2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-1714-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IROS51168.2021.9636101

Distributed Sampling-based Planning for Non-Myopic Active Information Gathering

Mariliza Tzes[1], Yiannis Kantaros[1], George J. Pappas[1]

***Abstract*— This paper addresses the problem of active information gathering for multi-robot systems. Specifically, we consider scenarios where robots are tasked with reducing uncertainty of dynamical hidden states evolving in complex environments. The majority of existing information gathering approaches are centralized and, therefore, they cannot be applied to distributed robot teams where communication to a central user is not available. To address this challenge, we propose a novel distributed sampling-based planning algorithm that can significantly increase robot and target scalability while decreasing computational cost. In our non-myopic approach, all robots build in parallel local trees exploring the information space and their corresponding motion space. As the robots construct their respective local trees, they communicate with their neighbors to exchange and aggregate their local beliefs about the hidden state through a distributed Kalman filter. We show that the proposed algorithm is probabilistically complete and asymptotically optimal. We provide extensive simulation results that demonstrate the scalability of the proposed algorithm and that it can address large-scale, multi-robot information gathering tasks, that are computationally challenging for centralized methods.**

I. INTRODUCTION

Recent advances in sensing and perception have enabled the deployment of mobile robots in unknown environments for information gathering missions such as environmental monitoring, surveillance, coverage, target tracking and active-SLAM [1]–[5]. To avoid the need for a central user computing sensor-based control policies for large multi-robot systems, distributed control frameworks are needed allowing robots to make decisions locally.

In this paper, we are interested in designing *distributed* control policies for a team of mobile sensing robots tasked with actively reducing the accumulated uncertainty of a dynamic hidden state over an a priori unknown horizon while satisfying user-specified accuracy requirements. In particular, we formulate the Active Information Acquisition (AIA) problem as a deterministic optimal control problem building upon the separation principle presented in [6] where offline/open-loop control policies are optimal. We then propose a novel distributed sampling-based method under which all robots incrementally, in parallel and asynchronously build their own directed trees that explore both their respective robot motion and information space. The robots communicate with each other over an underlying connected communication network to exchange their beliefs about the hidden state which are then fused using a Distributed Kalman Filter (DKF) [7].

Literature Review: Relevant works that address informative planning problems can be categorized into myopic/greedy [8]–[10] and non-myopic (search-based [11]–[14] and sampling-based [15]–[18]). In [10] a *decentralized*, myopic approach is introduced where the robots are driven along the gradient of the Mutual Information (MI) between the targets and the sensor observations. Efficient but suboptimal *search-based* controllers have also been proposed that rely on pruning the exploration process and on addressing the information gathering problem in a *decentralized* way via coordinate descent [12]–[14]. However, these approaches become computationally intractable as the planning horizon or the number of robots increases since decisions are made locally but *sequentially* across the robots. Monte Carlo Tree Search [19] has recently gain popularity for online planning in robotics. Best et. al [20] propose the Dec-MCTS algorithm that efficiently samples the individual action space of each robot contrary to our proposed method and then coordinates with a sparse approximation of the joint action space. To do so, the robots must exchange branches of their Monte-Carlo trees, burdening further the communication needs. Nonmyopic *sampling-based* approaches have also been proposed due to their ability to find feasible solutions very fast. Common in these works is that they are *centralized* and, therefore, as the number of robots or the dimensions of the hidden states increase, the state-space that needs to be explored grows exponentially and, as result, sampling-based approaches also fail to compute sensor policies because of either excessive runtime or memory requirements. A more scalable but centralized sampling-based approach is proposed in [21] that requires communication to a central user for both estimation and control purposes. Building upon our previous work [21], we propose a novel *distributed sampling-based* algorithm that scales well with the number of robots and planning horizon while allowing the robots to *locally* and *simultaneously* explore their physical and reachable information space, exchange their beliefs over an underlying communication network, and fuse them via a DKF.

Contributions: The contribution of this paper can be summarized as follows. *First*, we propose a *distributed* non-myopic sampling-based approach for information-gathering tasks. *Second*, we propose the first distributed sampling-based information gathering algorithm that is probabilistically complete and asymptotically optimal, while it signfinicantly decreases the computational complexity per iteration of its centralized counterpart [21]. *Third*, we provide exten-

This work was supported by the ARL grant DCIST CRA W911NF-17-2-0181.

[1]M. Tzes, Y. Kantaros and G. Pappas are with GRASP Lab, University of Pennsylvania, Philadelphia, PA 19104, USA, {mtzes, kantaros, pappasg}@seas.upenn.edu.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:39 UTC from IEEE Xplore. Restrictions apply.

sive simulation results that show that the proposed method can efficiently handle large-scale estimation tasks.

II. Problem Formulation

A. Robot Dynamics, Hidden State, and Observation Model

Consider a team of N mobile robots that reside in a complex environment $\Omega \subset \mathbb{R}^d$ where d is the dimension of the workspace. Obstacles of arbitrary shape are denoted as $\mathcal{O}$ and the obstacle-free area is $\Omega_{\text{free}} \coloneqq \Omega \backslash \mathcal{O}$. The *dynamics of the robots* are governed by the following equation

$$\mathbf{p}_i(t+1) = f_i(\mathbf{p}_i(t), \mathbf{u}_i(t)), \quad i = \{1, \dots, N\} \tag{1}$$

where $\mathbf{p}_i(t) \in \Omega_{\text{free}}$ describes the state of robot i at discrete time t (e.g position), $\mathbf{u}_i(t) \in \mathcal{U}_i$ is the control input applied to robot i at time t from a *finite* space $\mathcal{U}_i$ of admissible control inputs. The task of the robots is to collaboratively estimate a hidden state $\mathbf{x}(t)$ evolving in Ω_{free} governed by the following dynamics

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + \mathbf{w}(t), \quad \mathbf{w}(t) \sim \mathcal{N}(\mathbf{d}(t), \mathbf{Q}(t)) \tag{2}$$

where $\mathbf{x}(t) \in \mathbb{R}^{d_x}$ and $\mathbf{w}(t) \in \mathbb{R}^{d_w}$ denote the hidden states and the process noise at time t, respectively. Robots are equipped with sensors (e.g., cameras) which allow them to take measurements associated with the unknown state $\mathbf{x}(t)$. Hereafter, we assume that the robots can generate measurements as per the following *observation model*

$$\mathbf{y}_i(t) = M_i(\mathbf{p}_i(t))\mathbf{x}(t) + \mathbf{v}_i(t) \tag{3}$$

where $\mathbf{y}_i(t)$ is the measurement signal at time t and $\mathbf{v}_i(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i(t))$ is a sensor-state dependent measurement noise. Signals observed by robot i are independent of other robots' observations.

