Multi-Nonholonomic Robot Exploration with Line-of-Sight Maintenance in Unknown Environments

Zhang, Weijian; Street, Charlie; Mansouri, Masoumeh

Multi-Nonholonomic Robot Exploration with Line-of-Sight Maintenance in Unknown Environments

Weijian Zhang, Charlie Street, Masoumeh Mansouri

***Abstract*— Multi-robot exploration in cluttered, unknown environments depends on inter-robot communication and mutual observation, yet occlusions and nonholonomic motion can easily break line-of-sight (LoS) and fragment the team. This paper investigates distributed exploration with multiple nonholonomic robots under LoS connectivity constraints. We present a closed-loop exploration framework that couples frontier-driven goal selection with LoS-constrained planning. We maintain team connectivity by building a weighted minimum spanning tree online based on visibility and communication distance. This produces robot neighborhood relations that are necessary for enforcing LoS. Each robot then solves a distributed, optimization-based planning problem using only onboard sensing and information from its neighbors. This produces efficient and dynamically feasible trajectories while satisfying collision avoidance and LoS constraints to the robot's current neighbors. We validate our approach in simulation and on hardware, demonstrating improved exploration continuity and reduced team fragmentation in challenging environments.**

I. Introduction

In multi-robot systems, maintaining reliable communication and mutual observability is fundamental to effective coordination [1]. During cooperative exploration, robots must continuously exchange local observations and state information to support task coordination, map fusion, and motion planning. However, in cluttered environments communication links are often unreliable. Onboard radios are constrained by power and bandwidth, so link quality degrades with distance and relay availability [2]. More importantly, obstacles may block the line-of-sight (LoS) between robots, leading to communication outages or severe throughput reduction [3]. In applications such as search and rescue, inspection, and autonomous exploration, such failures can reduce team efficiency and even cause team fragmentation [4]; in hazardous environments, preserving mutual observation is also essential for situational awareness [5].

In much of the multi-robot coordination literature, communication links are often approximated by range-based models [5], [6], which ignore occlusions and can therefore overestimate effective connectivity in cluttered environments [7]. To address occlusion effects, several works incorporate LoS constraints into multi-robot coordination, either through connectivity-preserving controllers or by restricting robots to LoS-feasible regions [4], [8]–[11]. However, cooperative exploration introduces a tighter coupling between connectivity and motion: robots must keep making exploration progress while the map, visibility relations, and navigation goals evolve online. This is particularly challenging for nonholonomic robots, since LoS recovery or connectivity maintenance cannot be achieved by arbitrary instantaneous repositioning, but must instead be realized through dynamically feasible motions subject to heading and turning constraints. As a result, exploration decisions and trajectory generation must be updated online while simultaneously enforcing LoS, collision, and communication constraints. This remains an open challenge for efficient multi-robot exploration in unknown cluttered environments.

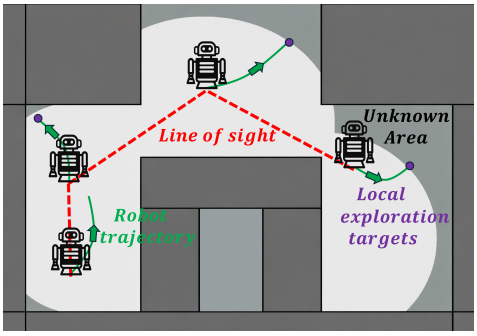

Fig. 1. During exploration in an unknown environment, four nonholonomic robots maintain line-of-sight (LoS) connectivity throughout the mission.

This paper studies cooperative exploration under LoS constraints with teams of nonholonomic robots operating in unknown environments (see Fig. 1). The team must fully explore the environment without breaking the LoS links required for communication and mutual observation. To address this problem, we develop a connectivity-aware exploration framework that tightly couples online mapping, exploration planning, and motion generation. The proposed framework maintains a sparse LoS-consistent neighbor structure, assigns exploration targets from the evolving map, and generates dynamically feasible trajectories by extracting local visible regions from onboard LiDAR observations and incorporating them into distributed motion planning. Unlike LoS-aware methods based on connectivity-preserving controllers [4], [8] or soft visibility costs [12], our framework integrates LoS reasoning with online exploration planning and nonholonomic trajectory generation. This enables dynamically feasible exploration under explicit occlusion, communication, and safety constraints, rather than maintaining connectivity only through reactive control or soft visibility biases. This enables the robot team to maintain LoS connectivity while continuing to explore cluttered environments with dynamically feasible motion.

Weijian Zhang, Charlie Street and Masoumeh Mansouri are with the School of Computer Science at the University of Birmingham, wxz163@student.bham.ac.uk and {c.l.street, m.mansouri}@bham.ac.uk

The main contributions of this paper are as follows: (1) We develop a cooperative exploration framework that enables teams of *nonholonomic robots* to explore unknown environments while maintaining LoS connectivity; (2) We incorporate LoS visibility constraints derived from onboard LiDAR observations directly into *distributed trajectory optimization*, allowing robots to generate dynamically feasible motions that maintain LoS connectivity while respecting collision and communication constraints. We validate our approach in both simulation and hardware.

II. Related Work

Communication Topology and Model. Multi-agent cooperation, including cooperative exploration tasks, often relies on a communication topology to organize information flow and coordination [13]. Existing designs include chain-like topologies that are simple to implement but often restrict task execution to the trailing agent [5], and spanning-tree topologies that maintain team-wide connectivity with few edges and allow multiple agents (as leaves) to work in parallel [4], [11], [14]. To better capture occlusions, visibility-weighted minimum spanning trees (MSTs) have been proposed to encode LoS effects directly into edge weights [9]. Such topologies maintain connectivity with a sparse set of links and are therefore attractive for scalable multi-robot coordination. Regarding the communication model, received signal power is influenced by path loss, shadowing, and multipath [15]. Many works adopt a range-based model for simplicity [5], [6], yet obstacle occlusions can severely degrade link quality and throughput [7]. Hence, in cluttered environments, LoS-based models provide a more practical approximation of inter-agent communication [4], [11]. Although richer channel models that account for reflection and diffraction exist [16], they are typically highly nonlinear and hard to integrate into collaborative motion planning. Therefore, we adopt a LoS-based communication model in this paper.

