doi: 10.82124/CEAS-GNC-2026-049 CEAS-GNC-2026-049

EuroGNC Navigation Guidance Control CEAS Madrid, Spain May 5th-7th uc3m Universidad Carlos III de Madrid AIAA 2026

Distributed Navigation Architecture for Close-Range Target Inspection

Matteo Capitanio PhD Student in Aerospace Engineering, Department of Aerospace Science and Technology (DAER), Politecnico di Milano, Milan, Italy. matteo.capitanio@polimi.it

Michele Maestrini Assistant Professor, Department of Aerospace Science and Technology (DAER), Politecnico di Milano, Milan, Italy. michele.maestrini@polimi.it

Mauro Massari Associate Professor, Department of Aerospace Science and Technology (DAER), Politecnico di Milano, Milan, Italy. mauro.massari@polimi.it

ABSTRACT

A significant fraction of Resident Space Objects, such as orbital debris or damaged satellites, are uncooperative and insufficiently known, thus requiring in-orbit inspection for applications ranging from debris mitigation to in-orbit servicing. In this context, Distributed Space Systems represent a promising solution, as they combine scalability, robustness to single-agent failures, availability of multi-view observations, and a more efficient use of time and resources. Specifically, this work presents a preliminary analysis of a distributed navigation architecture in a Low-Earth Orbit scenario, where two cooperative chasers, each equipped only with a monocular camera, collaborate to inspect a large uncooperative target. The primary chaser employs an Extended Kalman Filter in combination with its Multiplicative variant to estimate its relative position, velocity, orientation, and angular velocity with respect to the target object. However, due to the monocular camera's intrinsic limitations, the primary chaser alone would not achieve sufficiently accurate relative navigation in the close-range regime without the support of additional sensing. To address this, the secondary chaser, orbiting at a larger distance, provides complementary measurements which are incorporated in the filtering process: the primary inspector reconstructs the secondary's observations as a function of its current estimates, assuming that only the target's shape and inertial properties are known. Data exchange is enabled via Inter-Satellite Link technology, while differential carrier-phase measurements provide the relative position between the two chasers. Preliminary results show that this cooperative scheme effectively overcomes the limitations of monocular vision in close-range operations and enables the primary chaser to maintain accurate navigation during high-resolution inspection of the target.

Keywords: Distributed Space Systems, Relative Navigation, In-Orbit Inspection

Nomenclature

t	=	time	$\boldsymbol{a}$	=	attitude error
$\boldsymbol{r}$	=	position	$\boldsymbol{R}$	=	attitude direct cosine matrix
$\boldsymbol{v}$	=	velocity	$\boldsymbol{\omega}$	=	angular velocity
$\boldsymbol{q}$	=	attitude quaternion	$\boldsymbol{J}$	=	inertia matrix

$\boldsymbol{\tau}$	=	control torque	$\boldsymbol{x}$	=	state vector
$\boldsymbol{\Phi}$	=	state transition matrix	$\boldsymbol{u}$	=	control vector
n	=	mean motion	$\boldsymbol{z}$	=	measurement vector
$\boldsymbol{p}$	=	feature vector	$\boldsymbol{P}$	=	state covariance matrix
$\boldsymbol{K}$	=	camera's intrinsic matrix	$\boldsymbol{H}$	=	measurement Jacobian matrix
foc	=	focal length	$\boldsymbol{Q}$	=	process noise matrix
d	=	pixel dimension	$\mathcal{R}$	=	measurement noise matrix
u_c, v_c	=	central pixel coordinates	$\mathcal{K}$	=	Kalman gain matrix
m	=	number of visible features	$\boldsymbol{S}$	=	innovation matrix

1 Introduction and State of the Art

In recent years, In-Orbit Servicing (IOS) has emerged as a key enabler for the sustainable and cost-effective use of space assets. IOS encompasses a wide spectrum of operations, including satellite life extension through refueling or repair, upgrading of onboard systems, relocation to new orbits, and end-of-life disposal. These services not only reduce the burden of replacing malfunctional or aging spacecrafts but also contribute to mitigating the exponential growth of space debris, thereby ensuring the long-term safety and accessibility of the orbital environment. Moreover, IOS represents a cornerstone technology for future large-scale space infrastructures, supporting on-orbit assembly and manufacturing. Within this framework, in-orbit inspection plays a fundamental role as the precursor of servicing missions. Inspection involves the close-range characterization of a target object, providing essential information on its kinematics, geometry, and possible damages.
Pushing the boundaries further, Distributed Space Systems (DSS) are emerging as a promising paradigm for space science, exploration and IOS itself [1]. Beyond the intrinsic robustness and scalability, they can unlock a new range of servicing capabilities that would be infeasible or inefficient for a single monolithic spacecraft. For instance, multiple agents can collectively perform target detumbling by distributing the control effort and loads on each individual servicer, and conduct coordinated manipulation by allocating specialized roles to different agents. Even in the context of inspection, distributed solutions offer clear advantages. Multiple viewpoints allow for the simultaneous acquisition of complementary observations, potentially leading to faster and improved target reconstruction. Furthermore, cooperation can be exploited to improve the relative navigation performance of each agent within a more resilient and redundant estimation framework, enhancing fault tolerance and ensuring operations continuity even in the presence of sensor degradation or agent failure.

Several studies have explored relative navigation in DSS to enable coordinated operations. Giralo and D'Amico [2] present a Global Navigation Satellite System (GNSS) testbed for DSS, which demonstrates centimeter-level accuracy through Carrier-phase Differential GNSS (CDGNSS) processing in formation-flying scenarios. The testbed was validated against real data from the PRISMA mission [3], a pioneering formation-flying experiment that provided high-precision relative navigation. In particular, guidance and navigation were tested using a monocular camera and GPS receivers with ground-in-the-loop processing, and the Visual Based System short-range camera was used to determine the target's pose from tens of meters by tracking a set of LEDs located on target's panels [4]. In the same context, the CanX-4/5 dual CubeSat formation-flying mission demonstrated onboard CDGNSS processing in Low Earth Orbit (LEO) [5]. Using dual-frequency GPS and onboard double-difference carrier phase processing with an Extended Kalman Filter (EKF), the couple maintained sub-decimal relative positioning even during maneuvers, and relied on orbital propagation and state prediction to sustain navigation through temporary GPS outages. Capuano et al. [6] further expanded relative navigation options with a multi-sensor approach that utilizes carrier phase GNSS observations and monocular measurements to enable fast and robust autonomous relative pose estimation of cooperative spacecraft in degraded GNSS environments. The angles-only Absolute and Relative Trajectory Measurement System (ARTMS) [7] represents another significant ad-

vancement in distributed navigation, which will be demonstrated in the Starling Formation-Flying Optical Experiment (StarFOX), one of the four experiments of current NASA's Starling mission. Specifically, ARTMS integrates a sequential filter, which fuses local visual measurements with remote observations shared via Inter-Satellite Link (ISL). This approach enables distributed, cooperative navigation, where each observer estimates its own state as well as those of remote observers and targets. Building upon this concept, Kruger et al. [8] propose one of the most advanced GNC architectures, integrating both far- and close-range imagery, CDGNSS measurements with integer ambiguity resolution, and ISL into unified estimation and control pipelines. Although the algorithms are designed for general multi-observer, multi-target scenarios, the experimental validation focuses on a single chaser approaching cooperative targets. The authors also highlight plans for future development including more complex configurations with non-cooperative targets. As they state, such a distributed architecture is expected to improve relative navigation by increasing observability, enabling safe, autonomous proximity operations and rendezvous between multiple objects with maximum modularity and flexibility.

Alongside these architectural developments, the choice of the estimation algorithm plays a crucial role. Recent advancements in relative navigation have explored optimization-based approaches, such as Factor Graphs and incremental smoothing [9]. In a cooperative scenario, distributed Pose Graphs could theoretically be employed to fuse visual acquisitions from various chasers, offering enhanced accuracy by retaining past states and globally optimizing all the available measurements. However, as emphasized by Indelman et al. [9], these methods often impose a significant computational burden. Moreover, as highlighted by Williams et al. [10], smoothing algorithms can introduce latencies that are unsuitable for high-rate, real-time control loops. Conversely, sequential filters like the EKF provide immediate, low-latency state estimates with minimal memory footprint, making them the standard and most pragmatic choice for resource-constrained onboard processors in autonomous GNC applications.

Beyond the algorithmic choices, while previous efforts have primarily concentrated on relative navigation among multiple cooperative chasers or between a chaser and a cooperative target, the approach presented in this work provides a preliminary analysis of a distributed relative navigation architecture designed to enable close-range, high-resolution inspection of an uncooperative target object by leveraging multiple cooperative observers. Each chaser is equipped with a visible-wavelength monocular camera, while feature extraction and matching are assumed to be successfully executed. The primary chaser performs relative navigation by estimating its position, velocity, orientation, and angular velocity with respect to the target. However, monocular vision alone can be limited in close-range inspection due to the scarcity of trackable features. To mitigate this issue, the secondary chaser provides complementary observations that are integrated into the estimation process. The primary chaser reconstructs these measurements as a function of its own state estimates, thereby improving navigation accuracy. The scenario under consideration involves two chasers in LEO inspecting a common target with known shape and inertial properties, while real-time data sharing via ISL and precise relative positioning from CDGNSS enable the realization of the distributed navigation framework.

The presented paper is structured as follows. Section 2 provides a brief overview of the fundamental mathematical framework, including the relative translational and rotational dynamics, the Pinhole camera model, and the adopted estimation algorithms. Section 3 describes in detail the proposed methodology, outlining the computation of inertial orientations for target pointing, the simulation of the acquired measurements, and the design of the distributed navigation filter. Section 4 presents the reference scenario, discusses the simulation setup, and analyzes the results obtained under a standalone configuration before evaluating the nominal and degraded cooperative conditions. Finally, Section 5 summarizes the main findings and outlines potential directions for future developments.

2 Fundamentals

This section outlines a concise overview of the mathematical framework and algorithms utilized in developing the presented navigation architecture.

2.1 Translational and Rotational Relative Dynamics

Relative dynamics describe the motion of a spacecraft with respect to a target body, encompassing relative position, velocity, orientation and angular velocity. These dynamical laws are of paramount importance in proximity operations, formation-flying, and IOS, as they predict how relative position and attitude evolve over time under gravity, control forces and other external disturbances.

