Nonlinear Analysis: Hybrid Systems 62 (2026) 101747

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

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

# Temporal logic guided coverage control and path planning for unmanned aerial–ground vehicle systems☆,☆☆

Shiheng Zhang[1], Yangrui Zhang[1], Shaowen Miao, Yiding Ji*

*Robotics and Autonomous Systems Thrust of Systems Hub, The Hong Kong University of Science and Technology, Guangzhou 511453, China*

ARTICLE INFO

*Keywords:*
Signal temporal logic
Formal methods
Multi-agent systems
Coverage control
Path planning
Real-time monitoring

ABSTRACT

This paper investigates the problem of formal methods guided cooperative coordination of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) in complex and dynamic environments. A two-layer hierarchical framework is developed, where UAVs perform coverage and monitoring tasks as the perception layer, while UGVs complete safe navigation to avoid obstacles at the execution level. We leverage Spatial Aggregation Signal Temporal Logic (SaSTL) to formally specify both spatial and temporal behaviors of UAVs and UGVs, which allows real-time monitoring of collective task execution through the computation of robustness measures of properly defined SaSTL formulas. To ensure collision-free planning, the global SaSTL task is decomposed: an Attractive Potential Field (APF) based approach is first introduced to incorporates the ‘‘Eventually" formula fragments into the path planning objectives; then time-varying Control Barrier Functions (CBFs) are synthesized using a quadratic programming (QP) solver, which enforce safety constraints defined by ‘‘Always" formula fragments. Theoretical analysis confirms that our framework achieves optimal coverage control and safe path planning via SaSTL task monitoring. Extensive numerical simulations further validate the coordination performance of the proposed method and compare it with a baseline without SaSTL monitoring.

## 1. Introduction

Multi-agent systems (MASs), particularly heterogeneous teams comprising quadrotor unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), have garnered significant attention for their potential in complex and large-scale missions [1–4]. By leveraging the capabilities of different agents, MASs achieve superior scalability, flexibility and reliability that are unattainable by individual agents. Typical applications include autonomous environment monitoring, infrastructure inspection, and search and rescue after disasters, to name a few [5]. Among these scenarios, a core task is cooperative coverage control, which aims for strategic distribution of sensor-equipped agents to surveil a target environment.

However, real-world deployment of MASs are often hindered by environmental uncertainties, such as irregular obstacle distributions and dynamic terrain conditions. These uncertainties present two critical challenges: (1) limited local sensing capabilities

☆ This article is part of a Special issue entitled: ‘TC1.3 Discrete-event and hybrid systems(IFAC WC 2026)’ published in Nonlinear Analysis: Hybrid Systems.

☆☆ This work is partially supported by National Natural Science Foundation of China under grants 62303389 and 62373289; Guangdong Province Scientific Research Platform and Project Scheme under grant 2024KTSCX039; Youth Talent Support Program of Guangdong Association for Science and Technology under grant SKXRC2025463 and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No. 2023B1212010007).

* Corresponding author.
*E-mail addresses:* szhang593@connect.hkust-gz.edu.cn (S. Zhang), yzhang112@connect.hkust-gz.edu.cn (Y. Zhang), smiao585@connect.hkust-gz.edu.cn (S. Miao), jiyiding@hkust-gz.edu.cn (Y. Ji).

[1] Contributed equally.

https://doi.org/10.1016/j.nahs.2026.101747
Received 31 October 2025; Received in revised form 1 March 2026; Accepted 14 May 2026

Available online 1 June 2026

that restrict an agent's perception of global target locations, and (2) the risk of mission failure due to collisions or entry into restricted zones [6]. To overcome these challenges, hierarchical task decomposition is often employed, where complex global objectives are decomposed into multi-level coordinated ones for different agents. This paradigm has been widely applied in cyber–physical systems [7] and autonomous driving [8]. A representative application arises in post-earthquake rescue operations [9], where UAVs with high maneuverability and wide sensing ranges, serve as an aerial perception layer to identify targets and obstacles, and UGVs are responsible for ground-level navigation and precise target interaction to transport supplies to the designated sites.

Ensuring safety and task completion in such sequential and reactive missions necessitates the use of formal methods. Temporal logics, such as linear temporal logic (LTL) and signal temporal logic (STL), provide expressive languages for defining multi-agent tasks, ranging from path planning to persistent coverage [10–14]. However, tasks in in existing works are restrictive since they are often specified as sequential visits to target locations by individual agents, see, e.g., [6,15]. Additionally, conventional approaches primarily focus on the temporal evolution of individual trajectories, consequently fail to capture the collective spatial distribution properties of MASs, such as agent density or regional coverage.

To bridge these gaps, this paper leverages Spatial Aggregation Signal Temporal Logic (SaSTL), an extension of STL that incorporates spatial aggregation and counting operators into standard STL syntax [11,16–19]. Originated for smart city modeling [17], SaSTL is uniquely suited for multi-agent coverage problems as it formally defines collective spatial configuration of MASs. For instance, a certain area is covered by a minimum number of robots and each robot converges to its optimal point. This spatial-centric perspective is particularly appropriate for heterogeneous MASs where agents are coordinated to satisfy global spatial–temporal constraints.

Despite the promise, standard path planning approaches usually directly embed temporal logic formulas into quadratic programming (QP) into quadratic programming (QP), leading to high computational costs and poor scalability [20]. Through integrating Control Barrier Functions (CBFs) with STL is feasible for real-time safety enforcement [21,22], existing methods are often overly conservative to enforce multiple potentially conflicting specifications. An alternative approach encodes STL robustness into the control objectives through smooth approximations and leverages gradient-based optimization for controller synthesis [23].

Motivated by these situations, we propose a novel hierarchical UAV-UGV coverage-planning framework that is supported by spatio-temporal monitoring with SaSTL formulas. We adopt a "sense-then-act" coordination strategy to leverage the complementary capabilities of heterogeneous agents: UAVs serve as a perception layer to explore the environment, perform coverage tasks, then provide information to guide the safe path planning of UGVs at the execution layer. The two layers are connected through a unidirectional and sequential information flow, rather than an online feedback or tightly coupled control loop. Unlike conventional methods that focus on trajectories of individual agents, our approach utilizes SaSTL to formalize spatial aggregation requirements such as the number of agents in a particular region. This allows for the real-time monitoring of both the spatial configuration and temporal evolution of the MAS, ensuring successful task completion. The main contributions of this paper are summarized as follows:

(1) Our hierarchical framework functionally decomposes aerial perception of UAVs and ground navigation of UGVs. The spatial distribution and obstacle information are monitored by the UAVs, which are then used as fixed environmental constraints for the subsequent optimization-based path planning of UGVs.
(2) This work formalizes multi-agent coverage tasks by SaSTL formulas, especially the collective spatial distribution properties of heterogeneous MASs. Specifically, we model spatial aggregation requirements and monitor them in a real-time manner using properly defined SaSTL formulas. This is usually unattainable for conventional STL planning and control methods without introducing extensive auxiliary variables, since they primarily investigate temporal properties of individual agents.
(3) We decompose SaSTL specifications into *Eventually* and *Always* fragments for collision-free planning of UGVs. The former is incorporated into the planning objective function using an attractive potential field (APF) defined from manipulation of the robostness measure; and the latter is encoded as time-varying CBF constraints of a QP problem.
(4) We provide a comprehensive complexity analysis to prove that our SaSTL monitoring algorithm scales logarithmically $O(\log n)$ with respect to the number of agents $n$, which is highly competitive for potential applications in large-scale fleets.

The remainder of the paper is organized as follows. Section 2 introduces the preliminaries of CBFs and SaSTL, as well as multi-agent modeling of UAVs and UGVs. Section 3 first formulates the hierarchical coverage control-path planning problem of UAVs and UGVs, then presents the integrated coverage control and safe planning approach that is baded on real-time monitoring under SaSTL formulas. Substantial simulation results are provided in Section 4 to illustrate the performace of our approach against baseline methods, followed by conclusions and a brief discussion of future extension directions in Section 5.

## 2. Preliminaries and system modeling

### 2.1. Control barrier functions

Consider a general nonlinear control-affine system:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$. The functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous. For a continuous control input $\boldsymbol{u}(t)$, the trajectory $\boldsymbol{x}(t)$ is defined as the solution of (1) over time $t$. A continuously differentiable function $b(\boldsymbol{x}, t) : \mathbb{R}^n \to \mathbb{R}$ is referred to as a barrier function. The objective of a control barrier function (CBF) is to ensure the forward invariance of the safe set

$$C(t) := \{\boldsymbol{x} \in \mathbb{R}^n \mid b(\boldsymbol{x}, t) \geq 0\} \tag{2}$$

by appropriately designing the control input $\boldsymbol{u}$.

**Definition 1** (*Forward Invariance*)**.** A set $C(t)$ is said to be forward invariant under a control input $\boldsymbol{u}$ if, for any initial condition $\boldsymbol{x}(t_0) \in C(t_0)$, there exists a continuous trajectory $\boldsymbol{x}(t)$ such that $\boldsymbol{x}(t) \in C(t)$ for all $t \in [t_0, t_1]$.

That is, once the system state starts within the safe set, it remains there for all future time. To enforce this property, we introduce the concept of control barrier functions.