Line-of-Sight-Constrained Multi-Robot Motion Planning. Maintaining visibility during cooperative motion is challenging, especially when connectivity preservation must be coupled with inter-robot and robot–obstacle collision avoidance. Prior work can be broadly grouped into discrete and continuous connectivity formulations [1]. Discrete methods enforce connectivity only at selected time instants [17], whereas continuous methods maintain connectivity throughout execution, e.g., via potential-function controllers that keep the Laplacian Fiedler eigenvalue positive and quantify LoS using obstacle-to-segment distances [8], or via potential-field formulations in unknown environments [4]. Other approaches use control barrier certificates [14] or geometric Lloyd-type updates [9]. However, these methods are often difficult to extend to nonholonomic robots with control/state constraints. Optimization-based formulations, such as MILP [18], scale poorly with robots and obstacles, while convex restriction methods may be overly conservative in narrow spaces [10]. These limitations motivate planning approaches that can explicitly handle nonholonomic dynamics while maintaining LoS constraints in an efficient manner.

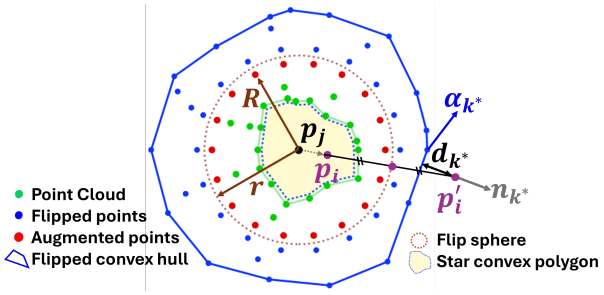

Fig. 2. Illustration of SCP generation in 2D.

III. Preliminaries

A. Robot Kinematics and Communication Graph

We consider a formation of N differential-drive nonholonomic robots performing exploration tasks in an *a priori* unknown environment. For robot $i \in \{1, 2, \ldots, N\}$, $\mathbf{p}_i(t) = [x_i(t),\, y_i(t)]^\top \in \mathbb{R}^2$ is the coordinate of the midpoint of the robot at time t and $\theta_i(t)$ is the robot's orientation in Cartesian space. We adopt a unicycle model [19] to describe robot kinematics, assuming pure rolling without lateral slipping. A trajectory for robot i is defined as $\mathcal{T}_i(t) \triangleq \big(\mathbf{p}_i(t),\, \theta_i(t),\, v_i(t),\, \omega_i(t)\big),\, t \in [0, T]$, where $(v_i(t), \omega_i(t))$ are the translational and angular velocities, respectively. Further, we model the team communication topology at each time step as an undirected, time-varying weighted graph $\mathcal{G}(t) = \langle \mathcal{V}(t), \mathcal{E}(t), w \rangle$, as in [4]. Here, $\mathcal{V}$ is the set of vertices where each vertex is a robot; $\mathcal{E}(t) \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set at time t, where the existence of an edge between vertices i and j implies robots i and j can communicate, and there are no obstacles along the line segment between i and j; and $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geq 0}$ assigns a nonnegative weight that reflects the strength of the connection between two robots. The neighbor set of robot i at time t is defined as $\mathcal{N}_i(t) = \{\, j \in \mathcal{V} \mid (i, j) \in \mathcal{E}(t) \,\}$. We assume that the initial graph $\mathcal{G}(0)$ is connected.

B. Visible Region Construction via Star-Convex Polygons

To enforce LoS during optimization, we introduce a *visible space* for each neighbor pair (i, j). Specifically, for robot i with respect to neighbor robot j, we define $\mathcal{S}_j$ as the set of collision-free positions of robot i from which robot j remains observable, i.e., the line segment connecting $\mathbf{p}_i$ and $\mathbf{p}_j$ is not occluded by any obstacle in the current map. Following [20], we approximate the visible space of robot j by a *Star-Convex Polygon* (SCP) $\mathcal{S}_j$. Here, $\mathcal{S}_j$ denotes the geometric region from which robot j remains visible, while the SCP *construction* refers to the procedure used to estimate this region from onboard point clouds. The construction process is illustrated in Fig. 2 and proceeds as follows. First, we extract the obstacle points surrounding robot j from the local point cloud (the green points in Fig. 2). These points represent nearby obstacles that may occlude the line-of-sight between robots. As the local point cloud can be sparse and incomplete, we add uniformly sampled points on the radius-R boundary to fill gaps and enforce a hard

range bound during hull construction. Next, the union of the local point-cloud samples and the augmented points is mapped through a ball-flipping transformation with respect to the sphere centered at $\mathbf{p}_j$ with radius r (shown as the blue points in Fig. 2):

$$\mathbf{p'}_i = \frac{2r}{\|\mathbf{p}_i - \mathbf{p}_j\|}(\mathbf{p}_i - \mathbf{p}_j) - (\mathbf{p}_i - \mathbf{p}_j), \tag{1}$$

where $\mathbf{p}_i$ is a point in the original point cloud and $\mathbf{p}'_i$ is its flipped position in the transformed space. The flipping transformation converts the non-convex visible region into a convex representation, so that the boundary can be efficiently obtained using a convex hull. Thirdly, we construct the convex hull of all the flipped points using the QuickHull algorithm [21]. The convex hull is the image of the visible space [20]. Finally, applying the inverse transformation of (1) to the convex hull yields the star-convex polygon $\mathcal{S}_j$, which approximates the visible region of robot j. The resulting polygon provides a compact geometric representation of the visibility constraint in the subsequent trajectory optimization.

As shown in Fig. 2, the visibility of a point $\mathbf{p}_i$ from the position $\mathbf{p}_j$ of robot j is determined by the maximal distance between the flipped point $\mathbf{p}'_i$ and all faces of S_j, defined as: $d^{ji}_{k^*} = \max_k \left\{ d^{ji}_k = \mathbf{n}^\top_{ji,k}(\mathbf{p}'_i - \alpha^{ji}_k) \right\}$, where $\mathbf{n}_{ji,k}$ is the outward normal vector of the k-th face of the convex hull composed of K faces, $\alpha^{ji}_k \in \mathbb{R}^2$ is an arbitrary point on face k, $\mathbf{p}'_i$ is the flipped point of $\mathbf{p}_i$ following spherical flipping, and the index k^* is defined as $k^* = \arg\max_{k\in\{1,\dots,K\}} d^{ji}_k$. If $d^{ji}_{k^*} > 0$, then $\mathbf{p}_i$ is visible from $\mathbf{p}_j$, i.e., it is within line-of-sight. To obtain a smooth approximation of $d^{ji}_{k^*}$, the log-sum-exp approximation is used:

$$d^{ji}_{k^*} \approx \frac{1}{\beta} \log \left(\sum_{k=1}^{K} e^{\beta d^{ji}_k} \right), \tag{2}$$

where $\beta > 0$ is a coefficient that controls the quality of approximation [20]. The above process is adopted from the hidden point removal (HPR) operator in point cloud visibility analysis [22].