Regarding the translational motion, a variety of differential equations under the two-body assumption exist in literature. For sake of simplicity, the model considered for this early analysis is the one described by the linearized equations of relative motion with respect to a circular reference orbit, also known as Clohessy-Wiltshire (CW) equations [11]. Given the constant orbital angular velocity, or mean motion, n, these equations are:

$$\begin{cases} \ddot{x} - 2n\dot{y} - 3n^2 x = 0 \\ \ddot{y} + 2n\dot{x} = 0 \\ \ddot{z} + n^2 z = 0 \end{cases} \tag{1}$$

where coordinates $\{x, y, z\}$ refer to radial, tangential and out-of-plane components of chaser position centered on the target object, namely defined in the Local-Vertical-Local-Horizontal (LVLH) reference frame. It is relevant to highlight that the CW equations are homogeneous because no external disturbance or control acceleration is applied. Featuring a time-invariant property, this model allows for the analytical expression of chaser's position and velocity as function of time and the initial conditions only. The same closed-form solution can be described through the State Transition Matrix (STM) $\mathbf{\Phi}_{\mathrm{CW}}$, which essentially provides a mapping between the initial state of the system at time t_0 and its state at a later time t.

$$\mathbf{\Phi}_{\mathrm{CW}}(t_0, t) = \begin{bmatrix} 4 - 3c_\theta & 0 & 0 & s_\theta/n & 2(1 - c_\theta)/n & 0 \\ 6(s_\theta - \Delta\theta) & 1 & 0 & 2(c_\theta - 1)/n & (4s_\theta - 3\Delta\theta)/n & 0 \\ 0 & 0 & c_\theta & 0 & 0 & s_\theta/n \\ 3ns_\theta & 0 & 0 & c_\theta & 2s_\theta & 0 \\ 6n(c_\theta - 1) & 0 & 0 & -2s_\theta & 4c_\theta - 3 & 0 \\ 0 & 0 & -ns_\theta & 0 & 0 & c_\theta \end{bmatrix} \tag{2}$$

where plainly $s_\theta = \sin(\Delta\theta)$, $c_\theta = \cos(\Delta\theta)$, $\Delta\theta = n\Delta t$ and $\Delta t = t - t_0$.

For what concerns the rotational motion between two reference frames, the relative attitude and relative angular velocity of the chaser with respect to the target must be first defined. Given the absolute orientations of body frames with respect to the inertial frame, namely $\boldsymbol{q}_{CI}$ and $\boldsymbol{q}_{TI}$ using the quaternion representation, and the body frames angular velocities $\boldsymbol{\omega}_{\mathrm{C}}$ and $\boldsymbol{\omega}_{\mathrm{T}}$, the corresponding relative quantities are defined according to:

$$\boldsymbol{q}_{\mathrm{rel}} = \boldsymbol{q}_{CI} \otimes (\boldsymbol{q}_{TI})^{-1} \qquad \boldsymbol{\omega}_{\mathrm{rel}} = \boldsymbol{\omega}_C - \boldsymbol{R}_{\boldsymbol{CT}}\boldsymbol{\omega}_T \tag{3}$$

The time evolution of the absolute orientations are described by the quaternion kinematic equation, while angular velocities can be propagated according to the Euler's equations valid for a rigid body [12]. Consequently, the relative counterparts can be obtained by integrating over time the following differential equations, respectively Eq. (4) for relative attitude and Eq. (5) for relative angular velocity.

$$\dot{\boldsymbol{q}}_{\text{rel}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{\text{rel}})\boldsymbol{q}_{\text{rel}} \qquad \boldsymbol{\Omega}(\boldsymbol{\omega}_{\text{rel}}) = \begin{bmatrix} 0 & -\omega^x_{\text{rel}} & -\omega^y_{\text{rel}} & -\omega^z_{\text{rel}} \\ \omega^x_{\text{rel}} & 0 & \omega^z_{\text{rel}} & -\omega^y_{\text{rel}} \\ \omega^y_{\text{rel}} & -\omega^z_{\text{rel}} & 0 & \omega^x_{\text{rel}} \\ \omega^z_{\text{rel}} & \omega^y_{\text{rel}} & -\omega^x_{\text{rel}} & 0 \end{bmatrix} \tag{4}$$

$$\dot{\boldsymbol{\omega}}_{\text{rel}} = \boldsymbol{J}_c^{-1}\left(\boldsymbol{\tau}_c - \boldsymbol{\omega}_c \times (\boldsymbol{J}_c\boldsymbol{\omega}_c)\right) - \boldsymbol{\omega}_c \times \boldsymbol{\omega}_{\text{rel}} + \boldsymbol{R}_{CT}\left(\boldsymbol{J}_t^{-1}\left[(\boldsymbol{R}_{CT}^{\top}(\boldsymbol{\omega}_c - \boldsymbol{\omega}_{\text{rel}})) \times (\boldsymbol{J}_t\boldsymbol{R}_{CT}^{\top}(\boldsymbol{\omega}_c - \boldsymbol{\omega}_{\text{rel}}))\right]\right) \tag{5}$$

considering J_C and J_T being respectively the inertia matrices of chaser and target, $\boldsymbol{\omega}_C$ the chaser's angular velocity and $\boldsymbol{\tau}_C$ the chaser's control torque for target pointing. To conclude, matrix $\boldsymbol{R}_{CT}$ in Eqs. (3) and (5) corresponds to the rotation described by $\boldsymbol{q}_{\text{rel}}$, as described by the conversion between attitude parameterizations [13].
All these dynamical models and mathematical relationships are used to generate the reference scenario and perform the prediction step required in the estimation process.

2.2 Pinhole Camera Model

In the context of visual navigation, monocular sensors are commonly modeled according to the Pinhole camera model [14], which returns the mathematical relationship between the three-dimensional spatial coordinates of a point and its projection onto the two-dimensional image plane of an ideal camera. Indeed, the model does not account for blur and geometric distortion of coordinates that most of the cameras have. On the other hand, it relies on the geometry of similar triangles to perform the projection, with the transformation governed by the camera's intrinsic parameters, such as focal length and central pixel coordinates. Due to its effectiveness and simplicity in approximating real camera behavior, the Pinhole model serves as a fundamental tool in computer vision, pose estimation and augmented reality, thereby supporting the autonomous operations of satellites in complex environments.
Let the 3D coordinates of point $\boldsymbol{p}_{\text{cam}}$ in camera frame be $\{X, Y, Z\}$. The corresponding pixel coordinates $\boldsymbol{p}_{\text{pixel}}$ on the image plane can be respectively expressed as:

$$\boldsymbol{p}_{\text{pixel}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{K}\frac{\boldsymbol{p}_{\text{cam}}}{Z} \qquad \boldsymbol{K} = \begin{bmatrix} foc/d & 0 & u_c \\ 0 & foc/d & v_c \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The matrix $\boldsymbol{K}$ is called the intrinsic matrix of the camera, and allows for the linear transformation of the homogeneous 3D coordinates of the point into the 2D pixel coordinates. Moreover, foc represents the focal length of the camera, d is the dimension of each pixel constituting the sensor, and $\{u_c, v_c\}$ denote the pixel coordinates of central point of the image.
The presented model is employed to model both acquired and expected measurements, particularly the 2D pixel coordinates of the selected target's features, as will be later shown.

2.3 Efficient Perspective-n-Point Algorithm

In computer vision, the Perspective-n-Point (PnP) problem involves determining the position and orientation of a camera from 2D image points and their corresponding 3D world coordinates. Accurate pose estimation is crucial for applications such as visual navigation and 3D reconstruction, where precise alignment between the camera's perspective and the physical environment is critical. In literature, many traditional methods rely on iterative optimization and high order complexity ($O(m^2)$). Alternatively, the Efficient PnP (EPnP) algorithm, developed by Vincent Lepetit et al. [15], provides a non-iterative and computationally efficient solution to the PnP problem with $O(m)$ time complexity for $m \geq 4$, where m is the number of points. In particular, EPnP expresses 3D points as a weighted sum of four virtual control points, transforming the problem of pose estimation into determining the coordinates of these control

points in the camera reference frame. Given the known 3D world coordinates and their corresponding 2D image projections, the four control points are combined using homogeneous barycentric coordinates, which are straightforward to compute and uniquely defined for each configuration. This representation can be applied in the camera coordinate system, enabling the formulation of a linear system without the need for 2D projection normalization. This closed-form solution is robust to noise and suitable for real-time applications, with an optional Gauss-Newton refinement step that improves accuracy while maintaining linear complexity with respect to the number of points.
Essentially, given the 3D world coordinates $\boldsymbol{p}_{\text{world}}$ and their corresponding 2D projections $\boldsymbol{p}_{\text{pixel}}$ of at least four points, EPnP efficiently solves for the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ that describe how the camera is oriented and placed with respect to the world reference system. Considering the Pinhole projection model (6), which relates the image points $\boldsymbol{p}_{\text{pixel}}$ to the 3D coordinates $\boldsymbol{p}_{\text{cam}}$ expressed in camera frame, $\boldsymbol{R}$ and $\boldsymbol{t}$ are determined such that the following relation holds:

$$\boldsymbol{p}_{\text{cam}} = \boldsymbol{R}\boldsymbol{p}_{\text{world}} + \boldsymbol{t} \tag{7}$$

In this work, the EPnP algorithm is applied to simulate pose measurements of chasers' cameras with respect to the target object, under the assumption that its 3D model is known.

2.4 Sequential Estimation Techniques

Kalman filtering is a consolidated method for sequentially estimating state variables and reducing the associated uncertainty. Many variants exist in literature, each designed to handle specific types of mathematical models for dynamics and measurements or state representations. Among these, the Extended Kalman Filter (EKF) is widely used for managing non-linear systems, where state and measurement models are linearized around the current estimate. However, for systems involving attitude estimation, where a quaternion representation is commonly used, the conventional EKF formulation may lead to inconsistencies due to the normalization constraint imposed by this parametrization. To overcome this issue, the Multiplicative Extended Kalman Filter (MEKF) estimates the attitude error as a three-dimensional, unconstrained vector relating the true orientation to a unit reference quaternion. This formulation preserves the quaternion normalization and ensures consistent attitude updates within the recursive filtering process.
Both these algorithms constitute the basis for the developed distributed filter for relative navigation in target's proximity, and they are briefly described in the following paragraphs.