Assumption 1: The dynamics of the state $\mathbf{x}(t)$ in (2), the control input $\mathbf{d}(t)$ the observation model (3), and process and measurement noise covariances $Q(t)$ and $R_i(t)$ are known for all time instants t. This assumption is common in the literature [6], [7] and it is required for the application of a Kalman filter to estimate the hidden states.

B. Distributed Kalman Filter for State Estimation

Given a Gaussian prior distribution for $\mathbf{x}(0)$, i.e., $\mathbf{x}(0) \sim \mathcal{N}(\boldsymbol{\mu}(0), \Sigma(0))$, and measurements, denoted by $\mathbf{y}_{i,0:t}$, that robot i has collected until a time instant t, robot i computes a Gaussian distribution that the hidden state follows at time t, denoted by $\hat{\mathbf{x}}_i(t) = \mathcal{N}(\boldsymbol{\mu}_i(t|\mathbf{y}_{i,0:t}), \Sigma_i(t|\mathbf{y}_{i,0:t}))$, where $\boldsymbol{\mu}_i(t|\mathbf{y}_{i,0:t})$ and $\Sigma_i(t|\mathbf{y}_{i,0:t})$ denote the a-posteriori mean and covariance matrix. To compute this local Gaussian distribution, we adopt the *Distributed Kalman Filter* (DKF) algorithm proposed in [7]. To this end, we assume that the robots formulate an underlying communication network modeled as an undirected graph $G = (V, E)$. The sets V and E denote the set of vertices, defined as $V \coloneqq \{1, \dots, N\}$ and indexed by the robots, and set of edges where existence of an edge (i, j) means that robots i and j can directly communicate with each other. Given, such a communication graph, every robot i updates its respective Gaussian distribution as follows:

$$\Omega_i(t+1) = \sum_{j \in \mathcal{N}_i \cup \{i\}} \kappa_{ij} \Omega_j(t) + M_i(t)^T R_i(t)^{-1} M_i(t) \tag{4}$$

where $\Omega_i(t) = \Sigma_i(t)^{-1}$, $\mathcal{N}_i$ is the set of nodes (neighbors) connected to robot i, $\kappa_{ij} > 0, \sum_{j \in \mathcal{N}_i \cup \{i\}} \kappa_{ij} = 1$.

C. Active Information Acquisition

The quality of measurements taken by robot i up to a time instant t, $\mathbf{y}_{i,0:t}$, can be evaluated using information measures, such as the mutual information between $\mathbf{y}_{i,0:t}$ and $\mathbf{x}(t)$ or the conditional entropy of $\mathbf{x}(t)$ given $\mathbf{y}_{i,0:t}$. Given initial robot states $\mathbf{p}_i(0) \in \Omega_{\text{free}}$ and a prior distribution of the hidden states $\mathbf{x}(0)$, our goal is to compute a sequence of control policies $\boldsymbol{\pi}_i(t)\colon \{\mathbf{u}_{i,0:t-1}, \mathbf{y}_{i,0:t}\} \to \mathbf{u}_i(t) \in \mathcal{U}_i$ and a planning horizon F, for all robots $i \in \{1, \dots, N\}$ and time instants $t = \{0, \dots, F\}$ which solves the following stochastic optimal control problem:

$$\min_{F} \max_{\boldsymbol{\pi}_{0:F}} \left[J(F, \boldsymbol{\pi}_{0:F}) = \sum_{i=1}^{N} \sum_{t=0}^{F} \mathbb{I}(\mathbf{x}(t+1); \mathbf{y}_{i,0:t}) \right] \tag{5a}$$

$$\mathbb{I}(\mathbf{x}(F+1); \mathbf{y}_{i,0:F}) \geq \epsilon \textit{ for at least one robot}, \tag{5b}$$

$$\mathbf{p}_i(t+1) = f_i(\mathbf{p}_i(t), \boldsymbol{\pi}_i(t)), \tag{5c}$$

$$\mathbf{p}_i(t+1) \in \Omega_{\text{free}}^N, \tag{5d}$$

$$\mathbf{y}_i(t) = M_i(\mathbf{p}_i(t))\mathbf{x}(t) + \mathbf{v}_i(t), \tag{5e}$$

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + \mathbf{w}(t) \tag{5f}$$

where the objective in (5a) captures the accumulated mutual information between the measurements received up to time t and the state $\mathbf{x}(t+1)$ of all robots and $\boldsymbol{\pi}_{0:F}$ stands for the concatenated sequence of control inputs applied to the robots from $t = 0$ until $t = F$. The constraint in (5b) requires that at the end of the planning there exists at least one robot that has a comprehensive view of the state $\mathbf{x}(F+1)$ with a certain confidence $\epsilon \geq 0$.