C. Problem Statement

We study cooperative exploration in an unknown cluttered environment with N robots, where the initial position of each robot is given. The robots that share map updates to maintain a merged occupancy grid map $\mathcal{M}(t)$. Let $\mathcal{F}(t)$ denote the frontier set extracted from $\mathcal{M}(t)$, i.e., the boundary between explored free space and unknown space. In addition to making exploration progress, the team must satisfy a time-varying *line-of-sight (LoS)* connectivity requirement specified by a graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} = \{1, \dots, N\}$ and each edge $(i, j) \in \mathcal{E}(t)$ indicates that robots i and j are required to remain mutually observable (and within communication range). Concretely, LoS means that the straight line segment connecting the robots' planar positions is not occluded by occupied cells in $\mathcal{M}(t)$. Our objective is to generate dynamically feasible and collision-free trajectories $\{\mathcal{T}_i\}_{i=1}^N$ that satisfy LoS constraints for all $(i, j) \in \mathcal{E}(t)$ over execution while keeping $\mathcal{G}(t)$ connected, and to continue exploration until $\mathcal{F}(t) = \emptyset$, indicating full coverage.

IV. LoS-Constrained Exploration Framework

This section presents our closed-loop LoS-constrained exploration framework, which integrates online topology construction, frontier generation, target assignment, and distributed trajectory optimization. At each planning cycle, robots first update the merged map and extract the current frontier set (Alg. 1, lines 2–5), where frontiers represent the boundaries between explored and unexplored regions and serve as candidate exploration targets. The coordinator then broadcasts these neighbor sets together with a global priority order (line 8). The priority order defines the sequence in which robots solve their trajectory optimizations, allowing robots planned earlier to act as references for subsequent robots. Given the assigned exploration targets and neighbor relations, each robot generates a dynamically feasible trajectory toward its target while maintaining LoS connectivity with its one-hop neighbors (lines 12–14). The resulting trajectories are executed in a receding-horizon manner, and the planning loop continuously repeats as the map and frontier set evolve until the frontier set becomes empty.

A. Online MST Topology Construction

To enforce LoS connectivity with minimal coupling, we maintain a sparse connected neighbor structure by computing a minimum spanning tree (MST) online. The MST is constructed centrally by the coordinator using the current robot states and map information, and the resulting neighbor sets are broadcast to all robots. Following the generalized connectivity weighting in [4], each robot pair (i, j) is assigned a link score that captures both distance-limited communication quality and LoS robustness. Specifically, links that violate basic feasibility (e.g., $\|\mathbf{p}_i - \mathbf{p}_j\| > d^{\max}_{\text{com}}$ or LoS is already occluded) are discarded. For feasible pairs, the link score is computed using the sensitivity-aware LoS metric in [4], which jointly captures the current LoS clearance and how rapidly it may decrease under small relative motions. Intuitively, links with larger LoS margin and lower sensitivity correspond to more reliable visibility relations and therefore receive higher scores. Here, sensitivity measures the tendency of the LoS margin to shrink under small motions; in particular, larger angular misalignment between the robot motion direction and the supporting face normal implies a higher risk of LoS loss. The final MST edge cost combines this LoS score with a distance-based communication penalty (see [4] for the exact formulation). The minimum-cost connected topology is then computed using Kruskal's algorithm [23]. The resulting MST contains $\mathcal{O}(N)$ active links and induces the neighbor sets $\mathcal{N}_i(t)$, which specify the pairs of robots required to maintain LoS constraints used during target assignment and distributed trajectory optimization.

Algorithm 1: Exploration

while True **do**
  for $i = 1$ **to** N **do in parallel**
    $\mathcal{S}_i \leftarrow$ **BuildSCP**$(i, \mathbf{p}_i, \mathcal{M})$;
    $\mathcal{F}_i \leftarrow$ **DetectFrontiers**$(i, \mathcal{M})$;
  $\mathcal{F} \leftarrow$ **MergeAndFilterFrontiers**$(\{\mathcal{F}_i\}_{\forall i}, \mathcal{M})$;
  if $\mathcal{F} = \emptyset$ **then**
    break;
  $\mathcal{G} \leftarrow$ **ComputeMSTTopology**$(\{\mathbf{p}_i, \mathcal{S}_i\}_{\forall i})$;
  $\{\mathcal{N}_i\}_{\forall i} \leftarrow$ **ExtractNeighbors**$(\mathcal{G})$;
  $\{\mathbf{g}_i\} \leftarrow$ **AssignTargets**$(\{\mathbf{p}_i\}_{\forall i}, \mathcal{F}, \{\mathcal{N}_i\}_{\forall i}, \mathcal{M})$;
  $\pi \leftarrow$ **ComputePriorityOrder**$(\{\mathcal{N}_i, \mathcal{S}_i, \mathbf{g}_i\}_{\forall i}, \mathcal{M})$;
  for $i \in \pi$ **do**
    $\{\mathcal{S}_j, \mathcal{T}_j\}_{j \in \forall \mathcal{N}_i} \leftarrow$ **GetNeighborInfo**$(i, \mathcal{N}_i)$;
    $\mathcal{T}_i \leftarrow$ **OptimizeTraj**$(\mathbf{g}_i, \{\mathcal{S}_j, \mathcal{T}_j\}_{j \in \forall \mathcal{N}_i}, \mathcal{M})$;
  Execute and update;

B. Frontier Generation

To drive exploration toward unexplored regions, we generate candidate goals from *frontiers*. At each planning cycle, robots share local observations to maintain a merged map $\mathcal{M}(t)$, from which a *global* frontier set is extracted. Specifically, we adopt the OpenCV-based detection and filtering pipeline in [24]. The occupancy grid is first converted into a binary image where free cells adjacent to unknown cells are identified as frontier pixels. Contour and edge operations are then applied to group frontier pixels into connected components, and clustering is used to produce a compact set of frontier centroids. Additional validity filtering removes candidates that lie in occupied regions or yield insufficient information gain. The resulting centroids constitute the frontier candidate set $\mathcal{F}(t)$, which serves as the exploration goal pool for the subsequent target assignment module.

