(Preprint) AAS 26-741

# DISTRIBUTED NAVIGATION STRATEGIES FOR MULTI-AGENT PROXIMITY OPERATIONS AROUND UNCOOPERATIVE TARGETS

Matteo Capitanio[*], Michele Maestrini[†], and Mauro Massari[‡]

The increasing adoption of Distributed Space Systems enables new capabilities for close-proximity operations around uncooperative damaged satellites or debris, requiring accurate and robust relative navigation. Two alternative frameworks are investigated: a centralized joint-state filter and a decentralized architecture, both of which leverage inter-agent information exchange. Sequential filtering techniques are employed to estimate chasers' relative state with respect to the target by fusing optical and auxiliary measurements under realistic communication constraints. Results show that covariance coupling improves translational accuracy in the centralized solution, whereas the decentralized architecture enhances attitude robustness during feature outages and removes the single point of failure.

## INTRODUCTION

Over recent years, the space sector is witnessing a paradigm shift, expanding beyond traditional monolithic spacecraft toward Distributed Space Systems (DSS). This architectural evolution is demonstrated by the successful implementation of large satellite constellations, precise formation flying, and distributed space sensor networks.[1] The combination of satellite miniaturization with the distribution of tasks among multiple coordinated units is paving the way for a new generation of space architectures.[2,3] Building upon the scalability, intrinsic redundancy, and cost-effectiveness of these multi-agent solutions, also the next frontier of close-proximity operations, such as target characterization, Rendezvous & Docking (RVD), Active Debris Removal (ADR), and in-orbit assembly and manufacturing, can be fundamentally enhanced or entirely unlocked. Specifically, rather than relying on a single monolithic servicer, deploying multiple cooperative chasers around a common, non-cooperative target object enables unprecedented operational capabilities. For instance, these include simultaneous multi-view inspection, continuous visual coverage during robotic manipulation, and cooperative target detumbling by distributing the control effort and loads across the swarm. However, executing these proximity activities necessitates highly autonomous, collision-free distributed control algorithms, which inherently rely on accurate and robust relative navigation. When transitioning to a multi-agent framework, providing this capability introduces a fundamental architectural trade-off on how the state estimation process is managed and fused across the network.

The first paradigm relies on centralized architectures, which represent the standard for theoretical optimality. In these frameworks, a central node or "mothership" gathers all measurements from the swarm to estimate the joint state of the entire DSS. For example, Sabatini et al.[4] propose a centralized visual navigation algorithm running on a primary spacecraft to identify and track multiple

[*]PhD Candidate in Aerospace Engineering, DAER, Politecnico di Milano, Via La Masa, 34, 20156 Milan, Italy
[†]Assistant Professor, DAER, Politecnico di Milano, Via La Masa, 34, 20156 Milan, Italy
[‡]Associate Professor, DAER, Politecnico di Milano, Via La Masa, 34, 20156 Milan, Italy

smaller satellites inspecting a damaged target. Similarly, Dennison et al.[5] developed a Simultaneous Navigation and Characterization (SNAC) framework where optical and radiometric measurements of an entire nanosatellite swarm are fused into a single, computationally robust centralized filter to estimate an asteroid's gravity field, rotational motion, and shape. While these filters explicitly capture complex inter-agent cross-correlations, they present stringent operational limitations: they introduce a single point of failure, poorly scale the computational complexity, and place a massive communication bottleneck on the master node.[6]

Conversely, the second paradigm explores distributed architectures, which mitigate these bottlenecks by shifting the computational effort to the individual agents. A significant contribution in this direction was made by Matsuka et al.,[7] who developed a distributed multi-target relative pose estimation algorithm to address the challenges of limited sensing and communication topologies in large cooperative spacecraft swarms. While their approach avoids the need for a global common reference frame by modeling relative dynamics assuming locally aligned Local-Vertical Local-Horizontal (LVLH) frames, the most notable contribution of their framework lies in its scalability. By restricting the pose estimation exclusively to a local subset of neighbors and sharing only relative measurements, the algorithm establishes a decentralized architecture highly suited for large-scale formations. Another notable state-of-the-art example is the Guidance, Navigation, and Control (GNC) architecture developed by Stanford's Space Rendezvous Laboratory.[8] While this framework supports multi-object navigation using optical measurements, carrier-phase differential GNSS (CDGNSS) and inter-satellite crosslinks, it presents a specific limitation when applied to non-cooperative close-proximity operations. Indeed, when CDGNSS cannot be employed and the swarm must rely on visual navigation, their target pose estimation pipeline restricts the problem to a single observer estimating the relative state of a single target. Furthermore, although the architecture enables asynchronous measurement sharing through event-driven updates, it assumes sufficiently synchronized inter-agent communication and does not explicitly model communication-induced latency or delayed measurement fusion. This becomes a critical limitation in realistic distributed vision-based navigation scenarios, where non-negligible delays between measurement acquisition and availability can degrade estimation consistency and overall system performance. Ultimately, the most critical drawback shared by both the Caltech and Stanford frameworks is their impact on efficiency. By including the states of other observers and targets in the local filters, both architectures inherently increase the state vector dimensionality, which partially reintroduces the computational burden at the agent level. Taken as a whole, these structural and operational limitations significantly constrain their applicability in realistic on-orbit servicing scenarios, where targets are uncooperative and the full exploitation of simultaneous multi-view geometry is paramount.

Recent preliminary research[9] has shown that a primary chaser can effectively overcome the inherent limitations of monocular vision at close range, such as the scarcity of target trackable features, by integrating local measurements with complementary observations provided by a secondary observer. While this cooperative strategy proved successful for the high-resolution inspection of an uncooperative target, the most effective paradigm for managing and exploiting such distributed information remains an open issue in the literature. Motivated by this gap, the presented paper provides a comprehensive comparison between a centralized joint-state filter and a decentralized relative navigation architecture to enable distributed close-proximity operations with uncooperative targets. To ensure a valid assessment, these frameworks are evaluated in a realistic testbed involving two cooperative chasers and a known satellite as a target, overcoming the assumption of perfect data exchange by enforcing realistic Inter-Satellite Link (ISL) communication constraints.

The outline of the paper is as follows. First, the two navigation architectures are proposed, detailing the specific dynamics employed, the sequential filtering logic, and the different frameworks. Following this, the paper presents the simulation environment and general setup, including the mission scenario, the optical measurements model, and the ISL model, which is essential for simulating a realistic communication channel. Next, the results are discussed, outlining the performance comparison between the two distributed strategies, together with their robustness and consistency. Finally, the key findings and possible developments are summarized in the conclusion.

## DISTRIBUTED NAVIGATION STRATEGIES

This section formalizes the distributed estimation strategies required to enable coordinated multi-agent operations. Specifically, the architectures presented herein are tailored for a reference scenario involving two cooperative chasers operating simultaneously around an uncooperative target. To perform relative navigation, the chasers are equipped with visible-wavelength monocular cameras to track target features and ISL systems to enable communication.
The core estimation process relies on a hybrid Extended Kalman Filter (EKF) and Multiplicative Extended Kalman Filter (MEKF)[10] framework. The MEKF provides an elegant solution to avoid the inconsistencies caused by the normalization constraints of standard quaternion parametrization. Indeed, it estimates the attitude error $\boldsymbol{a}$ as a three-dimensional, unconstrained vector that relates the global orientation to a unit reference quaternion.

Throughout this work, the following reference frames are adopted: $I$ denotes the inertial frame, $C$ and $T$ represent the chaser and target body-fixed frames, respectively, and $L$ refers to the target LVLH frame, defined by its radial, tangential, and out-of-plane components. In the adopted notation convention, superscripts denote the reference frame in which a quantity is expressed, while subscripts denote the relationship between two coordinate systems, defining the direction of a vector or the sense of a transformation. The only exception is the attitude error, for which the subscripts refer to the global rotation pair rather than the error itself.

### Relative Dynamics and Prediction Model

The relative translational dynamics is modeled through the Clohessy-Wiltshire (CW) equations,[11] which provide a linearized description of the chaser motion with respect to the target LVLH frame. Assuming no control input or external perturbations, the resulting dynamics is homogeneous and time-invariant, allowing the relative state to be analytically propagated from the initial conditions through the proper State Transition Matrix (STM) $\Phi_{CW}$ described in the literature. Conversely, the relative rotational dynamics are derived from quaternion kinematics and nonlinear Euler equations.[12] Given the absolute inertial orientations $\boldsymbol{q}_{CI}$ and $\boldsymbol{q}_{TI}$, alongside the body frames angular velocities $\boldsymbol{\omega}_{CI}^{C}$ and $\boldsymbol{\omega}_{TI}^{T}$ of chaser and target respectively, the relative attitude and angular velocity are defined as:

$$\boldsymbol{q}_{CT} = \boldsymbol{q}_{CI} \otimes \boldsymbol{q}_{TI}^{-1} \qquad \boldsymbol{\omega}_{CT}^{C} = \boldsymbol{\omega}_{CI}^{C} - \boldsymbol{R}_{CT}\boldsymbol{\omega}_{TI}^{T} \tag{1}$$

The time evolution of these relative quantities is obtained by numerically integrating the following differential equations.

$$\dot{\boldsymbol{q}}_{CT} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{CT}^{C})\boldsymbol{q}_{CT} \qquad \boldsymbol{\Omega}(\boldsymbol{\omega}_{CT}^{C}) = \begin{bmatrix} 0 & -\omega_{CT,x}^{C} & -\omega_{CT,y}^{C} & -\omega_{CT,z}^{C} \\ \omega_{CT,x}^{C} & 0 & \omega_{CT,z}^{C} & -\omega_{CT,y}^{C} \\ \omega_{CT,y}^{C} & -\omega_{CT,z}^{C} & 0 & \omega_{CT,x}^{C} \\ \omega_{CT,z}^{C} & \omega_{CT,y}^{C} & -\omega_{CT,x}^{C} & 0 \end{bmatrix} \tag{2}$$

$$\begin{aligned}\dot{\boldsymbol{\omega}}_{CT}^{C} = \boldsymbol{J}_C^{-1}\left(\boldsymbol{\tau}_C - \boldsymbol{\omega}_{CI}^{C}\times(\boldsymbol{J}_C\boldsymbol{\omega}_{CI}^{C})\right) - \boldsymbol{\omega}_{CI}^{C}\times\boldsymbol{\omega}_{CT}^{C} \quad + \\ + \quad \boldsymbol{R}_{CT}\left(\boldsymbol{J}_T^{-1}\left[(\boldsymbol{R}_{CT}^{\top}(\boldsymbol{\omega}_{CI}^{C}-\boldsymbol{\omega}_{CT}^{C}))\times(\boldsymbol{J}_T\boldsymbol{R}_{CT}^{\top}(\boldsymbol{\omega}_{CI}^{C}-\boldsymbol{\omega}_{CT}^{C}))\right]\right)\end{aligned} \tag{3}$$