The linear relation between measurement $\mathbf{y}_i(t)$ and state $\mathbf{x}(t)$ in (3) and the gaussian assumptions transform the stochastic optimal control problem (5) into a deterministic optimal control problem where open loop policies are optimal [6]. The principle is extended for our case where robots communicate with neighbors $\mathcal{N}_i$ according to a communication graph $G = (V, E)$ and fuse their beliefs via DKF (4) and an open-loop control sequence $\sigma = \mathbf{u}(0), \dots, \mathbf{u}(F)$ exists which is optimal in (5). Furthermore, Problem (5) can be transformed to the following deterministic control problem

$$\min_{F, \boldsymbol{u}_{0:F}} \left[J(F, \boldsymbol{u}_{0:F}) = \sum_{i=1}^{N} \sum_{t=0}^{F} \det \Sigma_i(t+1|\mathbf{y}_{i,0:t}) \right] \tag{6a}$$

$$\det \Sigma_i(F+1) \leq \delta(\epsilon) \textit{ for at least one robot} \tag{6b}$$

$$\mathbf{p}_i(t+1) \in \Omega_{\text{free}}^N, \tag{6c}$$

$$\mathbf{p}_i(t+1) = f_i(\mathbf{p}_i(t), \mathbf{u}_i(t)), \tag{6d}$$

$$\Omega_i(t+1) = \rho(\mathbf{p}_i(t), \Omega_{j \in \mathcal{N}_i \cup \{i\}}(t)), \tag{6e}$$

$$\Sigma_i(t+1) = \Omega_i(t+1)^{-1} \tag{6f}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:39 UTC from IEEE Xplore. Restrictions apply.

where $\rho(\cdot)$ is the DKF update rule in (4), $\boldsymbol{u}_{0:F}$ is the concatenated sequence of control inputs of all robots from $t=0$ until $t=F$ and $\delta(\epsilon)\geq 0$ is the resulting uncertainty threshold from the definition of mutual information and (5b). The robots collaboratively solve the objective (6a) with each constraint applying individually to them. Note that solving (6) requires the robots to exchange their covariance matrices over the communication network G due to (6e).

III. Distributed Sampling Based Active Information Acquisition

A novel distributed sampling-based method to solve (6) is proposed which is summarized in Algorithm 1. In the proposed method, each robot i builds, in parallel with all other robots, a local directed tree that explores both the information space and its robot motion space while exchanging information with neighboring robots collected in $\mathcal{N}_i$, opposed to the global tree built in [21] resulting in decreasing the computational complexity significantly.[1]

The proposed distributed sampling-based algorithm is presented in Algorithm 1. In what follows, we denote the tree built by robot i as $\mathcal{G}_i=\{\mathcal{V}_i,\mathcal{E}_i,J_{\mathcal{G}_i}\}$, where $\mathcal{V}_i$ is the set of nodes and $\mathcal{E}_i$ denotes the set of edges. The set of nodes collects states of the form $\mathbf{q}_i(t)=[\mathbf{p}_i(t),\Sigma_i(t),\mathcal{S}_i(t)]$ where $\mathcal{S}_i(t)$ is a set that collects nodes $\mathbf{q}_j\in\mathcal{V}_j$ for $j\in\mathcal{N}_i$, that participate in the update rule of $\Sigma_i(t)$ as per (4) and $[\mathcal{S}_i(t)]_j=\mathbf{q}_j(t)$, where $[\mathcal{S}_i(t)]_j$ denotes the j-th element in the set $\mathcal{S}_i(t)$; see Section III-A.2. The root of the tree is defined as $\mathbf{q}_i(0)=[\mathbf{p}_i(0),\Sigma_i(0),\mathcal{S}_i(0)]$ where $\mathbf{p}_i(0),\Sigma_i(0)$ are the initial state of the robot and prior covariance, respectively, and $\mathcal{S}_i(0)=\emptyset$.

The cost function $J_{\mathcal{G}_i}\colon\mathcal{V}_i\to\mathbb{R}_+$ describes the cost of reaching a node $\mathbf{q}_i(t)$ starting from the root of the tree $\mathbf{q}_i(0)$. The cost of the root $\mathbf{q}_i(0)$ is $J_{\mathcal{G}_i}(\mathbf{q}(0))=\det\Sigma_i(0)$, while the cost of $\mathbf{q}_i(t+1)=[\mathbf{p}_i(t+1),\Sigma_i(t+1),\mathcal{S}_i(t+1)]$ is equal to

$$J_{\mathcal{G}_i}(\mathbf{q}_i(t+1))=J_{\mathcal{G}_i}(\mathbf{q}_i(t))+\det\Sigma_i(t+1) \tag{7}$$

where $\mathbf{q}_i(t)$ is the parent node of $\mathbf{q}_i(t+1)$. Applying (7) recursively results in $J_{\mathcal{G}_i}(\mathbf{q}_i(t+1))=\sum_{k=0}^{t+1}\det\Sigma_i(k)$ which is equal to the interior sum of the objective in (6). Thus, the summation of costs of individual tree paths over all robots yields the objective function in (6a).

The tree $\mathcal{G}_i$ is initialized so that $\mathcal{V}_i=\{\mathbf{q}_i(0)\}$, $\mathcal{E}_i=\emptyset$ and $J_{\mathcal{G}_i}(\mathbf{q}_i(0))=\det\Sigma_i(0)$ [line 1, Alg. 1]. The construction of the trees lasts for $n_{\max}$ iterations and is based on two procedures *sampling* [line 4-5, Alg. 1] and *extending-the-tree* [line 7-18, Alg. 1]. A solution consisting of a terminal horizon F and a sequence of paths control inputs $\mathbf{u}_{0:F}$ is returned after $n_{\max}$ iterations.