**Definition 2** (*Control Barrier Functions*)**.** A continuously differentiable function $b(\boldsymbol{x}, t)$ is control barrier function (CBF) for system (1) if there exists an extended class $\mathcal{K}_\infty$ function $\alpha$ such that, for all $t \in [t_0, t_1], \boldsymbol{x}(t) \in C(t)$, the following condition holds:

$$\sup_{\boldsymbol{u} \in \mathcal{U}} \left[ L_f b(\boldsymbol{x}, t) + L_g b(\boldsymbol{x}, t)\boldsymbol{u} + \frac{\partial b(\boldsymbol{x}, t)}{\partial t} + \alpha(b(\boldsymbol{x}, t)) \right] \geq 0,$$

where $\mathcal{U}$ is the feasible control input set, $\alpha : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ is strictly increasing with $\alpha(0) = 0$. Here, $L_f b(\boldsymbol{x}, t) = \frac{\partial b(\boldsymbol{x},t)}{\partial \boldsymbol{x}} f(\boldsymbol{x})$ and $L_g b(\boldsymbol{x}, t) = \frac{\partial b(\boldsymbol{x},t)}{\partial \boldsymbol{x}} g(\boldsymbol{x})$ denote the Lie derivatives of $b$ along $f$ and $g$, respectively.

**Lemma 1** (*[24]*)**.** *If $b(\boldsymbol{x}, t)$ is a time-varying CBF, then any Lipschitz continuous controller $\boldsymbol{u} \in K_{cbf}(\boldsymbol{x})$, defined as*

$$K_{cbf}(\boldsymbol{x}) := \{\boldsymbol{u} \in \mathcal{U} : L_f b + L_g b\boldsymbol{u} + \frac{\partial b}{\partial t} + \alpha b \geq 0\},$$

*ensures the forward invariance of $C(t)$, and guarantees the safety of the system (1).*

### 2.2. Spatial aggregation signal temporal logic

Spatial aggregation signal temporal logic (SaSTL) is a formal language that specifies temporal properties with spatial information. An SaSTL formula is consists of a predicate $\mu$, defined by a predicate function $h(y(t, z)) : \mathbb{R}^n \to \mathbb{R}$:

$$\mu := \begin{cases} \text{True} & \text{if } h(y(t, z)) \geq 0, \\ \text{False} & \text{if } h(y(t, z)) < 0, \end{cases} \tag{3}$$

where $z$ is the spatial location. SaSTL extends Signal Temporal Logic (STL) by introducing two spatial operators: $\mathcal{A}_D^{\text{op}}$ for spatial aggregation and $C_D^{\text{op}}$ for spatial counting across a set of agents. The syntax is given as

$$\varphi ::= \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 U_{[t_a,t_b]} \varphi_2 \mid \mathcal{A}_D^{\text{op}} y \sim c \mid C_D^{\text{op}} \varphi \sim c,$$

where $\varphi_1$ and $\varphi_2$ are SaSTL formulas, $\sim \in \{<, \leq, =, >, \geq\}$, and $c \in \mathbb{R}$ is a constant. The temporal operators $U_{[t_a,t_b]}$, $F_{[t_a,t_b]}\varphi$ and $G_{[t_a,t_b]}\varphi$ denote *Until*, *Eventually*, and *Always*, respectively, while $\neg$ and $\wedge$ represent Boolean *Negation* and *Conjunction*.

Spatial relations are described by an undirected weighted graph $\mathcal{G} = (Z, \mathcal{E}, w)$, where $Z$ is the set of locations, $\mathcal{E} \subseteq Z \times Z$ is the set of edges connecting nodes and $w : \mathcal{E} \to \mathbb{R}_{\geq 0}$ defines the distances between agents. The spatial domain is specified as

$$D := ([d_1, d_2], \tau),$$
$$\tau := \text{True} \mid \text{pos} \mid \neg\tau \mid \tau_1 \vee \tau_2,$$

where $[d_1, d_2]$ specifies a spatial interval with radii $d_1, d_2 \in \mathbb{R}$, pos is the proposition, and $\tau$ encodes the spatial property over the set of propositions to be satisfied at each location. For a given location $z \in Z$, the set

$$Z^z_{([d_1,d_2],\tau)} := \{z' \in Z \mid 0 \leq d_1 \leq w(z, z') \leq d_2 \wedge z' \vDash \tau\}$$

collects the neighboring locations within range $[d_1, d_2]$ that satisfy $\tau$. Then for signal $y$, the set of values at time $t$ and location $z$ over $Z^z_D$ is defined as

$$\varpi_D(y, t, z) = \bigcup_{z' \in Z^z_D} \{y(t, z')\}.$$

An operation set $\text{op}(\varpi_D(y, t, z))$, with $\text{op} \in \{\min, \max, \text{sum}, \text{avg}\}$, computes the corresponding aggregation value over $\varpi_D$. SaSTL defines spatial aggregation operators $\mathcal{A}_D^{\text{op}} y \sim c$ for $\text{op} \in \{\min, \max, \text{sum}, \text{avg}\}$ to evaluate the aggregated values $\text{op}(\varpi_D(y, t, z))$ across locations $z \in Z^z_D$. Similarly, spatial counting operators $C_D^{\text{op}} \varphi \sim c$ are defined to quantify the satisfaction of traces across locations. Specifically, $C_D^{\text{op}} \varphi = \text{op}(\{\vartheta((y, t, z') \vDash \varphi) \mid z' \in Z^z_D\})$, where $\vartheta((y, t, z) \vDash \varphi) = 1$ if $(y, t, z) \vDash \varphi$, and $\vartheta((y, t, z) \vDash \varphi) = 0$ otherwise. $(y, t, z) \vDash \varphi$ indicates that the signal $y$ satisfies formula $\varphi$ at time $t$ and location $z$. Then SaSTL semantics is defined recursively as:

$$(y, t, z) \vDash \mu \Leftrightarrow h(y(t, z)) \geq 0$$
$$(y, t, z) \vDash \varphi_1 \wedge \varphi_2 \Leftrightarrow (y, t, z) \vDash \varphi_1 \wedge (y, t, z) \vDash \varphi_2$$
$$(y, t, z) \vDash F_{[t_a,t_b]}\varphi \Leftrightarrow \exists t' \in [t + t_a, t + t_b], (y, t', z) \vDash \varphi$$
$$(y, t, z) \vDash G_{[t_a,t_b]}\varphi \Leftrightarrow \forall t' \in [t + t_a, t + t_b], (y, t', z) \vDash \varphi$$
$$(y, t, z) \vDash \mathcal{A}_D^{\text{op}} y \sim c \Leftrightarrow \text{op}(\varpi_D(y, t, z)) \sim c$$

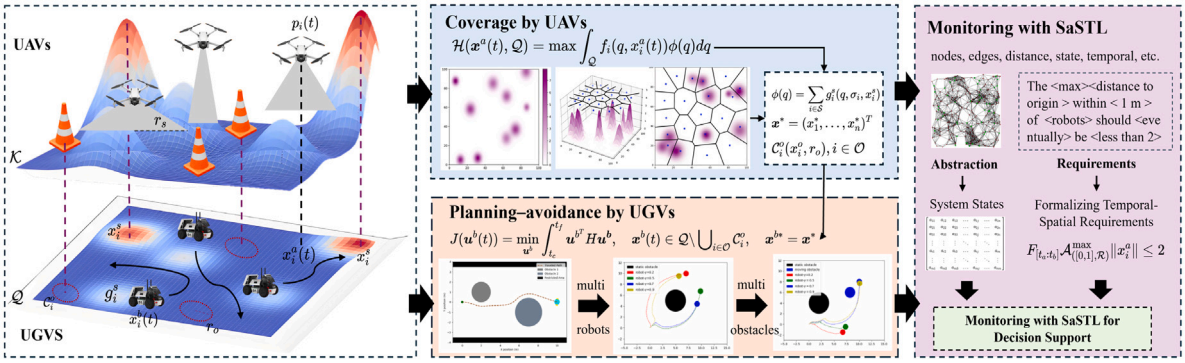



**Fig. 1.** Overview of the proposed two-layer framework for UAV coverage and UGV path planning under SaSTL monitoring. UAVs provide dynamic sensing coverage, while UGVs perform safe navigation and obstacle avoidance based on the generated spatial information.

$$(y,t,z) \vDash C_D^{\mathrm{op}}\varphi \sim c \Leftrightarrow \mathrm{op}(\vartheta((y,t,z') \vDash \varphi) \mid z' \in Z_D^z) \sim c$$

The robustness semantics $\rho$ quantifies the degree to which the signal $y$ satisfies an SaSTL formula [17]:

$$\rho(y,t,z,\mu) = h(y(t,z))$$
$$\rho(y,t,z,\varphi_1 \wedge \varphi_2) = \min\{\rho(y,t,z,\varphi_1), \rho(y,t,z,\varphi_2)\}$$
$$\rho(y,t,z,F_{[t_a,t_b]}\varphi) = \sup_{t' \in [t+t_a, t+t_b]} \rho(y,t',z,\varphi)$$
$$\rho(y,t,z,G_{[t_a,t_b]}\varphi) = \inf_{t' \in [t+t_a, t+t_b]} \rho(y,t',z,\varphi)$$
$$\rho(y,t,z,\mathcal{A}_D^{\mathrm{op}} y \sim c) = \begin{cases} \frac{\mathrm{sum}(\varpi_D) - c}{|\varpi_D|}, & \mathrm{op} = \mathrm{sum} \\ \mathrm{op}(\varpi_D) - c, & \mathrm{op} \setminus \mathrm{sum} \end{cases}$$
$$\rho(y,t,z,C_D^{\mathrm{op}}\varphi \sim c) = \begin{cases} \max_{z' \in Z_D^z}\{\rho(y,t,z',\varphi)\}, & \mathrm{op} = \max \\ \min_{z' \in Z_D^z}\{\rho(y,t,z',\varphi)\}, & \mathrm{op} = \min \\ \Gamma(\lceil c \rceil, \{\rho(y,t,z',\varphi) \mid z' \in Z_D^z\}), & \mathrm{op} = \mathrm{sum} \\ \Gamma(\lceil c \times |Z_D^z| \rceil, \{\rho(y,t,z',\varphi) \mid z' \in Z_D^z\}), & \mathrm{op} = \mathrm{avg}, \end{cases}$$

where $\Gamma(\delta, d)$ returns the $\delta$-th smallest number of set $d$. For op = sum, the operator specifies that at least $\lceil c \rceil$ locations satisfy $\varphi$. For op = avg, it indicates that at least $\lceil c \times |Z_D^z| \rceil$ locations satisfy $\varphi$, representing the average satisfaction ratio across the spatial domain. When no spatial variable $z$ is involved, SaSTL naturally reduces to STL.