The terms $J_C$ and $J_T$ refer respectively to the inertia matrices of the chaser and the target, and $\boldsymbol{\tau}_C$ corresponds to the chaser's control torque for target pointing. Lastly, matrix $\boldsymbol{R}_{CT}$ in Eqs. (1) and (3) corresponds to the rotation described by $\boldsymbol{q}_{CT}$, as described by the conversion between attitude parameterizations.[10] To complete the dynamics, the rate of change of the relative attitude error $\boldsymbol{a}_{CT}$ can be expressed as the cross product:

$$\dot{\boldsymbol{a}}_{CT} = -\frac{1}{2}\boldsymbol{\omega}_{CT}^{C}\times\boldsymbol{a}_{CT} \tag{4}$$

Since the relative attitude dynamics is coupled with the absolute rotational motion of the chaser, the STM is likewise affected by $\boldsymbol{a}_{CI}$ and $\boldsymbol{\omega}_{CI}^{C}$. To account for this dependency, an augmented STM, denoted as $\boldsymbol{\Phi}_{aug}$, is obtained by integrating the corresponding variational equation, $\dot{\boldsymbol{\Phi}}_{aug} = \boldsymbol{A}\boldsymbol{\Phi}_{aug}$, where the Jacobian matrix $\boldsymbol{A}$ depends on both the absolute and relative attitude and angular velocity dynamics. Its complete expression is given by:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial\dot{\boldsymbol{a}}_{CI}}{\partial\boldsymbol{a}_{CI}} & \frac{\partial\dot{\boldsymbol{a}}_{CI}}{\partial\boldsymbol{\omega}_{CI}^{C}} & \cancel{\frac{\partial\dot{\boldsymbol{a}}_{CI}}{\partial\boldsymbol{a}_{CT}}} & \cancel{\frac{\partial\dot{\boldsymbol{a}}_{CI}}{\partial\boldsymbol{\omega}_{CT}^{C}}} \\ \cancel{\frac{\partial\dot{\boldsymbol{\omega}}_{CI}^{C}}{\partial\boldsymbol{a}_{CI}}} & \frac{\partial\dot{\boldsymbol{\omega}}_{CI}^{C}}{\partial\boldsymbol{\omega}_{CI}^{C}} & \cancel{\frac{\partial\dot{\boldsymbol{\omega}}_{CI}^{C}}{\partial\boldsymbol{a}_{CT}}} & \cancel{\frac{\partial\dot{\boldsymbol{\omega}}_{CI}^{C}}{\partial\boldsymbol{\omega}_{CT}^{C}}} \\ \cancel{\frac{\partial\dot{\boldsymbol{a}}_{CT}}{\partial\boldsymbol{a}_{CI}}} & \cancel{\frac{\partial\dot{\boldsymbol{a}}_{CT}}{\partial\boldsymbol{\omega}_{CI}^{C}}} & \frac{\partial\dot{\boldsymbol{a}}_{CT}}{\partial\boldsymbol{a}_{CT}} & \frac{\partial\dot{\boldsymbol{a}}_{CT}}{\partial\boldsymbol{\omega}_{CT}^{C}} \\ \cancel{\frac{\partial\dot{\boldsymbol{\omega}}_{CT}^{C}}{\partial\boldsymbol{a}_{CI}}} & \frac{\partial\dot{\boldsymbol{\omega}}_{CT}^{C}}{\partial\boldsymbol{\omega}_{CI}^{C}} & \frac{\partial\dot{\boldsymbol{\omega}}_{CT}^{C}}{\partial\boldsymbol{a}_{CT}} & \frac{\partial\dot{\boldsymbol{\omega}}_{CT}^{C}}{\partial\boldsymbol{\omega}_{CT}^{C}} \end{bmatrix} \quad \text{where} \quad \begin{aligned} \frac{\partial\dot{\boldsymbol{a}}_{CI}}{\partial\boldsymbol{a}_{CI}} &= -\frac{1}{2}(\boldsymbol{\omega}_{CI}^{C})^{\wedge} \\ \frac{\partial\dot{\boldsymbol{a}}_{CI}}{\partial\boldsymbol{\omega}_{CI}^{C}} &= \frac{\partial\dot{\boldsymbol{a}}_{CT}}{\partial\boldsymbol{\omega}_{CT}^{C}} = \boldsymbol{I}_{3\times3} \\ \frac{\partial\dot{\boldsymbol{\omega}}_{CI}^{C}}{\partial\boldsymbol{\omega}_{CI}^{C}} &= -\boldsymbol{J}_C^{-1}\left[(\boldsymbol{\omega}_{CI}^{C})^{\wedge}\boldsymbol{J}_C - (\boldsymbol{J}_C\boldsymbol{\omega}_{CI}^{C})^{\wedge}\right] \\ \frac{\partial\dot{\boldsymbol{a}}_{CT}}{\partial\boldsymbol{a}_{CT}} &= -\frac{1}{2}(\boldsymbol{\omega}_{CT}^{C})^{\wedge} \end{aligned}$$

$$\begin{aligned} \frac{\partial\dot{\boldsymbol{\omega}}_{CT}^{C}}{\partial\boldsymbol{\omega}_{CI}^{C}} &= \frac{\partial\dot{\boldsymbol{\omega}}_{CI}^{C}}{\partial\boldsymbol{\omega}_{CI}^{C}} + \left(\boldsymbol{\omega}_{CT}^{C}\right)^{\wedge} + \boldsymbol{R}_{CT}\boldsymbol{J}_T^{-1}\boldsymbol{M}\boldsymbol{R}_{CT}^{\top} \\ \frac{\partial\dot{\boldsymbol{\omega}}_{CT}^{C}}{\partial\boldsymbol{\omega}_{CT}^{C}} &= -\left(\boldsymbol{\omega}_{CI}^{C}\right)^{\wedge} - \boldsymbol{R}_{CT}\boldsymbol{J}_T^{-1}\boldsymbol{M}\boldsymbol{R}_{CT}^{\top} \\ \text{with} \quad \boldsymbol{M} &= -\left(\boldsymbol{J}_T\boldsymbol{R}_{CT}^{\top}(\boldsymbol{\omega}_{CI}^{C}-\boldsymbol{\omega}_{CT}^{C})\right)^{\wedge} + \left(\boldsymbol{R}_{CT}^{\top}(\boldsymbol{\omega}_{CI}^{C}-\boldsymbol{\omega}_{Ct}^{C})\right)^{\wedge}\boldsymbol{J}_T \end{aligned} \tag{5}$$

The matrix $\boldsymbol{I}$ denotes the identity matrix, whereas the superscript "$\wedge$" represents the skew-symmetric operator. The term $\partial\dot{\boldsymbol{\omega}}_{CT}^{C}/\partial\boldsymbol{a}_{CT}$, together with unexplained Jacobians involved in this work, has been obtained using the *MATLAB*® Symbolic Math Toolbox, and the complete set of these symbolic derivations has been made available in a dedicated public repository*. The rotational STM $\boldsymbol{\Phi}_{rot}$ is then extracted from $\boldsymbol{\Phi}_{aug}$, and the full state transition matrix $\boldsymbol{\Phi}$, is constructed as:

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

*https://github.com/Matteo-Capitanio/Symbolic-Jacobians-AAS-2026.git

### Sequential Filtering Scheme

The overall objective is to accurately estimate a local state vector for each chaser, defined as:

$$\hat{\boldsymbol{x}}_i = \begin{bmatrix} \boldsymbol{r}^L_{C_iT} & \boldsymbol{v}^L_{C_iT} & \boldsymbol{a}_{C_iT} & \boldsymbol{\omega}^{C_i}_{C_iT} \end{bmatrix}^\top \tag{7}$$

where $\boldsymbol{r}^L_{C_iT}$ and $\boldsymbol{v}^L_{C_iT}$ are the relative position and velocity, while $\boldsymbol{a}_{C_iT}$ and $\boldsymbol{\omega}^C_{C_iT}$ represent the relative attitude error and angular velocity of the $i$-th chaser with respect to the target.
The framework integrates auxiliary measurements, namely the target ephemerides, chasers' inertial attitude and angular velocity, and their 3D-baseline provided by CDGNSS, which are assumed to be estimated via independent systems. To account for their associated uncertainties in both the prediction and update steps without explicitly augmenting the state vector, the filter exploits the Consider Covariance Analysis[13] approach (CCA), under the assumption that these uncertainties remain constant and uncorrelated over time.
For a generic iteration from $t_k$ to $t_{k+1}$, the a priori state estimate $\hat{\boldsymbol{x}}_k$ is propagated through the nonlinear process model as

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

where $f(\cdot)$ combines the translational and rotational dynamics previously introduced. The corresponding covariance $\boldsymbol{P}^+_k$ is propagated through the linearized STM $\boldsymbol{\Phi}_k$ defined in Eq. (6). However, since the relative rotational model explicitly depends on the chaser's absolute inertial angular velocity $\boldsymbol{\omega}^C_{CI}$, the previously introduced CCA approach is formally applied to incorporate this uncertainty, denoted with the covariance $\boldsymbol{P}_\omega$, into the prediction step. Its contribution is mapped into the local filter covariance through the sensitivity matrix $\boldsymbol{\Gamma}_k$, yielding:

$$\boldsymbol{P}^-_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{P}^+_k \boldsymbol{\Phi}^\top_k + \boldsymbol{\Gamma}_k \boldsymbol{P}_\omega \boldsymbol{\Gamma}^\top_k + \boldsymbol{\mathcal{Q}}_{k+1} \quad \text{where} \quad \boldsymbol{\Gamma} k = \begin{bmatrix} \mathbf{0}_{6\times3} \\ \boldsymbol{\Phi}_\omega \end{bmatrix} \tag{9}$$

Here, the zero block reflects the decoupling of the translational dynamics from the chaser's absolute angular velocity, while the lower block $\boldsymbol{\Phi}_\omega$ is extracted directly from the augmented rotational STM $\Phi_{aug}$, representing the sensitivity of $\boldsymbol{a}_{C_iT}$ and $\boldsymbol{\omega}^{C_i}_{C_iT}$ to the chaser's angular velocity. The term $\boldsymbol{\mathcal{Q}}_{k+1}$ refers to the overall process noise matrix computed for the time gap of the iteration.