To extract such a solution, we need first to define the goal set $\mathcal{X}_{g_i}\subseteq\mathcal{V}$ that collects all states $\mathbf{q}_i(t)=[\mathbf{p}_i(t),\Sigma_i(t),\mathcal{S}_i(t)]$ in the i-th tree that satisfy $\det\Sigma_i(t)\leq\delta$ [lines 19-20, Algorithm 1]. After $n_{\max}$ iterations, every robot i with non-empty goal set, collected in the set $\mathcal{N}^*$, selects a node in $\mathcal{X}_{g_i}$ and computes the path connecting it to the root of its tree. This path is then propagated to all robots in the network in a multi-hop fashion through the communication graph G. Using the sets $\mathcal{S}_i(t)$, all robots compute a path in their trees that they should follow so that robot $i\in\mathcal{N}^*$ can follow its selected/transmitted path. The resulting paths are then transmitted back to robot i which computes the total cost as per (6a). This process is repeated for all nodes in $\mathcal{X}_{g_i}$ and for all robots i in $\mathcal{N}^*$, [lines 22-25, Alg. 1]. Among all resulting team paths, the robots select the team path with the minimum cost as (6a), [lines 26-27, Alg. 1].

[1]Throughout the paper robot i shall be used as a reference. Same procedure is followed for all the robots unless stated differently.

Algorithm 1: Sampling-based Active Information Acquisition

Input: (i) maximum number of iterations $n_{\max}$, (ii) dynamics (1), (2), observation model (3), (iii) prior Gaussians $\mathcal{N}(\boldsymbol{\mu}_i(0),\Sigma_i(0))$, (iv) initial robot configurations $\boldsymbol{p}_i(0)$
Output: Terminal horizon F, and control inputs $\boldsymbol{u}_{0:F}$

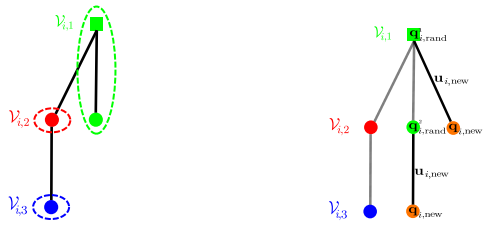
1 Initialize $\mathcal{V}_i=\{\mathbf{q}_i(0)\}$, $\mathcal{E}_i=\emptyset$, $\mathcal{V}_{i,1}=\{\boldsymbol{q}_i(0)\}$, $K_{i,1}=1$ and $\mathcal{X}_g=\emptyset$;
2 **for** $n=1,\ldots,n_{max}$ **do**
3 **parfor** $i=1,\ldots,N$
4 Sample a subset $\mathcal{V}_{i,k_{\text{rand}}}$ from $f_{\mathcal{V}_i}$;
5 Sample a control input $\boldsymbol{u}_{i,\text{new}}\in\mathcal{U}_i$ from $f_{\mathcal{U}_i}$ and compute $\boldsymbol{p}_{i,\text{new}}$;
6 **if** $\boldsymbol{p}_{i,new}\in\Omega^N_{free}$ **then**
7 **for** $\boldsymbol{q}_{i,rand}(t)=[\boldsymbol{p}_{i,rand}(t),\Sigma_{i,rand}(t),\mathcal{S}_{i,rand}(t)]\in\mathcal{V}_{i,k_{rand}}$ **do**
8 Compute sets $\mathcal{Q}_{ij}(t+1)$ for all $j\in\mathcal{N}_i$;
9 Sample node $\boldsymbol{q}_j$ from $f_{\mathcal{Q}_i}$ for all $j\in\mathcal{N}_i$ and compute $\mathcal{S}_{i,\text{new}}$;
10 Compute $\Sigma_{i,\text{new}}(t+1)=\rho(p_{i,\text{rand}}(t),\Sigma_{i,\text{rand}}(t),\Sigma_j(t))$;
11 Construct $\boldsymbol{q}_{i,\text{new}}(t+1)=[\boldsymbol{p}_{i,\text{new}}(t+1),\Sigma_{i,\text{new}}(t+1),\mathcal{S}_{i,\text{new}}]$;
12 Update set of nodes: $\mathcal{V}_i=\mathcal{V}_i\cup\{\boldsymbol{q}_{i,\text{new}}\}$;
13 Update set of edges: $\mathcal{E}_i=\mathcal{E}_i\cup\{(\boldsymbol{q}_{i,\text{rand}},\boldsymbol{q}_{i,\text{new}})\}$;
14 Compute cost of new state: $J_{\mathcal{G}_i}(\boldsymbol{q}_{i,\text{new}})=J_{\mathcal{G}_i}(\boldsymbol{q}_{i,\text{rand}})+\det\Sigma_{i,\text{new}}(t+1)$
15 **if** $\exists k\in\{1,\ldots,K_n\}$ *associated with same position as in* $\boldsymbol{q}_{i,new}$ **then**
16 $\mathcal{V}_{i,k}=\mathcal{V}_{i,k}\cup\{\boldsymbol{q}_{i,\text{new}}\}$;
17 **else**
18 $K_{i,n}=K_{i,n}+1$, $\mathcal{V}_{i,K_n}=\{\boldsymbol{q}_{i,\text{new}}\}$;
19 **if** $\boldsymbol{q}_{new}$ *satisfies* (6b) **then**
20 $\mathcal{X}_{g_i}=\mathcal{X}_{g_i}\cup\{\boldsymbol{q}_{\text{new}}\}$;
21 **parfor** $i\in\mathcal{N}^*$
22 **for** $\mathbf{q}_i(t_k)\in\mathcal{X}_{g_i}$ **do**
23 Compute and transmit path $\mathbf{q}_{i,0:t_k}=[\mathbf{q}_i(0),\ldots,\mathbf{q}_i(t_k)]$;
24 Receive corresponding paths $\mathbf{q}_{j,0:t_k}$;
25 Compute cost $J(t_k)$ as per 6a;
26 Compute minimum cost $J(t_{\text{end}})$;
27 Set $F=t_{\text{end}}$ and execute team paths $\mathbf{q}_{0:t_{\text{end}}}$;

A. Incremental Construction of trees

At each iteration n a node $\mathbf{q}_{i,\text{rand}}\in\mathcal{V}_i$ is selected to be expanded according to a sampling procedure described in Section III-A.1. A new node $\mathbf{q}_{i,\text{new}}=[\mathbf{p}_{i,\text{new}},\Sigma_{i,\text{new}},\mathcal{S}_{i,\text{new}}]$ is added to $\mathcal{V}_i$ and the edge $(\mathbf{q}_{i,\text{rand}},\mathbf{q}_{i,\text{new}})$ is added to $\mathcal{E}_i$; see Section III-A.2.

