Progress in Aerospace Sciences 165 (2026) 101238

Contents lists available at ScienceDirect

Progress in Aerospace Sciences

journal homepage: www.elsevier.com/locate/paerosci

Comprehensive review of swarm intelligence for space robotics

Zixuan Zhang, Zheng H. Zhu *

Department of Mechanical Engineering, York University, 4700 Keele Street, Toronto, M3J 1P3, Canada

ARTICLE INFO

Keywords:
Swarm intelligence
Space swarm robotics
Swarm controllability
Swarm scalability and adaptability
Swarm optimality and performance
Spacecraft formation control
Autonomous multi-spacecraft coordination

ABSTRACT

Swarm intelligence has emerged as a transformative paradigm for autonomous space robotics, enabling scalable, robust, and adaptive behaviors through decentralized coordination of multiple agents. Inspired by collective phenomena in nature, swarm intelligence provides solutions to the challenges of extreme space environments, where resilience, autonomy, and fault tolerance are crucial. This review explores recent advances in the modeling, control, and validation of swarm-based space robotic systems. Mathematical frameworks ranging from single- and double-integrator dynamics to orbital swarm dynamics are examined, alongside formation control strategies such as consensus-based, leader–follower, virtual structure, and behavior-based approaches. The review covers swarm controllability, scalability, and performance metrics, highlighting trade-offs between efficiency, robustness, and computational complexity. Emerging optimization paradigms, including bio-inspired algorithms, hybrid global-local strategies, and multi-objective optimization, are surveyed for their applicability to mission-critical tasks such as debris removal, and distributed satellite constellations. The review also investigates numerical simulation platforms and experimental testbeds associated with swarm intelligence, highlighting their role in bridging the gap between theory and deployment. Case studies of current and proposed space missions illustrate the transition of swarm intelligence from conceptual design to operational reality, while trends in reinforcement learning, blockchain integration, and large language model-guided swarms signal future research directions. By consolidating theoretical foundations, experimental progress, and mission applications, this paper outlines the opportunities and challenges of harnessing swarm intelligence for future space exploration and infrastructure.

1. Introduction and overview

Swarm intelligence (SI), a term widely used in swarm robotics, describes the collective capabilities of autonomous agents and has been extensively studied for distributed problem-solving tasks that emphasize decentralized and self-organizing approaches [1]. It is a subfield of artificial intelligence (AI) [2] and draws inspiration from the collective behavior observed in natural systems such as ant colonies, bird flocks, and fish schools, as shown in Figs. 1 and 2. In these systems, autonomous agents (e.g., ants, bees, or fish) collectively solve complex tasks in a distributed manner through local interactions with one another and their environment; through simple rule-based behaviors, these interactions give rise to emergent global behavior.

SI is particularly beneficial for space robots, as it enables robust and autonomous operation in extreme and unpredictable environments with minimum human intervention. Several mission concepts have explored the potential of SI in space in the early days, including NASA's Autonomous Nano-Technology Swarm (ANTS) mission, which envisioned 1000 cooperative spacecraft for Near Earth Asteroid exploration [4]; ESA's Asteroid Population Investigation & Exploration Swarm (APIES) mission, which proposed a leader-follower architecture with 18 follower spacecraft for asteroid population investigation [5], and CNES' FDIR concept, which aimed at fully distributed coordination and fault management in Earth-observation constellations [6]. Beyond these conceptual studies, SI principles have also seen practical application: SpaceX's Starlink mega-constellation exemplifies a successful engineering implementation of swarm-like behavior in spacecraft systems [7].

In general, SI can be categorized into four key areas, as illustrated in Fig. 3: (i) swarm-based optimization algorithms, (ii) swarm robotics control, (iii) swarm controllability, scalability and performance, and (iv) physical and cyber-physical swarm intelligence (see Fig. 4).

Swarm-based optimization algorithms encompass bio-inspired methods such as particle swarm optimization (PSO), ant colony optimization (ACO), and evolutionary algorithms, including genetic

* Corresponding author.
E-mail address: gzhu@yorku.ca (Z.H. Zhu).

https://doi.org/10.1016/j.paerosci.2026.101238
Received 29 August 2025; Received in revised form 4 February 2026; Accepted 16 June 2026
Available online 26 June 2026

Fig. 1. Flocking natural swarm [3].

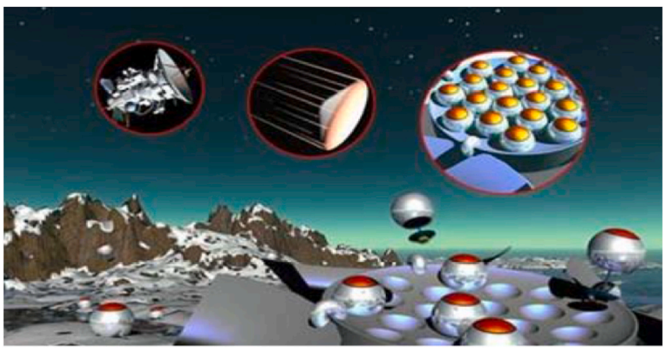


Fig. 2. Concept of a robotic swarm for exploring an unknown planet [3].

algorithms (GA) and differential evolution (DE), that address complex optimization problems by mimicking collective behavior observed in nature. Recent advancements in this area increasingly integrate SI with deep learning, reinforcement learning, and blockchain technologies to

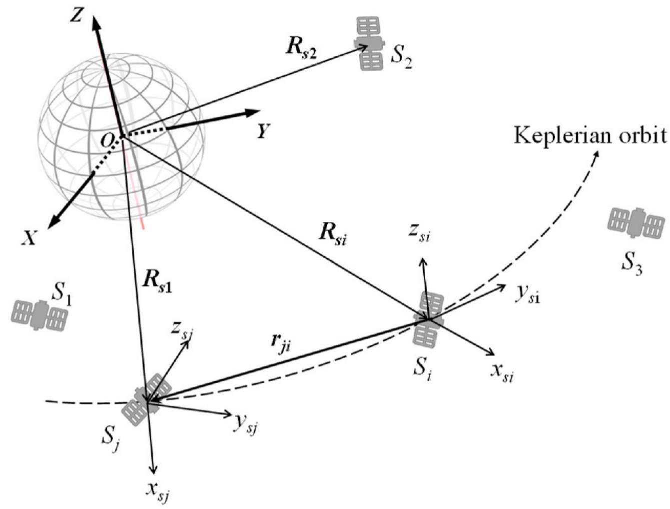


Fig. 4. A sketch of swarm dynamics in orbit.

improve scalability, adaptability, and optimization performance in dynamic environments.

Swarm robotics control is concerned with decentralized coordination among multiple robot agents, enabling the emergent collective behaviors through approaches such as pursuit–evasion, cyclic pursuit, formation tracking, and a variety of architectural paradigms such as structure-based, leader-follower, and behavior-based schemes. These enable swarm robotics to autonomously achieve formation control, collision avoidance, and recovery or reorganization following system disturbances.

Swarm controllability, scalability, and performance address the development of models, performance metrics, and optimization strategies. Core considerations include throughput, energy efficiency, robustness, trade-offs between global and local optimality, and

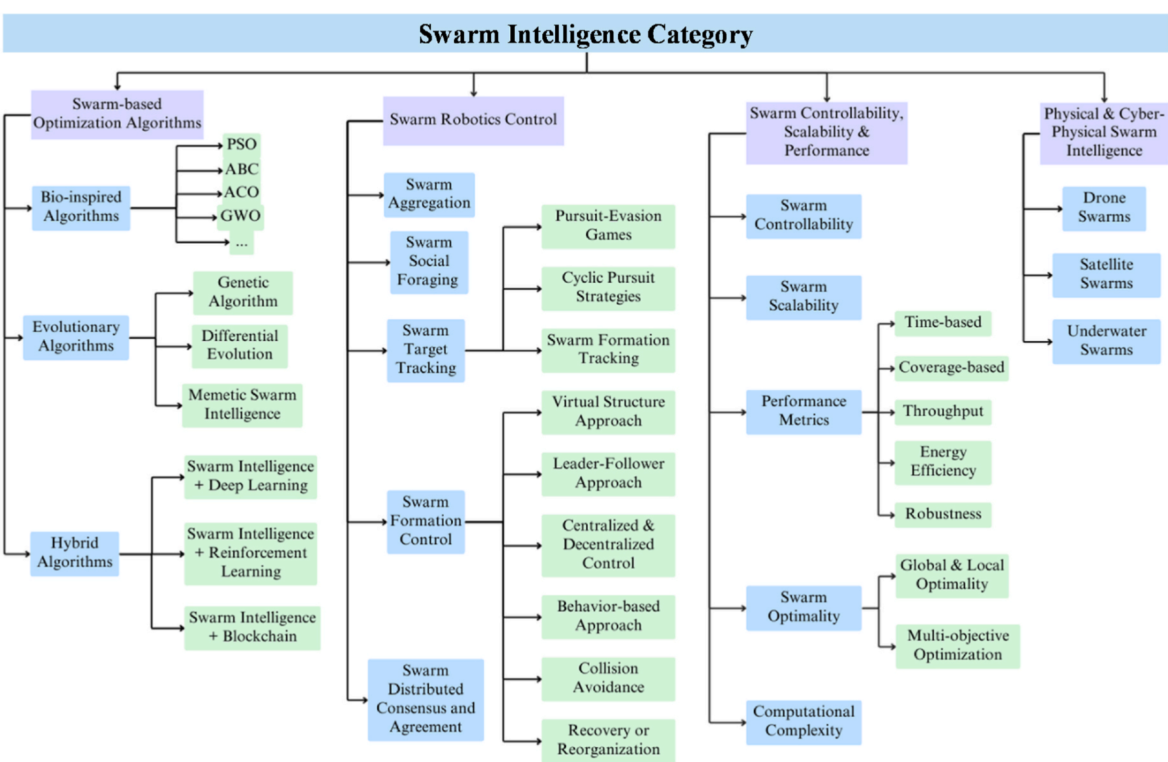


Fig. 3. Category of swarm intelligence.

computational complexity.

Finally, physical and cyber-physical swarm intelligence translates these theoretical principles into the real world. Applications range from drone swarms and satellite constellations to underwater swarm robotics, where tight integration of distributed agents with their operating environments facilitates robust, scalable, and adaptive collective behaviors.

This paper is organized around the above categories. Following this brief Introduction, Section 2 presents dynamic modeling approaches for swarm-based space robotic systems. Section 3 examines the scalability challenges inherent to SI and reviews optimization strategies critical to space robotic applications. Section 4 discusses the fundamental control principles that govern swarm behavior within the unique constraints of space environments. Section 5 provides an overview of test facilities and experimental platforms developed for validating swarm-based space robotic systems. Section 6 examines the practical applications of swarm intelligence in space robotics, identifies emerging research trends, and outlines future opportunities for the field. Finally, Section 7 concludes the paper.

2. Mathematical modeling and formation control of space robots in swarm

Mathematical modeling of robots (hereafter referred to as agents) is the foundation for analyzing the forces and interactions that dictate their collective motion and swarm behavior [8].

2.1. Mathematical models for space swarm agents

Space swarm systems consist of large numbers of relatively simple spacecraft that cooperate through local interactions to achieve global mission objectives. Mathematical modeling of such systems is essential for analysis, control design, verification, and performance evaluation.

2.1.1. Swarm orbital dynamics

When modeling and controlling swarm robotics and their formation configurations in space, it is essential to account for the effects of orbital dynamics as well as the relative dynamics in swarm configuration control. Assume a swarm system consisting of n agents in space, each following a Keplerian orbit. Introduce the Earth-centered inertial (ECI) coordinate frame with its origin at the center of Earth and its x–y plane coincident with the Earth's equatorial plane. The X-axis is directed toward the mean vernal equinox of the reference epoch, and the Z-axis is aligned with the Earth's rotation axis and is normal to the equatorial plane, pointing toward the North pole. The Y-axis completes the right-handed orthogonal triad.

The equation of motion for a single agent i in the ECI frame is given by:

$$\ddot{\boldsymbol{R}}_{S_i} = -\mu \boldsymbol{R}_{S_i} \Big/ \left|\boldsymbol{R}_{S_i}\right|^3 + \boldsymbol{a}_{S_i} \tag{1}$$

where $\boldsymbol{R}_{S_i} = (X_{S_i}, Y_{S_i}, Z_{S_i})$ is the altitude of the agent and μ is the gravitational parameter.

To control the relative motion of agents within the swarm, the relative motion of agent j with respect to agent i, both in Keplerian orbits, is generally described by the Tschauner–Hempel (TH) equations [9] in the local orbital frame $oxyz$ as follows:

$$\begin{bmatrix} \ddot{x}_{S_i} \\ \ddot{y}_{S_i} \\ \ddot{z}_{S_i} \end{bmatrix} = \begin{bmatrix} 2\dfrac{\mu}{R_{S_i}^3} x_{S_i} + 2\omega \dot{y}_{S_i} + \dot{\omega} y_{S_i} + \omega^2 x_{S_i} \\ -\dfrac{\mu}{R_{S_i}^3} y_{S_i} - 2\omega \dot{x}_{S_i} - \dot{\omega} x_{S_i} + \omega^2 y_{S_i} \\ -\dfrac{\mu}{R_{S_i}^3} z_{S_i} \end{bmatrix} + \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{2}$$

where the origin is attached to the mass center of agent i with x-axis pointing from the Earth's center towards the agent i, y-axis along-track and z-axis cross-track; the disturbance acceleration $\boldsymbol{a} = \left[a_x, a_y, a_z\right]$ includes the effects of solar radiation pressure, atmospheric drag, third-body gravitational perturbations (e.g., from the Moon and Sun), and Earth's oblateness (i.e., J_2 perturbation).

If the swarm is operating in a circular orbit, then $e = 0$, leading to $\omega = \mu / R_{S_i}^3, \dot{\omega} = 0$. Thus, the TH equations are reduced to the well-known Clohessy–Wiltshire (CW) equations [9] as

$$\begin{bmatrix} \ddot{x}_{S_i} \\ \ddot{y}_{S_i} \\ \ddot{z}_{S_i} \end{bmatrix} = \begin{bmatrix} 2\omega \dot{y}_{S_i} + 3\omega^2 x_{S_i} \\ -2\omega \dot{x}_{S_i} \\ -\omega^2 \dot{z}_{S_i} \end{bmatrix} + \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{3}$$

2.1.2. Position control model for space swarm agents

Position control constitutes a core component of space swarm coordination [10], as many collective objectives, such as rendezvous, formation keeping, coverage, and interferometric alignment, can be reduced to regulating the relative positions of agents. In contrast to terrestrial swarms, position control in space must explicitly account for orbital dynamics, limited actuation authority, and strict safety constraints. Consequently, most position control models are formulated at the guidance or acceleration command level and embedded within the underlying orbital dynamics.

For a swarm operating in a circular reference orbit, the relative motion of agent i with respect to a reference spacecraft is governed by the CW equations:

$$\begin{aligned} \ddot{x}_i &= 2\omega \dot{y}_i + 3\omega^2 x_i + a_{x,i} \\ \ddot{y}_i &= -2\omega \dot{x}_{ii} + a_{y,i} \\ \ddot{z}_i &= -\omega^2 z_i + a_{z,i} \end{aligned} \tag{4}$$

where $\omega = \mu / R^3$ is the orbital angular velocity and $\boldsymbol{a}_i = \left[a_{x,i}, a_{y,i}, a_{z,i}\right]^T$ represents the control acceleration produced by onboard actuators after compensating for orbital environmental disturbance terms discussed in Section 2.1.1.

Let $x_{d,i}(t)$ denote the desired relative state of agent i, defined either with respect to a leader spacecraft or as part of a prescribed geometric formation. The position control objective is to ensure

$$\lim_{t \to \infty} \left\| x_i(t) - x_{d,i}(t) \right\| = 0 \tag{5}$$

while satisfying orbital dynamics, actuator constraints and maintaining safe inter-agent separation.

The tracking error is defined as

$$e_i = x_i - x_{d,i} \tag{6}$$

The control problem is thus reduced to stabilizing the error dynamics induced by the CW model.

2.1.3. Velocity-position model for space swarm agents

In addition to pure position regulation, many space swarm missions require simultaneous coordination of both relative position and velocity. Examples include reconfiguration maneuvers, distributed synthetic aperture systems, rendezvous operations [11], and swarm reshaping under orbital perturbations. In such scenarios, the control objective must explicitly incorporate coupled velocity-position dynamics rather than treating velocity as a secondary state.

Unlike terrestrial multi-agent systems that often assume simple double-integrator dynamics, orbital motion introduces intrinsic coupling between position and velocity states due to Coriolis and gravitational gradient terms. Therefore, velocity cannot be controlled independently of position.

Defining the velocity-position state vector

$$x_i = \begin{bmatrix} p_i \\ v_i \end{bmatrix} = \begin{bmatrix} x_i & y_i & z_i & \dot{x}_i & \dot{y}_i & \dot{z}_i \end{bmatrix}^T \tag{7}$$

then, the controlled linear time-invariant system becomes

$$\dot{x}_i = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ A_p & A_v \end{bmatrix} x_i + \begin{bmatrix} 0_{3\times3} \\ I_{3\times3} \end{bmatrix} u_i \tag{8}$$

where

$$A_p = \begin{bmatrix} 3\omega^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\omega^2 \end{bmatrix}, A_v = \begin{bmatrix} 0 & 2\omega & 0 \\ -2\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9}$$

For coordinated swarm motion, the desired trajectory of agent i is specified as $x_{d,i}(t) = \left[p_{d,i}(t), v_{d,i}(t)\right]^T$. The control objective is defined as

$$\lim_{t\to\infty}\|p_i(t) - p_{d,i}(t)\| = 0, \quad \lim_{t\to\infty}\|v_i(t) - v_{d,i}(t)\| = 0 \tag{10}$$

Define the tracking error as

$$e_i = \begin{bmatrix} e_{p,i} \\ e_{v,i} \end{bmatrix} = \begin{bmatrix} p_i - p_{d,i} \\ v_i - v_{d,i} \end{bmatrix} \tag{11}$$

2.2. Swarm formation control for space swarm agents

Swarm formation control for space swarm agents [12] concerns the distributed coordination of multiple spacecraft to acquire, maintain, and reconfigure prescribed geometric patterns (e.g., rings, planes, tetrahedra, shells) while operating under space-specific constraints: limited propellant, thrust saturation, intermittent communications, navigation uncertainty, and the coupled dynamics of orbital motion. The core objective is to regulate relative states (position/velocity) so that the swarm's configuration satisfies a formation specification, typically while enforcing collision avoidance and minimizing fuel or time.

Swarm formation control for space swarm agents is typically organized around three closely related implementation styles that map naturally onto formation-flying missions in orbit: leader-follower, virtual-structure, and fully distributed graph-based control. In leader-follower approaches [13], one spacecraft (or a small subset) defines a reference trajectory or relative motion profile, and the remaining agents regulate their relative states to that leader and/or to local neighbors, a configuration widely studied for formation flying and relative maneuvering in near-circular orbits. In virtual-structure methods [14], the swarm is treated as a single rigid “virtual body” whose motion is planned at the group level, with each spacecraft tracking its assigned slot within the structure; this framing is popular for precision array geometries and has been developed explicitly for spacecraft formations, including versions that incorporate formation feedback and decentralized execution. In fully distributed architectures [15], formation objectives are encoded on an interaction graph and achieved through neighbor-based coordination laws (e.g., consensus-like coupling of relative position/velocity errors), which supports time-varying connectivity and local-information operation that is attractive for larger swarms and for reduced-communication scenarios in space.

A typical mission concept using a swarm of CubeSats in formation for inspection of a large spacecraft is shown in Fig. 5 [16].

For example, for swarm systems modeled by single integrator dynamics, a fundamental approach to formation control involves regulating the relative positions among neighboring agents. Assuming each agent can measure or estimate the relative position information of its immediate neighbors, a distributed control law can be formulated as [17].

$$u_i = -\sum_{j\in N_i} k_{ij}\left[(x_i - x_j) - (d_i - d_j)\right] \tag{12}$$

where N_i denotes the set of neighboring agents of agent i, $k_{ij} > 0$ is a scalar weight representing the interaction strength between the agent i and agent j, d_i and d_j represent the desired relative positions of the agent i and agent j in the formation configuration.

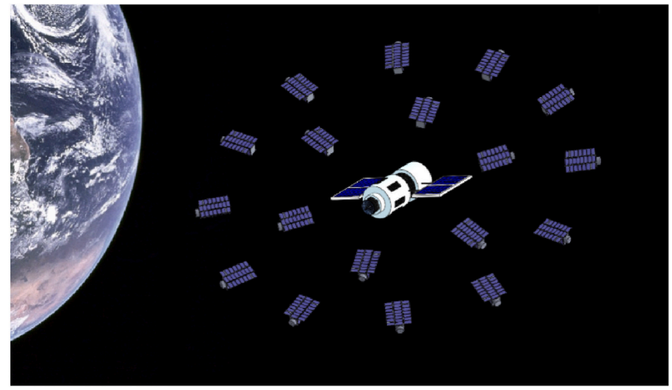


Fig. 5. A swarm of CubeSats is inspecting a bigger spacecraft [16].

As a result, the swarm collectively achieves and maintains the specified formation shape without requiring a centralized controller or global position information.