### 2.3. Modeling of UAV–UGV systems

Consider a three-dimensional space $\mathcal{K} \subset \mathbb{R}^3$ containing $s$ signals and $m$ obstacles, denoted by the sets $\mathcal{S}$ and $\mathcal{O}$, respectively. Let $\mathcal{Q} \subset \mathbb{R}^2$ denote the ground projection of $\mathcal{K}$. The projection of an obstacle $i \in \mathcal{O}$ onto $\mathcal{Q}$ is modeled as a circular region $C_i^o(x_i^o, r_o)$ with center $x_i^o \in \mathcal{Q}$ and radius $r_o \in \mathbb{R}_{>0}$. The environmental information is characterized by a spatial density function:

$$\phi(q) = \sum_{i \in \mathcal{S}} g_i^s(q, \sigma_i, x_i^s), \tag{4}$$

where $q \in \mathcal{Q}$, $x_i^s$ is the center of the signal, and $g_i^s(q, \sigma_i, x_i^s)$ is a Gaussian function defined as $\frac{1}{2\pi\sigma_i^2} \exp(-\frac{1}{2\sigma_i^2}\|q - x_i^s\|^2)$.

Consider a UAV–UGV system as a multi–agent system consisting of $n$ UAVs and UGVs, denoted by $\mathcal{V}$ and $\mathcal{R}$, respectively. Each UAV moves in $\mathcal{K}$, while each UGV operates on $\mathcal{Q}$. The UAV positions are denoted by $\boldsymbol{p}(t) = (p_1(t), \ldots, p_n(t))^\top \in \mathbb{R}^{3\times n}$, and the UGV positions by $\boldsymbol{x}^b(t) = (x_1^b(t), \ldots, x_n^b(t))^\top \in \mathbb{R}^{2\times n}$. Throughout this work, all UAVs are assumed to operate at the same altitude, and obstacle avoidance is not considered in the coverage-control layer. The ground projections of the UAV positions are represented as $\boldsymbol{x}^a(t) = (x_1^a(t), \ldots, x_n^a(t))^\top \in \mathbb{R}^{2\times n}$. For simplicity, the dynamics of UAVs and UGVs are modeled by first-order integrators:

$$\dot{\boldsymbol{x}}^a(t) = \boldsymbol{u}^a(t), \quad \dot{\boldsymbol{x}}^b(t) = \boldsymbol{u}^b(t), \tag{5}$$

where $\boldsymbol{u}^a(t)$ and $\boldsymbol{u}^b(t)$ are the control inputs constrained by minimum and maximum linear velocities, denoted by $\boldsymbol{u}_{\min} \le \boldsymbol{u}^a(t), \boldsymbol{u}^b(t) \le \boldsymbol{u}_{\max}$. Each UAV $i \in \mathcal{V}$ senses the environment within a circular region $C_i^a(x_i^a(t), r_s)$ of radius $r_s \in \mathbb{R}_{>0}$, where the sensing capacity

decreases with distance:

$$f_i(q, x_i^a(t)) = \begin{cases} \beta e^{-\lambda \|q - x_i^a(t)\|^2} - \gamma, & \|q - x_i^a(t)\| \le r_s, \\ 0, & \|q - x_i^a(t)\| > r_s, \end{cases}$$

where $\gamma = \beta e^{-\lambda r_s^2}$ ensures boundary continuity, $\beta \in (0, 1)$ denotes maximum sensing range, and $\lambda > 0$ is the attenuation coefficient. Two UAVs can communicate if $\|x_i^a(t) - x_j^a(t)\| \le r_c$, where $r_c \in \mathbb{R}_{>0}$.

**Remark 1.** The first-order integrator model (5) is a kinematic abstraction of the system, which primarily provides a clear and concise exposition of the control and monitoring process in the following sections. By focusing on the kinematic model, we highlight the core theoretical contributions regarding spatial aggregation and temporal logic encoding without the additional algebraic complexity associated with high-order dynamics. It is important to emphasize that this modeling choice does not restrict the fundamental nature of the coverage and planning problem studied in this work. This abstraction is widely used in robotic control architectures [11,25,26], where high-level planners generate motion or velocity references, and low-level controllers (e.g., PID, MPC, or geometric controllers) are responsible for tracking these references while handling the underlying vehicle dynamics [27–29]. When the tracking is sufficiently fast, the resulting closed-loop system can be well approximated by a first-order integrator.

**Remark 2.** For the sake of investigation on multi-agent coverage control and path planning of UAVs and UGVs, the SaSTL spatial graph $\mathcal{G} = (Z, \mathcal{E}, w)$ is defined where the set of nodes $Z$ represent the positions of the agents. The SaSTL specifications are defined over global spatial domains, including $D_1 = ([0, \infty], \mathcal{V})$ and $D_2 = ([0, r_s], S)$, where the spatial index sets represent the set of all agents $\mathcal{V}$ and the set of spatial signals $S$, respectively. In these global domains, spatial aggregation and counting operators are evaluated across entire agents and signals, i.e., $Z^z_{D_1} = \mathcal{V}$ and $Z^z_{D_2} = S$ for all locations $z$ and time steps $t$. This formulation facilitates the evaluation of the collective distribution and coverage performance of the complete UAV-UGV system relative to the entire mission area during monitoring, regardless of instantaneous local connectivity. Since specifications are defined over the global domains, the resulting $\mathcal{G}$ effectively serves as a complete graph, ensuring persistently well defined spatial relations throughout missions.

## 3. Hierarchical coverage control and path planning via SaSTL monitoring

### 3.1. Problem formulation

Consider a multi-agent system consisting of $n$ UAVs operating in a three-dimensional space $\mathcal{K}$ and $n$ UGVs navigating on a two-dimensional plane $\mathcal{Q}$, with complementary tasks. UAVs are responsible for coverage and monitoring, and UGVs undertake path planning and obstacle avoidance. This is a two-layer setting illustrated in Fig. 1. In the aerial layer, UAVs conduct coverage and transmit target and obstacle information to UGVs. In the ground layer, UGVs perform path planning by solving a QP, where obstacle avoidance is enforced via a SaSTL-based CBF approach. Both layers are formally monitored using SaSTL specifications, allowing real-time detection of requirement violations and adaptive responses, such as path modification or action rejection.

1. **Coverage–monitoring layer:**
   (a) *Coverage:* Each UAV $i \in \mathcal{V}$ moves to a designated target position $\boldsymbol{p}^* = (p_1^*, \dots, p_n^*)^\top \in \mathbb{R}^{3\times n}$ before time $t_c$. The corresponding ground projection $\boldsymbol{x}^* = (x_1^*, \dots, x_n^*)^\top \in \mathbb{R}^{2\times n}$ is determined such that the sensing impact over $\mathcal{Q}$ is maximized. During this process, UAVs detect both signal distribution $g_i^s$ and obstacle information $C_i^o$.
   (b) *Monitoring:* UAVs employ SaSTL semantics to monitor both their coverage performance and the UGVs' execution of path planning and obstacle avoidance. The mission is regarded as unsuccessful if any UAV fails to satisfy the coverage specification, or if any UGV collides with an obstacle or fails to reach its assigned target.
2. **Planning–avoidance layer:** The UGVs must reach their target positions $\boldsymbol{x}^*$ before $t_f$, to eliminate signal impacts, and avoid obstacles $\bigcup_{i\in\mathcal{O}} C_i^o$. This is formulated using timed reach-avoid formulas, which are a subset of SaSTL. Only safety and reachability behaviors are explicitly considered in this work, whereas the specific process of signal elimination is omitted.

Our objective is to design control laws to optimize the following cost functions under safety and input constraints:

$$\mathcal{H}(\boldsymbol{x}^{\boldsymbol{a}}(t), \mathcal{Q}) = \max \int_{\mathcal{Q}} f_i(q, x_i^a(t))\phi(q)dq, \tag{6a}$$

$$J(\boldsymbol{u}^b(t)) = \min_{\boldsymbol{u}^b} \int_{t_c}^{t_f} \boldsymbol{u}^{b^T} P \boldsymbol{u}^{\boldsymbol{b}}, \tag{6b}$$

$$\text{s.t.} \quad \boldsymbol{p}(t) \in \mathcal{K}, \boldsymbol{x}^a(t) \in \mathcal{Q}, \boldsymbol{x}^b(t) \in \mathcal{Q} \setminus \bigcup_{i\in\mathcal{O}} C_i^o \tag{6c}$$

$$\boldsymbol{u}^a(t), \boldsymbol{u}^b(t) \in \mathcal{U}, \Delta\boldsymbol{u}^a(t), \Delta\boldsymbol{u}^b(t) \in \Delta\mathcal{U} \tag{6d}$$

where $P$ is a positive definite matrix. Eq. (6a) represents the coverage objective, which maximizes the sensing impact on $\mathcal{Q}$. Eq. (6b) specifies the path planning objective for UGVs with quadratic costs. Constraint (6c) enforces feasible states within the operational environment and ensures obstacle avoidance, meanwhile constraint (6d) describes the admissible inputs and their variations.

