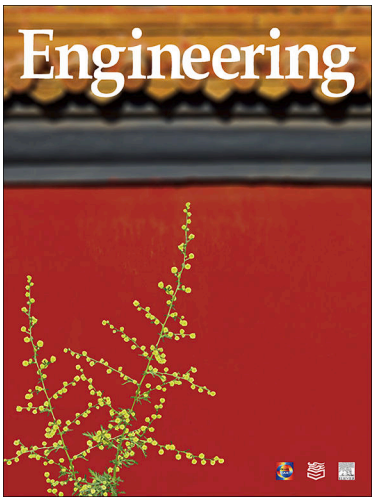
Journal Pre-proofs



Article

Behavior-Inspired Event-Triggered Control of Spacecraft Swarms for Autonomous Capture of Uncooperative Targets

Zixuan Zhang, Zheng H. Zhu, Dan Zhang

PII: S2095-8099(26)00253-5
DOI: https://doi.org/10.1016/j.eng.2026.04.014
Reference: ENG 2332

To appear in: *Engineering*

Received Date: 21 April 2026

Please cite this article as: Z. Zhang, Z.H. Zhu, D. Zhang, Behavior-Inspired Event-Triggered Control of Spacecraft Swarms for Autonomous Capture of Uncooperative Targets, *Engineering* (2026), doi: https://doi.org/10.1016/j.eng.2026.04.014

Behavior-Inspired Event-Triggered Control of Spacecraft Swarms for Autonomous Capture of Uncooperative Targets

Zixuan Zhang[1], Zheng H. Zhu[1,*], Dan Zhang[2]

[1] Department of Mechanical Engineering, York University, 4700 Keele Street, Toronto, Ontario, Canada

[2] Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract

This paper presents an event-triggered behavior-inspired decentralized control framework for a spacecraft swarm tasked with capturing an unknown, uncooperative target. The proposed approach integrates local swarm behaviors, event-triggered communication and control, and distributed target perception within a unified architecture. Three local behaviors are considered: inter-agent collision avoidance, target buffering, and sensor-pointing alignment. To reduce communication burden, each agent updates and broadcasts its state only when predefined local triggering conditions are satisfied. Meanwhile, onboard sensing and neighbor-to-neighbor information exchange are used to estimate the target position, motion, and geometry without prior knowledge of the target shape. A graduated non-convexity-based perception module is adopted to improve the robustness of landmark-based target reconstruction under partial observability and noisy measurements. Numerical simulations for tumbling ellipsoidal targets demonstrate that the proposed method enables autonomous search, chase, and pre-capture enclosure while maintaining safe inter-agent spacing and reducing communication compared with periodic update strategies.

Keyword: Spacecraft swarm; Active debris removal; Event-triggered; Decentralized control; Autonomous; Distributed target perception; Uncooperative target.

1. Introduction

The accelerating accumulation of orbital debris poses serious risks to operational spacecraft and the long-term sustainability of space activities, creating an urgent need for effective active debris removal (ADR) solutions [1]-[2]. ADR missions demand autonomous rendezvous, close-proximity operations, and capture of non-cooperative targets while operating under stringent sensing, communication, and actuation limitations. Most existing robotic ADR approaches depend on a single servicer spacecraft and typically assume some prior knowledge of the target's geometry or dynamics [3]-[4]. In practice, however, single-spacecraft architecture offers limited sensing coverage and reduced fault tolerance, which motivates the development of swarm-based multi-spacecraft ADR systems.

Prior studies have examined both centralized and decentralized control methods for spacecraft swarms. Centralized approaches can produce coordinated group behavior, but they

* Corresponding author: gzhu@yorku.ca

are constrained by poor scalability, communication bottlenecks, and vulnerability to single-point failures [5]. By contrast, decentralized methods, especially those influenced by swarm robotics and biological collectives [6], generate coordinated behavior through local interactions, thereby improving robustness and scalability [7]-[8]. In particular, behavior-inspired control [6] has become a promising paradigm for spacecraft swarms, enabling agents to accomplish collective tasks such as formation maintenance, collision avoidance, and target encirclement through simple local rules. Nevertheless, many of these methods depend on continuous or high-rate communication, which is impractical for space systems with limited bandwidth and onboard computational capability. To address this issue, event-triggered control and communication [9] have attracted growing interest as a means of reducing unnecessary updates while maintaining stability and coordination performance [10]. Despite these advances, most existing studies concentrate on formation control or cooperative tracking of known targets, and they do not address decentralized perception together with behavior-based swarm coordination for the capture of unknown targets.

To address these gaps, this paper develops a fully decentralized, event-triggered, behavior-inspired control framework for spacecraft swarms operating around unknown, uncooperative targets. The proposed framework integrates swarm-level local interaction rules with agent-level event-triggered communication and translational control, so that each agent updates its control input and broadcasts information only when locally defined triggering conditions are satisfied. In this manner, unnecessary communication and onboard computation are reduced while coordinated pursuit and enclosure behavior is maintained. In parallel, a distributed perception module based on onboard landmark sensing, neighbor-to-neighbor information fusion, and a graduated non-convexity (GNC) algorithm [11] is employed to estimate the target centroid, motion, and approximate geometry without requiring prior knowledge of the target shape.

The main contributions of this work are fourfold: (1) the development of a unified decentralized architecture that combines behavior-based swarm coordination, event-triggered communication and control, and distributed target perception for unknown-target operations; (2) the design of a self-organizing local behavior set that enforces inter-agent separation, target buffer-zone protection, and sensor-pointing alignment to support safe pre-capture enclosure; (3) the introduction of a distributed landmark-based perception pipeline, integrated with graduated non-convexity, for online estimation of target position, motion, and geometry under partial observability and noisy measurements; and (4) numerical simulations with tumbling ellipsoidal targets showing that the proposed method scales up to larger swarms, adapts to different target shapes, and significantly reduces communication relative to periodic update schemes.

2. Problem Formulation

2.1 Mathematical Model

Consider a spacecraft swarm and an uncooperative target (debris), as illustrated in Fig. 1. It is assumed that the target is in a circular orbit and the relative separation between the target and the swarm agents is negligible compared with the agents' orbital radius. Define the Earth-Centered Inertial (ECI) frame, $E_c = \{O_c, x_c, y_c, z_c\}$, whose origin is fixed at the Earth's center, the x_c-axis points towards the vernal equinox, the z_c-axis is aligned with the Earth's rotation axis (North Pole), and the y_c-axis is defined to complete a right-handed coordinate frame. The

motion of each spacecraft is described relative to the target using a local-vertical local-horizontal (LVLH) frame, $E_i = \{O_i, x_i, y_i, z_i\}$ [12]. The origin of this frame is placed at the target's center of mass, with the x_i-axis directed radially outward from the Earth's center, the y_i-axis oriented along the orbital path, and the z_i-axis pointing in the cross-track direction, consistent with the orbital angular momentum.

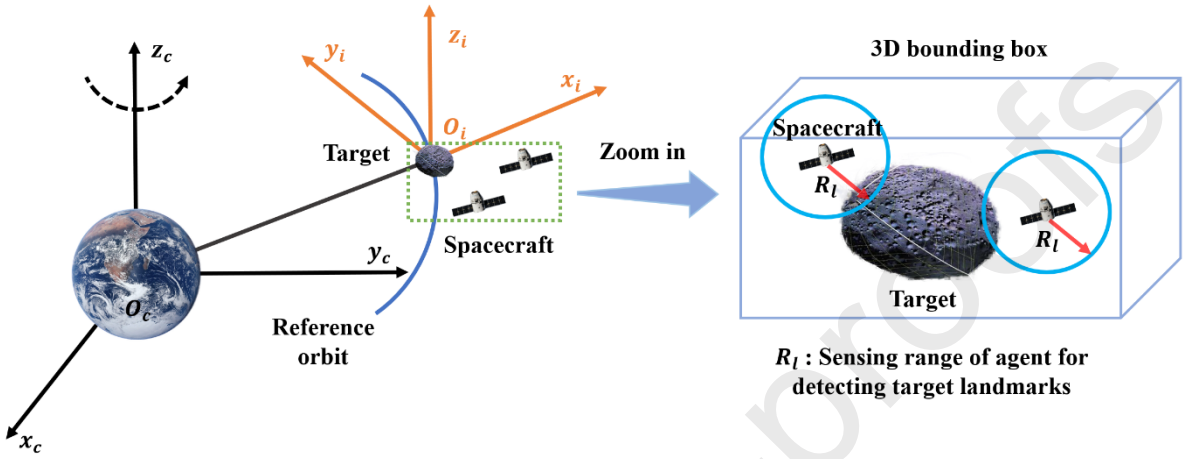


Fig. 1 Coordinate frames and close-up view of the spacecraft sensing ranges.

2.1.1 Spacecraft Agent Relative Motion Dynamics

Translational dynamics

Consider a swarm consisting of N spacecraft agents, denoted by $A_i, i \in \{1, \ldots, N\}$, operating in orbital space. All agents are assumed to have identical dynamic characteristics, sensing capabilities, and communication limitations. The translational motion of the i-th agent relative to the target is described by the Clohessy-Wiltshire (CW) equations [13], expressed as

$$\begin{cases} \dot{r}_i = v_i \\ \dot{v}_i = A_{cw} r_i + B_{cw} v_i + F_i / m_i \end{cases} \tag{1}$$

$$A_{cw} = \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix}, B_{cw} = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2}$$

where $r_i = [x_i \quad y_i \quad z_i]^T \in \mathbb{R}^3$ and $v_i = [\dot{x}_i \quad \dot{y}_i \quad \dot{z}_i]^T \in \mathbb{R}^3$ denote the relative position and velocity of agent i with respect to the target. Here, F_i represents the control force exerted by the agent, and m_i is its mass. The matrices A_{cw} and B_{cw} represent the gravity-gradient terms and Coriolis coupling associated with relative orbital motion. The quantity n denotes the mean angular velocity of the LVLH frame.