2.4.1 Extended Kalman Filter

A generic iteration from time step t_k to t_{k+1} is subdivided in state prediction, expected measurements forecasting, and state update by correcting the previous prediction with measurements acquired through the sensors. Particularly, state estimate mean $\hat{\boldsymbol{x}}_k$ is forward propagated until t_{k+1} according to the non-linear dynamical law:

$$\hat{\boldsymbol{x}}_{k+1}^{-} = \boldsymbol{f}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) \tag{8}$$

while its corresponding covariance $\boldsymbol{P}_k$ is propagated through the STM $\boldsymbol{\Phi}_k$ of the function $\boldsymbol{f}$ evaluated in $\hat{\boldsymbol{x}}_k$, as following:

$$\hat{\boldsymbol{P}}_{k+1}^{-} = \boldsymbol{\Phi}_k \boldsymbol{P}_k \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_{k+1} \qquad \boldsymbol{\Phi}_k = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{x}_k}\right|_{\hat{\boldsymbol{x}}_k} \tag{9}$$

with $\boldsymbol{u}_k$ being the control input at t_k, the matrix $\boldsymbol{Q}_{k+1}$ describes the process noise uncertainty, while the superscript "$-$" refers to the a priori estimate. The expected measurements are computed through the measurement model $\boldsymbol{h}$, and the residual is obtained as the difference between actual measurements $\boldsymbol{z}_{k+1}$ and the predicted ones.

Subsequently, the innovation matrix $\boldsymbol{S}_{k+1}$ is retrieved according to:

$$\boldsymbol{S}_{k+1} = \boldsymbol{H}_{k+1}\boldsymbol{P}^{-}_{k+1}\boldsymbol{H}^{\top}_{k+1} + \boldsymbol{\mathcal{R}}_{k+1} \qquad \boldsymbol{H}_{k+1} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_{k+1})}{\partial \boldsymbol{x}_{k+1}}\right|_{\hat{\boldsymbol{x}}^{-}_{k+1}} \tag{10}$$

The matrix $\boldsymbol{H}_{k+1}$ is the Jacobian of the measurement function with respect to the state, linking small state perturbations to the corresponding measurement variations, while $\boldsymbol{\mathcal{R}}_{k+1}$ denotes the measurement noise covariance matrix. The update step involves the computation of the Kalman gain $\boldsymbol{\mathcal{K}}_{k+1}$, defined as:

$$\boldsymbol{\mathcal{K}}_{k+1} = \boldsymbol{P}_{k+1}\boldsymbol{H}^{\top}_{k+1}\boldsymbol{S}^{-1}_{k+1} \tag{11}$$

Finally, the state estimate mean and covariance are updated according to the following equations, where the superscript "+" denotes the posterior estimate:

$$\hat{\boldsymbol{x}}^{+}_{k+1} = \hat{\boldsymbol{x}}^{-}_{k+1} + \boldsymbol{\mathcal{K}}_{k+1}\left[z_{k+1} - \boldsymbol{h}(\hat{\boldsymbol{x}}^{-}_{k+1})\right] \qquad \hat{\boldsymbol{P}}^{+}_{k+1} = (\boldsymbol{I} - \boldsymbol{\mathcal{K}}_{k+1}\boldsymbol{H}_{k+1})\boldsymbol{P}^{-}_{k+1} \tag{12}$$

The whole process is then repeated for the next time step by setting as new initial conditions the results of the last iteration, enabling sequential estimation.

2.4.2 Multiplicative Extended Kalman Filter

As previously mentioned, the MEKF is specifically tailored for systems involving attitude estimation, where quaternion representation is typically employed. The core concept of the algorithm is to estimate an unconstrained, three-component attitude error $\boldsymbol{a}$ while using a normalized four-component $\boldsymbol{q}_{\text{ref}}$ to provide a globally non-singular attitude representation. Specifically, the prediction step is arranged to preserve a zero attitude error, while the update through the Kalman gain assigns a finite value $\boldsymbol{a}^{+}_{k+1}$. Such information is then moved to the propagated reference quaternion through quaternion multiplication:

$$\hat{\boldsymbol{q}}^{+}_{\text{ref},k+1} = \delta\hat{\boldsymbol{q}}(\boldsymbol{a}^{+}_{k+1}) \otimes \boldsymbol{q}^{-}_{\text{ref},k+1} \tag{13}$$

The mathematical relationship between $\delta\boldsymbol{q}$ and $\boldsymbol{a}$ depends on the chosen attitude error representation, as specified by Markley [13]. To conclude, the attitude error estimate is reset to zero, preparing the filter for the next iteration.

3 Methodology

In the following section, the simulation setup for the presented preliminary analysis is described, with particular emphasis on the design of target pointing control law, the modeling of the acquired measurements, and the distributed navigation architecture developed for close-range target inspection. It is important to note that the first version of the presented filter is focused on one chaser, referred to as the primary chaser, or Ch.1, while the secondary observer, or Ch.2, continuously provides additional measurements from a different viewpoint. The overall scheme is shown in Fig. 1, in which the objective is indeed to refine the current state estimate, including primary chaser's relative position, velocity, orientation, and angular velocity with respect to the inspected object, whose shape and inertia properties are known.

3.1 Control Law for Target Pointing

As the focus of the presented analysis is on navigation performance, guidance and control laws are excluded from the loop, unlike in complete GNC implementations. For this reason, the chasers' orientations and control torques are precomputed from ground-truth trajectories and attitudes.
Specifically for the generic i-th chaser ($i = 1, 2$), its orientation with respect to the inertial reference

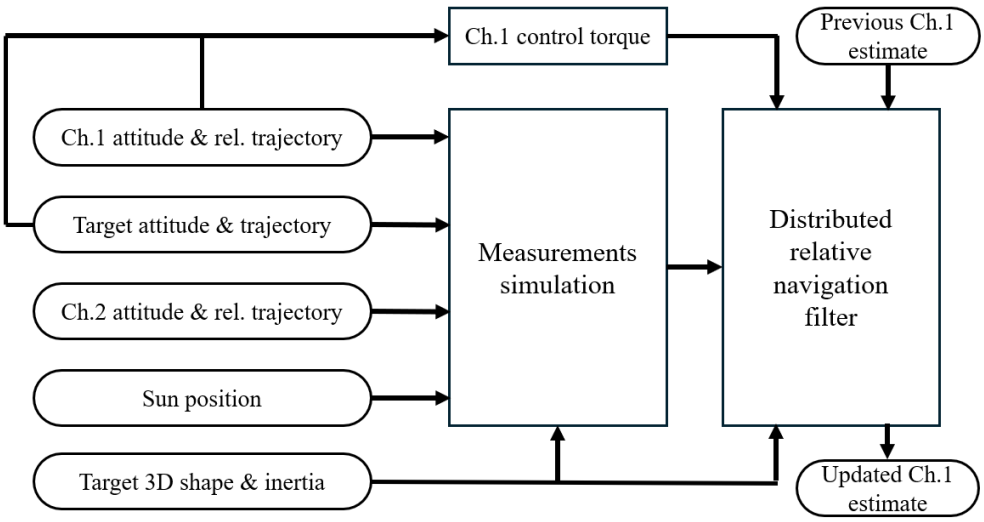

Fig. 1 Diagram representing the overall simulation scheme.

frame I is retrieved by aligning the camera axis towards the target. The quaternion $\boldsymbol{q}_{\mathrm{CiI}}$ corresponding to this rotation is computed as:

$$\boldsymbol{q}_{\mathrm{CiI,v}} = [0 \quad 0 \quad 1]^{\top} \times \boldsymbol{R}_{\mathrm{IL}} \left(-\frac{\boldsymbol{r}_{\mathrm{Ci}}^{\mathrm{rel}}}{||\boldsymbol{r}_{\mathrm{Ci}}^{\mathrm{rel}}||} \right) \qquad q_{\mathrm{CiI,0}} = 1 + [0 \quad 0 \quad 1]^{\top} \cdot \boldsymbol{R}_{\mathrm{IL}} \left(-\frac{\boldsymbol{r}_{\mathrm{Ci}}^{\mathrm{rel}}}{||\boldsymbol{r}_{\mathrm{Ci}}^{\mathrm{rel}}||} \right) \tag{14}$$

with vector $[0 \quad 0 \quad 1]^{\top}$ is the camera axis in body frame, $\boldsymbol{r}_{\mathrm{Ci}}^{\mathrm{rel}}$ is chaser's position relative to the target expressed in LVLH frame, and $\boldsymbol{R}_{\mathrm{IL}}$ is the rotation matrix required to pass from the LVLH frame to the inertial frame. This rotation is defined as the transpose matrix of $\boldsymbol{R}_{\mathrm{LI}}$ whose rows are respectively:

$$\boldsymbol{x}_{dir} = \frac{\boldsymbol{r}_{\mathrm{T,\,ECI}}}{||\boldsymbol{r}_{\mathrm{T,\,ECI}}||} \qquad \boldsymbol{y}_{dir} = \boldsymbol{z}_{dir} \times \boldsymbol{x}_{dir} \qquad \boldsymbol{z}_{dir} = \frac{\boldsymbol{r}_{\mathrm{T,\,ECI}} \times \boldsymbol{v}_{\mathrm{T,\,ECI}}}{||\boldsymbol{r}_{\mathrm{T,\,ECI}} \times \boldsymbol{v}_{\mathrm{T,\,ECI}}||} \tag{15}$$

where $\boldsymbol{r}_{\mathrm{T,\,ECI}}$ and $\boldsymbol{v}_{\mathrm{T,\,ECI}}$ are target's position and velocity expressed in Earth Centered Inertial (ECI) reference frame. The quaternion $\boldsymbol{q}_{\mathrm{CiI}}$ is formed by combining the vector component $\boldsymbol{q}_{\mathrm{CiI,v}}$ and the scalar component $q_{\mathrm{CiI,0}}$ into a 4-element vector, which is normalized to ensure it represents a valid rotation.
The resulting angular velocity and the control torque required to follow the obtained orientation can be retrieved as:

$$\boldsymbol{\omega}_{\mathrm{Ci}} = 2 \cdot \boldsymbol{q}_{\mathrm{CiI}}^{-1} \otimes \dot{\boldsymbol{q}}_{\mathrm{CiI}} \qquad \boldsymbol{\tau}_{\mathrm{Ci}} = \boldsymbol{J}_i \dot{\boldsymbol{\omega}}_{\mathrm{Ci}} + \boldsymbol{\omega}_{\mathrm{Ci}} \times (\boldsymbol{J}_i \boldsymbol{\omega}_{\mathrm{Ci}}) \tag{16}$$

The quaternion derivative $\dot{\boldsymbol{q}}_{\mathrm{CiI}}$ and $\dot{\boldsymbol{\omega}}_{\mathrm{Ci}}$ are approximated using a central finite difference scheme adopted for non-uniform time steps, while forward and backward three-point schemes are employed at the boundaries.