*1) **Sampling Strategy**:* The sampling procedure begins by first dividing the set of nodes $\mathcal{V}_i$ into a *finite* number of sets $\mathcal{V}_{i,k}\subseteq\mathcal{V}_i$. If $K_{i,n}$ is the number of created sets

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:39 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 1: Fig. 1a shows an example of the sets $\mathcal{V}_{i,k}$. The root of the tree is depicted by a square. There are three ($K_{i,n} = 3$) groups $\mathcal{V}_{i,k}$ and the members of each group have the same color. Group $\mathcal{V}_{i,1}$ consists of nodes $\mathbf{q}_i^1$ and $\mathbf{q}_i^2$ for which $\mathbf{p}_i^1 = \mathbf{p}_i^2$. Figure 1b illustrates the incremental construction of the tree.

during iteration n then $\mathcal{V}_i = \cup_{k=1}^{K_{i,n}} \mathcal{V}_{i,k}$. The set $\mathcal{V}_{i,k}$ collects nodes $\mathbf{q}_i = [\mathbf{p}_i, \Sigma_i, \mathcal{S}_i] \in \mathcal{V}_i$ that share the same robot state $\mathbf{p_i}$; see Fig. 1a. At iteration $n = 1$ it holds that $K_{i,1} = 1$ and $\mathcal{V}_i = \mathcal{V}_{i,1}$ [line 1, Alg. 1]. Once the sets $\mathcal{V}_{i,k}$ are created, the nodes $\mathbf{q}_{i,\text{rand}} \in \mathcal{V}_{i,k_{i,\text{rand}}}$ will be expanded where index $k_{i,\text{rand}}$ is sampled from a given distribution $f_{\mathcal{V}_i}(k_i|\mathcal{V}_i)\colon \{1,\dots,K_{i,n}\} \to [0,1]$ and points to the set $\mathcal{V}_{i,k_{i,\text{rand}}}$[line 4, Alg. 1]. Given nodes $\mathbf{q}_{i,\text{rand}}$ to be expanded with corresponding robot state $\mathbf{p}_{i,\text{rand}}$, a control input $\mathbf{u}_{i,\text{new}}$ is sampled from a distribution $f_{\mathcal{U}_i}(\mathbf{u}_i)\colon \mathcal{U}_i \to [0,1]$. The new robot state $\mathbf{p}_{i,\text{new}}$ is created by applying the selected control input $\mathbf{u}_{i,\text{new}}$ to $\mathbf{p}_{i,\text{rand}}$ according to (1) [line 5, Alg. 1].

***Remark** 1 (Diversity of trees):* The selection of which group $\mathcal{V}_{i,k}$ is expanded is independent between different robots. Due to the randomness imposed by the aforementioned sampling procedures, the constructed trees $\mathcal{G}_i$ at iteration n may have different structures among the robots.

*2) **Extending the tree**:* Once $\mathbf{p}_{i,\text{new}}$ is constructed, we check whether it belongs to the obstacle-free area Ω_{free} or not. In the latter case, the sample is rejected and the sampling procedure is repeated [line 6, Alg. 1]. Otherwise, the state $\mathbf{q}_{i,\text{new}}$ is constructed; see Fig. 1b. Specifically, given the parent node $\mathbf{q}_{i,\text{rand}} = [\mathbf{p}_{i,\text{rand}}, \Sigma_{i,\text{rand}}, \mathcal{S}_{i,\text{rand}}]$ and $\mathbf{p}_{i,\text{new}}$, the construction of $\mathbf{q}_{i,\text{new}}$ requires the computation of $\Sigma_{i,\text{new}}$ and $\mathcal{S}_{i,\text{new}}$. To what follows we assume that $\mathbf{q}_{i,\text{rand}}$ and $\mathbf{q}_{i,\text{new}}$ lie at depth t and $t+1$ respectively.[2] The update rule in (4) requires covariance matrix $\Sigma_i(t)$ obtained from node $\mathbf{q}_i(t)$ and $\Sigma_j(t)$ from neighbors $j \in \mathcal{N}_i$. From Remark 1, at depth t there may exist more than one nodes $\mathbf{q}_j(t) \in \mathcal{V}_j$ to provide $\Sigma_j(t)$ or no nodes. Moreover, the covariance matrix $\Sigma_i(t)$ of node $\mathbf{q}_i(t)$ was once constructed using information received by the set of nodes $\mathcal{S}_i(t) = \{\mathbf{q}_j(t-1) \mid \Sigma_j(t-1)$ *participated in computation of* $\Sigma_i(t)$ *in* (4), $\forall\, j \in \mathcal{N}_i\}$, with $|\mathcal{S}_i(t)| = |\mathcal{N}_i|$ and $[\mathcal{S}_i(t)]_j = \mathbf{q}_j(t-1)$. To maintain consistency with the update rule in (4) only children of the set of nodes $\mathcal{S}_i(t)$ are allowed to distribute their corresponding covariance matrices. For every robot $j \in \mathcal{N}_i$ the set of admissible nodes to share $\Sigma_j(t)$ with robot i is $\mathcal{Q}_{ij}(t) = \{\mathbf{q}_j(t) \mid \mathbf{q}_j(t)$ *is child of node* $[\mathcal{S}_i(t)]_j\}$. Given the set $\mathcal{Q}_{ij}(t)$ node $\tilde{\mathbf{q}}_j(t)$ is sampled from a given discrete distribution $f_{\mathcal{Q}_i}(\mathbf{q}_j)\colon \mathcal{Q}_{ij}(t) \to [0,1]$ and the set $\mathcal{S}_i(t+1)$ is created by collecting the sampled nodes from all $j \in \mathcal{N}_i$ [lines 8-9, Alg. 1]; see also Fig. 2. The covariance matrix $\Sigma_i(t+1)$ is then computed by applying the update rule in (4) and node $\mathbf{q}_{i,\text{new}}(t+1)$ is completed [lines 10-11, Alg. 1].

