction, which may allow relaxing computational requirements without compromising the achieved accuracy. The knowledge of the target geometry is typically exploited by entrusting the pose determination task to model-based algorithms [9]. A dual inertial multiplicative EKF can be found in [19], where the target-chaser pose estimates are provided to the filter by processing simulated LIDAR data with the Oriented, Unique, and Repeatable Clustered Viewpoint Feature Histograms (OUR-CVFH) algorithm. The concept of unit dual quaternion, which provides a concise and compact representation of the relative position and attitude parameters, is proposed in [20]. Two filtering schemes, i.e., a standard EKF and an adaptive fading factor EKF, are compared, using simulated pose measurements as provided by a vision-based system. A vision-based pose estimator (including algorithms for pose initialization and tracking which process simulated 2D corner features) is integrated with a filtering architecture in [21]. Specifically, two separate filters, i.e., an H∞ filter and a second-order, minimum-energy filter on the Lie group, are exploited for the relative orbital and rotational dynamics, respectively. Finally, a high-order EKF exploiting simulated pose estimate from a visual system is proposed in [22].

This analysis shows that very few works have addressed the relative navigation problem without relying on simulated pose estimates, i.e., considering also the task of processing raw data (either 2D images or 3D point clouds). Instead, the loosely-coupled relative navigation architecture proposed in this paper addresses all the processing steps required to estimate the target-chaser relative state starting from raw LIDAR measurements. State-of-the-art pose determination algorithms are exploited within the proposed architecture to play a twofold role. On one side, they are used within a processing strategy designed to initialize the filter state and covariance when no prior information is available about the target-chaser relative state. On the other side, the Iterative Closest Point (ICP) algorithm is adopted to provide updated pose measurements to the filter each time a new set of LIDAR data is acquired [13]. An autonomous failure detection strategy, based on the evaluation of the ICP-based cost function, is also exploited to avoid providing a wrong initialization to the filter as well as to avoid the filter to diverge, by commanding a new execution of the initialization process. Regarding the filtering scheme, an original Unscented Kalman Filter (UKF) is designed which uses as measurement for the update phase, relative position, attitude and angular velocity estimates. The choice of using an UKF scheme is motivated by the fact that it provides better performance than other filtering approaches (such as the Extended Kalman Filter) to deal with the non-linearities in the quaternion-based relative rotational dynamics model [23]. Finally, another important contribution of this work is that the measurement covariance matrix is not a fixed parameter selected by the user but rather it is updated after each implementation of the ICP algorithm. This allows properly weighting the level of trust in the relative position and attitude estimates.

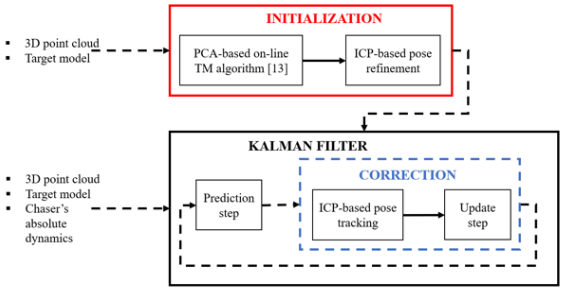

FIGURE 1. Block diagram summarizing the proposed LIDAR-based relative navigation architecture. The inputs required by the two main blocks are listed on the left side of the figure.

III. RELATIVE NAVIGATION ARCHITECTURE

The block diagram describing the proposed relative navigation architecture is shown in Fig. 1. It is composed of two main parts. The former, indicated as ''Initialization'', aims at providing a reliable estimate of the target-chaser pose parameters for starting the filtering process. It requires in input measured point clouds and a model of the target geometry (no prior knowledge of the relative state is needed). Details about the initialization strategy including fundamentals regarding the adopted model-based pose determination algorithms are collected in Section III-B. At the end of the initialization, an UKF scheme is adopted to estimate the entire relative navigation state. The main theoretical concepts about UKF are provided in Section III-C, while details of the proposed implementation are given in Section III-D. Due to the loosely-coupled nature of the proposed architecture, a pose determination block, relying on the ICP algorithm, is adopted within the update step of the UKF to properly correct the relative navigation state. Before discussing the details about the above-mentioned parts of the proposed architecture, preliminaries about the adopted coordinate systems and terminology for the relative state vector are collected in Section III-A.

A. COORDINATE SYSTEMS AND RELATIVE STATE PARAMETRIZATION

A list of the reference frames required for the mathematical derivation of the following sub-sections is provided below.

- The Inertial reference frame (IRF) is the classical Earth-centered inertial frame; the z and x axes point towards North and the first point of Aries, respectively, while the y axis completes the right-handed coordinate system.
- The Sensor reference frame (SRF) is the LIDAR coordinate system whose z-axis is directed along the boresight direction.
- The Chaser-fixed reference frame (CRF) is the body-reference frame of the chaser; for the sake of the mathematical simplicity, it is assumed that it is coincident and aligned with the SRF.
- The Target-fixed reference frame (TRF) is the body-reference frame of the target.

- The Hill's reference frame (HRF) has the same origin of the CRF. The x axis is in the chaser radial direction; the z axis is parallel to the angular momentum vector of the chaser and the y axis completes the right-handed triad.
- The Target orbit reference frame (TORF) has the z axis in the radial direction (toward Earth center), the x axis along the angular momentum vector, while the y axis completes the right-handed triad.

The proposed architecture aims to estimate a 13 × 1 state vector ($\boldsymbol{x}$) defined as follows,

$$\boldsymbol{x} = [\boldsymbol{\rho}\ \dot{\boldsymbol{\rho}}\ \boldsymbol{q}\boldsymbol{\omega_T}] \tag{1}$$

where $\boldsymbol{\rho}$ and $\dot{\boldsymbol{\rho}}$ are 3×1 vectors representing the relative position and velocity of the target with respect to the chaser, both expressed in HRF; $\boldsymbol{q}$ is a 4×1 vector representing the unit quaternion of the SRF attitude with respect to the TRF; finally, $\boldsymbol{\omega_T}$ is a 3×1 vector representing the target angular velocity and it is expressed in TRF. It is worth recalling that the target-chaser relative attitude is alternatively parameterized using the rotation matrix ($\boldsymbol{R}$) or also a 321 sequence of Euler angles (i.e. yaw, γ, pitch, β, and roll, α).

B. INITIALIZATION STRATEGY

In order to initialize the filter, an estimate of the target-chaser pose parameters as well as of the associated covariance must be provided. This task is entrusted to a combination of state-of-the-art model-based pose determination algorithms [13]. The initialization procedure is now detailed by also recalling, for the sake of clarity of the discussion, the main concepts regarding the adopted pose determination algorithms.

1) INITIAL POSE ACQUISITION AND REFINEMENT

Once the first point cloud is acquired by the LIDAR system on board the chaser, an initial coarse estimate of the pose parameters is firstly obtained applying the *on-line PCA-based Template Matching*(PCA-TM) algorithm [13]. It consists of the following phases. First, a tentative solution for the relative position vector of the chaser with respect to the target ($\boldsymbol{T}$) is computed as the centroid of the point cloud. Second, according to the Principal Component Analysis [24], the direction of the target main axis is estimated as the eigenvector corresponding to the maximum eigenvalue associated to the point cloud covariance matrix. While two rotational DOFs (represented by the α-β couple of Euler angles) can be directly derived from this direction, the remaining one (represented by γ, i.e., the rotation of the target about its main axis with respect to SRF) is computed applying a TM approach [13]. Since it is not possible to directly establish whether the target main axis is parallel or antiparallel to the direction estimated by the PCA, an ambiguity arises in the estimation of α and β, and, consequently, in the relative attitude quaternion. This ambiguity in the pose vector $\boldsymbol{p} = [\boldsymbol{T}, q]$ is solved applying twice the ICP algorithm (whose main concepts are recalled in the next sub-section) and choosing the solution characterized by the minimum value of the associated cost function [13]. To further refine the achieved pose solution and, consequently, to initialize the filter with a more accurate pose estimate, a short time interval (t_{INIT}, which can be set to a few seconds) is assigned during which the ICP algorithm is applied to update the pose parameters each time a new point cloud is acquired.

It is worth outlining that the PCA-TM algorithm is tailored to targets having a main geometric direction (indeed, most resident space objects, such as rocket bodies, have an elongated structure). However, if the target does not have such a principal direction, the initialization scheme can still be applied by substituting the PCA-TM algorithm with different state-of-the-art techniques able to provide an estimate of the pose parameters without relying on a prior knowledge [19], [26]–[30].

2) ICP FUNDAMENTALS

The ICP algorithm is an iterative technique which aims to find the best estimate of the rotation and translation necessary to align two datasets [30]. This problem is also called registration in the robotic research community. The implementation of the ICP algorithm can be easily customized to the application of interest considering that each ICP iteration is composed of multiple phases for which several variants exist [31]. Among these phases, three are mandatory, namely the matching step, the selection of the error metric function and of the minimization technique. In this work, the ICP algorithm is implemented, as described in [13], to estimate the pose solution which best aligns the measured point cloud (expressed in SRF) to a model point cloud (expressed in TRF) and obtained from the knowledge of the target geometry. While the details can be found in [13], it is here important to recall the definition of the ICP cost function (f_{ICP}), which is defined as the mean squared distance between corresponding model and measured points, as shown in (2),

$$f_{ICP}\left(\boldsymbol{T}, q\right) = \frac{1}{N_p} \sum_{i=1}^{N_p} \left| \boldsymbol{P}_M^i - \underline{\underline{R}}\left(q\right)^T \left(\boldsymbol{P}_L^i + \boldsymbol{T}\right) \right|^2 \tag{2}$$

where $\boldsymbol{P}_M^i$ and $\boldsymbol{P}_L^i$ are the corresponding points in the model and measured point cloud, respectively, while N_p is the size of the measured point cloud. At each ICP iteration, after the matching process, the pose parameters are estimated as the ones that minimize this cost function using a closed-form solution based on unit quaternions. The iterative process is ended when the variation of f_{ICP} between two consecutive iterations becomes lower than 10^{-6} m^2, or if the iteration number exceeds a maximum limit (e.g. 30 iterations).