Attitude dynamics

Each agent is modeled as a rigid body with inertia matrix $J_i \in \mathbb{R}^{3\times3}$. Its attitude dynamics is governed by Euler's equation

$$J_i\dot{\omega}_i = \tau_i - \omega_i \times (J_i\omega_i) \tag{3}$$

where $\omega_i = [\omega_{i1}, \omega_{i2}, \omega_{i3}]^T \in \mathbb{R}^3$ is the angular velocity of the agent expressed in its body-fixed frame and τ_i is the control torque applied in the body frame.

The orientation of the *i*-th agent with respect to the LVLH frame is represented by the unit quaternion $q_i = [q_{i0}, q_{iv}^T]^T \in \mathbb{R}^4$ satisfying $\|q_i\| = 1$, where q_{i0} is the scalar part. The quaternion is related to its angular velocity by

$$\dot{q}_i = \frac{1}{2}\Omega(\omega_i)q_i \tag{4}$$

where $\Omega(\omega_i)$ is the standard quaternion multiplication matrix given by

$$\Omega(\omega_i) = \begin{bmatrix} 0 & -\omega_{i1} & -\omega_{i2} & -\omega_{i3} \\ \omega_{i1} & 0 & \omega_{i3} & -\omega_{i2} \\ \omega_{i2} & -\omega_{i3} & 0 & \omega_{i1} \\ \omega_{i3} & \omega_{i2} & -\omega_{i1} & 0 \end{bmatrix} \tag{5}$$

Compact state-space form

Define the state vector of agent i as $X_i = [r_i^T, v_i^T, q_i^T, \omega_i^T]^T \in \mathbb{R}^{13}$ and u_i as the corresponding control input for the agent i

$$u_i = \begin{bmatrix} F_i \\ \tau_i \end{bmatrix} \in \mathbf{R}^{6\times1} \tag{6}$$

where F_i and τ_i represent the control force and control torque applied to agent i, which is generated by the onboard actuator.

Then, the relative motion of agent i can be expressed in the following control-affine form,

$$\dot{X}_i = f(X_i) + G_iu_i + D_id_i \tag{7}$$

where

$$f(X_i)=\begin{bmatrix} v_i \\ 0_3 \\ (1/2)\Omega(\omega_i)q_i \\ -J_i^{-1}\left[\omega_i\times(J_i\omega_i)\right] \end{bmatrix},\ G_i=\begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ (1/m_i)I_3 & 0_{3\times3} \\ 0_{4\times3} & 0_{4\times3} \\ 0_{3\times3} & J_i^{-1} \end{bmatrix},\ D_i=\begin{bmatrix} 0_{3\times3} \\ I_3 \\ 0_{4\times3} \\ 0_{3\times3} \end{bmatrix} \tag{8}$$

Here, $d_i=A_{cw}r_i+B_{cw}v_i$ represents the orbital coupling term induced by the CW relative motion dynamics. For robust control design, this term is treated as a bounded additive disturbance acting on the translational acceleration channel and satisfying $\|d_i\|\le\|A_{cw}\|\|r_i\|+\|B_{cw}\|\|v_i\|=3n^2\|r_i\|+2n\|v_i\|$. For example, consider a representative low-Earth-orbit scenario with $r_{\max}$ = 100m and $v_{\max}$ = 0.1m/s, the disturbance yields a disturbance magnitude on the order of 10^{-4} to 10^{-3} m/s^2, depending on the orbital mean motion n.

2.1.2 Target Tumbling Kinematics

The target is assumed to be significantly larger than the agents and is modeled as a rigid body freely floating and tumbling in space. Its geometry is represented by a set of landmarks L_i detectable by agent i, where each landmark corresponds to a geometric feature defining the target's shape and size. The target's rotational motion is described by the angular velocity ω_T^L of the target principal reference frame (TPRF) with respect to the LVLH frame, expressed in the LVLH frame [13]. Agents detect these landmarks and exchange such information with neighboring agents to estimate the target's relative position and velocity in the frame E_i.

To characterize the motion of target surface features, consider a point k on the target surface, representing either a surface feature or a contact point. The position of this point in the LVLH frame is denoted by r_k, obtained by transforming sensor measurements from the agent reference frames into the LVLH frame. Since the origins of the LVLH frame and the TPRF coincide, the linear velocity of point k in the LVLH frame is given by the rigid-body kinematic relation

$$v_k=\omega_T^L\times r_k \tag{9}$$

These expressions describe the motion of target surface features and contact points induced by the target tumbling motion and are used by the agents to track moving features and perform velocity matching during the capture phase. Furthermore, a three-dimensional bounding box is defined to restrict the permissible region of agent motion, such that

$$S=\left\{(x_i,y_i,z_i):x_{\min}\le x_i\le x_{\max},y_{\min}\le y_i\le y_{\max},z_{\min}\le z_i\le z_{\max}\right\} \tag{10}$$

Finally, three local behaviors are defined to support decentralized coordination of the swarm:

1. Local communication

Each agent communicates only with its neighboring agents that lie within its

communication range.

2. Collision avoidance and separation

Safe separation is maintained through inter-agent repulsive actions derived from the negative gradient of an artificial potential field [14], together with an additional buffer-zone potential that prevents agents from approaching the uncooperative target too closely.

3. Pointing behavior

Each agent directs its sensor toward detected target landmarks by reducing attitude and angular velocity tracking error through a proportional-derivative (PD) controller.

Collectively, these decentralized behaviors allow the swarm to coordinate autonomously, avoid collisions, and maintain reliable sensing performance in uncertain orbital environments.

2.2 Local Communication Protocol

Assume that each agent functions within a decentralized communication architecture based on local interactions, sharing information only with neighboring agents located inside its communication range R_c. Information such as target-detection events is transmitted through neighbor-to-neighbor relays, allowing the swarm to adapt collectively without requiring any individual agent to have global knowledge of the entire group [15]. Communication delays are neglected in the current work.

In addition, each agent is further assumed to possess a sensing range R_l for detecting target landmarks and a collision awareness range R_h for identifying nearby obstacles or neighboring agents [16]. These ranges are assumed to satisfy the relation $R_h \le R_l \le R_c$.

2.3 Collision Avoidance Mechanism

2.3.1 Inter-Agent Separation

To avoid collisions between agents, mutual separation is enforced through repulsive forces derived from an artificial potential function U_{ij}, defined as in [17]

$$U_{ij}\left(r_i, r_j\right) = \begin{cases} \dfrac{1}{2} k_r \left(\dfrac{1}{d_{ij}^n} - \dfrac{1}{R_h} \right)^2, & d_{ij}^n \le R_h \\ 0, & d_{ij}^n > R_h \end{cases} \tag{11}$$

where $k_r > 0$ denotes the repulsion gain, R_h represents the maximum collision awareness range among agents, and $d_{ij} = \left\| r_i - r_j \right\| = \sqrt{\left(r_i - r_j\right)^T \cdot \left(r_i - r_j\right)}$ is the Euclidean distance between agent i and j. The repulsive force exerted on the agent i by the agent j is obtained from the negative gradient of the potential function with respect to the position of the agent i:

$$F_{ij}^{rep} = -\nabla_{r_i} U_{ij}\left(r_i, r_j\right) = \begin{cases} k_r n\left(\dfrac{1}{d_{ij}^n} - \dfrac{1}{R_h}\right)\dfrac{r_i - r_j}{d_{ij}^{n+2}}, & d_{ij}^n \le R_h \\ 0, & d_{ij}^n > R_h \end{cases} \tag{12}$$

By Newton's third law, the agent j experiences the equal and opposite force $F_{ji}^{rep} = -F_{ij}^{rep}$. The total collision avoidance force on the agent i is the sum over all neighbors within range:

$$F_i^{col} = \sum_{j \neq i, d_{ij} \le R_h} F_{ij}^{rep} \tag{13}$$

2.3.2 Agent-Target Buffering

Since the target is both unknown and uncooperative, the agents have no prior knowledge of its physical characteristics and are therefore unable to model detailed contact interactions or define direct repulsive forces. Accordingly, a protective buffer zone R_{buffer} is introduced around the estimated target boundary to ensure safe approach during the pursuit and enclosure phases.

Let $x_{s,i}$ denote the closest point on the estimated target surface to the agent i, expressed in the LVLH frame. This point is a particular target surface point as described in Section 2.1.2, and its motion is induced by the target tumbling kinematics. The outward unit normal at $x_{s,i}$ is denoted by n_i. The corresponding safe offset point on the buffer shell is defined as

$$x_{g,i} = x_{s,i} + R_{buffer} n_i \tag{14}$$

Based on this local geometric information, a repulsive action is activated whenever the agent approaches the protected region, and additional damping may be introduced near the boundary to suppress inward motion and reduce oscillatory behavior.