### 3.2. Coverage control

The coverage objective in (6a) aims to maximize the sensing impact of UAVs over the target region $\mathcal{Q}$. A widely adopted approach employs Voronoi tessellation [30] to partition $\mathcal{Q}$ into disjoint regions assigned to UAVs. Specifically, each UAV computes the centroid of its Voronoi cell based on its position, neighboring positions, and sensing radius. UAVs are steered toward their centroids and converge to a Voronoi-based equilibrium configuration. The Voronoi tessellation associated with UAV $i$ at time $t$ is given by:

$$V_i(x_i^a(t)) = \{q \in \mathcal{Q} \mid \|q - x_i^a(t)\| \leq \left\|q - x_j^a(t)\right\|, \forall j \neq i\}.$$

Accordingly, (6a) can be reformulated as

$$\mathcal{H}(\boldsymbol{x}^a(t), \boldsymbol{V}(t)) = \sum_{i\in\mathcal{V}} \int_{V_i(x_i^a(t))} f_i(q, x_i^a(t))\phi(q)dq, \tag{7}$$

where $\boldsymbol{V}(t) = (V_1(x_1^a(t)), \dots, V_n(x_n^a(t)))^\top$ denotes the Voronoi tessellation of the UAVs. Following a gradient-based optimization method, the derivative of $\mathcal{H}$ with respect to $x_i^a(t)$ can be expressed as

$$\begin{aligned}\frac{\partial \mathcal{H}(\boldsymbol{x}^a(t), \boldsymbol{V}(t))}{\partial x_i^a(t)} &= \int_{V_i(t)} \frac{\partial f_i(q, x_i^a(t))}{\partial x_i^a(t)} \phi(q,t)dq \\ &= M_{V_i}(x_i^a(t) - c_{V_i}(t)),\end{aligned}$$

where $M_{V_i}(t)$ and $c_{V_i}(t)$ denote the mass and centroid of the Voronoi cell $V_i(t)$, respectively:

$$M_{V_i}(t) = \int_{V_i(t)} 2\beta\lambda e^{-\lambda\left\|q - x_i^a(t)\right\|^2} \phi(q)\, dq, \tag{8a}$$

$$c_{V_i}(t) = \frac{\int_{V_i(t)} 2\beta\lambda e^{-\lambda\left\|q - x_i^a(t)\right\|^2} \phi(q) q\, dq}{M_{V_i}(t)}. \tag{8b}$$

Thus, the control law for each UAV $i$ is designed as

$$u_i^a(t) = -\kappa_a(x_i^a(t) - c_{V_i}(t)), \tag{9}$$

where $\kappa_a > 0$ is a control gain.

**Theorem 1.** *Consider a swarm $\mathcal{V}$ of $n$ UAVs governed by the control law (9). Then, each UAV $i \in \mathcal{V}$ asymptotically converges to the centroid $c_{V_i} = x_i^*$ of its Voronoi cell $V_i$.*

**Proof.** Define the Lyapunov candidate function as

$$V(t) = \frac{1}{2}\sum_{i\in\mathcal{V}} \|x_i^a(t) - c_{V_i}(t)\|^2.$$

Taking its time derivative along system trajectories yields

$$\dot{V}(t) = \sum_{i\in\mathcal{V}} (x_i^a(t) - c_{V_i}(t))^T \dot{x}_i^a(t).$$

Substituting $\dot{x}_i^a(t) = u_i^a(t) = -\kappa_a(x_i^a(t) - c_{V_i}(t))$, we get

$$\dot{V}(t) = -\kappa_a \sum_{i\in\mathcal{V}} \|x_i^a(t) - c_{V_i}(t)\|^2 \leq 0, \tag{10}$$

therefore, $V(t)$ is non-increasing. By LaSalle's invariance principle [31], the trajectories converge to the largest invariant set satisfying $\dot{V}(t) = 0$, i.e., $x_i^a(t) = c_{V_i}(t)$ for all $i \in \mathcal{V}$. Hence, each UAV asymptotically converges to the centroid of its Voronoi cell. ■

### 3.3. Monitoring with SaSTL

To formally specify the monitoring tasks, we employ SaSTL formulas defined as follows:

$$\psi ::= \text{True} \mid \mu \mid \neg\mu \mid \psi_1 \wedge \psi_2, \tag{11a}$$

$$\begin{aligned}\varphi ::= {} & F_{[a,b]}\psi \mid G_{[a,b]}\psi \mid \psi_1 U_{[a,b]}\psi_2 \mid \\ & \mathcal{A}_D^{\text{op}} y \sim c \mid C_D^{\text{op}}\varphi \sim c \mid \varphi_1 \wedge \varphi_2,\end{aligned} \tag{11b}$$

where $\psi$ denotes the non-temporal formula, and $\varphi$ stands for the temporal formula. The objective in (6) can be encoded as a monitoring SaSTL specification that consists of four formulas:

$$\varphi_1 := F_{[t_0:t_c]}\mathcal{A}^{\min}_{([0,r_s],S)}\|x_i^s - x_j^a\| \leq 3\sigma_i, \tag{12a}$$

$$\varphi_2 := C^{\min}_{([0,\infty),\mathcal{V})}\left(F_{[t_0:t_c]}(\|x_i^a - c_{V_i}\| \le \varepsilon)\right) = 1, \tag{12b}$$

$$\varphi_3 := G_{[t_c:t_f]}(\|\boldsymbol{x}^b - \boldsymbol{x}^o\| \ge r_o), \tag{12c}$$

$$\varphi_4 := F_{[t_c:t_f]}(\|\boldsymbol{x}^b - \boldsymbol{x}^*\| \le \varepsilon), \tag{12d}$$

where the range of $3\sigma_i$ is generally considered as the main distribution region of signal $i$, implying that 99.7% of the signal is concentrated within a radius of $3\sigma$. Eqs. (12a)–(12b) specify UAV-related coverage objectives:

(1) each signal is covered by at least one UAV;
(2) each UAV converges to the centroid of its Voronoi centroid.

In particular, $\varphi_1$ requires that, for every signal $i$, there exists at least one UAV $j$ within distance $3\sigma_i$ during the interval $[t_0, t_c]$. The formula $\varphi_2$ ensures that each UAV eventually converges to its Voronoi centroid $c_{V_i}$ and remains within an $\varepsilon$-neighborhood, where $\varepsilon > 0$ is the error bound. Alternative formulations of $\varphi_1$ share the same semantics and include two conditions below:

$$C^{\max}_{([0,r_s],S)}\left(F_{[t_0:t_c]}(\|x_i^s - x_j^a\| \le 3\sigma_i)\right) = 1,$$
$$C^{\text{sum}}_{([0,r_s],S)}\left(F_{[t_0:t_c]}(\|x_i^s - x_j^a\| \le 3\sigma_i)\right) \ge s,$$

where each signal $i$ is covered by at least one UAV within the distribution range $3\sigma_i$. Eqs. (12c)–(12d) define UGV tasks. Formula $\varphi_3$ enforces obstacle avoidance by requiring the UGV obstacle distance to remain greater than $r_o$ during $[t_c, t_f]$. Formula $\varphi_4$ guarantees that each UGV eventually reaches its designated target $\boldsymbol{x}^*$ within $[t_c, t_f]$ and subsequently remains within a $\varepsilon$-neighborhood.

The detailed monitoring process is presented in *Algorithm 1*, which evaluates requirements $\varphi_1$–$\varphi_4$. The inputs of the monitor include locations, environmental information, time intervals, and graph topology, and the outputs are numerical values indicating the satisfaction degree for each requirement. The algorithms in [17] are adopted to compute the operators $\mathcal{A}^{\text{op}}_{\mathcal{D}}$ and $C^{\text{op}}_{\mathcal{D}}$. Additionally, the distributed parallel search algorithm `deScan()` [32] is adopted to identify locations satisfying $\mathcal{D}$. In essence, $\mathcal{A}^{\text{op}}_{\mathcal{D}}$ computes the aggregated signal values over the spatial domain, whereas $C^{\text{op}}_{\mathcal{D}}$ computes the corresponding cumulative values in the same region.

The computational complexity of Algorithm 1 is determined by the structure of SaSTL formulas and the efficiency of spatial queries. Inspired by formal verification methods [33,34], the complexity of monitoring logical and temporal operators in SaSTL is comparable with that of STL. Our monitoring approach decouples the logical evaluation from the system dimension, ensuring linear complexity growth with the formula size rather than the state space size, similar to the lightweight observer designs in [35,36].

To achieve the decomposition, a spatial indexing structure called Range Tree [37] is leveraged to organize agent positions, which allows logarithmic time retrieval of agents within a spatial domain $\mathcal{D}$. Formally, let $|\varphi|$ denote the number of operators in the SaSTL syntax tree, $T_{\max}$ the maximum temporal horizon, $n$ the total number of agents, and $|Z|_{\max}$ the maximum number of agents satisfying a spatial predicate. Given existing SaSTL monitoring results [17], the overall time complexity is derived and bounded by:

$$O\big(|\varphi| \times T_{\max} \times (\log(n) + |Z|_{\max})\big).$$

This logarithmic dependence on $n$, i.e., $O(\log n)$, confirms that the monitoring overhead remains manageable even as the fleet size increases, validating the scalability of the proposed SaSTL framework.

### 3.4. Path planning with obstacle avoidance

This subsection integrates SaSTL semantics into the path planning problem with obstacle avoidance. Motivated by the method in [38], the *Eventually* specification $\varphi_4$ is incorporated into the objective function (6b) through an APF, and the *Always* specification $\varphi_3$ is enforced as a hard constraint by a CBF.