3.2 Simulated Measurements Modeling

Within the simulation framework, the synthetic measurements modeling plays a key role in enabling a quasi-realistic test of the navigation architecture presented in this work. Fig. 2 illustrates the logic behind the generation of simulated measurements: in particular, the acquired measurements correspond to those utilized by the filter to compute innovation and gain, while the so defined auxiliary measurements are data assumed available from external entities, i.e. other onboard sensors or ground stations.
More specifically, the 3D position of target's traceable features $\boldsymbol{p}_{\mathrm{T}}$ defined in the respective body frame T are converted and translated into camera frame through the following relation:

$$\boldsymbol{p}_{\mathrm{cam}} = \boldsymbol{R}_{CiT} \boldsymbol{p}_{\mathrm{T}} - \boldsymbol{R}_{CiI} \boldsymbol{R}_{IL} \boldsymbol{r}_{\mathrm{Ci}}^{\mathrm{rel}} \tag{17}$$

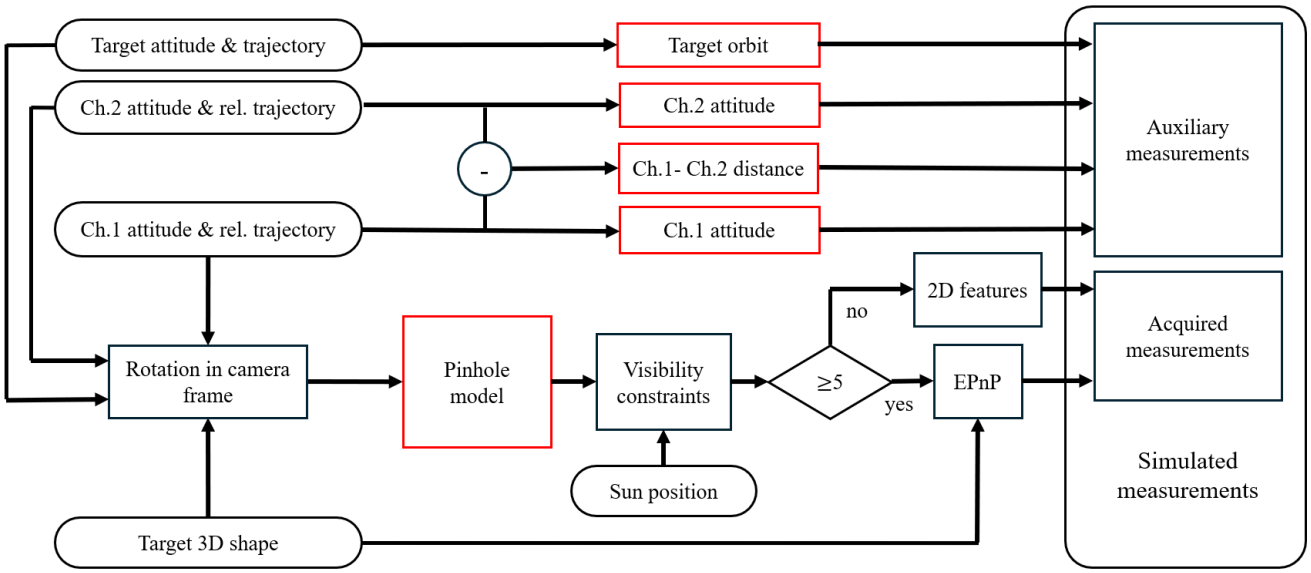

Fig. 2 Diagram of the synthetic measurements modeling. Measurement noise is indicated by red boxes.

where matrix $\boldsymbol{R}_{CiT}$ represents the relative rotation of the i-th chaser with respect to the target, and matrix $\boldsymbol{R}_{CiI}$ is the absolute rotation of $Ch.i$'s body frame with respect to the inertial frame. Subsequently, the Pinhole model described in Sec. 2.2 is applied to convert $\boldsymbol{p}_{\text{cam}}$ into the 2D pixel coordinates $\boldsymbol{p}_{\text{pix}}$.

To improve the realism of the simulated cameras, several visibility constraints are imposed on the 2D measurements. First, depending on the relative distance between chaser and target, a feature may fall outside the camera's Field Of View (FOV), thus becoming undetectable. Given the FOV dimensions, this constraint is easily implemented. Second, a geometric condition is enforced: a point is considered invisible if it lies behind the target relative to the chaser, as it would be occluded by the target's structure. Assuming a simplified target geometry, this condition is verified by checking the sign of the scalar product between the norm vector of each target face to which the feature belongs and the feature-to-camera direction vector. Finally, an illumination constraint is applied using a similar criterion: a face is illuminated only if the scalar product between the face normal vector and the Sun position, expressed in the target body frame, is positive. For both the geometric and illumination checks, features associated with multiple faces (e.g. edges or vertices) are considered visible if at least one of the corresponding faces satisfies the visibility condition; otherwise, they are discarded.

At this point, the EPnP algorithm briefly recalled in Sec. 2.3 is exploited to retrieve relative pose measurements position $\boldsymbol{t}_{\text{EPnP}}$ and rotation $\boldsymbol{R}_{\text{EPnP}}$ whenever the number of the remaining 2D measurements is larger than or equal to five. As stated before, the algorithm requires at least four correspondences to properly function. However, since solutions with four points yeld a higher position error, the threshold was increased by one. Otherwise, if fewer features are available, their 2D pixel coordinates are directly employed as acquired measurements.

In addition to the camera-derived acquisitions, the presented architecture also relies on a set of auxiliary measurements that provide complementary but necessary information. These include the target's orbit, chasers' inertial orientation, and their mutual position, all of them computed as function of the ground-truth trajectories and orientations. The reference target's orbit is assumed to be provided by a ground station after some orbit determination process, while the chasers' orientation with respect to the inertial frame are possibly given by attitude estimation filters and onboard attitude sensors. Finally, the secondary chaser's position with respect to the primary one is assumed to be retrieved by CDGNSS data processing.

To conclude the measurements modeling discussion, it is worth noting that predefined uncertainties are included to simulate realistic sensor acquisitions. Specifically, measurement noise is added to the 2D features after the Pinhole projection, as well as to all auxiliary measurements, as highlighted by the red boxes in Fig. 2.

3.3 Distributed Relative Navigation Filter

Building upon the previously defined control and synthetic measurement models, this section presents the distributed navigation architecture developed to estimate the relative state of the primary chaser with respect to the target. While Ch.1 runs the estimation process onboard, Ch.2 provides complementary visual or pose measurements from a different perspective, effectively increasing the spatial diversity of observations. The overall structure related to a generic iteration of the sequential filter is depicted in Fig. 3. As anticipated, this filter is a combination of the EKF with a MEKF, both deepened in Sec. 2.4, which integrate the system dynamics and acquired measurements for robust estimation.

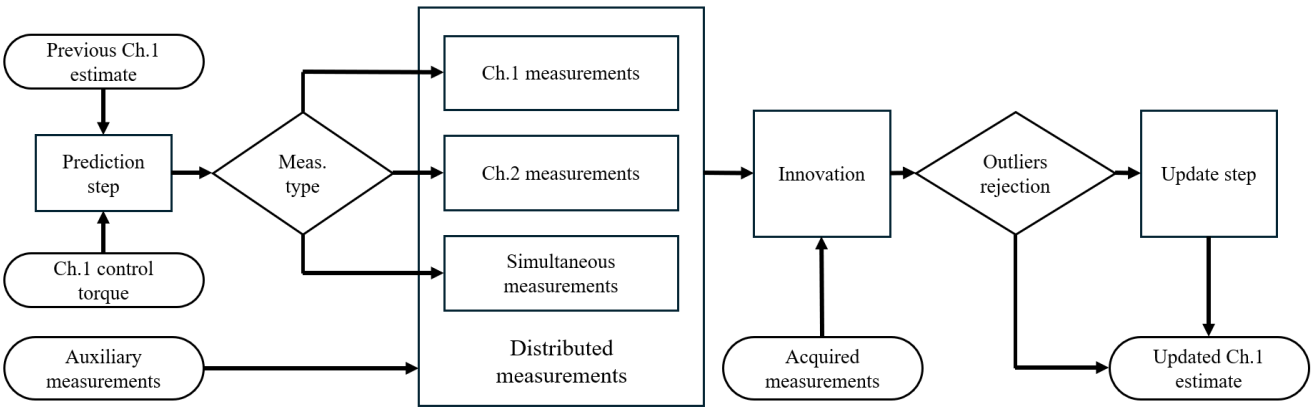

Fig. 3 Diagram of the distributed navigation filter architecture.

At current time t_k, the filter state comprises the Ch.1 relative position, velocity, orientation error and angular velocity:

$$\hat{\boldsymbol{x}}_k = \begin{bmatrix} \boldsymbol{r}_{\mathrm{C1}}^{\mathrm{rel}} & \boldsymbol{v}_{\mathrm{C1}}^{\mathrm{rel}} & \boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}} & \boldsymbol{\omega}_{\mathrm{C1}}^{\mathrm{rel}} \end{bmatrix}^\top \tag{18}$$

State propagation is performed according to the dynamical models presented in Sec. 2.1, embedding the primary chaser's control torque $\boldsymbol{\tau}_{\mathrm{C1}}$. Since the relative attitude error $\boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}}$ is reset to zero at the previous iteration, as required by the MEKF variant, the propagation of the relative angular velocity relies on the kinematics of the reference quaternion $\boldsymbol{q}_{\mathrm{ref},k}^{\mathrm{rel}}$ alone. However, although $\boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}}$ is instantaneously null, its dynamics affects the integration of the STM associated with the relative attitude motion, $\boldsymbol{\Phi}_{\mathrm{rot}}$. Specifically, the rate of change of the orientation error can be expressed as the cross product:

$$\dot{\boldsymbol{a}}_{\mathrm{C1}}^{\mathrm{rel}} = -\frac{1}{2}\boldsymbol{\omega}_{\mathrm{C1}}^{\mathrm{rel}} \times \boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}} \tag{19}$$