2.4 Pointing Alignment

Accurate perception requires each agent to maintain its sensor's field of view oriented toward detected target landmarks, which is particularly important for camera- or LIDAR-based systems. When attitude control capability is available, each agent aligns a selected body-fixed axis with the direction of the landmark using locally defined control laws [6]. Pointing behavior is activated only after the chase mode is triggered, to reduce unnecessary energy consumption during the search phase.

For the attitude subsystem, a quaternion-based nonlinear computed-torque controller is adopted. Let the current attitude quaternion be q_i, and q_{id} denote the desired attitude quaternion computed at each timestep by aligning the agent's body-fixed x-axis with the direction vector from the agent toward the estimated target centroid $\hat{x}_{T,i}$.

During chase mode, the desired pointing direction is the line-of-sight vector toward the freshest estimated target centroid,

$$s_i = \hat{x}_{T,i} - x_i \tag{15}$$

The desired quaternion is obtained via the shortest-arc rotation from $[1,0,0]^T$ to s_i.

Then the attitude tracking error is defined by the error quaternion q_{ie}

$$q_{ie} = q_{id}^{*} \otimes q_i \tag{16}$$

where $\otimes$ denotes quaternion multiplication and q_{id}^{*} is the conjugate of q_{id}. The sign ambiguity of the unit quaternion is resolved by enforcing its scalar part $q_{ie0} \geq 0$. If $q_{ie0} < 0$, then q_{ie} is negated to ensure the shortest rotational path is tracked. The angular velocity tracking error is defined as

$$e_{\omega_i} = \omega_i - \omega_{id} \tag{17}$$

where ω_{id} is the desired angular velocity expressed in the body frame. For fixed-attitude stabilization tasks, $\omega_{id} = 0$ is adopted, which gives $e_{\omega_i} = \omega_i$.

Based on the above dynamics, the control torque is designed as [18]

$$\tau_i = \omega_i \times (J_i \omega_i) - K_{q_i} q_{iev} - K_{\omega_i} e_{\omega_i} \tag{18}$$

where K_{q_i} and K_{ω_i} are positive definite feedback gain matrices. The first term compensates for the gyroscopic coupling in the rigid-body rotational dynamics, while the second and third terms provide proportional attitude correction and angular-rate damping, respectively.

Substituting the control law into the attitude dynamics yields

$$J_i \dot{\omega}_i + \omega_i \times (J_i \omega_i) = \omega_i \times (J_i \omega_i) - K_{q_i} q_{iev} - K_{\omega_i} e_{\omega_i} \tag{19}$$

Therefore, the closed-loop dynamics reduce to

$$J_i \dot{\omega}_i = -K_{q_i} q_{iev} - K_{\omega_i} e_{\omega_i} \tag{20}$$

This closed-loop form has a standard damped second-order structure, in which the quaternion error term acts as a restoring component and the angular velocity feedback provides damping. As a result, the nonlinear coupling term is explicitly canceled, which improves the transient response and facilitates gain tuning, especially in aggressive rotational maneuvers or tumbling target tracking scenarios. Combined with the separation and buffering behaviors, this pointing control enables coordinated target tracking and enclosure.

3. Event-triggered Decentralized Collaborative Swarm Control

In contrast to conventional continuous or time-triggered coordination schemes, the proposed strategy enables each agent to update its control action and communicate with

neighboring agents only when specified local triggering events occur, thereby reducing unnecessary computational and communication burdens [19]. Within this framework, each agent autonomously operates in one of three behavioral modes: search, chase, and dock. Transitions between these modes are governed by explicitly defined mission events.

Event 1: Target Detection

When an agent detects the uncooperative target within its sensing range, it switches from the search mode to the chase mode and simultaneously initiates an event-triggered broadcasting to its neighboring agents.

Event 2: Target Information Reception

When an agent reaches the boundary of the search region, it is redirected inward by a randomized thrust applied at the boundary, then transitions to the chase mode and joins the cooperative pursuit and localization process through local information exchange.

Event 3: Target Perception Completion

Once the swarm has collectively obtained a reliable estimate of the target's position and geometric features, the agents enter the dock mode and carry out coordinated engagement with the target.

Event 4: Event-Triggered Control Update

During the chase and dock modes, each agent adopts an event-triggered mechanism for both the translational PD controller and local state broadcasting. State information is transmitted only when the local deviation exceeds a prescribed threshold.

The overall event-triggered decentralized control architecture for collaborative swarm capture of an unknown and uncooperative target is illustrated in Fig. 2.

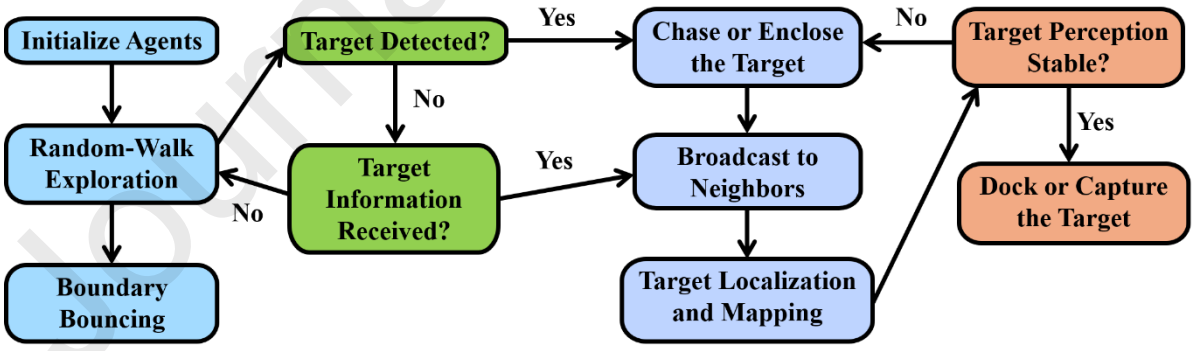


Fig. 2 Event-triggered decentralized swarm control pipeline.

3.1 Random Walk Exploration with Energy-Constrained Bouncing Maneuver

To mitigate the risk that individual agents may fail to detect the target, the search mode incorporates a stochastic exploration mechanism. In this phase, each agent follows a random walk strategy combined with a boundary-bouncing maneuver to prevent it from drifting beyond the prescribed search region. When an agent reaches the boundary of the search region, it is redirected inward by a randomized thrust applied at the boundary,

$$F_{bnc} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T \tag{21}$$

with

$$f_x = \begin{cases} -F_{thr}, & x > x_{\max} \\ F_{thr}, & x < x_{\min} \\ 0, & otherwise \end{cases}, \; f_y = \begin{cases} -F_{thr}, & y > y_{\max} \\ F_{thr}, & y < y_{\min} \\ 0, & otherwise \end{cases}, \; f_z = \begin{cases} -F_{thr}, & z > z_{\max} \\ F_{thr}, & z < z_{\min} \\ 0, & otherwise \end{cases} \tag{22}$$

where $F_{rnd} = \alpha F_{thr} \cdot rand(1)$, $F_{thr} > 0$ denotes a constant thrust magnitude, α is a scaling factor associated with the stochastic perturbation, and $rand(1)$ generates a random number in the interval [0, 1]. The resulting term F_{rnd} represents the stochastic control input applied to the agent.

To improve energy efficiency, a limited boundary penetration is permitted to balance fuel consumption against strict enforcement of the search-region constraint.

3.1.1 Penetration depth

For a specified outward normal velocity at the instant of boundary crossing $v_n > 0$, the penetration depth under a constant inward thrust is given by $\delta(v_n) = m v_n^2 / (2F_{thr})$. To ensure that $\delta(v_n) \le \delta_{\max}$ for all $v_n \le v_{\max}$, the thrust must satisfy $F_{thr} \ge m v_{\max}^2 / (2\delta_{\max})$, where $\delta_{\max}$ is a tunable design parameter. This relationship indicates a trade-off: reducing the allowable penetration requires a larger corrective thrust and, consequently, greater fuel expenditure, and vice versa.

In summary, during the search mode, the translational control input is composed of the inter-agent collision-avoidance force and the boundary-bouncing thrust, whereas the rotational control input is provided by the pointing torque that maintains the desired sensor orientation.

$$u_{i,search} = \begin{bmatrix} F_{bnc} + F_i^{col} \\ \tau_{pnt} \end{bmatrix} \in \mathbf{R}^{6\times 1} \tag{23}$$

3.2 Chase and Enclose the Target

Once the target is detected, agents switch from the search phase to the chase phase. This transition is regulated by a decentralized event-triggered PD controller [19], which coordinates both agent motion and local state broadcasting among neighboring agents. Agents that either directly detect the target or receive target related information initiate the localization and mapping process, refining target geometry estimates and propagating them through neighbor-to-neighbor communication. The control input in the chase phase is formulated as follows:

$$u_{i,chase} = \begin{bmatrix} F_{i,chase} + F_i^{col} \\ \tau_{pnt} \end{bmatrix} \in \mathbf{R}^{6\times 1} \tag{24}$$

3.2.1 Event-Triggered Broadcasting Strategy

Given the inherent limitations in inter-satellite communication bandwidth and onboard power availability, it is necessary to adopt a control framework that minimizes both computational and communication demands. In this context, event-triggered strategies are particularly suitable for decentralized coordination in resource-constrained multi-agent systems [19]. Accordingly, an event-triggered broadcasting mechanism is integrated with the decentralized PD controller.

