2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Scalable Active Information Acquisition for Multi-Robot Systems

Yiannis Kantaros, and George J. Pappas

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561244

***Abstract*— This paper proposes a novel highly scalable nonmyopic planning algorithm for multi-robot Active Information Acquisition (AIA) tasks. AIA scenarios include target localization and tracking, active SLAM, surveillance, environmental monitoring and others. The objective is to compute control policies for multiple robots which minimize the accumulated uncertainty of a static hidden state over an *a priori* unknown horizon. The majority of existing AIA approaches are centralized and, therefore, face scaling challenges. To mitigate this issue, we propose an online algorithm that relies on decomposing the AIA task into local tasks via a dynamic space-partitioning method. The local subtasks are formulated online and require the robots to switch between exploration and active information gathering roles depending on their functionality in the environment. The switching process is tightly integrated with optimizing information gathering giving rise to a hybrid control approach. We show that the proposed decomposition-based algorithm is probabilistically complete for homogeneous sensor teams and under linearity and Gaussian assumptions. We provide extensive simulation results showing that the proposed algorithm can address large-scale estimation tasks that are computationally challenging to solve using existing centralized approaches.**

I. Introduction

This paper addresses the problem of designing control policies for a team of mobile homogeneous sensors which minimize the accumulated uncertainty of a static landmarks located at uncertain positions over an *a priori* unknown horizon while satisfying user-specified accuracy thresholds; see Figure 1. First, we formulate this AIA problem as a centralized stochastic optimal control problem which generates an optimal terminal horizon and a sequence of optimal control policies given measurements to be collected in the future. Under Gaussian and linearity assumptions we can convert the problem into a deterministic optimal control problem, for which optimal control policies can be designed *offline*. To design sensor policies, we propose a novel algorithm that relies on decomposing the centralized deterministic optimal control problem into local ones via dynamically tessellating the environment into Voronoi cells assuming global information about the hidden state is available. The local subproblems are formulated and solved online and require the robots to switch between exploration and information gathering roles depending on their functionality in the environment. Specifically, a robot switches to an information gathering role if landmarks are estimated to be located inside its Voronoi cell. In this case, sensor-based control actions to localize the landmarks residing within its Voronoi cell are generated by applying our recently proposed sampling-based approach that simultaneously explores both the robot motion space and the information space reachable by the sensors [1]. On the other hand, a robot adopts an exploration role if no landmarks are estimated to be within its Voronoi region. In this case, control actions are generated using existing area coverage or exploration methods that force the robots to spread in the environment and discover landmarks [2]–[7]. As the robots navigate the workspace, they update their Voronoi cells and their roles accordingly. We show that this hybrid control approach results in distributing the burden of information gathering among the robots. Also, we show that the proposed algorithm is probabilistically complete under Gaussian and linearity assumptions. We provide extensive simulation results demonstrating that our algorithm can address large scale estimation tasks that involve hundreds of robots and hidden states with hundreds of dimensions. Finally, we show that the proposed algorithm can also design sensor policies when the linearity assumptions are relaxed.

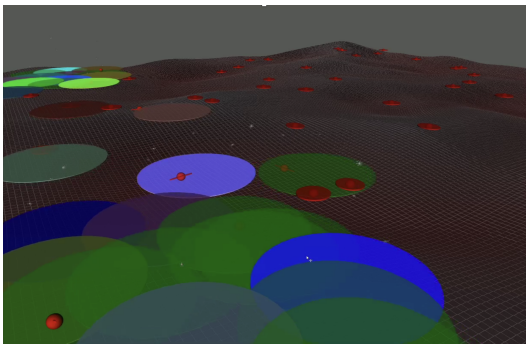


Fig. 1. Landmark localization scenario: 30 UAVs with limited field-of-view (colored disks) navigating an environment to localize 50 landmarks (red spheres) of interest. Red ellipses denote the uncertainty about the landmark positions; see [8].

Literature Review: Relevant approaches to accomplish AIA tasks are typically divided into greedy and nonmyopic. Greedy approaches rely on computing controllers that incur the maximum immediate decrease of an uncertainty measure as, e.g., in [9]–[13], while they are often accompanied with suboptimality guarantees due to submodular functions that quantify the informativeness of paths [14]. Although myopic approaches are usually preferred in practice due to their computational efficiency, they often get trapped in local optima. To mitigate the latter issue, nonmyopic *search-based* approaches have been proposed that sacrifice computational efficiency in order to design optimal paths. For instance, optimal controllers can be designed by exhaustively searching

The authors are with with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA, 19104, USA. {kantaros, pappasg}@seas.upenn.edu. This work was supported by the ARL grant DCIST CRA W911NF-17-2-0181.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:05:17 UTC from IEEE Xplore. Restrictions apply.

the physical and the information space [15]. More computationally efficient but suboptimal 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 [16]–[18]. However, these approaches become computationally intractable as the planning horizon or the number of robots increases as decisions are made locally but *sequentially* across the robots. Nonmyopic *sampling-based* approaches have also been proposed due to their ability to find feasible solutions very fast, see e.g., [19]–[22]. 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. More scalable but centralized sampling-based approaches that can solve target tracking problems and logic-based AIA tasks involving tens of robots and long planning horizons have also been proposed recently in [1], [23]. Scalability comparisons among search-based and sampling-based methods can be found in [1]. In his paper, we propose a new AIA algorithm that is highly scalable due to the fact that the global AIA task is decomposed into local ones while the robots make local decisions *simultaneously* as they navigate the workspace assuming all-to-all communication for information exchange purposes is available.