C. Target Assignment

At each planning cycle, the coordinator performs target assignment and assigns each robot i a planar target $\mathbf{g}_i(t) \in \mathbb{R}^2$ from the current frontier set $\mathcal{F}(t)$ under the MST-induced one-hop neighbors $\mathcal{N}_i(t)$. Alg. 2 summarizes the target assignment procedure. We first select an anchor robot–frontier pair $(i^\star, g^\star)$ that minimizes a composite cost

$$c(i, g) = \|\mathbf{p}_i(t) - g\|_2 + \lambda_{\text{los}} \ell(i, g), \tag{3}$$

where the first term measures the Euclidean robot–frontier distance and $\ell(i, g)$ denotes a penalty reflecting whether the corresponding robot–target segment satisfies the LoS visibility condition. The anchor establishes a stable reference direction for the current planning cycle, mitigating collective switching when multiple frontiers compete in cluttered environments. Intuitively, this corresponds to selecting one robot to commit to a nearby exploration direction first, while the remaining robots adjust their assignments relative to this reference, which helps avoid oscillatory target switching that may otherwise occur on robots when multiple frontiers have similar costs (lines 1–2).

The remaining robots are then assigned by solving a minimum-cost bipartite matching problem using the Hungarian algorithm [25]. The assignment cost uses the same metric

Algorithm 2: **AssignTargets**

Input: Robot positions $\{\mathbf{p}_i(t)\}$, frontier set $\mathcal{F}(t)$, MST neighbors $\{\mathcal{N}_i(t)\}$, map $\mathcal{M}(t)$
Output: Targets $\{\mathbf{g}_i(t)\}$
1 $(i^\star, g^\star) \leftarrow \arg\min_{i \in \mathcal{V},\, g \in \mathcal{F}(t)} c(i, g)$;
2 $\mathbf{g}_{i^\star}(t) \leftarrow g^\star$;
3 $\{\mathbf{g}_i(t)\}_{i \neq i^\star} \leftarrow$ **AssignRemainingTargets**$(\{\mathbf{p}_i(t)\}_{i \neq i^\star}, \mathcal{F}(t) \setminus \{g^\star\}, c)$;
4 **while** $\neg$ ***MSTFeasible***$(\{\mathbf{g}_i(t)\}, \{\mathcal{N}_i(t)\}, \mathcal{M}(t))$ **do**
5 $i \leftarrow$ **SelectMaxDeviation**$(\{\mathbf{g}_k(t)\}, i^\star)$;
6 $r \leftarrow$ **SelectRefNeighbor**$(i, \mathcal{N}_i(t), \{\mathbf{g}_k(t)\}, \mathcal{M}(t))$;
7 $\mathbf{g}_i(t) \leftarrow$ **BestCoordTarget**$(r, \{\mathbf{p}_k(t)\}, \mathcal{N}_i(t), \mathcal{M}(t))$;
8 **return** $\{\mathbf{g}_i(t)\}_{\forall i}$;

$c(i, g)$, thereby jointly accounting for geometric proximity and LoS feasibility (line 3). Since the initial assignment may violate connectivity-consistent exploration constraints, we iteratively repair the allocation by verifying feasibility only on the active MST edges (line 4), namely the edges of the current minimum spanning tree that define the required one-hop neighbor relations for maintaining LoS connectivity. The feasibility predicate **MSTFeasible**$(\cdot)$ checks whether all active MST pairs satisfy LoS visibility, admissible inter-target distance bounds, and heading-consistency constraints, while additionally enforcing a minimum separation between assigned targets to avoid duplicate or nearly overlapping goal assignments. Whenever infeasibility is detected, the anchor assignment remains fixed, and the robot whose exploration target exhibits the largest anchor-referenced deviation is selected (line 5). Its exploration target is then replaced by a *coordination target*, i.e., a temporary goal that repositions the robot to help maintain LoS connectivity with its neighbors while other robots continue exploring. The coordination target is generated relative to a reference neighbor $r \in \mathcal{N}_i(t)$ (lines 6–7). Specifically, candidate coordination targets are sampled on a circular ring centered at the reference neighbor with radius d_{des}, which corresponds to the desired inter-robot distance used to maintain visibility-based connectivity. Samples that intersect occupied or inflated obstacle cells, violate minimum inter-robot clearance, or break LoS visibility with the reference neighbor are discarded. Among the remaining feasible samples, the one maximizing the LoS margin is selected. The repair loop terminates once all MST edges satisfy the LoS visibility and distance constraints, yielding the final target set (line 8). The resulting assignments maintain exploration efficiency while remaining consistent with the sparse LoS connectivity structure, thereby providing well-conditioned reference goals for the subsequent distributed trajectory optimization.

D. Priority-Based Distributed Trajectory Optimization

Given the MST neighbor sets $\{\mathcal{N}_i(t)\}$ and the assigned targets $\{\mathbf{g}_i(t)\}$, each robot independently solves a local trajectory optimization problem. While trajectory optimization is performed in a distributed manner on each robot, the planning order is determined centrally by the coordinator through a priority-based sequential planning strategy. At each planning cycle, the coordinator computes a global order $\pi(t)$

by sorting robots in ascending order of the priority cost:

$$c_i(t) = w_e(1-\mathbb{I}(i)) + w_m\,\kappa_i(t) + w_d\,\|\mathbf{p}_i(t)-\mathbf{g}_i(t)\|_2, \quad (4)$$

where w_e, w_m, and w_d are weight parameters. $\mathbb{I}(\cdot)$ is the indicator function, i.e., $\mathbb{I}(i) = 1$ if robot i is assigned an exploration target and 0 otherwise. Moreover, $\kappa_i(t) \triangleq \min_{j\in\mathcal{N}_i(t)} d^{ij}_{k^*}$ is robot i's minimum LoS margin to its one-hop neighbors (defined in (2)), where larger values indicate safer visibility and smaller values indicate a higher risk of LoS loss. Accordingly, the priority rule schedules exploration-target robots first and, within each group, prioritizes robots with smaller LoS margins and shorter distance-to-target, since their motions must be resolved earlier to give the rest of the team more room to preserve connectivity. Robots are optimized sequentially according to $\pi(t)$. When planning for robot i, all already-planned higher-priority trajectories $\{\mathcal{T}_j\}_{j\prec i}$ are treated as time-parameterized moving obstacles, while LoS constraints are enforced only with its one-hop MST neighbors $\mathcal{N}_i(t)$. The priority order is computed centrally by the coordinator, whereas trajectory optimization and trajectory sharing are distributed: after solving, robot i broadcasts $\mathcal{T}_i$ directly to lower-priority robots.