Furthermore, since the relative attitude dynamics depend on the absolute rotational motion of Ch.1, the STM is also influenced by $\boldsymbol{a}_{\mathrm{C1}}$ and $\boldsymbol{\omega}_{\mathrm{C1}}$. For this reason, an augmented STM $\boldsymbol{\Phi}_{\mathrm{aug}}$ is retrieved by integrating the variational equation ($\dot{\boldsymbol{\Phi}}_{\mathrm{aug}} = \boldsymbol{A}\boldsymbol{\Phi}_{\mathrm{aug}}$), where the Jacobian $\boldsymbol{A}$ is a function of both the absolute and relative attitude and angular velocity dynamics. The complete expression of $\boldsymbol{A}$ is given as:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}}{\partial \boldsymbol{a}_{\mathrm{C1}}} & \frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}} & \cancel{\frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}}{\partial \boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}}}} & \cancel{\frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}^{\mathrm{rel}}}} \\ \cancel{\frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}}{\partial \boldsymbol{a}_{\mathrm{C1}}}} & \frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}} & \cancel{\frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}}{\partial \boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}}}} & \cancel{\frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}^{\mathrm{rel}}}} \\ \cancel{\frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{a}_{\mathrm{C1}}}} & \cancel{\frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}}} & \frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}}} & \frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}^{\mathrm{rel}}} \\ \cancel{\frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{a}_{\mathrm{C1}}}} & \frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}} & \frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}}} & \frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}^{\mathrm{rel}}} \end{bmatrix} \quad \text{where} \quad \begin{aligned} \frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}}{\partial \boldsymbol{a}_{\mathrm{C1}}} &= -\frac{1}{2}(\boldsymbol{\omega}_{\mathrm{C1}})^\wedge \\ \frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}} &= \frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}^{\mathrm{rel}}} = \boldsymbol{I}_{3\times 3} \\ \frac{\partial \dot{\boldsymbol{\omega}}_{\mathrm{C1}}}{\partial \boldsymbol{\omega}_{\mathrm{C1}}} &= -\boldsymbol{J}_{\mathrm{C1}}^{-1}\left[(\boldsymbol{\omega}_{\mathrm{C1}})^\wedge \boldsymbol{J}_{\mathrm{C1}} - (\boldsymbol{J}_{\mathrm{C1}}\boldsymbol{\omega}_{\mathrm{C1}})^\wedge\right] \\ \frac{\partial \dot{\boldsymbol{a}}_{\mathrm{C1}}^{\mathrm{rel}}}{\partial \boldsymbol{a}_{\mathrm{C1}}^{\mathrm{rel}}} &= -\frac{1}{2}(\boldsymbol{\omega}_{\mathrm{C1}}^{\mathrm{rel}})^\wedge \end{aligned} \tag{20}$$

$$\frac{\partial \dot{\omega}_{C1}^{rel}}{\partial \omega_{C1}} = \frac{\partial \dot{\omega}_{C1}}{\partial \omega_{C1}} + \left(\omega_{C1}^{rel}\right)^{\wedge} + \boldsymbol{R}_{CT}\boldsymbol{J}_T^{-1}\left[-\left(\boldsymbol{J}_T\boldsymbol{R}_{CT}^{\top}(\omega_{C1}-\omega_{C1}^{rel})\right)^{\wedge} + \left(\boldsymbol{R}_{CT}^{\top}(\omega_{C1}-\omega_{C1}^{rel})\right)^{\wedge}\boldsymbol{J}_T\right]\boldsymbol{R}_{CT}^{\top}$$

$$\frac{\partial \dot{\omega}_{C1}^{rel}}{\partial \omega_{C1}^{rel}} = -(\omega_{C1})^{\wedge} - \boldsymbol{R}_{CT}\boldsymbol{J}_T^{-1}\left[-\left(\boldsymbol{J}_T\boldsymbol{R}_{CT}^{\top}(\omega_{C1}-\omega_{C1}^{rel})\right)^{\wedge} + \left(\boldsymbol{R}_{CT}^{\top}(\omega_{C1}-\omega_{C1}^{rel})\right)^{\wedge}\boldsymbol{J}_T\right]\boldsymbol{R}_{CT}^{\top}$$

The last term $\partial\dot{\omega}_{C1}^{rel}/\partial\boldsymbol{a}_{C1}^{rel}$, along with all subsequent complex derivatives and Jacobians in this work, has been derived using the ®MATLAB Symbolic Math Toolbox to ensure mathematical consistency. The matrix $\boldsymbol{I}$ is the identity matrix while the superscript "∧" denotes the skew-symmetric matrix operator. Finally, the useful STM $\boldsymbol{\Phi}_{rot}$ is extrapolated from $\boldsymbol{\Phi}_{aug}$, and the complete STM $\boldsymbol{\Phi}_k$ is retrieved as:

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{\Phi}_{CW} & \boldsymbol{0}_{6\times6} \\ \boldsymbol{0}_{6\times6} & \boldsymbol{\Phi}_{rot} \end{bmatrix} \tag{21}$$

with matrix $\boldsymbol{\Phi}_{CW}$ given by Eq. (2). At this point, the covariance matrix $\boldsymbol{P}_k$ associated with the state estimate mean can be propagated until t_{k+1}, concluding the prediction step.
Subsequently, the algorithm models the expected measurements and associated measurement Jacobians based on those available at time t_{k+1}, which may be acquired from Ch.1, Ch.2 or simultaneously from both. Moreover, the same logic employed in the simulation of acquired measurements is applied to decide whether to use pose measurements or directly the 2D coordinates of the detected features. With the measurements acquired by the generic i-th Ch.i, the expected measurement vector and the corresponding Jacobian are composed as follows.

- If pose measurements are available through EPnP method:

$$\hat{\boldsymbol{z}}_{k+1} = \begin{bmatrix} -\hat{\boldsymbol{R}}_{CiI}\hat{\boldsymbol{R}}_{IL}\hat{\boldsymbol{r}}_{Ci}^{rel} & \hat{\boldsymbol{a}}_{Ci}^{rel} \end{bmatrix}^{\top} \qquad \boldsymbol{H}_{k+1} = \begin{bmatrix} -\hat{\boldsymbol{R}}_{CiI}\hat{\boldsymbol{R}}_{IL} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \dfrac{\partial \boldsymbol{a}_{Ci}^{rel}}{\partial \boldsymbol{a}_{C1}^{rel}} & \boldsymbol{0}_{3\times3} \end{bmatrix} \tag{22}$$

  The derivative $\partial\boldsymbol{a}_{Ci}^{rel}/\partial\boldsymbol{a}_{C1}^{rel}$ results in the identity matrix for primary measurements, while it is symbolically retrieved for secondary measurements. The obtained rotation matrix $\boldsymbol{R}_{EPnP}$ is converted into the corresponding quaternion $\boldsymbol{q}_{EPnP}$, which is then used to compute the true relative orientation error with respect to the reference attitude described with $\boldsymbol{q}_{ref,k+1}^{rel}$. Consequently, its expected counterpart $\hat{\boldsymbol{a}}_{Ci}^{rel}$ is a null vector.
- If 2D features are directly used, because less than five features are detected:

$$\hat{\boldsymbol{z}}_{k+1} = \boldsymbol{K}\frac{1}{Z}\left(\hat{\boldsymbol{R}}_{CiT}\boldsymbol{p}_T - \hat{\boldsymbol{R}}_{CiI}\hat{\boldsymbol{R}}_{IL}\hat{\boldsymbol{r}}_{Ci}^{rel}\right)$$

$$\boldsymbol{H}_{k+1} = \begin{bmatrix} -\boldsymbol{B}\hat{\boldsymbol{R}}_{CiI}\hat{\boldsymbol{R}}_{IL} & \boldsymbol{0}_{2\times3} & \boldsymbol{B}\dfrac{\partial \boldsymbol{p}_{cam}}{\partial \boldsymbol{a}_{Ci}^{rel}} & \boldsymbol{0}_{2\times3} \end{bmatrix} \quad \text{with} \quad \boldsymbol{B} = \begin{bmatrix} \dfrac{foc}{d\,Z} & 0 & -\dfrac{foc\,X}{d\,Z^2} \\ 0 & \dfrac{foc}{d\,Z} & -\dfrac{foc\,Y}{d\,Z^2} \end{bmatrix} \tag{23}$$

  The derivative $\partial\boldsymbol{p}_{cam}/\partial\boldsymbol{a}_{Ci}^{rel}$ is symbolically computed, taking into account the relation between the attitude error and the reference quaternion expressed by Eq. (13). In this case, it is relevant to remark that $\hat{\boldsymbol{z}}_{k+1}$ has dimension $2m$ while the Jacobian $\boldsymbol{H}_{k+1}$ is a $[2m \times 12]$, with m being the number of modeled features that have correspondence with the actual detected ones. The visibility constraints are applied to the simulated features as explained in Sec. 3.2, while the modeled ones are selected accordingly, without enforcing these constraints.

In the above expressions, matrices $\hat{\boldsymbol{R}}_{CiI}$ and $\hat{\boldsymbol{R}}_{IL}$ are computed as functions of the auxiliary measurements, i.e. target's orbit and chasers' inertial orientations respectively. Furthermore, in the specific case of secondary measurements, the algorithm computes Ch.2's relative position and orientation with respect to the target by employing Eqs. (24) and (25), linking them to Ch.1 relative state.

$$\hat{\boldsymbol{r}}_{C2}^{rel} = \hat{\boldsymbol{r}}_{C1}^{rel} + \hat{\boldsymbol{d}}_{C2C1} \tag{24}$$

$$\hat{\boldsymbol{q}}^{\text{rel}}_{\text{C2T}} = \hat{\boldsymbol{q}}_{\text{C2I}} \otimes (\hat{\boldsymbol{q}}_{\text{C1I}})^{-1} \otimes \hat{\boldsymbol{q}}^{\text{rel}}_{\text{C1T}} \tag{25}$$

where $\hat{\boldsymbol{d}}_{\text{C2C1}}$ is the auxiliary relative position between the two observers, $\hat{\boldsymbol{q}}_{\text{C1I}}$ and $\hat{\boldsymbol{q}}_{\text{C2I}}$ are their auxiliary inertial attitudes, and $\hat{\boldsymbol{q}}^{\text{rel}}_{\text{C1T}}$ corresponds to the propagated reference quaternion $\boldsymbol{q}^{\text{rel}}_{\text{ref},k+1}$.
To include the uncertainty associated with the auxiliary measurements, for which a constant uncertainty is assumed over time, the Consider Covariance Analysis [16] approach is exploited. In particular, the covariance matrix of each of the g auxiliary measurements is projected into the measurement space and added to the main contributors in the computation of the Innovation matrix $\boldsymbol{S}_{k+1}$, extending Eq. 10 to:

$$\boldsymbol{S}_{k+1} = \boldsymbol{H}_{k+1}\boldsymbol{P}^{-}_{k+1}\boldsymbol{H}^{\top}_{k+1} + \boldsymbol{\mathcal{R}}_{k+1} + \sum_{j=1}^{g} \boldsymbol{H}_j \boldsymbol{P}_j \boldsymbol{H}_j^{\top} \tag{26}$$

where $\boldsymbol{P}_j$ denotes the covariance matrix of the j-th auxiliary measurement, and $\boldsymbol{H}_j$ is its corresponding measurement Jacobian. While the auxiliary covariance matrices are assumed constant over time, the associated Jacobians are retrieved by differentiating the measurement vector with respect to each auxiliary variable. These analytical expressions are likewise generated via the ®MATLAB Symbolic Math Tool. Ultimately, this approach allows the filter to account for auxiliary measurement uncertainties without explicitly augmenting the state vector. Moreover, when 2D features are employed, the measurement noise matrix $\boldsymbol{\mathcal{R}}_{k+1}$ is defined as a function of the camera noise. In the case of EPnP-based measurements, instead, this matrix is obtained as the covariance of the simulated EPnP outputs, which are themselves generated from noisy 2D feature acquisitions.
Next, the outlier-rejection process is carried out based on the Chi-Squared χ^2 test [17]. At this aim, the following Square Mahalanobis Distance (SMD) is employed as test statistics:

$$\text{SMD} = [\boldsymbol{h}(\boldsymbol{x}_{k+1}) - \hat{\boldsymbol{z}}_{k+1}]^{\top} \boldsymbol{S}^{-1}_{k+1} [\boldsymbol{h}(\boldsymbol{x}_{k+1}) - \hat{\boldsymbol{z}}_{k+1}] \tag{27}$$