3. Space swarm intelligence controllability, scalability and optimality

3.1. Swarm controllability

Space swarm controllability [18] extends the classical notion of controllability to distributed aerospace systems operating under strict physical and resource constraints. In classical control theory, controllability refers to the ability to drive a system from any initial state to any desired state using admissible control inputs. For space swarms [19], this definition must be refined because actuator saturation, collision avoidance requirements, limited Δv budgets, communication constraints, and navigation uncertainty make full state controllability neither required nor practically achievable. Instead, controllability must be interpreted with respect to mission level objectives such as formation shape regulation, collective trajectory tracking, safe reconfiguration, and task driven performance metrics like coverage or baseline geometry. A space swarm can therefore be considered practically controllable if a feasible distributed control policy exists that drives mission relevant collective errors to within tolerance while respecting physical and operational constraints [20].

Assume CW equations are described in the state-space form

$$\dot{z}_i = A_o z_i + B_o u_i \tag{13}$$

where z_i is the state that includes relative position and velocity of spacecraft i. Even if the pair (A_o, B_o) is theoretically controllable, physical actuation limits (e.g., low-thrust propulsion) and orbital coupling often render some motion directions harder or costlier to achieve in practice. Controllers thus are designed not only for reachability but also for cost-optimal execution, especially over extended mission horizons.

In realistic space swarm systems [21], communication coupling interacts with the underlying orbital dynamics. A commonly adopted distributed spacecraft swarm model that captures both communication topology and physical dynamics can be expressed as [22]:

$$\dot{Z} = (I_N \otimes A_o - L \otimes K)Z + (B_L \otimes B_o)u \tag{14}$$

where Z stacks all spacecraft states, A_o models individual orbital dynamics, L is the network Laplacian, K is a feedback gain, B_L selects leader agents. In this framework, controllability depends on the interplay between network topology, leader selection, and individual spacecraft dynamics. Modes associated with combined eigenvalues of L and A_o may be poorly controllable if either the network is not designed to excite them or if control authority along specific physical axes is insufficient.

Since the early 2000s, bio-inspired and decentralized control paradigms have emerged to enhance swarm controllability and robustness while reducing complexity. In these frameworks, agents interact locally based on biologically inspired rules, such as attraction-repulsion, alignment, or stigmergy, without relying on identifiers, memory, or centralized planning. For example, Asri & Zhu [23] demonstrated that simple attraction-repulsion and coverage rules can produce cohesion, flexibility, and resilient collective behaviors.

More recently, swarm controlled by large language models (LLMs) has emerged as a conceptual advance [24]. Based on LLMs, these control strategies investigate the integration of cognitive-level coordination and decision-making within agent swarms. While showing promises for tasks requiring high-level reasoning or human interaction, LLM-guided swarms raise new challenges regarding decentralization, explainability, and adherence to the core principles of swarm intelligence [25]. Specifically, within the space domain, NASA's Cognitive Space Gateway (CSG) [26] has been employed as a learning-enabled, adaptive routing approach for CubeSat and SmallSat swarm topologies to improve multi-hop network performance under intermittently connected, dynamic orbital network conditions, thereby enhancing link reliability, throughput, and routing resilience in distributed spacecraft swarms.

3.2. Swarm scalability

Swarm scalability refers to the capability of a swarm system to sustain or improve collective performance as the number of agents increases [27]. It is a fundamental concern in swarm robotics and multi-agent systems. A graphical presentation for swarm scalability, robustness, and adaptability is shown in Fig. 6 [27]. While early studies focused on interference-limited performance, current approaches integrate analytical, algorithmic, and empirical methodologies, enabling scalability in complex and high-dimensional environments.

In the context of space swarms, scalability assumes additional significance due to the unique characteristics of the orbital environment. Unlike terrestrial swarm robotics, spacecraft operate under coupled orbital dynamics, limited propulsion authority, constrained onboard computation [28], sparse and intermittent inter-satellite communication links, and strict collision-avoidance requirements [29]. To address scalability under such constraints, Lu et al. [30] proposed a decentralized guidance and target assignment strategy for large modular satellite swarms, enabling distributed onboard computation and collision-free trajectory planning despite limited communication capabilities. Similarly, Skobelev et al. [31] proposed a multi-agent adaptive scheduling approach for large-scale space observation systems, demonstrating how negotiation-based coordination among agents can support scalable planning for dozens to hundreds of spacecrafts.

3.2.1. Foundational models of swarm scalability

Foundational models of swarm scalability originate from graph-theoretic consensus systems, statistical physics flocking models, mean-field game theory, distributed optimization, and networked control systems. These theoretical frameworks provide mathematical descriptions of how collective behavior evolves as the number of agents increases. In space applications [32], these models must be extended to account for nonlinear orbital dynamics, propulsion constraints, collision-avoidance requirements, and intermittent communication, leading to hybrid formulations that integrate network theory with astrodynamics.

Early analytical work by Lerman and Galstyan (2002) [33] modeled homogeneous behavior-based swarms in foraging tasks without explicit communication. Swarm performance $P(N)$ was expressed as,

$$P(N) = N \cdot p(N) \tag{15}$$

where N is the agent number in the swarm, $p(N)$ denotes per-agent performance and was found to decrease with N due to interference:

$$p(N) \propto \frac{1}{1 + \alpha N} \tag{16}$$

Here, α is the interference factor, which leads to sublinear scaling, $P(N) < N$, and an optimal swarm size that maximizes the performance $P(N)$ before interference effects dominated.

Recognizing the lack of formal scalability metrics for measurable, reproducible models for verification in the above works, Hecker and Moses [34] developed an evolutionary tuning framework within the central-place foraging algorithm, which shifted scalability analysis to search-based optimization by defining efficacy as:

$$E = \frac{P(N)}{N} \tag{17}$$

where E measures the performance per robot, and a decreasing E with increasing N indicates sublinear scaling. By optimizing error tolerance, flexibility, and scalability in foraging, they showed that evolved swarms could maintain higher E across a larger N.

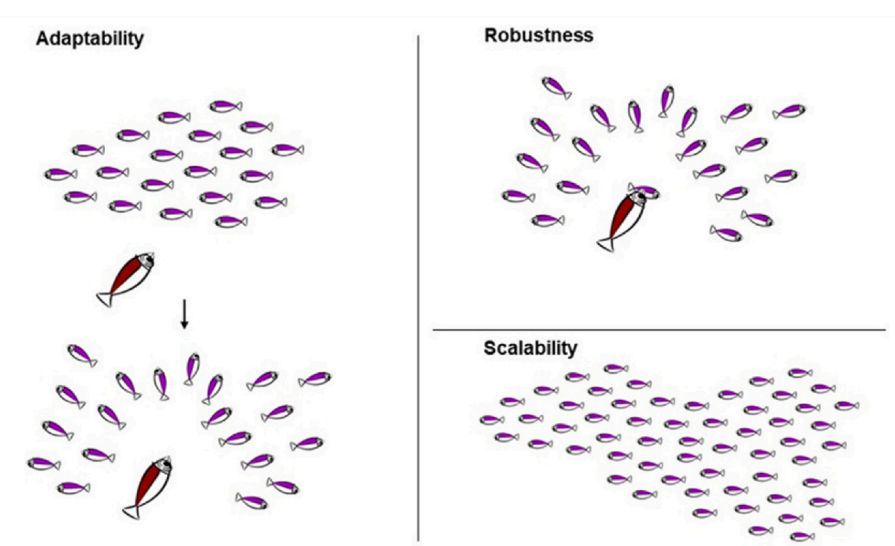


Fig. 6. Swarm advantages scalability, robustness, and adaptability [27].

3.2.2. Quantitative metrics and theoretical scalability laws

Quantitative measures for large-scale swarm evaluation beyond small-scale lab experiments were formalized by Harwell and Gini [35], who introduced quantitative metrics for scalability, emergence, and flexibility that could be applied to systems involving thousands of agents, such that:

$$S(N)=\frac{P(N)}{N\cdot P(1)}\begin{cases}>1 & \text{Superlinear scaling}\\ =1 & \text{Linear scaling}\\ <1 & \text{Sublinear scaling}\end{cases} \tag{18}$$

They validated their approach through simulations with up to 16,384 robots, comparing projected and observed performance curves to quantify scalability.

Adhikari [36] challenged the traditional assumption that scalability is inherently bounded by interference by Lerman & Galstyan [33]. Through physical robot experiments involving conveyor bucket brigades and collaborative pulling, they identified superlinear scaling law:

$$P(N)\propto N^{\beta} \tag{19}$$

where $\beta>1$ indicates superlinear scaling (collaborative synergy), $\beta=1$ indicates linear scaling (ideal additive behavior), and $\beta<1$ indicates sublinear scaling (typical due to interference). This scaling law indicates that coordination synergies lead to higher per-robot efficiency, a superlinear scaling ($\beta>1$) in swarm performance, as swarm size grows. This is the first real-world evidence of scaling synergy, directly contradicting earlier interference-limited models proposed.

On the theoretical front, Hamann and Reina [37] unified the three major scalability laws using a microscopic agent interaction framework:

1. Amdahl's Law ([38]):

$$S(N)=\frac{N}{1+N(1-\sigma)} \tag{20}$$

where σ is the serial (coordination) fraction. This formulation highlights that the achievable speedup is inherently limited by the portion of the computation that cannot be parallelized.

2. Gustafson's Law ([39]):

$$S(N)=N+\sigma(1-N) \tag{21}$$

which assumes problem size grows with N, making serial overhead less impactful. This formulation reflects that, under scaled-speedup assumptions, the effective parallel workload grows proportionally with N, causing the serial fraction to asymptotically approach zero in relative contribution.

3. Gunther's Universal Scalability Law ([40]):

$$S(N)=\frac{N}{1+\sigma(N-1)+\kappa N(N-1)} \tag{22}$$

where σ and κ describe contention (shared resources) and coherence delay (e.g., communication costs), respectively. In this formulation, the κ-dependent term introduces a quadratic interaction cost, capturing the loss of scalability that arises from all-to-all coordination among units.

These scalability laws originate from multiprocessor parallel computing, where they describe how speedup is limited by serial workload, communication overhead, and contention for shared resources. When applied to swarms, the same principles capture how coordinational effort, inter-agent communication, and shared information pathways constrain collective performance as the number of agents increases.

Milner et al. [41] further advanced empirical scalability assessment by introducing Swarm Performance Indicators (SPIs), which measure Key Performance Indicator changes in per-agent efficiency between swarm sizes and can be embedded directly into the swarm design process. One typical SPI metric was formulated as:

$$\text{Scalability SPI}=\frac{P(N_2)/N_2}{P(N_1)/N_1} \tag{23}$$

for two swarm sizes N_1 and N_2. A value SPI >1 implies increasing per-agent efficiency, while SPI ≤ 1 implies saturation or decline. These SPIs are meant to be used alongside failure mode thresholds.

Although these scaling laws were originally derived and validated in terrestrial swarm robotics, their implications extend directly to space swarm systems. In orbital environments, performance scaling is further shaped by communication topology, collision-avoidance constraints, propulsion limitations, and distributed estimation coupling. As the number of spacecraft increases [42], communication overhead and safety constraints may grow quadratically in fully connected architectures, potentially driving sublinear behavior, whereas locality-preserving interaction graphs and distributed control strategies can maintain near-linear scaling. Therefore, the classical performance law $P(N)\propto N^{\beta}$ provides a conceptual foundation, but its realization in space swarms critically depends on how orbital dynamics and resource constraints influence the effective value of β.

3.2.3. Algorithmic strategies for scalable swarm control

Recent work has demonstrated scalable bio-inspired and consensus-based control frameworks for spacecraft swarms operating in complex mission scenarios. For example, Asri and Zhu [43] proposed a two-phase decentralized control architecture for autonomous capture of uncooperative debris, combining flocking/anti-flocking behaviors with a consensus-based synchronization protocol and auction-driven distributed task allocation. In a related behavior-based framework, Izzo and Pettazzi [44] proposed a distributed satellite path-planning method that enables identical spacecraft to autonomously achieve desired formations using limited sensing, where desired velocities are generated through equilibrium-shaped high-level behaviors. Similarly, Pinciroli et al. [45] proposed a scalable distributed control strategy for pico-satellite swarms based on artificial potential fields, enabling autonomous formation of hexagonal lattice structures using only local neighbor information and demonstrating scalability up to 500 satellites.

3.2.4. Scalability in large-scale optimization problems

Large-scale space swarm missions are inherently formulated as constrained optimization problems, where coordination, collision avoidance, fuel efficiency, and task allocation must be jointly optimized across multiple agents [46]. As the number of spacecraft increases, the dimensionality of the decision space and the number of coupling constraints grow rapidly, often leading to combinatorial or quadratic complexity in centralized formulations [47].

In the context of optimization, scalability is often analyzed by considering two competing factors: the improvement in task performance $P(N)$ as agents are added, and the increase in coordination cost $C(N)$. In an ideal scenario, task performance grows linearly with the number of agents, while coordination costs grow only logarithmically, such that,

$$P(N)\propto N,\quad C\,(N)\propto \log N \quad\Rightarrow\quad \text{Scalability}\propto\frac{N}{\log N} \tag{24}$$

in contrast, poor scalability arises when coordination costs grow more quickly than task performance, eroding the benefits of additional agents, such that,

$$C\,(N)\propto N^2 \quad\Rightarrow\quad \text{Scalability}\propto\frac{N}{N^2}=\frac{1}{N} \tag{25}$$

Speedup and efficiency metrics are frequently used to assess these dynamics, with speedup defined as $S(N)=P(1)\,/P(N)$ and efficiency defined in Eq. (17). These metrics help evaluate how well a swarm can divide labor and execute tasks in parallel, providing deeper insights into swarm behavior under varying loads and sizes.

3.3. Swarm performance metrics

While decentralized swarms are widely recognized for their superior robustness, scalability, and adaptability compared to centralized systems, due to their distributed nature and absence of single points of failure, these qualitative characteristics lack well-recognized, standardized, and quantitative metrics [41]. Most metrics to date evaluate either individual agent behavior or small agent groups, rather than the collective performance of the swarm. Recent review studies on swarm intelligence, including comprehensive analyses of principles, algorithms, applications, and evaluation approaches [48], highlight that systematic performance assessment remains a critical research challenge, particularly in terms of scalability, robustness, and interpretability. Several key metrics, drawn from swarm robotics, distributed AI, and control theory, have emerged as standard indicators. In artificial swarm design, Haque et al. [49] proposed an empirical evaluation framework demonstrating that communication model–task pairing significantly influences swarm performance, highlighting that performance assessment must account for the interaction between coordination topology and task structure rather than treating communication as a fixed background assumption. Complementing this perspective, Harwell and Gini [50] proposed a set of quantitative metrics for swarm engineering, including measures of scalability, flexibility under environmental perturbations, congestion effects, and emergent self-organization, derived using tools from time-series analysis and queueing theory. Their work emphasizes the need to bridge intuitive design principles with rigorous analytical performance indicators.

In spacecraft swarms, these performance metrics take on heightened importance due to strict limitations on communication bandwidth, long signal delays, and constrained power and propulsion resources [51], as highlighted in recent overviews of emerging swarm missions [52] that emphasize the need to balance communications, energy, and control capabilities across distributed small satellite systems such as Starling and HelioSwarm. Time-based metrics directly impact fuel usage for formation keeping and proximity operations; throughput relates to the rate of scientific data collection or coordinated imaging; energy efficiency governs mission lifetime for power-limited CubeSat swarms; and robustness reflects the ability to tolerate radiation-induced faults, communication dropouts, and agent failures without jeopardizing formation safety or mission success.

3.3.1. Time-based metrics

Time-oriented performance metrics quantify how quickly a space swarm can achieve and maintain a mission objective under orbital dynamics, limited actuation, and intermittent communications. These metrics are essential because many space-swarm tasks are time-windowed (e.g., Earth observation access windows, inter-satellite link opportunities, eclipse periods) and because orbital motion couples time directly to geometry. In constellation and distributed satellite systems, temporal measures such as revisit time, access time, and response time are widely adopted as primary performance indicators [53,54].

The most fundamental of these is Time to Completion, defined as $T_c = T_{end} - T_{start}$. Lower T_c indicates better performance, as they reflect faster task execution. Closely related temporal metrics, including revisit interval and maneuver/reconfiguration time, are frequently used to assess coverage persistence and mission responsiveness in satellite constellations and scalable swarms [53]. This metric is widely applied in time-sensitive applications such as search and rescue, exploration, and foraging. In space swarms, "completion" should be stated explicitly, e.g., (i) achieving a target formation within tolerance, (ii) completing a coverage plan over a region, or (iii) reaching a required baseline distribution for distributed aperture sensing. In Earth observation missions, for example, completion may correspond to satisfying coverage and revisit constraints over a defined region of interest [54].

A more refined metric, Normalized Completion Time, scales T_c based on factors like swarm size or task complexity to allow fair comparisons across different setups. Another variant, Mean Task Execution Time per Agent, defined as T_c/N , evaluates per-agent efficiency [55]. This is useful when the task can be parallelized (e.g., distributed search/coverage). These temporal metrics are often combined with efficiency ratios, e.g., (Task Completion Rate)/T_c, to assess the trade-off between speed and accuracy. Time-based metrics also play a vital role in multi-objective optimization to balance goals like speed, energy consumption, and robustness, especially in evolutionary swarm systems and human-swarm teaming contexts [41].

3.3.2. Coverage-based metrics

Coverage-based performance metrics assess how effectively a swarm can explore, sense, or monitor an environment over time. In swarm robotics, the core quantity is typically an area coverage ratio; for spacecraft swarms, the same concept extends naturally to time-windowed and often 3D/volumetric coverage due to orbital geometry [56], line-of-sight constraints, and sensor field-of-view (FoV).

To capture spatial efficiency, researchers also track redundant coverage, how often the same area is revisited, a factor that enhances persistence in surveillance but reduces efficiency in exploration. Coverage over Time metrics capture how quickly the swarm progresses, which is critical in dynamic or time-sensitive tasks [57]. In complex environments, weighted coverage, defined as $\sum_i w_i \cdot c_i$ where w_i is the importance weight and c_i is the coverage status of the region i, is applied [58].

Distributed spacecraft operate in a three-dimensional orbital environment, and their sensing regions depend on sensor volumes, field-of-view cones, and relative orbital geometry. As a result, effective coverage depends on lateral distribution as well as altitude, orbital phase, line-of-sight conditions, and pointing constraints. Volumetric coverage ratios [59], defined as the fraction of a target three-dimensional region observed by at least one spacecraft, better represent typical on-orbit sensing missions such as debris detection or distributed space-weather monitoring.

For large-scale constellations, recent high-order analytical expansion methods enable rapid and high-precision evaluation of satellite visibility and rise/set events, significantly improving computational efficiency for mega-constellation coverage analysis [60]. Complementarily, resilient distributed coverage control strategies based on local Voronoi feedback demonstrate how satellites can autonomously regulate their spatial configuration under Hill–Clohessy–Wiltshire orbital dynamics to maintain scalable and fault-tolerant coverage performance [61].

3.3.3. Throughput-based metrics

Throughput quantifies the number of completed tasks per unit time and is a key metric in transport, logistics, or delivery applications. It is defined as $N_{\text{Tasks completed}}/T_c$. Higher throughput indicates a swarm's ability to perform many tasks efficiently within a given timeframe. This metric is often normalized by the number of agents to assess per-agent contribution and system scalability, defined as $N_{\text{Tasks completed}}/(N \cdot T_c)$ [62].

For spacecraft swarms, "tasks" are frequently constrained by orbital access and communications (e.g., time-limited observation windows, contact opportunities, downlink capacity). Recent work on dynamic nanosatellite scheduling with distributed ground stations demonstrates that intelligent pass allocation and cooperative reception strategies can significantly increase network good-put and overall throughput under realistic link variability and limited ground contact duty cycles [63].

3.3.4. Energy efficiency-based metrics

Energy efficiency is a critical performance metric in swarm systems, particularly in resource-constrained environments such as spacecraft swarms, where propulsion capability, onboard power, and communication energy are limited. It measures how effectively a swarm converts available energy into mission-relevant performance and is essential for long-duration autonomy, scalability, and sustainable operation. A common formulation is E = (Useful work done)/(Total energy consumed). The definition of useful work varies by applications, for example, the number of items retrieved (foraging), the area covered (exploration), the objects delivered (transport), or the tasks completed (assembly).

In orbital environments, energy efficiency is strongly influenced by orbital perturbations and fuel expenditure. Disturbances such as J2 and atmospheric drag increase propulsion demands for formation maintenance and collision avoidance. In this context, Morgan et al. [64] proposed swarm-keeping guidance strategies that minimize propellant usage while ensuring collision-free motion under J2 and drag perturbations, highlighting the importance of disturbance-aware energy metrics. Energy efficiency is also critical during fault recovery. Kang et al. [65] proposed a hybrid multi-objective optimization framework for fault reconstruction in giant satellite swarms, incorporating total energy consumption, fuel balance, and swarm health into the objective function. Their work underscores that energy efficiency serves as both an operational and system-level performance constraint.