Next, the set of nodes and edges are updated and the cost of node $\mathbf{q}_{i,\text{new}}$ is computed as $J_{\mathcal{G}_i}(\mathbf{q}_{i,\text{new}}(t+1)) = J_{\mathcal{G}_i}(\mathbf{q}_{i,\text{rand}}(t)) + \det \Sigma_{i,\text{new}}(t+1)$ [lines 12-14, Alg. 1]. Finally, the sets $\mathcal{V}_{i,k}$ are updated, so that if there already exists a subset $\mathcal{V}_{i,k}$ with the same configuration as the state $\mathbf{p}_{i,\text{new}}$, then $\mathcal{V}_{i,k} = \mathcal{V}_{i,k} \cup \{\mathbf{q}_{i,\text{new}}(t+1)\}$. Otherwise, a new subset $\mathcal{V}_{i,k} = \{\mathbf{q}_{i,\text{new}}\}$ is created and the number of subsets increases by one, i.e. $K_{i,n} = K_{i,n} + 1$ [lines 16-18, Alg. 1].

***Remark** 2 (Communication):* The update of $\Sigma_{i,\text{new}}$ in Section III-A.2 is the only procedure for which robots need to communicate with each other to build their trees. The information robot i sends to neighbors $\mathcal{N}_i$ is the set $\mathcal{S}_{i,\text{rand}}$. In exchange it receives covariance matrices Σ_j.

***Remark** 3 (Set of candidate nodes):* In the case where the set of candidate nodes $\mathcal{Q}_{ij}(t)$ is empty, covariance matrix of node $[\mathcal{S}_i(t)]_j$ will be transmitted.

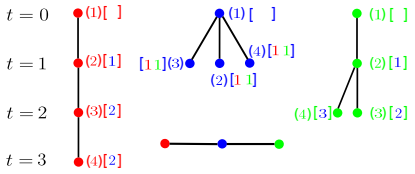


Fig. 2: This Figure illustrates the computation of $\Sigma_{i,\text{new}}$ and $\mathcal{S}_{i,\text{new}}$ for a group of three robots, $i \in \{\text{red, blue, green}\}$. The communication graph is depicted at the bottom. The growth of the trees for four iterations is presented where node (k) is added to the trees. In square brackets, sets $\mathcal{S}_i$ are described. Red node (4) awaits information from the blue robot, while the parent red node (3) transmits its own $\mathcal{S}_{\text{red}}$, i.e. informs that it once communicated with the blue node (2). Blue node (2) is a leaf therefore $\mathcal{Q}_{\text{red}} = \{2\}$ and so $\mathcal{S}_{\text{red,new}} = [2]$. Similarly for green node (4), $\mathcal{Q}_{\text{green}} = \{2,3,4\}$ and blue node (3) is sampled from $f_{\mathcal{Q}}$ to constitute $\mathcal{S}_{\text{green,new}}$.

IV. Complexity, Completeness and Optimality

In this section, we present the formal guarantees of Algorithm 1 and present its computational complexity per iteration compared to its centralized counterpart [21]. The proofs of the following theorems can be found in [22].

***Assumption** 2 (**Probability mass function** $f_{\mathcal{V}_i}$):* (i) Probability mass function $f_{\mathcal{V}_i}(k_i|\mathcal{V}_i)$ satisfies $f_{\mathcal{V}_i}(k_i|\mathcal{V}_i) \geq \epsilon, \forall k_i \in \{1,\dots,K_{i,n}\}, \forall n \geq 0$ for some $\epsilon > 0$ that remains constant across all iterations (ii) $k_{i,\text{rand}}$ are drawn independently across iterations.

***Assumption** 3 (**Probability mass function** $f_{\mathcal{U}_i}$):* (i) Probability mass function $f_{\mathcal{U}_i}$ satisfies $f_{\mathcal{U}_i}(\mathbf{u}_i) \geq \zeta, \forall \mathbf{u}_i \in \mathcal{U}_i, \forall n \geq 0$, for some $\zeta > 0$ that remains constant across all iterations (ii) Samples $\mathbf{u}_i$ are drawn independently across iterations.

***Assumption** 4 (**Probability mass function** $f_{\mathcal{Q}_i}$):* (i) Probability mass function $f_{\mathcal{Q}_i}$ satisfies $f_{\mathcal{Q}_i}(\mathbf{q}_j) \geq \xi, \forall \mathbf{q}_j \in \mathcal{Q}_{ij}, \forall n \geq 0$, where $\xi > 0$ (ii) Samples $\mathbf{q}_j$ are drawn independently across iterations.

***Theorem** 1 (**Probabilistic Completeness**): If there exists a solution to Problem 6, then Algorithm 1 is probabilistically*

[2]To simplify notation we drop the *rand* and *new* subscripts since the relation breaks down to parent and child node where depths t and $t+1$ discriminates them.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:39 UTC from IEEE Xplore. Restrictions apply.

complete, i.e., feasible paths $\boldsymbol{q}_{i,0:F} = \boldsymbol{q}_i(0), \ldots, \boldsymbol{q}_i(F)$, $\boldsymbol{q}_i(f) \in \mathcal{V}_i$, *for all* $f \in \{0, \ldots, F\}$ *and* $i \in \{1, \ldots, N\}$, *will be found with probability* 1, *as* $n \to \infty$.

***Theorem** 2 (**Asymptotic Optimality**): Assume that there exists an optimal solution to Problem 6. Then, Algorithm 1 is asymptotically optimal, i.e., the optimal paths* $\boldsymbol{q}^*_{i,0:F} = \boldsymbol{q}_i(0), \boldsymbol{q}_i(1), \boldsymbol{q}_i(2), \ldots, \boldsymbol{q}_i(F)$ *for all* $i = \{1, \ldots, N\}$, *will be found with probability* 1, *as* $n \to \infty$.