The **contribution** of this paper can be summarized as follows. *First*, we propose a new highly scalable and non-myopic method that can quickly design control policies achieving desired levels of uncertainty in AIA tasks that involve sensor teams with hundreds of robots and hidden state with hundreds of dimensions. *Second*, we show that the proposed algorithm is probabilistically complete under linearity and Gaussian assumptions. *Third*, we provide extensive simulation results that show that the proposed method can efficiently handle large-scale estimation tasks.

II. Problem Definition: Centralized Active Information Acquisition

Consider N mobile robots that reside in an environment $\Omega \subset \mathbb{R}^d$ with obstacles of arbitrary shape located at $O \subset \Omega$, where d is the dimension of the workspace. The dynamics of the N robots are described by $\mathbf{p}_j(t+1) = \mathbf{f}_j(\mathbf{p}_j(t), \mathbf{u}_j(t))$, for all $j \in \mathcal{N} := \{1, \ldots, N\}$, where $\mathbf{p}_j(t) \in \Omega_{\text{free}} := \Omega \backslash O$ stands for the state (e.g., position and orientation) of robot j in the obstacle-free space Ω_{free} at discrete time t, $\mathbf{u}_j(t) \in \mathcal{U}_j$ stands for a control input in a *finite* space of admissible controls $\mathcal{U}_j$. Hereafter, we compactly denote the dynamics of all robots as $\mathbf{p}(t+1) = \mathbf{f}(\mathbf{p}(t), \mathbf{u}(t))$, where $\mathbf{p}(t) \in \Omega_{\text{free}}^N$, $\forall t \geq 0$, and $\mathbf{u}(t) \in \mathcal{U} := \mathcal{U}_1 \times \cdots \times \mathcal{U}_N$.

The environment is occupied by $M > 0$ *static* landmarks located at uncertain positions $\mathbf{x}_i$, $i \in \mathcal{M} := \{1, \ldots, M\}$. In particular, the robots are not aware of the true landmark positions but they have access to probability distributions over their positions. Such distributions can be user-specified or produced by Simultaneous Localization and Mapping (SLAM) methods [24], [25]. Specifically, we assume that the position of landmark i follows an initially known Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_i(0), \Sigma_i(0))$, where $\hat{\mathbf{x}}_i(0)$ and $\Sigma_i(0)$ denote the expected position and the covariance matrix at time $t = 0$, respectively. Hereafter, we compactly denote the hidden state by $\mathbf{x}$, i.e., $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_M^T]$. Similarly, we denote by $\hat{\mathbf{x}}(0)$ the vector that stacks the expected positions $\hat{\mathbf{x}}_i(0)$ of all landmarks and $\Sigma(0)$ the block diagonal covariance matrix where the diagonal elements are the individual matrices $\Sigma_i(0)$.

The robots are equipped with sensors to collect measurements associated with the hidden states as per the following stochastic linear *observation model* $\mathbf{y}_j(t) = \mathbf{M}(\mathbf{p}_j(t))\mathbf{x} + \mathbf{v}(t)$, where $\mathbf{y}_j(t)$ is the measurement signal at discrete time t taken by robot $j \in \mathcal{N}$ associated with the landmark positions $\mathbf{x}$. Also, $\mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(t))$ is sensor-state-dependent Gaussian noise with covariance $\mathbf{R}(t)$. Note that we assume that the robots are homogeneous in terms of their sensing capabilities.[1] For simplicity, we compactly denote the observation models of all robots as $\mathbf{y}(t) = \bar{\mathbf{M}}(\mathbf{p}(t))\mathbf{x} + \bar{\mathbf{v}}(t)$, $\bar{\mathbf{v}}(t) \sim \mathcal{N}(\mathbf{0}, \bar{\mathbf{R}}(t))$. Hereafter, we assume the robots can exchange their collected measurements via an all-to-all communication network allowing them to always maintain a common/global estimate of the hidden state. The quality of measurements taken by all robots up to a time instant t, collected in a vector denoted by $\mathbf{y}_{0:t}$, can be evaluated using information measures, such as the mutual information between $\mathbf{y}_{0:t}$ and $\mathbf{x}$, the conditional entropy of $\mathbf{x}$ given $\mathbf{y}_{0:t}$, or the determinant of the a-posteriori covariance matrix, denoted by $\Sigma(t|\mathbf{y}_{0:t})$. Note that the a-posteriori mean, denoted by $\hat{\mathbf{x}}(t|\mathbf{y}_{0:t})$. and a-posteriori covariance matrix $\Sigma(t|\mathbf{y}_{0:t})$ can be computed using probabilistic inference methods, e.g., Kalman filter.