The filter dynamically switches between two update modes based on target feature visibility. It uses a tightly coupled mode to process raw 2D pixel coordinates when fewer than five features are tracked, and a loosely coupled mode when five or more are available, relying on $C$-to-$T$ frame relative pose measurements. The computation of the expected measurement vector $\hat{\boldsymbol{z}}_{k+1}$ and the Jacobian $\boldsymbol{H}_{k+1}$ as functions of $\hat{\boldsymbol{x}}^-_{k+1}$ is detailed in the next paragraphs, as it represents the main difference between the centralized and decentralized architectures. Letting $\boldsymbol{z}(t_{k+1})$ be the actual acquired measurement vector, the innovation is simply defined as $\boldsymbol{y}_{k+1} = \boldsymbol{z}(t_{k+1}) - \hat{\boldsymbol{z}}_{k+1}$. Finally, the CCA approach is applied again to account for auxiliary measurement uncertainties during the update step. Indeed, the covariance matrix of each of the $g$ auxiliary measurements is projected into the measurement space and added to the main contributors to compute the innovation covariance matrix $\boldsymbol{S}_{k+1}$:

$$\boldsymbol{S}_{k+1} = \boldsymbol{H}_{k+1} \boldsymbol{P}^-_{k+1} \boldsymbol{H}^\top_{k+1} + \boldsymbol{\mathcal{R}}_{k+1} + \sum_{j=1}^{g} \boldsymbol{H}_j \boldsymbol{P}_j \boldsymbol{H}^\top_j \tag{10}$$

where $\boldsymbol{\mathcal{R}}_{k+1}$ represents the measurement noise matrix, $\boldsymbol{P}_j$ denotes the covariance matrix of the $j$-th auxiliary measurement, and $\boldsymbol{H}_j$ is its corresponding sensitivity Jacobian. The latter are likewise generated symbolically and have been made available in the aforementioned public repository.

Before proceeding with the state update, an outlier-rejection process is performed based on the Chi-Squared ($\chi^2$) test.[14] To this end, the Square Mahalanobis Distance (SMD) of the innovation is employed as the test statistic:

$$\text{SMD} = \boldsymbol{y}_{k+1}^\top \boldsymbol{S}_{k+1}^{-1} \boldsymbol{y}_{k+1} \tag{11}$$

This step is of paramount importance to prevent filter inconsistencies caused by spurious measurements. If the test has a positive outcome (i.e., SMD $< \chi^2_{lim}$), the standard EKF update equations are applied to compute the Kalman gain $\boldsymbol{K}_{k+1}$, the updated state $\hat{\boldsymbol{x}}_{k+1}^+$ and covariance $\boldsymbol{P}_{k+1}^+$:

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^- \boldsymbol{H}_{k+1}^\top \boldsymbol{S}_{k+1}^{-1} \tag{12}$$

$$\hat{\boldsymbol{x}}_{k+1}^+ = \hat{\boldsymbol{x}}_{k+1}^- + \boldsymbol{K}_{k+1} \boldsymbol{y}_{k+1} \qquad \boldsymbol{P}_{k+1}^+ = \left(\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{H}_{k+1}\right) \boldsymbol{P}_{k+1}^- \tag{13}$$

Conversely, if the measurements are rejected by the $\chi^2$ test, the update step is skipped, and the a posteriori state and covariance remain equal to the predicted ones. It is worth noting that, in the tightly coupled mode, the $\chi^2$ test is performed individually for each pixel, and the subsequent update is carried out using only the subset of pixels that successfully pass the threshold.
Finally, following the MEKF architecture, the relative attitude error $\boldsymbol{a}_{CT}$ estimated within $\hat{\boldsymbol{x}}_{k+1}^+$ is extracted to correct the reference quaternion $\boldsymbol{q}_{CT}$. The update is performed via a quaternion multiplication:

$$\boldsymbol{q}_{C_T}^+ = \delta \boldsymbol{q} \left( \hat{\boldsymbol{a}}_{C_T}^+ \right) \otimes \boldsymbol{q}_{C_T}^- \tag{14}$$

The mathematical relationship between $\delta \boldsymbol{q}$ and $\boldsymbol{a}$ depends on the adopted attitude error parameterization, as described by F. L. Markley.[10] After updating the reference quaternion, the attitude error component in the local state vector is reset to zero, thus finishing the current iteration.

To conclude the filtering core scheme, a dedicated time-management logic is presented in Algorithm 1. This allows the event-driven estimation process to effectively handle communication latencies and delayed measurements received via ISL.

**Algorithm 1:** Handling of Measurements Logic - Generic iteration $t_j \rightarrow t_{j+1}$

**Result:** Updated state estimate $\hat{\boldsymbol{x}}_{j+1}^+$ and covariance $\boldsymbol{P}_{j+1}^+$

**if** $j > 1$ **then** Save backup state estimate $\hat{\boldsymbol{x}}_{j-1}$, $\boldsymbol{P}_{j-1}$, and measurements $\boldsymbol{z}(t_j)$
**for** *each measurement at* $t_{j+1}$ **do**
    Read measurement timestamp $t_m$
    **if** $t_m < t_{j+1}$ *(delayed measurement)* **then**
        **if** $t_m > t_j$ **then**
            Propagate, update, and propagate: $t_j \rightarrow t_m^+ \rightarrow t_{j+1}$
        **else**
            **if** $t_m < t_{j-1}$ **then**
                Skip measurement
            **else**
                Propagate backup $\hat{\boldsymbol{x}}_{j-1}$, $\boldsymbol{P}_{j-1}$ and update: $t_{j-1} \rightarrow t_m^+$
                Propagate, update with backup $\boldsymbol{z}(t_j)$, and propagate: $t_m \rightarrow t_j^+ \rightarrow t_{j+1}$
    **else**
        Propagate and update: $t_j \rightarrow t_{j+1}^+$

### Centralized Joint-State Filter

In the centralized architecture, the state estimation process is entirely performed by a primary node, namely Chaser 1, which acts as the mothership of the DSS. As illustrated in Figure 1, the global EKF/MEKF framework fuses the local optical measurements acquired by Chaser 1, the remote optical acquisitions transmitted by Chaser 2 via the ISL, and the set of auxiliary measurements. To simultaneously estimate the kinematics of both agents, the state vector is augmented to concatenate the local states of the two chasers:

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{r}^L_{C_1T} & \boldsymbol{v}^L_{C_1T} & \boldsymbol{a}_{C_1T} & \boldsymbol{\omega}^{C_1}_{C_1T} & \boldsymbol{r}^L_{C_2T} & \boldsymbol{v}^L_{C_2T} & \boldsymbol{a}_{C_2T} & \boldsymbol{\omega}^{C_2}_{C_2T} \end{bmatrix}^\top \tag{15}$$

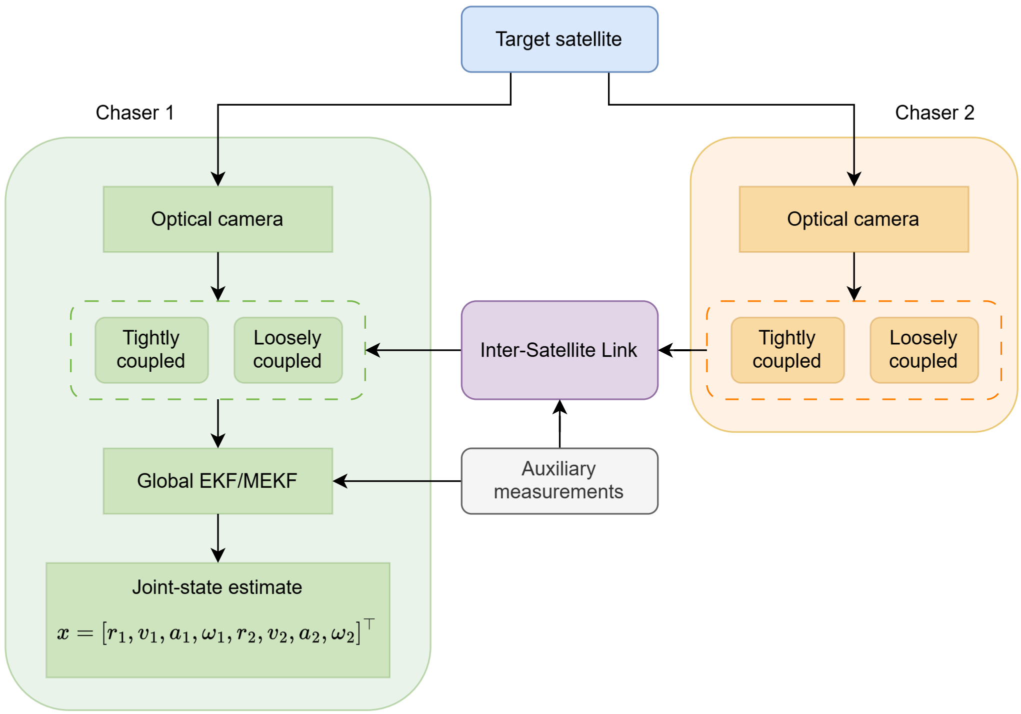


**Figure 1. Centralized Joint-State Filter Diagram.**

Consequently, all the filter quantities are dimensionally expanded to accommodate the joint state. The augmented state covariance matrix $\boldsymbol{P}$, the process noise matrix $\boldsymbol{Q}$, and the state transition matrix $\boldsymbol{\Phi}$ are constructed as block-diagonal matrices composed of the individual chasers' matrices, under the assumption that the relative dynamics of the two agents are mutually decoupled.