As anticipated in the previous sub-section, the value of f_{ICP} at the end of the iterative process (f_{END}) plays a key role in the initialization procedure as it allows solving the ambiguity between the two pose solutions arising from the application of the PCA-TM algorithm. This is possible since f_{END} is a quantitative measure of the quality of the registration process and, consequently, of the accuracy in the estimated pose parameters [13]. This result is exploited in this work to

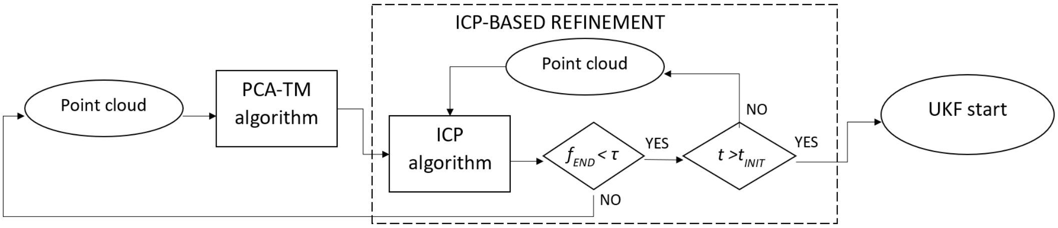

FIGURE 2. Flow diagram summarizing the filter initialization procedure. The ICP-based refinement block is enclosed in a dashed rectangular box to indicate that this phase is conducted over a time interval (t_{INIT}). Specifically, a new point cloud is acquired and the ICP algorithm is applied until the time spent for the starting of the initialization process reaches (t) reaches the value assigned to t_{INIT}. At each ICP implementation, if the failure detection condition ($f_{END} < \tau$) is not satisfied, a new point cloud is acquired, and the initialization process is restarted.

define an autonomous failure detection strategy based on the evaluation of f_{END}. Specifically, each time the ICP algorithm is applied during the initialization process, f_{END} is compared to a threshold (τ) selected by the user. If f_{END} is larger than τ, a new point cloud must be acquired, and the initialization process is restarted. A flow diagram summarizing the initialization process, including this ICP-based failure detections strategy is reported in Fig. 2.

Besides the pose parameters, the ICP algorithm must also provide an estimate of the associated covariance matrix to the UKF. This task is carried out by adapting the approach proposed in [32] to the ICP formulation used in this work. According to [32], the covariance matrix associated to the ICP-based pose estimate can be estimated as in (3).

$$cov(\boldsymbol{p}) = \left(\frac{\partial^2 f_{ICP}}{\partial \boldsymbol{p}^2}\right)^{-1} \left(\frac{\partial^2 f_{ICP}}{\partial \boldsymbol{s} \partial \boldsymbol{p}}\right) \times cov(\boldsymbol{s}) \left[\left(\frac{\partial^2 f_{ICP}}{\partial \boldsymbol{p}^2}\right)^{-1} \left(\frac{\partial^2 f_{ICP}}{\partial \boldsymbol{s} \partial \boldsymbol{p}}\right)\right]^T \quad (3)$$

where $\boldsymbol{p}$ is the pose estimate vector and $cov(\boldsymbol{s})$ is the $3Np$-by-$3Np$ diagonal matrix which measures the error level in the acquired point cloud, computed applying an offline statistical analysis which considers the noise level in the simulated LIDAR system.

C. UNSCENTED KALMAN FILTER

The Kalman filter is an algorithm able to provide the optimal estimate of a state vector, given a set of measurements over time, under a set of assumptions [33]. Specifically, the temporal evolution of the state vector and the relation between the state vector and the measurements are ruled by linear models (dynamic process and observation models, respectively), which are affected by random Gaussian white noises uncorrelated with each other. However, in the most general case, the dynamic and observation models are ruled by non-linear functions (f and h), as shown in (4) and (5),

$$\dot{x} = f(\boldsymbol{x}) + \boldsymbol{w} \quad (4)$$

$$z = h(\boldsymbol{x}) + \boldsymbol{v} \quad (5)$$

where $\boldsymbol{x}$ and $\boldsymbol{z}$ are the state and measurement vector, respectively, while $\boldsymbol{w}$ and $\boldsymbol{v}$ are the process and measurement noise, respectively. The process and measurement covariance matrixes are given by (6) and (7),

$$\boldsymbol{Q} = E\{\boldsymbol{w}\boldsymbol{w}^T\} \quad (6)$$

$$\boldsymbol{R} = E\{\boldsymbol{v}\boldsymbol{v}^T\} \quad (7)$$

where $E\{\}$ is the expected value operator. To deal with the non-linearity in the dynamic and observation models, different techniques can be exploited, such as the EKF [33] and UKF [34]. In this paper, an UKF scheme is implemented within the proposed relative navigation architecture. The UKF algorithm is based on the concept of the unscented transform, which allows avoiding the derivation of the Jacobian matrices required in the Extended Kalman Filter formulation [34]. The main steps of the discrete time formulation of the UKF algorithm are described in the following. First, the columns of the state covariance matrix ($\boldsymbol{P}$) are used to construct the so-called sigma-points (χ), i.e., a set of state vectors chosen so that their sample mean and sample covariance are equal to the mean and covariance of the current state. They can be computed as shown in (8),

$$\begin{aligned} \boldsymbol{\chi}_{k-1}^{1} &= \hat{x}_{k-1} \\ \boldsymbol{\chi}_{k-1}^{i} &= \hat{x}_{k-1} + \sqrt{L+\lambda}\boldsymbol{W_i}, i = 2, \ldots .L+1 \\ \boldsymbol{\chi}_{k-1}^{i} &= \hat{x}_{k-1} - \sqrt{L+\lambda}\boldsymbol{W_i}, i = L+2, \ldots .2L+1 \end{aligned} \quad (8)$$

where k is the current discrete time instant, $\boldsymbol{W_i}$ the i^th column vector of the Cholesky decomposition of $\boldsymbol{P}$, L is the dimension of the state vector and λ is a scale factor computed as shown in (9),

$$\lambda = \alpha^2 (L + c) - L \quad (9)$$

where α and c are user-defined parameters. The parameter α defines the spread of the sigma points around the mean of the statistical distribution, and it is a positive integer lower than 1, while c is another setting parameter typically set to zero [23].

The sigma points are projected ahead in time by means of the dynamic model (prediction step). Then, an estimate of the mean (10) and covariance (11) of the state vector is obtained assuming that the predicted sigma points sample a Gaussian distribution.

$$\hat{x}_k = \sum_{i=1}^{2L} w_m^i f(\boldsymbol{\chi}_{\boldsymbol{k-1}}^{\boldsymbol{i}}) \quad (10)$$

$$\boldsymbol{P}_{xx} = \sum_{i=1}^{\boldsymbol{2L}} w_c^i \left(\boldsymbol{\chi}_{\boldsymbol{k}}^{\boldsymbol{i}} - \hat{x}_k\right)\left(\boldsymbol{\chi}_{\boldsymbol{k}}^{\boldsymbol{i}} - \hat{x}_k\right)^{\boldsymbol{T}} + \boldsymbol{Q} \quad (11)$$

In the equations above, w_C^i and w_M^i are weights defined as follows.

$$\begin{aligned} w_m^1 &= \frac{\lambda}{L+\lambda} \\ w_c^1 &= \frac{\lambda}{L+\lambda} + (3-\alpha^2) \\ \boldsymbol{w_m^i} &= \boldsymbol{w_c^i} = \frac{1}{2\,(L+\lambda)} \end{aligned} \tag{12}$$

The predicted sigma-points are also projected in the measurement space in order to get an estimate of the mean and covariance matrix of the state vector in that space (13-15).

$$\boldsymbol{\gamma_k^i} = \boldsymbol{h}(\boldsymbol{\chi_k^i}) \tag{13}$$

$$\hat{z}_k = \sum\nolimits_{i=1}^{2L} w_m^i \boldsymbol{\gamma_i} \tag{14}$$

$$\boldsymbol{P}_{zz} = \sum\nolimits_{\boldsymbol{i=1}}^{\boldsymbol{2L}} w_c^i \left(\boldsymbol{\gamma_k^i} - \hat{z}_k\right)\left(\boldsymbol{\gamma_k^i} - \hat{z}_k\right)^{\boldsymbol{T}} + \boldsymbol{R} \tag{15}$$

In the UKF, the Kalman gain is computed by means of (16),

$$\boldsymbol{K}_k = \boldsymbol{P}_{xz}\boldsymbol{P}_{zz}^{-1} \tag{16}$$

where the cross-covariance matrix P_{xz}, is computed as shown in (17)

$$\boldsymbol{P}_{xz} = \sum\nolimits_{i=1}^{\boldsymbol{2L}} w_c^i \left(\boldsymbol{\chi_k^i} - \hat{x}_k\right)\left(\boldsymbol{\gamma_k^i} - \hat{z}_k\right)^{\boldsymbol{T}} \tag{17}$$