Given the initial robot configuration $\mathbf{p}(0)$ and the Gaussian distributions $\mathcal{N}(\mathbf{x}_i(0), \Sigma_i(0))$ for the hidden states, our goal is to select a *finite* horizon $F \geq 0$ and compute control inputs $\mathbf{u}(t)$, for all time instants $t \in \{0, \ldots, F\}$, that solve the following stochastic optimal control problem:

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

$$\det \Sigma(F|\mathbf{y}_{0:F}) \leq \delta, \tag{1b}$$

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

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

$$\mathbf{y}(t) = \bar{\mathbf{M}}(\mathbf{p}(t))\mathbf{x} + \bar{\mathbf{v}}(t), \tag{1e}$$

where the constraints hold for all time instants $t \in \{0, \ldots, F\}$. In (1a), $\mathbf{u}_{0:F}$ stands for the sequence of control inputs applied from $t = 0$ until $t = F$. In words, the objective function (1a) captures the cumulative uncertainty in the estimation of $\mathbf{x}$ after fusing information collected by all robots from $t = 0$ up to time F. The first constraint (1b) requires the terminal uncertainty of $\mathbf{x}$ to be below a user-specified threshold δ. The second constraint (1c) requires that the robots should never collide with obstacles. The last

[1] This assumption is used in Section IV to ensure that the proposed decomposition-based approach is complete.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:05:17 UTC from IEEE Xplore. Restrictions apply.

two constraints capture the robot dynamics and the sensor model.

The Active Information Acquisition (AIA) problem in (1) is a stochastic optimal control problem for which, in general, closed-loop control policies are optimal. Nevertheless, given the linear observation models, we can apply the separation principle presented in [26] to convert (1) to the following deterministic optimal control problem.

$$\min_{F,\mathbf{u}_{0:F}} \left[J(F,\mathbf{u}_{0:F}) = \sum_{t=0}^{F} \det \Sigma(t) \right] \tag{2a}$$

$$\det \Sigma(F) \leq \delta, \tag{2b}$$

$$\mathbf{p}(t) \in \Omega_{\text{free}}^{N}, \tag{2c}$$

$$\mathbf{p}(t+1) = \mathbf{f}(\mathbf{p}(t),\mathbf{u}(t)), \tag{2d}$$

$$\Sigma(t+1) = \rho(\mathbf{p}(t+1),\Sigma(t)), \tag{2e}$$

where $\rho(\cdot)$ stands for the Kalman Filter Ricatti map. Note that open loop (offline) policies are optimal solutions to (2). The problem addressed in this paper can be summarized as follows.

Problem 1: (Active Information Acquisition) Given an initial robot configuration $\mathbf{p}(0)$ and a Gaussian prior distribution $\mathcal{N}(\hat{\mathbf{x}}(0),\Sigma(0))$ for the hidden state $\mathbf{x}$, select a horizon F and compute control inputs $\mathbf{u}(t)$ for all time instants $t \in \{0,\dots,F\}$ as per (2).

Throughout the paper we make the following assumption allows for application of a Kalman filter; see [26].

Assumption 2.1: The measurement noise covariance matrices $\mathbf{R}(t)$ are known for all time $t \geq 0$.

Remark 2.2 (Optimal Control Problem (1)*):* In (1), any other optimality metric, not necessarily information-based, can be used in place of (1a) as long as it is always positive. If non-positive metrics are selected, e.g., the entropy of $\mathbf{x}$, then (1) is not well-defined, since the optimal terminal horizon F is infinite. On the other hand, in the first constraint (1b), any uncertainty measure can be used without any restrictions, e.g., scalar functions of the covariance matrix, or mutual information. Moreover, note that without the terminal constraint (1b), the optimal solution of (1) is all robots to stay put, i.e., $F = 0$.

III. Distributed Active Information Acquisition

A centralized and offline solution to (2) is proposed in [1] that is shown to be optimal under linearity and Gaussian assumptions; nevertheless, a centralized solution can incur a high computational cost as it requires exploring a high dimensional joint space composed of the space of multi-robot states and covariance matrices. To mitigate these challenges, we propose a distributed and online - but sub-optimal - approach that allows the robots to make local decisions, assuming all-to-all communication, to solve (2) as they navigate the environment.

A. Decomposition of AIA Task & Robot Roles

The proposed algorithm is summarized in Algorithm 1. The key idea relies on decomposing (2) into local optimal control problems, that are formulated and solved online to generate individual robot actions. Decomposition of (2) into local problems is attained by assigning landmarks to each robot online via a *dynamic* space-partitioning method.

Specifically, at any time $t \geq 0$, the environment is decomposed into Voronoi cells generated by the robot positions $\mathbf{p}_j(t)$. The Voronoi cell assigned to robot j at time t, denoted by $V_j(t)$, is defined as $V_j(t) = \{\mathbf{q} \in \Omega \mid \|\mathbf{p}_j(t) - \mathbf{q}\| \leq \|\mathbf{p}_e(t) - \mathbf{q}\|,\ \forall e \neq j\}$, i.e., $V_j(t)$ collects all locations $\mathbf{q}$ in the workspace that are closer to robot j than to any other robot $e \neq j$ at time $t \geq 0$. Observe that it holds that $\bigcup_{j\in\mathcal{N}} V_j(t) = \Omega$.