***Theorem** 3 (**Complexity Per Iteration**): Let* $d_{\max}$ *denote the maximum degree of the vertices of the communication graph* G, N *be the number of robots and* k *be the number of nodes* $\mathbf{q}_{rand}$ *to be expanded in one iteration. Then the computational complexity per iteration of Algorithm* 1 *and the centralized approach [21] is* $\mathcal{O}(kd_{\max})$ *and* $\mathcal{O}(kN)$ *respectively, under a worst-case scenario complexity analysis.*

V. Numerical Experiments

In this section, we present numerical experiments that illustrate the performance of Algorithm 1 for the application of target localization and tracking problem. We are interested in examining the scalability of Algorithm 1 and compare performances for various communication networks. All case studies have been implemented using Python 3.6 on a computer with Intel Core i10 1.3GHz and 16Gb RAM.

In this section, each robot uses differential drive dynamics with $\mathbf{p}_i(t)$ describing the position and orientation of robot i at time t. The control commands use motion primitives $\{\nu_i, \omega_i\}$, $\nu_i \in \{0, 0.2, 1\}$m/s, $\omega_i \in \{0, \pm 5, \pm 10, \pm 20, \pm 30, \pm 45, \pm 60\}^\circ$/s. The hidden state $\mathbf{x}(t)$ now collects the positions of all targets at time t, i.e $\mathbf{x}(t) = [\mathbf{x}_1(t)^T, \ldots, \mathbf{x}_M(t)^T]^T$, where $\mathbf{x}_k(t)$ is the position of target k at time t and M is the number of targets. In this application, we require the constraint (6f) to hold for all targets $\mathbf{x}_k(F+1)$ for some δ_i^k, i.e $\det \Sigma_i^k \leq \delta_i^k$ where Σ_i^k is the corresponding covariance matrix of target k. Furthermore, we assume that the robots are equipped with omnidirectional, range-only, line-of-sight sensors with limited range of 1m. We further use the observation model as in [21, Section VI.A]. The observation model is linearized about the predicted target position so that the separation principle can still be applied. In all case studies the targets are modeled as linear systems and the threshold parameter is selected to be $\delta_i^k = 1.8 \times 10^{-5}$ for all targets. The robots reside in the $10m \times 10m$ environment shown in Fig. 4. The DKF parameters are chosen as $k_{ii} = 0.75$ and $k_{ij} = 0.25/|\mathcal{N}_i|$ if $\det \Sigma_i^l \leq \det \Sigma_j^l$ for a target l; otherwise the values interchange. Finally, we incorporate a target assignment similar to that of [21, Section V].

A. Biased Sampling

In what follows, we design the mass functions $f_{\mathcal{V}_i}, f_{\mathcal{U}_i}$ as in [21, Section V] enabling Algorithm 1 to address large-scale estimation problems. The mass function $f_{\mathcal{S}_i}$ is designed such that when more than one candidate nodes $\mathbf{q}_j \in \mathcal{Q}_{ij}$ are available, node $\mathbf{q}_j^*$ with the smallest uncertainty, i.e $|\Sigma_j^*| = \arg\min_{\mathbf{q}_j \in \mathcal{Q}_{ij}} |\Sigma_j|$ is selected with higher probability. Given the set $\mathcal{Q}_{ij}$ the mass function $f_{\mathcal{S}_i}$ is the following

TABLE I: Scalability Analysis

	Connected		Fully	Centralized
N/M	A.D	Runtime / F	Runtime / F	Runtime / F
10/10	5.60	2.09 secs / 28	1.45 secs / 15	7.670 secs / 11
10/20	5.00	3.63 secs / 33	2.93 secs / 26	15.03 secs / 18
10/35	5.40	5.69 secs / 40	3.37 secs / 25	25.37 secs / 19
15/20	8.10	2.72 secs / 26	2.60 secs / 26	14.30 secs / 13
15/35	8.50	4.41 secs / 34	3.77 secs / 26	25.24 secs / 15
20/20	11.1	2.93 secs / 28	2.11 secs / 17	23.14 secs / 14
20/25	10.5	4.00 secs / 33	2.51 secs / 19	21.27 secs / 12
20/35	10.7	4.64 secs / 32	3.51 secs / 24	26.02 secs / 11
30/50	17.0	5.32 secs / 26	4.29 secs / 19	46.02 secs / 10

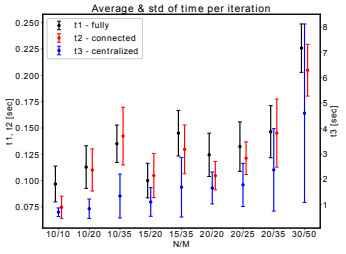


Fig. 3: This Figure illustrates the average time needed for a completion of one iteration across the entire planning horizon F for N robots, M targets for three cases (i) random connected network (red), (ii) fully connected network (black) and (iii) centralized method (blue).

$$f_{\mathcal{S}_i}(\boldsymbol{q}_j) = \begin{cases} p_{\mathcal{S}_i}, & \text{if } \boldsymbol{q}_j = \boldsymbol{q}_j^* \\ (1 - p_{\mathcal{S}_i}) \frac{1}{|\mathcal{Q}_{ij}|}, & \text{otherwise,} \end{cases} \tag{8}$$

where $p_{\mathcal{S}_i} \in (0.5, 1)$ is the probability that node $\mathbf{q}_j$ is selected to provide covariance matrix $\Sigma_j(t)$.

B. Scalability Performance

In this section, we examine the scalability of the proposed distributed scheme with respect to the number of robots N and targets M for different communication networks. In Table I we provide averaged results from ten Monte-Carlo trials for random and fully connected communication networks and the centralized approach in [21]. Especially for the random communication networks we state the average degree (A.D) of the vertices pointing to the robots. The results come from a sequential implementation of Algorithm 1 and the time required by the "slowest" robot to build its tree is reported in Table I for the distributed scheme.