Consider an *Eventually* formula $\varphi_F = F_{[t_a,t_b]}\psi$, where the robustness measure of the atomic formula $\psi$ is defined as $\rho(\boldsymbol{x}^b, t, \psi) = h(\boldsymbol{x}^b(t))$, and the cumulative robustness of $\varphi_F$ is given by $\rho(\boldsymbol{x}^b, t, \varphi_F) = \sum_{t' \in [t_a,t_b]} \rho(\boldsymbol{x}^b, t + t', \psi)$. For real-time feedback controller design, $\rho(\boldsymbol{x}^b, t, \psi)$ is employed as the optimization objective, enabling the optimal control satisfying $\varphi_F$ to be obtained by gradient descent [23]. Task satisfaction is enforced through the APF formulation as follows:

$$A_{\varphi_F}(\boldsymbol{x}^b, t) = \frac{1}{2}\kappa_f \parallel \rho(\boldsymbol{x}^b, t, \text{True}) - \rho(\boldsymbol{x}^b, t, \psi) \parallel^2, \tag{13}$$

where $\kappa_f > 0$ is a tuning parameter. The gradient with respect to $\rho(\boldsymbol{x}^b, t, \psi)$ is

$$\nabla A_{\varphi_F} = \frac{\partial A_{\varphi_F}(\boldsymbol{x}^b, t)}{\partial \rho(\boldsymbol{x}^b, t, \psi)} = \kappa_f \|\rho(\boldsymbol{x}^b, t, \text{True}) - \rho(\boldsymbol{x}^b, t, \psi)\|.$$

Since $\nabla A_{\varphi_F}(\boldsymbol{x}^b, t)$ decreases along trajectories satisfying $\mu = \text{True}$, it can be incorporated in the minimization objective of the path planning problem. Moreover, the *Eventually* specification $\varphi_4$ can be decomposed into $n$ atomic formulas as

$$\begin{aligned}\varphi_4 &:= F_{[t_c:t_f]}(\|\boldsymbol{x}^b - \boldsymbol{x}^*\| \le \varepsilon) \\ &:= \bigcap_{i \in \mathcal{R}} F_{[t_c:t_f]}(\|x_i^b - x_i^*\| \le \varepsilon) = \bigcap_{i \in \mathcal{R}} \varphi_4^i.\end{aligned} \tag{14}$$

**Algorithm 1** SaSTL Monitoring Algorithm

**Input:** Signals $g_i^s(q, \sigma_i, x_i^s)$, time interval $[t_0, t_f]$, locations $(x_i^a, x_i^b, x^o, x^*)$, parameters $r_s, r_o, \varepsilon$, graph $\mathcal{G}$
**Output:** Satisfaction values $\rho_{\varphi_1}, \rho_{\varphi_2}, \rho_{\varphi_3}, \rho_{\varphi_4}$

**begin**

// Task $\varphi_1$: Signal covered by UAV
$\rho_{\varphi_1} \leftarrow -\infty$;
**for** $t \in [t_0 : t_c]$ **do**
  $\varrho := 0$; $v := +\infty$;
  $Z^{x_i^s}_{[0,r_s],\mathcal{S}} := \texttt{deScan}(x_i^s, \mathcal{G}, ([0, r_s], \mathcal{S}))$;
  **for** $x_j^a \in Z^{x_i^s}_{[0,r_s],\mathcal{S}}$ **do**
    $v := \min(v, \|x_i^s(t) - x_j^a(t)\|)$;
    $\varrho := \varrho + 1$;
  **if** $\varrho == 0$ **then return** $\rho_t \leftarrow -\infty$;
  **else return** $\rho_t \leftarrow r_s - v$;
  $\rho_{\varphi_1} \leftarrow \max(\rho_{\varphi_1}, \rho_t)$;
// Task $\varphi_2$: UAV convergence to centroid $c_{V_i}$
$\rho_{\varphi_2} \leftarrow -\infty$;
**for** $t \in [t_0 : t_c]$ **do**
  $\varrho := 0$; $E := \text{EmptyList}()$;
  $Z^{x_i^a}_{[0,\infty],\mathcal{V}} := \texttt{deScan}(x_i^a, \mathcal{G}, ([0, \infty], \mathcal{V}))$;
  **for** $c_{V_i} \in Z^{x_i^a}_{[0,\infty],\mathcal{V}}$ **do**
    $\rho_t \leftarrow \varepsilon - \|x_i^a(t) - c_{V_i}(t)\|$;
    E.add($\rho_t$);
    $\varrho \leftarrow \varrho + 1$;
  **if** $\varrho == 0$ **then return** $\rho_t \leftarrow -\infty$;
  **else return** $\rho_t \leftarrow E.\min()$;
  $\rho_{\varphi_2} \leftarrow \max(\rho_{\varphi_2}, \rho_t)$;
// Task $\varphi_3$: UGV obstacle avoidance
$\rho_{\varphi_3} \leftarrow +\infty$;
**for** $t \in [t_c : t_f]$ **do**
  $\rho_t \leftarrow \|x^b(t) - x^o(t)\| - r_o$;
  $\rho_{\varphi_3} \leftarrow \min(\rho_{\varphi_3}, \rho_t)$;
// Task $\varphi_4$: UGV reaching the target
$\rho_{\varphi_4} \leftarrow -\infty$;
**for** $t \in [t_c : t_f]$ **do**
  $\rho_t \leftarrow \varepsilon - \|x^b(t) - x^*(t)\|$;
  $\rho_{\varphi_4} \leftarrow \max(\rho_{\varphi_4}, \rho_t)$;
**return** $(\rho_{\varphi_1}, \rho_{\varphi_2}, \rho_{\varphi_3}, \rho_{\varphi_4})$;

Accordingly, the objective function (6b) is reformulated as

$$J^*(\boldsymbol{u}^b, \boldsymbol{x}^b) = \min \boldsymbol{u}^{b^T} \boldsymbol{P} \boldsymbol{u}^b + \sum_{i \in \mathcal{R}} \nabla A_{\varphi_4^i}(x_i^b, t). \tag{15}$$

Although classical APF based methods may suffer from local minima, we alleviate this issue in the proposed approach by the structural design of the SaSTL-based synthesis. Specifically, the *Always* and *Eventually* specifications are first decoupled, with safety enforced as hard constraints and task completion encoded as an objective term. Obstacle avoidance is then enforced via CBF constraints rather than repulsive potential fields, preemptively preventing undesired scenarios where attractive and repulsive forces cancel out. Moreover, the APF in (13) is constructed based on atomic *Eventually* formulas whose robustness measure $\rho(\boldsymbol{x}^b, t, \psi)$ is norm-based, resulting in a convex quadratic potential with respect to the robustness metric. As a result, the optimization problem does not end with local minima from the APF component; instead, the gradient descent direction $\nabla A_{\varphi_4^i}$ in (15) consistently guides the system toward the global minimum. Finally, when multiple *Eventually* formulas are involved, we decompose time-ordered task specifications $\varphi_4^i$ to exclude conflicting STL requirements.

For obstacle avoidance, the *Always* specification $\varphi_3$ is encoded as hard constraints using a CBF. Since multiple obstacles may exist, $\varphi_3$ is expressed as a conjunction of atomic formulas:

$$\varphi_3 = \bigwedge_{l=1}^{k} \varphi_G^l, \quad \varphi_G^l := G_{[t_a,t_b]} \psi^l. \tag{16}$$

For each $\varphi_G^l$, a candidate CBF is defined as $b_l(\boldsymbol{x}^b, t) = h_l(\boldsymbol{x}^b(t))$ through the robustness of $\psi^l$, such that

$$b_l(\boldsymbol{x}^b, t) \geq \rho(\boldsymbol{x}^b, t, \psi^l), \forall t' \in [t + t_a, t + t_b]. \tag{17}$$

If $b_l(\boldsymbol{x}^b, t) \geq 0$ holds for all $t \geq 0$, then $(\boldsymbol{x}^b, t) \vDash \varphi_G^l$ is satisfied. To facilitate the conjunction of *Always* operators, the minimum of $h_l(\boldsymbol{x}^b(t))$ is smoothly approximated following the manipulation in [21]:

$$\hat{h}(\boldsymbol{x}^b(t)) := -\frac{1}{\eta} \ln(\sum_{l=1}^{k} \exp(-\eta h_l(\boldsymbol{x}^b(t)))), \eta > 0, \tag{18}$$

which provides a smooth under-approximation of the minimum function, i.e., $\hat{h}(\boldsymbol{x}^b(t)) \leq \min h_l(\boldsymbol{x}^b(t))$. According to this, a composite CBF is constructed from individual CBFs below:

$$b(\boldsymbol{x}^b, t) = -\frac{1}{\eta} \ln(\sum_{l=1}^{k} \exp(-\eta b_l(\boldsymbol{x}^b, t))). \tag{19}$$

The CBF constraint in *Definition* 2 is then expressed as

$$\frac{\partial b(\boldsymbol{x}^b, t)}{\partial \boldsymbol{x}^b}(f(\boldsymbol{x}^b) + g(\boldsymbol{x}^b)\boldsymbol{u}^b) + \frac{\partial b(\boldsymbol{x}^b, t)}{\partial t} \geq -\alpha(b(\boldsymbol{x}^b, t)), \tag{20}$$

where $\alpha(\cdot)$ is a class-$\mathcal{K}$ function that ensures the forward invariance of the safe set.