Given $V_j(t)$, robot j is responsible for *actively decreasing the uncertainty of the landmarks that are expected/estimated to lie within its Voronoi cell*. Formally, the landmarks that robot j is responsible for at time t are collected in the following set:

$$\mathcal{A}_j(t) = \{i \in \mathcal{M} \mid (\hat{\mathbf{x}}_i(t) \in \mathcal{V}_j(t)) \wedge (\det \Sigma_j(t) > \delta)\}. \tag{3}$$

In words, $\mathcal{A}_j(t)$ collects the landmarks that have not been localized with accuracy determined by δ and are estimated to be closer to robot j than to any other robot. By definition of the set of $\mathcal{A}_j(t)$ and the fact that $\bigcup_{j\in\mathcal{N}} V_j(t) = \Omega$, we have that every landmark is assigned to exactly one robot and that there are no unassigned landmarks, i.e., $\bigcup_{j\in\mathcal{N}} \mathcal{A}_j(t) = \mathcal{M}$. Notice that the set $\mathcal{A}_j(t)$ may be empty in case the estimated positions of all landmarks are outside the j-th Voronoi cell. Depending on the emptiness of the sets $\mathcal{A}_j(t)$, the robots switch between *exploration* and *AIA* roles in the workspace, as follows.

1) Local AIA Strategy: If at time $t = \bar{t}$ there are landmarks assigned to robot j, i.e., $\mathcal{A}_j(\bar{t}) \neq \emptyset$, then robot j solves the following local AIA problem:

$$\min_{F_j,\mathbf{u}_{j,\bar{t}:F_j}} \left[J(F_j,\mathbf{u}_{\bar{t}:F_j}) = \sum_{t=\bar{t}}^{F_j} \sum_{i\in\mathcal{A}_j(\bar{t})} \det \Sigma_i(t) \right] \tag{4a}$$

$$\det \Sigma_i(F_j) \leq \delta, \forall i \in \mathcal{A}_j(\bar{t}) \tag{4b}$$

$$\mathbf{p}_j(t) \in \Omega_{\text{free}}, \tag{4c}$$

$$\mathbf{p}_j(t+1) = \mathbf{f}_j(\mathbf{p}_j(t),\mathbf{u}_j(t)), \tag{4d}$$

$$\Sigma(t+1) = \rho(\mathbf{p}_j(t+1),\Sigma(t)), \tag{4e}$$

where the constraints (4c)-(4e) hold for all time instants $t \in [\bar{t}, F_j]$. Note that (4) can viewed as local version of (2). In words, (4) requires robot j to compute a terminal horizon F_j and a sequence of control inputs over this horizon, denoted by $\mathbf{u}_{j,\bar{t}:F_j}$, so that (i) the accumulated uncertainty over the assigned targets is minimized (see (4a)), (ii) the terminal uncertainty of the assigned targets is below a user-specified threshold (see (4b)) while (iii) avoiding obstacles and respecting the robot dynamics and the Kalman filter Ricatti equation (see (4c)-(4e)). Note that in (4e), the Kalman filter Ricatti equation is applied using only the position of robot j and not the multi-robot state as opposed to (2e). In other words, in (4e), the covariance matrix $\Sigma(t)$ is computed using only local information. For simplicity, we do not add any dependence to robot j to $\Sigma(t)$. Equivalence between the

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:05:17 UTC from IEEE Xplore. Restrictions apply.

centralized AIA problem (2) and the local AIA problems in (4) is discussed in Section IV. A computationally efficient method to quickly solve the nonlinear optimal control problem (4) is discussed in Section III-B.

2) Exploration Strategy: If at time $t = \bar{t}$, there are no landmarks assigned to robot j, i.e., $\mathcal{A}_j(\bar{t}) = \emptyset$, then robot j is responsible for *exploring* the environment so that eventually it holds that $\mathcal{A}_j(t) \neq \emptyset$. In this way, the burden of localizing the landmarks is shared among the robots. The latter can be accomplished if these robots adopt an exploration/mapping or an area coverage strategy. For instance, in [4], [27] a frontiers-based exploration strategy is proposed. Specifically, dummy "exploration" landmarks with locations at the current map frontiers are considered with a known Gaussian prior on their locations. This fake uncertainty in the exploration-landmark locations promises information gain to the robots. In [3], [28], area coverage methods are proposed assuming robots with range-limited sensors. Specifically, the robots follow a gradient-based policy to maximize an area coverage objective that captures the part of the workspace that lies within the sensing range of all robots. Alternatively, robots with $\mathcal{A}_j(\bar{t}) = \emptyset$ can be recruited from robots e with $\mathcal{A}_e(\bar{t}) \neq \emptyset$ to help them localize the corresponding landmarks.

B. Local Planning for Non-myopic AIA

In this section, we present a computationally efficient method to solve the local optimal control problem (4). Specifically, to solve (4), we employ our recently proposed sampling-based algorithm [1] that incrementally constructs a directed tree that explores both the information space and the physical space.