From a quantitative perspective, energy efficiency in spacecraft swarms is typically characterized through propellant consumption and total Δv expenditure, as maneuvering and station-keeping dominate long-term resource usage. For spacecraft swarms, energy efficiency is most commonly evaluated through propellant consumption and total Δv expenditure since maneuvering and station-keeping dominate long-term resource usage. Accordingly, optimal spacecraft swarm reconfiguration via chief-orbit refinement formulates swarm reconfiguration as a convex optimization problem that explicitly minimizes total and maximum Δv across the swarm, directly linking Δv reduction to improved energy efficiency and lifetime extension [66]. Centralized Δv-optimal guidance strategies for under-actuated N-satellite formations further treat propellant expenditure as the primary performance objective, demonstrating how trajectory optimization reduces maneuver cost while satisfying safety and control constraints [67].

3.3.5. Robustness-based metrics

Robustness in swarm robotics refers to the swarm's ability to maintain acceptable performance despite individual agent failures or external disturbances. In spacecraft swarms, robustness is especially critical due to (i) environmental perturbations (e.g., J_2, drag, SRP) [68], (ii) intermittent communications and delays, (iii) navigation uncertainty, and (iv) actuator faults and bounded thrust authority.

From a spacecraft formation-design perspective, robustness is frequently quantified through safety-constrained performance measures, such as guaranteed minimum separation distances and bounded relative motion under perturbations. Robust formation control strategies explicitly compensate for nonlinearities and modeling uncertainties, ensuring convergence of relative position and attitude errors within prescribed neighborhoods despite disturbances [69]. At the estimation level, robustness metrics include sensitivity to measurement noise, non-Gaussian disturbances, and partial sensing degradation. In this context, Wang et al. [70] proposed a decentralized Huber-based cubature filtering algorithm for formation-flying spacecraft that improves robustness to non-Gaussian noise and modeling uncertainties, outperforming decentralized EKF-based approaches under sensing disturbances. At the mission level, Joyner and Plice [71] proposed an active swarm resiliency strategy for the HelioSwarm mission, enabling reconfiguration into contingency configurations following deputy failures, thereby preserving mission objectives under partial agent loss.

Beyond control and estimation, swarm robustness must account for agent loss and degraded sensing. Distributed coverage control based on local Voronoi feedback explicitly evaluates resilient performance under satellite failure and biased deployment, maintaining coverage objectives using only local relative information [61]. Additionally, communication topology strongly influences robustness under link degradation or agent removal. Studies of satellite swarm communication networks show that certain structured topologies provide improved performance retention under node loss, revealing trade-offs between efficiency and robustness in hostile or disrupted environments [72].

One common metric is performance degradation, defined as $R = P_{fail}/P_{no\ \min\ al}$ where P is task performance before and after agent failure, measuring degradation in task success when agents fail. A robust system shows little to no degradation, e.g. $R \approx 1$, which is essential in unpredictable or harsh environments. Complementing this, survivability measures the ratio of functioning agents over time, such as $S = N_{active}(t)/N_{initial}$ [73]. This metric reflects how well the swarm maintains operational capacity as agents fail. To assess the swarm limit, a fault tolerance threshold defines the maximum number or percentage of agent failures the swarm can withstand before performance drops below a critical level. Another key metric is Time to Recover (or resilience time), which is the duration the swarm takes to return to acceptable performance after a disruption, indicating its adaptability. Finally, the metric behavioral stability quantifies how consistent the swarm's behavior remains under varying conditions, such as noise, delays, or changing environments.

3.4. Swarm optimality

Swarm optimality refers to the ability of a swarm system to achieve a collective objective in the most efficient or effective way. This may involve minimizing resource usage (e.g., energy, time, communication overhead), maximizing performance metrics (e.g., task completion, coverage, accuracy), or balancing trade-offs between competing objectives. "Optimality" in space swarms concerns optimal swarm reconfiguration and maneuver efficiency [66], optimal control algorithms for swarm performance [74], and numerical and computational optimality techniques. It is typically a multi-objective notion balancing propellant expenditure, safety, sensing performance, and autonomy under orbital dynamics, sparse communication, and actuator limits [75]. Mathematically, swarm optimality can be cast as the minimization of a global cost function J over the agent population:

$$\min_{\{x_i,u_i\}} J = \sum_{i=1}^{N} J_i(x_i, u_i, \mathscr{N}_i) \tag{26}$$

where J_i is the local cost for the agent i (e.g., task error, energy, latency), x_i is the state of the agent i, u_i is the control input of the agent i, $\mathscr{N}_i$ is the neighborhood of the agent i defining local interactions (e.g., via communication or sensing), and N is the total number of agents.

To ensure emergent optimality, global optimization is often decomposed into a hierarchical swarm model [76].

$$\min_{\{x_i,u_i\}} J_{hier} = \sum_{l=1}^{L} w_l \sum_{j=1}^{n_l} \sum_{k=1}^{m_j} f_{j,k}^{l}\left(x_{j,k}^{l}\right) \tag{27}$$

where L is the number of hierarchy layers, n_l is the number of subgroups at level l, m_j is the number of agents in group j at level l, $x_{j,k}^{l}$ is the state of agent k in group j at level l, $f_{j,k}^{l}$ is the cost function for group i at level l, and w_l is the level-wise importance weight.

At each level, the agents or subgroups apply local optimization strategies, such as particle swarm optimization, to refine solutions before propagating updates up or down the hierarchy. This framework supports scalable convergence and modular coordination. A generalized schematic of this hierarchical swarm optimization process is illustrated in Fig. 7.

In a broader sense, swarm optimality is generally classified into global, local, and multi-objective optimization, depending on the scope of the optimization, which will be outlined below.

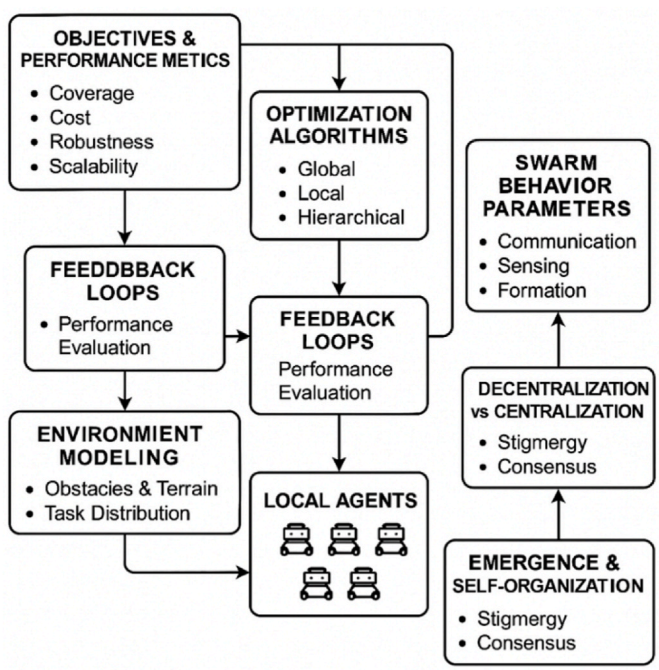


Fig. 7. Generalized schematic of hierarchical swarm optimization process.

3.4.1. Global and local optimality

3.4.1.1. Global optimality. Global optimality refers to the process of identifying the best possible solution, i.e., the global optimum, within the entire search space of an optimization problem [77]. In swarm-based optimization frameworks, this typically implies that agents share or retain knowledge of the best solution discovered by the population. Such shared information guides the collective search process, enabling a dynamic balance between exploration (searching new regions of the solution space) and exploitation (refining promising candidate solutions).

In spacecraft systems, global optimality assumes particular importance due to the nonlinear, constrained, and mission-critical nature of space operations [66]. For example, Ding et al. [78] proposed a homotopy-based continuation framework for low-thrust fuel-optimal transfers between Sun–Earth Halo orbits within the CRTBP, combining Pontryagin's Maximum Principle with particle swarm optimization for initial costate estimation, and demonstrating efficient computation of globally feasible homoclinic transfer windows under multirevolution and bang-bang control structures.

3.4.1.2. Local optimality. In contrast to global optimality, which seeks the best solution across the entire search space, local optimality focuses on identifying the best solution within a restricted neighborhood or subset of agents [79]. This localized optimization may not converge to the global optimum but can be computationally efficient and contextually effective. Instead of relying on a global best solution, agents leverage a local best that represents the best-known solution within their defined neighborhood, which is determined by spatial proximity or topological structures such as ring or star formations. For instance, traditional path planning methods struggle with complex constraint spaces, and many classical nature-inspired algorithms often converge to local optima due to limited optimization capability.

In space contexts, local optimality frequently arises in real-time decision-making scenarios where computational resources and response times are constrained. Chen et al. [80] introduced a dynamic control framework for real-time debris avoidance in space missions by defining four guidance rules to generate the shortest collision-free trajectory. Ehresmann et al. [81] further demonstrated a heuristic optimization approach using real flight mission data, showing that local strategies, when coupled with sufficient seed-point diversity, can effectively approximate global solutions.

3.4.1.3. Hybrid global and local optimality. To enhance convergence, adaptability, and scalability across complex optimization tasks, a hybrid strategy is proposed in the literature to integrate global exploration and local exploitation to efficiently solve complex optimization problems.

In space swarms, hybrid global–local optimality typically arises through hierarchical or layered optimization architectures that decompose the problem into combinatorial coordination and continuous trajectory refinement components. Decentralized model predictive control combined with sequential convex programming enables each spacecraft to solve local convex optimal control problems while coordinating to achieve swarm-level reconfiguration objectives [82]. Distributed auction-based allocation determines target assignments, while convex programming ensures collision-free, fuel-efficient trajectories, supporting scalability to large swarms. Hierarchical optimization approaches further extend this concept by incorporating dynamic communication topologies into the coordination layer [83]. Global topology planning is coupled with lower-level convex trajectory refinement to maintain safety and connectivity under time-varying communication constraints.

3.4.2. Multi-objective optimization

Multi-objective optimization (MOO) in swarm robotics addresses the simultaneous optimization of conflicting goals such as task efficiency, energy consumption, collision avoidance, communication overhead, and system robustness via Pareto-optimal solutions that balance trade-offs across multiple performance metrics.

In space applications, Oyekan [84] explored the use of an autonomous micro-robot swarm for the repair of high-value space structures. Their MOO framework effectively balances exploration, exploitation, and constraint satisfaction (e.g., obstacle avoidance) and enables reliable swarm behavior in dynamic and hazardous space environments. Yin et al. [85] proposed a multi-objective orbital maneuver optimization framework for Earth observation satellite (EOS) systems, considering response time, imaging resolution, and fuel consumption simultaneously. The authors further proposed an adaptive feedback learning NSGA-II (AFL-NSGA-II), which enhances convergence performance and maintains solution diversity in multi-satellite scheduling problems. From a formation-design perspective, Hoskins and Atkins [86] proposed a multi-objective particle swarm optimization framework that balances fuel consumption and observation time through Pareto-optimal trade-off analysis. Similarly, Stolfi and Danoy [87] proposed an evolutionary orbital formation algorithm that optimizes swarm distribution while minimizing propellant usage, demonstrating the effectiveness of evolutionary MOO methods for large-scale orbital formation design.

3.5. Computational complexity

Swarm-based algorithms are valued for their robustness and adaptability in swarm robotics, yet their computational complexity limits efficiency and scalability in real-world applications. In space swarm systems, where onboard computational resources, communication bandwidth, and energy budgets are limited, complexity considerations become mission-critical [28,88]. In large-scale spacecraft swarms, collision-free reconfiguration quickly becomes computationally prohibitive due to nonlinear dynamics, combinatorial assignment, and coupled collision constraints. Centralized optimization scales poorly with swarm size, making distributed model predictive control attractive for decoupling variables and enabling parallel onboard computation [28]. Similarly, trajectory planning under orbital perturbations exhibits rapidly increasing computational burden as the swarm grows [89]. Linearized models, assignment–trajectory decomposition, and shrinking-horizon feedback enable near-real-time reconfiguration while preserving safety and scalability.

Computational complexity arises primarily from agent interactions, frequent objective function evaluations, and iterative updates driven by both individual and collective intelligence. Additional factors include swarm size, search space dimensionality, objective function evaluation cost, and convergence criteria (see Table 1).

Table 2 below provides a comparison of the time and space complexity of several widely used swarm-based optimization algorithms, including PSO [90], ACO [91], Artificial Bee Colony (ABC) [92], Firefly Algorithm (FA) [93], Bat Algorithm (BA) [94], Grey Wolf Optimizer (GWO) [95], Whale Optimization Algorithm (WOA) [96], Salp Swarm Algorithm (SSA) [97], Dragonfly Algorithm (DA) [98], Rat Swarm Optimizer (RSO) [99], and Tuna Swarm Optimization (TSO) [100].

The complexities listed in Table 2 represent asymptotic upper bounds, expressed by the function $\mathrm{O}(\mathrm{T}, \mathrm{N}, \mathrm{D})$ [101]. These estimates reflect worst-case scenarios while ignoring constant factors and lower-order terms. In practice, actual complexity depends on implementation details, problem size, and other practical considerations.

4. Space swarm robotics control fundamentals

Control of space warm robotics encompasses several fundamental strategies, including aggregation, social foraging, target tracking, formation control, and distributed consensus. Each of these approaches is examined in detail in the following sections.

4.1. Swarm aggregation

Aggregation is a fundamental collective behavior observed in natural swarms and forms a cornerstone for swarm robotics. In the domain of space swarm robotics, aggregation refers to the ability of multiple spacecraft or robotic agents to converge and form cohesive clusters through distributed control laws. This capability enables cooperative operations such as on-orbit assembly, debris capture, communication relaying, coordinated sensing, and modular reconfiguration, and serves as primitive behavior supporting higher-level functionalities including formation building, cooperative inspection, and task allocation.

The task of aggregation for a swarm of N agents requires the definition of a control input, $u_i(t), \forall i \in \{1, \ldots, N\}$, that drives all agents to converge within a bounded vicinity of one another, while maintaining distinct state trajectories $x_i(t)$ [17], such that,

$$\lim_{t \to \infty} \| x_i(t) - x_j(t) \| \le \varepsilon \ \forall i, j \in \{1, \ldots, N\} \tag{28}$$

where $\| x_i(t) - x_j(t) \|$ denotes the inter-agent distances, ε denotes the ultimate swarm size, determined jointly by the number of agents and the controller parameters.

The parameter ε characterizes the steady-state cluster size and depends on controller gains, interaction topology, and the number of agents N. Unlike idealized point-mass models, physical space robots possess finite geometry, safety envelopes, and actuation constraints. Therefore, ε cannot be arbitrarily small and must scale appropriately with agent size, navigation uncertainty, and collision-avoidance margins.

A key requirement in space swarm aggregation [102] is strict collision avoidance under orbital dynamics. Relative motion is governed by nonlinear equations (e.g., nonlinear orbital models), and small relative velocity errors can lead to significant separation drift. Consequently, aggregation control laws often incorporate repulsive potential terms, barrier functions, or constraint-based model predictive control to guarantee safety.

Furthermore, aggregation in space must account for limited thrust authority and propellant consumption. Continuous attraction-based control strategies may be energetically inefficient; therefore, event-triggered [103] or impulsive aggregation schemes are frequently considered to reduce fuel expenditure while maintaining convergence properties.

From a systems perspective, aggregation efficiency depends on interaction topology, communication latency, and sensing accuracy. Graph connectivity and algebraic connectivity directly influence convergence rate and robustness.

A graphical sample of the aggregation process, illustrating the increase in aggregation efficiency, is shown in Fig. 8 [104].

In space applications, Harwell and Gini [35] proposed a two-stage mission concept for space debris removal where a swarm identifies, classifies, and aggregates space debris for coordinated operation. Saaj et al. [105] studied autonomous formation flying at distances under 50m using artificial potential fields. Their results demonstrated stable swarm reconfiguration and precise formation maintenance, both of which are essential for on-orbit servicing and cooperative space exploration.

Table 1
Key differences between global optimality and local optimality.

Feature	Global Optimality	Local Optimality
Focus	Entire search space.	Limited neighborhood or region.
Convergence	Guides all particles toward one global best.	Multiple regions can be explored simultaneously.
Risk	Premature convergence to a false global optimum.	Slower convergence, but better diversity.
Algorithm Behavior	Exploitation-focused.	Balances exploration and exploitation.

4.2. Swarm social foraging

Swarm social foraging [106] draws inspiration from natural systems such as bird flocks and wolf packs, where individuals cooperate to locate, harvest, and distribute resources more efficiently than alone. This collective behavior, driven by the dual objectives of maximizing energy intake while minimizing risk and expenditure, has motivated the development of algorithmic frameworks in swarm robotics.

In swarm robotics, social foraging algorithms enable autonomous agents to collaboratively explore environments, identify targets, and collect resources while adapting to dynamic and uncertain conditions. Consequently, social foraging offers a robust paradigm for designing resilient swarm systems with efficiencies comparable to those observed in nature. An example of such behavior is shown in Fig. 9, which depicts 50 Kilobots performing a collective foraging task [107].

In contrast to terrestrial systems, space swarm robotics operate under strict constraints imposed by orbital mechanics, limited propulsion capability, communication latency, radiation exposure, and high mission risk. Therefore, the objective of social foraging in space is not merely resource collection, but the coordinated optimization of mission-level performance metrics such as fuel efficiency, coverage rate, communication load, and operational safety. Tan et al. [108] proposed the Lunarminer framework for biomimetic swarm-based lunar water ice extraction. Their simulations demonstrated improved task allocation, reduced extraction time and energy consumption, and strong fault tolerance under partial robot failure.

Environmental conditions play a critical role in shaping agent behavior during space social foraging. In orbital or planetary contexts, the environment can be abstracted as follows:

- **Favorable regions**: Resource-rich asteroid zones, scientifically valuable planetary sites, debris clusters for removal missions, or designated mission targets.
- **Unfavorable regions**: Collision-prone orbital corridors, high-radiation areas, gravitational perturbation zones, or dynamically unstable regions that must be avoided.

The design challenge lies in enabling space swarm agents to

Table 2
Temporal–spatial complexity and scalability comparison of swarm-based control and optimization algorithms.

Algorithm Category	Typical Space Application	Time Complexity	Space Complexity	Scalability	Notes for Space Swarms
Centralized Optimal Control (e.g., centralized MPC)	Formation reconfiguration, fuel-optimal planning	$O(N^3) – O(N^4)$	$O(N^2)$	Poor	Coupled state optimization; computational bottleneck beyond tens of satellites
Distributed MPC (Decoupled/ Parallel)	Collision-free reconfiguration	$O(N \cdot I \cdot d^3)$	$O(N \cdot d^2)$	Moderate	Scales linearly in N if coupling constraints are limited
Consensus-Based Control	Attitude/orbit synchronization	$O(N \cdot k \cdot I)$	$O(N \cdot d)$	Good	Depends on communication topology (graph sparsity critical)
Artificial Potential Fields (APF)	Formation keeping, collision avoidance	$O(N^2)$ (global) $O(N \cdot k)$ (local)	O(N)	Moderate to good	Quadratic if global interaction; scalable if locality enforced
Task Allocation Algorithms	Target assignment, observation scheduling	$O(N^2 \log N)$	$O(N^2)$	Moderate	Communication overhead increases with swarm size
Voronoi-Based Coverage Control	Distributed Earth observation	O(N log N) (2D partitioning)	O(N)	Good	Efficient for distributed sensing missions
Random Finite Set (RFS) Methods	Large probabilistic swarms	O(N) (intensity-based)	O(N)	Very Good	Reduces dimensionality; avoids explicit pairwise coupling
Bio-Inspired/Swarm Optimization (PSO, GWO, WOA, SSA, etc.)	Global mission optimization	$O(N \cdot I)$	O(N)	Moderate	Iterative convergence; may be slow for real-time large-scale swarms

N - number of agents (satellites or spacecrafts), I - number of iterations, d - dimension of state space, k - number of local neighbors in the communication topology.

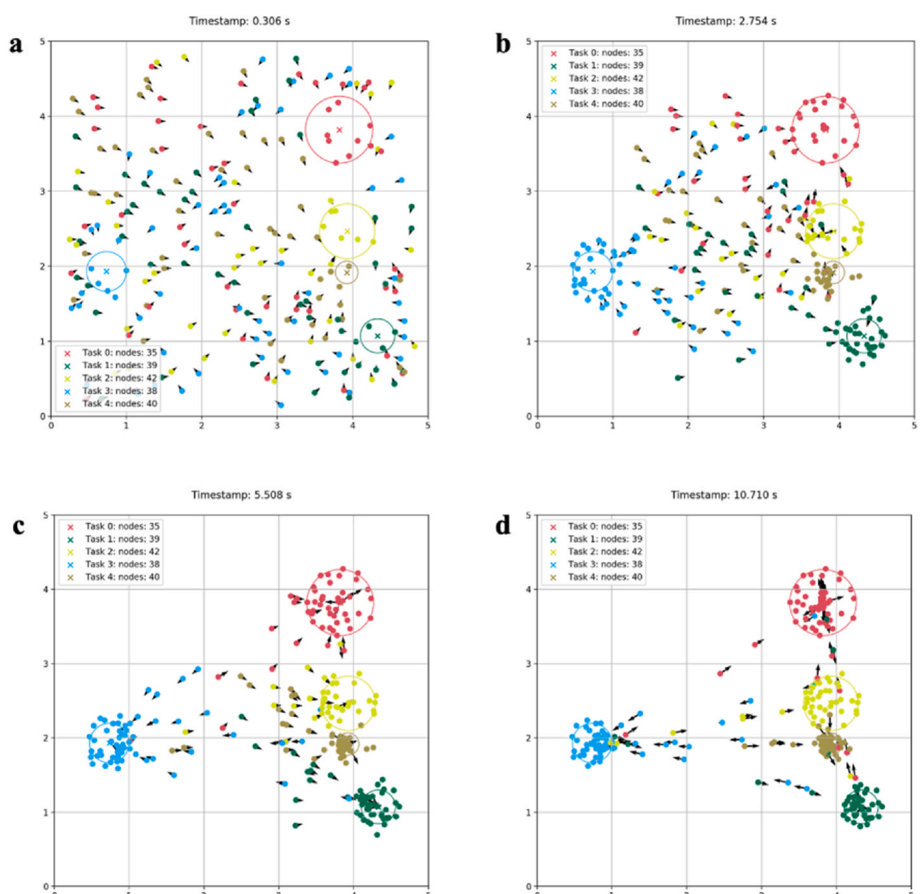


Fig. 8. Illustration of a random aggregation process [104].