Finally, the state vector and the covariance matrix are updated as in (18) and (19)

$$\hat{x}_{\boldsymbol{k}}^{+} = \hat{x}_{\boldsymbol{k}}^{-} + \boldsymbol{K_k}\left(z_k - \boldsymbol{h}\left(\hat{x}_{\boldsymbol{k}}^{-}\right)\right) \tag{18}$$

$$\boldsymbol{P}_k = \boldsymbol{P}_{k-1} - \boldsymbol{K}_k\boldsymbol{P}_{zz}\boldsymbol{K}_k^T \tag{19}$$

D. UKF FOR LOOSELY-COUPLED RELATIVE NAVIGATION

First, the dynamic and observation models adopted in the proposed UKF implementation are defined. Recalling that the state vector defined in (1) includes information about target-chaser relative position, velocity and attitude, as well as about the absolute angular velocity of the target, different dynamics models are needed for the different state variables. With regards to the relative translational dynamics, the relative motion model presented in [35] is adopted. It is obtained from the combination of the two-body problem of the target and chaser without any further assumption on the shape of the orbit. It comprises a system of non-linear differential equations expressed in HRF, as shown in (20),

$$\begin{aligned} \ddot{x} &= \frac{\mu}{r_c^2} - \frac{\mu\,(r_c+x)}{\left[(r_c+x)^2+y^2+z^2\right]^{\frac{3}{2}}} + 2\dot{y}\omega_H + y\dot{\omega}_H + x\omega_H^2 \\ \ddot{y} &= -\frac{\mu y}{\left[(r_c+x)^2+y^2+z^2\right]^{\frac{3}{2}}} - 2\dot{x}\omega_H + x\dot{\omega}_H + y\omega_H^2 \\ \ddot{z} &= -\frac{\mu z}{\left[(r_c+x)^2+y^2+z^2\right]^{\frac{3}{2}}} \end{aligned} \tag{20}$$

where μ is the Earth's gravitational constant, $\boldsymbol{r_C}$ is the norm of the position vector of the chaser spacecraft, ω_{H} and $\dot{\omega}_{\mathrm{H}}$ are the angular velocity and acceleration of HRF in IRF, respectively, while x, y and z are the Cartesian components of the relative position vector $\boldsymbol{\rho}$.

The relative attitude model is represented by (21) which describes the kinematic evolution in time of the relative attitude quaternion [36], while the absolute rotational dynamics of the target is described by the well-known Euler equation (22) under the free-body assumption (i.e., disturbance torques are not included in the model).

$$\dot{q} = \frac{1}{2}[0\boldsymbol{\omega}_{T/C}] \otimes \boldsymbol{q} \tag{21}$$

$$\dot{\omega}_T = -I_T^{-1}(\boldsymbol{\omega}_T \times I_T\boldsymbol{\omega}_T) \tag{22}$$

In (21), $\omega_{T/C}$ or two different values e chaser with respect to the target in TRF and the symbol ⊗ indicates a quaternion product; in (22), I_T is the inertia matrix of the target, assumed to be known.

It is worth highlighting that a relative rotational dynamic model can be used instead of considering the absolute rotational dynamics of the target. Both possibilities have been investigated in the literature (see [16] and [17], respectively). The latter choice is here preferred since it does not require to model the control torques acting on the chaser spacecraft.

As regards the observation model, each time a new LIDAR point cloud is acquired, the ICP algorithm (see Section III-B) is initialized based on the predicted state vector thus producing an accurate estimate of the pose parameters ($\boldsymbol{p} = [\boldsymbol{T}, q]$) and of the associated covariance. In this case, the measurement vector (z) is equal to $\boldsymbol{p}$, the observation model is represented by the following set of linear equations, as shown in (23),

$$z = \boldsymbol{p} = \begin{bmatrix} -R_{HRF}^{SRF} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & I_{3x3} & 0_{3x3} \end{bmatrix} \begin{pmatrix} \boldsymbol{\rho} \\ \dot{\rho} \\ \boldsymbol{q} \\ \boldsymbol{\omega_T} \end{pmatrix} \tag{23}$$

where R_{HRF}^{SRF} is the rotation matrix from HRF to SRF. It is worth outlining that the autonomous failure detection strategy used within the initialization scheme (see Section III-B) is similarly applied also at this stage of the proposed relative navigation architecture. Specifically, if f_{END} is larger than τ, the pose measurements are not used to update the state vector (i.e., only the prediction block of the UKF is adopted). If this condition occurs N_f consecutive times, the initialization process must be restarted.

Due to the unit norm constraint in the attitude quaternion, a singularity problem arises in the estimation of the covariance matrixes in the filter implementation making it no more invertible (see (16)). So, while the quaternion parameterization is used to represent the attitude state, the attitude error is represented by a three-parameters attitude metric, namely the Gibbs vector ($\boldsymbol{g}$) [37]. This strategy leads to a reduction in the dimension of the covariance matrixes, thus avoiding singularity issues. Clearly this is possible since, under the assumption of small angular errors, the relation between the quaternion error ($\delta\boldsymbol{q}$) and the corresponding Gibbs vector is given by (24) [37].

$$\boldsymbol{\delta q} \approx \begin{bmatrix} 1 \\ \boldsymbol{g} \end{bmatrix} \tag{24}$$

Due to the weak coupling between the attitude and the angular velocity of the target in the dynamic model (see (21) and (22)), the effect of the pose measurements alone in the update of the angular velocity of the target is limited. So, in order to improve the related estimation accuracy level, a measure of the angular velocity of the target ($\boldsymbol{\omega}_{T,EST}$) can be supplied to the filter exploiting a time sequence of attitude measurements (and, consequently, at a lower frequency than the LIDAR frame rate).

The procedure to compute ω_{TEST} is now described. Given the attitude quaternion from the implementation of the ICP algorithm, and the inertial attitude of the chaser ($\boldsymbol{q}_{C/I}$, supplied by its onboard navigation system), the inertial attitude of the target ($\boldsymbol{q}_{T/I}$) can be computed.

$$\boldsymbol{q}_{T/I} = \boldsymbol{q} \otimes \boldsymbol{q}_{C/I} \tag{25}$$

The above-estimated attitude can be expressed in the form a 313 sequence of Euler angles, namely ψ, θ, and φ through (26).

$$\begin{aligned} \varphi &= atan2(2q_1q_3 - 2q_0q_2, 2q_2q_3 + 2q_0q_1) \\ \theta &= \cos^{-1}(q_3^2 - q_2^2 - q_1^2 + q_0^2) \\ \psi &= atan2(2q_1q_3 + 2q_0q_2, 2q_0q_1 - 2q_2q_3) \end{aligned} \tag{26}$$

Given the estimates of these Euler angles at different time instants, their first derivative ($\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$) can be computed by exploiting a first-order, finite-difference scheme [38]. Finally, the 313 kinematics equation (27) are used to get a measure of the target angular velocity expressed in TRF [39].

$$\begin{aligned} &\boldsymbol{\omega}_{T,EST} \\ &= \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\varphi} \end{pmatrix} \\ &\quad + \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} \dot{\theta} \\ 0 \\ 0 \end{pmatrix} \\ &\quad + \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \\ &\quad \times \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \end{aligned} \tag{27}$$

This angular velocity estimate is provided to the filter at a lower frame rate (i.e., 0.1 Hz in this work) than the one characterizing the pose measurements provided by the ICP algorithm, in order to smooth the error introduced by the numerical derivative of ψ, θ and φ (which is related to the noise in the relative attitude quaternion estimates). So, when a measure of the target angular velocity is available, the observation model is modified, as shown in (28).

$$\boldsymbol{z} = \begin{bmatrix} -R_{HRF}^{SRF} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & I_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & I_{3x3} \end{bmatrix} \begin{pmatrix} \boldsymbol{\rho} \\ \dot{\boldsymbol{\rho}} \\ \boldsymbol{q} \\ \boldsymbol{\omega_T} \end{pmatrix} \tag{28}$$

Clearly, the dimension of the measurement vector $\boldsymbol{z}$ is changed adaptively depending on the availability of the target angular velocity estimates.

IV. SIMULATION ENVIRONMENT, SCENARIO AND RESULTS

To fully assess performance of the proposed relative navigation architecture, a numerical simulation environment, which can reproduce the operation of a scanning LIDAR, as well as the target-chaser relative dynamics corresponding to a realistic OOS or ADR scenario, is exploited. Fig. 3 summarizes the structure of the simulation environment, which is composed of three modules dedicated to the relative dynamics, LIDAR measurement and relative navigation process, respectively. The relative dynamics module (highlighted in blue in Fig. 3) is conceived to reproduce inspection or monitoring trajectories of a chaser with respect to an uncooperative target. Indeed, target monitoring is one of the key phases of a close-range rendezvous manoeuvre [40].

Regarding the translational dynamics, the module requires in input the mean orbit parameters of the target, and a set of parameters which characterize the size and shape of the desired monitoring trajectory. Using the trajectory design method described in [41], the differences between the target-chaser mean orbit parameters which characterize the desired trajectory can be computed (and, consequently, also the mean orbit parameters of the chaser are derived). At this point, the absolute orbital dynamics is numerically propagated for both the chaser and the target by a numerical orbit propagator in which the main perturbations are included (i.e., aerodynamic drag, harmonics up to the fourth order and solar radiation pressure). Clearly, the relative position and velocity information can be obtained from the two absolute orbital dynamics. With regards to the relative rotational dynamics, the absolute attitude of the target is obtained by integrating the Euler's equations considering the gravity gradient torque as only disturbance. Then, the rotation matrix describing the attitude between TRF and SRF is obtained by assuming that the chaser attitude is controlled to point the LIDAR boresight axis toward the target geometric center.