In what follows, we denote by $\mathcal{G}_j = \{\mathcal{V}_j, \mathcal{E}_j, J_{\mathcal{G},j}\}$ the tree constructed by robot j to solve (4), where $\mathcal{V}_j$ is the set of nodes and $\mathcal{E}_j \subseteq \mathcal{V}_j \times \mathcal{V}_j$ denotes the set of edges. The set of nodes $\mathcal{V}_j$ contains states of the form $\mathbf{q}_j(t) = [\mathbf{p}_j(t), \Sigma(t)]$. The function $J_{\mathcal{G}} : \mathcal{V}_j \to \mathbb{R}_+$ assigns the cost of reaching node $\mathbf{q}_j \in \mathcal{V}_j$ from the root of the tree. The root of the tree, denoted by $\mathbf{q}_j(0)$, is constructed so that it matches the initial state $\mathbf{p}_j(0)$ of robot j and the prior covariance $\Sigma(0)$, i.e., $\mathbf{q}_j(0) = [\mathbf{p}_j(0), \Sigma(0)]$. For simplicity of notation, hereafter we drop the dependence of the tree on robot j. The cost of the root $\mathbf{q}(0)$ is $J_{\mathcal{G}}(\mathbf{q}(0)) = \det \Sigma(0)$, while the cost of a node $\mathbf{q}(t+1) = [\mathbf{p}_j(t+1), \Sigma(t+1)] \in \mathcal{V}$, given its parent node $\mathbf{q}(t) = [\mathbf{p}_j(t), \Sigma(t)] \in \mathcal{V}$, is computed as $J_{\mathcal{G}}(\mathbf{q}(t+1)) = J_{\mathcal{G}}(\mathbf{q}_j(t)) + \det \Sigma(t+1)$. Observe that by applying this cost function recursively, we get that $J_{\mathcal{G}}(\mathbf{q}(t+1)) = J(t, \mathbf{u}_{0:t+1})$ which is the objective function in (4).

The tree $\mathcal{G}$ is initialized so that $\mathcal{V} = \{\mathbf{q}(0)\}$, $\mathcal{E} = \emptyset$, and $J_{\mathcal{G}}(\mathbf{q}(0)) = \det \Sigma(0)$. Also, the tree is built incrementally by adding new states $\mathbf{q}_{\text{new}}$ to $\mathcal{V}$ and corresponding edges to $\mathcal{E}$, at every iteration n of the sampling-based algorithm, based on a *sampling* and *extending-the-tree* operation. After taking $n_{\max} \geq 0$ samples, where $n_{\max}$ is user-specified, the sampling-based algorithm terminates and returns a solution to Problem 1, i.e., a terminal horizon F and a sequence of control inputs $\mathbf{u}_{0:F}$. To extract such a solution, we need first to define the set $\mathcal{X}_g \subseteq \mathcal{V}$ that collects all states $\mathbf{q}(t) = [\mathbf{p}_j(t), \Sigma(t)] \in \mathcal{V}$ of the tree that satisfy $\det \Sigma(F_j) \leq \delta$, for all landmarks that belong to $\mathcal{A}_j(t)$, which is the constraint (4b). Then, among all nodes $\mathcal{X}_g$, we select the node $\mathbf{q}(t) \in \mathcal{X}_g$, with the smallest cost $J_{\mathcal{G}}(\mathbf{q}(t))$, denoted by $\mathbf{q}(t_{\text{end}})$. Then, the terminal horizon is $F = t_{\text{end}}$, and the control inputs $\mathbf{u}_{0:F}$ are recovered by computing the path $\mathbf{q}_{0:t_{\text{end}}}$ in $\mathcal{G}$ that connects $\mathbf{q}(t_{\text{end}})$ to the root $\mathbf{q}(0)$, i.e., $\mathbf{q}_{0:t_{\text{end}}} = \mathbf{q}(0), \ldots, \mathbf{q}(t_{\text{end}})$. Note that satisfaction of the constraints (4c)-(4e) is guaranteed by construction of $\mathcal{G}$; see [1]. The core operations of this algorithm, '*sample*' and '*extend*' used to construct the tree $\mathcal{G}$ can be found in [1].

Algorithm 1: Local Planning for Global AIA Task

Input: (i) robot dynamics; observation model; (iii) prior Gaussian $\mathcal{N}(\hat{\mathbf{x}}(0), \Sigma(0))$; (iv) initial robot configuration $\mathbf{p}_j(0)$
Output: Terminal horizon F, and control inputs $\mathbf{u}_{0:F}$

```
; t = 0;
Compute Voronoi cell V_j(0) and set of assigned targets A_j(0);
while ∃j s.t. det Σ_j(t) > δ do
    if A_j(t) ≠ ∅ then
        if (t = 0) ∨ (A_j(t) ≠ A_j(t − 1)) then
            Compute sequence of controllers u_{j,t:F_j} for AIA [1];
            Apply control input u_j(t);
        else
            Apply the next control input from the most recently computed sequence of control inputs;
    else
        Compute next control input u_j(t) to explore V_j(t);
    Collect sensor measurements;
    Update x̂(t + 1) and Σ(t + 1);
    Update Voronoi cell V_j(t + 1) and set A_j(t + 1);
    t = t + 1;
```

C. Overview of Distributed Hybrid AIA Algorithm

At any time $t \geq 0$ every robot j adopts a role in the environment based on the set $\mathcal{A}_j(t)$. If $\mathcal{A}_j(t) = \emptyset$, then robot j adopts an exploration mode aiming to eventually become responsible for localizing landmarks. If $\mathcal{A}_j(t) \neq \emptyset$, then robot j solves (4) to actively decrease the uncertainty of the assigned landmarks. In both cases, robot j generates a sequence of controls denoted by $\mathbf{u}_{j,t:F_j}$. Once the robots apply their respective control input $\mathbf{u}_j(t)$, they communicate via an all-to-all connected network to share their measurements and positions so that they can (i) maintain a global estimate of the hidden state and (ii) update their Voronoi cells $V_j(t+1)$. Based on the new Voronoi cells, the robots locally compute the new sets of assigned targets $\mathcal{A}_j(t+1)$. The robots for which it holds that $\mathcal{A}_j(t+1) = \mathcal{A}_j(t)$ do not update their corresponding sequence of control actions, i.e., they apply the control input $\mathbf{u}_j(t+1)$ computed at time t. On the other hand, the robots that satisfy $\mathcal{A}_j(t+1) \neq \mathcal{A}_j(t)$ recompute their control actions to accomplish either their assigned AIA or exploration tasks.