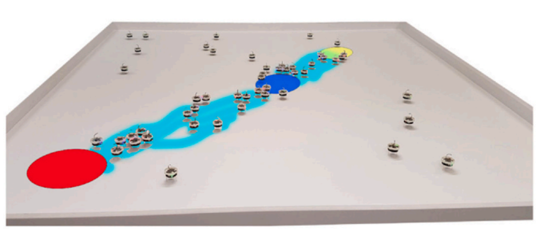


Fig. 9. Collective foraging with 50 Kilobots [107].

efficiently converge toward favorable regions while maintaining cohesion formation and avoiding hazards under dynamic constraints.

Let the environment be represented by a scalar function $\sigma(x) : R^n \rightarrow R$, referred to as the space resource-risk profile, where $x \in R^n$ denotes the position of an agent in orbital or planetary coordinates, such that,

$$\sigma(x) \begin{cases} < 0 & \text{favorable region} \\ = 0 & \text{neutral region} \\ > 0 & \text{hazardous region} \end{cases} \tag{29}$$

For any two points $(x_1, x_2) \in R^n$, if $\sigma(x_1) < \sigma(x_2)$, then x_1 is more favorable than x_2. The space swarm social foraging problem is to design a distributed control input $u_i(t), \forall i \in \{1, \dots, N\}$ for each agent i such that all agents converge to and remain within a bounded neighborhood of a local minimum c_{σ_j} of $\sigma(x)$, while satisfying spacecraft dynamics constraints. Specifically, the following condition must be satisfied [17]:

$$\lim_{t\to\infty} \|x_i(t) - c_{\sigma_j}\| \le \varepsilon_j, \forall i \in \{1, \dots, N\} \tag{30}$$

for some bounded $\varepsilon_j > 0$ and $c_{\sigma_j} \in R^n$. Simultaneously, the control strategy must ensure: (i) Avoidance of undesirable local maxima corresponding to hazardous regions, (ii) Preservation of inter-agent cohesion to maintain communication and sensing connectivity, (iii) Safe spacing constraints to prevent inter-satellite collision, (iv) Attraction toward favorable regions while repelling agents from hazardous zones. Thus, all agents will remain confined near c_{σ_j} while preserving inter-agent spacing $\|x_i(t) - x_j(t)\| \le \varepsilon$ and $\varepsilon > 0$.

The principles of social foraging and swarm intelligence have also been extended to space applications. In spacecraft attitude control, Cooper and Smeresky [109] reviewed evolutionary algorithms for spacecraft attitude control, highlighting their potential for adaptive, decentralized optimization in highly dynamic orbital environments. In satellite formation, Nag and Summerer [110] investigated collision avoidance strategies for satellite clusters to perform coordinated evasive maneuvers in response to an external threat, while preserving overall formation integrity. After the threat diminishes, the control system autonomously reconfigures the cluster to its original state, ensuring mission continuity with minimal disruption.

4.3. Swarm target tracking

Swarm target tracking for space swarm robotics refers to the coordinated control of multiple spacecraft or robotic units to continuously monitor and follow one or more moving targets or dynamic environmental phenomena [16,111]. Representative tasks include pursuit of non-cooperative objects, distributed encirclement for surveillance or containment, and cooperative escort during proximity operations. These missions require adaptability to maneuvering or uncertain target dynamics, environmental disturbances, and strict operational constraints such as limited thrust, collision avoidance, and communication latency [112].

4.3.1. Pursuit-evasion games

A fundamental problem in swarm target tracking is the pursuit-evasion game [113], where one group of agents (pursuers) attempts to capture or enclose another group (evaders), often under incomplete information regarding evader trajectories. In space swarm applications, this framework models scenarios such as debris interception, inspection of non-cooperative spacecraft, asteroid containment, or defensive space operations [114].

Pursuit-evasion problems have been extensively studied within differential game theory, robotics, and aerospace due to their relevance to surveillance, interception, and cooperative tracking tasks. Fig. 10 illustrates the coordinate system of spacecraft pursuit-evasion game [115].

Numerous studies have focused on pursuit-evasion game applications in spacecraft systems. Li and Luo [116] investigated an orbital impulsive pursuit-evasion game under spacecraft dynamic constraints using differential game theory. He et al. [117] proposed a hierarchical meta-heuristic optimization framework and demonstrated its applicability to spacecraft trajectory design and pursuit-evasion scenarios.

4.3.2. Cyclic pursuit strategies

A notable decentralized strategy is cyclic pursuit, wherein each agent follows another in a closed-loop pattern. This approach, inspired by natural swarm behaviors, enables robust group dynamics through simple local rules. Fig. 11 provides a trajectory example of cyclic pursuit [118].

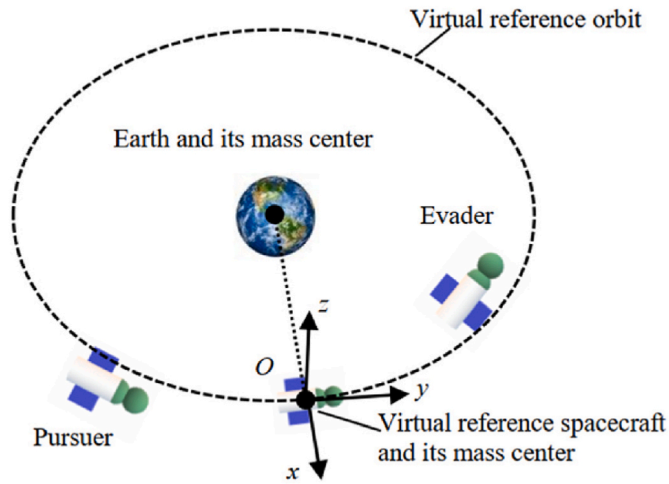


Fig. 10. Illustration of spacecraft pursuit-evasion game [115].

Cyclic pursuit strategies have evolved from multi-agent formulations to targeted applications in guidance and control. Kim et al. [119] established a cooperative control framework for N agents in 3D space, where each of N agents tracks its successor agent $i+1$ mod N, providing a foundational methodology for distributed coordination. Extending these concepts, Mukherjee et al. [120] introduced an enhanced cyclic pursuit scheme capable of adapting to the additional complexity of intercepting maneuvering targets, demonstrating the strategy's versatility for practical mission scenarios.

4.3.3. Swarm formation tracking

Swarm tracking extends beyond simple pursuit to include interception, encirclement, and escorting of targets, while maintaining prescribed formations [121]. The primary objective is to enable a swarm of agents to perform precise and continuous tracking of one or more moving targets, while adapting in real time to variations in target behavior, environmental conditions, and the swarm's own operational dynamics. In space swarm systems, this capability is essential for applications such as non-cooperative spacecraft inspection, debris monitoring, asteroid escort, and distributed sensing of dynamic orbital phenomena. For instance, cooperative space swarm formation tracking with safety constraints, e.g., collision avoidance, saturation, and uncertainties, is studied in Ref. [122] and formation tracking under adversarial/actuation attacks is investigated in Ref. [112]. An illustration of SpaceX Starlink satellite constellation in low Earth orbit is shown in Fig. 12 [123].

Unlike traditional trajectory tracking problems, swarm tracking is complicated by partial target observability, where agents may lack complete information, such as target velocity or acceleration. Mathematically, the problem can be defined as follows: given a set of desired inter-agent distances $\{d_{ij} | \forall i,j \in \{1,\dots,N\}, i \neq j\}$, where d_{ij} represents the desired distance between agents i and j, the goal is to design control inputs $u_i(t), i = 1, \dots, N$, such that the following conditions are satisfied [17],

$$\lim_{t\to\infty} x_T(t) \in conv\{x_1(t), \dots, x_N(t)\} \tag{31}$$

$$\lim_{t\to\infty} \left| \|x_i(t) - x_j(t)\| - d_{ij} \right| \le \varepsilon, \forall i \neq j \in \{1, \dots, N\} \tag{32}$$

where $con\{x_1(t), \dots, x_N(t)\}$ denotes the convex hull formed by the agent positions $x_1(t), \dots, x_N(t)$ and $\varepsilon > 0$ is a small constant.

For spacecraft swarms, relative motion is shaped by differential orbital parameters, since small differences in semi-major axis, inclination, or right ascending node cause natural drift that requires continuous formation adjustment [124]. These dynamics are commonly modeled

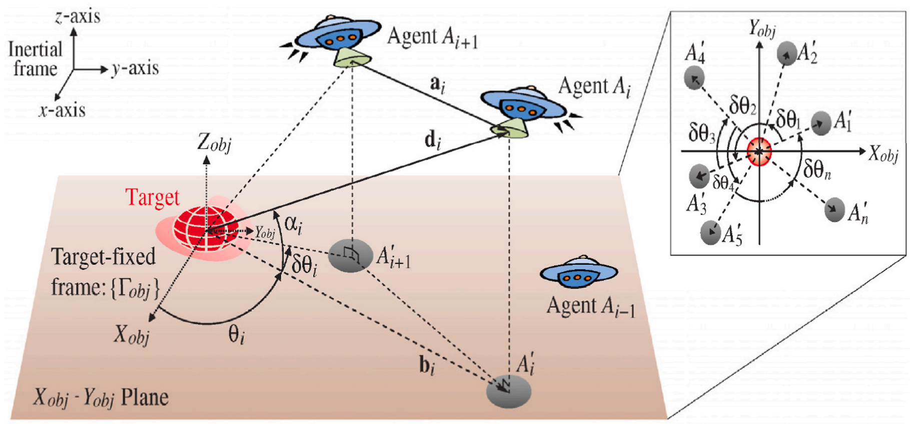


Fig. 11. Trajectories of agents with a cyclic pursuit strategy [118].

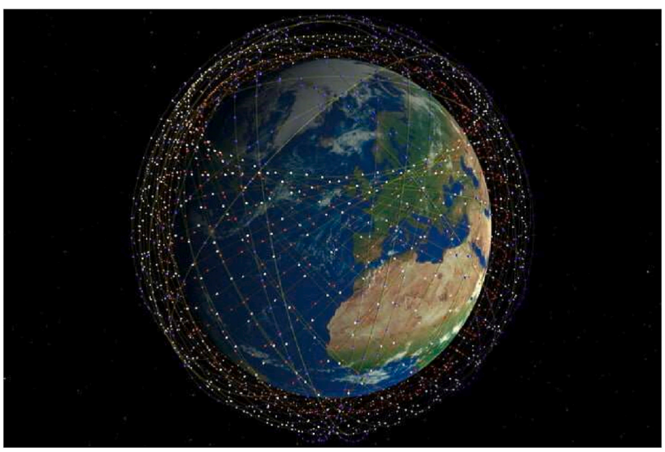


Fig. 12. SpaceX Starlink satellite constellation in low Earth orbit [123].

using the Clohessy-Wiltshire or Hill equations, which form the basis for many tracking and formation-keeping controllers. Tracking performance is further influenced by orbital perturbations, including the Earth's J2 term [114], atmospheric drag in low Earth orbit, solar radiation pressure, and third-body effects. Controllers must compensate for these disturbances while operating under limited propulsion and attitude control capability, which makes spacecraft tracking significantly more complex than terrestrial swarm tracking.

Topological connectivity, which is the minimum requirement for cooperative behavior, has been used to design fuel-efficient distributed swarm-keeping controllers that maintain connectivity through topology regulation and selective impulse maneuvers under J2 perturbations [125]. Building on this idea, behavior-driven impulsive maneuver strategies have also been formulated as optimization problems and solved with reinforcement learning to support target–attacker–defender interactions in orbital tracking scenarios [126].

4.3.4. *Algorithmic developments*

In addition to the above three architectural approaches, several algorithmic strategies have been proposed for enhancing formation control:

- Hybrid evolutionary algorithm (EA) was developed and validated using the ARGoS simulator and experiments with E-Puck2 robots for optimizing formation acquisition and maintenance [127].
- Distributed nonlinear model predictive control (MPC) scheme was introduced to minimize inter-satellite communication while preventing collisions with both cooperative spacecraft and external obstacles [128].
- Geometric formation structures were introduced to ensure stability at the agent level, though with the risk of emergent instabilities at the system level [129].
- Reinforcement learning techniques have been leveraged to address complex swarm guidance and control challenges in dynamic environments [130].

4.4. *Swarm formation control*

Swarm formation control ensures that agents maintain predefined geometric patterns during collective motion. This functionality is critical for applications such as aperture synthesis, distributed sensing, and coordinated inspections, and deep-space interferometry missions. In space robotics, precise formation maintenance directly influences measurement accuracy, communication geometry, and mission robustness [131]. An illustrative example of swarm formation control for multiple spacecraft systems is shown in Fig. 13.

The swarm formation control problem can generally be decomposed into three stages [132]:

(i) Formation acquisition (stabilization): guiding agents from arbitrary initial positions into a target geometric configuration,

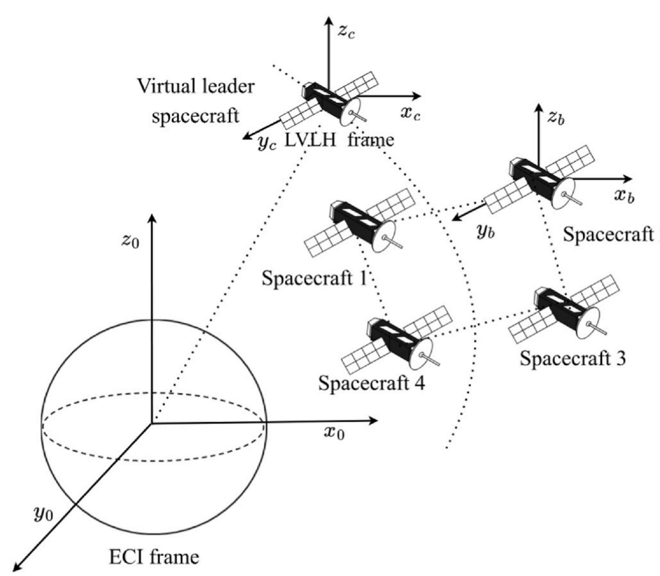


Fig. 13. Spacecraft swarm formation illustration [112].

(ii) Formation maintenance: ensuring cohesion and stability under disturbances, model uncertainties, and environmental perturbations (e.g., J2 effects, drag, solar radiation pressure),
(iii) Formation reconfiguration (switching): enabling adaptation of the formation to evolve mission requirements or environmental conditions [133].

4.4.1. Virtual structure approach

The virtual structure approach models the swarm as a single rigid body, in which each agent is assigned to a fixed position relative to a global virtual coordinate system, rather than relying purely on local interactions or decentralized decision-making [134]. Under this paradigm, the formation behaves as an idealized rigid entity whose collective motion is prescribed at the group level and then tracked by individual agents. This abstraction enables precise coordinated maneuvers and structured motion, even in dynamically varying environments. A graphical illustration of the virtual structure and its corresponding virtual rigid-body representation [134] is shown in Fig. 14.

Implementation of the virtual structure method typically follows three steps:

(i) Defining virtual structure dynamics that prescribe the global trajectory or maneuver of the swarm.
(ii) Mapping global dynamics onto individual agents, assigning each agent a fixed position within the virtual structure.
(iii) Applying tracking controllers to ensure that agents accurately follow their assigned trajectories.

Through this procedure, the swarm maintains strict geometric consistency while executing coordinated motion.

In spacecraft formation flying, the virtual structure approach has been applied by defining the desired rigid-body dynamics of the formation and projecting them onto individual spacecraft [135]. This allows the spacecraft swarm to be treated as a cohesive entity, facilitating long-term precision maneuvers such as deep-space interferometry and distributed aperture synthesis. Robustness improvements have been reported, such as the use of μ-synthesis-based controllers by Shahbazi et al. [136], which enhance tolerance to model uncertainties and environmental disturbances.

4.4.2. Leader-follower approach

The leader-follower approach is a well-established strategy in swarm formation control [137]. In this framework, one or more agents are designated as leaders that guide the swarm along a predefined trajectory or toward a specific mission goal, while the remaining agents act as followers. The followers adjust their positions by maintaining desired inter-agent distances either to the leaders or to neighboring peers. An illustrative example of a multi-group leader-follower swarm is shown in Fig. 15 [138].

The leader-follower framework is particularly effective in spacecraft formation flying [139], where a leader spacecraft maintains a reference orbit, while followers adjust their relative positions with respect to the leader. The advantage of this method lies in its efficiency: only the leader executes full orbital maneuvers, while followers require relative-state control, thereby reducing fuel consumption and system complexity. A schematic of multi-spacecraft leader-follower swarm system is shown in Fig. 16.

An important extension is the virtual leader variant [140], in which a virtual reference agent, rather than a physical leader, defines the desired formation trajectory. This allows followers to synchronize with the virtual entity via consensus-based or distributed control laws.

Numerous studies have advanced the leader-follower paradigm across different spacecraft robotic domains:

- **Robust and Finite-Time Leader–Follower Attitude Consensus**: Several studies have advanced robust leader–follower attitude coordination for spacecraft under uncertainties and disturbances. Feng et al. [141] incorporated Gaussian process regression to compensate for inertia uncertainty, actuator saturation, and external disturbances within a directed communication graph.
- **Leader-Follower Extensions: Game-Theoretic and Flexible Spacecraft Dynamics**: Beyond rigid-body coordination, Endo and Morimoto [142] extended leader–follower consensus to boundary-controlled flexible spacecraft modeled by hybrid PDE-ODE dynamics, achieving simultaneous attitude synchronization and elastic vibration suppression.
- **Resilient and Swarm-Level Formation Control**: At the formation level, Jin et al. [131] proposed a J2-aware probabilistic swarm guidance framework integrating Markov-chain-based distributed guidance with MPC tracking to achieve scalable formation under orbital perturbations. Cui et al. [112] developed a distributed Lyapunov-based MPC approach resilient to actuation attacks and input saturation, ensuring recursive feasibility and closed-loop stability. These works address resilience and large-scale swarm formation tracking under realistic space environment constraints.

4.4.3. Centralized and decentralized control approach

Swarm control strategies are broadly categorized into centralized and decentralized architectures, each with distinct advantages and inherent limitations. These paradigms shape the way multiple agents—such as spacecraft, drones, or ground robots—coordinate to accomplish collective objectives in dynamic environments.

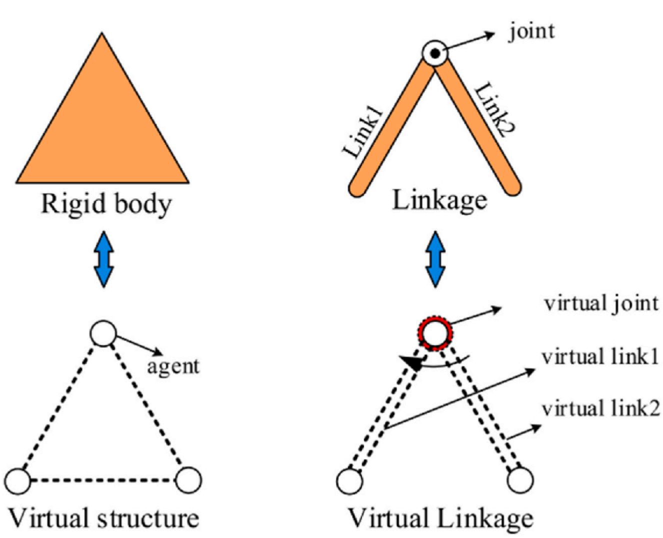


Fig. 14. Illustration of virtual structure and virtual linkage [134].

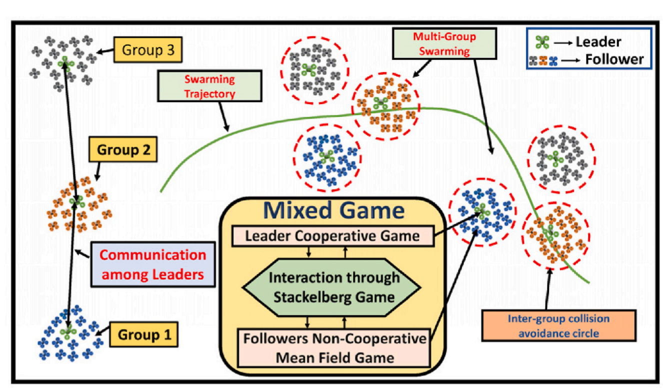


Fig. 15. An illustration of a multi-group leader-follower swarm [138].