The target-chaser relative attitude and position parameters are then used as input by the LIDAR measurement module (orange block in Fig. 3). Once the LIDAR Field of View (FOV) and angular resolution (δ_{LOS}) are assigned, this module is able to generate a set of raw LIDAR data, i.e., a point cloud, in three steps [27]. First, a purely geometric point cloud is obtained applying a ray tracing algorithm (no source of noise considered). Second, the detection process is simulated by computing for each backscattered laser beam the probability of detection as a function of the probability of

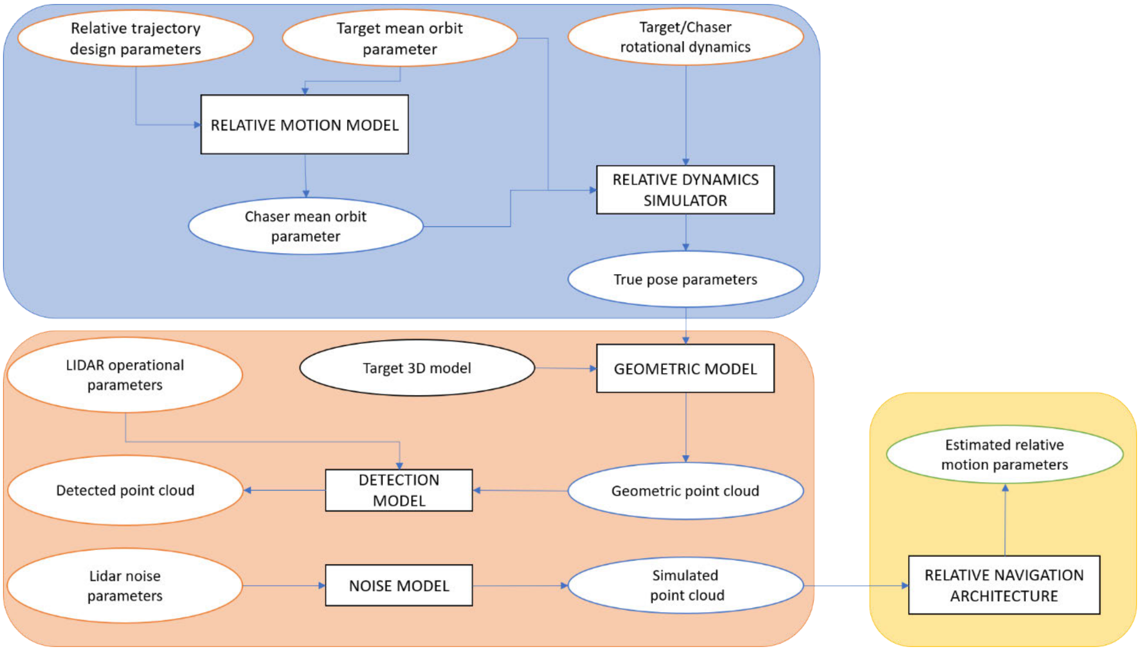
Relative trajectory design parameters
Target mean orbit parameter
Target/Chaser rotational dynamics
RELATIVE MOTION MODEL
RELATIVE DYNAMICS SIMULATOR
Chaser mean orbit parameter
True pose parameters
LIDAR operational parameters
Target 3D model
GEOMETRIC MODEL
Estimated relative motion parameters
Detected point cloud
DETECTION MODEL
Geometric point cloud
Lidar noise parameters
NOISE MODEL
Simulated point cloud
RELATIVE NAVIGATION ARCHITECTURE

FIGURE 3. Flow diagram summarizing the main blocks of the simulation environment. The three main modules are highlighted with colors: relative dynamics module (blue); LIDAR measurement module (orange); relative navigation module (yellow).

false alarm and the Signal to Noise Ratio [42]. This can be done by evaluating the signal mean value based on the LIDAR equation, while the noise is determined considering the background and the thermal noise contributions [43]. Finally, the detected point cloud is modified considering the sensor measurement uncertainties, which are simulated as a gaussian white noise on the measured range (σ_ρ) and laser beam direction (σ_{LOS}) [27]. The possibility to produce outliers as a percentage of the detected points ($O_{\%}$) is also considered. Specifically, they are randomly extracted among the elements of the measured point cloud and their range uncertainty is set to four times σ_ρ. The measured point clouds obtained through this procedure are the main input data for testing the relative navigation module (yellow block in Fig. 3) in which the algorithmic approach described in Section III is applied obtaining a temporal evolution of the estimated relative state which must be compared to the ground truth for performance assessment.

A. SIMULATION AND TEST CASE

The target considered in this simulation scenario is ENVISAT, an eight-tones Earth observation satellite declared inoperative by ESA in 2012. The LIDAR measurement simulator described in the previous section, as well as the model-based pose determination algorithms included in the relative navigation architecture, require information about the target geometry (e.g., size, shape, inertia, surface material). In this work, ENVISAT geometry is modeled as a combination of cuboid-shaped elements representing the main body, the solar array and the synthetic aperture radar (SAR) antenna (these two latter elements are attached to the main body by two appendixes), as depicted in Fig. 4.

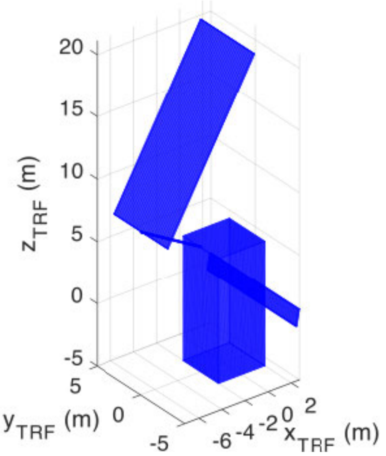

FIGURE 4. Simplified model of ENVISAT geometry. It is represented in the form of a 3D point cloud which is used as model point cloud within the ICP algorithm.

A synthesis of the main information, collected from the literature [44]–[46] adopted to generate this model and used for the LIDAR point cloud generation process are collected in Table 1.

A safety ellipse is designed around ENVISAT for the simulated monitoring scenario. The mean orbital parameters defining this trajectory are collected in Table 2, and the resulting safety ellipse, with a range varying from 25 to 57 meters, is depicted in Fig. 5.

With regards to the rotational dynamics, the scenario is generated assuming that, at the initial time, the TRF is aligned with the orbital reference frame of the target (TORF). The resulting time variations of the Euler angles representing the

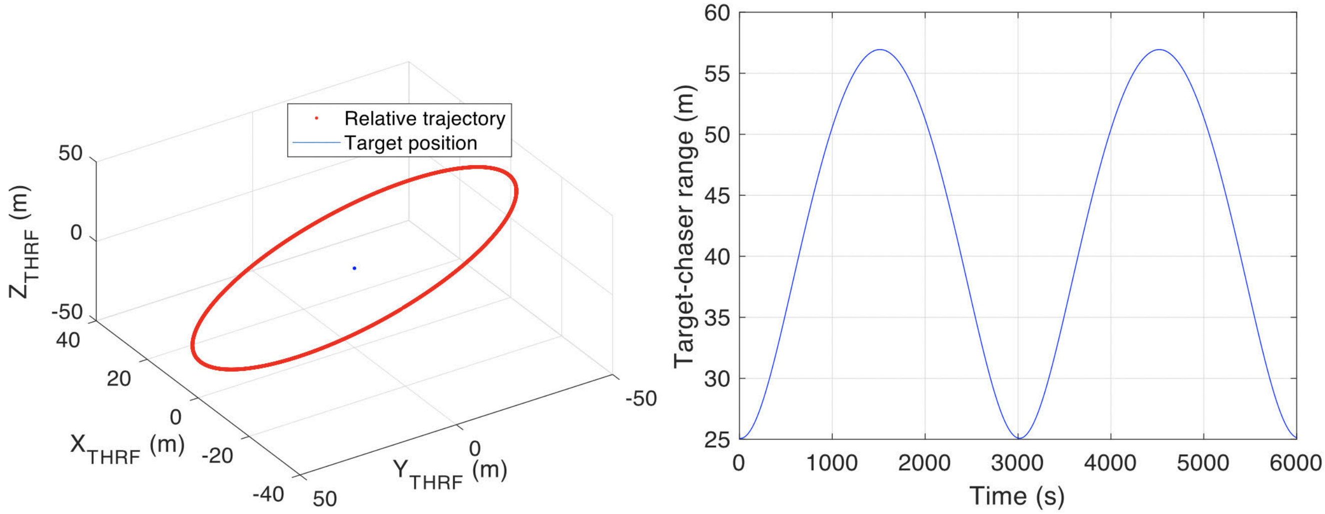

FIGURE 5. Safety ellipse designed around ENVISAT. (Left) Relative trajectory expressed in the THRF. (Right) Time-history of the target-chaser range.

TABLE 1. ENVISAT model: Geometrical and inertial characteristics.

Element	Dimension	Moments of inertia (kg·m^2)			
Main-body	4 m × 4 m x 10 m (cuboid)	I_{xx}	129112.2	I_{xy}	-344.2
Solar panel	6 m × 15 m	I_{yy}	124825.7	I_{xz}	2171.4
SAR antenna	1.3 m × 10 m	I_{zz}	17023.3	I_{yz}	397.1

TABLE 2. Target and chaser mean orbit parameters to get the desired safety ellipse. The target ones are taken from a set of True Line Elements relative to a recent measurement. The chaser ones are the output of the "relative motion model" mentioned in Fig. 3 [41].

Initial orbit parameters	Chaser	Target
Semi-major axis (km)	7135.31918	71353.31929
Inclination (°)	98.291	98.291
Eccentricity	0.0014519	0.0014553
Argument of perigee (°)	252.4586	252.4608
Right Ascension of ascending node (°)	191.3391	191.3393
Mean Anomaly (°)	186.35689	186.35478

relative attitude of the target with respect to the chaser are depicted in Fig. 6.