D. Extensions

1) Mobile Landmarks: Mobile landmarks can also be considered that are governed by the following known but noisy dynamics: $\mathbf{x}_i(t+1) = \mathbf{g}_i(\mathbf{x}_i(t), \mathbf{a}_i(t), \mathbf{w}_i(t))$, where $\mathbf{a}_i(t)$ and $\mathbf{w}_i(t) \in \mathbb{R}^{d_w}$ are the control input and the process

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:05:17 UTC from IEEE Xplore. Restrictions apply.

noise for landmark i at discrete time t. We assume that the process noise $\mathbf{w}_i(t)$ is uncertain and follows a known normal distribution, i.e., $\mathbf{w}(t) \sim \mathcal{N}(0, \mathbf{Q}(t))$, where $\mathbf{Q}(t)$ is the covariance matrix at time t. Assuming that the control inputs $\mathbf{a}_i(t)$ and the covariance matrices $\mathbf{Q}(t)$ are known to the robots for all $t \geq 0$, [1] can be used to design AIA paths.[2]

2) Nonlinear Observation Models: In Section II, we assumed a sensor model that is linear with respect to the landmark positions allowing us to compute offline the a-posteriori covariance matrix without the need of measurements. This assumption can be relaxed by computing the a-posteriori covariance matrices using the linearized observation model about the estimated landmark positions; see Section V.

IV. Completeness, Optimality & Convergence

In this section, we show that Algorithm 1 is complete for homogeneous sensor networks and static hidden states. This result is formally stated in the following theorem and its proof can be found on [29].

Theorem 4.1 (Completeness): Assume that (i) given the initial multi-robot state $\mathbf{p}(0)$ and the prior distributions $\mathcal{N}(\hat{\mathbf{x}}_i(0), \Sigma_i(0))$, there exists a solution to the centralized optimal control problem (2); (ii) the robot dynamics allow all robots to reach any obstacle-free location in Ω. Under assumptions (i)-(ii), Algorithm 1 is probabilistically complete, i.e., at any time instant t it will generate a terminal horizon F_j and a sequence of control actions $\mathbf{u}_{j,t:F_j}$ for all robots with $\mathcal{A}_j(t) \neq \emptyset$ that solve the respective local problems (4).

Remark 4.2 (Online vs Offline Planning): According to Theorem 4.1, Algorithm 1 can find at time $t = 0$ feasible (offline) paths that solve the centralized problem (4). As it will be shown in Section V, the benefit of dynamically assigning landmarks to robots and accordingly re-planning online is that the state estimation burden is shared among all robots resulting in shorter terminal horizons.

Remark 4.3 (Implementation): A more computationally efficient approach to apply [1] is to define the goal region $\mathcal{X}_g$ so that it requires only one landmark - instead of all - in $\mathcal{A}_j(t)$ to be localized with accuracy δ. Such an approach may not yield the optimal solution to (4) due to its myopic nature but it does not sacrifice completeness. This can be shown by following the same logic as in the proof of Theorem 4.1.

Remark 4.4 (Optimality): To solve the local problem (4), information that may be collected by the rest of the robot team is neglected; see also Section III-C. As a result, the synthesized paths may not constitute an optimal solution to (2). This local design of paths sacrifices optimality but it allows for addressing large-scale estimation tasks.

V. Numerical Experiments

In this section, we present numerical experiments that illustrate the performance of Algorithm 1. The sampling-based AIA algorithm [1] is implemented as in Remark 4.3 using the biased density functions $f_{\mathcal{V}}$ and $f_{\mathcal{U}}$ designed in [1] with parameters $p_{\mathcal{V}} = p_{\mathcal{U}} = 0.9$. All case studies have been implemented using MATLAB 2016b on a computer with Intel Core i7 3.1GHz and 16Gb RAM. Simulation videos can be found in [8].

Robot Dynamics & Sensors: Hereafter, we consider robots with differential drive dynamics, where $\mathbf{p}_j(t)$ captures both the position and the orientation of the robots. Specifically, the available motion primitives are $u \in \{0, 0.1\}$m/s and $\omega \in \{0, 5, 10, \ldots, 350, 355\}$ deg/s. Moreover, we assume that the robots are equipped with omnidirectional, range-only, sensors with limited range of 0.4m while they reside in 10×10 workspace. Every robot j can generate measurements associated with landmark i as per the model $y_{j,i} = \ell_{j,i}(t) + v(t)$ if $(\ell_{j,i}(t) \leq 2)$, where $\ell_{j,i}(t)$ is the distance between landmark i and robot j, and $v(t) \sim \mathcal{N}(0, \sigma^2(\mathbf{p}_j(t), \mathbf{x}_i(t)))$ is the measurement noise. Also, we model the measurement noise so that σ increases linearly with $\ell_{j,i}(t)$, with slope 0.25, as long as $\ell_{j,i}(t) \leq 2$; if $\ell_{j,i}(t) > 2$, then σ is infinite. Observe that this observation model is nonlinear and, therefore, the separation principle, discussed in Section II, does not hold. Thus, we execute the sampling-based algorithm to solve the local sub-problems (4) using the linearized observation model about the estimated landmark positions. Also, the robots with no assigned landmarks, i.e., the robots with $\mathcal{A}_j(t) = \emptyset$, navigate the environment so that they maximize an area coverage metric that captures the region that lies within the sensing range of all robots [28]. The latter forces these robots to spread in the environment so that landmarks eventually reside within their Voronoi cells distributing the AIA burden.