A major distinction of the centralized approach is the direct integration of the CDGNSS inter-satellite baseline $\boldsymbol{b}^L_{C_2C_1}$ as a direct measurement within the filter. The corresponding expected measurement model $\hat{\boldsymbol{z}}_{\boldsymbol{b}}$ and its associated Jacobian $\boldsymbol{H}_{\boldsymbol{b}}$ are defined as:

$$\hat{\boldsymbol{z}}_{\boldsymbol{b}}(\hat{\boldsymbol{x}}^-) = \boldsymbol{r}^L_{C_2T} - \boldsymbol{r}^L_{C_1T} \qquad \boldsymbol{H}_{\boldsymbol{b}} = \begin{bmatrix} -\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times9} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times9} \end{bmatrix} \tag{16}$$

Regarding optical acquisitions, the expected measurement vector and the corresponding Jacobian are composed as follows based on the active measurement mode.

First, the rotational matrix $\boldsymbol{R}_{IL}$ required to pass from the LVLH frame to the inertial frame $I$ is defined as the transpose matrix of $\boldsymbol{R}_{LI}$ whose rows are respectively:

$$\hat{\boldsymbol{x}}_L = \frac{\boldsymbol{r}_T^I}{||\boldsymbol{r}_T^I||} \qquad \hat{\boldsymbol{y}}_L = \hat{\boldsymbol{z}}_L \times \hat{\boldsymbol{x}}_L \qquad \hat{\boldsymbol{z}}_L = \frac{\boldsymbol{r}_T^I \times \boldsymbol{v}_T^I}{||\boldsymbol{r}_T^I \times \boldsymbol{v}_T^I||} \tag{17}$$

where $\boldsymbol{r}_T^I$ and $\boldsymbol{v}_T^I$ are target's position and velocity expressed in Earth Centered Inertial (ECI) reference frame.
Therefore, if pose measurements are available:

$$\hat{\boldsymbol{z}}_{opt_1} = \begin{bmatrix} -\boldsymbol{R}_{C_1I}\boldsymbol{R}_{IL}\boldsymbol{r}_{C_1T}^L \\ \boldsymbol{a}_{C_1T} \end{bmatrix} \qquad \boldsymbol{H}_{opt_1} = \begin{bmatrix} -\boldsymbol{R}_{C_1I}\boldsymbol{R}_{IL} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \end{bmatrix} \tag{18}$$

$$\hat{\boldsymbol{z}}_{opt_2} = \begin{bmatrix} -\boldsymbol{R}_{C_2I}\boldsymbol{R}_{IL}\boldsymbol{r}_{C_2T}^L \\ \boldsymbol{a}_{C_2T} \end{bmatrix} \qquad \boldsymbol{H}_{opt_2} = \begin{bmatrix} \mathbf{0}_{3\times12} & -\boldsymbol{R}_{C_2I}\boldsymbol{R}_{IL} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times12} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{19}$$

with matrices $\boldsymbol{R}_{C_iI}$ computed as functions of the inertial orientations of the chasers, included in the auxiliary measurements. According to the MEKF formulation, the expected attitude errors $\boldsymbol{a}_{C_iT}$ are null vectors. As a consequence, the real measurement $\boldsymbol{z}_{opt_i}$ contains the actual attitude error, which is obtained as the deviation between the true pose measurement quaternion and the associated reference $\boldsymbol{q}_{C_iT}$.
Conversely, if fewer than five features are detected, the 2D features are used directly. Specifically, the 3D position of a target's traceable feature $\boldsymbol{p}_T^T$, defined in the target's body frame $T$, is transformed into the $i$-th chaser's camera frame. Assuming this camera frame is aligned with its respective body frame, the transformation is performed through the following relation:

$$\boldsymbol{p}_{cam_i} = \boldsymbol{R}_{C_iT}\boldsymbol{p}_T^T - \boldsymbol{R}_{C_iI}\boldsymbol{R}_{IL}\boldsymbol{r}_{C_iT}^L \tag{20}$$

By projecting this 3D position onto the image plane, the expected 2D measurement vectors and their corresponding Jacobians for the two chasers are obtained as:

$$\hat{\boldsymbol{z}}_{opt_1} = \boldsymbol{K}\frac{1}{Z_1}\boldsymbol{p}_{cam_1} \qquad \boldsymbol{H}_{opt_1} = \begin{bmatrix} -\boldsymbol{B}_1\boldsymbol{R}_{C_1I}\boldsymbol{R}_{IL} & \mathbf{0}_{2\times3} & \boldsymbol{B}_1\dfrac{\partial \boldsymbol{p}_{cam_1}}{\partial \boldsymbol{a}_{C_1T}} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times12} \end{bmatrix} \tag{21}$$

$$\hat{\boldsymbol{z}}_{opt_2} = \boldsymbol{K}\frac{1}{Z_2}\boldsymbol{p}_{cam_2} \qquad \boldsymbol{H}_{opt_2} = \begin{bmatrix} \mathbf{0}_{2\times12} & -\boldsymbol{B}_2\boldsymbol{R}_{C_2I}\boldsymbol{R}_{IL} & \mathbf{0}_{2\times3} & \boldsymbol{B}_2\dfrac{\partial \boldsymbol{p}_{cam_2}}{\partial \boldsymbol{a}_{C_2T}} & \mathbf{0}_{2\times3} \end{bmatrix} \tag{22}$$

where the camera intrinsic matrix $\boldsymbol{K}$ and the matrices $\boldsymbol{B_i}$ are defined as:

$$\boldsymbol{K} = \begin{bmatrix} f/d & 0 & u_c \\ 0 & f/d & v_c \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{B}_i = \begin{bmatrix} \dfrac{f}{d\ Z_i} & 0 & -\dfrac{f\ X_i}{d\ Z_i^2} \\ 0 & \dfrac{f}{d\ Z_i} & -\dfrac{f\ Y_i}{d\ Z_i^2} \end{bmatrix} \tag{23}$$

Let $\{X_i, Y_i, Z_i\}$ be the 3D coordinates of the feature point $\boldsymbol{p}_{cam_i}$ expressed in the $i$-th camera frame. Furthermore, $f$ represents the focal length of the camera, $d$ is the physical dimension of a single pixel on the sensor, and $(u_c, v_c)$ denote the pixel coordinates of the image's principal point. The derivatives $\partial\boldsymbol{p}_{cam_i}/\partial\boldsymbol{a}_{C_iT}$ are symbolically computed, taking into account the relation between the attitude errors and the reference quaternions expressed by Eq. (14). Finally, it is worth remarking that in the tightly coupled mode, the vectors $\hat{\boldsymbol{z}}_{opt_i}$ have dimension $[2m \times 1]$ while the Jacobians $\boldsymbol{H}_{opt_i}$ are $[2m \times 24]$, with $m$ being the number of modeled features that matched the detected ones. Importantly, whenever multiple independent observations, such as optical and CDGNSS data, share the same timestamp, the respective estimate updates are executed sequentially rather than in a single augmented step. Assuming uncorrelated measurement noises, this block-wise approach allows the outlier-rejection mechanism to isolate specific anomalies, preventing a single spurious reading from invalidating an entire batch of concurrent valid data.

### Decentralized Architecture

In contrast to the centralized framework, the decentralized architecture distributes the navigation process, maintaining an independent EKF/MEKF onboard each chaser. As illustrated in Figure 2, each node solely estimates its own local state vector, stated in Eq. 7, effectively halving the state dimensionality compared to the joint-state filter. To achieve this, the local filter fuses the optical observations acquired onboard with the remote optical data transmitted by the cooperative spacecraft via the ISL. Consistent with the centralized approach, auxiliary measurements remain essential for the prediction and update steps; these parameters are similarly exchanged through the ISL and systematically integrated into the local estimation pipeline using the CCA approach.

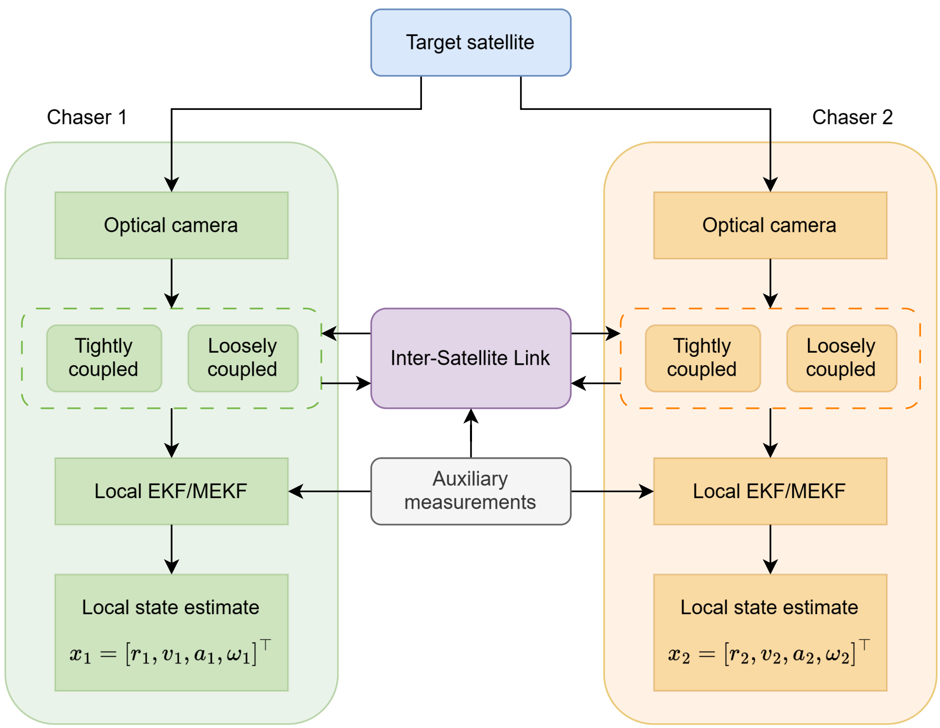


**Figure 2. Decentralized Architecture Diagram.**

A pivotal aspect of this architecture is the employment of the inter-satellite baseline $\boldsymbol{b}$ and the absolute inertial orientations $\boldsymbol{q}_{C_1I}$ and $\boldsymbol{q}_{C_2I}$ as auxiliary parameters to mathematically relate the local chaser's state to the observations acquired by the remote one. The expected measurement vectors and their associated Jacobians are derived hereafter. For the sake of clarity, the formulations are generalized for a generic $i$-th chaser, where the subscript $j$ denotes the remote cooperative agent.