This passage is of paramount importance in order to prevent inconsistency between the expected and the true simulated measurements. Whereas this test has a positive outcome, i.e. SMD $< \chi^2_{lim}$, the filter proceeds with the update step as defined in Eqs. 11 and 12, while the reference quaternion is corrected according to Eq. 13. Conversely, if the measurements are rejected, the update step is skipped, and the current filter state and covariance remain equal to the predicted ones. After the appropriate reset of variables, the current iteration ends and the process continues with the next set of acquired measurements.

4 Preliminary Analysis and Results

This section outlines the reference scenario adopted for the analysis and presents preliminary yet significant results. The scenario covers the characteristics of the chasers and target object, including their reference trajectories and attitude profiles, and monocular camera specifications. The initial and assumed uncertainties for both the filter estimates and the auxiliary measurements are also detailed. Subsequently, the standalone primary chaser navigation is assessed to demonstrate the limitations of single-agent operations in close proximity. To conclude, the proposed distributed architecture is evaluated under nominal and degraded operating conditions, highlighting its robustness and potential benefits.

4.1 Description of the Reference Scenario

Given the preliminary nature of this study, the developed architecture relies on simplifying assumptions intended to investigate the presented approach. In particular, the two chasers and the target objects are modeled as simple cuboids, whose dimensions and masses are reported in Table 1. Furthermore, the features correspond to the vertices of the target geometry, while feature detection and matching are assumed to be ideal. Although this simplification neglects outliers and false matches typical of visual imagery, it allows for isolating and assessing the performance of the estimation logic.

The target object orbits in a circular equatorial LEO with a radius of 8000 km, starting from a random initial orientation. Its initial angular velocity is randomly generated and has a magnitude of a few hundredths of a degree per second. The primary chaser is positioned close to the target object to simulate the close-range inspection mission, maintaining a minimum relative distance of approximately 15 m, while the secondary observer orbits at a greater distance. Both the chasers trajectories are assumed to be ideal football orbits, as depicted in Fig. 4, while their inertial attitudes are defined according to Sec. 3.1. Both the chasers are equipped with the same camera, whose characteristics are defined in Table 2.

Table 1 Chasers and target physical properties.

	Ch.1	**Ch.**2	**Target**
Length (m)	0.2	0.2	10
Depth (m)	0.3	0.3	3
Height (m)	0.8	0.2	5
Mass (kg)	50	10	8000

Table 2 Optical cameras characteristics.

Parameter	**Value**
Sensor size	2048×2048
Focal length, foc	100 mm
Pixel size, d	5.5 μm
FOV	$20.59° \times 20.59°$

Minimal process noise is considered in the filtering operation, since reality is simulated with the same dynamical models utilized for predicted estimates. Table 3 shows the selected values for camera noise, auxiliary measurements noise and initial estimate uncertainties. It is worth noting that the attitude uncertainties are expressed in the three-dimensional attitude error vector. The two observers acquire images at different frequencies, enabling the filter to process primary and secondary measurements either independently or simultaneously. Chaser 1 acquires images at a frequency of 0.1 Hz, whereas Chaser 2 operates at 0.07 Hz.

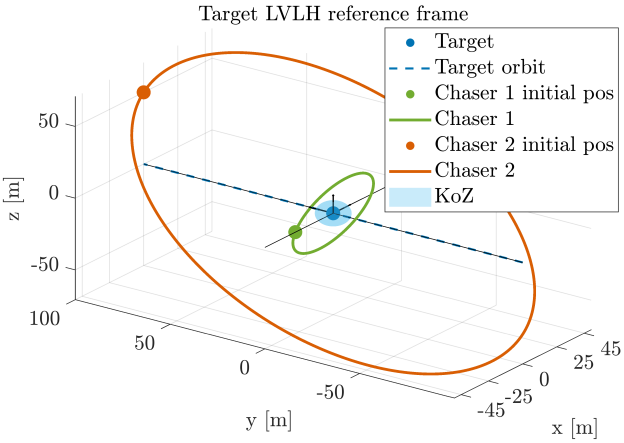

Fig. 4 Chasers' relative orbits.

Fig. 5 illustrates the distribution of the target's features as observed from the two chasers' viewpoints during the inspection phase. Specifically, the colored points correspond to the features projected onto each image plane over time. The black rectangle denote the FOV, while the black squares enclose non-visible features, either because they fall outside the FOV, are not illuminated, or are occluded by the target geometry. This visualization emphasizes the different viewpoints of the two chasers: Ch.1, operating at a closer range, captures a more detailed portion of the target surface but detects fewer features, which may cause its standalone navigation to fail.

Table 3 Initial uncertainties and measurements noise.

Variable	**Uncertainty** (3σ)
Camera noise	5 pix
Target orbit in ECI	100 m, 0.5 m/s
Ch.1 & Ch.2 inertial attitude	1e-3
Ch.1 & Ch.2 relative position	0.10 m
Ch.1 relative pos. & vel. at t_0	1 m, 0.01 m/s
Ch.1 relative att. & ang. vel. at t_0	1e-2, 0.057 deg/s

In further support of this, Fig. 6 depicts the number of visible features tracked by the two chasers. Notably, as Ch.1 reaches its minimum distance from the target, which occurs twice per orbit, the number of detectable features drastically drops to zero. On the other hand, Ch.2 observes a larger number of features throughout the entire orbit, thereby supporting the primary chaser's relative navigation. Even with a dual-viewpoint, at least one feature is unobservable due to the illumination constraint.

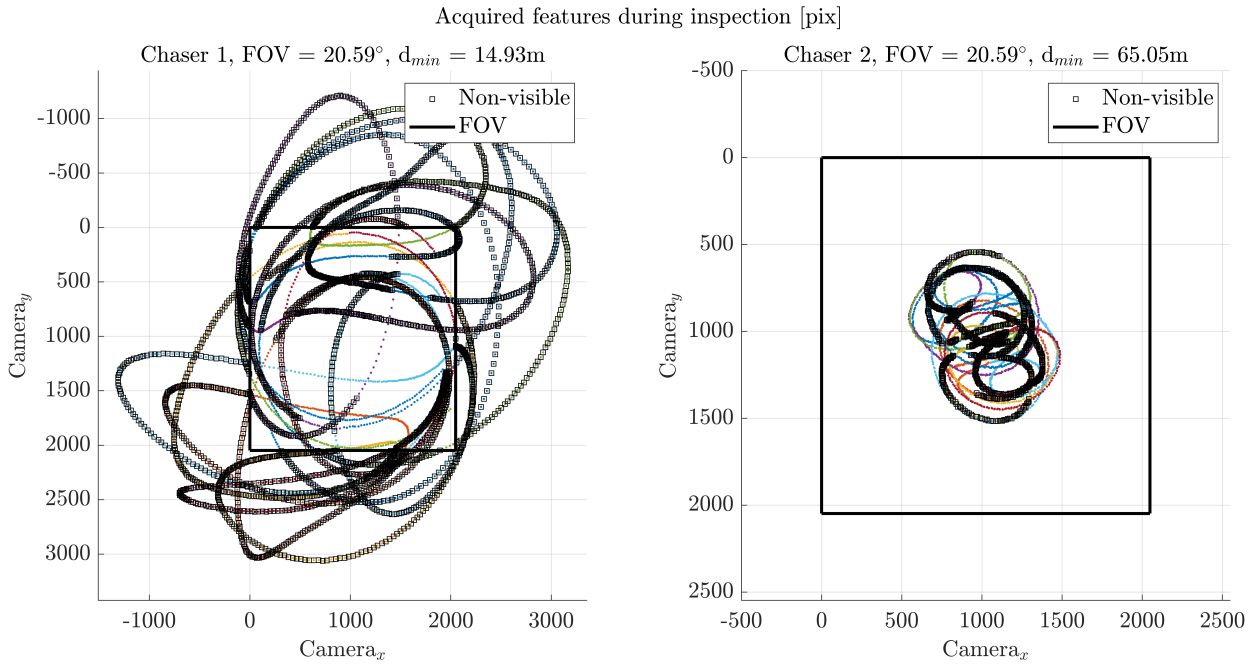

Fig. 5 Acquired features from Ch.1 and Ch.2 perspectives. Colored points denote visible features.

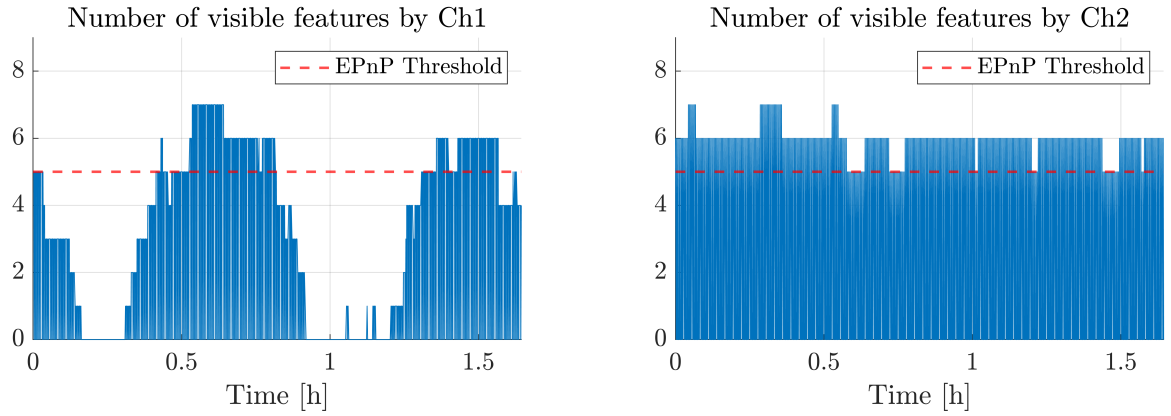

Fig. 6 Number of acquired features over time by the two chasers.