Scalability Analysis: First, we examine the scalability performance of Algorithm 1 with respect to the number of robots and the number of landmarks. The results are summarized in Table I. In all case studies of Table I, the parameter δ is selected to be $\delta = 1.8 \times 10^{-6}$, for all $i \in \mathcal{M}$, while all robots initially reside in the bottom-left corner of the 10m$\times 10$m environment shown in Figure 2. In Table I the first line corresponds to the number of robots and landmarks. In the second line, T_{plan} refers to the average runtime in seconds required for a robot to compute a control action regardless of their role in the workspace (see lines 6 and 11 in Alg. 1). Similarly, T_{vor} shows the average runtime in seconds to compute the Voronoi cells per iteration of Algorithm 1 (see line 15 in Alg. 1). Note that in our implementation, the Voronoi tessellation has been computed in a centralized fashion and, therefore, the corresponding runtime increases as the number of robots increases. Nevertheless, this runtime is always quite small and, therefore, the Voronoi partitioning does not compromise scalability. In the third and fourth row, T_{total} and F refer to the mean total runtime in minutes and total number of discrete time instants of 10 simulations, along with the respective standard deviation, required to find the first feasible solution. Note that Algorithm 1 has not been implemented in parallel across the robots; instead, in our implementation the robots make local but sequential decisions, which explains the increasing runtime T_{total} as the number N of robots increases. Observe in Table I that for a given number of landmarks, increasing the number of

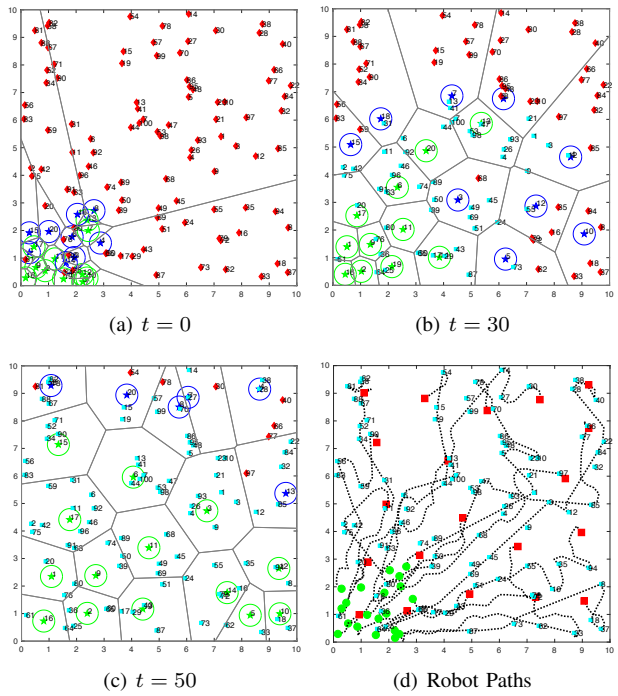
[2]Note that in this case, Algorithm 1 is not complete; see also the proof of Theorem 4.1.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:05:17 UTC from IEEE Xplore. Restrictions apply.

TABLE I

SCALABILITY ANALYSIS: THE RUNTIMES T_{PLAN}, T_{VOR}, AND T_{TOTAL} ARE IN SECS, SECS, AND MINS, RESPECTIVELY.

	N=20 M=10	N=50 M=10	N=100 M=10	N=10 M=100	N=20 M=100	N=50 M=100	N=100 M=100	N=300 M=100	N=100 M=200	N=300 M=200
$T_{\text{plan}}/T_{\text{vor}}$	0.12/0.002	0.13/0.02	0.12/0.05	0.18/0.009	0.19/0.014	0.17/0.03	0.20/0.05	0.26/0.36	0.31/0.06	0.36/0.19
T_{total}	1.7	5.1	9.2	2.9	3.5	7.8	17.8	78.1	30.6	98.4
F	34±2	32±2	30±4	82±5	70±4	63±2	60±2	58±3	68±3	58±7

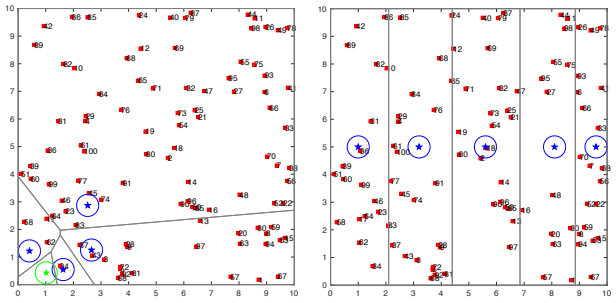


(a) $t = 0$ (b) $t = 30$ (c) $t = 50$ (d) Robot Paths

Fig. 2. Landmark localization scenario: Figures 2(a)-2(c) show the configurations of $N = 20$ robots at various time instants towards localizing $M = 100$ static landmarks. The red diamonds and cyan squares correspond to accurately localized and non-localized landmarks, respectively. The blue and green stars correspond to robots that navigate for information gathering and area coverage purposes, respectively. The circle centered at each robot position illustrates the sensing range while the gray segments depict the Voronoi cells. Figure 2(d) shows the robot paths where the green and red squares denote the initial and final robot positions.