The chase-mode control law for the agent i is expressed as follows:

$$F_{i,chase} = -K_p\left(r_i - r_T^i\right) - K_v\left(v_i - v_T^i\right) \tag{25}$$

where K_p and K_v are positive control gains selected to yield a damped convergent response.

Based solely on locally accessible information, each agent determines whether its current state should be transmitted to neighboring agents. Let $r_i\left(t_k^i\right)$ denote the most recently broadcast position of the agent i, where t_k^i represents the *k*-th triggering instant of the agent i. An analogous definition is used for the velocity state.

During the interval between two successive triggering instants, t_k^i and t_{k+1}^i, the control input is held constant using the latest transmitted state:

$$F_{i,chase}\left(t\right) = -K_p[r_i(t_k^i) - r_T^i(t_k^i)] - K_v[v_i(t_k^i) - v_T^i(t_k^i)] \tag{26}$$

The corresponding sampling errors in position and velocity are then defined as $\hat{r}_i(t) = r_i(t_k^i) - r_i(t)$ and $\hat{v}_i(t) = v_i(t_k^i) - v_i(t)$. Based on these errors, the event-triggering condition in the LVLH frame is formulated as

$$f_i\left(t, \hat{r}_i(t), \hat{v}_i(t)\right) > 0 \tag{27}$$

$$f_i\left(t, \hat{r}_i(t), \hat{v}_i(t)\right) = \left\| \begin{bmatrix} \hat{r}_i(t) \\ \hat{v}_i(t) \end{bmatrix} \right\| - \left(c_0 + c_1 e^{-\beta t} + c_d d_{\max}\right) \tag{28}$$

where c_0, c_1, β, and c_d are prescribed design parameters. Following [19], the parameters are selected as $c_0 = 0.05$, $c_1 = 0.2499$, $\beta = 0.1$, $c_d = 1.0$.

Consequently, the triggering-time sequence for the agent i is recursively determined by

$$t_{k+1}^i = \inf\left\{t : t > t_k^i, f_i\left(t, \hat{r}_i(t), \hat{v}_i(t)\right) > 0\right\} \tag{29}$$

3.2.2 Unknown Target Localization and Mapping

Since neither the shape nor the size of the target is known a priori, each agent uses onboard sensors (e.g., LIDAR) to detect target landmarks and infer the target state from locally available measurements. Once landmarks are detected, the agent updates its estimate of the

target centroid and motion using the visible landmark set. To improve robustness under noisy measurements, partial observability, and possible outliers, a GNC [11] algorithm is employed to extract a consistent local target representation from the sensed landmarks. This representation is then used to update the target centroid estimate $\hat{x}_{T,i}$, velocity estimate $\hat{v}_{T,i}$, and, when available, geometric parameters describing the local target shape.

When direct measurements are unavailable, agents exploit local communication and update their target information using neighboring estimates received within the communication range. In general form, the distributed update is given by

$$\hat{x}_{T,i} = \sum_{j \in N_i} w_{ij} \hat{x}_{T,j}, \quad \hat{v}_{T,i} = \sum_{j \in N_i} w_{ij} \hat{v}_{T,j} \tag{30}$$

where $w_{ij} > 0$ and $\sum_{j \in N_i} w_{ij} = 1$. The same decentralized fusion principle can be extended to any geometric parameters estimated from the target observations. In this way, the swarm collectively reconstructs target position, motion, and approximate geometry through local sensing and neighbor-to-neighbor information exchange, without requiring global knowledge of the swarm or prior knowledge of the target shape.

3.3 Dock or Capture the Target

During the chase phase, the agents perform target detection and state broadcasting while progressively forming a hovering enclosure around the target, thereby establishing a pre-capture configuration for the subsequent docking stage. Once the target position and relevant characteristics have been estimated with sufficient reliability, the swarm transitions to the final docking phase for capture. During this phase, the agents continue to exchange information to coordinate their approach, synchronize their motions, and update their trajectories in real time, thereby ensuring cohesive collective operation and reducing the risk of collision.

4. Simulation and Discussion

The proposed event-triggered decentralized swarm control framework is verified via numerical simulations on a laptop with an AMD Ryzen 5 4600H processor (3.00 GHz) and 16GB of RAM. The simulation time step is 0.05 *s*, and each run lasts 550 *s*. The agent parameters are defined based on a representative 8U CubeSat platform: m_i = 10 kg, $\tau_{\max}$ = 0.01 *Nm*, and $F_{\max} = F_{thr}$ = 0.1 *N*. The target translational velocity and angular velocity are v_T = [0.01, 0.01, 0.0] m/s and ω_T = 0.72 deg/s. Other parameters are listed in Table 1.

Table 1. Numerical simulation parameters

Symbol	Parameter	Value
FOV	Field of view angle	75°

K_{q_i}	Attitude proportional gain	diag([1.8, 1.8, 1.8])
K_{ω_i}	Attitude derivative gain	diag([0.25, 0.25, 0.25])
K_p	Translational proportional gain	0.5
K_v	Translational derivative gain	1.2
k_r	Inter-agent repulsion gain	8.0
$k_{r,target}$	Repulsive gain between agent & target	0.4
R_{buffer}	Buffering zone around the target (m)	1.0
R_c	Communication range (m)	9.5
R_h	Collision awareness range (m)	5.5
R_l	Sensing range for landmarks (m)	8.5

4.1 Simulation Scenarios

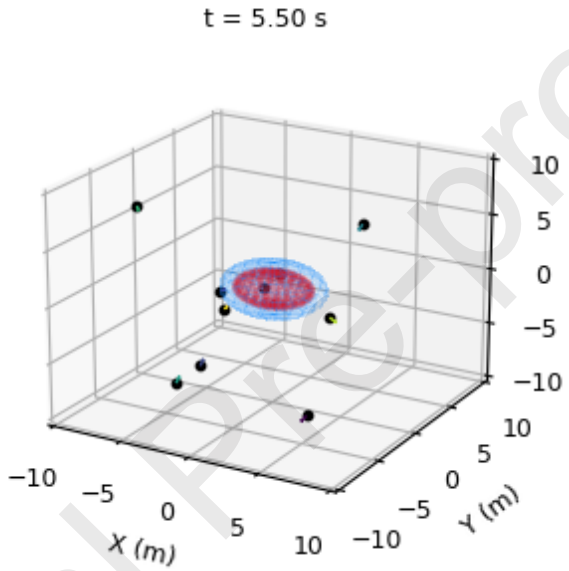
All agents in the swarm share identical physical and control parameters. Two representative mission scenarios were designed as follows:

(i) Ten agents are randomly initialized within a 3D space of [-10, 10]×[-10, 10]×[-10, 10] m, with a minimum inter-agent separation of 2 m. The target is an ellipsoidal body with semi-axes $[2.6, 1.9, 1.4]^T$ m. It is initially at $[0.0, 0.0, 0.0]^T$ m and moves with constant velocity $[0.01, 0.01, 0.0]^T$ m/s. The target also tumbles about the normalized axis $[0.2, 0.1, 1.0]^T$, with initial attitude quaternion $[1.0, 0.0, 0.0, 0.0]^T$.

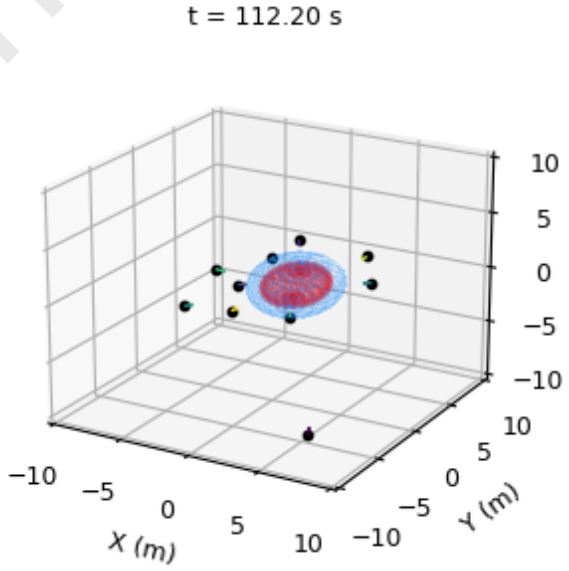
(ii) Twenty agents are randomly initialized within [-10, 10]×[-10, 10]×[-10, 10] m, with a minimum inter-agent separation of 2 m. The target is an ellipsoidal body with semi-axes $[4.0, 1.3, 1.0]^T$ m. It is initially at $[0.0, 0.0, 0.0]^T$ m and moves with constant velocity $[0.01, 0.01, 0.0]^T$ m/s. The target also tumbles about the normalized axis $[0.2, 0.1, 1.0]^T$, with initial attitude quaternion $[1.0, 0.0, 0.0, 0.0]^T$.