E. Ego-Robot Trajectory Optimization

At this stage, robot i has been assigned a target $\mathbf{g}_i(t)$ and has received the planned trajectories of all higher-priority robots. It then independently solves a local continuous-time trajectory optimization problem over a receding horizon. We adopt a polynomial trajectory representation with $\mathcal{C}^{h-1}$ continuity and minimize a quadratic control effort with time regularization [26]. The θ–s representation is chosen instead of Cartesian x–y parameterization, as it avoids the singularities associated with direction reversals and provides a smooth polynomial form for nonholonomic motion. The optimization problem can be formulated as follows:

$$\begin{aligned}
\min_{\mathbf{c}_i,\mathbf{T}_i}\quad & J_i^{total} = \int_0^T \boldsymbol{\sigma}_i^{(h)}(t)^\top \mathbf{W}\,\boldsymbol{\sigma}_i^{(h)}(t)\,dt \;+\; \lambda_T\,T \\
\text{s.t.}\quad & \boldsymbol{\sigma}_i^{[h-1]}(0) = \boldsymbol{\sigma}_{i,0}^{[h-1]}, \quad \boldsymbol{\sigma}_i^{[h-1]}(T) = \boldsymbol{\sigma}_{i,f}^{[h-1]}, \\
& \boldsymbol{\sigma}_i^{(k)}(T_{i,m}^-) = \boldsymbol{\sigma}_i^{(k)}(T_{i,m}^+), \quad k=0,\dots,h-1,\ \forall m, \\
& [x_i^f, y_i^f]^\top = \mathbf{g}_i(t), \quad T = \sum_{m=1}^{M} T_{i,m}, \quad T_{i,m} > 0, \\
& \mathcal{G}(\cdot) \le 0,\ t\in[0,T],
\end{aligned} \quad (5)$$

where $\mathbf{c}_i$ stacks the polynomial coefficients of the motion-state trajectory and $\mathbf{T}_i \triangleq [T_{i,1},\dots,T_{i,M}]^\top$ are the segment durations, with total horizon $T \triangleq \sum_{m=1}^{M} T_{i,m}$. Here, the trajectory is represented as a continuous piecewise-polynomial curve composed of M consecutive segments over the planning horizon. We optimize a motion-state trajectory with flat output $\boldsymbol{\sigma}_i(t) = [\theta_i(t),\, s_i(t)]^\top$, where $\theta_i(t)$ denotes the robot heading and $s_i(t)$ is the arc length traveled along the path (i.e., the accumulated forward distance). In this representation, the commanded angular and linear speeds satisfy $\omega_i(t) = \dot\theta_i(t)$ and $v_i(t) = \dot s_i(t)$. Each component of $\boldsymbol{\sigma}_i(t)$, i.e., the heading $\theta_i(t)$ and the arc length $s_i(t)$, is represented as an M-segment polynomial of order $2h-1$, and $\boldsymbol{\sigma}_i^{(h)}(t)$ denotes the h-th time derivative. The boundary operator $\boldsymbol{\sigma}_i^{[h-1]}(t)$ stacks derivatives up to order h–1, i.e., $\boldsymbol{\sigma}_i^{[h-1]}(t) \triangleq [\boldsymbol{\sigma}_i(t), \dot{\boldsymbol{\sigma}}_i(t), \dots, \boldsymbol{\sigma}_i^{(h-1)}(t)]$, so constraints in (5) enforce the prescribed initial/final states $\boldsymbol{\sigma}_{i,0}^{[h-1]}$ and $\boldsymbol{\sigma}_{i,f}^{[h-1]}$. To ensure smooth transitions between polynomial segments, we impose continuity constraints at their boundaries. Derivatives up to order $h-1$ are matched at each segment junction, i.e., $\boldsymbol{\sigma}_i^{(k)}(T_{i,m}^-) = \boldsymbol{\sigma}_i^{(k)}(T_{i,m}^+)$, $k=0,\dots,h-1$, at $t = \sum_{\ell=1}^{m} T_{i,\ell}$, yielding $\mathcal{C}^{h-1}$ continuity across segments. The terminal Cartesian position $[x_i^f, y_i^f]^\top$ is computed by forward integrating the unicycle kinematics driven by $(v_i,\omega_i) = (\dot s_i, \dot\theta_i)$ over $[0,T]$ (as in [26]) and is constrained to match the assigned target $\mathbf{g}_i(t)$. The inequality set $\mathcal{G}(\cdot)\le 0$ collects all path and dynamic constraints evaluated along the trajectory, including obstacle avoidance on the merged map, velocity limits, acceleration limits, and the segment-duration balance regularizer to prevent degenerate time allocations [26]. Inter-robot collision avoidance, communication-range constraints, and LoS feasibility are incorporated as additional inequality penalties and are detailed next.

Inter-Robot Collision Avoidance. In robot i's local optimization, we add J_i^R to penalize proximity to the predicted trajectories of higher-priority robots. Let $\mathbf{p}_\phi(t)$ denote the predicted trajectory of a higher-priority robot $\phi \prec i$, where $\phi \prec i$ means $c_\phi(t) < c_i(t)$ and $c_i(t)$ is the priority cost defined in (4). When computing robot i trajectory, the trajectories of all higher-priority robots are treated as fixed. The collision-avoidance penalty is evaluated at discrete trajectory samples $\{t_j\}$ within the planning horizon:

$$\begin{aligned}
J_i^R &= \sum_j \sum_{\phi\prec i} \lambda_R \Big[\max\{\psi_r(t_j,\phi),\,0\}\Big]^3, \\
\psi_r(t_j,\phi) &= d_r^2 - \|\mathbf{p}_i(t_j) - \mathbf{p}_\phi(t_j)\|_2^2,
\end{aligned} \quad (6)$$

where $d_r > 0$ is the desired inter-robot clearance and $\lambda_R > 0$ is a penalty weight. Only constraint-violating samples, i.e., those with $\psi_r(t_j,\phi) > 0$, contribute to the collision penalty. For such samples, the gradient of the penalty with respect to the robot position $\mathbf{p}_i(t_j)$ is

$$\frac{\partial J_i^R}{\partial \mathbf{p}_i(t_j)} = -6\lambda_R\,\psi_r(t_j,\phi)^2\big(\mathbf{p}_i(t_j) - \mathbf{p}_\phi(t_j)\big), \quad (7)$$

and is propagated to the trajectory parameters for gradient-based optimization.