robots tends to decrease the terminal horizon. Note that the terminal horizon depends on the initial robot configuration. For instance, assuming N robots that are initially uniformly distributed and $M = 100$ landmarks, the terminal horizon when $N = 20$ and $N = 100$ is 35±3 and 15±2, respectively. Figure 2 shows the paths that 20 robots followed to localize $M = 100$ landmarks; observe that the robots dynamically switch roles.

Online vs Offline Planning: Second, we examine the effect of dynamically updating the Voronoi partitioning - and, consequently, the landmarks assigned to the robots - and accordingly re-planning. The results are summarized in Table II and pertain to a case study with $N = 5$ and $M = 100$. In Table II, the first row corresponds to three different initial configurations illustrated in Figure 3. The second and the third row show the mean terminal planning horizon F_{offline} and F_{online} of 10 simulations required to localize



(a) Deployment 1 (b) Deployment 2

Fig. 3. Graphical depiction of initial multi-robot deployments in Table II. The third initial deployment in Table II differs from the second one only in that the y-coordinate of all robot positions is 0.1m.

TABLE II

ONLINE VS OFFLINE PLANNING

N=5, M=100	Deployment 1	Deployment 2	Deployment 3
F_{offline}	782 ± 44	190 ± 30	210 ± 30
F_{online}	141 ± 16	137 ± 8	146 ± 26

all landmarks without and with re-planning, respectively. Observe that the total time is always larger when offline planning is considered. This becomes more pronounced in configuration 1 (Fig. 3(a)), as there are robots that have to stay within a small region the whole time which is not the case when online re-planning is considered due to its adaptive nature. Thus, in an online setting the state estimation burden is shared among all robots resulting in shorter terminal horizons.

Effect of Communication: Third, we evaluate the effect of all-time and all-to-all communication among the robots on the estimation performance. Specifically, here we consider a scenario with $N = 50$ robots and $M = 100$ static landmarks and we assume that communication occurs intermittently and periodically, i.e., every $T > 0$ time units. We observed that increasing the period T results in longer terminal planning horizons F, as expected, since the robots plan paths without always having access to global estimates of the hidden state. In particular, for $T = 1$ (all-time communication), $T = 2$, $T = 10$, and $T = 15$, the average planning horizon of 10 simulation studies was 63, 70, 125, and 150, respectively.

VI. CONCLUSION

In this paper, we proposed a new highly scalable, non-myopic, and probabilistically complete planning algorithm for AIA tasks. Extensive simulation studies demonstrated scalability and validated the theoretical analysis.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:05:17 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] 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.

[2] J. Cortés, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[3] Y. Kantaros, M. Thanou, and A. Tzes, "Distributed coverage control for concave areas by a heterogeneous robot–swarm with visibility sensing constraints," *Automatica*, vol. 53, pp. 195–207, 2015.

[4] C. Leung, S. Huang, and G. Dissanayake, "Active slam using model predictive control and attractor based exploration," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5026–5031.

[5] A. J. Smith and G. A. Hollinger, "Distributed inference-based multi-robot exploration," *Autonomous Robots*, vol. 42, no. 8, pp. 1651–1668, 2018.

[6] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019.

[7] J. Wang and B. Englot, "Autonomous exploration with expectation-maximization," in *Robotics Research*. Springer, 2020, pp. 759–774.

[8] Simulation_Video, https://vimeo.com/468212369.

[9] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[10] R. Graham and J. Cortés, "A cooperative deployment strategy for optimal sampling in spatiotemporal estimation," in *47th IEEE Conference on Decision and Control*, 2008, pp. 2432–2437.

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

[12] B. Charrow, V. Kumar, and N. Michael, "Approximate representations for multi-robot control policies that maximize mutual information," *Autonomous Robots*, vol. 37, no. 4, pp. 383–400, 2014.

[13] F. Meyer, H. Wymeersch, M. Fröhle, and F. Hlawatsch, "Distributed estimation with information-seeking control in agent networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2439–2456, 2015.

[14] M. Corah and N. Michael, "Distributed submodular maximization on partition matroids for planning on large sensor networks," in *IEEE Conference on Decision and Control*, Miami Beach, FL, 2018, pp. 6792–6799.

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

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] Y. Kantaros, Q. Jin, and G. Pappas, "Sensor-based temporal logic planning in uncertain semantic maps," *https://arxiv.org/pdf/2012.10490.pdf*, 2020.

[24] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," *arXiv preprint arXiv:1910.02490*, 2019.

[25] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *IEEE International Conference on Robotics and Automation*, Singapore, May-June 2017, pp. 1722–1729.

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

[27] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, 1997, pp. 146–151.

[28] J. Cortés, S. Martinez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.

[29] Y. Kantaros and G. J. Pappas, "Technical report: Scalable active information acquisition for multi-robot systems," in *https://arxiv.org/abs/2103.09364*, 2021.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:05:17 UTC from IEEE Xplore. Restrictions apply.