When the filter operates in loosely coupled mode:

$$\hat{\boldsymbol{z}}_{opt_i} = \begin{bmatrix} -\boldsymbol{R}_{C_iI}\boldsymbol{R}_{IL}\boldsymbol{r}^L_{C_iT} \\ \boldsymbol{a}_{C_iT} \end{bmatrix} \qquad \boldsymbol{H}_{opt_i} = \begin{bmatrix} -\boldsymbol{R}_{C_iI}\boldsymbol{R}_{IL} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix} \tag{24}$$

$$\hat{\boldsymbol{z}}_{opt_j} = \begin{bmatrix} -\boldsymbol{R}_{C_jI}\boldsymbol{R}_{IL}\boldsymbol{r}^L_{C_jT} \\ \boldsymbol{a}_{C_jT} \end{bmatrix} \qquad \boldsymbol{H}_{opt_j} = \begin{bmatrix} -\boldsymbol{R}_{C_jI}\boldsymbol{R}_{IL} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \dfrac{\partial \boldsymbol{a}_{C_jT}}{\partial \boldsymbol{a}_{C_iT}} & \boldsymbol{0}_{3\times3} \end{bmatrix} \tag{25}$$

Otherwise, when the tightly coupled mode is requested:

$$\hat{\boldsymbol{z}}_{opt_i} = \boldsymbol{K}\frac{1}{Z_i}\boldsymbol{p}_{cam_i} \qquad \boldsymbol{H}_{opt_i} = \begin{bmatrix} -\boldsymbol{B}_i\boldsymbol{R}_{C_iI}\boldsymbol{R}_{IL} & \boldsymbol{0}_{2\times3} & \boldsymbol{B}_i\dfrac{\partial \boldsymbol{p}_{cam_i}}{\partial \boldsymbol{a}_{C_iT}} & \boldsymbol{0}_{2\times3} \end{bmatrix} \tag{26}$$

$$\hat{\boldsymbol{z}}_{opt_j} = \boldsymbol{K}\frac{1}{Z_j}\boldsymbol{p}_{cam_j} \qquad \boldsymbol{H}_{opt_j} = \begin{bmatrix} -\boldsymbol{B}_j\boldsymbol{R}_{C_jI}\boldsymbol{R}_{IL} & \boldsymbol{0}_{2\times3} & \boldsymbol{B}_j\dfrac{\partial \boldsymbol{p}_{cam_j}}{\partial \boldsymbol{a}_{C_iT}} & \boldsymbol{0}_{2\times3} \end{bmatrix} \tag{27}$$

To process the remote acquisitions, the $i$-th chaser must mathematically map its local state estimate into the remote measurement space. This is achieved by reconstructing the expected relative position and reference quaternion of the $j$-th chaser as follows:

$$\boldsymbol{r}^L_{C_jT} = \boldsymbol{r}^L_{C_iT} + \boldsymbol{b}^L_{C_jC_i} \tag{28}$$

$$\boldsymbol{q}_{C_jT} = \boldsymbol{q}_{C_jI} \otimes (\boldsymbol{q}_{C_iI})^{-1} \otimes \boldsymbol{q}_{C_iT} \tag{29}$$

The partial derivatives $\partial\boldsymbol{a}_{C_jT}/\partial\boldsymbol{a}_{C_iT}$ and $\partial\boldsymbol{p}_{cam_j}/\partial\boldsymbol{a}_{C_iT}$ are evaluated with the *MATLAB*® Symbolic Math Toolbox, whereas all remaining variables are defined consistently with the centralized formulation.

## SIMULATION ENVIRONMENT AND SETUPS

This section details the high-fidelity simulation environment established to validate the proposed distributed navigation architectures. It outlines the reference mission scenario, the modeling of the optical measurements and the realistic ISL, together with filters initialization and tuning.

### Mission Scenario

The swarm is tasked with operating in proximity to an uncooperative object modeled after the Sentinel-1A satellite. As illustrated in Figure 3, the target presents a set of recognizable features, which have been selected through *Blender*®. Moreover, a subset of these features is randomly designated as defective to simulate permanent damage on the target. The orbital regime corresponds to the nominal Sun-synchronous orbit of Sentinel-1A, while its attitude initial conditions are defined by the quaternion $\mathbf{q}_{TI}(t_0) = [0.6866, 0.3166, 0.3546, 0.5501]^\top$, and the angular velocity $\boldsymbol{\omega}^T_{TI}(t_0) = [3\cdot10^{-5}, 2\cdot10^{-3}, 2\cdot10^{-4}]^\top$ rad/s.
The two chasers are modeled as rigid cuboids, and their physical properties and initial orbital conditions with respect to the $L$ frame are summarized in Table 1.

The orbital trajectories are generated using a proprietary orbital simulator, which accounts for major perturbations such as Earth's gravitational harmonics, solar radiation pressure, atmospheric drag, and third-body gravity (Sun and Moon). At this stage, no translational control laws are imposed. Conversely, the attitude motion is propagated by integrating Euler's equations for rigid

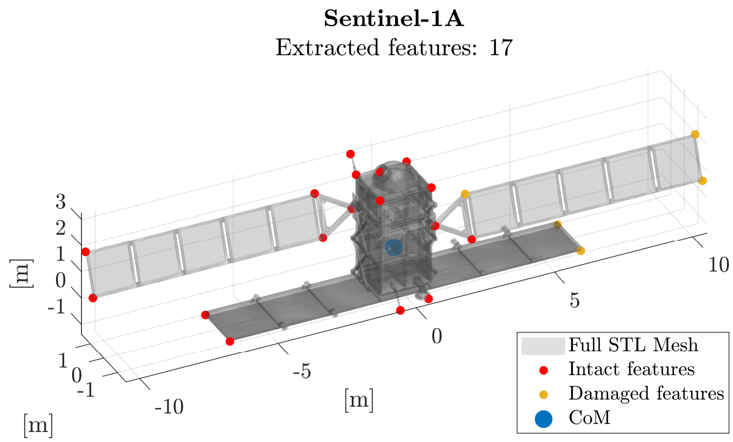


**Figure 3. Target object CAD with selected features.**

**Table 1. Chasers' physical properties and initial relative state**

| | Chaser 1 | Chaser 2 |
|---|---|---|
| Mass (kg) | 50 | 10 |
| Dimensions (m) | $0.2 \times 0.3 \times 0.8$ | $0.2 \times 0.3 \times 0.2$ |
| Initial rel. position, $\boldsymbol{r}^L_{C_iT}(t_0)$ (m) | $[0, 20, -20]^\top$ | $[0, 100, 50]^\top$ |
| Initial rel. velocity, $\boldsymbol{v}^L_{C_iT}(t_0)$ (m/s) | $[0.01, 0, 0.02]^\top$ | $[0.05, 0, 0.05]^\top$ |

bodies.[12] Environmental disturbance torques are neglected, as attitude control is enforced to ensure target pointing. Since closed-loop control design falls outside the scope of this work, these pointing commands are computed directly from ground-truth data. The $i$-th chaser's desired inertial orientation is obtained by aligning its camera optical axis (the third axis of its body-fixed frame $C_i$) with the chaser-to-target direction expressed in frame $I$. The reference angular velocity and the required control torque are subsequently derived as:

$$\boldsymbol{\omega}^{C_i}_{C_iI} = 2 \cdot \boldsymbol{q}^{-1}_{C_iI} \otimes \dot{\boldsymbol{q}}_{C_iI} \qquad \boldsymbol{\tau}_{C_i} = \mathbf{J}_{C_i}\dot{\boldsymbol{\omega}}^{C_i}_{C_iI} + \boldsymbol{\omega}^{C_i}_{C_iI} \times (\boldsymbol{J}_{C_i}\boldsymbol{\omega}^{C_i}_{C_iI}) \tag{30}$$

The time derivatives $\dot{\boldsymbol{q}}_{C_iI}$ and $\dot{\boldsymbol{\omega}}^{C_i}_{C_iI}$ are numerically approximated using a non-uniform central finite difference scheme, with forward and backward three-point schemes employed at the boundaries.

## Optical Measurements Model and Setup

The visual acquisition simulation begins with the conversion of the target's features $\boldsymbol{p}^T_T$, expressed in its body-fixed frame $T$, into the camera frame. Eq. 20 is employed to perform this transformation. Subsequently, the Pinhole camera model[15] is used to map the three-dimensional coordinates $\boldsymbol{p}_{cam}$ onto the two-dimensional image plane with the camera intrinsic matrix $\boldsymbol{K}$ defined with Eq. 23.

To ensure a high-fidelity simulation environment, the extraction of 2D measurements is subjected to three visibility constraints. First, a Field of View (FOV) constraint requires that the projected feature falls within the physical limits of the optical sensor. The FOV bounds are computed as:

$$\text{FOV}_i = 2\arctan\left(\frac{d \cdot 2i_c}{2f}\right) \qquad i \in \{u, v\} \tag{31}$$

Second, to account for geometric occlusions, a ray-tracing algorithm, based on radius-triangle intersection method,[16] is utilized to verify that the line of sight between the camera and the feature is not obstructed by the target's own structure. Finally, an illumination condition is enforced by evaluating the direction of the solar vector relative to the surface normal associated with the feature, ensuring that points in complete shadow are discarded.

A realistic pixel noise model is subsequently applied to the valid measurements. Rather than assuming a constant variance, the Gaussian noise variance $\sigma^2_{pix_i}$ for each pixel axis is computed dynamically as the sum of squared independent physical contributions:[17]

$$\sigma^2_{pix_i} = \sigma^2_{std} + \sigma^2_{defocus} + \sigma^2_{motion,i} + \sigma^2_{dist} \tag{32}$$

These components capture distinct optical and dynamic artifacts. The standard noise $\sigma_{std}$ accounts for the intrinsic electronic and quantization uncertainties. The defocus blur $\sigma_{defocus}$ arises from the discrepancy between the feature's depth $Z$ and the camera's focus distance $Z_f$. This spatial spread[18] is scaled by the focal length $f$, pixel size $d$, and aperture $N$:

$$\sigma_{defocus} = k_{def} \frac{f^2}{Nd(Z_f - f)} \left| \frac{Z - Z_f}{Z} \right| \tag{33}$$

Additionally, motion blur captures the image smear caused by the relative transverse velocity $v_i$ during the exposure time $t_{exp}$:[19]

$$\sigma_{motion,i} = k_{mot} f \frac{v_i t_{exp}}{dZ} \tag{34}$$

Finally, perimetric distortion, or vignetting, introduces a radial quadratic error that progressively increases with the Euclidean distance $\rho$ from the image's principal point up to the maximum sensor radius $\rho_{max}$.[20] This effect is expressed as:

$$\sigma_{dist} = k_{vig} \left( \frac{\rho}{\rho_{max}} \right)^2 \tag{35}$$

To replicate the criticalities typically associated with onboard image processing algorithms, two independent probabilistic degradation events are introduced at each timestep. Each event possesses a separate $50\%$ chance of occurring and can affect up to a maximum of three visible features. The first simulates a detection failure by entirely removing the selected points from the available set. The second emulates a matching failure, in which the true pixel coordinates are randomly reassigned across the image plane, generating critical outliers.