B. RESULTS

Results from different numerical simulations are now presented to evaluate performance of the proposed LIDAR-based relative navigation architecture.

Before entering the details, a set of error metrics must be properly defined. With regards to the initialization phase, the error in the estimation of the pose parameters, i.e., the components of $\boldsymbol{T}$ (Tx, Ty and Tz), and the triplet of 321 Euler angles corresponding to q (α, β and γ), is evaluated. With regards to the UKF, the error level is measured as the difference between the Euclidean norms of the true and estimated vectors for the relative position (29) and velocity (30) as well as for the target angular velocity (31).

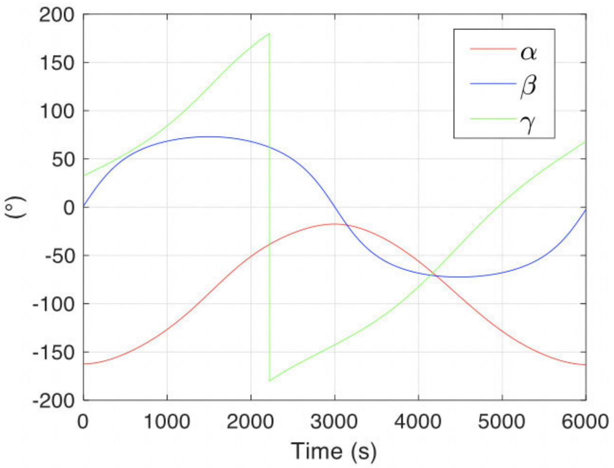

FIGURE 6. Time history of the euler angles (321 sequence) describing the attitude of SRF with respect to TRF.

$$|\boldsymbol{\rho}|_{ERR} = |\boldsymbol{\rho}|_{TRUE} - |\boldsymbol{\rho}|_{EST} \quad (29)$$

$$|\dot{\boldsymbol{\rho}}|_{ERR} = |\dot{\boldsymbol{\rho}}|_{TRUE} - |\dot{\boldsymbol{\rho}}|_{EST} \quad (30)$$

$$|\boldsymbol{\omega}_T|_{ERR} = |\boldsymbol{\omega}_T|_{TRUE} - |\boldsymbol{\omega}_T|_{EST} \quad (31)$$

Similarly, the relative attitude accuracy is expressed in terms of a single parameter (Φ_{ERR}), which is the equivalent Euler angle corresponding to the quaternion error ($\boldsymbol{q}_{ERR}$, (32)) defined as in (33),

$$\boldsymbol{q}_{ERR} = \boldsymbol{q}_{TRUE} \otimes \boldsymbol{q}_{EST}^{-1} \quad (32)$$

$$\phi_{ERR} = 2\cos^{-1}(q_{ERR}) \quad (33)$$

where q_{ERR} is the scalar part of the quaternion error.

Relative navigation performance achieved by the proposed architecture is now analysed for the scenario presented in Section IV-A. To this aim, the setting for the LIDAR operational and noise parameters is chosen to be consistent with performance of spaceborne systems [26], [47], [48].

Specifically, the simulated LIDAR is assumed to scan 40°-by-40° FOV, with δ_{LOS} equal to 1°, and update rate (f_L) of 1 Hz, while the noise parameters are selected considering typical performance of spaceborne, scanning LIDARs ($\sigma_{LOS} = 0.0007°$, $\sigma_\rho = 25$ mm) [26]. With regards to the presence of outliers in the measured point cloud, $O_\%$ is set to 5% for the set of detected points backscattered from the surfaces of the solar array and SAR antenna, while a larger value (7%) is considered for the measurements corresponding to the main body. This is done to better model the fact that ENVISAT is characterized by the presence of several devices on the main body surface which may produce outliers or generate multipath phenomena (and, consequently, larger errors in the measurements). For the sake of clarity, the setting parameters are listed in Table 3.

TABLE 3. LIDAR operational and noise parameters.

Operational parameters		Noise parameters	
FOV	40°x40°	σ_ρ	2,5 cm
δ_{LOS}	1°	σ_{LOS}	0.0007°
f_L	1 Hz	$O_\%$	5-7%

TABLE 4. Results of the initialization phase of the relative navigation architecture.

Pose parameters	Error after PCA-TM	Error after ICP-based pose refinement	
		Start	End ($t = T_{INIT}$)
α (°)	-19.65	-0.15	-0.06
β (°)	-19.16	0.12	0.08
γ (°)	2.56	-0.03	-0.04
T_X (m)	-0.365	$-1.7\cdot10^{-3}$	$0.4\cdot10^{-3}$
T_Y (m)	-0.777	$-2.1\cdot10^{-3}$	$-0.2\cdot10^{-3}$
T_Z (m)	-3.715	$1.8\cdot10^{-3}$	$4.8\cdot10^{-3}$

The results of the initialization process (T_{INIT} is set to 10 s) are collected in Table 4. Specifically, the PCA-TM algorithm provides a coarse estimate of the pose parameters which is exploited to initialize the ICP-based refinement process. In particular, the metric-order error in the relative position vector is justified by the use of the centroiding approach (which is not able to identify the position of the TRF origin, but rather it measures the average distance of the detected points distributed on the surfaces of the target in the FOV). Despite the relatively coarse accuracy achieved at the acquisition step the corresponding pose solution falls in the field of convergence of the ICP algorithm as shown by the accuracy level achieved at the start of the refinement process. Also, it is worth highlighting the improved accuracy in the estimation of α, β, T_X and T_Y (in particular, for these latter two terms, the error reduces around one order of magnitude) at the end of the refinement process, while the error level in γ and T_Z remains approximately constant. Besides the pose parameters, the UKF requires also an initialization of the target-chaser relative velocity and of the target angular velocity. The former is initialized as a null vector due to the proximity flight condition, while the latter is computed from the attitude measurements as already described in Section IIID.

At this point, the temporal variation of the error metrics selected to assess the UKF accuracy is depicted in Fig. 7 for the time duration of a single relative orbit. Despite the presence of a limited number of peaks, the relative navigation filter ensures an accuracy of 1-centimetre order in the relative distance (Fig. 7a), few-millimetres/s order in the relative velocity (Fig. 7b) and sub-degree order in the relative attitude (Fig. 7c). These peaks are associated to the periodic occurrence of unfavourable observation conditions of the target in the LIDAR FOV [13], [49]. For instance, they can be associated to self-occlusion phenomena, i.e., most of the target appearance is occluded by a single geometrical element (like the huge solar array in the ENVISAT case).

The presence of these peaks can be predicted on line by looking at the time variation of the ICP cost function at convergence, depicted in Fig. 8. Since the threshold value on $f_{END}(\tau)$ is set to 10^{-2} m^2 (thus being able to recognize major errors in the pose determination process), the autonomous failure detection strategy does not lead to the restart of the initialization process. Anyway, the worsening in the accuracy of the pose estimates (associated to the increase in f_{END} in Fig. 8) is considered thanks to the on-line computation of the ICP-based covariance matrix (see Section III-B).

By comparing the results in Fig. 7a with the time variation of the true target-chaser range (see Fig. 5-right), it is interesting to highlight that the error in the estimated relative distance gets slightly larger at increasing range. This can be explained considering that the average number of detected points reduces significantly with the range, so, the point cloud becomes extremely sparse, as shown in Fig. 9, and, consequently, the pose determination process gets more challenging.

Finally, with regards to the target angular velocity determination, the time variation of the Euclidean norm for both the true and estimated vectors (rather than the corresponding error, as done for the other output elements of the UKF) is depicted in Fig. 7d. This choice allows better showing that, while the mean estimation error is practically zero ($1.6\cdot 10^{-6}$ rad/s), the instantaneous error has an oscillatory behaviour. However, the peaks in this error are extremely low (i.e., they reach at most 3% of the true value of the target angular velocity).

In order to statistically characterize the estimation accuracy achieved by the proposed relative navigation architecture, a set of 100 simulations is run over the same scenario. Indeed, at each run, random extractions are done (*i*) for the detection

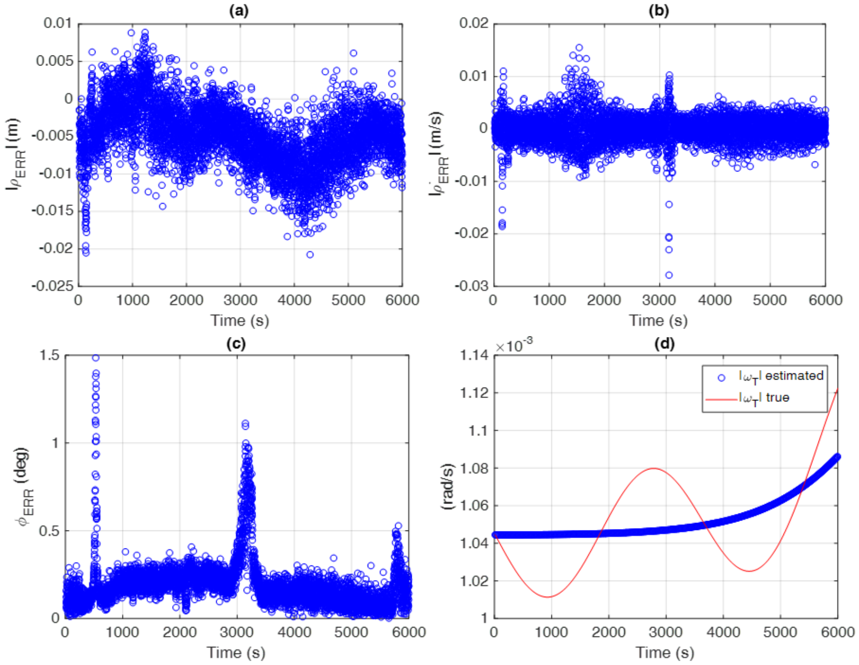

FIGURE 7. Time variation during one relative orbit of the error in the estimated (a) relative position, (b) relative velocity and (c) relative attitude. (d) Time variation during one relative orbit of the norm of the true and estimated target angular velocity.

process, (*ii*) to select the outliers among the elements of the detected point cloud corresponding to laser shots backscattered from the same target surface (based on the associated value of $O_{\%}$), (*iii*) and to evaluate the range and pointing errors associated to each detected point. So, the selected error metrics are first averaged at each time step over the 100 simulations. Then, their root mean square (*rms*) values can be computed over time. The results of this procedure are reported in Table 5.