Finally, the feasibility issue of the optimization problem may arise in the presence of potential conflicts between safety-critical and liveness-driven specifications. To resolve this problem, we introduce a slack variable $\delta$ into the CBF constraints. Combining the APF-based objective (15) and the CBF-based constraints (20), the path planning problem is reformulated as the following QP:

$$J(\boldsymbol{u}^b, \boldsymbol{x}^b, \delta) = \min_{\boldsymbol{u}^b, \delta} \boldsymbol{u}^{bT} \boldsymbol{P} \boldsymbol{u}^b + \sum_{i \in \mathcal{R}} \nabla A_{\varphi_4^i}(x_i^b, t) + \kappa \delta^2, \tag{21}$$

$$\text{s.t. } \frac{\partial b(\boldsymbol{x}^b, t)}{\partial \boldsymbol{x}^b}\left(f(\boldsymbol{x}^b) + g(\boldsymbol{x}^b)\boldsymbol{u}^b\right) + \frac{\partial b(\boldsymbol{x}^b, t)}{\partial t} \geq -\alpha(b(\boldsymbol{x}^b, t)) - \delta,\ \delta \geq 0. \tag{22}$$

where $\kappa > 0$ is a large penalty coefficient and $\delta \geq 0$ is the slack variable. The slack variable is penalized in the objective function with a sufficiently large weight, ensuring that safety violations occur only when strictly necessary and are minimized in magnitude and duration. This formulation establishes a clear priority structure between the *Always* specifications, enforced as (relaxed) safety constraints, and the *Eventually* specifications, encoded as liveness objectives. The introduction of $\delta$ ensures that the admissible control set is non-empty even under extreme environmental constraints, preventing optimization solver failures.

By allowing minimal and penalized relaxation of the CBF constraints, the proposed QP formulation improves feasibility even in highly constrained or transiently conflicting scenarios. In practice, when the APF-driven liveness objective directs the agent toward an obstacle (a conflict), the QP solver prioritizes the CBF constraint. The agent will follow the safe set's boundary, effectively bypassing the obstacle while maintaining the minimum possible deviation from the APF's intended path. This synthesis mitigates conservativeness and provides a robust compromise between task completion and collision avoidance.

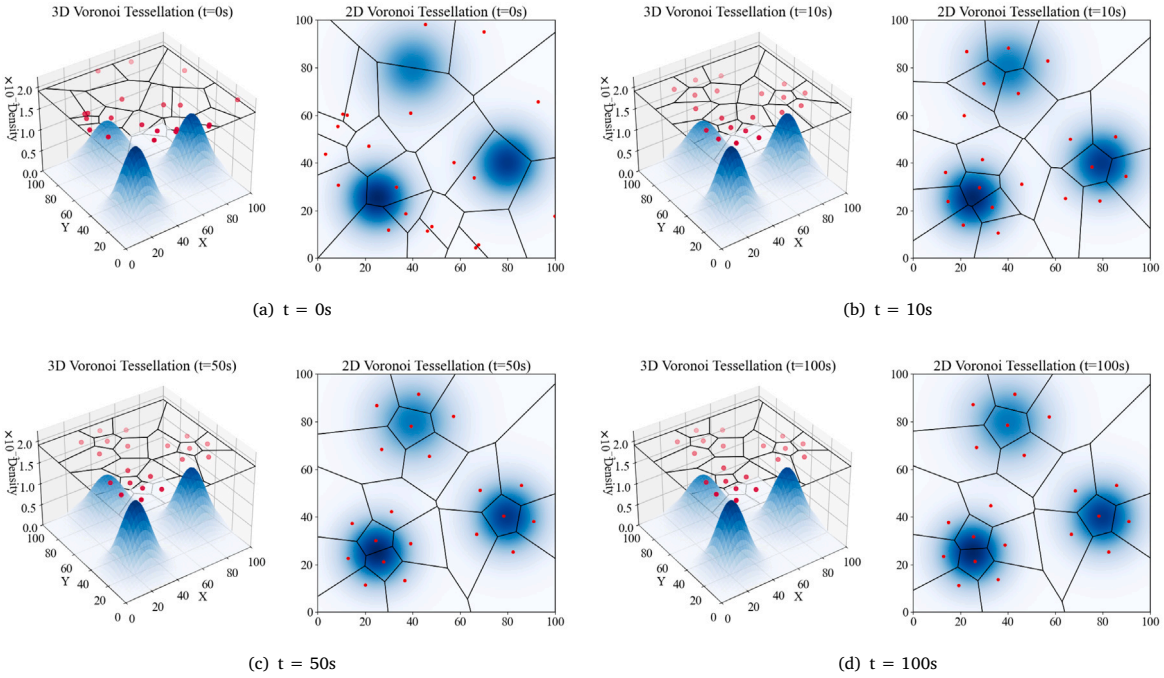
It is worth noting that although the paper adopts a first-order integrator model $f(x) = 0$ and $g(x) = I$ for clarity, our CBF-based formulation in (21)–(22) also works for general nonlinear control-affine systems of the form $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$. For agents with higher-order or nonlinear dynamics, such as double-integrator or nonholonomic models, our framework can be extended by employing standard techniques, including High-Order Control Barrier Functions (HOCBFs) [39,40] or tracking-based controllers [41,42], to map high-level planning commands to physical control inputs. Importantly, the SaSTL monitoring layer operates on the resulting state trajectories and therefore does not require fundamental modification when applied to complex dynamic systems.

**Remark 3.** In principle, it is also feasible to design a controller by incorporating the robustness degree of SaSTL into the objective function of coverage control. However, this turns out to be unnecessary in the present context. The coverage task involves only a single *Eventually* specification over one time interval, rather than multiple avoidance tasks across different intervals. As a result, embedding robustness into the objective would not yield any notable performance improvement, which justifies our design choice.

## 4. Numerical simulations

Consider a 3D space $\mathcal{K}$ and a 2D plane $\mathcal{Q}$, each with side length 100 m, where 20 UAVs and 20 UGVs are deployed, respectively. There are 3 environmental signals located at $x_1^s = [25, 25]$, $x_2^s = [80, 40]$, and $x_3^s = [40, 80]$, with corresponding variances of $\sigma_1^2 = 100$, $\sigma_2^2 = 110$, and $\sigma_3^2 = 150$, respectively. Each UAV has a communication radius of $r_c = 20$ m and a sensing radius of $r_s = 50$ m, with parameters $\beta = 1$ and $\lambda = 0.02$. The control gains are set as $\kappa_a = 0.6$, $\kappa_f = 2$, the penalty coefficient is 100, and the bounded error is $\varepsilon = 0.15$ m. The coverage task is required to be completed within $t_c = 100$ s, and the path planning task within $t_f = 200$ s.

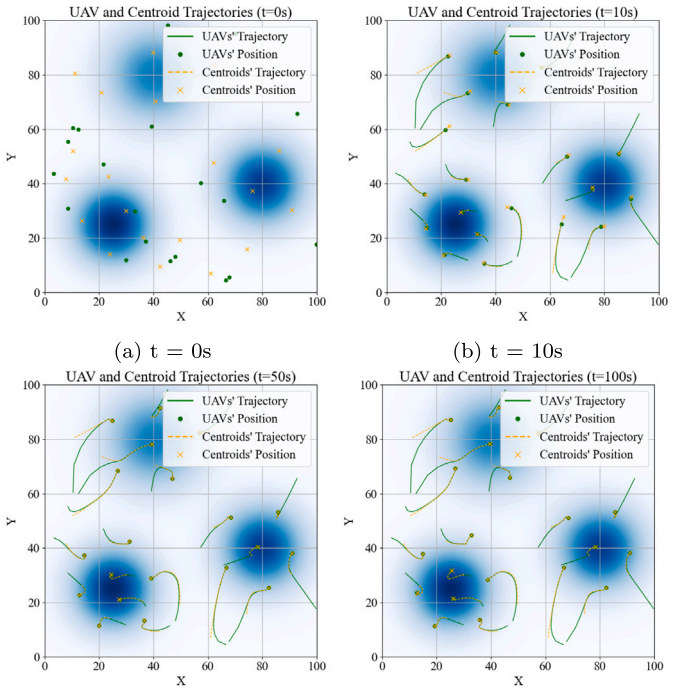
The UAVs first perform the coverage task, as illustrated in Figs. 2 and 3. At $t = 0$ s, the UAVs are randomly distributed in $\mathcal{K}$, and each UAV constructs its Voronoi tessellation according to the positions of its neighboring UAVs, as shown in Fig. 2(a). At this stage, not all UAVs cover a signal, and their positions do not coincide with the centroids of their respective Voronoi cells, as shown in Fig. 3(a). The green circles and yellow crosses represent the UAVs and the centroids of the Voronoi cells, respectively.




(a) t = 0s (b) t = 10s

(c) t = 50s (d) t = 100s

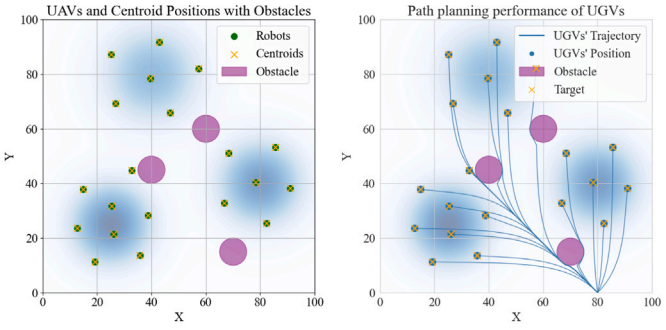
**Fig. 2.** UAV coverage evolution under the Voronoi-based control law at different time instants. The 3D and 2D Voronoi tessellations illustrate how UAVs gradually redistribute to improve spatial coverage over the mission area.