Following these degradation steps, if the sensor successfully maintains the tracking of at least 5 valid features, the relative pose of the chaser with respect to the target is computed by solving the Perspective-n-Point (PnP) problem via the EPnP algorithm.[21] To guarantee robustness against the severe outliers introduced by matching failures, the EPnP solver is wrapped within a RANdom SAmple Consensus (RANSAC) architecture.[22] This robust estimator iteratively samples minimal subsets of points, evaluates the resulting pose hypotheses by discarding features with a reprojection error exceeding a predefined tolerance, and returns the optimal pose measurement, i.e. translation and rotation, based on the largest consensus set. Conversely, if fewer than 5 features remain available, the system aborts the pose estimation process and directly saves the raw 2D pixel coordinates.

Table 2 summarizes the key parameters used to model the camera, the realistic noise, and the RANSAC logic.

**Table 2. Optical Camera and Measurement Simulation Parameters**

| Parameter | Value |
|---|---|
| Sensor size (pixels) | $2048 \times 2048$ |
| Focal length, $f$ (mm) | 31 |
| Pixel size, $d$ ($\mu$m) | 5.5 |
| Aperture, $N$ | 5.6 |
| Focus distance, $Z_f$ (m) | 20 |
| Exposure time, $t_{exp}$ (s) | 0.005 |
| Standard pixel noise, $\sigma_{std}$ (pix) | 1.8 |
| Defocus coefficient, $k_{defocus}$ | 0.8 |
| Motion coefficient, $k_{mot}$ | 0.5 |
| Vignetting coefficient, $k_{vig}$ | 1.2 |
| RANSAC Max Iterations | 100 |
| RANSAC Reproj. tol. (pix) | 10 |

### Inter-Satellite Link Model and Setup

The ISL model adopted in this work aims to reproduce the main physical and protocol-level effects of a realistic radio-frequency crosslink enabling communication between two cooperative spacecraft. The implemented architecture mirrors the physical and data-link layers of a typical Open System Interconnection (OSI) framework,[23] which dictate the overall communication performance.

The system utilizes a deterministic Time Division Multiple Access (TDMA) architecture for channel scheduling. Each spacecraft is assigned a specific transmission slot within a periodic frame. This minimizes frequency allocation complexity and mitigates co-channel interference compared to Frequency Division Multiple Access (FDMA) and Code Division Multiple Access (CDMA) approaches.[24] Given the network time, data transmission occurs only if the transmitter ID matches the current slot index, otherwise the packet is discarded. Specifically, this mechanism emulates distributed time-synchronized access while introducing realistic clock synchronization imperfections.

The link budget is evaluated through the Received Signal Strength Indicator (RSSI) $P_{rx}$, computed in the logarithmic domain:

$$P_{rx} = P_{tx} + G_{tx} + G_{rx} - L_s \tag{36}$$

where $P_{tx}$ is the transmitted power, while $G_{tx}$ and $G_{rx}$ are the transmitter and receiver antenna gains respectively, determined by the satellites' relative line of sight and their inertial attitudes. The term $L_s$ represents the free space losses, computed as a function of the relative distance $b$ and the carrier wavelength $\lambda$:

$$L_s = 20\ log_{10}\left(\frac{4\pi b}{\lambda}\right) \tag{37}$$

The system employs multiple toroidal antennas mounted on the satellites to achieve almost omnidirectional coverage. Subsequently, the model validates the link by checking for receiver saturation and Signal-to-Noise Ratio (SNR) threshold against thermal noise.

The model accounts for data corruption by evaluating the Bit Error Rate (BER) as a function of the SNR.[24] Furthermore, packet payload integrity is managed through a dictionary-based structure that enforces specific bit-resolution and saturation constraints for each field. This mechanism simulates quantization effects while ensuring that data remain within defined operational bounds. Whenever the total bit count exceeds the allocated packet size, the payload is rejected as an overflow error.

Finally, total communication latency $T_{lat}$ is modeled as the sum of propagation delay, hardware processing, and transmission time, expressed as:

$$T_{lat} = \frac{d}{c} + \tau_{hw} + \tau_{cpu} + \frac{B_{pkt}}{R_{data}} \tag{38}$$

where $c$ is the speed of light, $\tau_{hw}$ represents the cumulative hardware delay of the transceiver pair, $\tau_{cpu}$ is the computational processing latency, and the last term accounts for the serialization delay given the packet size $B_{pkt}$ and the selected data rate $R_{data}$.

The parameters and their respective values used to model the ISL system are detailed in Table 3.

**Table 3. Inter-Satellite Link (ISL) Simulation Parameters**

| Parameter | Value |
|---|---|
| Frequency (GHz) | 2.3 (S-band) |
| Tx power, $P_{tx}$ (mW) | 250 |
| Antenna gain, $G_{peak}$ (dBi) | 2.5 |
| Antenna FOV (deg) | 60 |
| Bandwidth, $B$ (MHz) | 10 |
| Hardware delay, $\tau_{hw}$ (s) | $\sim 1 \times 10^{-7}$ |
| CPU delay, $\tau_{cpu}$ (s) | $\sim 1 \times 10^{-2}$ |
| Time slot (s) | 0.6 |
| Data rate, $R_{data}$ (Mbps) | 1 |
| Max packet size, $B_{pkt}$ (bits) | 4096 |

## Filters Initialization and Tuning

The initial uncertainties of local states and the constant ones of the auxiliary measurements are summarized in Table 4. The initial state estimates are generated by taking a sample given the associated distribution around the ground-truth. The same methodology is applied to generate all the auxiliary measurements.

**Table 4. Filter Initialization and Auxiliary Measurements Uncertainties**

| Parameter | Uncertainty ($3\sigma$) |
|---|---|
| *Local State Variables* | |
| Relative position at $t_0$ (m) | 1 |
| Relative velocity at $t_0$ (m/s) | 0.01 |
| Relative attitude error at $t_0$ (rad) | $1 \times 10^{-2}$ |
| Relative angular velocity at $t_0$ (rad/s) | $1 \times 10^{-3}$ |
| *Auxiliary Measurements* | |
| Target absolute position (m) | 100 |
| Target absolute velocity (m/s) | 0.5 |
| Absolute inertial attitude error (rad) | $1 \times 10^{-3}$ |
| Absolute inertial angular velocity (rad/s) | $1 \times 10^{-4}$ |
| CDGNSS inter-satellite baseline (m) | 0.10 |

The process noise covariance matrix $\mathcal{Q}$ is mathematically formulated assuming a Continuous-White Noise Acceleration (CWNA) model for both translational and rotational dynamics. Given a generic continuous-time power spectral density matrix $\boldsymbol{Q}_{c,i}$, the corresponding $6 \times 6$ discrete block $\boldsymbol{Q}_i$ is derived via analytical integration over the sampling interval $\Delta t$:

$$\boldsymbol{Q}_i = \begin{bmatrix} \frac{\Delta t^3}{3}\boldsymbol{Q}_{c,i} & \frac{\Delta t^2}{2}\boldsymbol{Q}_{c,i} \\ \frac{\Delta t^2}{2}\boldsymbol{Q}_{c,i} & \Delta t \boldsymbol{Q}_{c,i} \end{bmatrix} \tag{39}$$

The final process noise matrix $\boldsymbol{\mathcal{Q}}$ is then assembled as a larger block-diagonal matrix containing the respective $\boldsymbol{Q}_i$ blocks for each dynamically uncoupled state component. Specifically, the diagonal terms for translation and rotation $\boldsymbol{Q}_{c,i}$ matrices are $5 \times 10^{-10}$ $\text{m}^2/\text{s}^3$ and $1 \times 10^{-13}$ $\text{rad}^2/\text{s}^3$.

Finally, the measurement noise covariance matrix $\boldsymbol{\mathcal{R}}$ dynamically adapts based on the active observation mode. When the filter operates in the tightly coupled mode, $\boldsymbol{\mathcal{R}}$ is constructed as a block-diagonal matrix, where each block corresponds to the noise variance of a tracked 2D feature and is assigned a constant value of 6 $\text{pix}^2$. Conversely, when operating in the loosely coupled mode, $\boldsymbol{\mathcal{R}}$ is derived empirically a priori by evaluating the sample covariance of the translation and rotation errors produced by the EPnP-RANSAC solver under simulated operational conditions. To account for worst-case noise realizations, these covariances are conservatively inflated using scale factors of 1.5 and 2.2 for the translational and rotational components, respectively.

## SIMULATION RESULTS

The numerical results and performance of the proposed distributed frameworks are presented in this section. To validate and compare the architectures, the assessment is divided into two main parts. First, a comparative analysis based on a single run is performed to investigate the impact of CDGNSS baseline measurements and architectural choices. Following this, a Monte Carlo (MC) analysis is presented to evaluate the robustness and the statistical consistency of the filters.
For all the simulations, chaser 1 and chaser 2 acquire optical measurements at frequencies of 0.1 Hz and 0.07 Hz, respectively, over half the target's orbital period. For sake of conciseness, results related to velocities are omitted.

### Impact of Architecture and Baseline Measurements

The inclusion of inter-agent baseline measurements and the structural choice of the navigation architecture profoundly impact the state estimation accuracy.