TABLE 5. Statistics of the error metrics defined for performance assessment of the developed relative navigation architecture computed over one relative orbits (averaged on 100 simulations). The operational and noise parameters for the simulated LIDAR are listed in Table 3.

	$\lvert\rho\rvert_{ERR}$ (m)	$\lvert\dot{\rho}\rvert_{ERR}$ (m/s)	Φ_{ERR} (°)	$\lvert\omega\rvert_{ERR}$ (rad/s)
rms	$5.6\cdot10^{-3}$	$1.2\cdot10^{-3}$	0.23	$2.32\cdot10^{-5}$

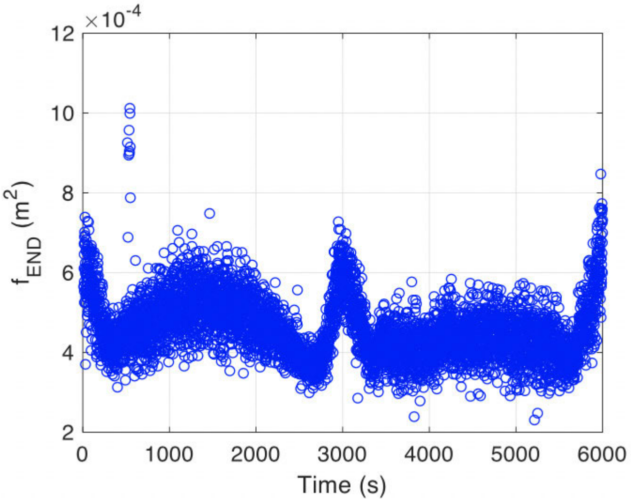

FIGURE 8. Time-variation of the ICP cost function at convergence during one relative orbit.

For the sake of completeness, the filter performance can be compared with the one achievable applying only the pose determination algorithm. For instance, the corresponding time variations of the error in the norm of the relative position is depicted in Fig. 10.

1) ROBUSTNESS ANALYSIS

The goal of this section is to fully characterize the performance of the proposed relative navigation architecture by evaluating its robustness and sensitivity against variations of the operational and noise parameters of the simulated LIDAR system.

a: LIDAR NOISE

First, the effect on the UKF performance obtained increasing the level of noise in the LIDAR measurements is analyzed. To this aim, a set of 100 simulations is carried out considering three different values of σ_ρ, increased with respect to the reference case (i.e., 2.5 cm, for which the results are presented in the previous sub-section). The results of a statistical analysis,

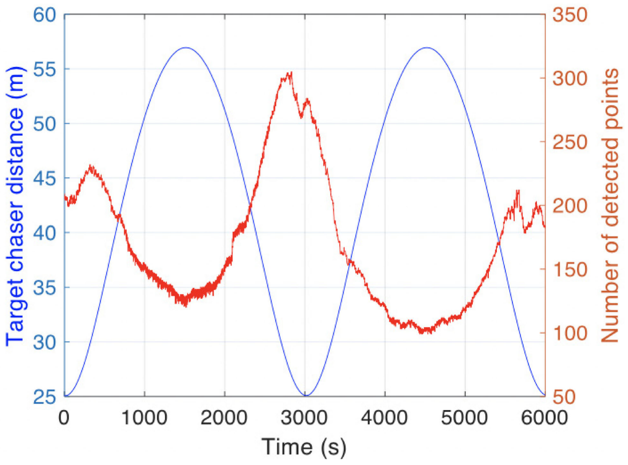

FIGURE 9. Time variation of the true target-chaser range (blue) and of the size of the detected point cloud (red) during one relative orbit.

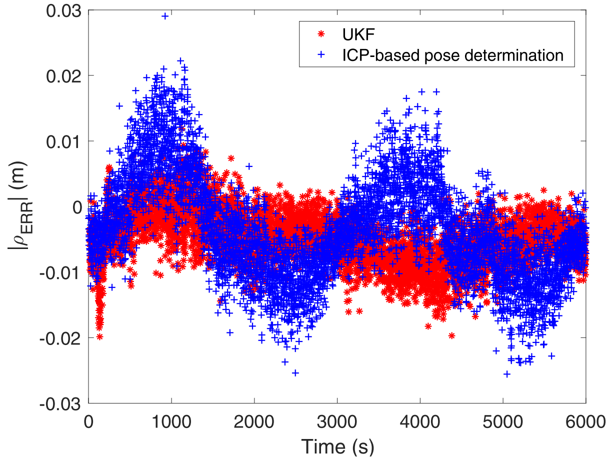

FIGURE 10. Time variation of during one relative orbit of the error in the estimated relative position given by UKF (red) and pose determination algorithm (blue).

TABLE 6. Statistics of the error metrics defined for performance assessment of the developed relative navigation architecture computed over one relative orbits (averaged on 100 simulations). Effect of the LIDAR range measurement accuracy. δ_{LOS} 1°, $f_L = 1$ Hz.

σ_ρ (cm)	$\|\boldsymbol{\rho}\|_{ERR}$ (m)	$\|\dot{\boldsymbol{\rho}}\|_{ERR}$	Φ_{ERR} (°)	$\|\boldsymbol{\omega}\|_{ERR}$ (rad/s)
	rms	*rms*	*rms*	*rms*
2.5	$5.6 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	0.23	$2.32 \cdot 10^{-5}$
5	$11.4 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	0.26	$2.35 \cdot 10^{-5}$
7.5	$16.4 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	0.32	$2.38 \cdot 10^{-5}$
10	$19.8 \cdot 10^{-3}$	$3.5 \cdot 10^{-3}$	0.43	$2.41 \cdot 10^{-5}$

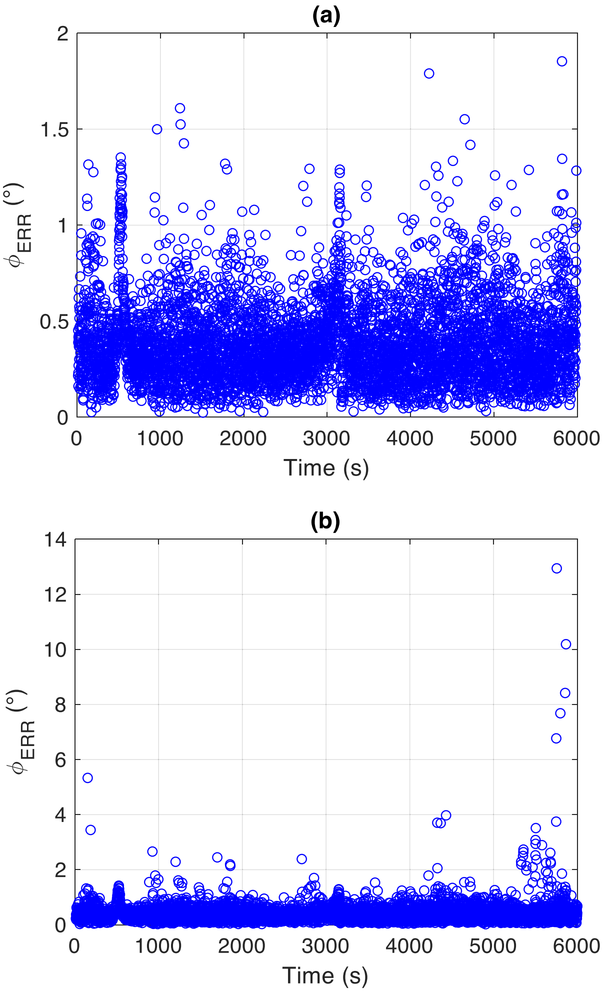

FIGURE 11. Time variation of φ_{ERR} during one relative orbit. (a) ICP-based failure detection and re-initialization enabled. (b) ICP-based failure detection and re-initialization disabled. $\sigma_\rho = 10$ cm. $\delta_{LOS} = 1°$, $f_L = 1$ H.

done adopting the same procedure described in the previous sub-section, are collected in Table 6.

As expected, the increase in the range measurement noise produces a reduction in the accuracy level achieved by the considered performance metrics. However, on the one side, the order of magnitude in the relative attitude and target angular velocity estimation error does not change, i.e., it remains below 0.5° and around $2 \cdot 10^{-5}$ rad/s, respectively. On the other side, the rms error in the relative position and velocity estimates becomes one order of magnitude larger when σ_ρ is increased from 2.5 cm to 10 cm. Anyway, it can be stated that the proposed UKF-based relative navigation architecture can provide a satisfying accuracy (cm-level and mm/s-level accuracy in relative position and velocity, respectively) despite dealing with an extremely large, conservative value of the range measurement noise.

The results of the simulations carried out setting σ_ρ to 10 cm also allow highlighting the advantage provided by an autonomous failure detection strategy in the pose determination process as the one adopted in this paper (see Section III-B and Section III-D). For instance, if τ is set to 10^{-2} m^2 and N_f is set to 3, the time variation of Φ_{ERR} for a single run is depicted in Fig. 11 either enabling or disabling the ICP-based failure detection strategy.