(a) t = 0s (b) t = 10s

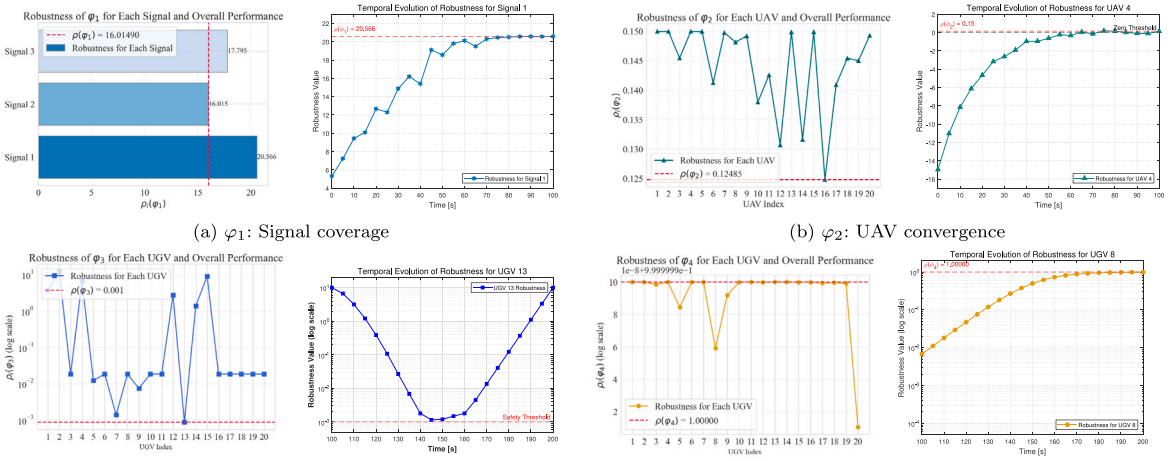
(c) t = 50s (d) t = 100s

**Fig. 3.** Temporal evolution of UAV and centroid trajectories at different time intervals from $t = 0$ s to $t = 100$ s. The green solid lines and dots represent the UAVs' trajectories and current positions, while the orange dashed lines and crosses denote the moving centroids, visualizing the dynamic tracking and convergence process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)




(a) Environmental information (b) UGV path planning results

**Fig. 4.** Environmental information and corresponding UGV path planning results.




(a) $\varphi_1$: Signal coverage (b) $\varphi_2$: UAV convergence

(c) $\varphi_3$: Obstacle avoidance (d) $\varphi_4$: Target reaching

**Fig. 5.** Evaluation of SaSTL robustness for UAV and UGV task monitoring across four specifications: (a) signal coverage $\varphi_1$, (b) UAV convergence $\varphi_2$, (c) obstacle avoidance $\varphi_3$, and (d) target reaching $\varphi_4$. In each subfigure, the left plot visulizes the overall performance across different agents/signals, while the right plot illustrates the corresponding temporal evolution.

**Table 1**
Robustness results under multiple simulations.

| Task semantics | Simulation number = 100 | | | Success rate |
|---|---|---|---|---|
| | sat | vio | $\rho_{ave}$ | |
| $\varphi_1$ | 100 | 0 | $1.576 \times 10^{1}$ | 100% |
| $\varphi_2$ | 94 | 6 | $8.923 \times 10^{-2}$ | 94% |
| $\varphi_3$ | 97 | 3 | $2.744 \times 10^{-2}$ | 97% |
| $\varphi_4$ | 100 | 0 | $9.999 \times 10^{-1}$ | 100% |

Once the UAVs start executing the coverage control algorithm , their positions gradually converge to the corresponding centroids of their respective Voronoi cells. As shown in Figs. 3(b) and 3(c), the green solid lines represent the UAV trajectories, and the yellow dotted lines denote the centroid trajectories. Meanwhile, the Voronoi tessellations evolve dynamically, as shown in Figs. 2(b) and 2(c). At $t = 50$ s, all UAVs successfully cover a signal, thus their positions lie within the $3\sigma$ range of the signal. When $t = 100$ s, the coverage state remains nearly unchanged compared to that at $t = 50$ s, indicating completion of the coverage task. This result demonstrates that the proposed control law maximizes the coverage performance and achieves the objective defined in (6a).

After completing the coverage task, the UAVs transmit the environmental information at $t = 100$ s to the UGVs, including the final position projections $\boldsymbol{x}^*$, the centroid coordinates $c_{V_i}$, and the obstacle information $C_i^o$, as shown in Fig. 4(a). Based on this information, the UGVs perform path planning with obstacle avoidance, as illustrated in Fig. 4(b). Three circular obstacles with radii of $r_o = 5$ m are considered, and the UGVs are initialized at the same starting point $[80, 0]$, representing a rescue center or a base station in practical applications. Each UGV then navigates toward its assigned target along the shortest feasible path and

**Table 2**
Performance comparison between the proposed framework and non-monitoring baseline approach.

| Metrics | Baseline (No monitor) | Proposed framework |
|---|---|---|
| Success Rate ($\varphi_1$–$\varphi_4$) | 75% | **100%** |
| Min. Safety Margin ($\varphi_3$) | $0.02 \pm 0.01$ | **0.28 ± 0.05** |
| Avg. Robustness Degree $\rho$ | 0.08 | **0.42** |
| Relaxation Variable $\delta$ (Avg.) | 1.25 | **0.14** |
| Computing Time per Step (ms) | **2.1** | 4.5 |

*Note:* The baseline employs the same QP-CBF controller but lacks the SaSTL-based proactive strategy adjustment.

avoids obstacles, which is guaranteed by solving the optimal control problem with CBF constraints. As shown in Fig. 4(b), the UGVs successfully reach their designated targets without any collisions or violations of safety constraints.

To evaluate the performance of the SaSTL monitoring guided coverage control-path planning scheme, we compute the robustness degrees of formulas $\varphi_1$–$\varphi_4$ in the simulation using Algorithm 1. Fig. 5 illustrates the temporal evolution of the robustness values for 20 UAV coverage tasks and 20 UGV planning tasks. Unlike binary satisfaction results, these quantitative values reveal the safety margins and execution quality of the agents throughout the simulation. Specifically, Fig. 5(a) shows the robustness of $\varphi_1$, which quantifies the coverage margin between UAV sensing regions and the assigned signals. Larger values indicate stronger coverage redundancy. Fig. 5(b) depicts the robustness of $\varphi_2$ for each UAV, reflecting the convergence margin toward the centroid of the corresponding Voronoi cell. The differences across indices indicate that through all agents satisfied the specifications, their convergence margins varied due to different initial spatial configurations and neighbor interactions.

More importantly, Fig. 5 illustrates the temporal evolution of the robustness degrees $\rho(y, t, z, \varphi)$ for the core specifications, which capture the dynamic constraints of the system. In Fig. 5(c), which depicts the obstacle avoidance specification ($\varphi_3$), the robustness degree for each UGV represents its minimum distance margin to obstacles over the entire trajectory $[t_c, t_f]$. A temporary dip in the robustness (e.g., for UGV 7 and UGV 13) corresponds to a UGV approaching an obstacle and nearing the safety boundary, while the subsequent increase reflects the execution of an avoidance maneuver. This illustrates the predictive capability of the SaSTL monitor, as decreasing robustness values indicate imminent constraint activation before an actual violation occurs. Similarly, in Fig. 5(d), a lower robustness value (UGV 8) indicates a case where the agent reached the target region $\boldsymbol{x}^*$ with a minimal spatial or temporal margin relative to the deadline $t_f$. Increasing robustness values indicate that the UGV is progressing toward the target region with a growing satisfaction margin. In general, the fact that all values remain positive confirms satisfaction of specifications, while the variations provide a dynamic interpretation of how strictly the constraints were met under varying conditions.

We then conduct 100 simulations under random initial conditions, and the results are summarized in Table 1. `sat` and `vio` denote the number of satisfied and violated cases for each SaSTL formula, respectively; and $\rho_{\text{ave}}$ represents the average robustness degree. It can be observed that all four task semantics achieve a high satisfaction rate, with $\varphi_1$ and $\varphi_4$ satisfied in all simulations, while $\varphi_2$ and $\varphi_3$ exhibit minor violations. Specifically, minor violations of $\varphi_2$ occur when some UAVs slightly exceed the prescribed convergence tolerance $\varepsilon$. Such deviations can be mitigated by appropriately enlarging the error bound. Regarding $\varphi_3$, occasional violations arise when the centroids of certain UAV Voronoi cells coincide with obstacle locations, leading to rare UGV collisions, which are unlikely to occur in practical applications due to inherent physical constraints. However, these situations are extremely rare, hence are regarded as special edge cases that do not affect the validity of the proposed framework.

To quantify the advantages of our SaSTL-based framework, we conduct a comparative study. First, a non-hierarchical approach, where a single controller directly enforces all spatial–temporal specifications, is unsuitable for our scenario. In that sense, each agent is required to have global environmental knowledge (e.g., signal distributions $S$ and global obstacle maps), which contradicts with the limited sensing capabilities of UGVs. Thus, we compare the proposed method with a non-monitored baseline, where agents rely solely on the lower-level QP controller without the robustness-driven feedback from the SaSTL monitor. As shown in Table 2, although the baseline controller (Eq. (21)) utilizes CBFs to prevent collisions, it lacks the foresight provided by SaSTL robustness degrees. In scenarios with high-density obstacles, the baseline's robustness $\rho(\varphi_3)$ often approaches the zero-threshold, leaving no margin for sensor noise or unexpected disturbances. Conversely, the proposed framework utilizes the monitor to detect early warnings of low robustness, allowing for proactive trajectory adjustments. This ensures that the system achieves a 35% higher average robustness margin compared to the baseline, significantly enhancing the reliability of the mission in complex environments.