4.2 Scenario (i) – 10 agents

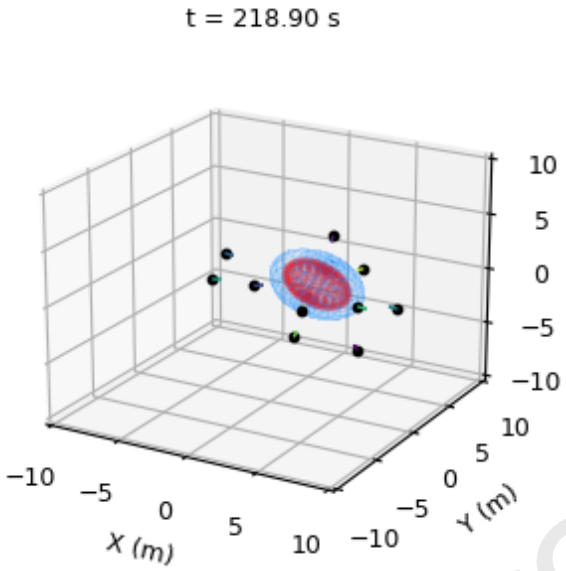
The pre-capture enclosure process for the ellipsoidal target is shown in Fig. 3. The target translates along the x- and y-directions while tumbling, and the agents gradually transition from dispersed search to target-oriented motion. In the early stage, the agents search for the target while adjusting their headings according to the sensing and pursuit requirements. As time progresses, more agents acquire the target and move toward the buffering zone surrounding the ellipsoid. The snapshots show that the swarm progressively gathers around the moving target and forms a distributed hovering pattern around its body, while maintaining safe inter-agent spacing and respecting the target buffer shell. By approximately 200–300 s, most active agents have converged to the neighborhood of the target and remain coordinated around it.



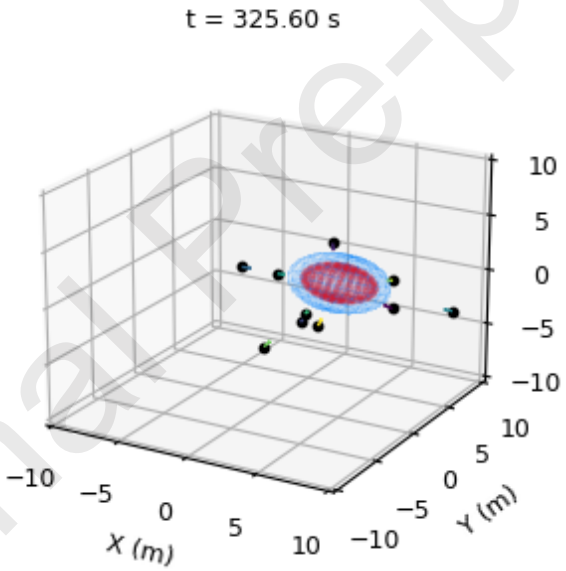
(a)



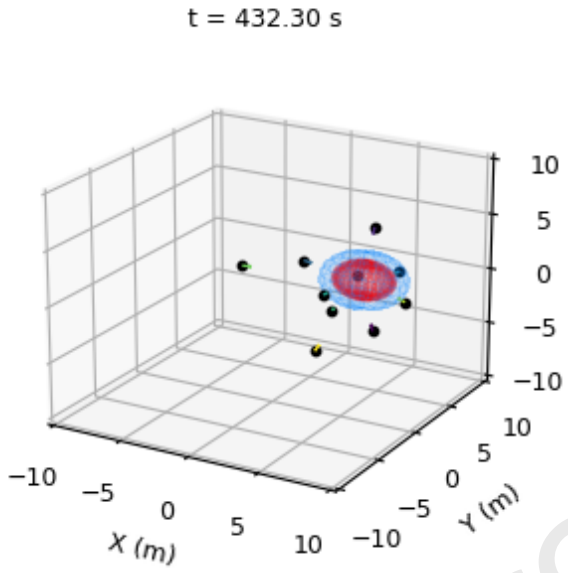
(b)



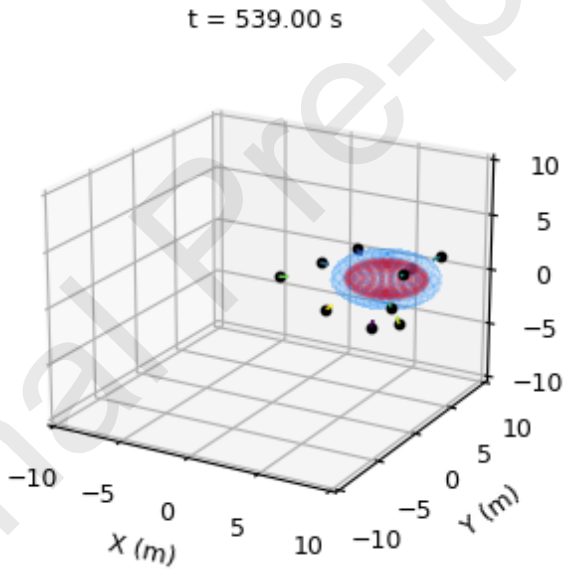
(c)



(d)



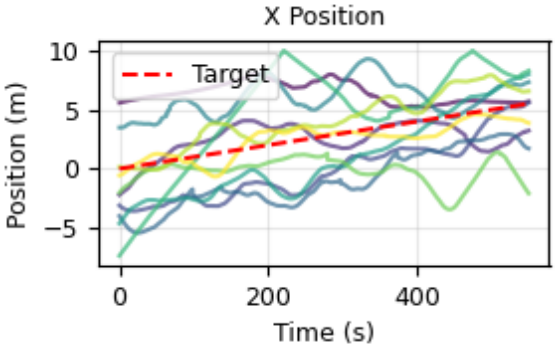
(e)

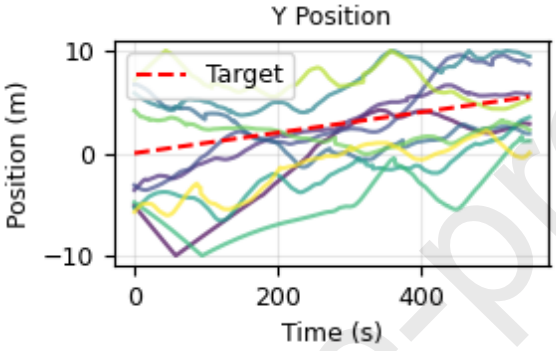


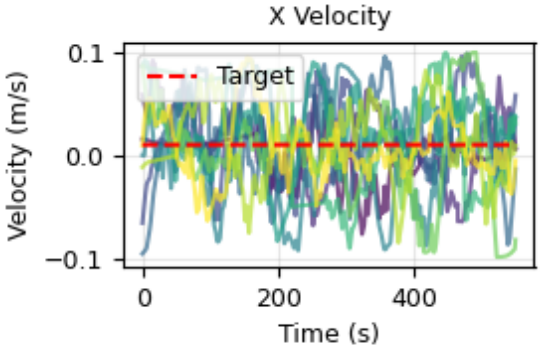
(f)

Fig. 3 Hovering of 10 agents around the ellipsoidal target.

Fig. 4 shows that the position time histories (top) of the agents progressively align with the target trajectory (red dashed line). In particular, the target drifts linearly in the x- and y-directions and remains constant in z, and most agents gradually follow this translational trend. The velocity time histories (bottom) remain bounded and show partial synchronization during the chase phase, although some dispersion persists because the ellipsoidal target induces geometry-dependent motion and the agents do not acquire the target at the same time.







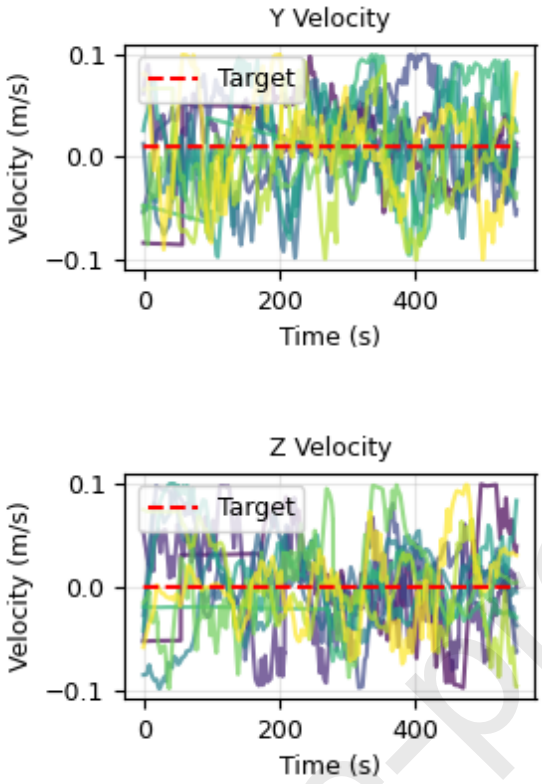
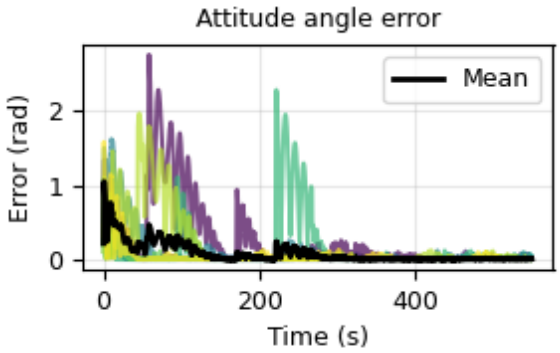


Fig. 4 Time histories of agent and target positions and velocities for the ellipsoidal target (10 agents).

Fig. 5 shows that the attitude angle error and angular velocity error are relatively large during the initial search phase, since the agents start from dispersed configurations and must rapidly adjust their pointing directions while trying to detect a translating and tumbling ellipsoidal target; however, both swarm-mean errors (black curves) decay quickly and remain small for most of the mission, indicating effective attitude tracking once pursuit is established. The occasional spikes in individual-agent curves should not be interpreted as instability, but rather as transient responses to abrupt updates in the desired pointing direction caused by target tumbling, target-estimate refreshes, and asynchronous switching from search to chase mode. Therefore, the main conclusion is that the attitude controller maintains bounded and generally low tracking errors at the swarm level.