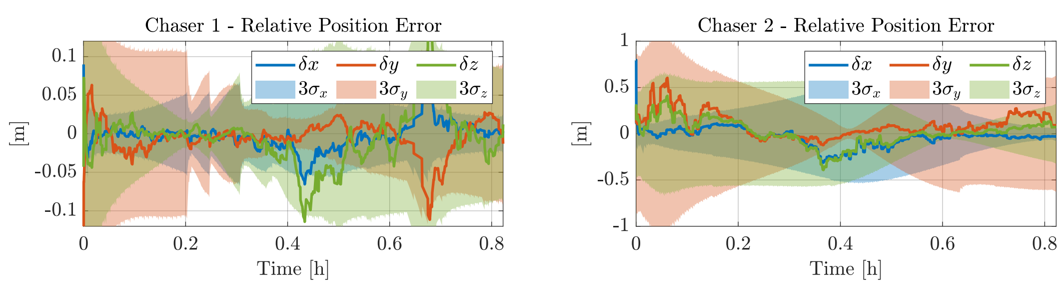


**Figure 4. Results of Centralized Joint-State Filter operating without CDGNSS measurements.**

To provide a reference point, the centralized filter is first evaluated without incorporating CDGNSS baseline measurements. As depicted in Figure 4, this configuration yields low positional accuracy due to distinct geometric challenges faced by each spacecraft. Because of its closer proximity to the target, Chaser 1 suffers from a reduced number of visible features, negatively impacting both the quality and quantity of information extracted from its observations. Conversely, Chaser 2 operates at a greater relative distance, which exacerbates the noise levels inherent in the EPnP optical measurements, leading to larger positional uncertainty.

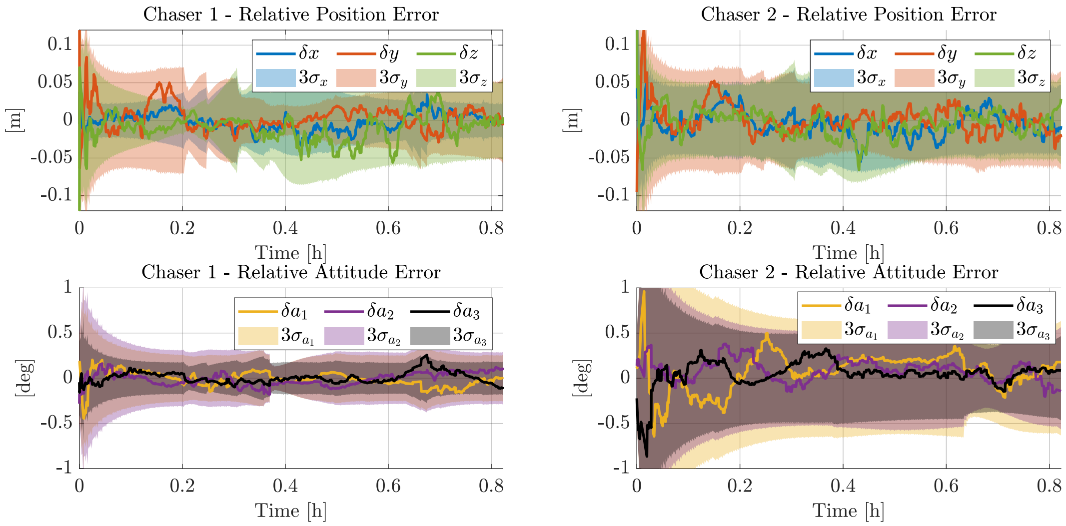


**Figure 5. Results of Centralized Joint-State Filter operating with CDGNSS measurements.**

Figure 5 shows the position and attitude error estimates when the CDGNSS baseline is integrated as a direct measurement within the centralized filter. In this configuration, the positional accuracy improves significantly, particularly for Chaser 2. Despite this translational enhancement, the centralized solution still exhibits low precision in estimating the relative attitude error for the secondary agent. This limitation arises because the centralized joint-state formulation lacks a direct observational link between the attitude states of the two chasers, unlike the CDGNSS baseline which explicitly correlates their positions.

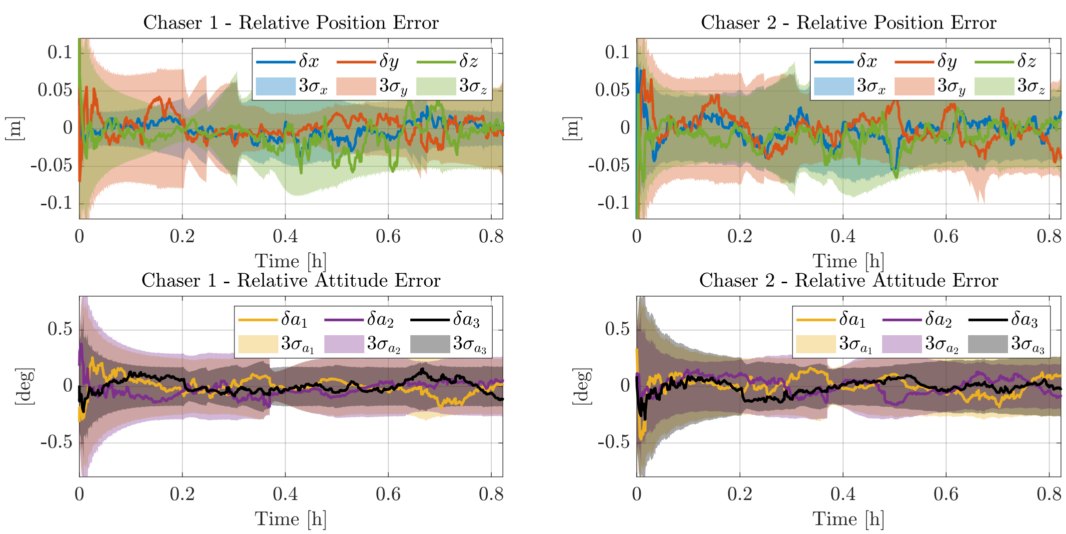


**Figure 6. Results of Decentralized Architecture.**

Finally, the results of the decentralized architecture are reported in Figure 6. This approach successfully mitigates the aforementioned shortcomings: by locally fusing remote optical observations and mapping them into the local state, the decentralized framework establishes the necessary correlations. This coupling yields a marked improvement in the rotational estimation accuracy while maintaining highly accurate translational estimates, effectively compensating for limited feature visibility of Chaser 1 end mitigating the higher measurement noise affecting Chaser 2.

### Navigation Robustness and Consistency

To comprehensively evaluate the robustness and statistical consistency of the proposed frameworks, a MC campaign is conducted (500 samples). The initial states, auxiliary measurements, and optical realizations are randomized for each run according to the uncertainties defined in Table 4.

Figures 7 and 8 display the temporal evolution of the relative position and attitude errors across all realizations for the centralized and decentralized architectures, respectively. Both frameworks demonstrate robust statistical consistency. Indeed, for each state component, the formal $3\sigma$ bounds estimated by the filters, denoted by three times the Root Mean Variance (3RMV), closely track the empirical $3\sigma$ bounds, represented by three times the Root Mean Square Error (3RMSE). The minor discrepancies between these metrics indicate a slightly conservative filter behavior. This underconfidence suggests that the measurement noise matrix $\mathcal{R}$ could be further optimized. Moreover, neither architecture exhibits divergence, confirming that the outlier-rejection and time-management algorithms successfully handle measurement delays and sporadic optical failures.

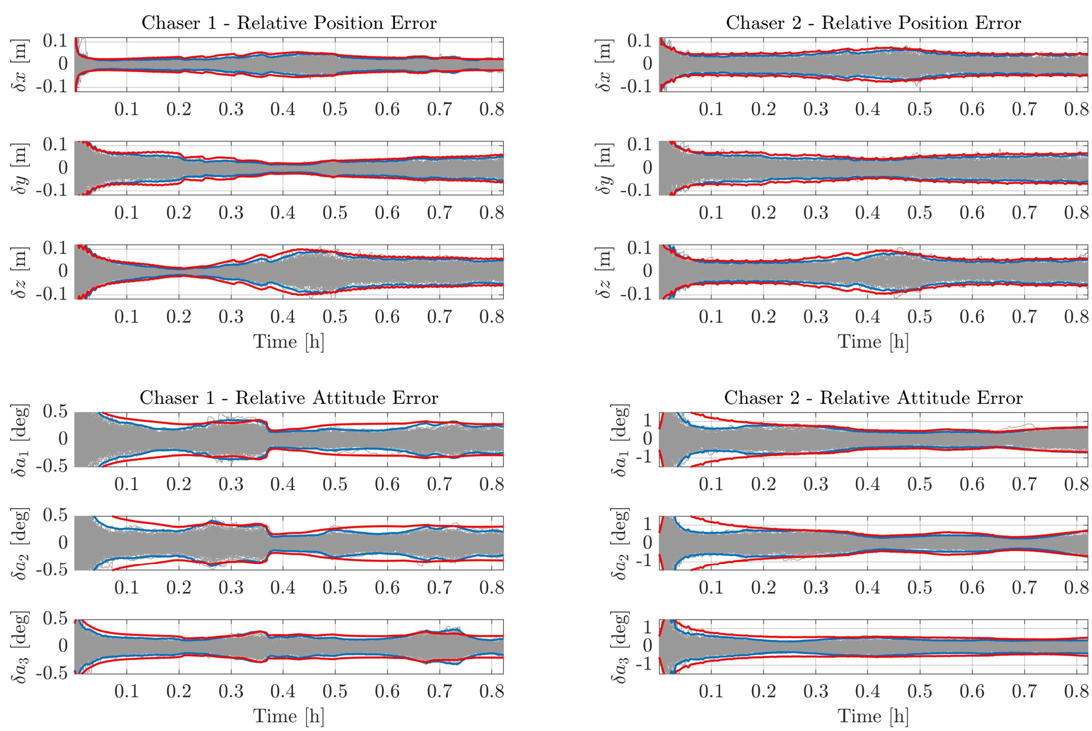


**Figure 7. MC results of Centralized Joint-State Filter. Gray lines represent error realizations, red lines indicate the estimated $3\sigma$ bounds (3RMV), and blue lines denote empirical $3\sigma$ bounds (3RMSE).**