## 5. Conclusion

This paper proposed a two-layer integrated coverage control and path planning framework for UAV–UGV cooperation under SaSTL specifications. UAVs perform aerial coverage and monitor the explored area to gather spatial information. In addition, UGVs are responsible for collision free path planning at the ground level. Complex spatial–temporal specifications are properly decomposed into *Eventually* and *Always* fragments, then the proposed framework integrates APF and time-varying CBFs to achieve both task satisfaction and safety in real time. The SaSTL-based monitoring mechanism further provides quantitative evaluation of system performance and specification satisfaction. Numerical simulations demonstrated that our approach achieves robust cooperation between UAVs and UGVs. Future extensions will involve implemention of our framework on high-fidelity nonlinear UAVs and UGVs, also experiments on robotic platforms to validate its effectiveness under real-world dynamics and large-scale physical constraints.

## CRediT authorship contribution statement

**Shiheng Zhang:** Writing – review & editing, Formal analysis. **Yangrui Zhang:** Resources. **Shaowen Miao:** Data curation. **Yiding Ji:** Supervision, Funding acquisition.

## 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.

## Data availability

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

## References

[1] F.F. Lizzio, E. Capello, G. Guglieri, A review of consensus-based multi-agent UAV implementations, J. Intell. Robot. Syst. 106 (2) (2022) 43.
[2] S. Zhang, Z.-W. Liu, G. Wen, Y.-W. Wang, Accelerated distributed optimization algorithm with malicious nodes, IEEE Trans. Netw. Sci. Eng. 11 (2) (2024) 2238–2248.
[3] S. Zhang, Y. Ji, Resilient consensus control with privacy protection for battery energy storage systems, IEEE Control. Syst. Lett. 9 (2025) 2133–2138.
[4] S. Zhang, Y. Ji, Distributed resilient consensus and demand tracking in battery energy storage systems under adversarial attacks, in: IEEE 19th International Conf. on Control and Automation, 2025, pp. 268–273.
[5] M. Zhong, C.G. Cassandras, Distributed coverage control and data collection with mobile sensor networks, IEEE Trans. Autom. Control 56 (10) (2011) 2445–2455.
[6] J.M. Palacios-Gasós, Z. Talebpour, E. Montijano, C. Sagüés, A. Martinoli, Optimal path planning and coverage control for multi-robot persistent coverage in environments with obstacles, in: IEEE International Conf. on Robotics and Automation, 2017, pp. 1321–1327.
[7] S. Miao, J. Komenda, F. Lin, Hierarchical supervisory control of networked and cyber-attacked discrete-event systems, Automatica 183 (2026) 112578.
[8] K. Zhang, Z. Li, X. Yin, L. Han, Resource provision for cloud-enabled automotive vehicles with a hierarchical model, IEEE Trans. Syst. Man, Cybern.: Syst. 53 (3) (2022) 1466–1478.
[9] M.H. Yousefi, B. Behnam, S. Farahani, An auxiliary framework to facilitate earthquake search and rescue operations in urban regions, Nat. Hazards 120 (12) (2024) 11107–11131.
[10] O. Maler, D. Nickovic, Monitoring temporal properties of continuous signals, in: Intl. Symp. on Formal Techniques in Real-Time and Fault-Tolerant Sys., 2004, pp. 152–166.
[11] D. Sun, J. Chen, S. Mitra, C. Fan, Multi-agent motion planning from signal temporal logic specifications, IEEE Robot. Autom. Lett. 7 (2) (2022) 3451–3458.
[12] D. Gujarathi, I. Saha, MT*: Multi-robot path planning for temporal logic specifications, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2022, pp. 13692–13699.
[13] M. Guo, J. Tumova, D.V. Dimarogonas, Communication-free multi-agent control under local temporal tasks and relative-distance constraints, IEEE Trans. Autom. Control 61 (12) (2016) 3948–3962.
[14] Y. Zhang, T. Jiang, Z. Cai, Y. Ji, Safety-critical control under timed reach-avoid specifications: A backup control barrier function approach, IEEE Control. Syst. Lett. 10 (2026) 109–114.
[15] C.I. Vasile, C. Belta, Sampling-based temporal logic path planning, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 4817–4822.
[16] L. Nenzi, L. Bortolussi, V. Ciancia, M. Loreti, M. Massink, Qualitative and quantitative monitoring of spatio-temporal properties with SSTL, Log. Methods Comput. Sci. 14 (Modal and temporal logics) (2018).
[17] M. Ma, E. Bartocci, E. Lifland, J.A. Stankovic, L. Feng, A novel spatial–temporal specification-based monitoring system for smart cities, IEEE Internet Things J. 8 (15) (2021) 11793–11806.
[18] W. Liu, S. Alsalehi, N. Mehdipour, E. Bartocci, C. Belta, Quantifying the satisfaction of spatio-temporal logic specifications for multi-agent control, IEEE Trans. Autom. Control 70 (8) (2025) 5098–5113.
[19] Y. Zhao, X. Yu, B. Hoxha, G. Fainekos, J. Deshmukh, L. Lindemann, STL-GO: Spatio-temporal logic with graph operators for distributed systems with multiple network topologies, ACM Trans. Embed. Comput. Syst. 24 (5s) (2025) 1–23.
[20] C. Belta, S. Sadraddini, Formal methods for control synthesis: An optimization perspective, Annu. Rev. Cont. Robot. Auto. Sys. 2 (1) (2019) 115–140.
[21] L. Lindemann, D.V. Dimarogonas, Control barrier functions for signal temporal logic tasks, IEEE Control. Syst. Lett. 3 (1) (2019) 96–101.
[22] X. Huang, L. Li, J. Chen, Multi-agent system motion planning under temporal logic specifications and control barrier function, Control. Theory Technol. 18 (3) (2020) 269–278.
[23] I. Haghighi, N. Mehdipour, E. Bartocci, C. Belta, Control from signal temporal logic specifications with smooth cumulative quantitative semantics, in: IEEE 58th Conf. on Decision and Control, 2019, pp. 4361–4366.
[24] A.D. Ames, X. Xu, J.W. Grizzle, P. Tabuada, Control barrier function based quadratic programs for safety critical systems, IEEE Trans. Autom. Control 62 (8) (2017) 3861–3876.
[25] L. Wang, A.D. Ames, M. Egerstedt, Safety barrier certificates for collisions-free multirobot systems, IEEE Trans. Robot. 33 (3) (2017) 661–674.
[26] M. Faghihi, M. Yadegar, M. Bakhtiaridoust, N. Meskin, J. Sharifi, P. Shi, Distributed optimal coverage control in multi-agent systems: Known and unknown environments, Automatica 173 (2025) 112031.
[27] D. Mellinger, V. Kumar, Minimum snap trajectory generation and control for quadrotors, in: IEEE International Conference on Robotics and Automation, 2011, pp. 2520–2525.
[28] C. Gao, X. Wang, X. Chen, B.M. Chen, A hierarchical multi-UAV cooperative framework for infrastructure inspection and reconstruction, Control. Theory Technol. 22 (3) (2024) 394–405.
[29] Z. Wang, S. Wang, Y. Xie, T. Xiong, C. Wang, Autonomous navigation of mobile robots: A hierarchical planning–control framework with integrated DWA and MPC, Sensors (Basel, Switzerland) 25 (7) (2025) 2014.
[30] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks, IEEE Trans. Robot. Autom. 20 (2) (2004) 243–255.
[31] H.K. Khalil, Nonlinear systems, 3rd edition, Pearson, 2001.
[32] R.E. Ladner, M.J. Fischer, Parallel prefix computation, J. ACM 27 (4) (1980) 831–838.

[33] A. Donzé, T. Ferrere, O. Maler, Efficient robust monitoring for STL, in: International Conference on Computer Aided Verification, Springer, 2013, pp. 264–279.
[34] M. Kamali, L.A. Dennis, O. McAree, M. Fisher, S.M. Veres, Formal verification of autonomous vehicle platooning, Sci. Comput. Program. 148 (2017) 88–106.
[35] X. Song, Z. Peng, S. Song, V. Stojanovic, Interval observer design for unobservable switched nonlinear partial differential equation systems and its application, Internat. J. Robust Nonlinear Control 34 (16) (2024) 10990–11009.
[36] V. Djordjevic, L. Dubonjic, M.M. Morato, D. Prsic, V. Stojanovic, Sensor fault estimation for hydraulic servo actuator based on sliding mode observer, Math. Model. Control. 2 (1) (2022) 34–43.
[37] G.S. Lueker, A data structure for orthogonal range queries, in: 19th Annual Symposium on Foundations of Computer Science (Sfcs 1978), IEEE, 1978, pp. 28–34.
[38] Y. Zhang, S. Zhang, S. Miao, Y. Ji, A task decomposition approach for signal temporal logic control, in: 2025 China Auto. Congress, 2025, pp. 3242–3247.
[39] W. Xiao, C. Belta, High-order control barrier functions, IEEE Trans. Autom. Control 67 (7) (2022) 3655–3662.
[40] X. Tan, W.S. Cortez, D.V. Dimarogonas, High-order barrier functions: Robustness, safety, and performance-critical control, IEEE Trans. Autom. Control 67 (6) (2022) 3021–3028.
[41] A.J. Taylor, P. Ong, T.G. Molnar, A.D. Ames, Safe backstepping with control barrier functions, in: IEEE 61st Conf. on Decision and Control, 2022, pp. 5775–5782.
[42] S. Koga, M. Krstic, Safe PDE backstepping QP control with high relative degree CBFs: Stefan model with actuator dynamics, IEEE Trans. Autom. Control 68 (12) (2023) 7195–7208.