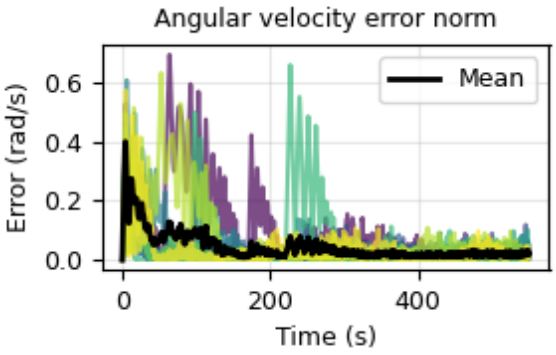


Fig. 5 Attitude tracking errors of the agents for the ellipsoidal target case (10 agents).

Fig. 6 shows the sample error norms of all agents relative to the trigger threshold (dashed line), confirming that the decentralized event-triggered strategy keeps the tracking error bounded while avoiding continuous communication. Most agents exhibit the typical event-triggered pattern, in which the error evolves close to the threshold and is reset after communication events. The different error profiles also reflect the asynchronous nature of target acquisition: agents that detect the target early generate more updates, whereas agents that join the chase later show delayed activation in their error histories. This indicates that the decentralized strategy remains effective even when the target is not acquired simultaneously by all agents.

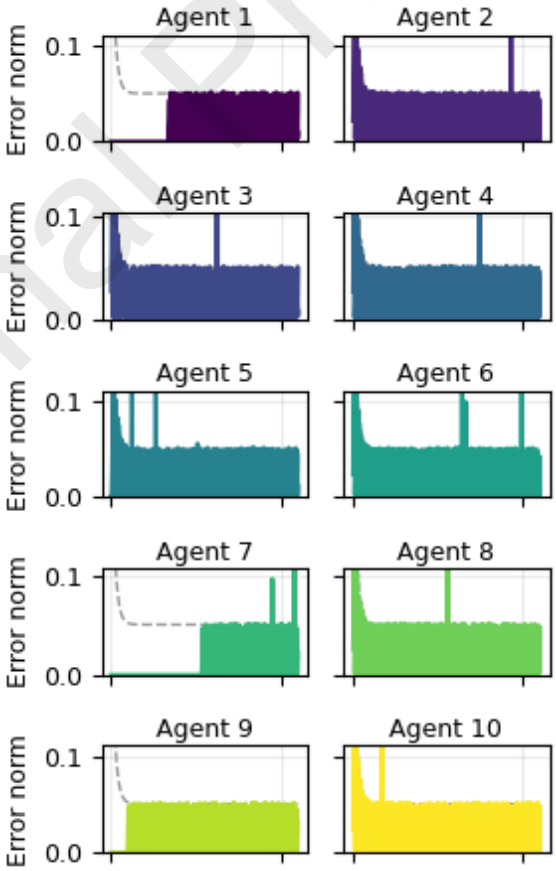


Fig. 6 Agent sample errors with trigger threshold for the ellipsoidal target case (10

agents).

Fig. 7 shows that the online GNC-based ellipsoid reconstruction remains consistently close to the ground-truth semi-axes throughout the simulation, with the estimated semi-axes staying near 2.6 m, 1.9 m, and 1.4 m, respectively. After a short initial transient, all three estimates settle rapidly and fluctuate only within a narrow band around their true values, indicating that the perception module can reliably recover the target geometry despite measurement noise, target tumbling, and decentralized sensing. Overall, this figure confirms that the proposed GNC-based shape estimation provides an accurate and stable geometric description of the moving ellipsoidal target, which in turn supports the geometry-aware buffering and tracking behavior observed in the swarm.

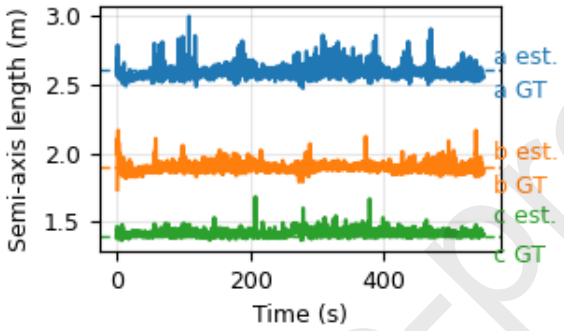
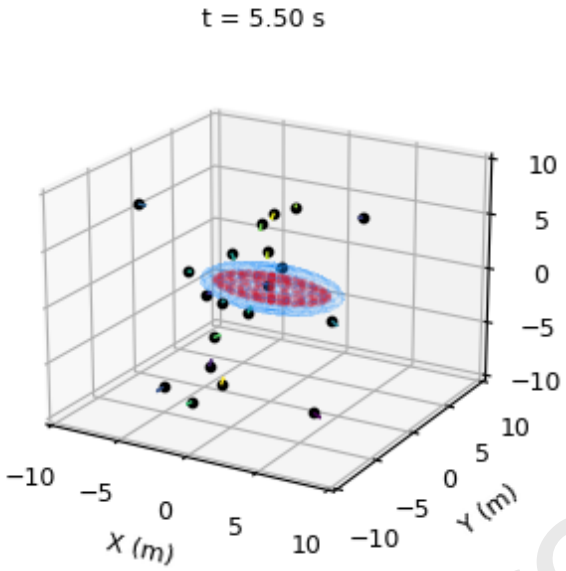


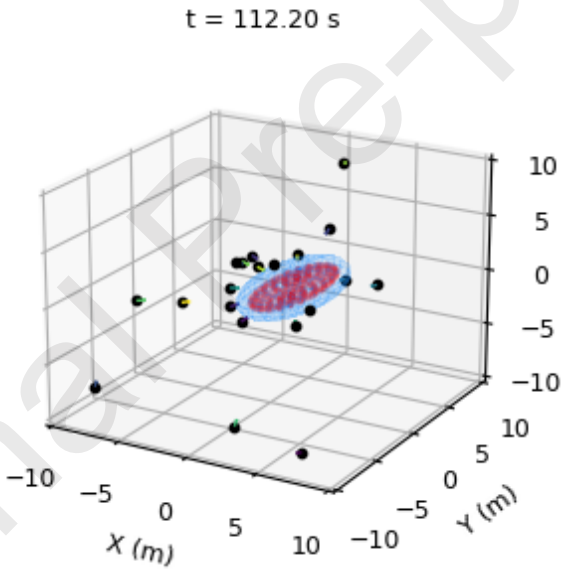
Fig. 7 GNC-based ellipsoid semi-axis estimation versus ground truth for the ellipsoidal target case (10 agents).

4.3 Scenario (ii) – 20 agents

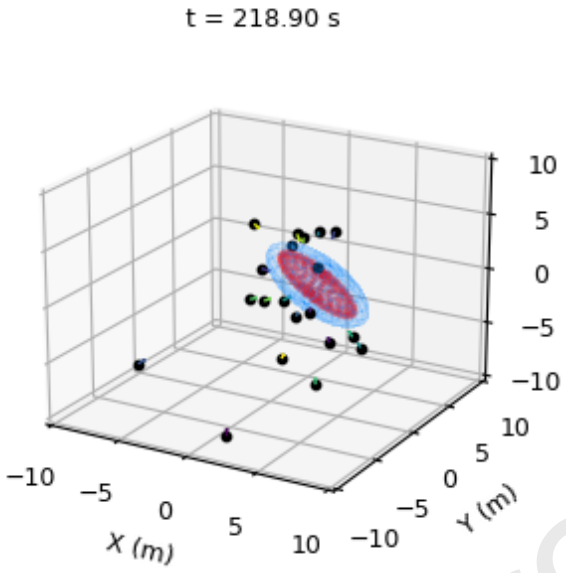
The pre-capture enclosure process for the elongated ellipsoidal target is shown in Fig. 8. The target translates along the x- and y-directions while tumbling, and the agents gradually transition from dispersed search to target-oriented motion. In the early stage, most agents search for the target, while a few agents detect it immediately and enter the chase mode. As time progresses, more agents acquire the target and move toward the buffering zone surrounding the ellipsoid. The snapshots show that the swarm progressively gathers around the moving target and forms a distributed hovering pattern around its elongated body, while maintaining safe inter-agent spacing and respecting the target buffer shell. By approximately 200–300 s, most active agents have converged to the neighborhood of the target and remain coordinated around it.