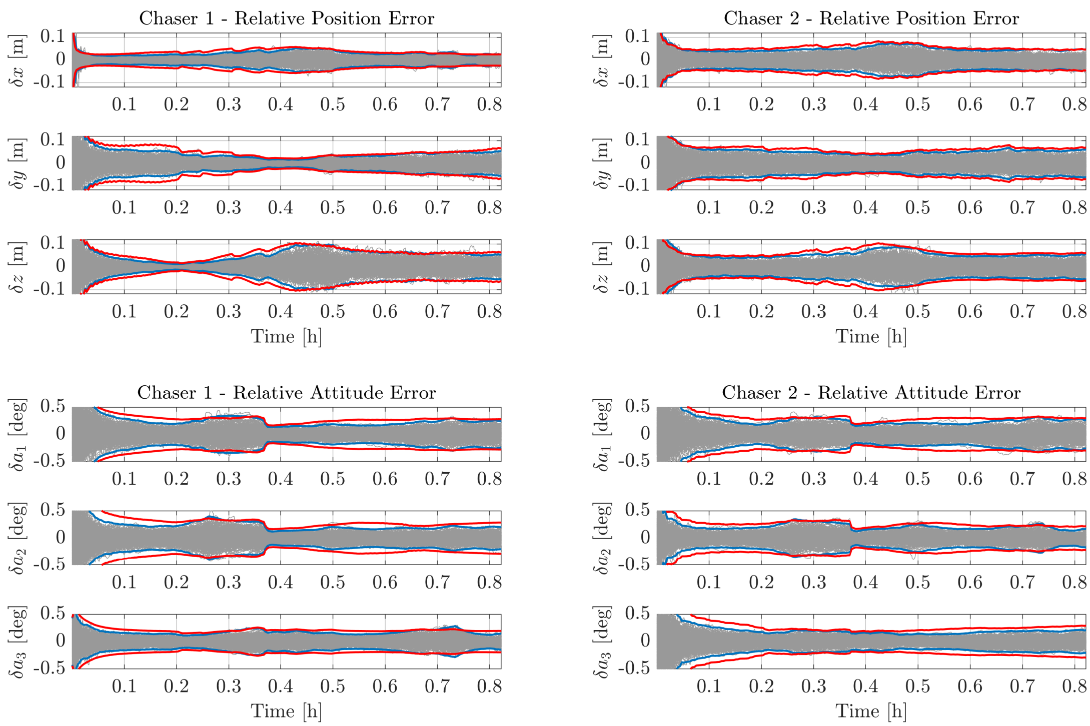


**Figure 8. MC results of Decentralized Architecture. Gray lines represent error realizations, red lines indicate the estimated $3\sigma$ bounds (3RMV), and blue lines denote empirical $3\sigma$ bounds (3RMSE).**

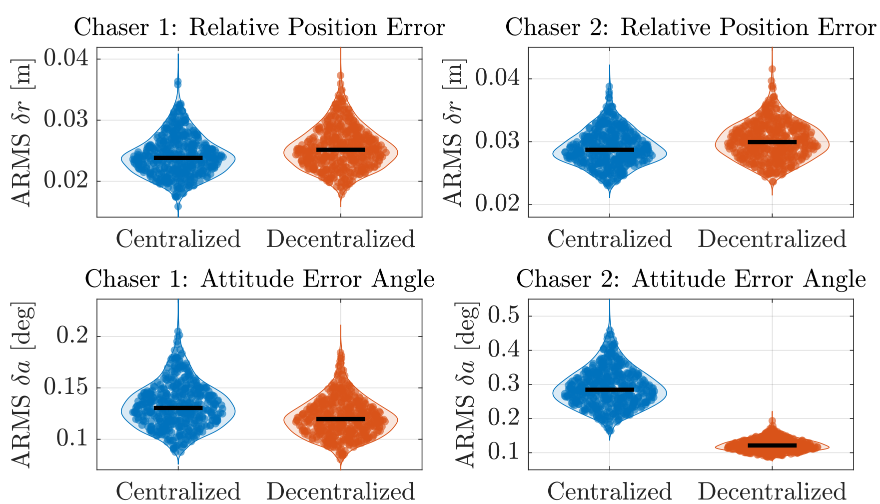


**Figure 9. Violin plots with the ARMSE of state variables. Black lines denote the ARMSE median.**

To better quantify the overall estimation accuracy across the entire simulated campaign, Figure 9 presents violin plots of the temporal Average Root Mean Square Error (ARMSE) of each state variable for both architectures. The translational performance remains largely comparable between the two frameworks. Indeed, the centralized filter shows only a marginal statistical advantage for

Chaser 2, a direct result of integrating the CDGNSS baseline as a direct measurement. However, the operational value of the decentralized architecture becomes evident in the rotational estimates. By mathematically mapping remote optical observations into the local state, the decentralized framework improves the relative attitude estimation for Chaser 2, reducing the median ARMSE by over $50\%$ and significantly narrowing its statistical dispersion. Ultimately, these results confirm that the decentralized architecture not only eliminates the single point of failure inherent to centralized systems but also provides superior robustness for multi-agent navigation.

## CONCLUSIONS

This study investigates two different frameworks for relative navigation during multi-agent proximity operations around uncooperative targets: a centralized joint-state filter and a decentralized architecture, which are designed to effectively manage communication delays and measurement outliers. To validate these approaches, high-fidelity optical and ISL models were implemented. The results demonstrate that the centralized solution improves translation accuracy by directly integrating CDGNSS measurements of the chasers' relative position. Conversely, the proposed decentralized architecture proves superior by enhancing the robustness of attitude estimation and eliminating the single point of failure inherent to centralized nodes.

Future developments will focus on the validation of the decentralized architecture across a broader range of operational scenarios. Furthermore, subsequent research will evaluate whether incorporating other cooperative agents into the distributed system can further improve state estimation accuracy. Another area of focus will be managing the double-counting of information to enable the exchange of processed estimates rather than raw measurements. This approach ensures that an agent can cooperate with the system without requiring explicit knowledge of the other agents' measurement models. Finally, future iterations will introduce range and range-rate measurements derived from the ISL system. These parameters will be investigated to operate either as supporting measurements or as direct substitutes for the chasers' 3D baseline measurements, ultimately enabling the system to autonomously operate in GNSS-denied environments.

## ACKNOWLEDGMENT

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.

## REFERENCES

[1] M. K. Ben-Larbi, K. F. Pozo, T. Haylok, M. Choi, B. Grzesik, A. Haas, D. Krupke, H. Konstanski, V. Schaus, S. P. Fekete, *et al.*, "Towards the automated operations of large distributed satellite systems. Part 1: Review and paradigm shifts," *Advances in Space Research*, Vol. 67, No. 11, 2021, pp. 3598–3619.

[2] S. D'Amico, M. Pavone, S. Saraf, A. Alhussien, T. Al-Saud, S. Buchman, R. Byer, and C. Farhat, "Miniaturized autonomous distributed space system for future science and exploration," *8th International Workshop on Satellite Constellations and Formation Flying*, International Astronautical Federation's Astrodynamics Committee Delft, The . . . , 2015, pp. 8–10.

[3] D. J. Barnhart, T. Vladimirova, and M. N. Sweeting, "Very-small-satellite design for distributed space missions," *Journal of Spacecraft and Rockets*, Vol. 44, No. 6, 2007, pp. 1294–1306.

[4] M. Sabatini, R. Volpe, and G. Palmerini, "Centralized visual based navigation and control of a swarm of satellites for on-orbit servicing," *Acta Astronautica*, Vol. 171, 2020, pp. 323–334.

[5] K. Dennison, N. Stacey, and S. D'Amico, "Autonomous asteroid characterization through nanosatellite swarming," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 59, No. 4, 2023, pp. 4604–4624.

[6] N. Stacey and S. D'Amico, "Autonomous swarming for simultaneous navigation and asteroid characterization," *AAS/AIAA Astrodynamics Specialist Conference*, Vol. 1, 2018, p. 76.

[7] K. Matsuka, E. S. Lupu, Y. K. Nakka, R. Foust, S.-J. Chung, and F. Hadaegh, "Distributed multi-target relative pose estimation for cooperative spacecraft swarm," 2019.

[8] J. J. Kruger, T. Guffanti, T. H. Park, M. Murray-Cooper, S. Y. Low, T. Bell, S. D'Amico, C. W. Roscoe, and J. Westphal, "Adaptive End-to-End Architecture for Autonomous Spacecraft Navigation and Control During Rendezvous and Proximity Operations," *AIAA SCITECH 2024 Forum*, 2024, p. 0430.

[9] M. Capitanio, M. Maestrini, M. Massari, *et al.*, "Distributed Navigation Architecture for Close-Range Target Inspection," *8th CEAS Specialist Conference on Guidance, Navigation and Control-EuroGNC*, 2026, pp. 1–19.

[10] F. L. Markley, "Attitude error representations for Kalman filtering," *Journal of guidance, control, and dynamics*, Vol. 26, No. 2, 2003, pp. 311–317.

[11] K. Alfriend, S. R. Vadali, P. Gurfil, J. How, and L. Breger, *Spacecraft formation flying: Dynamics, control and navigation*, Vol. 2. Elsevier, 2009.

[12] F. L. Markley and J. L. Crassidis, "Attitude kinematics and dynamics," *Fundamentals of Spacecraft Attitude Determination and Control*, pp. 67–122, Springer, 2014.

[13] F. L. Markley and J. R. Carpenter, "Generalized linear covariance analysis," *The Journal of the Astronautical Sciences*, Vol. 57, No. 1, 2009, pp. 233–260.

[14] W. G. Cochran, "The $\chi 2$ test of goodness of fit," *The Annals of mathematical statistics*, 1952, pp. 315–345.

[15] P. Sturm, "Pinhole camera model," *Computer Vision: A Reference Guide*, pp. 983–986, Springer, 2021.

[16] T. Möller and B. Trumbore, "Fast, Minimum Storage Ray-Triangle Intersection," *Journal of Graphics Tools*, Vol. 2, No. 1, 1997, pp. 21–28.

[17] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[18] S. K. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Transactions on Pattern analysis and machine intelligence*, Vol. 16, No. 8, 1994, pp. 824–831.

[19] V. Peretroukhin, L. Clement, and J. Kelly, "Get to the point: Active covariance scaling for feature tracking through motion blur," *Pan*, Vol. 1, 2015, p. 0.

[20] D. B. Goldman, "Vignette and exposure calibration and compensation," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 32, No. 12, 2010, pp. 2276–2288.

[21] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O (n) solution to the PnP problem," *International journal of computer vision*, Vol. 81, No. 2, 2009, pp. 155–166, 10.1007/s11263-008-0152-6.

[22] K. G. Derpanis, "Overview of the RANSAC Algorithm," *Image Rochester NY*, Vol. 4, No. 1, 2010, pp. 2–3.

[23] R. Sun, D. Maessen, J. Guo, and E. Gill, "Enabling inter-satellite communication and ranging for small satellites," *9th Symposium on Small Satellites Systems and Services, Funchal, Portugal*, Vol. 31, 2010.

[24] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 4, 2016, pp. 2442–2473.