Communication-Range Constraint. Within robot i's local optimization, we further penalize the trajectory when its distance to any current one-hop neighbor exceeds the maximum communication range $d_{\text{com}}^{\max}$. Evaluated at the same discrete samples $\{t_j\}$, we add

$$\begin{aligned}
J_i^C &= \sum_j \sum_{k\in\mathcal{N}_i(t)} \lambda_C \Big[\max\{\psi_c(t_j,k),\,0\}\Big]^3, \\
\psi_c(t_j,k) &= \|\mathbf{p}_i(t_j) - \mathbf{p}_k(t_j)\|_2^2 - \big(d_{\text{com}}^{\max}\big)^2,
\end{aligned} \quad (8)$$

where $\lambda_C > 0$ is a penalty weight and $\mathbf{p}_k(t)$ denotes the predicted trajectory of neighbor k (treated as fixed when $k \prec$

i under the priority-based scheme; otherwise taken from the latest available broadcast/previous cycle). Only samples with $\psi_c(t_j,k)>0$, i.e., those violating the communication-range constraint, contribute to the penalty. For these samples, the penalty term and its gradient are given by:

$$\frac{\partial J_i^C}{\partial \mathbf{p}_i(t_j)} = 6\lambda_C\,\psi_c(t_j,k)^2\big(\mathbf{p}_i(t_j)-\mathbf{p}_k(t_j)\big), \tag{9}$$

and the gradient is propagated to the trajectory parameters during optimization.

Star Convex Polytope Constraints. Recall from Sec. III-B that $\mathcal{S}_j$ is the star-convex approximation of robot j's visible region, and that the spherical flipping transform converts LoS maintenance into a geometric constraint in the transformed space. Thus, maintaining visibility of robot j is equivalent to requiring that the flipped relative position of robot i remains outside $\mathcal{S}_j$:

$$d_{k^*}^{ji} \approx \frac{1}{\beta}\log\left(\sum_{k=1}^{K} e^{\beta d_k^{ji}}\right) > d_{\min}, \tag{10}$$

where $d_{\min}$ is a user-defined safe margin. During robot i's local optimization, we collect the $\{\mathcal{S}_j\}_{j\in\mathcal{N}_i(t)}$ received from its one-hop neighbors and enforce visibility with respect to all of them. Since the assigned target $\mathbf{g}_i(t)$ may lie outside a neighbor's current visible region, directly constraining the full trajectory to remain inside $\mathcal{S}_j$ would be overly restrictive. Instead, we impose the SCP constraint only over the initial horizon segment $[0,\delta T]$, where δT denotes the initial portion of the planning horizon corresponding to the immediate motion phase. Although visibility is fundamentally a hard constraint, we use a soft penalty here for optimization efficiency: the adopted trajectory parameterization allows the penalty weight to be set sufficiently large, so that the penalized problem remains a close approximation of the original hard-constrained one while preserving smoothness for efficient optimization. This design guarantees short-term visibility consistency, while the receding-horizon planning repeatedly enforces the same constraint at each cycle, thereby maintaining LoS throughout execution. To enforce the visibility constraint in a differentiable form, we introduce a penalty that increases as the flipped point approaches the boundary of the star-convex polygon:

$$\begin{aligned}\min_{\mathbf{c}_i,\mathbf{T}_i}\quad & J_i^{total}+J_i^R+J_i^C+\int_0^{\delta T}\lambda_S\sum_{j\in\mathcal{N}_i}{J_{ji}}^S(t)\,dt,\\ {J_{ji}}^S &= \max\left\{d_{\min}-\frac{1}{\beta}\log\left(\sum_{k=1}^{K}e^{\beta d_k^{ji}}\right),0\right\}^3,\end{aligned} \tag{11}$$

where ${J_{ji}}^S$ is $\mathcal{C}^2$ and admits second-order derivatives. The resulting penalty is differentiable, and its gradient with respect to $\mathbf{p}_i$ indicates how the robot position should change to restore visibility:

$$\frac{\partial {J_{ji}}^S}{\partial \mathbf{p}_i(t)} = \begin{cases}-3\lambda_S\left({J_{ji}}^S\right)^2\bar{\mathbf{n}}_{ji}\,\mathbf{M}(\mathbf{p}_{ji}), & {J_{ji}}^S>0,\\ \mathbf{0}, & {J_{ji}}^S\le 0,\end{cases} \tag{12}$$

where

$$\bar{\mathbf{n}}_{ji}=\frac{\sum_{k=1}^{K}e^{\beta d_k^{ji}}\mathbf{n}_{ji,k}}{\sum_{k=1}^{K}e^{\beta d_k^{ji}}},\qquad \mathbf{p}_{ji}=\mathbf{p}_i(t)-\mathbf{p}_j(t),$$

and $\mathbf{M}(\mathbf{p}_{ji})$ is the Jacobian induced by the spherical flipping transform (1):

$$\mathbf{M}(\mathbf{p}_{ji})=\frac{2r}{\|\mathbf{p}_{ji}\|^3}\left(\|\mathbf{p}_{ji}\|^2\mathbf{I}-\mathbf{p}_{ji}\mathbf{p}_{ji}^\top-\frac{\|\mathbf{p}_{ji}\|^3}{2r}\mathbf{I}\right). \tag{13}$$

We incorporate the remaining inequality constraints into the objective via penalty terms, resulting in an unconstrained optimization problem that can be solved efficiently with a gradient-based solver [27]. In practice, we optimize the polynomial coefficients and segment durations using iterative gradient updates with analytic derivatives, allowing real-time planning as the robot executes the trajectory.

LoS-aware Velocity Constraints. Even when the planned trajectory is nominally LoS-feasible, an exploration robot may still move too aggressively and enlarge the LoS gap faster than the neighboring robots responsible for maintaining connectivity can react, especially in cluttered environments with evolving occlusions. To mitigate this effect, we modulate the admissible speed of exploration robots as a function of their current LoS margin. Specifically, for robots assigned exploration targets, we reuse the minimum one-hop LoS margin $\kappa_i(t)\triangleq\min_{j\in\mathcal{N}_i(t)}d_{ij}^{\text{los}}(t)$ and define a smooth scale factor $s_i(t)=1-\exp(-\gamma\,\kappa_i(t))\in(0,1)$, where $\gamma\in(0,1)$ is a constant parameter. We then tighten the velocity bounds as

$$|v_i(t)|\le s_i(t)\,v_{\max},\qquad |\omega_i(t)|\le s_i(t)\,\omega_{\max}. \tag{14}$$

As the LoS margin $\kappa_i(t)$ decreases, indicating a higher risk of losing visibility with neighbors, the scaling factor $s_i(t)$ shrinks and the exploration robot slows down. This provides neighboring robots more time to adjust their motion and restore LoS margins.