4.2 Standalone Primary Chaser Navigation

Before assessing the distributed architecture, a baseline simulation featuring the primary chaser operating in standalone mode is conducted. As anticipated by the visibility analysis in Fig. 6, the close-range trajectory forces the primary chaser into periods of partial or total feature loss.

To assess the primary chaser's standalone capabilities, a Monte Carlo (MC) simulation is conducted over one target orbit, randomly varying the initial estimates, target's orientation and angular rate, and Sun position on the Ecliptic plane at t_0. While the cooperative scenarios analyzed in later sections employ 500 realizations to establish robust statistical bounds, this baseline evaluation is deliberately limited to 20 runs. This reduced sample size is statistically sufficient to demonstrate the systematic divergence of the standalone approach. Fig. 7 illustrates the resulting navigation performance without the support of the secondary observer: as foreseeable, the scarcity of visual features causes the filter to diverge rapidly, with position errors quickly exceeding tens of meters. By the time a sufficient number of features become visible again, the accumulated estimation error is too large for the filter to recover and converge. This outcome confirms that a standalone monocular approach, constrained by the limited FOV of the selected optical sensor, is inadequate for this specific inspection profile, thereby providing a strong justification for the proposed solution.

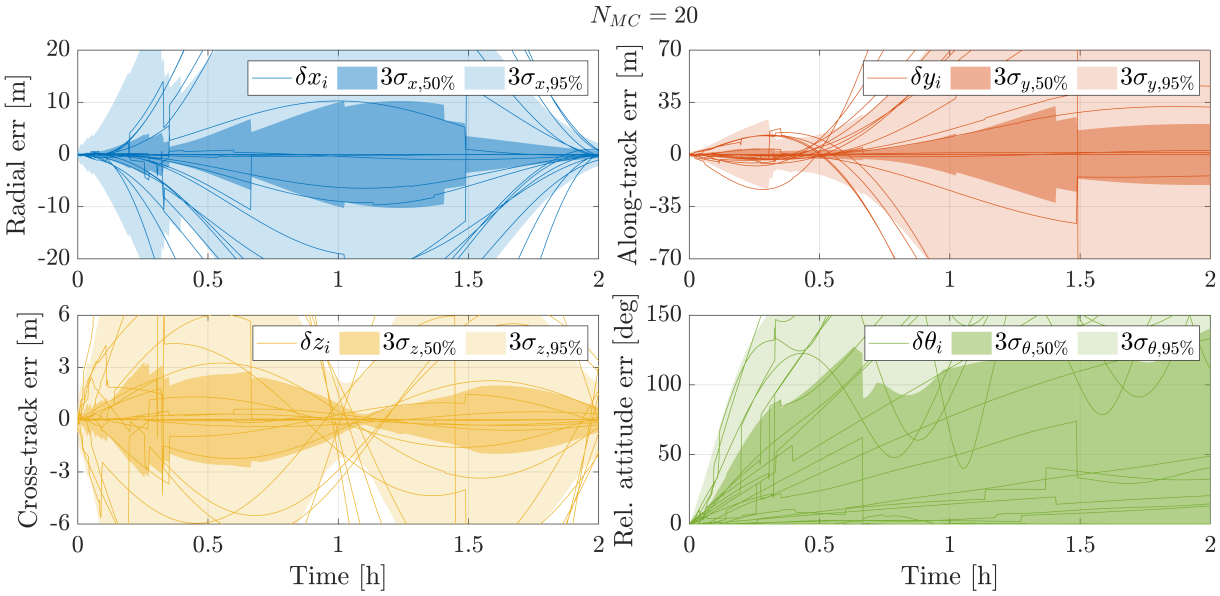

Fig. 7 MC realizations of relative position and attitude (Euler angle) errors in standalone condition.

4.3 Results in Nominal Conditions

During nominal operations, the scenario corresponds to that described in Sec 4.1. Ch.1 performs relative navigation with respect to the target object by fusing its own observations with those acquired by the secondary chaser, effectively exploiting the suggested distributed architecture.
Building upon the baseline analysis, the same Monte Carlo framework is applied to evaluate this cooperative setup, this time utilizing the full set of 500 samples. Unlike the divergent behavior observed in the standalone case, the continuous integration of secondary measurements ensures stable and accurate estimation. Fig. 8 illustrates the time evolution of Ch.1 relative position and attitude estimated errors, while Tables 4 and 5 summarize the overall analysis results. In particular, both average and final Root Mean Square Errors (RMSEs) of each estimated variable are reported together with the corresponding median and 95-th percentile of the 3σ realizations. While the median represents the typical 3σ uncertainty over the considered time frame, the 95-th percentile quantifies the robustness against worst-case realizations. All state components are correctly estimated, achieving uncertainty levels comparable with the current

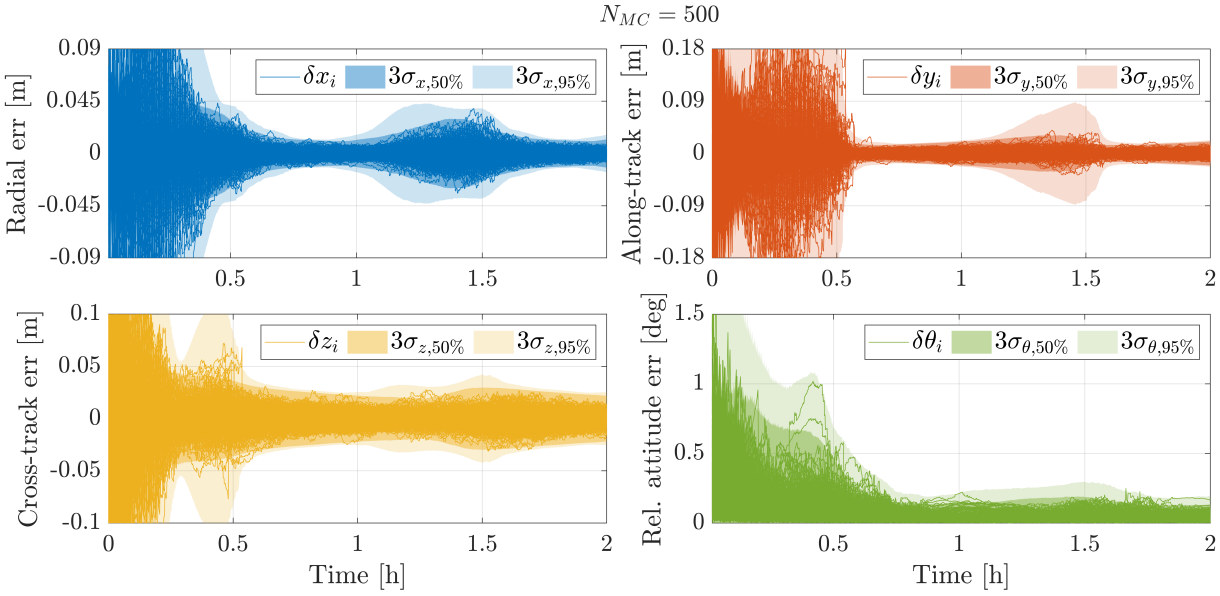

Fig. 8 MC realizations of relative position and attitude (Euler angle) errors in nominal conditions.

Table 4 MC average results in nominal conditions.

Estimated Variable	$\overline{RMSE}$	$\overline{3\sigma}_{\text{med}}$	$\overline{3\sigma}_{\text{P95}}$
Ch.1 relative position (mm)	[9.4, 14.1, 10.9]	[31.2, 40.4, 41.3]	[103.8, 139.1, 85.3]
Ch.1 relative velocity (mm/s)	[0.08, 0.09, 0.09]	[0.86, 0.89, 0.87]	[1.05, 1.1, 1.01]
Ch.1 relative attitude (deg)	0.083	0.324	0.523
Ch.1 relative ang. vel. (deg/s)	[7E-6, 6E-6, 5E-6]	[0.003, 0.003, 0.003]	[0.004, 0.004, 0.004]

Table 5 MC final results in nominal conditions.

Estimated Variable	$RMSE_f$	$3\sigma_{f,\text{med}}$	$3\sigma_{f,\text{P95}}$
Ch.1 relative position (mm)	[2.7, 5.5, 5.9]	[12.6, 21.0, 22.2]	[18.2, 23.1, 26.1]
Ch.1 relative velocity (mm/s)	[0.01, 0.01, 0.01]	[0.05, 0.06, 0.06]	[0.06, 0.06, 0.06]
Ch.1 relative attitude (deg)	0.041	0.134	0.195
Ch.1 relative ang. vel. (deg/s)	[5E-7, 4E-7, 3E-7]	[1E-4, 2E-4, 1E-4]	[2E-4, 2E-4, 1E-4]

literature. Position and attitude remain within a few centimeters and few tenths of a degree respectively, while velocity and angular velocity estimates exhibit even higher accuracy. The comparison between average and final metrics highlights the filter's reliability and its convergence toward lower errors and uncertainties. Moreover, the spikes in the 3σ bounds can be associated to the reduced number of features observed by the Ch.1, while secondary measurements are characterized by higher uncertainty, as better evidenced in the degraded case simulation.

4.4 Results in Degraded Conditions

To assess the robustness of the developed architecture, the same Monte Carlo analysis is repeated under degraded conditions. In this scenario, the primary chaser relies exclusively on measurements provided by the secondary to maintain relative navigation around the target. This setup represents the extreme situation in which the Ch.1's optical camera is non-operational, or issues in feature detection and matching occur. As in the nominal case, the MC realizations of relative position and attitude errors are outlined in Fig. 9, while average and final RMSE values with medians and 95-th percentiles of the 3σ bounds are reported in Tables 6 and 7.