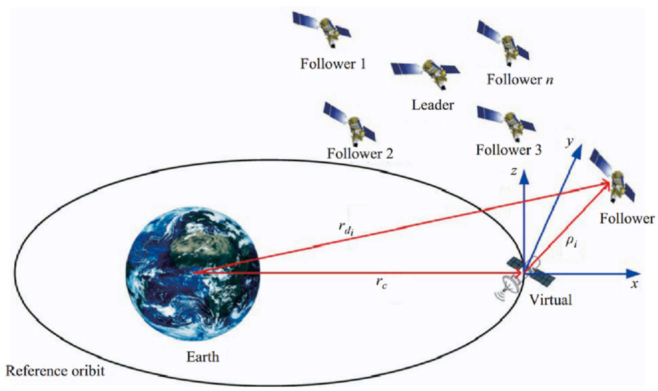


Fig. 16. A schematic of multi-spacecraft leader-follower swarm system [140].

4.4.3.1. Centralized control. In centralized control, a single supervisory unit orchestrates the behavior of the entire swarm by collecting global information, processing it, and dispatching commands to all agents [143]. This architecture ensures consistent global coordination and simplifies decision-making, as the central authority enforces a predefined mission objective. Typical applications include formation keeping, area coverage, and coordinated inspections, where optimization algorithms are often employed to improve efficiency and task execution [144].

A notable example is the Mother-Children satellite architecture proposed by Sabatini et al. [145], in which a large Mother spacecraft deploys a swarm of Children spacecraft to inspect a damaged satellite. Due to safety constraints, the Mother spacecraft cannot approach the target, making centralized coordination essential for mission execution. Despite these advantages, centralized control suffers from critical drawbacks, most notably:

- Single point of failure: A malfunction of the central controller or a disruption in inter-agent communication can collapse the entire system.
- Limited scalability: Adding more agents substantially increases communication and computational demands.

4.4.3.2. Decentralized control. In contrast, decentralized (or distributed) control eliminates dependence on a single supervisory unit by enabling each agent to make autonomous decisions based on local sensing and peer-to-peer interactions. This paradigm allows swarms to self-organize, adapt to environmental changes, and maintain resilience under partial failures [146]. Owing to these properties, decentralized control is particularly attractive for space robotics, where robustness, scalability, adaptability, and autonomy are mission critical.

A primary advantage of decentralized control in space swarm robotics is its robustness and fault tolerance - the failure of individual agents does not compromise the overall swarm. For instance, Cui et al. [112] developed a distributed Lyapunov-based MPC framework that ensures resilient spacecraft swarm formation tracking under actuation attacks and input saturation, while incorporating collision avoidance and stability guarantees through online constrained optimization.

Scalability is another defining property of decentralized architectures. Such systems can accommodate additional agents or dynamic task reallocations without substantial redesign of the control structure. In space robotics, Farrag et al. [147] proposed a satellite swarm configuration that leverages commercial-off-the-shelf radio frequency communication subsystems, illustrating scalable inter-satellite communication strategies for multi-satellite coordination.

Local decision-making in decentralized control also facilitates real-time adaptation to dynamic and uncertain conditions while reducing bandwidth requirements and communication latency. In the space application, Koenig et al. [148] explored decentralized formation acquisition strategies in perturbed orbits to highlight adaptability to realistic mission constraints.

Finally, by reducing the need for continuous human supervision, decentralized control enables real-time autonomous decision-making. This autonomy supports modular mission design, reconfigurable objectives, and resilience against hybrid cyber-physical attacks. Complementing this line of research, Cui et al. [149] proposed a resilient time-varying formation-tracking framework against DoS and actuation attacks, tailored for spacecraft formation. By integrating distributed state estimation, event-triggered communication, and adaptive output-feedback control, the method ensures uniformly ultimately bounded tracking errors for spacecraft under hybrid cyber-physical threats.

A comparison of centralized and decentralized swarm formation control is summarized in Table 3.

4.4.4. Behavior-based approach

The behavior-based control approach relies on decentralized local interaction rules inspired by biological swarms, where global formation patterns emerge from agent-level behaviors such as alignment, cohesion, and separation [152]. Unlike centralized or globally planned strategies, behavior-based control emphasizes simplicity, scalability, and robustness, enabling swarms to operate effectively under limited communication and sensing constraints. A representative illustration of behavior-based swarm mechanisms is shown in Fig. 17 [152].

Extending to space robotics, numerous studies demonstrate the potential of behavior-based control for autonomous spacecraft swarms. Asri and Zhu [23] proposed a decentralized scheme for the coordinated

Table 3
Comparison of centralized and decentralized swarm control approaches.

Aspect	Centralized Control	Decentralized Control
Architecture	Single central controller collects global data, processes decisions, and issues commands to all agents.	Each agent makes autonomous decisions based on local sensing and peer-to-peer interactions.
Coordination	High precision in global coordination; suitable for tasks requiring rigid formation or strict supervision.	Emergent coordination through local rules; supports self-organization and adaptive collective behavior.
Advantages	• Simplified decision-making logic • •Strong global optimization capability • •Effective for small swarms.	•High robustness and fault tolerance (no single point of failure) •Scalable to large swarms •Adaptive in dynamic environments •Supports autonomy and modular missions.
Disadvantages	•Single point of failure •Communication bottlenecks •Poor scalability as swarm size increases.	•Limited global optimality guarantees •Higher algorithmic complexity •Coordination challenges under constrained communication.
Applications	•Mother–Children spacecraft inspection [145] •Coordinated inspection and coverage tasks [144]	•Decentralized pose estimation •Asteroid mission swarm operations [150] •Formation acquisition in perturbed orbits [148] •Robust reconfiguration and safety [112]
Typical Algorithms	•Centralized optimization (e.g., mission-level planning, global task assignment)	•Distributed MPC [112] •Reinforcement learning for adaptive behaviors [151]
Suitability for Space Missions	Effective for tightly constrained, small-scale missions with limited agents.	Well-suited for large, autonomous, and long-duration missions in dynamic or uncertain space environments.

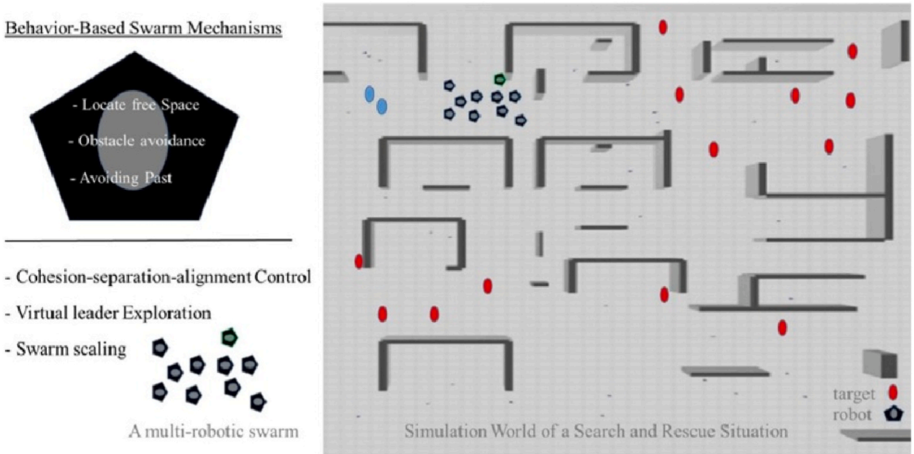


Fig. 17. Behavior-based multi-robotic swarm [152].

capture of a tumbling space target, leveraging flocking-inspired guidance to manage proximity operations. Izzo and Pettazzi [44] developed a distributed path-planning technique, wherein the spacecraft maintains relative geometry using only local sensory information. Their approach enables spacecraft to dynamically form enclosures around targets, utilizing a switching control strategy based on sperm dynamics, which optimizes individual trajectories for flexible and scalable encirclement.

While formation control provides a foundation, the practical deployment of spacecraft swarms must address key operational challenges, including collision avoidance, adaptive recovery and reorganization, and autonomous navigation in uncertain environments, which will be examined in the following sections.

4.4.5. Collision avoidance

Collision avoidance is a critical component of swarm formation control, particularly in decentralized systems where real-time adaptability and scalability are required [153]. In such frameworks, agents rely on reactive behavioral control, often modeled using artificial forces. Repulsive forces ensure that agents maintain a minimum safe distance from one another, while attractive forces guide them back to their intended formation or trajectory once the risk subsides. This decentralized mechanism enables rapid responsiveness to dynamic environments, including moving obstacles or nearby agents, without dependence on centralized coordination.

In the field of spacecraft swarms, their close-proximity operations pose significant challenges, particularly in maintaining safe distances while preserving formation configuration. For example, Zhang et al. [154] proposed a trajectory planning approach by an improved PSO algorithm to satisfy stringent safety requirements for coordinating swarm chaser spacecraft in the safe capture of non-cooperative targets. With event-triggered updates to reduce computational and communication burdens within spacecraft swarms, Sun et al. [103] introduced collision avoidance communication subgroups. Figs. 18 and 19 illustrate key aspects of these strategies: satellite collision risk zones and the structure of collision avoidance communication groups.

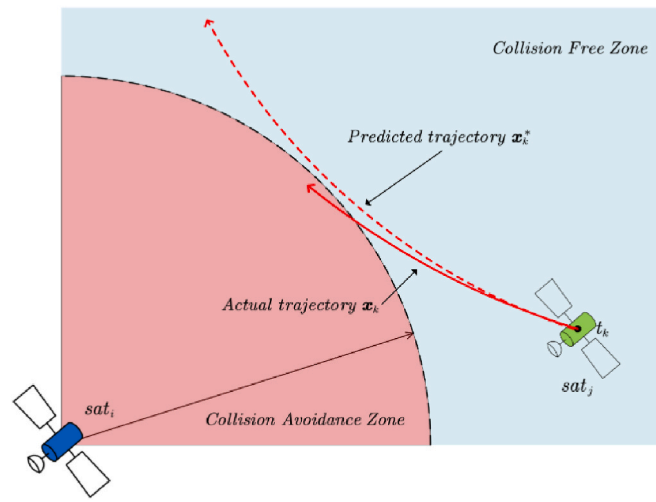


Fig. 18. Illustration of the satellite j entering the collision risk zone of satellite i [103].

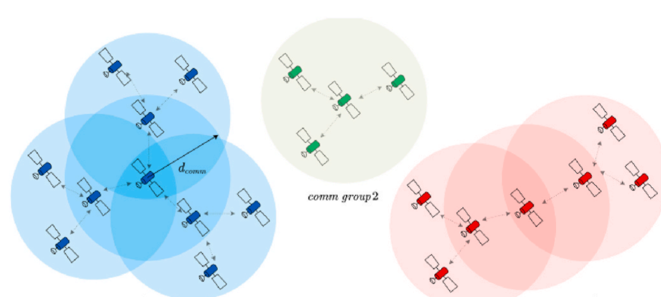


Fig. 19. Schematic of the collision avoidance communication group [103].

4.4.6. Swarm formation recovery or reorganization

In swarm robotics, recovery and reorganization are essential for sustaining collective functionality in the presence of agent loss, environmental disturbances, or mission reconfiguration. In the context of space systems, formation recovery and reorganization [155] refer to the capability of a spacecraft swarm to autonomously restore, adapt, or reconfigure its geometric structure in response to faults, orbital perturbations, communication degradation, or evolving mission objectives.

4.4.6.1. Behavioral and topological adaptation. During formation navigation, swarms often operate under conditions that test their adaptability and resilience. Highly constrained environments, sudden reconfiguration commands, and inefficient task assignments create situations where traditional coordination methods may falter. To address these complexities, researchers have designed several frameworks.

One such advancement is found in the work of Yu et al. [156], who introduced a self-organizing control framework based on dynamic topologies. This framework supports essential behaviors such as aggregation, dispersion, and formation transitions during maneuvering.

In space applications, Kempf et al. [157] proposed a self-organizing control-loop recovery framework for fractionated spacecraft, addressing data loss and controller failures through adaptive event-based networked predictive control and distributed failure compensation. An auction-based task redistribution mechanism ensures scalable and stable formation control for satellites.

4.4.6.2. Formation recovery strategies. To address the challenge of slow recovery from disordered states caused by abrupt formation changes, Quan et al. [158] proposed the Global-Remap–Local-Remap (GRLR) strategy. This method combines centralized recalibration of key formation parameters with distributed local trajectory replanning. The GRLR activates only when destabilization is detected, minimizing unnecessary computation. Recovery begins by assessing constraint awareness and solving an Alignment and Task Assignment (ALAS) problem, ensuring efficient reorganization. A schematic illustration of the GRLR strategy and ALAS method is shown in Fig. 20.

4.4.6.3. Spacecraft swarm recovery and optimization. In space applications, swarm recovery requires advanced optimization and control strategies tailored to dynamic and uncertain space environments, ensuring the resilience of long-duration missions. Venigalla et al. [159] introduced a virtual swarm optimization method for recovering missed thrust, directly controlling recovery margins in low-thrust trajectory design. By quantifying the maximum allowable coasting durations before corrective thrusting is required, their approach enhances mission robustness through principles of swarm intelligence. Liu et al. [160] addressed manufacturing-oriented recovery through a bi-objective evolutionary optimization framework for multi-robot machining of spacecraft components, jointly optimizing base positioning and task allocation to improve workspace coverage and operational feasibility in-space advanced manufacturing.

4.5. *Swarm distributed consensus and agreement*

Distributed agreement and consensus-seeking constitute fundamental coordination mechanisms in spacecraft swarms [43], enabling agents to establish common reference quantities or maintain coherent operational states using only local sensing and inter-satellite communication. In orbital environments, where centralized supervision is limited by latency and bandwidth constraints, these distributed mechanisms provide the basis for autonomous collective behavior.

Distributed agreement concerns convergence to a common discrete decision variable, such as task allocation, leader selection, or maneuver sequencing in cooperative orbital missions. Through local information exchange, spacecraft collectively determine unified decisions without centralized control. Consensus seeking [161], in contrast, applies to continuous state variables, where agents iteratively align quantities such as relative position, velocity, attitude, or orbital elements until agreement is achieved. Table 4 shows the consensus advances and applications.

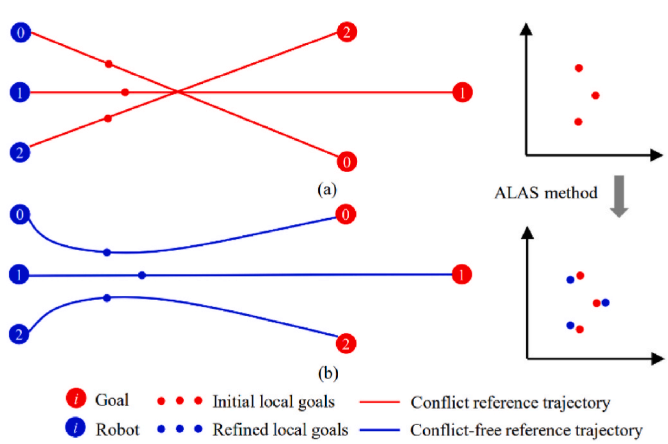


Fig. 20. Illustration of the GRLR strategy and the ALAS method [158].

5. Numerical simulation and experimental validation of swarm intelligence

5.1. *Numerical simulation software*

Testing swarm robotics algorithms on physical robots is costly, time-consuming, and operationally complex. Numerical simulation offers a practical alternative, enabling design, validation, and optimization in controlled virtual environments before hardware deployment. Various numerical simulation software have been developed and utilized in swarm robotics research, including ARGoS (Autonomous Robots Go Swarming), V-REP, Gazebo, Webots, NetLogo, swarm-bot, SwarmSim, BeeGround, and MASON. Table 5 provides a comparison of the properties of various swarm robotics simulation software.

As swarm robotics continues to evolve, simulation platforms will remain essential in driving innovation and facilitating the deployment of intelligent multi-robot systems in real-world applications.

5.2. *Experimental validation*

While numerical simulation is crucial for algorithm development, experimental validation is indispensable for assessing swarm robotics systems under realistic conditions. Such validation typically integrates hardware components (such as robots, sensors, and communication devices) with software modules (including control algorithms, logging, and analysis), enabling the implementation and evaluation of swarm behaviors across diverse platforms. Experimental work in swarm robotics can be broadly classified into three categories: laboratory-scale experiments and hardware-in-the-loop testing.

Although these platforms provide useful insight into decentralized control and collective behavior, their fidelity to spacecraft dynamics is limited by their planar motion and the absence of orbital forces. They can reproduce collision-free motion, relative-state control, and communication effects, but they cannot emulate differential orbital motion, gravitational perturbations, or impulsive thrust-based actuation. Consequently, controllers tested on planar platforms must be adapted to Hill-frame or Clohessy-Wiltshire dynamics for on-orbit use, and propulsion and perturbation effects must be reintroduced through simulation or hardware-in-the-loop methods.

5.2.1. *Laboratory-scale experiments*

Laboratory platforms allow controlled and cost-effective validation of swarm algorithms at small to medium scales. There are three different scale systems in terms of their size.

(a) Small-scale and low-cost platforms:

Kilobot [180] in Fig. 21(a) and Droplet [181] in Fig. 21(b) are among the smallest and most cost-effective platforms that are widely used for large-scale studies involving hundreds to thousands of robots. Kilobots employ vibration-based locomotion and infrared communication, while Droplets offer even lower-cost hardware with similar capabilities.

(b) Educational and modular systems:

The e-puck [182] and e-puck2 [183] are popular choices for education and vision-based research. The e-puck provides modularity, cameras, proximity sensors, and Bluetooth communication, making it suitable for both teaching and advanced experiments in universities, as shown in Fig. 21(c). Similarly, Pheeno [184], combining Raspberry Pi and Arduino with Wi-Fi, supports multi-robot systems for flexible educational and research applications, as shown in Fig. 21(d).

Table 4
Consensus advances and applications.

Types	Definition	Advantages	Disadvantages	Applications
Average consensus [162]	Agents converge to the average of their initial status; used for distribution control	Simple and scalable; effective for distributed estimation and sensing.	Convergence may be slow; sensitive to graph connectivity.	Sensor fusion, distributed averaging, cooperative estimation in space swarms.
Finite-time consensus [163]	Guarantees convergence within finite time; critical for time-sensitive missions.	Fast convergence; robust to delays; critical in urgent maneuvers.	Requires precise gain design; may be sensitive to noise.	Fast satellite synchronization, collision avoidance in orbit.
Leader-Follower Consensus [164]	Leaders define reference trajectories; followers synchronize accordingly.	Clear structure; simple implementation; allows trajectory tracking.	Vulnerable to leader failure; relies on stable leader tracking.	Chief-deputy spacecraft formations, UAV/UGV leader-follower control.
Output Consensus/ Output Agreement [165]	Consensus on subset of states (e.g., heading) while other states follow constraints.	Efficient when only subset of states matters; reduces complexity.	Partial consensus may not guarantee global stability.	Heading/attitude consensus in UAV/spacecraft formations.
State Consensus/State Agreement [166]	All agents converge on their full state vectors (e.g., position and velocity).	Ensures full state alignment; supports tight coordination.	May lead to collisions in physical swarms if geometry is not regulated.	Tight formation flying, spacecraft rendezvous and docking.
Robust Consensus under Noise/ Failures [167]	Ensures convergence under noisy measurements, delays, or link failures.	Improves reliability under imperfect communication and sensing.	Complex design; higher computational cost.	Fault-tolerant satellite constellations, noisy inter-satellite links.
Event-Triggered Consensus [168]	Updates triggered only when conditions are met; reduces communication load.	Energy- and bandwidth-efficient; reduces unnecessary communication.	Triggering rules may reduce accuracy if poorly tuned.	Resource-constrained satellite swarms, power-limited systems.
Consensus with Switching Topologies [169]	Handles time-varying communication graphs due to dynamic or intermittent links.	Captures realistic dynamic topologies; supports mobility.	Analysis is complex; stability proofs challenging.	Satellite swarms with intermittent LOS, dynamic reconfiguration.
Cluster/Group Consensus [170]	Agents converge to multiple agreement values within subgroups or clusters.	Supports task partitioning; flexible in heterogeneous missions.	Possible loss of global cohesion across subgroups.	Multi-cluster satellite constellations, heterogeneous swarm missions.
Adaptive Consensus [171]	Gains or communication weights adapt online to improve convergence under uncertainties.	Adaptive to uncertainty; increases robustness and flexibility.	May require online computation and adaptation overhead.	Adaptive spacecraft formation flying, missions with uncertain parameters.

Table 5
Comparison of swarm robotics simulation software.