F. Emergency Stop under LoS Violation

Each exploration robot monitors its local LoS margin and activates an emergency safeguard when needed, such as at doorways or sharp turns. This is to prevent exploration robots from moving too aggressively and breaking team connectivity. Recall that $\kappa_i(t)$ denotes robot i's minimum LoS margin to its required neighbors, i.e., the distance to the visibility boundary. When this margin becomes too small, indicating a high risk of LoS loss, we trigger a safety stop: if $\kappa_i(t)<\delta_{\text{stop}}$, the exploration robot immediately stops and holds its current state. Meanwhile, coordination robots continue moving toward their coordination targets to restore a comfortable LoS margin. Once the margin recovers above a hysteresis threshold $\kappa_i(t)\ge\delta_{\text{resume}}$, the exploration robot resumes planning and proceeds toward its assigned exploration target. This safeguard mitigates persistent team fragmentation and improves robustness in narrow or highly occluded environments.

V. Experimental Results

In this section, we demonstrate the efficacy of our approach in simulation and on hardware. All simulations are run on Ubuntu 20.04 with an Intel Core i9 processor @ 3.2GHz and 16GB of RAM. All software is implemented in C++ using the Gazebo and Robot Operating System (ROS). Each robot is equipped with an omnidirectional 2D LiDAR, and the laser points that hit neighboring robots are removed from the point cloud. Each robot performs mapping using the gmapping package [28], and the submaps generated by individual robots are merged into a global map using the multi-robot map merge package [29]. We use L-BFGS [27] to solve the optimization problems in (5). Experimental runs can be viewed in the supplementary video. All parameters are tuned by hand to maximize performance. Parameter values are given in the documentation for our corresponding software release.

A. Simulation Experiment

To the best of our knowledge, no prior work has addressed exploration in unknown environments with teams of nonholonomic robots under visibility constraints. Existing studies on visibility-preserving multi-robot systems either rely on prior knowledge of the environment [11], assume omnidirectional robots [4], [8], or impose overly conservative formulations of visibility constraints [9], [10], [12]. As a result, there are no existing approaches that can be directly compared with our method. Instead, we evaluate the effectiveness of the proposed exploration framework through experiments conducted in several challenging environments.

We evaluate the proposed method in 25 complex unknown environments, three representative examples of which are shown in Fig. 3. All experiments are conducted using a team of four differential-drive nonholonomic robots. These environments contain many corners and densely arranged walls, requiring frequent sharp turns and making LoS maintenance particularly challenging. In each experiment, the initial robot formation is randomly generated while ensuring that the initial minimum spanning tree (MST) satisfies the LoS constraints. To assess whether the proposed method can preserve LoS connectivity throughout exploration, we record the formation trajectories and the resulting maps, as shown in Fig. 3(a)–Fig. 3(c), together with the minimum value of the distance metric defined in (10) over all robot pairs in the MST graph at each time step, i.e., $\min_{\mathcal{V}\in\mathcal{G}(t)} d_{k^*}^{ji}$, as shown in Fig. 3(d)–Fig. 3(f). The results show that the maximum distance remains consistently above the predefined LoS margin threshold $d_{\min}$, (set to 4.0 in all experiments), This indicates that every edge in the MST graph satisfies the required visibility margin throughout exploration, and thus the proposed method preserves LoS-constrained graph connectivity. We further evaluate the computation time of each module in the four-robot system, including frontier detection, frontier filtering, MST generation, and distributed trajectory optimization. Table I reports the median per-cycle runtime of each module, aggregated over all planning cycles from the 25 test scenarios. The results show that

TABLE I

Median computation time of each module under the 4-robot setting.

Module	Median time
MST generation (Sec. IV-A)	$46ms$
Frontier detection (Sec. IV-B)	$492ms$
Frontier filtering (Sec. IV-B)	$217ms$
Distributed trajectory optimization (Sec.IV-D)	$53ms$

MST generation is lightweight and can be executed online with negligible overhead. Frontier detection and frontier filtering are relatively more time-consuming, but they can naturally operate at a lower update frequency. In contrast, the distributed trajectory optimization remains highly efficient, enabling rapid reactions to potential LoS loss during exploration. Overall, these results indicate that the proposed framework is suitable for real-time operation. Nevertheless, the current centralized MST construction may become a scalability bottleneck as the team size increases.

B. Real World Experiment

We further demonstrate the performance of the proposed framework in real-world experiments. As shown in Fig. 4, three TurtleBot4 robots equipped with 2D LiDAR sensors start exploring the environment from given initial positions. The results show that the robots are able to consistently maintain LoS connectivity while exploring and mapping the environment, even in the presence of previously unknown obstacles. The complete experiments are shown in the supplementary video.

VI. Conclusion

This paper proposed a LoS-constrained multi-robot exploration framework for cluttered unknown environments. We combine online MST-based topology construction, LoS-aware target assignment, and priority-based distributed trajectory optimization to generate dynamically feasible, collision-free motions while maintaining visibility connectivity. Future work includes scaling to larger teams and incorporating uncertainty-aware neighbor prediction.

References

[1] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot exploration of communication-restricted environments: A survey," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 48–57, 2018.

[2] M. Z. Iskandarani, "Effect of number of nodes and distance between communicating nodes on wsn characteristics," *environments*, vol. 1, p. 6, 2022.

[3] U. T. Virk and K. Haneda, "Modeling human blockage at 5g millimeter-wave frequencies," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 3, pp. 2256–2266, 2019.

[4] R. Bai, S. Yuan, K. Li, H. Guo, W.-Y. Yau, and L. Xie, "Realm: Real-time line-of-sight maintenance in multi-robot navigation with unknown obstacles," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7363–7369, IEEE, 2025.

[5] V. S. Varadharajan, D. St-Onge, B. Adams, and G. Beltrame, "Swarm relays: Distributed self-healing ground-and-air connectivity chains," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5347–5354, 2020.