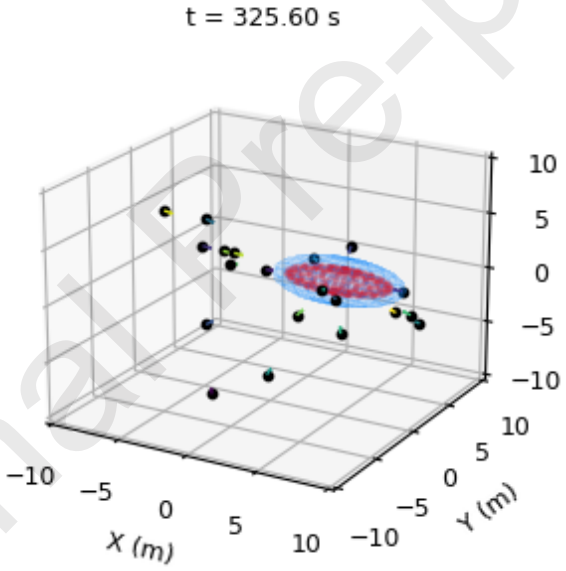
(a)



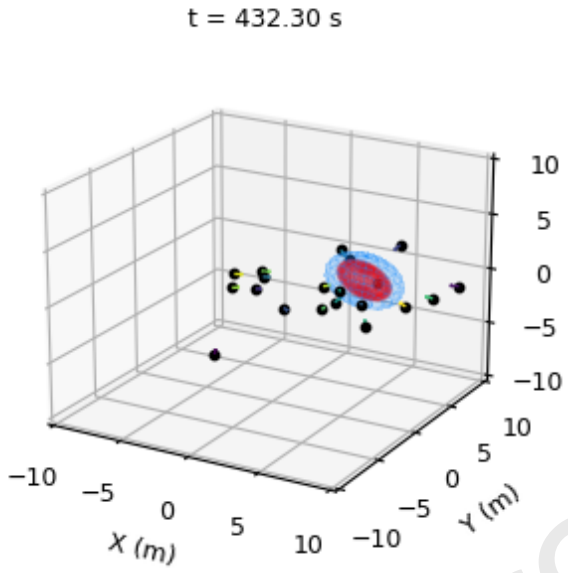
(b)



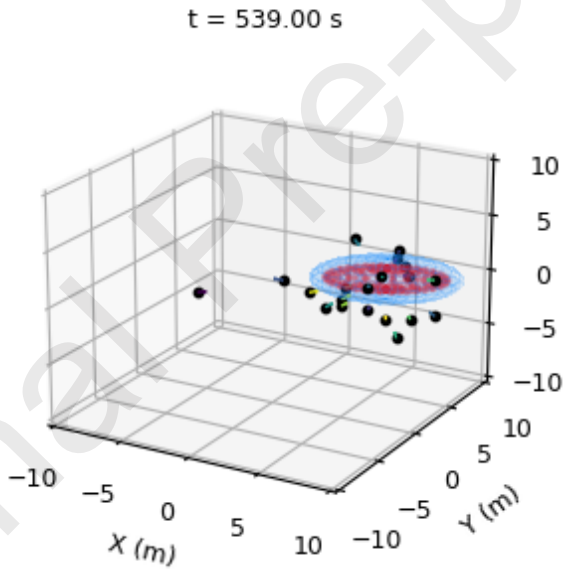
(c)



(d)



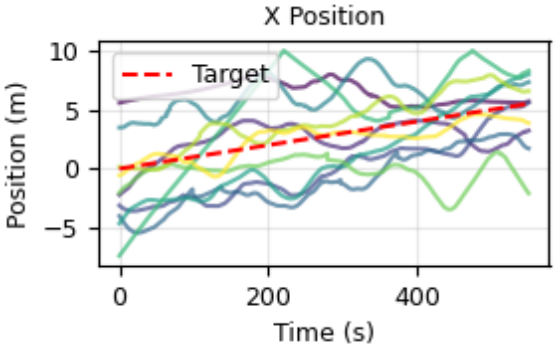
(e)

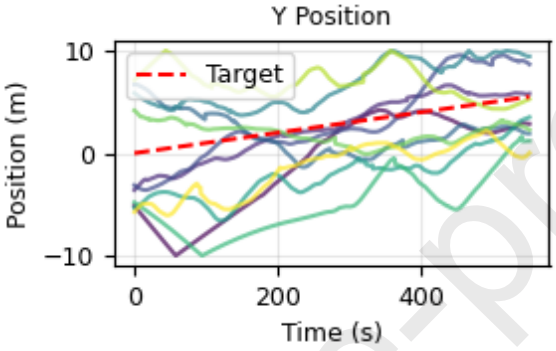


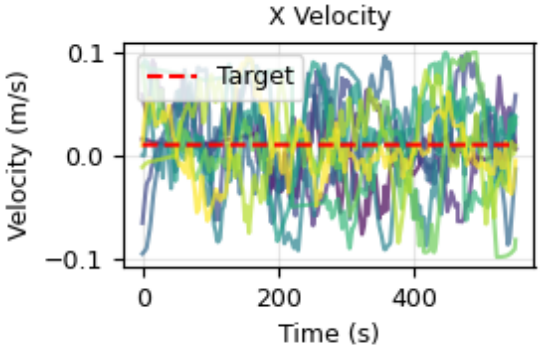
(f)

Fig. 8 Hovering of 20 agents around the elongated target.

Fig. 9 shows that the position time histories (top) of the agents progressively align with the target trajectory (red dashed line). In particular, the target drifts linearly in the x- and y-directions and remains constant in z, and most agents eventually follow this translational trend. The velocity time histories (bottom) remain bounded and show partial synchronization during the chase phase, although the spread is larger than in the 10-agent ellipsoidal-target case because the elongated target induces more complex geometry-dependent motion and some agents enter the chase mode later than others. Overall, the plots confirm that most agents achieve sustained target-following behavior under the moving and tumbling target condition.







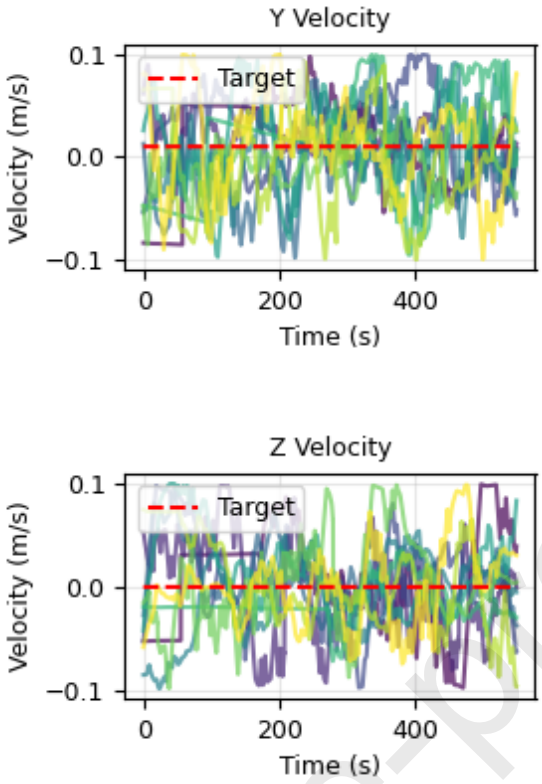
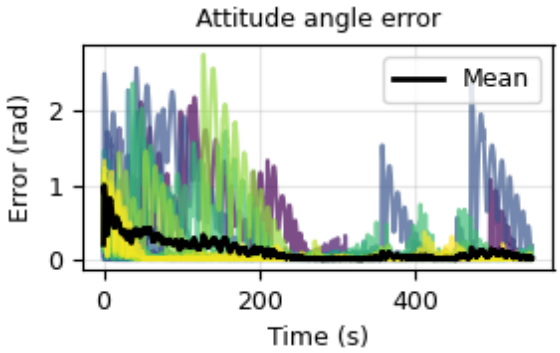


Fig. 9 Time histories of agent and target positions and velocities for the elongated target (20 agents).

Fig. 10 shows that the attitude angle error and angular velocity error are relatively large during the initial search phase, since the agents start from dispersed configurations and must rapidly adjust their pointing directions while trying to detect a translating and tumbling elongated target; however, both swarm-mean errors (black curves) decay quickly and remain small for most of the mission, indicating effective attitude tracking once pursuit is established. The occasional spikes in individual-agent curves should not be interpreted as instability, but rather as transient responses to abrupt updates in the desired pointing direction caused by target tumbling, target-estimate refreshes, and asynchronous switching from search to chase mode. Therefore, the main conclusion is that the attitude controller maintains bounded and generally low tracking errors at the swarm level.



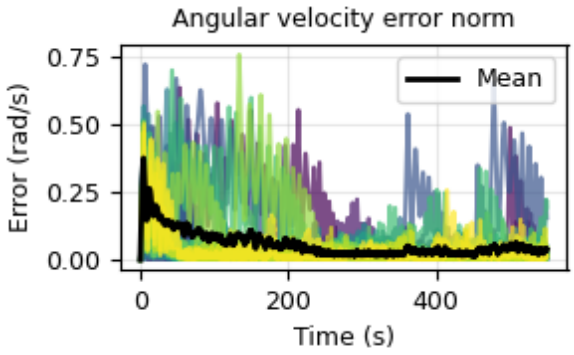


Fig. 10 Attitude tracking errors of the agents for the elongated target case (20 agents).

Fig. 11 shows the sample error norms of all agents relative to the trigger threshold (dashed line), confirming that the decentralized event-triggered strategy keeps the tracking error bounded while avoiding continuous communication. Most agents exhibit the typical event-triggered pattern, in which the error evolves close to the threshold and is reset after communication events. The different error profiles also reflect the asynchronous nature of target acquisition: agents that detect the target early generate more updates, whereas late-entry agents trigger fewer events. Notably, agent 6 did not enter the chase phase and therefore never triggered events. This indicates that the decentralized strategy remains robust even when not all agents acquire the target simultaneously.