Software	Physics Engine	Programming Languages	Dimension	Robot Models	Pros	Cons	Application Cases
ARGoS [172]	Box2D, ODE	C++, Python	3D	Custom	High performance, customizable	Steeper learning curve	Large-scale swarm simulations
Gazebo [173]	ODE, Bullet, etc.	C++, Python (ROS)	3D	Pre-built + custom	High-fidelity, ROS integration	Computationally intensive	High-fidelity swarm simulations
Webots [174]	ODE	C++, Python, Java, MATLAB	3D	Pre-built + custom	User-friendly, cross-platform	Limited scalability, commercial license	Education, research, prototyping
NetLogo [175]	None	NetLogo language	2D (basic 3D)	Abstract agents	Easy to learn, great for teaching	Abstract modeling only	Teaching, prototyping, abstract behaviors
V-REP (CoppeliaSim) [127]	Bullet, ODE, Vortex	Lua, C++, Python, MATLAB	3D	Custom	Modular, multiple physics engines	Steeper learning curve	Research, education, prototyping
SwarmSim [176]	None	Java	2D	Abstract agents	Lightweight, open source	Abstract modeling only	Swarm intelligence research
MASON [177]	None	Java	2D, 3D	Abstract agents	Open-source, flexible	Abstract modeling only	Swarm intelligence, social simulations
swarm-bot [178]	Vortex	Not mentioned	3D	Real, detailed, simplified s-bots	Self-organization, scalability, robustness, simplicity	Limited real robots, sim-reality gap, fitness tracking	Aggregation, coordinated motion, transport
Bee-Ground [179]	Unity Physics Engine	C#, Python (via Unity), TensorFlow 2.0	2D, 3D	Custom (e.g., MONA robot)	High scalability (1000+), ROS & URDF support, bio-inspired simulation, TensorFlow integration	Moderate learning curve, depends on Unity setup	Large-scale bio-inspired swarm robotics simulations (e.g., BEECLUST)

(c) Medium to advanced platforms

Mona [185] shown in Fig. 22(a) with omnidirectional movement and modularity, is ideal for medium-sized swarms and reconfigurable systems. Khepera IV [186], Fig. 22(b), though expensive, offers high processing power and sensing capabilities for advanced research. Colias [187], Fig. 22(c), provides an affordable yet functional alternative for low-cost swarm studies.

5.2.2. Hardware-in-the-loop testing

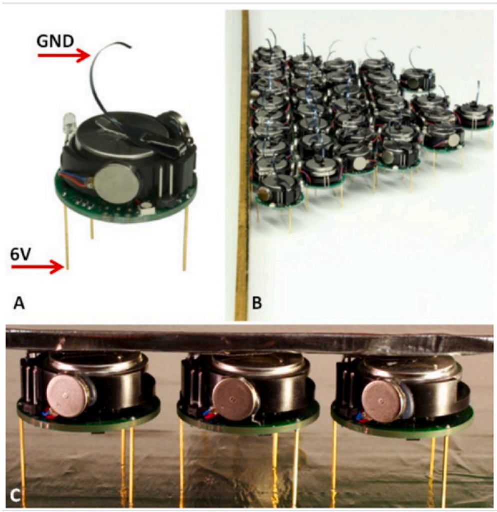
Hardware-in-the-loop (HIL) platforms integrate real robotic hardware with simulation or virtual environments, allowing algorithms to be tested under controlled but partially physical conditions.

(a) Leader–follower and formation studies:

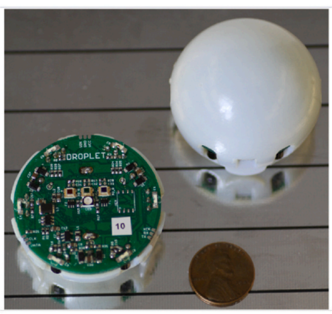
Bono et al. [188] evaluated distributed swarm algorithms using six Elisa-3 robots for multi-vehicle leader–follower formations.

(b) Self-organization platforms:

Zhou et al. [189] employed the SwarmBang system, a compact

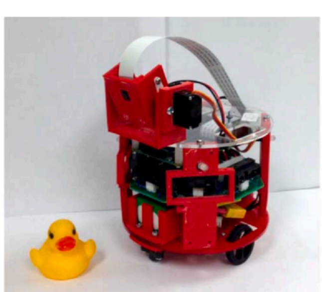


(a) An individual and a group of Kilobots [181].



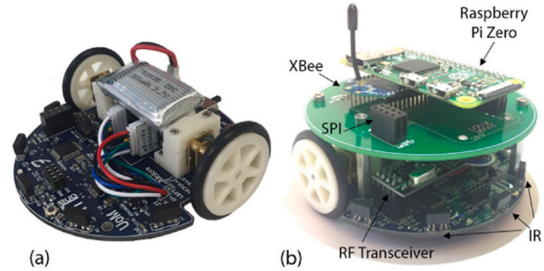
(b) The Droplet swarm robotics platform [182].

(c) The e-puck robot [183].



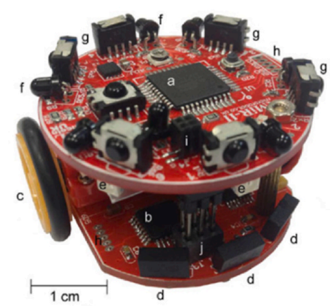
(d) The Pheeno core module [184].

Fig. 21. (a) Kilobots; (b) Droplet; (c) e-puck; (d) Pheeno.



(a) Mona robot and its expansion [185].

(b) The Khepera IV robot [186].



(c) Colias mobile robot [187].

Fig. 22. (a) Mona; (b) Khepera; (c) Colias.

mobile platform tailored for studying self-organized behaviors, see Fig. 23.

(c) Multi-quadrotor testbeds:

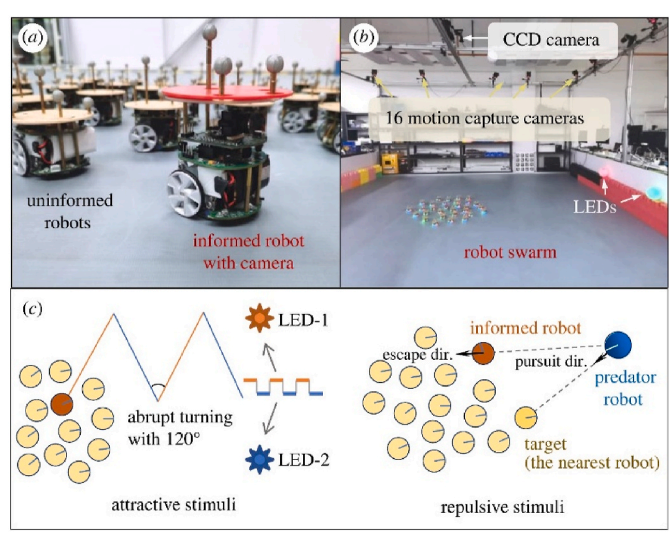


Fig. 23. Experimental setup of a swarm robotic system utilizing SwarmBang robots [190].

Deng et al. [191] demonstrated swarm control using quadrotors equipped with Raspberry Pi 4B processors, XBee radios, and CubeOrange autopilots, operating under ROS. Their physical setup, shown in Fig. 24, validated both low-level behaviors (e.g., aggregation, pattern formation) and high-level tasks (e.g., collective foraging, transport).

In summary, each platform offers distinct advantages, and the choice depends on the swarm's requirements, such as scale, complexity, and budget. Despite challenges posed by hardware limitations and environmental noise, these platforms remain essential for gaining practical insights into swarm behavior and performance. Table 6 shows a comparison of some popular swarm robotics hardware platforms in terms of their features, capabilities, and limitations.

6. Swarm intelligence application in space and future directions

6.1. Swarm intelligence applications in space

Swarm intelligence has emerged as a pivotal framework for addressing complex space mission challenges by optimizing operations through decentralized, scalable, and fault-tolerant agent coordination. Applications span on-orbit servicing and debris removal, satellite swarms and formation flying, autonomous rendezvous, and emerging directions in deep learning, motion planning, and distributed networking [192].

6.1.1. On-orbit servicing and debris removal

Swarm intelligence algorithms such as Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) have been widely applied to mission planning and optimization. PSO is extensively applied to trajectory design, enabling efficient debris capture with minimal fuel [90]. GWO excels in supporting dynamic task allocation through hierarchical search strategies [95]. ACO is ideal for pheromone-inspired path planning for debris removal [91]. ABC improves resource allocation and sensor deployment [92]. These algorithms are computationally efficient, robust, and capable of handling high-dimensional and dynamic optimization problems, making them well-suited for space applications.

6.1.2. Satellite swarms and formation flying

Swarm intelligence has also been applied to decentralized constellation design and precision formation flying for optimizing Earth observation, communication, and environmental monitoring. Yang et al. developed a constellation design method based on imaging swath, subgroup formation flying, and global payload coverage [193]. The Starling Formation-Flying Optical Experiment (StarFOX) is the first in-space test of autonomous, angles-only navigation with four CubeSats in formation in low Earth orbit [194]. The proposed Virtual Super Optics Reconfigurable Swarm (VISORS) mission is planned to use two 6U CubeSats, spaced 40 m apart, to capture high-resolution extreme ultraviolet images of active solar regions [195].

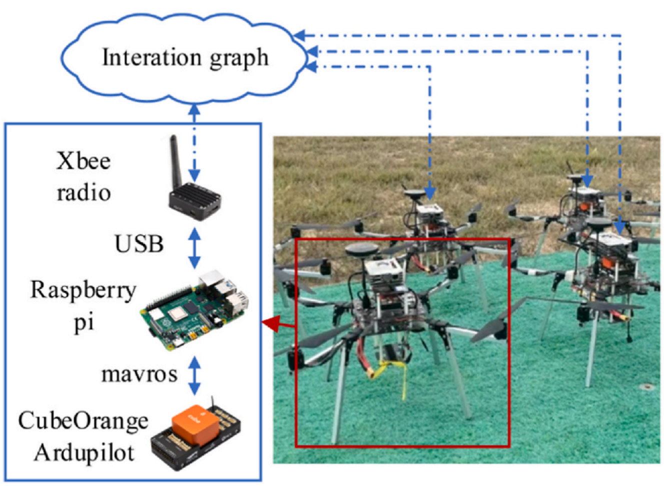


Fig. 24. Multi-quadrotor physical platform [191].

6.1.3. Autonomous rendezvous and debris management

Swarm intelligence can play a key role in facilitating coordinated rendezvous with non-cooperative targets in space debris management and satellite servicing [23,196]. Mahendrakar et al. developed the Multipurpose Autonomous Rendezvous Vision-Integrated Navigation system (MARVIN), which utilizes a swarm of drones to rendezvous with non-cooperative space objects autonomously [197]. Tsukamoto et al. introduced Neural-Rendezvous, a deep learning framework for real-time, autonomous, and accurate interception of fast-moving objects like Interstellar objects [198]. Although not launched into space, these system and mission concepts are currently actively tested through hardware-in-the-loop experiments using drones and simulation to validate their feasibility.

6.1.4. Communication constraints and proximity-operation challenges

Spacecraft swarms operate under strict communication [199] and proximity-operation constraints that influence coordination and control. Limited bandwidth, intermittent line-of-sight, and antenna-pointing restrictions introduce delays and packet loss, reducing the consistency of shared state estimates and slowing consensus convergence; when control actions rely on outdated neighbor information, distributed formation keeping and cooperative planning can become unstable. These communication challenges are compounded during close-proximity operations [200] such as formation flight and on-orbit servicing, where spacecraft must account for differential orbital motion, gravitational perturbation, sensor uncertainty, and plume impingement. Small navigation errors or delayed responses can rapidly escalate into collision risk, requiring conservative safety margins, precise relative navigation, and fault-tolerant control laws capable of reacting quickly to disturbances.

6.2. Prospective research directions

The current research on swarm intelligence focuses on optimizing distributed target-capture strategies for unmanned aerial and surface vehicles. While these studies demonstrate scalability and robustness, a significant gap remains in translating swarm intelligence algorithms to space robotics, particularly in active on-orbit servicing and debris removal [192].

Despite their potential, swarm intelligence systems for on-orbit servicing and debris removal face several challenges, including communication constraints, limited autonomy, and the need for real-time adaptation. Limited bandwidth and communication delays in space can hinder swarm coordination, necessitating robust communication protocols. Increasing the autonomy of swarm members will reduce reliance on ground control and improve mission efficiency. Additionally, swarm algorithms must adapt to dynamic environments in real-time, such as unexpected debris movements or satellite failures. Future research should prioritize the development of advanced communication systems, the enhancement of autonomous capabilities, and the improvement of real-time decision-making to fully unlock the potential of swarm intelligence in space applications.

7. Conclusion

Swarm intelligence offers a transformative framework for addressing the complexities of space robotics, enabling scalable, adaptive, and resilient multi-agent coordination. This review has examined the fundamental dynamic models of swarm agents, including continuous-time and discrete-time systems, which underpin the design of effective

Table 6
Comparison of swarm robotics hardware platforms.

Platform	Locomotion	Comm.	Sensors	Processing	$/robot	Swarm Size	User Cases
Kilobot	Vibration	IR	Ambient light	8-bit ATmega	~$14	1000+	Large-scale swarm experiments
e-puck	Wheeled	Bluetooth, IR	Camera, Proximity, accelerometer	32-bit dsPIC	~$80	10-50	Education, vision-based tasks
Mona	Omni directional	Wi-Fi, IR	Proximity, IMU, RGB LEDs	ARM Cortex-M4	~$500	10–100	Modular swarm robotics
Colias	Wheeled	IR	Proximity, ambient light	ATmega328P	~$50	50–100	Low-cost swarm experiments
Pheeno	Wheeled	Wi-Fi, IR	Proximity, IMU, camera (opt.)	Raspberry Pi + Arduino	~$200	10–50	Multi-robot systems, education
Droplet	Vibration	IR	Ambient light	8-bit ATmega	~$20	100+	Large-scale swarm experiments
Khepera IV	Wheeled	Wi-Fi, IR	Camera, proximity, IMU	ARM Cortex-A8	~ $2000	10–20	Advanced swarm research, vision-based tasks

swarm control strategies. Core control mechanisms, such as aggregation, social foraging, formation control, and distributed consensus, were highlighted for their role in achieving robust, decentralized coordination. Aggregation supports self-organization, while social foraging optimizes task allocation. Formation control, approached through virtual structures, leader–follower frameworks, and decentralized schemes, ensures spatial organization, with decentralized strategies offering particular advantages in robustness and scalability. Consensus-seeking algorithms further enhance collaboration by enabling distributed agreement without centralized control. Scalability and optimality remain central challenges for large-scale swarms operating in high-dimensional environments. Multi-objective optimization methods are crucial for balancing competing goals, including fuel efficiency, mission success, and collision avoidance. Numerical simulations and experimental studies confirm the practical value of these approaches, demonstrating their effectiveness in realistic mission contexts.

Looking ahead, advancing autonomy, enabling real-time adaptation, and strengthening inter-agent communication will be essential to fully realize the potential of swarm systems. By overcoming these challenges, swarm intelligence can drive significant progress in space robotics, supporting sustainable and efficient operations in on-orbit servicing, debris removal, and planetary exploration.

CRediT authorship contribution statement

Zixuan Zhang: Formal analysis, Investigation, Validation, Visualization, Writing – original draft. **Zheng H. Zhu:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

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.

Acknowledgements

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

Data availability

No data was used for the research described in the article.

References

[1] L. Bayındır, A review of swarm robotics tasks, Neurocomputing 172 (2016) 292–321, https://doi.org/10.1016/j.neucom.2015.05.116.
[2] A.E. Ezugwu, A.M. Ikotun, O.O. Oyelade, L. Abualigah, J.O. Agushaka, C.I. Eke, A.A. Akinyelu, A comprehensive survey of clustering algorithms: state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, Eng. Appl. Artif. Intell. 110 (2022) 104743, https://doi.org/10.1016/j.engappai.2022.104743.
[3] I. Sample, Robot swarms: scientists work to harness hive mentality, Guardian (2015). https://www.theguardian.com/technology/2015/sep/18/robot-swarms-drone-scientists-hive-mentality. (Accessed 27 August 2025).
[4] S.A. Curtis, J. Mica, J. Nuth, G. Marr, M. Rilee, M. Bhat, ANTS (autonomous nano technology swarm): an artificial intelligence approach to asteroid belt resource exploration, in: Proc. 51st Int. Astronautical Congress (IAC), Int. Astronautical Federation, Rio de Janeiro, Brazil, 2000.
[5] P. D'Arrigo, S. Santandrea, The APIES mission to explore the asteroid belt, Adv. Space Res. 38 (2006) 2060–2067, https://doi.org/10.1016/j.asr.2006.09.017.
[6] G. Bonnet, C. Tessier, Multi-agent collaboration: a satellite constellation case, in: A. Cesta, N. Fakotakis (Eds.), STAIRS 2008, IOS Press, Amsterdam, 2008, pp. 24–35, https://doi.org/10.3233/978-1-58603-893-9-24.
[7] L. Pedroso, P. Batista, Distributed decentralized receding horizon control for very large-scale networks with application to satellite mega-constellations, Control Eng. Pract. 141 (2023) 105728, https://doi.org/10.1016/j.conengprac.2023.105728.
[8] S.-J. Chung, A.A. Paranjape, P. Dames, S. Shen, V. Kumar, A survey on aerial swarm robotics, IEEE Trans. Robot. 34 (2018) 837–855, https://doi.org/10.1109/TRO.2018.2857475.
[9] T.E. Carter, J. Brient, Linearized impulsive rendezvous problem, J. Optim. Theor. Appl. 86 (1995) 553–584, https://doi.org/10.1007/BF02192159.
[10] Y. Takahashi, H. Sakamoto, S. Sakai, Simultaneous control of relative position and absolute attitude for electromagnetic spacecraft swarm, in: AIAA SCITECH 2021 FORUM, 2021.
[11] P. Di, Y. Yao, Z. Lin, Z. Yin, Trajectory optimization of spacecraft autonomous far-distance rapid rendezvous based on deep reinforcement learning, Adv. Space Res. 75 (2025) 790–806, https://doi.org/10.1016/j.asr.2024.09.066.
[12] S. Starin, A. Sparks, R. Yedavalli, Spacecraft formation flying maneuvers using linear-quadratic regulation with no radial axis inputs, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, 2001, p. 4029.
[13] H.-T. Liu, J. Shan, D. Sun, Adaptive Synchronization Control of Multiple Spacecraft Formation Flying, J. Dyn. Syst. Meas. Control 129 (2007) 337–342, https://doi.org/10.1115/1.2719772.
[14] W. Ren, R.W. Beard, Decentralized scheme for spacecraft formation flying via the virtual structure approach, J. Guid. Control Dynam. 27 (2004) 73–82, https://doi.org/10.2514/1.9287.
[15] Z. Zhao, H. Zuo, Q. Shen, W. Wang, Event-triggered distributed model predictive control for spacecraft formation with collision avoidance, IFAC-PapersOnLine 59 (2025) 905–910, https://doi.org/10.1016/j.ifacol.2025.11.268.
[16] K. Matsuka, E.S. Lupu, Y.K. Nakka, R. Foust, S.-J. Chung, F. Hadaegh, Distributed multi-target relative pose estimation for cooperative spacecraft swarm, in: Proc. 10th Int. Workshop on Satellite Constellations and Formation Flying (IWSCFF), Univ. Strathclyde, Glasgow, UK, 2019, pp. 16–19.
[17] V. Gazi, K.M. Passino, Swarm Stability and Optimization, first ed., Springer, Berlin, 2011.
[18] Y. Takubo, W. Manuel, E. Foss, S. D'Amico, Safe and optimal N-Spacecraft swarm reconfiguration in non-keplerian cislunar orbits, J. Guid. Control Dynam. (2025) 1–20, https://doi.org/10.2514/1.G009290.
[19] E.G. Asri, Z.H. Zhu, An introductory review of swarm technology for spacecraft on-orbit servicing, Int. J. Mech. Syst. Dyn. 4 (2024) 3–21, https://doi.org/10.1002/msd2.12098.
[20] X. Bai, M. Xu, Phase-space density control based on configuration invariant of spacecraft swarm, Adv. Space Res. 75 (2025) 2271–2288, https://doi.org/10.1016/j.asr.2024.10.033.
[21] J. Wu, M. Liu, Y. Wang, X. Cao, Event-trigger-based cluster coordinated control of spacecraft swarm under switching topology, Aero. Sci. Technol. 135 (2023) 108200, https://doi.org/10.1016/j.ast.2023.108200.