TABLE 7. Statistics of the error metrics defined for performance assessment of the developed relative navigation architecture computed over one relative orbits (averaged on 100 simulations). Effect of the LIDAR update rate. $\delta_{LOS} = 1°$, $\sigma_\rho = 2.5$ *cm*.

f_L (Hz)	$\lvert\rho\rvert_{ERR}$ (m)	$\lvert\dot{\rho}\rvert_{ERR}$ (m/s)	Φ_{ERR} (°)	$\lvert\omega\rvert_{ERR}$ (rad/s)
	rms	*rms*	*rms*	*rms*
1	$5.6\cdot10^{-3}$	$1.2\cdot10^{-3}$	0.23	$2.32\cdot10^{-5}$
0.50	$5.8\cdot10^{-3}$	$1.2\cdot10^{-3}$	0.36	$2.29\cdot10^{-5}$
0.33	$5.9\cdot10^{-3}$	$1.4\cdot10^{-3}$	0.50	$1.47\cdot10^{-4}$
0.25	$6.2\cdot10^{-3}$	$1.6\cdot10^{-3}$	0.60	$1.30\cdot10^{-4}$
0.20	$6.5\cdot10^{-3}$	$1.8\cdot10^{-3}$	0.67	$2.77\cdot10^{-5}$

b: LIDAR UPDATE RATE

The effect on performance of changing the rate at which pose estimates are provided to the UKF is now evaluated. To this aim, a set of 100 simulations is carried out considering four different values of f_L (i.e., 0.5 Hz, 0.33 Hz, 0.25 Hz and 0.20 Hz), reduced with respect to the reference case (i.e., 1 Hz). The results of a statistical analysis, done adopting the same procedure described in the previous sub-section, are collected in Table 7, and compared with the reference case. This analysis shows the robustness of the proposed relative navigation architecture against a significant reduction in the LIDAR update rate (down to 1 point cloud acquired and processed each 5 seconds). Considering that f_L can be interpreted as the rate at which the point clouds (acquired at higher frequency) are processed, this result is important since it allows relaxing the requirements regarding the computational efficiency characterizing the implementation of the pose determination algorithms. Clearly, these requirements can still be more stringent, i.e., pose measurements may be required even at higher frame rate when dealing with particularly aggressive relative dynamics, e.g., in the case of rapidly tumbling target.

c: LIDAR ANGULAR RESOLUTION

Finally, the effect on the achieved accuracy level obtained reducing the LIDAR angular resolution is analyzed. To this aim, δ_{LOS} is increased from 1° to 1.5°. As a consequence, the average number of detected points during one relative orbit reduces from 172 to 76, as also highlighted in Fig. 12.

Given the huge size of ENVISAT (it has a maximum frontal section of around 21 × 10 m^2), the point clouds provided in input to the pose determination algorithms are extremely sparse. Like in the previous cases, a set of 100 simulations is carried out setting δ_{LOS} to 1.5°. The results of a statistical analysis, done adopting the same procedure described in the previous sub-section, are collected in Table 8, and compared with the reference case ($\delta_{LOS} = 1°$).

Despite dealing with a significant sparseness in the raw LIDAR data processed by the pose determination algorithm, the level of accuracy achieved by the proposed relative navigation architecture does not change (only a slight worsening in each performance metric can be highlighted).

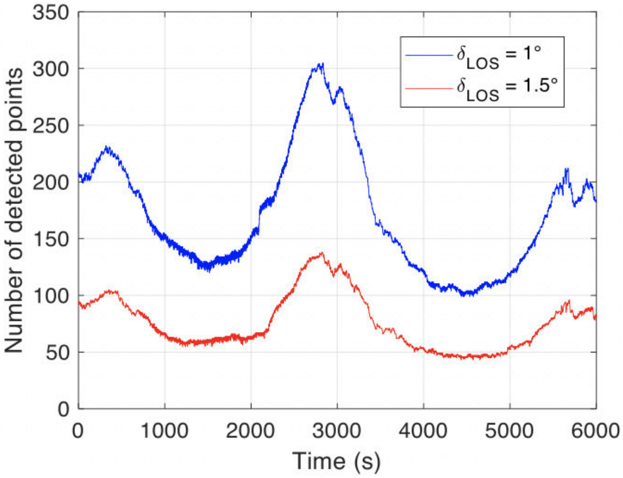

FIGURE 12. Time variation of the number of points detected by the LIDAR during one relative orbit for two different values of the LIDAR angular resolution. $\delta_{LOS} = 1°$ (blue curve). $\delta_{LOS} = 1.5°$ (red curve).

TABLE 8. Statistics of the error metrics defined for performance assessment of the developed relative navigation architecture computed over one relative orbits (averaged on 100 simulations). Effect of the LIDAR angular resolution. $f_L = 1$Hz, $\sigma_\rho = 2.5$ *cm*.

δ_{LOS}	$\lvert\rho\rvert_{ERR}$ (m)	$\lvert\dot{\rho}\rvert_{ERR}$ (m/s)	Φ_{ERR} (°)	$\lvert\omega\rvert_{ERR}$ (rad/s)
	rms	*rms*	*rms*	*rms*
1°	$5.6\cdot10^{-3}$	$1.2\cdot10^{-3}$	0.23	$2.32\cdot10^{-5}$
1.5°	$6.2\cdot10^{-3}$	$1.5\cdot10^{-3}$	0.28	$2.38\cdot10^{-5}$

V. CONCLUSION

In the frame of on-orbit servicing and active debris removal applications, an original architecture for relative navigation of a chaser spacecraft operating in close-proximity of uncooperative space targets was proposed. By combining an Unscented Kalman Filter (UKF) with model-based pose determination algorithms according to a loosely-coupled configuration, the architecture was designed to provide accurate estimates of the target-chaser relative rotational and orbital state using a LIDAR as relative navigation sensor. In particular, the pose determination algorithms are used not only for the relative state initialization phase (no prior information about the target-chaser relative state), but also to provide measurements about target-chaser relative position and attitude, as well as about the target absolute angular velocity (at lower rate than the one at which LIDAR point clouds are acquired), required for the update stage of the UKF. For the sake of improving reliability against unfavorable observation conditions of the target in the LIDAR Field-of-View, a processing strategy was also introduced to autonomously detect poorly accurate pose estimates, during the initialization or filtering phases, and, if necessary, to command the re-start of the initialization process. To test the robustness and the accuracy of the proposed approach, an extensive campaign of numerical simulations was carried out within a virtual environment, where the relative dynamics between two spacecraft flying in close-proximity and the operation of a scanning time-of-flight LIDAR were realistically reproduced.

The simulated test case was set considering a huge debris in Low Earth Orbit (i.e., ENVISAT) as target, around which a safe trajectory for monitoring purposes was designed.

Considering noise and operational parameters relative to spaceborne LIDARs, results showed capability of the proposed method to achieve satisfying level of accuracy in the estimate of the relative motion parameters. A sensitivity analysis was also performed through additional simulations characterized by increased noise level in the LIDAR range measurements, reduced update rate of the available pose estimates, and reduced LIDAR angular resolution. These additional results demonstrated robustness of the relative navigation architecture in terms of the achieved relative state accuracy, as the root mean square values associated to the selected error metrics suffered only a very slight worsening (without changes in the order of magnitude), despite dealing which highly noisy or sparse point clouds, as well as low-frequency pose estimates. This latter aspect is particularly important as it may allow relaxing computational time requirements associated to the implementation of the pose determination algorithms. Future work will be addressed to further test proposed approach performance considering different targets and different close-proximity maneuvers (like the final approach phase of a rendezvous). Another interesting point for future research is relative to the development of strategies to deal with partially known targets, for which LIDAR measurements can also be used for shape and/or inertia parameters reconstruction.

REFERENCES

[1] H. Klinkrad, *Space Debris*. Berlin, Germany: Springer-Verlag, 2010.

[2] R. A. Williamson, "Assuring the sustainability of space activities," *Space Policy*, vol. 28, no. 3, pp. 154–160, 2012.

[3] C. Bonnal, J.-M. Ruault, and M.-C. Desjean, "Active debris removal: Recent progress and current trends," *Acta Astron.*, vol. 85, pp. 51–60, Apr./May 2013.

[4] D. Kessler, N. Johnson, J.-C. Liou, and M. Matney, "The Kessler syndrome: Implications to future space operations," *Adv. Astronaut. Sci.*, vol. 137, p. 2010, Jan. 2010.

[5] W.-J. Li, D. Y. Cheng, X. G. Liu, Y. B. Wang, W. H. Shi, Z. X. Tang, F. Gao, F. M. Zeng, H. Y. Chai, W. B. Luo, and Q. Cong, "On-orbit service (OOS) of spacecraft: A review of engineering developments," *Prog. Aerosp. Sci.*, vol. 108, pp. 32–120, Jul. 2019.

[6] B. L. Benedict, "Investing in satellite life extension-fleet planning options for spacecraft owner/operators," in *Proc. AIAA SPACE Conf. Expo.*, 2014, p. 4445.

[7] National Aeronautics and Space Administration, "NASA Technology Roadmaps TA 4: Robotics and Autonomous Systems," NASA Technol. Roadmaps, Washington, DC, USA, Tech. Rep., Jul. 2015, p. 188. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/2015_nasa_technology_roadmaps_ta_4_robotics_and_autonomous_systems_final.pdf

[8] W. Fehse, "Rendezvous with and capture/removal of non-cooperative bodies in orbit: The technical challenges," *J. Space Saf. Eng.*, vol. 1, no. 1, pp. 17–27, 2014.