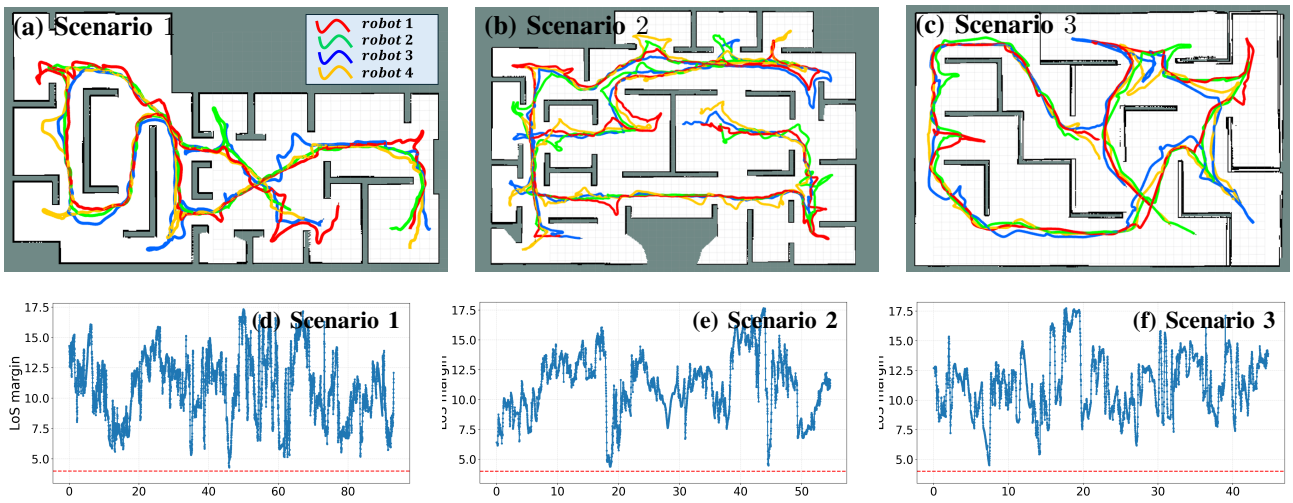

Fig. 3. Exploration results in three representative environments. 3(a)-3(c) Recorded formation trajectories of the four robots and the corresponding maps obtained after exploration. 3(d)-3(f) The minimum pairwise LoS distance, at each time step (blue), together with the predefined threshold $d_{\min}$ (red).

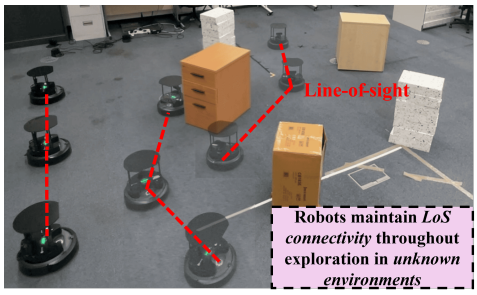

Fig. 4. Real-world exploration experiment with three TurtleBot4. Red dashed lines indicate the LoS connectivity links between robots.

[6] D. M. S. Tan, Y. Ma, J. Liang, Y. C. Chng, Y. Cao, and G. Sartoretti, "Ir 2: Implicit rendezvous for robotic exploration teams under sparse intermittent connectivity," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13245–13252, IEEE, 2024.

[7] S. Gil, S. Kumar, D. Katabi, and D. Rus, "Adaptive communication in multi-robot systems using directionality of signal strength," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 946–968, 2015.

[8] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[9] M. Boldrer, P. Bevilacqua, L. Palopoli, and D. Fontanelli, "Graph connectivity control of a mobile robot network with mixed dynamic multi-tasks," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1934–1941, 2021.

[10] Y. Chen, S. Wang, J. Li, and M. Guo, "Multi-uav deployment in obstacle-cluttered environments with los connectivity," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2772–2779, IEEE, 2025.

[11] L. Xia, B. Deng, J. Pan, X. Zhang, P. Duan, B. Zhou, and H. Cheng, "Relink: Real-time line-of-sight-based deployment framework of multi-robot for maintaining a communication network," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8152–8159, 2023.

[12] B. Jeon, Y. Lee, and H. J. Kim, "Integrated motion planner for real-time aerial videography with a drone in a dense environment," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1243–1249, IEEE, 2020.

[13] M. Mesbahi and M. Egerstedt, "Graph theoretic methods in multiagent networks," 2010.

[14] W. Luo, S. Yi, and K. Sycara, "Behavior mixing with minimum global and subgroup connectivity maintenance for large-scale multi-robot systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9845–9851, IEEE, 2020.

[15] J. Fink, A. Ribeiro, and V. Kumar, "Robust control for mobility and wireless communication in cyber–physical systems with application to robot teams," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 164–178, 2011.

[16] Y. Luo, Y. Wang, H. Chen, C. Wu, X. Lyu, J. Zhou, J. Ma, F. Zhang, and B. Zhou, "Fermi: Flexible radio mapping with a hybrid propagation model and scalable autonomous data collection," *arXiv preprint arXiv:2504.14862*, 2025.

[17] G. A. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, 2012.

[18] A. Caregnato-Neto, M. R. Maximo, and R. J. Afonso, "A line of sight constraint based on intermediary points for connectivity maintenance of multiagent systems using mixed-integer programming," *European Journal of Control*, vol. 68, p. 100671, 2022.

[19] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[20] T. Liu, Q. Wang, X. Zhong, Z. Wang, C. Xu, F. Zhang, and F. Gao, "Star-convex constrained optimization for visibility planning with application to aerial inspection," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7861–7867, IEEE, 2022.

[21] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[22] S. Katz and A. Tal, "On the visibility of point clouds," in *Proceedings of the IEEE international conference on computer vision*, pp. 1350–1358, 2015.

[23] P. B. Guttoski, M. S. Sunye, and F. Silva, "Kruskal's algorithm for query tree optimization," in *11th international database engineering and applications symposium (IDEAS 2007)*, pp. 296–302, IEEE, 2007.

[24] J. A. Placed, J. J. G. Rodríguez, J. D. Tardós, and J. A. Castellanos, "Explorb-slam: Active visual slam exploiting the pose-graph topology," in *Iberian Robotics conference*, pp. 199–210, Springer, 2022.

[25] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[26] M. Zhang, N. Chen, H. Wang, J. Qiu, Z. Han, Q. Ren, C. Xu, F. Gao, and Y. Cao, "Universal trajectory optimization framework for differential drive robot class," *IEEE Transactions on Automation Science and Engineering*, 2025.

[27] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[28] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, pp. 2432–2437, IEEE, 2005.

[29] J. Hörner, "Map-merging for multi-robot system," 2016.