[22] A. Rahmani, M. Ji, M. Mesbahi, M. Egerstedt, Controllability of multi-agent systems from a graph-theoretic perspective, SIAM J. Control Optim. 48 (2009) 162–186, https://doi.org/10.1137/060674909.
[23] 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.
[24] M.A.U. Rahman, M. Schranz, LLM-powered swarms: a new frontier or a conceptual stretch?, ArXiv Preprint, https://doi.org/10.48550/arXiv.2506.14496, 2025.
[25] P. Li, Z. An, S. Abrar, L. Zhou, Large language models for multi-robot systems: a survey, ArXiv Preprint, https://doi.org/10.48550/arXiv.2502.03814, 2025.
[26] R. Lent, R. Dudukovich, A. Gannon, R. Short, Applying the cognitive space gateway to swarm topologies, in: 2021 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW), IEEE, 2021, pp. 1–7, https://doi.org/10.1109/CCAAW50069.2021.9527307.
[27] M. Schranz, G.A. Di Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. Şekercioğlu, M. Sende, Swarm intelligence and cyber-physical systems: concepts, challenges and future trends, Swarm Evol. Comput. 60 (2021) 100762, https://doi.org/10.1016/j.swevo.2020.100762.
[28] J. Chen, B. Wu, Z. Sun, Computationally efficient distributed model predictive control of satellite swarms, Adv. Space Res. 77 (2026) 2183–2202, https://doi.org/10.1016/j.asr.2025.11.014.
[29] J. Chen, B. Wu, Z. Sun, D. Wang, Distributed safe trajectory optimization for large-scale spacecraft formation reconfiguration, Acta Astronaut. 214 (2024) 125–136, https://doi.org/10.1016/j.actaastro.2023.10.012.
[30] S. Lu, Y. Zhang, Q. Hu, Swarms of modular satellites decentralized guidance and target assignment strategy, Adv. Space Res. 72 (2023) 3787–3801, https://doi.org/10.1016/j.asr.2023.07.027.
[31] P. Skobelev, E. Simonova, V. Galuzin, A. Galitskaya, V. Travin, Swarm of satellites: implementation and experimental study of multi-agent solution for adaptive scheduling of large-scale space observation systems, 267–278, https://doi.org/10.1007/978-3-030-85739-4_22, 2021.
[32] C. Adams, B. Kempa, W. Vaughan, N. Cramer, Development of a high-performance, heterogenous, scalable test-bed for distributed spacecraft, in: 2023 IEEE Aerospace Conference, IEEE, 2023, pp. 1–8, https://doi.org/10.1109/AERO55745.2023.10115695.
[33] K. Lerman, A. Galstyan, Mathematical model of foraging in a group of robots: effect of interference, Auton. Robots 13 (2002) 127–141, https://doi.org/10.1023/A:1019633424543.
[34] J.P. Hecker, M.E. Moses, Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms, Swarm Intell. 9 (2015) 43–70, https://doi.org/10.1007/s11721-015-0104-z.
[35] J. Harwell, M. Gini, Swarm engineering through quantitative measurement of swarm robotic principles in a 10,000 robot swarm, in: S. Kraus (Ed.), Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI-19), IJCAI, Macao, China, 2019, pp. 336–342.
[36] S. Adhikari, Study of Scalability in a Robot Swarm Performance and Demonstration of Superlinear Performance in Conveyor Bucket Bridges and Collaborative Pulling, University of Toledo, 2021. PhD thesis.
[37] H. Hamann, A. Reina, Scalability in computing and robotics, IEEE Trans. Comput. 71 (2022) 1453–1465, https://doi.org/10.1109/TC.2021.3089044.
[38] G.M. Amdahl, Validity of the single processor approach to achieving large scale computing capabilities, in: Proc. Spring Joint Computer Conf. (AFIPS '67 Spring), ACM Press, New York, USA, 1967, pp. 483–485, https://doi.org/10.1145/1465482.1465560.
[39] J.L. Gustafson, Reevaluating Amdahl's law, commun, ACM 31 (1988) 532–533, https://doi.org/10.1145/42411.42415.
[40] N.J. Gunther, A simple capacity model of massively parallel transaction systems, in: Proc. Int. CMG Conf., Computer Measurement Group, San Diego, CA, USA, 1993, pp. 1–9.
[41] E. Milner, M. Sooriyabandara, S. Hauert, Swarm performance indicators: metrics for robustness, fault tolerance, scalability and adaptability, ArXiv Preprint (2023), https://doi.org/10.48550/arXiv.2311.01944.
[42] Y. Li, Satellites swarm cooperation for pursuit-attachment tasks with transformer-based reinforcement learning, ArXiv Preprint (2024), https://doi.org/10.48550/arXiv.2406.01061.
[43] 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.
[44] D. Izzo, L. Pettazzi, Autonomous and distributed motion planning for satellite swarm, J. Guid. Control Dynam. 30 (2007) 449–459, https://doi.org/10.2514/1.22736.
[45] C. Pinciroli, M. Birattari, E. Tuci, M. Dorigo, M. del R. Zapatero, T. Vinko, D. Izzo, Self-organizing and scalable shape formation for a swarm of pico satellites, in: 2008 NASA/ESA Conference on Adaptive Hardware and Systems, IEEE, 2008, pp. 57–61, https://doi.org/10.1109/AHS.2008.41.
[46] C. Liu, Y. Luo, X. Yue, S. Li, Hierarchical cooperative adaptive model predictive control for swarm self-assembly of large-scale spacecraft, Aero. Sci. Technol. 170 (2026) 111538, https://doi.org/10.1016/j.ast.2025.111538.
[47] X. Bai, J. Yuan, M. Xu, Density evolution control in large-scale spacecraft swarms via physics-informed deep learning, Aero. Sci. Technol. 169 (2026) 111458, https://doi.org/10.1016/j.ast.2025.111458.
[48] H.M. Zangana, Z.B. Sallow, M.H. Alkawaz, M. Omar, Unveiling the collective wisdom: a review of swarm intelligence in problem solving and optimization, Inform : Jurnal Ilmiah Bidang Teknologi Informasi Dan Komunikasi 9 (2024) 101–110, https://doi.org/10.25139/inform.v9i2.7934.
[49] M. Haque, C. McGowan, Y. Guo, D. Kirkpatrick, J.A. Adams, Communication model-task pairing in artificial swarm design, ArXiv Preprint (2018), https://doi.org/10.48550/arXiv.1801.07327.
[50] J. Harwell, M. Gini, Improved swarm engineering: aligning intuition and analysis, ArXiv Preprint, https://doi.org/10.48550/arXiv.2012.04144, 2021.
[51] G. Tb, R. AL, Embedded systems for space exploration and satellite communication, World J. Adv. Res. Rev. 11 (2021) 279–290, https://doi.org/10.30574/wjarr.2021.11.1.0341.
[52] B. Hine, B. Wickizer, H. Cannon, Spacecraft swarm missions. Moffett Field, 2025. https://ntrs.nasa.gov/api/citations/20250001985/downloads/Spacecraft_Swarm_Missions_v2.pdf. (Accessed 3 February 2026).
[53] S.W. Paek, S. Kim, O. de Weck, Optimization of reconfigurable satellite constellations using simulated annealing and genetic algorithm, Sensors 19 (2019) 765, https://doi.org/10.3390/s19040765.
[54] P.G. Buzzi, D. Selva, N. Hitomi, W.J. Blackwell, Assessment of constellation designs for Earth observation: application to the TROPICS mission, Acta Astronaut. 161 (2019) 166–182, https://doi.org/10.1016/j.actaastro.2019.05.007.
[55] A. Balachandran, N. Harasha, N. Lynch, Swarm algorithms for dynamic task allocation in unknown environments, ArXiv Preprint (2024), https://doi.org/10.48550/arXiv.2409.09550.
[56] A.T. Takano, B.G. Marchand, Optimal constellation design for space based situational awareness applications, in: H. Schaub, B.C. Gunter, R.P. Russell, W. T. Cerven (Eds.), ASTRODYNAMICS 2011, PTS I - IV, 2012, pp. 2139–2158.
[57] E. Larkin, M. Antonov, On assessing the temporal characteristics of reaching the milestone by a swarm, in: Proc. Int. Conf. (Springer LNCS), Springer, Cham, 2020, pp. 46–55, https://doi.org/10.1007/978-3-030-53956-6_5.
[58] G. Duflo, G. Danoy, E.-G. Talbi, P. Bouvry, Learning to optimise a swarm of UAVs, Appl. Sci. 12 (2022) 9587, https://doi.org/10.3390/app12199587.
[59] L.M. Sauter, R.J. Proulx, H. Musoff, Satellite constellation design for mid-course ballistic missile intercept, in: S.L. Coffey, S.L. Coffey, A.P. Mazzoleni, K.K. Luu, R. A. Glover (Eds.), Spaceflight Mechanics, 119, 2004, pp. 3087–3105. Pt 1-3, 2005.
[60] J. Wang, J. Zhu, W. Zhao, J. Jiang, X. Bai, M. Xu, Quick coverage analysis method for mega constellation based on high-order expansion, J. Guid. Control Dynam. 48 (2025) 2527–2540, https://doi.org/10.2514/1.G008954.
[61] T. Yasuda, M. Bando, S. Hokamoto, Resilient distributed coverage control of satellite swarms via local voronoi feedback, Adv. Space Res. (2025), https://doi.org/10.1016/j.asr.2025.11.029.
[62] Y.T. dos Passos, X. Duquesne, L.S. Marcolino, On the throughput of the common target area for robotic swarm strategies – extended version, ArXiv Preprint (2022), https://doi.org/10.48550/arXiv.2201.09335.
[63] R. Ronen, B. Ben-Moshe, Maximizing nanosatellite throughput via dynamic scheduling and distributed ground stations, Sensors 25 (2025) 7538, https://doi.org/10.3390/s25247538.
[64] D. Morgan, S.-J. Chung, L. Blackmore, B. Acikmese, D. Bayard, F.Y. Hadaegh, Swarm-keeping strategies for spacecraft under J2 and atmospheric drag perturbations, J. Guid. Control Dynam. 35 (2012) 1492–1506, https://doi.org/10.2514/1.55705.
[65] G. Kang, Z. Yang, X. Yuan, J. Wu, Fault reconstruction for a giant satellite swarm based on hybrid multi-objective optimization, Appl. Sci. 13 (2023) 6674, https://doi.org/10.3390/app13116674.
[66] C. Lippe, S. D'Amico, Optimal spacecraft swarm reconfiguration through chief orbit refinement, Acta Astronaut. 183 (2021) 162–175, https://doi.org/10.1016/j.actaastro.2021.03.011.
[67] A. Mahfouz, G. Gaias, F. Dalla Vedova, H. Voos, Delta-V-optimal centralized guidance strategy for under-actuated N-satellite formations, Acta Astronaut. 228 (2025) 69–87, https://doi.org/10.1016/j.actaastro.2024.11.048.
[68] A.W. Koenig, S. D'Amico, Robust and safe N-Spacecraft swarming in perturbed near-circular orbits, J. Guid. Control Dynam. 41 (2018) 1643–1662, https://doi.org/10.2514/1.G003249.
[69] H. Liu, Y. Tian, F.L. Lewis, Y. Wan, K.P. Valavanis, Robust formation flying control for a team of satellites subject to nonlinearities and uncertainties, Aero. Sci. Technol. 95 (2019) 105455, https://doi.org/10.1016/j.ast.2019.105455.
[70] Y. Wang, X. Wang, N. Cui, Robust decentralised state estimation for formation flying spacecraft, IET Radar Sonar Navig. 13 (2019) 814–823, https://doi.org/10.1049/iet-rsn.2018.5348.
[71] M. Joyner, L. Plice, Active swarm resiliency in the HelioSwarm mission, in: 37th Annual Small Satellite Conference, 2023.
[72] V. Mann, A. Sivaram, L. Das, V. Venkatasubramanian, Robust and efficient swarm communication topologies for hostile environments, Swarm Evol. Comput. 62 (2021) 100848, https://doi.org/10.1016/j.swevo.2021.100848.
[73] J. Czyzowicz, S. Dobrev, E. Kranakis, E. Pacheco, Survivability of swarms of bouncing robots, in: Proc. Int. Conf. (Springer LNCS), Springer, Cham, 2014, pp. 622–633, https://doi.org/10.1007/978-3-642-54423-1_54.
[74] J.L.S. Cotta, O. Qasem, P. do V. Pereira, H. Gutierrez, Control of small spacecraft by optimal output regulation: a reinforcement learning approach, ArXiv Preprint (2023), https://doi.org/10.48550/arXiv.2307.09428.
[75] C. Lippe, S. D'Amico, Safe, Delta-v-Efficient spacecraft swarm reconfiguration using lyapunov stability and artificial potentials, J. Guid. Control Dynam. 45 (2022) 213–231, https://doi.org/10.2514/1.G006253.
[76] H. Chen, Y. Zhu, K. Hu, X. He, Hierarchical swarm model: a new approach to optimization, Discrete Dynam Nat. Soc. 2010 (2010), https://doi.org/10.1155/2010/379649.
[77] M.N. Omidvar, X. Li, X. Yao, A review of population-based metaheuristics for large-scale black-box global optimization—part I, IEEE Trans. Evol. Comput. 26 (2022) 802–822, https://doi.org/10.1109/TEVC.2021.3130838.

[78] J. Ding, X. Bai, M. Xu, Homotopy method with initial costate evaluation for low-thrust fuel-optimal homoclinic halo orbit transfer, J. Aero. Eng. 38 (2025), https://doi.org/10.1061/JAEEEZ.ASENG-5901.

[79] A.O. de Sá, N. Nedjah, L. de M. Mourelle, Distributed efficient localization in swarm robotics using min–max and particle swarm optimization, Expert Syst. Appl. 50 (2016) 55–65, https://doi.org/10.1016/j.eswa.2015.12.007.

[80] R. Chen, M. Dong, Y. Bai, Y. Zhao, X. Chen, Trajectory planning and control of spacecraft avoiding dynamic debris swarm, Aero. Sci. Technol. 151 (2024) 109273, https://doi.org/10.1016/j.ast.2024.109273.

[81] M. Ehresmann, G. Herdrich, S. Fasoulas, An automated system analysis and design tool for spacecrafts, CEAS Space J 14 (2022) 327–354, https://doi.org/10.1007/s12567-021-00383-3.

[82] D. Morgan, S.-J. Chung, F.Y. Hadaegh, Model predictive control of swarms of spacecraft using sequential convex programming, J. Guid. Control Dynam. 37 (2014) 1725–1740, https://doi.org/10.2514/1.G000218.

[83] C. Liu, Y. Luo, X. Yue, E. Capello, A hierarchical optimization approach for spacecraft swarm trajectories considering dynamic communication topologies, Aero. Sci. Technol. 168 (2026) 111250, https://doi.org/10.1016/j.ast.2025.111250.

[84] J. Oyekan, Multi-objective optimisation of robotic active particle swarms for continuous repair of large scale high value structures, in: Proc. IEEE Congr. Evol. Comput. (CEC 2021), IEEE, Kraków, Poland, 2021, pp. 1312–1318, https://doi.org/10.1109/CEC45853.2021.9504749.

[85] Q. Yin, G. Wu, G. Sun, Y. Gu, Multi-objective orbital maneuver optimization of multi-satellite using an adaptive feedback learning NSGA-II, Swarm Evol. Comput. 93 (2025) 101835, https://doi.org/10.1016/j.swevo.2024.101835.

[86] A. Hoskins, E. Atkins, Satellite formation design with a multi-objective optimization technique, in: AIAA/AAS Astrodynamics Specialist Conference and Exhibit, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2006, https://doi.org/10.2514/6.2006-6013.

[87] D.H. Stolfi, G. Danoy, Spacecraft swarm orbital formation optimisation using evolutionary techniques, in: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2023, pp. 771–774, https://doi.org/10.1145/3583133.3590651.

[88] Z. Hao, X. Yue, H. Wen, C. Liu, Full-state-constrained non-certainty-equivalent adaptive control for satellite swarm subject to input fault, IEEE/CAA J. Autom. Sin. 9 (2022) 482–495, https://doi.org/10.1109/JAS.2021.1004216.

[89] H. Basu, Y. Pedari, M. Almassalkhi, H.R. Ossareh, Computationally efficient collision-free trajectory planning of satellite swarms under unmodeled orbital perturbations, J. Guid. Control Dynam. (2023) 1–16, https://doi.org/10.2514/1.G007206.

[90] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proc. Int. Conf. Neural Netw. (ICNN '95), IEEE, Perth, Australia, 1995, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968.

[91] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B. 26 (1996) 29–41, https://doi.org/10.1109/3477.484436.

[92] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471, https://doi.org/10.1007/s10898-007-9149-x.

[93] O. Watanabe, T. Zeugmann (Eds.), Stochastic Algorithms: Foundations and Applications, Springer, Berlin, Heidelberg, 2009, https://doi.org/10.1007/978-3-642-04944-6.

[94] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: J.R. González, D. A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, Berlin, Heidelberg, 2010, pp. 65–74, https://doi.org/10.1007/978-3-642-12538-6_6.

[95] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (2014) 46–61, https://doi.org/10.1016/j.advengsoft.2013.12.007.

[96] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Software 95 (2016) 51–67, https://doi.org/10.1016/j.advengsoft.2016.01.008.

[97] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Software 114 (2017) 163–191, https://doi.org/10.1016/j.advengsoft.2017.07.002.

[98] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (2016) 1053–1073, https://doi.org/10.1007/s00521-015-1920-1.

[99] G. Dhiman, M. Garg, A. Nagar, V. Kumar, M. Dehghani, A novel algorithm for global optimization: rat swarm optimizer, J. Ambient Intell. Hum. Comput. 12 (2021) 8457–8482, https://doi.org/10.1007/s12652-020-02580-0.

[100] L. Xie, T. Han, H. Zhou, Z.-R. Zhang, B. Han, A. Tang, Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization, Comput. Intell. Neurosci. 2021 (2021), https://doi.org/10.1155/2021/9210050.

[101] Big O notation, (n.d.). https://en.wikipedia.org/wiki/Big_O_notation (accessed March 18, 2025).

[102] C.M. Saaj, V. Lappas, D. Richie, H. Schaub, D. Izzo, Hybrid propulsion system for spacecraft swarm aggregation using coulomb force, J. Br. Interplanet. Soc. (JBIS) 60 (2007) 268–274.

[103] Z. Sun, B. Wu, J. Chen, Distributed event-triggered model predictive control for spacecraft swarm, IEEE Trans. Aero. Electron. Syst. (2024) 1–20, https://doi.org/10.1109/TAES.2024.3485601.

[104] P. Oikonomou, S. Pappas, Decentralized bioinspired non-discrete model for autonomous swarm aggregation dynamics, Appl. Sci. 10 (2020) 1067, https://doi.org/10.3390/app10031067.

[105] C.M. Saaj, V. Lappas, H. Schaub, D. Izzo, Hybrid propulsion system for formation flying using electrostatic forces, Aero. Sci. Technol. 14 (2010) 348–355, https://doi.org/10.1016/j.ast.2010.02.009.

[106] V. Gazi, B. Fidan, Y.S. Hanay, M.I. Köksal, Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques, Turk. J. Electr. Eng. Comput. Sci. 15 (2007) 149–168.

[107] M.S. Talamali, T. Bose, M. Haire, X. Xu, J.A.R. Marshall, A. Reina, Sophisticated collective foraging with minimalist agents: a swarm robotics test, Swarm Intell. 14 (2020) 25–56, https://doi.org/10.1007/s11721-019-00176-9.

[108] J. Tan, N. Melkoumian, D. Harvey, R. Akmeliawati, Lunarminer framework for nature-inspired swarm robotics in lunar water ice extraction, Biomimetics 9 (2024) 680, https://doi.org/10.3390/biomimetics9110680.

[109] M. Cooper, B. Smeresky, An overview of evolutionary algorithms toward spacecraft attitude control, in: Advances in Spacecraft Attitude Control, IntechOpen, 2020, https://doi.org/10.5772/intechopen.89637.

[110] S. Nag, L. Summerer, Behaviour based, autonomous and distributed scatter manoeuvres for satellite swarms, Acta Astronaut. 82 (2013) 95–109, https://doi.org/10.1016/j.actaastro.2012.04.030.

[111] J. Kruger, S. D'Amico, Autonomous angles-only multitarget tracking for spacecraft swarms, Acta Astronaut. 189 (2021) 514–529, https://doi.org/10.1016/j.actaastro.2021.08.049.

[112] Y. Cui, Y. Chen, D. Yang, Z. Shu, T. Huang, X. Gong, Resilient formation tracking of spacecraft swarm against actuation attacks: a distributed lyapunov-based model predictive approach, IEEE Trans. Syst. Man Cybern. Syst 53 (2023) 7053–7065, https://doi.org/10.1109/TSMC.2023.3292426.

[113] L. Schenato, Songhwai Oh, S. Sastry, P. Bose, Swarm coordination for pursuit evasion games using sensor networks, in: Proc. IEEE Int. Conf. Robot. Autom. (ICRA), IEEE, Barcelona, Spain, 2005, pp. 2493–2498, https://doi.org/10.1109/ROBOT.2005.1570487.

[114] Z. Mu, M. Ji, P. Guo, Q. Zhang, B. Xiao, L. Cao, J. Yu, Pursuit-evasion game for spacecraft with incomplete information under J_2 perturbation, IEEE Trans. Circ. Syst. I 72 (2025) 4346–4358, https://doi.org/10.1109/TCSI.2025.3560303.

[115] Q. Wang, P. Li, T. Lei, X. Liu, G. Cai, A fuel optimization method for the pursuer in the spacecraft pursuit-evasion game, J. Comput. Nonlinear Dynam. 17 (2022), https://doi.org/10.1115/1.4053669.

[116] Z. Li, Y. Luo, Orbital impulsive pursuit–evasion game formulation and stackelberg equilibrium solutions, J. Spacecraft Rockets 62 (2025) 349–364, https://doi.org/10.2514/1.A35956.

[117] H. He, P. Shi, Y. Zhao, Hierarchical optimization algorithm and applications of spacecraft trajectory optimization, Aerospace 9 (2022) 81, https://doi.org/10.3390/aerospace9020081.

[118] J.-G. Park, Y. Kim, T.-H. Kim, Distributed coordination D-Stabilization in cyclic pursuit formations of dynamical multi-agent systems, Actuators 13 (2024) 495, https://doi.org/10.3390/act13120495.

[119] T.-H. Kim, T. Sugie, Cooperative control for target-capturing task based on a cyclic pursuit strategy, Automatica 43 (2007) 1426–1431, https://doi.org/10.1016/j.automatica.2007.01.018.