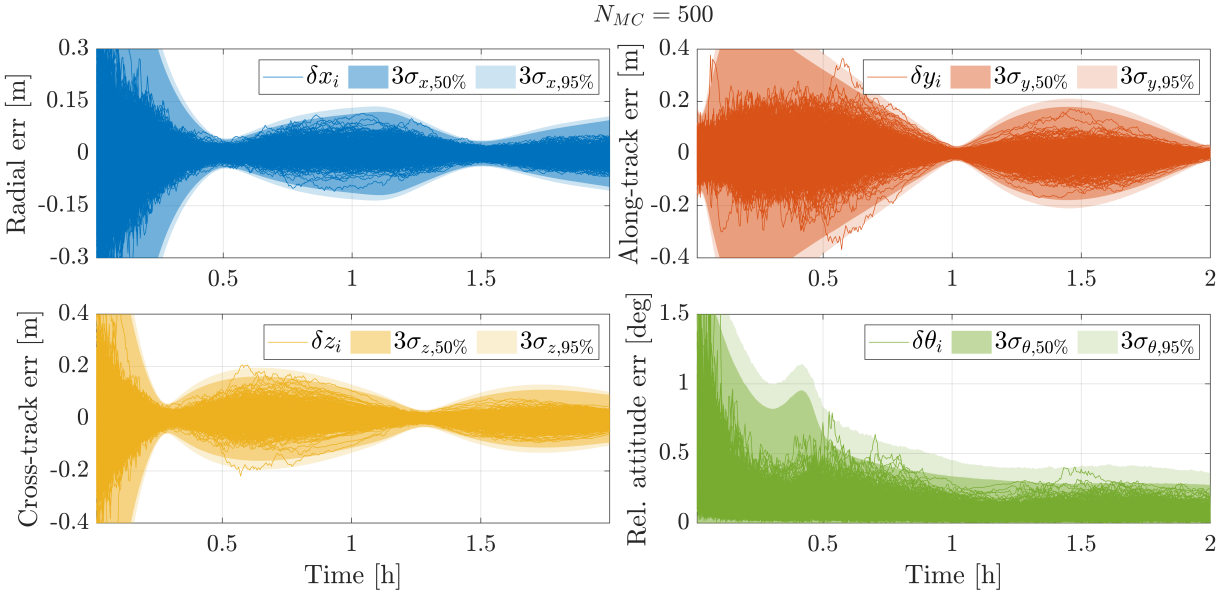

Fig. 9 MC realizations of relative position and attitude (Euler angle) errors in degraded conditions.

Table 6 MC average results in degraded conditions.

Estimated Variable	$\overline{RMSE}$	$\overline{3\sigma}_{\text{med}}$	$\overline{3\sigma}_{\text{P95}}$
Ch.1 relative position (mm)	[30.5, 44.6, 33.1]	[159.5, 200.9, 156.9]	[184.5, 237.7, 184.2]
Ch.1 relative velocity (mm/s)	[0.12, 0.13, 0.11]	[0.93, 0.90, 0.86]	[0.98, 0.95, 0.90]
Ch.1 relative attitude (deg)	0.135	0.528	0.687
Ch.1 relative ang. vel. (deg/s)	[7E-6, 7E-6, 8E-6]	[0.003, 0.003, 0.003]	[0.003, 0.003, 0.003]

Table 7 MC final results in degraded conditions.

Estimated Variable	$RMSE_f$	$3\sigma_{f,\text{med}}$	$3\sigma_{f,\text{P95}}$
Ch.1 relative position (mm)	[21.6, 6.79, 20.1]	[95.7, 27.5, 90.7]	[107.1, 28.85, 102.7]
Ch.1 relative velocity (mm/s)	[0.01, 0.04, 0.01]	[0.09, 0.15, 0.08]	[0.10, 0.17, 0.09]
Ch.1 relative attitude (deg)	0.079	0.274	0.361
Ch.1 relative ang. vel. (deg/s)	[1E-6, 2E-6, 9E-7]	[0.0002, 0.002, 0.002]	[0.002, 0.002, 0.003]

Even in the degraded case, the filter is robust and consistent with respect to uncertainties. As expected, Ch.1 relative navigation performance deteriorates due to the exclusive use of secondary measurements. These are inherently noisier due to the larger operating range and depend on a higher number of auxiliary measurements compared to direct ones. Consequently, the filter weights the prediction step more heavily, causing the estimate to follow the underlying periodic orbital dynamics. This effect manifests as the low-frequency oscillation depicted in the plot. Nevertheless, the uncertainty remains bounded, ensuring that relative navigation can still be reliably carried out, and enhancing the overall redundancy of the system.

5 Conclusions and Future Developments

This study presents a distributed architecture for relative navigation during close-range inspection of a target object using two cooperative chasers. The proposed filter performs sequential estimation of primary chaser's relative position, velocity, attitude and angular velocity by processing measurements acquired by both Ch.1 and Ch.2. Particularly, the paper presents the chaser's pointing control laws, the simulation of acquired measurements and the design of the distributed navigation filter, which combines an Extended Kalman Filter with its Multiplicative variant. Monte Carlo analyses are carried out in both nominal and degraded conditions, demonstrating the potential advantages and robustness of the architecture. Specifically, the baseline simulation demonstrated that the primary chaser alone fails to perform reliable relative navigation due to features scarcity. Conversely, by processing the measurements acquired by the secondary chaser, the architecture allows the primary inspector to accurately estimate its relative pose with respect to the target, even under degraded conditions where it relies solely on secondary measurements.

Future work will focus on extending this preliminary analysis towards more realistic scenarios, by increasing the fidelity of both dynamical and measurement models implemented in the filter. Additional studies will investigate configurations where the far chaser also exploits measurements sent from the closer one, since feature detection and matching tend to fail at larger distances from the target. Further developments will assess navigation performance in cooperative setups characterized by more than two chasers, with the ultimate goal of enabling swarm-based proximity operations. Furthermore, the integration of distributed closed-loop guidance and control laws will be explored to enhance target observability.

Acknowledgments

This work was carried out within the *Space It Up!* project funded by the Italian Space Agency (ASI) and the Italian Ministry of University and Research (MUR) under contract No. 2024-5-E.0, CUP No. I53D24000060005.

Declaration of Use of Artificial Intelligence

Artificial Intelligence was employed exclusively for language-related tasks, including proofreading, translation, and manuscript reorganization. OpenAI's ChatGPT tool was used at this scope and did not contribute to the generation of images, results or any substantial content. The authors are fully responsible for the integrity of the presented work and for all the scientific contributions, encompassing literature review, research design, coding, and manuscript structure.

References

[1] Simone D'Amico, Marco Pavone, Shailendhar Saraf, Abdulaziz Alhussien, Turki Al-Saud, Sasha Buchman, Robert Byer, and Charbel Farhat. Miniaturized autonomous distributed space system for future science and exploration. In *International Workshop on Satellite Constellations and Formation Flying*, pages 1–20. International Astronautical Federation's Astrodynamics Committee Delft, 2015.

[2] Vincent Giralo and Simone D'Amico. Development of the stanford gnss navigation testbed for distributed space systems. In *Proceedings of the 2018 International Technical Meeting of The Institute of Navigation*, pages 837–856, Reston, Virginia, 2018. doi: 10.33012/2018.15544.

[3] Per Bodin, Robin Larsson, Fredrik Nilsson, Camille Chasset, Ron Noteborn, and Matti Nylund. PRISMA: An in-orbit test bed for guidance, navigation, and control experiments. *Journal of Spacecraft and Rockets*, 46(3):615–623, 2009. doi: 10.2514/1.40161.

[4] Michel Delpech, Jean-Claude Berges, Thomas Karlsson, and Fabien Malbet. Results of PRISMA/FFIORD extended mission and applicability to future formation flying and active debris removal missions. *International Journal of Space Science and Engineering 5*, 1(4):382–409, 2013.

[5] Niels H Roth, Ben Risi, C Grant, and R Zee. Flight results from the canx-4 and canx-5 formation flying mission. In *Proceedings of the Small Satellites, Systems & Services Symposium (4S), Valletta, Malta*, volume 30, 2016.

[6] Vincenzo Capuano, Alexei Harvard, and Soon-Jo Chung. On-board cooperative spacecraft relative navigation fusing gnss with vision. *Progress in Aerospace Sciences*, 128:100761, 2022.

[7] Justin Kruger, Adam W Koenig, and Simone D'Amico. Starling formation-flying optical experiment (starfox): System design and preflight verification. *Journal of Spacecraft and Rockets*, 60(6):1755–1777, 2023.

[8] Justin J Kruger, Tommaso Guffanti, Tae Ha Park, Mason Murray-Cooper, Samuel Y Low, Toby Bell, Simone D'Amico, Christopher W Roscoe, and Jason Westphal. Adaptive end-to-end architecture for autonomous spacecraft navigation and control during rendezvous and proximity operations. In *AIAA SCITECH 2024 Forum*, page 0430, 2024. doi: 10.2514/6.2024-0430.

[9] Vadim Indelman, Stephen Williams, Michael Kaess, and Frank Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *2012 15th International Conference on Information Fusion*, pages 2154–2161. IEEE, 2012.

[10] Stephen Williams, Vadim Indelman, Michael Kaess, Richard Roberts, John J Leonard, and Frank Dellaert. Concurrent filtering and smoothing: A parallel architecture for real-time navigation and full smoothing. *The International Journal of Robotics Research*, 33(12):1544–1568, 2014.

[11] Kyle Alfriend, Srinivas Rao Vadali, Pini Gurfil, Jonathan How, and Louis Breger. *Spacecraft formation flying: dynamics, control and navigation*, volume 2. Elsevier, 2009.

[12] F Landis Markley and John L Crassidis. Attitude kinematics and dynamics. In *Fundamentals of Spacecraft Attitude Determination and Control*, pages 67–122. Springer, 2014.

[13] F Landis Markley. Attitude error representations for Kalman filtering. *Journal of guidance, control, and dynamics*, 26(2):311–317, 2003. doi: 10.2514/2.5048.

[14] Peter Sturm. Pinhole camera model. In *Computer Vision: A Reference Guide*, pages 983–986. Springer, 2021.

[15] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O (n) solution to the PnP problem. *International journal of computer vision*, 81(2):155–166, 2009. doi: 10.1007/s11263-008-0152-6.

[16] F Landis Markley and J Russell Carpenter. Generalized linear covariance analysis. *The Journal of the Astronautical Sciences*, 57(1):233–260, 2009.

[17] William G Cochran. The $\chi 2$ test of goodness of fit. *The Annals of mathematical statistics*, pages 315–345, 1952.