We can notice that our distributed approach computes feasible paths very fast for all connectivity cases regardless of the number of robots and targets. The planning horizon F of the random connected networks is always larger since any transmitted information needs more timesteps until diffused opposed to the centralized approach which fuses observations of all robots and therefore the desired uncertainty threshold is reached faster. In Fig. 3 the average time of completion of one iteration is presented for all scenarios of Table I. We can see that the distributed approach improves the computational complexity per iteration by at least a factor of 10.

C. Effect of the Communication Network Structure

In this section we examine the performance of Algorithm 1 for different communication graph densities. Specifically, in Fig. 4, the trajectories for three different communication

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:39 UTC from IEEE Xplore. Restrictions apply.

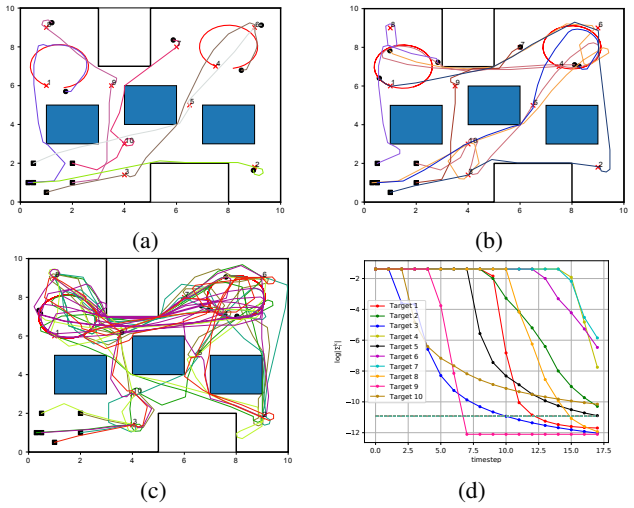


Fig. 4: Trajectories of $N = 6$ robots and $M = 10$ targets for three communication networks (a) fully connected, (b) connected and (c) no communication. The initial positions of the targets are depicted as red crosses and the trajectory of moving targets can be seen in solid red color. The initial and final position of the robots are depicted by black squares and circles respectively. In Fig. 4d the evolution of the uncertainties $\log\det\Sigma_i^k$ over time are presented for the "purple" robot of Fig. 4a. The dashed line depicts the threshold $\log\delta_i^k$.

scenarios are presented for $N = 6$ robots and $M = 10$ targets (i) fully connected graph (ii) connected graph with three edges average degree and (iii) no communication. In the current example, the planning terminates once a robot realizes that the targets have been identified either by itself and/or by the neighboring robots. The returned planning horizons of the aforementioned cases are $F = [17, 29, 120]$, where for the disconnected network the maximum planning horizon out of the individual planning horizons of each robot is stated. The density of the communication network affects the returned paths and the planning horizon F. Specifically, the less dense a graph is the more the planning horizon F increases, since the fusion of the robots' beliefs requires further time. As a result, robots may revisit targets that have already been identified as in the case of Fig. 4b, 4c. In Fig. 4d we can see that the robot becomes responsible only for two targets and realizes that there exist robots that will reduce the remaining uncertainties below the desired threshold.

VI. Conclusions

A new distributed sampling-based algorithm for active information gathering is introduced in this paper followed by formal guarantees. The proposed scheme is tested via extensive simulations showing that feasible paths are quickly computed for large-scale problems, improving significantly also the computational time of existing scalable centralized method. In the future, time-varying communication graphs will be incorporated in the proposed framework.

References

[1] K.-C. Ma, L. Liu, and G. S. Sukhatme, "Informative planning and online learning with sparse gaussian processes," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4292–4298.

[2] E. W. Frew and J. Elston, "Target assignment for integrated search and tracking by active robot networks," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2354–2359.

[3] M. Tzes, S. Papatheodorou, and A. Tzes, "Visual area coverage by heterogeneous aerial agents under imprecise localization," *IEEE control systems letters*, vol. 2, no. 4, pp. 623–628, 2018.

[4] D. J. Bucci and P. K. Varshney, "Decentralized multi-target tracking in urban environments: Overview and challenges," in *2019 22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–8.

[5] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri, "Active slam and exploration with particle filters using kullback-leibler divergence," *Journal of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 291–311, 2014.

[6] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6447–6454.

[7] N. Atanasov, R. Tron, V. M. Preciado, and G. J. Pappas, "Joint estimation and localization in sensor networks," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6875–6882.

[8] T. H. Chung, J. W. Burdick, and R. M. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 2416–2422.

[9] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2009.

[10] P. Dames, M. Schwager, V. Kumar, and D. Rus, "A decentralized control policy for adaptive information gathering in hazardous environments," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 2807–2813.

[11] J. Le Ny and G. J. Pappas, "On trajectory optimization for active sensing in gaussian process models," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 6286–6292.

[12] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.

[13] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.

[14] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4775–4782.

[15] D. Levine, B. Luders, and J. P. How, "Information-rich path planning with general constraints using rapidly-exploring random trees," in *AIAA Infotech at Aerospace Conference, Atlanta, GA*, 2010.

[16] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.

[17] R. Khodayi-mehr, Y. Kantaros, and M. M. Zavlanos, "Distributed state estimation using intermittently connected robot networks," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 709–724, 2019.

[18] X. Lan and M. Schwager, "Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016.

[19] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[20] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-mcts: Decentralized planning for multi-robot active perception," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 316–337, 2019.

[21] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering." in *Robotics: Science and Systems*, 2019, pp. 22–26.

[22] M. Tzes, Y. Kantaros, and G. J. Pappas, "Technical report: Distributed sampling-based planning for non-myopic active information gathering," 2021.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:39 UTC from IEEE Xplore. Restrictions apply.