[9] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations," *Prog. Aerosp. Sci.*, vol. 93, pp. 53–72, Aug. 2017.

[10] X. Clerc and I. Retatt, "Astrium vision on space debris removal," in *Proc. 63rd Int. Astron. Congr.*, 2012, pp. 2873–2885.

[11] G. S. Aglietti, B Taylor, S Fellowes, T Salmon, I Retat, A. Hall, T. Chabot, A. Pisseloup, C. Cox, A. Mafficini, and N. Vinkoff, "The active space debris removal mission RemoveDebris. Part 2: In orbit operations," *Acta Astronaut.*, to be published.

[12] K. Hassani and W. S. Lee, "Optimal tuning of linear quadratic regulators using quantum particle swarm optimization," in *Proc. Int. Conf. Control. Dyn. Syst. Robot.*, vol. 59, 2014, pp. 1–8.

[13] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Pose estimation for spacecraft relative navigation using model-based algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 431–447, Feb. 2017.

[14] L. Pasqualetto Cassinis, R. Fonod, and E. Gill, "Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft," *Prog. Aerosp. Sci.*, vol. 110, Oct. 2019, Art. no. 100548.

[15] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *J. Field Robot.*, vol. 30, pp. 803–831, Sep. 2013.

[16] V. Pesce, M. Lavagna, and R. Bevilacqua, "Stereovision-based pose and inertia estimation of unknown and uncooperative space objects," *Adv. Space Res.*, vol. 59, no. 1, pp. 236–251, Jan. 2017.

[17] R. Volpe, M. Sabatini, and G. B. Palmerini, "Pose and shape reconstruction of a noncooperative spacecraft using camera and range measurements," *Int. J. Aerosp. Eng.*, vol. 2017, Aug. 2017, Art. no. 4535316.

[18] X. Wang, Z. Wang, and Y. Zhang, "Stereovision-based relative states and inertia parameter estimation of noncooperative spacecraft," *Proc. Inst. Mech. Eng. G, J. Aerosp. Eng.*, vol. 233, no. 7, pp. 2489–2502, 2018.

[19] J. O. Woods and J. A. Christian, "LIDAR-based relative navigation with respect to non-cooperative objects," *Acta Astron.*, vol. 126, pp. 298–311, Sep./Oct. 2016.

[20] X. Hou, C. Ma, Z. Wang, and J. Yuan, "Adaptive pose and inertial parameters estimation of free-floating tumbling space objects using dual vector quaternions," *Adv. Mech. Eng.*, vol. 9, no. 10, pp. 1–17, 2017.

[21] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, and M. Grassi, "Autonomous relative navigation around uncooperative spacecraft based on a single camera," *Aerosp. Sci. Technol.*, vol. 84, pp. 1070–1080, Jan. 2019.

[22] F. Cavenago, P. Di Lizia, M. Massari, and A. Wittig, "On-board spacecraft relative pose estimation with high-order extended Kalman filter," *Acta Astronaut.*, vol. 158, pp. 55–67, May 2019.

[23] J. J. LaViola, "A comparison of unscented and extended Kalman filtering for estimating quaternion motion," in *Proc. Amer. Control Conf.*, vol. 3, Jun. 2003, pp. 2435–2440.

[24] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[25] P. Jasiobedzki, S. Se, T. Pan, M. Umasuthan, and M. Greenspan, "Autonomous satellite rendezvous and docking using LIDAR and model based vision," *Proc. SPIE*, vol. 5798, pp. 54–65, May 2005.

[26] S. Ruel, T. Luu, M. Anctil, and S. Gagnon, "Target localization from 3D data for on-orbit autonomous rendezvous & docking," in *Proc. IEEE Aerosp. Conf.*, Mar. 2008, pp. 1–11.

[27] R. Opromolla, G. Rufino, G. Fasano, and M. Grassi, "LIDAR-based autonomous pose determination for a large space debris," in *Proc. Int. Astron. Congr. (IAC)*, vol. 2, 2014, pp. 1512–1520.

[28] T. Boge, S. Tuttas, and U. Stilla, "Fast initial pose estimation of spacecraft from LiDAR point cloud data," in *Proc. 13th Symp. Adv. Space Technol. Robot. Autom. (ASTRA)*, Noordwijk, The Netherlands, May 2015, pp. 1–11.

[29] L. Liu, G. Zhao, and Y. Bo, "Point cloud based relative pose estimation of a satellite in close range," *Sensors*, vol. 16, no. 6, p. 824, 2016.

[30] P. J. Besl and D. N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[31] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Modeling*, 2001, pp. 145–152.

[32] S. M. Prakhya, L. Bingbing, Y. Rui, and W. Lin, "A closed-form estimate of 3D ICP covariance," in *Proc. 14th IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, 2015, pp. 526–529.

[33] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA, USA: MIT Press, 1974.

[34] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Proc. SPIE*, vol. 3068, pp. 182–193, Jul. 1997.

[35] P. Gurfil and K. V. Kholshevnikov, "Manifolds and metrics in the relative spacecraft motion problem," *J. Guid. Control Dyn.*, vol. 29, no. 4, pp. 1004–1010, 2006.

[36] J. M. Galante, J. Van Eepoel, C. D'Souza, and B. Patrick, "Fast Kalman filtering for relative spacecraft position and attitude estimation for the raven ISS hosted payload," *Adv. Astronaut. Sci.*, vol. 157, no. 1, pp. 179–196, 2016.

[37] F. L. Markley and F. H. Bauer, "Attitude representations for Kalman filtering," *Adv. Astronaut. Sci.*, vol. 109, pp. 133–151, Mar. 2003.

[38] M. D. Shuster, "A survey of attitude representations," *J. Astron. Sci.*, vol. 41, no. 4, pp. 439–517, Dec. 1993.

[39] S. Chakraverty, N. Mahato, P. Karunakar, and T. Rao, "Finite difference method," in *Advanced Numerical and Semi-Analytical Methods for Differential Equations*, 1st ed. Hoboken, NJ, USA: Wiley, 2019, doi: 10.1002/9781119423461.ch5.

[40] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*. Cambridge, U.K.: Cambridge Univ. Press 2003.

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

[42] G. Euliss, A. Christiansen, and R. Athale, "Analysis of laser-ranging technology for sense and avoid operation of unmanned aircraft systems: The tradeoff between resolution and power," *Proc. SPIE*, vol. 6962, Apr. 2008, Art. no. 696208.

[43] R. D. Richmond and S. C. Cain, *Direct-Detection LADAR Systems*. Bellingham, WA, USA: SPIE, 2010.

[44] E. A. Spatial, "Envisat-1," ESA, Paris, France, Tech. Rep., 1998. [Online]. Available: https://earth.esa.int/support-docs/pdf/mis_sys.pdf

[45] Y.-L. Desnos, C. Buck, J. Guijarro, J.-L. Suchail, R. Torres, and E. Attema, "ASAR—Envisat's advanced synthetic aperture radar," *ESA Bull.*, vol. 102, pp. 91–100, Jan. 2000.

[46] B. F. Lagaune, "Determination of the attitude state of space debris objects based on Satellite Laser Ranging, using Envisat as a test case," M.S. thesis, Dept. Astrodynamics Space Missions, TU Delft, Delft, The Netherlands, 2016.

[47] J. Christian and S. Cryan, "A survey of LIDAR technology and its use in spacecraft relative navigation," in *Proc. AIAA Guid., Navigat., Control (GNC) Conf.*, 2013, p. 4641.

[48] J. Sommer and I. Ahrns, "GNC for a rendezvous in space with an uncooperative target," in *Proc. 5th Int. Conf. Spacecraft Formation Flying Missions Technol.*, 2013, pp. 1–15.

[49] A. Rhodes, J. Christian, and T. Evans, "A concise guide to feature histograms with applications to LIDAR-based spacecraft relative navigation," *J. Astronaut. Sci.*, vol. 64, pp. 414–445, Jan. 2017.

ROBERTO OPROMOLLA received the M.S. degree *(summa cum laude)* and the Ph.D. degree (Hons.) in aerospace engineering from the University of Naples "Federico II," in 2012 and 2016, respectively. In 2015, he was a Visiting Researcher with the Autonomous Systems Group, Cranfield University, U.K. He is currently an Assistant Professor with the University of Naples "Federico II". He coauthored more than 30 scientific articles published by high impact factor journals and presented at international conferences organized by AIAA, IAA, and IEEE. In the space field, his research activity deals with autonomous relative navigation of multisatellite systems based on active and passive electro-optical sensors, spacecraft formation flying dynamics, and star tracker for satellite attitude determination. In the field of aeronautics, his research interests include autonomous navigation of UAVs based on LIDAR, visual and depth cameras, airborne visual detection, and tracking for sense and avoid and collaborative swarm applications. He has collaborated within national and international research programs with funding from both public institutions and private companies. Since 2018, he has been a member of the IAA Permanent Committee on Space Debris. He is also a Reviewer for top-rated journals in the field of aerospace engineering, and Guest Editor for a Special Issue of Sensors MDPI, entitled LIDAR for autonomous vehicles.

ALESSIA NOCERINO was born in Massa di Somma, Naples, Italy, in 1995. She received the B.S. and M.S. degrees in aerospace engineering from the University of Naples "Federico II," in 2016 and 2019, respectively, where she is currently pursuing the Ph.D. degree in industrial engineering. Her research interests include relative navigation technologies and algorithms for on-orbit servicing and active debris removal applications, spacecraft attitude determination and control, guidance, and control of a microsatellite for de-orbiting by means of aerodynamic drag.