[120] D. Mukherjee, D. Ghose, Target capturability using agents in cyclic pursuit, J. Guid. Control Dynam. 39 (2016) 1034–1045, https://doi.org/10.2514/1.G001495.

[121] Y. Chen, Y. Jiao, M. Wu, H. Ma, Z. Lu, Group target tracking for highly maneuverable unmanned aerial vehicles swarms: a perspective, Sensors 23 (2023) 4465, https://doi.org/10.3390/s23094465.

[122] C. Zhou, Y. Qiao, X. Huang, S. Li, L. Ma, Safe and robust satellite formation control via dynamic constraint following with input saturation, Aero. Sci. Technol. 169 (2026) 111436, https://doi.org/10.1016/j.ast.2025.111436.

[123] M. Williams, Starlink's satellites will be orbiting at a much lower altitude, reducing the risks of space junk, Phys. Oceanogr. (2019). https://phys.org/news/2019-05-starlink-satellites-orbiting-altitude-space.html. (Accessed 1 February 2026).

[124] S. Bandyopadhyay, R. Foust, G.P. Subramanian, S.-J. Chung, F.Y. Hadaegh, Review of formation flying and constellation missions using nanosatellites, J. Spacecraft Rockets 53 (2016) 567–578, https://doi.org/10.2514/1.A33291.

[125] Q. Chen, H. Jin, C. Wei, Y. Liao, Distributed impulsive satellite swarm-keeping based on connectivity and relative cartesian states, J. Guid. Control Dynam. 48 (2025) 269–281, https://doi.org/10.2514/1.G008144.

[126] H. Qian, Z. Chen, X. Wang, B. Xiao, L. Meng, Y. Ma, A swarm-independent behaviors-based orbit maneuvering approach for target-attacker-defender games of satellites, Inf. Sci. 699 (2025) 121790, https://doi.org/10.1016/j.ins.2024.121790.

[127] M. Freese, S. Singh, F. Ozaki, N. Matsuhira, Virtual robot experimentation platform V-REP: a versatile 3D robot simulator, in: N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, O. VonStryk (Eds.), SIMPAR 2010: Simulation, Modeling, and Programming for Autonomous Robots, Springer, Darmstadt, Germany, 2010, pp. 51–62, https://doi.org/10.1007/978-3-642-17319-6_8.

[128] D. Ruggiero, E. Capello, Model predictive control for spacecraft swarm proximity operations, in: Proc. Annu. Conf. Soc. Instrum. Control Eng. (SICE), IEEE, Suzhou, China, 2023, pp. 894–900, https://doi.org/10.23919/SICE59929.2023.10354185.

[129] V.P. Gehlot, M.J. Balas, A theoretical framework for the stable operation of autonomous spacecraft formations in the hill-clohessy-wiltshire frame, in: Proc. IEEE SoutheastCon, IEEE, Huntsville, AL, USA, 2019, pp. 1–7, https://doi.org/10.1109/SoutheastCon42311.2019.9020597.

[130] N. Yielding, J. Curro, S. Cain, Satellite guidance with multi-agent reinforce learning for triangulating a moving object in a relative orbit frame, in: Proc. IEEE/

ION Position, Location Navig. Symp. (PLANS), IEEE, Monterey, CA, USA, 2023, pp. 1311–1321, https://doi.org/10.1109/PLANS53410.2023.10139951.
[131] W. Jin, T. Chen, H. Hu, L. Wang, Spacecraft formation with J2 perturbation via real-time probabilistic swarm guidance framework, IEEE Trans. Aero. Electron. Syst. 61 (2025) 14672–14686, https://doi.org/10.1109/TAES.2025.3585904.
[132] H. Wang, S. Liu, M. Lv, B. Zhang, Two-level hierarchical-interaction-based group formation control for MAV/UAVs, Aerosp 9 (2022) 510, https://doi.org/10.3390/aerospace9090510.
[133] H. Huang, G. Ma, Y. Zhuang, Y. Lv, Optimal spacecraft formation reconfiguration with collision avoidance using particle swarm optimization, Inf. Technol. Control 41 (2012) 143–150, https://doi.org/10.5755/j01.itc.41.2.854.
[134] Y. Liu, J. Gao, C. Liu, F. Zhao, J. Zhao, Reconfigurable formation control of multi-agents using virtual linkage approach, Appl. Sci. 8 (2018) 1109, https://doi.org/10.3390/app8071109.
[135] K.T. Alfriend, S.R. Vadali, P. Gurfil, J.P. How, L.S. Breger, Spacecraft formation flying: dynamics, control, and navigation, in: Spacecraft Formation Flying, Elsevier, Oxford, 2010, pp. 1–11, https://doi.org/10.1016/B978-0-7506-8533-7.00206-2.
[136] B. Shahbazi, M. Malekzadeh, H.R. Koofigar, Robust constrained attitude control of spacecraft formation flying in the presence of disturbances, IEEE Trans. Aero. Electron. Syst. 53 (2017) 2534–2543, https://doi.org/10.1109/TAES.2017.2704160.
[137] H. Cai, S. Guo, H. Gao, A dynamic leader–follower approach for line marching of swarm robots, Unmanned Syst. 11 (2023) 67–82, https://doi.org/10.1142/S2301385023500024.
[138] S. Dey, H. Xu, Intelligent distributed swarm control for large-scale Multi-UAV systems: a hierarchical learning approach, Electronics (Basel) 12 (2022) 89, https://doi.org/10.3390/electronics12010089.
[139] P. Pereira, B.J. Guerreiro, P. Lourenço, Distributed model predictive control method for spacecraft formation flying in a leader–follower formation, IEEE Trans. Aero. Electron. Syst. 59 (2023) 3213–3223, https://doi.org/10.1109/TAES.2022.3224692.
[140] J. Chen, Y. Yu, Z. Zheng, J. Yuan, Distributed composite state observer-based cooperative control for spacecraft swarm, Astrodynamics 9 (2025) 821–835, https://doi.org/10.1007/s42064-024-0243-3.
[141] Z. Feng, M. Liu, L. Zhang, X. Cao, G. Duan, Leader-follower attitude coordination control for spacecraft formations with Gaussian processes, IEEE Trans. Aero. Electron. Syst. 61 (2025) 11571–11585, https://doi.org/10.1109/TAES.2025.3568420.
[142] T. Endo, Y. Morimoto, Leader-follower attitude consensus for a network of boundary-controlled flexible spacecraft, Mech. Eng. J. 12 (2025) 24–414, https://doi.org/10.1299/mej.24-00414.
[143] B. Doerr, R. Linares, P. Zhu, S. Ferrari, Random finite set theory and centralized control of large collaborative swarms, J. Guid. Control Dynam. 44 (2021) 505–521, https://doi.org/10.2514/1.G004861.
[144] D. Ivanov, R. Gondar, U. Monakhova, A. Guerman, M. Ovchinnikov, Electromagnetic uncoordinated control of a ChipSats swarm using magnetorquers, Acta Astronaut. 192 (2022) 15–29, https://doi.org/10.1016/j.actaastro.2021.12.014.
[145] 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.
[146] N. Delurgio, S. D'Amico, Closed-form modeling and control of spacecraft swarms in eccentric orbits, in: Proc. IEEE Aerosp. Conf. 2024, IEEE, Big Sky, 2024, pp. 1–20, https://doi.org/10.1109/AERO58975.2024.10521046. MT, USA.
[147] A. Farrag, S. Othman, T. Mahmoud, A.Y. ELRaffiei, Satellite swarm survey and new conceptual design for Earth observation applications, Egypt, J. Remote Sens. Space Sci 24 (2021) 47–54, https://doi.org/10.1016/j.ejrs.2019.12.003.
[148] A.W. Koenig, S. D'Amico, Safe spacecraft swarm deployment and acquisition in perturbed near-circular orbits subject to operational constraints, Acta Astronaut. 153 (2018) 297–310, https://doi.org/10.1016/j.actaastro.2018.01.037.
[149] Y. Cui, Y. Huang, Q. Zhao, X. Yu, T. Huang, Resilient time-varying formation-tracking of multiagent systems against hybrid attacks with applications to spacecraft formation, IEEE Trans. Cybern. 55 (2025) 625–637, https://doi.org/10.1109/TCYB.2024.3492035.
[150] C. Lippe, S. D'Amico, Spacecraft swarm dynamics and control about asteroids, Adv. Space Res. 67 (2021) 3426–3443, https://doi.org/10.1016/j.asr.2020.06.037.
[151] C. Yang, H. Zhang, W. Fu, Pattern control for large-scale spacecraft swarms in elliptic orbits via density fields, Chin. J. Aeronaut. 35 (2022) 367–379, https://doi.org/10.1016/j.cja.2021.03.020.
[152] A. Din, M. Jabeen, K. Zia, A. Khalid, D.K. Saini, Behavior-based swarm robotic search and rescue using fuzzy controller, Comput. Electr. Eng. 70 (2018) 53–65, https://doi.org/10.1016/j.compeleceng.2018.06.003.
[153] X. Long, F. Gao, Cooperative attack based on small-unit UAV swarms formation with trajectory tracking, J. Intell. Fuzzy Syst. 45 (2023) 2949–2965, https://doi.org/10.3233/JIFS-231180.
[154] J. Zhang, J. Zhang, Q. Zhang, X. Wei, Obstacle avoidance path planning of space robot based on improved particle swarm optimization, Symmetry (Basel). 14 (2022) 938, https://doi.org/10.3390/sym14050938.
[155] D. Lee, K.D. Kumar, M. Sinha, Fault detection and recovery of spacecraft formation flying using nonlinear observer and reconfigurable controller, Acta Astronaut. 97 (2014) 58–72, https://doi.org/10.1016/j.actaastro.2013.12.002.
[156] D. Yu, C.L.P. Chen, C.-E. Ren, S. Sui, Swarm control for self-organized system with fixed and switching topology, IEEE Trans. Cybern. 50 (2020) 4481–4494, https://doi.org/10.1109/TCYB.2019.2952913.
[157] F. Kempf, J. Scharnagl, S. Heil, K. Schilling, Self-organizing control-loop recovery for predictive networked formation control of fractionated spacecraft, Aerospace 9 (2022) 529, https://doi.org/10.3390/aerospace9100529.
[158] L. Quan, L. Yin, T. Zhang, M. Wang, R. Wang, S. Zhong, X. Zhou, Y. Cao, C. Xu, F. Gao, Robust and efficient trajectory planning for formation flight in dense environments, IEEE Trans. Robot. 39 (2023) 4785–4804, https://doi.org/10.1109/TRO.2023.3301295.
[159] C. Venigalla, J.A. Englander, D.J. Scheeres, Multi-objective low-thrust trajectory optimization with robustness to missed thrust events, J. Guid. Control Dynam. 45 (2022) 1255–1268, https://doi.org/10.2514/1.G006056.
[160] S. Liu, W. Tian, J. Shen, B. Li, P. Li, Base position and task assignment optimization concerning productivity and machining performance for multi-robot systems in aerospace manufacturing, Robot. Autonom. Syst. 171 (2024) 104536, https://doi.org/10.1016/j.robot.2023.104536.
[161] X. Wang, J. Wu, X. Wang, Distributed attitude consensus of spacecraft formation flying, J. Syst. Eng. Electron. 24 (2013) 296–302, https://doi.org/10.1109/JSEE.2013.00037.
[162] R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Automat. Control 49 (2004) 1520–1533, https://doi.org/10.1109/TAC.2004.834113.
[163] X. Wang, Y. Hong, Finite-time consensus for multi-agent networks with second-order agent dynamics, in: Proc. IFAC World Congr., Seoul, Korea, 2008, pp. 15185–15190, https://doi.org/10.3182/20080706-5-KR-1001.02568.
[164] X.-H. Wang, H.-B. Ji, Leader-follower consensus for a class of nonlinear multi-agent systems, Int. J. Control Autom. Syst. 10 (2012) 27–35, https://doi.org/10.1007/s12555-012-0104-3.
[165] J. Xi, Z. Shi, Y. Zhong, Output consensus analysis and design for high-order linear swarm systems: partial stability method, Automatica 48 (2012) 2335–2343, https://doi.org/10.1016/j.automatica.2012.06.041.
[166] Wei Ren, R.W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, IEEE Trans. Automat. Control 50 (2005) 655–661, https://doi.org/10.1109/TAC.2005.846556.
[167] W. Lin, L. Zhixin, G. Lei, Robust consensus of multi-agent systems with noise, in: Proc. Chin. Control Conf. (CCC), IEEE, Harbin, China, 2006, pp. 737–741, https://doi.org/10.1109/CHICC.2006.4347503.
[168] D.V. Dimarogonas, K.H. Johansson, Event-triggered control for multi-agent systems, in: Proc. IEEE Conf. Decision and Control (CDC), IEEE, Shanghai, China, 2009, pp. 7131–7136, https://doi.org/10.1109/CDC.2009.5399776.
[169] M. Cao, A.S. Morse, B.D.O. Anderson, Reaching a consensus in a dynamically changing environment: a graphical approach, SIAM J. Control Optim. 47 (2008) 575–600, https://doi.org/10.1137/060657005.
[170] J. Qin, C. Yu, Cluster consensus control of generic linear multi-agent systems under directed topology with acyclic partition, Automatica 49 (2013) 2898–2905, https://doi.org/10.1016/j.automatica.2013.06.017.
[171] Z. Li, W. Ren, X. Liu, M. Fu, Consensus of multi-agent systems with general linear and lipschitz nonlinear dynamics using distributed adaptive protocols, IEEE Trans. Automat. Control 58 (2013) 1786–1791, https://doi.org/10.1109/TAC.2012.2235715.
[172] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L.M. Gambardella, M. Dorigo, ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems, Swarm Intell. 6 (2012) 271–295, https://doi.org/10.1007/s11721-012-0072-5.
[173] L. Pitonakova, M. Giuliani, A. Pipe, A. Winfield, feature and performance comparison of the V-REP, gazebo and ARGoS robot simulators, in: M. Giuliani, T. Assaf, M.E. Giannaccini (Eds.), Towards Autonomous Robotic Systems, Springer, Cham, 2018, pp. 357–368.
[174] A.J. Ijspeert, A. Martinoli, A. Billard, L.M. Gambardella, Collaboration through the exploitation of local interactions in autonomous collective robotics: the stick pulling experiment, Auton. Robots 11 (2001) 149–171, https://doi.org/10.1023/A:1011227210047.
[175] A. Chella, S. Gaglio, M. Mannone, G. Pilato, V. Seidita, F. Vella, S. Zammuto, Quantum planning for swarm robotics, Robot. Autonom. Syst. 161 (2023) 104362, https://doi.org/10.1016/j.robot.2023.104362.
[176] D.B. Worth, B.G. Woolley, D.D. Hodson, SwarmSim: a framework for modeling swarming unmanned aerial vehicles using hardware-in-the-loop, J. Def. Model. Simul. 18 (2021) 105–124, https://doi.org/10.1177/1548512917715154.
[177] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, G. Balan, MASON: a multiagent simulation environment, Simulation 81 (2005) 517–527, https://doi.org/10.1177/0037549705058073.
[178] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T.H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, L.M. Gambardella, Evolving self-organizing behaviors for a swarm-bot, Auton. Robots 17 (2004) 223–245, https://doi.org/10.1023/B:AURO.0000033973.24945.f3.
[179] S. Lim, S. Wang, B. Lennox, F. Arvin, BeeGround - An open-source simulation platform for large-scale swarm robotics applications, in: Proc. IEEE Int. Conf. Autom. Robot. Appl. (ICARA), IEEE, Prague, Czech Republic, 2021, pp. 75–79, https://doi.org/10.1109/ICARA51699.2021.9376494.
[180] M. Rubenstein, C. Ahler, R. Nagpal, Kilobot: a low cost scalable robot system for collective behaviors, in: Proc. IEEE Int. Conf. Robot. Autom. (ICRA), IEEE, St. Paul, MN, USA, 2012, pp. 3293–3298, https://doi.org/10.1109/ICRA.2012.6224638.
[181] J. Klingner, A. Kanakia, N. Farrow, D. Reishus, N. Correll, A stick-slip omnidirectional powertrain for low-cost swarm robotics: mechanism, calibration, and control, in: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), IEEE,

Chicago, IL, USA, 2014, pp. 846–851, https://doi.org/10.1109/IROS.2014.6942658.
[182] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, A. Martinoli, The e-puck, a robot designed for education in engineering, in: Proc. 9th Conf. Auton. Robot Syst. Competitions, Castelo Branco, Portugal, 2009, pp. 59–65.
[183] D.H. Stolfi, G. Danoy, Evolutionary swarm formation: from simulations to real world robots, Eng. Appl. Artif. Intell. 128 (2024) 107501, https://doi.org/10.1016/j.engappai.2023.107501.
[184] S. Wilson, R. Gameros, M. Sheely, M. Lin, K. Dover, R. Gevorkyan, M. Haberland, A. Bertozzi, S. Berman, Pheeno, A versatile swarm robotic research and education platform, IEEE robot, Autom. Lett. 1 (2016) 884–891, https://doi.org/10.1109/LRA.2016.2524987.
[185] F. Arvin, J. Espinosa, B. Bird, A. West, S. Watson, B. Lennox, Mona: an affordable open-source Mobile robot for education and research, J. Intell. Rob. Syst. 94 (2019) 761–775, https://doi.org/10.1007/s10846-018-0866-9.
[186] J.M. Soares, I. Navarro, A. Martinoli, The khepera IV Mobile robot: performance evaluation, sensory data and software toolbox, in: A. Bicchi, W. Burgard, D. Prattichizzo (Eds.), Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2016), Springer, Cham, 2016, pp. 767–781, https://doi.org/10.1007/978-3-319-27146-0_59.
[187] F. Arvin, J. Murray, C. Zhang, S. Yue, Colias: an autonomous micro robot for swarm robotic applications, Int. J. Adv. Rob. Syst. 11 (2014), https://doi.org/10.5772/58730.
[188] A. Bono, G. Fedele, G. Franze, A swarm-based distributed model predictive control scheme for autonomous vehicle formations in uncertain environments, IEEE Trans. Cybern. 52 (2022) 8876–8886, https://doi.org/10.1109/TCYB.2021.3070461.
[189] Y. Zhou, Y. Tao, X. Lei, X. Peng, Self-organized swarm robot for multi-target trapping based on self-regulated density interaction, Inf. Sci. 661 (2024) 120119, https://doi.org/10.1016/j.ins.2024.120119.
[190] X. Lei, Y. Xiang, M. Duan, X. Peng, Exploring the criticality hypothesis using programmable swarm robots with vicsek-like interactions, J. R. Soc., Interface 20 (2023), https://doi.org/10.1098/rsif.2023.0176.
[191] Y. Deng, B. Zhu, H. Duan, Bioinspired bearing–based target enclosing control for unmanned aerial vehicle swarm, IEEE/ASME Trans. Mechatronics (2024) 1–11, https://doi.org/10.1109/TMECH.2024.3457826.
[192] C. Calderón-Arce, J.C. Brenes-Torres, R. Solis-Ortega, Swarm robotics: simulators, platforms and applications review, Computation 10 (2022) 80, https://doi.org/10.3390/computation10060080.
[193] Y. Yang, X. Wu, J. Li, J. Zang, J. Lu, R. Zgeib, Configuration design method of mega constellation for low Earth orbit observation, Space: Sci. Technol. 4 (2024), https://doi.org/10.34133/space.0175.
[194] J. Kruger, A.W. Koenig, S. D'Amico, Starling formation-flying optical experiment (StarFOX): system design and preflight verification, J. Spacecraft Rockets 60 (2023) 1755–1777, https://doi.org/10.2514/1.A35598.
[195] A.W. Koenig, S. D'Amico, E.G. Lightsey, Formation flying orbit and control concept for virtual super optics reconfigurable swarm mission, J. Guid. Control Dynam. 46 (2023) 1657–1670, https://doi.org/10.2514/1.G007334.
[196] E.G. Asri, Z.H. Zhu, Shape reconstruction of unknown tumbling target using factor graph-based dynamic SLAM, in: AIAA Scitech 2025 Forum, AIAA, Reston, Virginia, USA, 2025, https://doi.org/10.2514/6.2025-0180.
[197] T. Mahendrakar, S. Holmberg, A. Ekblad, E. Conti, R.T. White, M. Wilde, I. Silver, Autonomous rendezvous with non-cooperative target objects with swarm chasers and observers, ArXiv Preprint (2023), https://doi.org/10.48550/arXiv.2301.09059.
[198] H. Tsukamoto, S.-J. Chung, Y.K. Nakka, B. Donitz, D. Mages, M. Ingham, Neural-rendezvous: provably robust guidance and control to encounter interstellar objects, J. Guid. Control Dynam. 47 (2024) 2525–2543, https://doi.org/10.2514/1.G007671.
[199] M.G. Finley, M. Martinez-Ledesma, W.R. Paterson, M.R. Argall, D.M. Miles, J. C. Dorelli, E. Zesta, Generalized time-series analysis for in situ spacecraft observations: anomaly detection and data prioritization using principal components analysis and unsupervised clustering, Earth Space Sci. 11 (2024), https://doi.org/10.1029/2024EA003753.
[200] H. Holt, R. Armellin, Reinforcement learning enhanced LQR and control lyapunov functions for spacecraft proximity operations, IEEE Trans. Robot. 41 (2025) 5117–5129, https://doi.org/10.1109/TRO.2025.3600160.