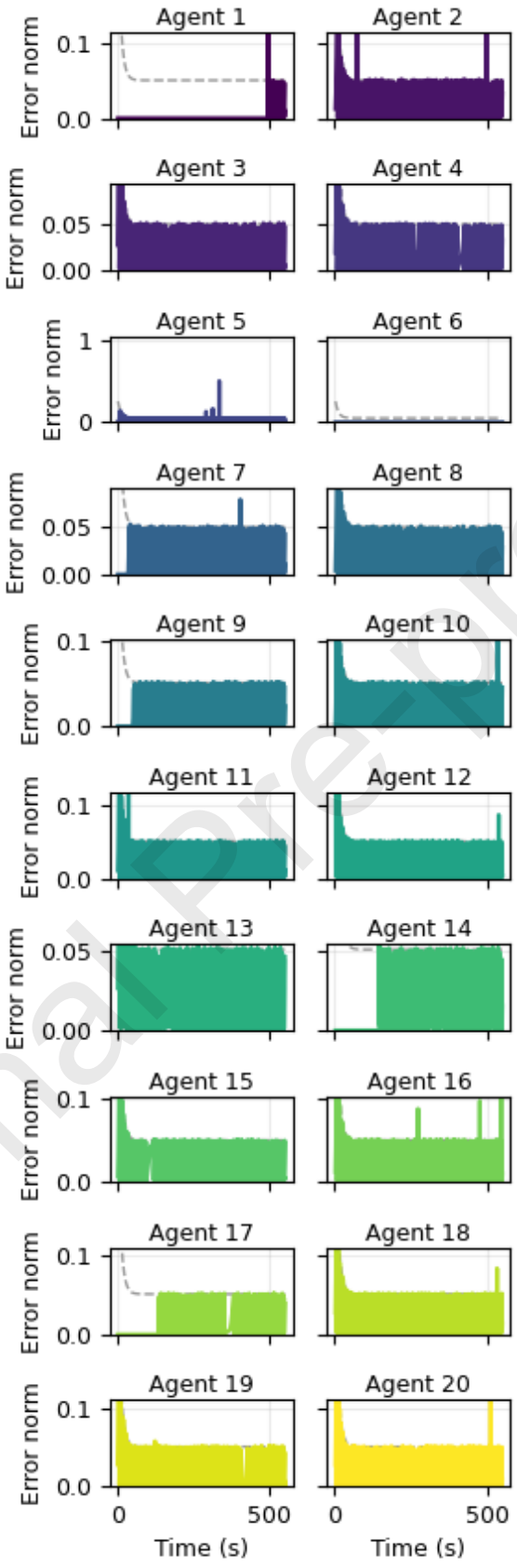


Fig. 11 Agent sample errors with trigger threshold for the elongated target case (20 agents).

Fig. 12 shows that the online GNC-based ellipsoid reconstruction remains consistently

close to the ground-truth semi-axes throughout the simulation, with the major axis estimate staying near 4.0 m and the two minor-axis estimates remaining close to 1.3 m and 1.0 m, respectively. After a short initial transient, all three estimates settle rapidly and fluctuate only within a narrow band around their true values, indicating that the perception module can reliably recover the elongated target geometry despite measurement noise, target tumbling, and decentralized sensing. Overall, this figure confirms that the proposed GNC-based shape estimation provides an accurate and stable geometric description of the moving ellipsoidal target, which in turn supports the geometry-aware buffering and tracking behavior observed in the swarm.

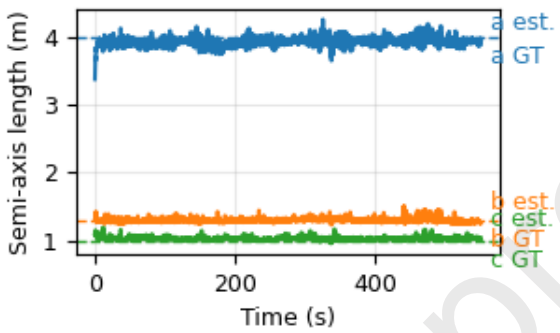


Fig. 12 GNC-based ellipsoid semi-axis estimation versus ground truth for the elongated target case (20 agents).

Declaration of competing interest

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

Acknowledgments

This work was supported by the Discovery Grant (RGPIN-2024-06290, RGPIN-2022-04624) and Collaborative Research and Training Experience Program Grant (555425-2021) of the Natural Sciences and Engineering Research Council of Canada.

References

[1] P. Zhao, J. Liu, C. Wu, Survey on research and development of on-orbit active debris removal methods, Sci. China Technol. Sci. 63 (2020) 2188–2210. https://doi.org/10.1007/s11431-020-1661-7.

[2] C.P. Mark, S. Kamath, Review of Active Space Debris Removal Methods, Space Policy 47 (2019) 194–206. https://doi.org/10.1016/j.spacepol.2018.12.005.

[3] NASA TechPort, Low-Cost CubeSat for Active Removal of Sizable Space Debris (TechPort #154517), (2021). https://techport.nasa.gov/projects/154517?utm_source=chatgpt.com (accessed October 5, 2025).

[4] F. Basana, F. Branz, Simulation of robotic space operations with minimum base reaction manipulator, Journal of Space Safety Engineering 9 (2022) 440–448.

https://doi.org/10.1016/j.jsse.2022.06.005.

[5] M. Sabatini, R. Volpe, G.B. Palmerini, Centralized visual based navigation and control of a swarm of satellites for on-orbit servicing, Acta Astronaut. 171 (2020) 323–334. https://doi.org/10.1016/j.actaastro.2020.03.015.

[6] E.G. Asri, Z.H. Zhu, Capturing an Unknown Uncooperative Target with a Swarm of Spacecraft, in: AIAA Scitech 2024 Forum, AIAA, Orlando, FL, USA, 2024. https://doi.org/10.2514/6.2024-0625.

[7] N. Zhou, Y. Xia, R. Chen, Finite-time fault-tolerant coordination control for multiple Euler–Lagrange systems in obstacle environments, J. Franklin Inst. 354 (2017) 3405–3429. https://doi.org/10.1016/j.jfranklin.2017.02.018.

[8] K. Xu, Y. Li, J. Sun, S. Du, X. Di, Y. Yang, B. Li, Targets capture by distributed active swarms via bio-inspired reinforcement learning, Sci. China Phys. Mech. Astron. 68 (2025) 218711. https://doi.org/10.1007/s11433-024-2494-0.

[9] Z. Sun, B. Wu, J. Chen, Distributed Event-Triggered Model Predictive Control for Spacecraft Swarm, IEEE Trans. Aerosp. Electron. Syst. (2024) 1–20. https://doi.org/10.1109/TAES.2024.3485601.

[10] J. Wu, M. Liu, Y. Wang, X. Cao, Event-trigger-based cluster coordinated control of spacecraft swarm under switching topology, Aerosp. Sci. Technol. 135 (2023) 108200. https://doi.org/10.1016/j.ast.2023.108200.

[11] H. Yang, P. Antonante, V. Tzoumas, L. Carlone, Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection, IEEE Robot. Autom. Lett. 5 (2020) 1127–1134. https://doi.org/10.1109/LRA.2020.2965893.

[12] T.A. Lovell, D.A. Spencer, Relative Orbital Elements Formulation Based upon the Clohessy-Wiltshire Equations, J. Astronaut. Sci. 61 (2014) 341–366. https://doi.org/10.1007/s40295-014-0029-6.

[13] E.G. Asri, Z.H. Zhu, Bioinspired consensus-based spacecraft swarm control for autonomous capture of uncooperative targets, Acta Astronaut. 234 (2025) 419–428. https://doi.org/10.1016/j.actaastro.2025.05.009.

[14] H.D. Curtis, Orbital mechanics for engineering students, Butterworth-Heinemann, 2019.

[15] N. Wang, W. Jia, H. Wu, Y. Wang, Event-Triggered Self-Organizing Swarm Control of Distributed Unmanned Surface Vehicles, IEEE Transactions on Intelligent Transportation Systems 26 (2025) 3431–3445. https://doi.org/10.1109/TITS.2024.3521961.

[16] Z. Pan, Z. Sun, H. Deng, D. Li, A Multilayer Graph for Multiagent Formation and Trajectory Tracking Control Based on MPC Algorithm, IEEE Trans. Cybern. 52 (2022) 13586–13597. https://doi.org/10.1109/TCYB.2021.3119330.

[17] Z. Pan, C. Zhang, Y. Xia, H. Xiong, X. Shao, An Improved Artificial Potential Field Method for Path Planning and Formation Control of the Multi-UAV Systems, IEEE Transactions on Circuits and Systems II: Express Briefs 69 (2022) 1129–1133.

https://doi.org/10.1109/TCSII.2021.3112787.

[18] B. Wie, H. Weiss, A. Arapostathis, Quarternion feedback regulator for spacecraft eigenaxis rotations, Journal of Guidance, Control, and Dynamics 12 (1989) 375–380. https://doi.org/10.2514/3.20418.

[19] G.S. Seyboth, D. V. Dimarogonas, K.H. Johansson, Event-based broadcasting for multi-agent average consensus, Automatica 49 (2013) 245–252. https://doi.org/10.1016/j.automatica.2012.08.042.

Highlights

- Event-triggered distributed control in autonomous spacecraft swarms
- Behavior-based local control for target autonomous search, chase, safe enclosure
- Distributed local sensing detect target motion and geometry collectively
- Robust reconstruction handles noisy, partial landmark observations.
- Simulations demonstrate feasibility and scalability of swarm control.

Declaration of interests

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

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: