Resilient Active Information Acquisition With Teams of Robots

Brent Schlotfeldt, *Student Member, IEEE*, Vasileios Tzoumas, *Member, IEEE*, and George J. Pappas, *Fellow, IEEE*

***Abstract*—Emerging applications of collaborative autonomy, such as *multitarget tracking*, *unknown map exploration*, and *persistent surveillance*, require robots plan paths to navigate an environment while maximizing the information collected via on-board sensors. In this article, we consider such information acquisition tasks but in adversarial environments, where attacks may temporarily disable the robots' sensors. We propose the first receding horizon algorithm, aiming for robust and adaptive multirobot planning against any number of attacks, which we call *Resilient Active Information acquisitioN* (RAIN). RAIN calls, in an online fashion, a *robust trajectory planning* (RTP) subroutine that plans attack-robust control inputs over a look-ahead planning horizon. We quantify RTP's performance by bounding its suboptimality. We base our theoretical analysis on notions of curvature introduced in combinatorial optimization. We evaluate RAIN in three information acquisition scenarios: *multitarget tracking*, *occupancy grid mapping*, and *persistent surveillance*. The scenarios are simulated in C++ and a unity-based simulator. In all simulations, RAIN runs in real time, and exhibits superior performance against a state-of-the-art baseline information acquisition algorithm, even in the presence of a high number of attacks. We also demonstrate RAIN's robustness and effectiveness against varying models of attacks (worst case and random), as well as varying replanning rates.**

***Index Terms*—Algorithm design and analysis, autonomous robots, combinatorial mathematics, multiagent systems, reactive-sensor-based mobile planning, robotics in hazardous fields.**

I. Introduction

Active information acquisition has a long history in robotics [1], [2]. The *"active"* characterization captures the idea that robots that move *purposefully* in the environment, *acting* as mobile sensors instead of static, can achieve superior sensing performance. Indeed, active information acquisition has recently been extended to a variety of collaborative (multirobot) autonomy tasks, such as *multitarget tracking* [3], [4], *exploration and mapping* (including *simultaneous localization and mapping*) [5]–[7], *monitoring of hazardous environments* [3], [8], and *persistent surveillance* [9]–[11], Fig. 1.

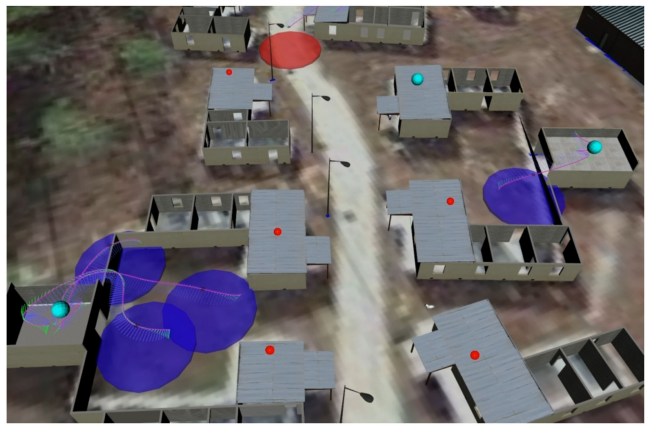


Fig. 1. *Persistent surveillance under attacks.* Unity simulation environment depicting a five-robot team engaging in a *persistent surveillance* task for monitoring a set of buildings. Some robots are under attack. The attacks can disable the sensing capabilities of the robots, at least temporarily. Each blue disc indicates the field of view of a (nonattacked) robot, whereas each red disc indicates an attacked robot. In this adversarial environment, the robots must resiliently plan trajectories to (re)visit all the building landmarks to continue acquiring information despite the attacks.

Robotics research has enabled the solution of several classes of information acquisition problems, both for single-robot and multirobots scenarios, with methods that include search-based [3], [12], sampling-based [13], [14], and gradient-based planning [15]. The methods can also differ on problem parameters, such as the length of the planning horizon—cf., *myopic* (or *one-step-ahead*) planning [16] versus *nonmyopic* (or *long-horizon*) planning [3]—and the type of target process (*e.g.*, Gaussian [17], Gaussian mixture, [5], [18], occupancy grid map [7], [19], and nonparametric [20]).

But most robotics research places not emphasis on robustifying information acquisition against attacks (or failures) that may temporarily disable the robots' sensors, *e.g.*, smoke-cloud attacks that can block temporarily the field of view of multiple sensors. For example, Saulnier *et al.* [21] focus on formation control, instead of information acquisition; Mitra *et al.* [22] focus on state estimation against byzantine attacks, *i.e.*, attacks that corrupt the robots' sensors with adversarial noise, instead of disabling them; and Prorok [23] focuses on a trajectory scheduling in transportation networks when travel times can

Manuscript received August 10, 2020; revised January 30, 2021; accepted April 17, 2021. Date of publication June 10, 2021; date of current version February 8, 2022. This work was supported in part by ARL CRA DCIST W911NF-17-2-0181 and in part by the Rockefeller Foundation. This article was recommended for publication by Associate Editor R. Tron and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. *(Corresponding author: Brent Schlotfeldt.)*

Brent Schlotfeldt and George J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA (e-mail: brentsc@seas.upenn.edu; pappasg@seas.upenn.edu).

Vasileios Tzoumas is with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA (e-mail: vtzoumas@umich.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2021.3082212.

Digital Object Identifier 10.1109/TRO.2021.3082212

be uncertain, instead on trajectory planning for information acquisition. An exception is [16], which however is limited to multitarget tracking based on myopic planning, instead of nonmyopic.

Related work has also been developed in combinatorial optimization [24], [25], paving the way for robust combinatorial optimization against attacks: Orlin *et al.* [26] propose algorithms for submodular set function optimization against worst-case attacks, but under the assumption the attacks can remove (disable) only a limited number of elements from the optimized set. Instead, Tzoumas *et al.* [27] propose algorithms for optimization against any number of removals. And, recently, Tzoumas *et al.* [28] extended the algorithms to matroid constraints, enabling the application of the algorithms in robotics since multirobot path planning can be cast as a matroid-constrained optimization problem [7].

A. Contributions

In this article, in contrast to the aforementioned works, we extend the *attack-free active information acquisition* to account for attacks against the robots while simultaneously performing nonmyopic multirobot planning. We make the following three key contributions.

1) *Receding horizon formulation and algorithm:* Section II formalizes the attack-aware active information acquisition problem as a finite horizon control optimization problem named *Resilient Active Information acquisitioN* (P-RAIN)—in the acronym, the "P" stands for "problem." (P-RAIN) is a sequential mixed-integer optimization problem: it jointly optimizes the robots' control inputs such that the robots' planned paths are robust against worst-case attacks that may occur at each time step. The upper bound to the number of attacks is assumed known and constant. (P-RAIN)'s information objective function is assumed nondecreasing in the number of robots (a natural assumption, since the more robots the more information one typically can collect).
   Section III proposes RAIN, the first receding horizon algorithm for the (P-RAIN). RAIN calls in an online fashion *robust trajectory planning* (RTP), a subroutine that plans attack-robust control inputs over a planning horizon. RTP's planning horizon is typically less than the acquisition problem's finite horizon, for computational reasons. For the same reason, RTP assumes constant attacks. RTP is presented in Section IV.
2) *Performance analysis:* Although no performance guarantees exist for the nonlinear combinatorial optimization problem (P-RAIN), Section V provides suboptimality bounds on RTP's performance, *i.e.*, on the algorithm used by RAIN to approximate a solution to (P-RAIN) *locally*, in a receding horizon fashion. The theoretical analysis is based on notions of curvature introduced for combinatorial optimization; namely, the notions of *curvature* [29] and *total curvature* [30]. The notions aim to bound the worst-case complementarity of the robots' planned paths in their ability to jointly maximize (P-RAIN) information acquisition objective function.
3) *Experiments:* Section VI evaluates RAIN across three multirobot information acquisition tasks: *multitarget tracking*, *occupancy grid mapping*, and *persistent surveillance*. All evaluations demonstrate the necessity for attack-resilient planning, via a comparison with a state-of-the-art baseline information acquisition algorithm, namely, *coordinate descent* [6]. Specifically, RAIN runs in real time and exhibits superior performance in all experiments. RAIN's effectiveness is accentuated the higher the numbers of attacks is (see Section VI-A). RAIN remains effective even against nonworst-case attacks, specifically, random (see Section VI-B). Even when high replanning rates are feasible (see Section VI-C), in which case coordinate descent can adapt at each time step against the observed attacks, RAIN still exhibits superior performance. The algorithm is implemented in C++ and a unity-based simulator.

B. Comparison With Preliminary Results in [31]

This article extends the results in [31], by introducing novel problem formulations, algorithms, and numerical evaluations, as well as by including all proofs (see Appendixes), which were omitted from Schlotfeldt *et al.* [31]. Particularly, the receding horizon formulation (P-RAIN) is novel, generalizing the (P-RTP) formulation first presented in [31]. The algorithm RAIN is also first presented here. Additionally, the simulation evaluations on *occupancy grid mapping* and *persistent surveillance* are new, and have not been previously published. They also include for the first time a sensitivity analysis of RAIN against varying models of attacks (worst case and random), as well as varying replanning rates.

II. RAIN Problem

We present the optimization problem of RAIN(see Section II-B). To this end, we first formalize the (attack-free) *active information acquisition problem* (see Section II-A). We also use the notation, which is as follows.

1) $\phi_{\mathcal{V},\,\tau:\tau+\tau'} \triangleq \{\phi_{i,t}\}_{i\in\mathcal{V},\,t\in[\tau,\ldots,\tau+\tau']}$, for any variable of the form $\phi_{i,t}$, where $\mathcal{V}$ denotes a set of robots (*i.e.*, $i \in \mathcal{V}$ is robot i), and $[\tau,\ldots,\tau+\tau']$ denotes a discrete time interval ($\tau \geq 1$, whereas $\tau' \geq 0$).
2) $w \sim \mathsf{N}(\mu,\Sigma)$ denotes a Gaussian random variable w with mean μ and covariance Σ.

A. Active Information Acquisition in the Absence of Attacks

Active information acquisition is a control input optimization problem over a finite-length time horizon: it aims to jointly optimize the control inputs for a team of mobile robots so that the robots, acting as mobile sensors, maximize the acquired information about a target process. Evidently, the optimization must account for the robot dynamics, target process, sensor model, robots' communication network, and information acquisition objective function.

1) Robot Dynamics: We assume noise-less, nonlinear robot dynamics, adopting the framework introduced in [6]

$$x_{i,t} = f_i(x_{i,t-1},\, u_{i,t-1}), \quad i \in \mathcal{V}, \quad t = 1,2,\ldots \tag{1}$$

where $\mathcal{V}$ denotes the set of available robots, $x_{i,t} \in \mathbb{R}^{n_{x_{i,t}}}$ denotes the state of robot i at time t,[1] and $u_{i,t} \in \mathcal{U}_{i,t}$ denotes the control input to robot i; $\mathcal{U}_{i,t}$ denotes the *finite* set of admissible control inputs to the robot.

2) Target Process: We assume any target process

$$y_t = g(y_{t-1}) + w_t, \quad t = 1, 2, \dots \tag{2}$$

where y_t denotes the target's state at time t, and w_t denotes Gaussian process noise; we consider $w_t \sim \mathsf{N}(\mu_{w_t}, \Sigma_{w_t})$.

3) Sensor Model: We assume measurements of the form

$$z_{i,t} = h_i(x_{i,t},\ y_t) + v_{i,t}(x_{i,t}), \quad t = 1, 2, \dots \tag{3}$$

where $z_{i,t}$ denotes the measurement by robot i at time t, and $v_{i,t}$ denotes measurement noise; we consider $v_{i,t} \sim \mathsf{N}(\mu_{v_{i,t}}((x_{i,t}), \Sigma_{v_{i,t}}((x_{i,t}))$. Both the noise and sensor function h_i depend on $x_{i,t}$, as it naturally is the case for, *e.g.*, bearing and range measurements (cf., Section VI-A).

4) Communication Network Among Robots: We assume centralized communication, *i.e.*, all robots can communicate with each other at any time.

5) Information Acquisition Objective Function: The information acquisition objective function captures the acquired information about the target process, as collected by the robots during the task via their measurements. In this article, in particular, we consider objective functions of the additive form

$$J_{\mathcal{V},\,1:T_{TASK}} \triangleq \sum_{t=1}^{T_{TASK}} J(y_t \mid u_{\mathcal{V},1:t},\ z_{\mathcal{V},1:t}) \tag{4}$$

where T_{TASK} denotes the duration of the information acquisition task, and $J(y_t \mid u_{\mathcal{V},1:t},\ z_{\mathcal{V},1:t})$ is an information metric, such as the *conditional entropy* [6] (also, cf., Section VI-A) or the *mutual information* [7] (also, cf., Section VI-B), where we make explicit only the metric's dependence on $u_{\mathcal{V},1:t}$ and $z_{\mathcal{V},1:t}$ (and we make implicit the metric's dependence on the initial conditions y_0 and $x_{i,0}$, and on the noise parameters, *i.e.*, the means and covariances of w_t and v_t for $t = 1, \dots, T_{TASK}$).

Problem ((Attack-Free) Active Information Acquisition): At time $t = 0$, find control inputs $u_{\mathcal{V},\,1:T_{TASK}}$ by solving the optimization problem

$$\max_{\substack{u_{\mathcal{V},t} \in \mathcal{U}_{\mathcal{V},t} \\ t = 1 : T_{TASK}}} \quad J_{\mathcal{V},\,1:T_{TASK}}. \tag{5}$$

Equation (5) captures a control input optimization problem where across a task-length horizon, the control inputs of all robots are jointly optimized to maximize $J_{\mathcal{V},\,1:T_{TASK}}$.

Solving (5) can be challenging, mainly due to the nonlinearity of (1)–(3), the duration of the task T_{TASK}, which acts as a look-ahead planning horizon (the longer the planning horizon is, the heavier (5) is in computing an optimal solution), and that at $t = 0$, *no* measurements have been realized yet.

To overcome the aforementioned challenges, *online* solutions to (5) have been proposed [6], similar to the *receding horizon control* solution—also known as *model predictive control*—for the finite-horizon optimal control problem [32, Ch. 12]. Specifically, per the receding horizon approach, one aims to solve (5) sequentially in time, by solving at each $t = 1, \dots, T_{TASK}$ an easier version of (5), but of the same form as (5), where the look-ahead horizon T_{TASK} is replaced by a shorter T_{PLAN} ($T_{PLAN} \leq T_{TASK}$), and (1)–(3) are replaced by their linearizations given the current $x_{\mathcal{V},t}$ and current estimate of y_t.

[1] $x_{i,t}$ in (1) belongs to an appropriate state space, such as $SE(2)$ or $SE(3)$, depending on the problem instance.

B. Active Information Acquisition in the Presence of Attacks

Equation (5) may suffer, however, from an additional challenge: the presence of attacks against the robots, which, if left unaccounted, can compromise the effectiveness of any robot plans per (5). In this article, in particular, we consider the presence of the following type of attacks.

1) Attack Model: At each t, an attack $\mathcal{A}_t$ can *remove* at most α robots from the information acquisition task ($\mathcal{A}_t \subseteq \mathcal{V}$ and $|\mathcal{A}_t| \leq \alpha$), in the sense that any removed robot i ($i \in \mathcal{A}_t$) cannot acquire any measurement $z_{i,t}$. In selecting the attack, the attacker has perfect knowledge of the state of the system. The attacker can select the worst-case attack (cf., Problem 1). Nevertheless, the attacker cannot necessarily prevent the robots from moving according to their preplanned path, nor can cause any communication loss among the robots.

Problem 1 (P-RAIN): At time $t = 0$, find control inputs $u_{\mathcal{V},\,1:T_{TASK}}$ by solving the optimization problem

$$\max_{\substack{u_{\mathcal{V},t} \in \mathcal{U}_{\mathcal{V},t} \\ t = 1 : T_{TASK}}} \quad \min_{\substack{\mathcal{A}_t \subseteq \mathcal{V},\, |\mathcal{A}_t| \leq \alpha \\ t = 1 : T_{TASK}}} \quad J_{\mathcal{V}\setminus\mathcal{A}_t,\,1:T_{TASK}}. \tag{P-RAIN}$$

(P-RAIN) goes beyond (5) by accounting for the attacks $\mathcal{A}_t$ ($t = 1, \dots, T_{TASK}$). This is expressed in (P-RAIN) with the minimization step, which aims to prepare an optimal solution $u_{\mathcal{V},t}$ against any worst-case attack that may happen at t.

Remark 1 (Need for (P-RAIN)): Reformulating (attack-free) (5) as in (P-RAIN) may seem unnecessary, since we consider that the attacker cannot cause any communication loss among the nonattacked robots (cf., the *attack model* defined earlier): indeed, if the nonattacked robots can *instantaneously observe the attacks at each* t, and *instantaneously replan at the same moment* t, then (P-RAIN) is unnecessary. However, replanning instantaneously in practice is impossible, due to computationally induced and algorithmic delays [6], as well as due to delays induced by the temporal discretization of the robot and target dynamics. Thus, for the duration where replanning is impossible, the plan needs to account for the worst-case attacks.

III. Receding Horizon Approximation: RAIN Algorithm

In solving (P-RAIN), one has to overcome not only the challenges involved in (5) (cf., Section II) but also the additional challenge of the worst-case attacks $\mathcal{A}_t$ (which are unknown *a priori*). We develop an online approximation procedure for (P-RAIN), summarized in Algorithm 1.

A. Intuitive Description

RAIN proposes a receding horizon solution to (P-RAIN), which enables online reaction to the history of attacks and, thus, is resilient, by executing the following steps.

1) *Initialization* (line 1): At $t = 0$, the acquisition task has not started and no attacks are assumed possible ($\mathcal{A}_0 = \emptyset$).
2) *Receding Horizon Planning* (lines 2–17). At each $t = 1, \ldots, T_{TASK}$, RAIN executes the receding horizon steps, which are as follows.

a) RTP (lines 1–1)*:* Given the current estimate $\hat{y}_t$ of the target process, all robots jointly optimize their control inputs by solving Problem 2, presented next, which is of the same form as (P-RAIN) but where (i) the look-ahead horizon T_{TASK} is replaced by a shorter T_{PLAN} ($T_{PLAN} \leq T_{TASK}$), and (ii) the attack is considered fixed over the look-ahead horizon.

Problem 2 (RTP): At time t, find attack-robust control inputs $u_{\mathcal{V},\, t+1:t+T_{PLAN}}$ by solving the optimization problem

$$\max_{\substack{u_{\mathcal{V},t'} \in \mathcal{U}_{\mathcal{V},t'} \\ t' = t+1 \,:\, t+T_{PLAN}}} \quad \min_{\mathcal{A} \subseteq \mathcal{V},\, |\mathcal{A}| \leq \alpha} \quad J_{\mathcal{V}\setminus\mathcal{A},\, t+1:t+T_{PLAN}}. \tag{P-RTP}$$

Both aforementioned (i) and (ii) intend to make (P-RTP) computationally tractable, so (P-RTP) can be solved in real time for the purposes of receding horizon planning. Particularly, we assume that the algorithm we propose for (P-RTP), RTP, is called in RAIN every T_{REPLAN} steps.[2]

Remark 2 (Role of T_{REPLAN}): T_{REPLAN} is chosen so that a receding horizon plan can always be generated in the duration it takes to compute a solution to (P-RTP) via RTP, *e.g.*, if one timestep—real-time interval from any t to $t+1$—has duration 0.5s, and solving (P-RTP) via RTP requires 2s, then $T_{REPLAN} = 4$ steps. Generally, $T_{REPLAN} \geq 1$ steps. Factors that influence the required time to solve (P-RTP) include the size of the robot team, the length of the planning horizon T_{PLAN} [3], the need for linearization of (1)–(3), and the number of possible attacks α—evidently, the latter factor is unique to (P-RAIN), in comparison to the attack-free (5).

b) *Control Execution* (lines 1–1)*:* Each robot i use their computed $u_{i,t}$ to make the next step in the environment (in the meantime, the real time changes from t to $t+1$ by the completion of the step).

c) *Attack Observation* (line 11)*:* RAIN observes the current attack, which affects the robots while they execute $u_{i,t}$.

d) *Measurement Collection* (lines 12–14)*:* The measurements from all nonattacked robots are collected.

e) *Estimate Update* (line 15)*:* Given all received measurements up to the current time, the estimate of y_t is updated.

f) *Time Update* (line 16)*:* RAIN updates the time counter to match it with the real time.

[2]RTP's pseudocode is presented in Algorithm 2, and described in more detail in Section IV. We quantify RTP's performance guarantees in Section V.

Algorithm 1: Resilient Active Information acquisitioN (RAIN).

Input: RAIN receives the inputs:
- *Offline:* Duration T_{TASK} of information acquisition task; look-ahead horizon T_{PLAN} for planning trajectories ($T_{PLAN} \leq T_{TASK}$); replanning rate T_{REPLAN} ($T_{REPLAN} \leq T_{PLAN}$); model dynamics f_i of each robot i's state $x_{i,t}$, including initial condition $x_{i,0}$ ($i \in \mathcal{V}$); sensing model h_i of each robot i's sensors, including $\mu_{v_{i,t}}$ and $\Sigma_{v_{i,t}}$; model dynamics g of target process, including initial condition y_0, and μ_{w_t} and covariance Σ_{w_t}; objective function J; number of attacks α.
- *Online:* At each $t = 1, \ldots, T_{TASK}$, observed (i) attack $\mathcal{A}_t$ (*i.e.*, robot removal $\mathcal{A}_t \subseteq \mathcal{V}$), and (ii) measurements $z_{i,t}$ from each non-attacked robot $i \in \mathcal{V} \setminus \mathcal{A}_t$.

Output: At each $t = 1, \ldots, T_{TASK}$, estimate $\hat{y}_t$ of y_t.

1: // Initialize //
2: $t = 0$; $\hat{y}_t = y_0$; $\mathcal{A}_t = \emptyset$; $z_t = \emptyset$;
3: // Execute resilient active information acquisition task //
4: **while** $t < T_{TASK}$ **do**
5: //(Re)plan robust trajectories for all robots //
6: **if** $t \mod T_{REPLAN} = 0$ **then**
7: $\mathcal{I}_t = \{t, \{f_i, x_{i,t}, h_i, \mu_{w_t}, \Sigma_{w_t}, \mu_{v_{i,t}}, \Sigma_{v_{i,t}}\}_{i\in\mathcal{V}}, g, \hat{y}_t,$
8: $z_t, T_{PLAN}, \alpha\}$; // Denote by $\mathcal{I}_t$ the information needed by the RTP algorithm, called in the next line.
9: $u_{\mathcal{V},\, t+1:t+T_{PLAN}} = RTP(\mathcal{I}_{0:t})$; // Plan robust trajectories for all robots with look-ahead planning horizon T_{PLAN}.
10: **end if**
11: // Execute current step of trajectory computed by RTP//
12: **for all** $i \in \mathcal{V}$ **do**
13: $x_{i,t+1} = f_i(x_{i,t}, u_{i,t})$;
14: **end for**
15: Observe $\mathcal{A}_{t+1}$; // Determined by environment/attacker.
16: // Integrate measurements from non-attacked robots //
17: **for all** $i \in \mathcal{V} \setminus \mathcal{A}_{t+1}$ **do**
18: Receive measurement $z_{i,t+1}$; // Only measurements from non-attacked robots are received.
19: **end for**
20: **update** Estimate $\hat{y}_{t+1}$ of y_{t+1} given $z_{1\,:\,t+1}$; // $z_{1\,:\,t}$ collects all available measurements up to the time t, *i.e.*, $z_{1\,:\,t} \triangleq \{z_{i,\tau} : i \in \mathcal{V} \setminus \mathcal{A}_\tau, \tau = 1, \ldots, t\}$.
21: $t = t + 1$; // Time update.
22: **end while**

IV. RTP Algorithm

We present RTP, which is used as a subroutine in RAIN, in a receding horizon fashion (cf., Section III). RTP's pseudocode is presented in Algorithm 2. RTP's performance is quantified in Section V. We next give an intuitive description of RTP.

A. Intuitive Description

RTP's goal is to maximize (P-RTP)'s objective function $J_{\mathcal{V}\setminus\mathcal{A},\, t+1:t+T_{PLAN}}$, despite a worst-case attack $\mathcal{A}$ that removes up to α robots from $\mathcal{V}$. In this context, RTP aims to fulfill (P-RTP)'s goal with a two-step process, where RTP partitions robots into two sets (the set of robots $\mathcal{L}$, and the set of robots $\mathcal{V} \setminus \mathcal{L}$; cf., RTP's lines 2–2), and then, RTP appropriately selects

Algorithm 2: Robust Trajectory Planning (RTP).

Input: Look-ahead horizon T_{PLAN} for planning trajectories; current time t; set of robots $\mathcal{V}$; model dynamics f_i of each robot i's state, including current state $x_{i,t}$; sensing model h_i of each robot i's sensors, including $\mu_{v_{i,t}}$ and $\Sigma_{v_{i,t}}$; model dynamics g of target process y_t, including current estimate $\hat{y}_t$, and μ_{w_t} and covariance Σ_{w_t}; objective function J; measurement history $z_{1:t}$; number of attacks α.

Output: Control inputs $u_{i,t'}$, for all robots $i \in \mathcal{V}$ and all times $t' = t+1, \ldots, t+T_{PLAN}$.

// Step 1: Generate *bait* robot set (to approximate a worst-case attack assumed constant $\forall t' \in [t+1, t+T_{PLAN}]$) //

1: **for all** $i \in \mathcal{V}$ **do** // Compute the value of (P-RTP) assuming (i) only robot i exists and (ii) no attacks will happen.

2: $J^{\star}_{\{i\},\, t,\, 0} \triangleq \max_{\substack{u_{i,t'} \in \mathcal{U}_{i,t'} \\ t' = t+1 \,:\, t+T_{PLAN}}} J_{\{i\},\, t+1:t+T_{PLAN}}$;

3: **end for**

4: Find a subset $\mathcal{L}$ of α robots such that $J^{\star}_{\{i\},\, t,\, 0} \geq J^{\star}_{\{j\},\, t,\, 0}$ for all $i \in \mathcal{L}$ and $j \in \mathcal{V} \setminus \mathcal{L}$; // $\mathcal{L}$ is the *bait* robot set ($\mathcal{L} \subseteq \mathcal{V}$ and $|\mathcal{L}| = \alpha$).

5: **for all** $i \in \mathcal{L}$ **do** // Assign to each robot $i \in \mathcal{L}$ the trajectory that achieves $J^{\star}_{\{i\},\, t,\, 0}$.

6: $u_{i,\, t+1:t+T_{PLAN}} = \arg \max_{\substack{u_{i,t'} \in \mathcal{U}_{i,t'} \\ t'=t+1 \,:\, t+T_{PLAN}}} J_{\{i\},t+1:t+T_{PLAN}}$;

7: **end for**

// Step 2: Remaining robots, $\mathcal{V} \setminus \mathcal{L}$, plan assuming (i) only robots in $\mathcal{V} \setminus \mathcal{L}$ exists and (ii) no attacks will happen //

8: $u_{\mathcal{V}\setminus\mathcal{L},t+1:t+T_{PLAN}} = \arg \max_{\substack{u_{\mathcal{V}\setminus\mathcal{L},t'} \in \mathcal{U}_{\mathcal{V}\setminus\mathcal{L},t'} \\ t'=t+1 \,:\, t+T_{PLAN}}} J_{\mathcal{V}\setminus\mathcal{L},t+1:t+T_{PLAN}}$.

the robots' control inputs in each of the two sets (cf., RTP's lines 2–2). In particular, RTP picks $\mathcal{L}$ aiming to guess the worst-case removal of α robots from $\mathcal{V}$, *i.e.*, to guess the optimal solution to the minimization step in (P-RTP). Thus, intuitively, $\mathcal{L}$ is aimed to act as a "bait" to the attacker. Since guessing the optimal "bait" is, in general, intractable [33], RTP aims to approximate it by letting $\mathcal{L}$ be the set of α robots with the α largest marginal contributions to $J_{\cdot,\, t+1:t+T_{PLAN}}$ (RTP's lines 5–7). Then, RTP assumes that the robots in $\mathcal{L}$ are nonexistent, and plans the control inputs for the remaining robots (RTP's line 8).

V. Performance Guarantees of RTP

Performance guarantees are unknown for RAIN, and, correspondingly, (P-RAIN) is a mixed-integer, sequential control optimization problem, with limited *a priori* information on the measurements and attacks that are going to occur during the task-length, look-ahead time horizon). Nevertheless, in this section, we quantify RTP's performance, which is used by RAIN in a receding horizon fashion to approximate a solution to (P-RAIN) locally (cf., RAIN's lines 2–17), by picking sequentially in time control inputs given a shorter, computationally feasible look-ahead time horizon (cf., Section III), and the history of the so far observed measurements and attacks.

Particularly, in this section, we bound RTP's approximation performance and running time. We use properties of the objective function $J_{\mathcal{V}\setminus\mathcal{A},\, t+1:t+T_{PLAN}}$ in (P-RTP) as a function of the set of robots; namely, the following notions of curvature.

A. Curvature and Total Curvature

We present the notions of *curvature* and *total curvature* for set functions. We start with the notions of *modularity*, and of *nondecreasing* and *submodular* set functions.

Definition 1 (Modularity [24]): Consider a finite (discrete) set $\mathcal{V}$. A set function $h : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is modular if and only if $h(\mathcal{A}) = \sum_{v \in \mathcal{A}} h(v)$, for any $\mathcal{A} \subseteq \mathcal{V}$.

Hence, if h is modular, then $\mathcal{V}$'s elements complement each other through h. Specifically, Definition 1 implies $h(\{v\} \cup \mathcal{A}) - h(\mathcal{A}) = h(v)$, for any $\mathcal{A} \subseteq \mathcal{V}$ and $v \in \mathcal{V} \setminus \mathcal{A}$.

Definition 2 (Nondecreasing Set Function [24]): Consider a finite (discrete) set $\mathcal{V}$. $h : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is *nondecreasing* if $h(\mathcal{B}) \geq h(\mathcal{A})$ for all $\mathcal{A} \subseteq \mathcal{B}$.

Definition 3 (Submodularity [24, Proposition 2.1]): Consider a finite (discrete) set $\mathcal{V}$. $h : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is *submodular* if $h(\mathcal{A} \cup \{v\}) - h(\mathcal{A}) \geq h(\mathcal{B} \cup \{v\}) - h(\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B}$ and $v \in \mathcal{V}$.

Therefore, h is submodular if and only if the return $h(\mathcal{A} \cup \{v\}) - h(\mathcal{A})$ diminishes as $\mathcal{A}$ grows, for any v. If h is submodular, then $\mathcal{V}$'s elements substitute each other, in contrast to h being modular. Particularly, consider h to be nonnegative (without loss of generality): then, Definition 3 implies $h(\{v\} \cup \mathcal{A}) - h(\mathcal{A}) \leq h(v)$. Thereby, v's contribution to $h(\{v\} \cup \mathcal{A})$'s value is diminished in the presence of $\mathcal{A}$.

Definition 4 (Curvature [29]): Consider a finite (discrete) $\mathcal{V}$, and a nondecreasing submodular $h : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that $h(v) \neq 0$ for any $v \in \mathcal{V}$, without loss of generality. Then, h's *curvature* is defined as

$$\kappa_h \triangleq 1 - \min_{v \in \mathcal{V}} \frac{h(\mathcal{V}) - h(\mathcal{V} \setminus \{v\})}{h(v)}. \tag{6}$$

Definition 4 implies $\kappa_h \in [0, 1]$. If $\kappa_h = 0$, then $h(\mathcal{V}) - h(\mathcal{V} \setminus \{v\}) = h(v)$, for all $v \in \mathcal{V}$, *i.e.*, h is modular. Instead, if $\kappa_h = 1$, then there exist $v \in \mathcal{V}$ such that $h(\mathcal{V}) = h(\mathcal{V} \setminus \{v\})$, that is, v has no contribution to $h(\mathcal{V})$ in the presence of $\mathcal{V} \setminus \{v\}$. Overall, κ_h represents a measure of how much $\mathcal{V}$'s elements complement (and substitute) each other.

Definition 5 (Total Curvature [30, Sec. 8]): Consider a finite (discrete) set $\mathcal{V}$ and a monotone $h : 2^{\mathcal{V}} \mapsto \mathbb{R}$. Then, h's *total curvature* is quantified as

$$c_h \triangleq 1 - \min_{v \in \mathcal{V}} \min_{\mathcal{A}, \mathcal{B} \subseteq \mathcal{V} \setminus \{v\}} \frac{h(\{v\} \cup \mathcal{A}) - h(\mathcal{A})}{h(\{v\} \cup \mathcal{B}) - h(\mathcal{B})}. \tag{7}$$

Definition 5 implies $c_h \in [0, 1]$, similarly to Definition 5 for κ_h. When h is submodular, then $c_h = \kappa_h$. Generally, if $c_h = 0$, then h is modular, whereas if $c_h = 1$, then (7) implies Definition 5's assumption that h is nondecreasing.

B. Performance Analysis for RTP

We quantify suboptimality bounds on RTP's approximation performance and upper bounds on the running time RTP requires. We use the following notation.

1) $J^\star_{\mathcal{V},\,t,\,\alpha}$ is the optimal value of (P-RTP)

$$J^\star_{\mathcal{V},t,\alpha} \triangleq \max_{\substack{u_{\mathcal{V},t'}\in\mathcal{U}_{\mathcal{V},t'}\\ t'=t+1\,:\,t+T_{PLAN}}} \min_{\mathcal{A}\subseteq\mathcal{V},\,|\mathcal{A}|\,\leq\,\alpha} J_{\mathcal{V}\setminus\mathcal{A},\,t+1:t+T_{PLAN}}.$$

2) $\mathcal{A}^\star$ is an optimal removal of α robots from $\mathcal{V}$ per (P-RTP)

$$\mathcal{A}^\star \triangleq \arg\min_{\mathcal{A}\subseteq\mathcal{V},\,|\mathcal{A}|\leq\alpha} J_{\mathcal{V}\setminus\mathcal{A},\,t+1:t+T_{PLAN}}.$$

We also use the following definitions.

Definition 6 (Normalized Set Function [24]): Consider a discrete set $\mathcal{V}$. $h:2^{\mathcal{V}}\mapsto\mathbb{R}$ is *normalized* if $h(\emptyset)=0$.

Definition 7 (Nonnegativeness [24]): Consider a discrete set $\mathcal{V}$. $h:2^{\mathcal{V}}\mapsto\mathbb{R}$ is *nonnegative* if $h(\mathcal{A})\geq 0$ for all $\mathcal{A}$.

Theorem 1 (Performance of RTP *):* Consider an instance of (P-RTP). Assume the robots in $\mathcal{V}$ can solve optimally the (attack-free) information acquisition problem in (5).

1) *Approximation performance:* RTP returns control inputs $u_{\mathcal{V},\,1:t+T_{PLAN}}$ such that if $J_{\cdot,t+1:t+T_{PLAN}}:2^{\mathcal{V}}\mapsto\mathbb{R}$ is nondecreasing and, without loss of generality, normalized and nonnegative, then

$$\frac{J_{\mathcal{V}\setminus\mathcal{A}^\star,\,t+1:t+T_{PLAN}}}{J^\star_{\mathcal{V},\,t,\,\alpha}} \geq (1-c_{J_{\cdot,t+1:t+T_{PLAN}}})^2. \quad (8)$$

If, in addition, $J_{\cdot,t+1:t+T_{PLAN}}$ is submodular, then

$$\frac{J_{\mathcal{V}\setminus\mathcal{A}^\star,\,t+1:t+T_{PLAN}}}{J^\star_{\mathcal{V},\,t,\,\alpha}} \geq \max\left(1-\kappa_{J_{\cdot,t+1:t+T_{PLAN}}},\frac{1}{1+\alpha}\right). \quad (9)$$

2) *Running time:* If ρ upper bounds the running time for solving the (attack-free) information acquisition problem in (5), then RTP terminates in $O(|\mathcal{V}|\rho)$ time.

Theorem 1's bounds in (8) and (9) compare RTP's selection $u_{\mathcal{V},\,1:t+T_{PLAN}}$ against an optimal selection of control inputs that achieves the optimal value $J^\star_{\mathcal{V},\,t,\,\alpha}$ for (P-RTP). Particularly, (8) and (9) imply that for nondecreasing and nondecreasing and submodular functions $J_{\cdot,t+1:t+T_{PLAN}}$, RTP guarantees a value for (P-RTP), which can be close to the optimal. For example, (9)'s lower bound $1/(1+\alpha)$ is nonzero for any finite number of robots $|\mathcal{V}|$, and, notably, it equals 1 in the attack-free case (RTP is exact for $\alpha=0$, per Theorem 1's assumptions). More broadly, when $\kappa_{J_{\cdot,t+1:t+T_{PLAN}}}<1$ or $c_{J_{\cdot,t+1:t+T_{PLAN}}}<1$, RTP's selection $u_{\mathcal{V},\,1:t+T_{PLAN}}$ is close to the optimal, in the sense that Theorem 1's bounds are nonzero. Functions with $\kappa_{J_{\cdot,t+1:t+T_{PLAN}}}<1$ include the $\log\det$ of positive-definite matrices [34]; objective functions of this form are the conditional entropy and mutual information when used for batch-state estimation of stochastic processes [35]. Functions with $c_{J_{\cdot,t+1:t+T_{PLAN}}}<1$ include the average minimum square error (mean of the trace of a Kalman filter's error covariance across a finite time horizon) [36].

Theorem 1's curvature-dependent bounds in (8) and (9) also make a first step toward separating the classes of nondecreasing and nondecreasing and submodular functions into functions for which (P-RTP) can be approximated well, and functions for which it cannot. Indeed, when either $\kappa_{J_{\cdot,t+1:t+T_{PLAN}}}$ or $c_{J_{\cdot,t+1:t+T_{PLAN}}}$ tend to zero, RTP becomes exact. For example, (8)'s term $1-c_{J_{\cdot,t+1:t+T_{PLAN}}}$ increases as $c_{J_{\cdot,t+1:t+T_{PLAN}}}$ decreases, and its limit is equal to 1 for $c_{J_{\cdot,t+1:t+T_{PLAN}}}\to 0$. Notably, however, the tightness of Theorem 1's bounds is an open problem. For example, although for the attack-free problem in (5), a bound $O(1-c_{J_{\cdot,t+1:t+T_{PLAN}}})$ is known to be optimal (the tightest possible in polynomial time and for a worst-case $J_{\cdot,t+1:t+T_{PLAN}}$) [30, Th. 8.6], the optimality of (8) is an open problem.

Overall, Theorem 1 quantifies RTP's approximation performance when the robots in $\mathcal{V}$ solve optimally the (attack-free) information acquisition problems in RTP's lines 22, and 8. Among those, however, the problems in lines 6 and 8 are computationally challenging, being multirobot coordination problems; only approximation algorithms are known for their solution. Such an approximation algorithm is the recently proposed *coordinate descent* [6, Sec. IV]. Coordinate descent has the advantages of having a provably near-optimal approximation performance. Therefore, we next quantify RTP's performance when the robots in $\mathcal{V}$ solve the problems in RTP's line 6, and line 8 using coordinate descent.[3]

Proposition 1 (Approximation Performance of: RTP *via Coordinate Descent):* Consider an instance of (P-RTP). Assume the robots in $\mathcal{V}$ solve the (attack-free) information acquisition problem in (5) suboptimally in the case of multiple robots *($|\mathcal{V}|\geq 2$)* via coordinate descent [6, Sec. IV], and optimally in the case of a single robot *($|\mathcal{V}|=1$)*. Then, the following hold.

1) *Approximation performance:* RTP returns control inputs $u_{\mathcal{V},\,1:t+T_{PLAN}}$ such that if $J_{\cdot,t+1:t+T_{PLAN}}:2^{\mathcal{V}}\mapsto\mathbb{R}$ is nondecreasing and, without loss of generality, normalized and nonnegative, then

$$\frac{J_{\mathcal{V}\setminus\mathcal{A}^\star,\,t+1:t+T_{PLAN}}}{J^\star_{\mathcal{V},\,t,\,\alpha}} \geq \frac{1}{2}(1-c_{J_{\cdot,t+1:t+T_{PLAN}}})^3. \quad (10)$$

If $J_{\cdot,t+1:t+T_{PLAN}}$ is also submodular, then

$$\frac{J_{\mathcal{V}\setminus\mathcal{A}^\star,\,t+1:t+T_{PLAN}}}{J^\star_{\mathcal{V},\,t,\,\alpha}} \geq \frac{1}{2}\max\left(1-\kappa_{J_{\cdot,t+1:t+T_{PLAN}}},\frac{1}{1+\alpha}\right). \quad (11)$$

2) *Running time:* If ρ_{CD} upper bounds the running time for solving the information acquisition problem in (5) via coordinate descent, then RTP terminates in $O(\rho_{CD})$ time.

Proposition 1's suboptimality bounds are discounted versions of Theorem 1's bounds: (10) is the discounted (8) by the factor $(1-c_{J_{\cdot,t+1:t+T_{PLAN}}})/2$; and (11) is the discounted (9) by the factor $1/2$. The source of the discounting factors is the requirement in Proposition 1 that the robots in $\mathcal{V}$ can solve only suboptimally (via coordinate descent) the information acquisition problem in (5) (and, in effect, the problems in RTP's lines 6 and 8). In more detail, in Lemma 5, located in Appendix B, we prove

[3]We refer to Appendix B for a description of coordinate descent.

that for nondecreasing objective functions, coordinate descent guarantees the suboptimality bound $(1 - c_{J_{\cdot,t+1:t+T_{PLAN}}})/2$ for (5) [which is the discounting factor to (8), resulting in (10)], whereas for nondecreasing and submodular functions, coordinate descent is known to guarantee the suboptimality bound $1/2$ for (5) [which is the discounting factor to (9), resulting in (11)] [6].

Proposition 1 also implies that if the robots in $\mathcal{V}$ use coordinate descent to solve the (attack-free) information acquisition problems in RTP's lines 6 and 8, then RTP has the same order of running time as coordinate descent. The proof of Proposition 1 is found in Appendix D.

VI. Applications and Experiments

We present RAIN's performance in applications. We present three applications of RAIN with teams of robots: *multitarget tracking* (see Section VI-A), *occupancy grid mapping* (see Section VI-B), and *persistent surveillance* (see Section VI-C). We show RAIN effectiveness, even as we vary key parameters in (P-RAIN), which is as follows.

1) The *number of attacks* α, among the permissible values $\{0, 1, \ldots, |\mathcal{V}|\}$, to test RAIN's performance to both small and high attack numbers (see Section VI-A).
2) The *attack model*, beyond the worst-case model prescribed by (P-RAIN)'s problem formulation, to test RAIN's sensitivity against *non*worst-case failures; particularly, random failures (see Section VI-B).[4]
3) The *replanning rate* T_{REPLAN}, among the permissible values $\{1, 2, \ldots, T_{PLAN}\}$, to test RAIN's performance even when the replanning rate is high (see Section VI-C).[5]

Common Experimental Setup Across Applications

1) *Approximation Algorithm for (Attack-Free) Information Acquisition Problem in* (5)*(and in effect for the problems in* RTP*'s lines 2, 6, and 8):* In the multirobot case (pertained to RTP's lines 6, and 8), the algorithm used for approximating a solution to (5) is the *coordinate descent* [6] (also, cf., Appendix B). Evidently, coordinate descent does not account for the possibility of attacks, and for this reason, we also use it as a baseline to compare RAIN with. In the single robot case (pertained to RTP's line 2), (5) reduces to a single-robot motion planning problem, and for its solution, we use reduced value iteration (ARVI algorithm [4]), except for the application of *occupancy grid mapping* (see Section VI-B) where we use forward value iteration [3].
2) *Worst-Case Attack Approximation:* Computing the worst-case attack requires brute-force, since the minimization step in (P-RAIN) is NP-hard [37]. The consequence is that solving for the worst-case attack requires solving an exponential number of instances of the information acquisition problem in (5), prohibiting real-time navigation performance by the robots, even for small teams of robots ($|\mathcal{V}| \geq 5$). In particular, the running time required to solve (5), even via coordinate descent, can be exponential in the number of robots and task length horizon, namely, $\mathcal{O}(|\mathcal{U}|^{|\mathcal{V}|T_{TASK}})$ [6] ($\mathcal{U}$ denotes the set of admissible control inputs to each of the robots in $\mathcal{V}$, assumed the same across all robots). Hence, we approximate the worst-case attacks by solving the minimization step in (P-RAIN) via a greedy algorithm [24].
3) *Computational Platform:* Experiments are implemented in C++, and run on an Intel Core i7 CPU laptop.

A. Resilient Multitarget Tracking

In *resilient multitarget tracking*, a team of mobile robots is tasked to track the locations of multiple moving targets, even in the presence of a number of attacks against the robots. For the purpose of assessing RAIN's effectiveness against various number of attacks, we will vary the number of attacks across scenarios, where we will also possibly vary the number of robots and targets. In more detail, the experimental setup and simulated scenarios are described below.

1) Experimental Setup: We specify the used *robot dynamics*, *target process*, *sensor model*, and *information acquisition objective function as follows.*

a) Robot dynamics: Each robot i has unicycle dynamics in $\mathrm{SE}(2)$, discretized with a sampling period τ, such that

$$\begin{pmatrix} x_{i,t+1}^{(1)} \\ x_{i,t+1}^{(2)} \\ \theta_{i,t+1} \end{pmatrix} = \begin{pmatrix} x_{i,t}^{(1)} \\ x_{i,t}^{(2)} \\ \theta_{i,t} \end{pmatrix} + \begin{pmatrix} \nu_i \operatorname{sinc}(\frac{\omega_i \tau}{2}) \cos(\theta_{i,t} + \frac{\omega_i \tau}{2}) \\ \nu_i \operatorname{sinc}(\frac{\omega_i \tau}{2}) \sin(\theta_{i,t} + \frac{\omega_i \tau}{2}) \\ \tau \omega_i \end{pmatrix} \tag{12}$$

where (ν_i, ω_i) is the control input (linear and angular velocity).

b) Target dynamics: The targets move according to double integrator dynamics, which are assumed corrupted with additive Gaussian noise. Specifically, if M denotes the number of targets, then $y_t = [y_{t,1}^\top, \ldots, y_{t,M}^\top]^\top$, where $y_{t,m}$ is the planar coordinates and velocities of target m, and

$$\begin{aligned} y_{t+1,m} &= \begin{bmatrix} I_2 & \tau I_2 \\ 0 & I_2 \end{bmatrix} y_{t,m} + w_t, w_t \\ &\sim \mathsf{N}\left(0, q \begin{bmatrix} \tau^3/3 I_2 & \tau^2/2 I_2 \\ \tau^2/2 I_2 & \tau I_2 \end{bmatrix}\right) \end{aligned}$$

with q being a noise diffusion parameter.

c) Sensor model: The robots' sensor model consists of a range and bearing for each target $m = 1, \ldots, M$

$$z_{t,m} = h(x_t, y_{t,m}) + v_t, \quad v_t \sim \mathsf{N}(0, V(x_t, y_{t,m}))$$

$$h(x, y_m) = \begin{bmatrix} r(x, y_m) \\ \alpha(x, y_m) \end{bmatrix} \triangleq \begin{bmatrix} \sqrt{(y^1 - x^1)^2 + (y^2 - x^2)^2} \\ \tan^{-1}((y^2 - x^2)(y^1 - x^1)) - \theta \end{bmatrix}.$$

[4]For random failures, one would expect RAIN's performance to be the same, or improve, since RAIN is designed to withstand the worst case.

[5]When the replanning rate tends, ideally, to infinity, in that the robots can instantaneously observe all attacks and replan (which, however, is in practice impossible due to algorithmic, computational, and communication delays), then it is expected that RAIN's advantage over a nonresilient algorithm with the same replanning rate, such as coordinate descent, would diminish.

Since the sensor model is nonlinear, we linearize it around the currently predicted target trajectory. Particularly, given

$$\nabla_y h(x, y_m) = \frac{1}{r(x, y_m)} \times \begin{bmatrix} (y^1 - x^1) & (y^2 - x^2) & 0_{1x2} \\ -\sin(\theta + \alpha(x, y_m)) & \cos(\theta + \alpha(x, y_m)) & 0_{1x2} \end{bmatrix}$$

the observation model for the joint target state can be expressed as a block diagonal matrix containing the linearized observation models for each target along the diagonal

$$H \triangleq \operatorname{diag}\left(\nabla_{y_1} h(x, y_1), \ldots, \nabla_{y_M} h(x, y_M)\right).$$

The sensor noise covariance grows linearly in range and in bearing, up to σ_r^2 and σ_b^2, where σ_r and σ_b are the standard deviation of the range and the bearing noise, respectively. The model here also includes a limited range and field of view, denoted by the parameters r_{sense} and ψ, respectively.

d) Information acquisition objective function: For the information objective function we use, the time averaged log determinant of the covariance matrix, which is equivalent to conditional entropy for Gaussian variables [3]. This objective function is nondecreasing, yet not necessarily submodular [38]. Overall, we solve an instance of (P-RAIN) per the aforementioned setup and the objective function

$$J_{\mathcal{V},\,1:T_{TASK}} \triangleq \frac{1}{T_{TASK}} \sum_{t=1}^{T_{TASK}} \log\det(\Sigma_{\mathcal{V},\,t})$$

where $\Sigma_{\mathcal{V},\,t}$ is the Kalman filtering error covariance at t, given the measurements collected up to t by the robots in $\mathcal{V}$ [3].[6]

2) Simulated Scenarios: We consider multiple scenarios of the experimental setup introduced earlier: across scenarios, we vary the number of robots n, the number of targets M, and the number of attacks, α (cf., first column of Table I). Additionally, the robots and targets are restricted to move inside a $64 \times 64\text{m}^2$ environment (see Fig. 2). The admissible control input values to each robot are the $\mathcal{U} = \{1, 3\}\text{m/s} \times \{0, \pm 1, \pm 3\}\text{rad/s}$. At the beginning of each scenario, we fix the initial positions of both the robots and targets, and the robots are given a prior distribution of the targets before starting the simulation. The targets start with a zero velocity, and in the event that a target leaves the environment, its velocity is reflected to remain in bounds. Finally, across all simulations: $T_{PLAN} = T_{REPLAN} = 25$ steps, $\tau = 0.5\text{s}$, $r_{\text{sense}} = 10\text{m}$, $\psi = 94^\circ$, $\sigma_r = .15\text{m}$, $\sigma_b = 5^\circ$, and $q = .001$. We use $T_{TASK} = 500$.

3) Compared Techniques: We compare RAIN with *coordinate descent*. We consider two performance measures: the average entropy and average root-mean-square error (RMSE) per target, averaged over the robots in the team.

4) Results: The results, averaged across ten Monte Carlo runs, are depicted in Fig. 3 and Table I. In Fig. 3, RAIN's performance is observed to be superior both in terms of the average entropy and the RMSE. Particularly, as the number of attacks grows (cf., second rows of plots in Fig. 3), RAIN's benefits are accentuated in comparison to the nonresilient coordinate descent. Similarly, Table I demonstrates that RAIN achieves a lower mean RMSE than coordinate descent and, crucially, is highly effective in reducing the peak estimation error; in particular, RAIN achieves a performance that is 2 to 30 times better in comparison to the performance achieved by the nonresilient algorithm. We also observe that the impact of

[6]The multitarget tracking scenarios are dependent on a prior distribution of the target's initial conditions y_0 and $\Sigma_{0|0}$, assumed here known. Yet, if a prior distribution is unknown, then an exploration strategy can be incorporated to find the targets by placing exploration landmarks at the map frontiers [6].

TABLE I
RESILIENT MULTITARGET TRACKING RESULTS: PERFORMANCE COMPARISON OF RAIN WITH COORDINATE DESCENT (NOTED AS NonRes IN THE TABLE), FOR A VARIETY OF CONFIGURATIONS, WHERE n DENOTES THE NUMBER OF ROBOTS ($n = |\mathcal{V}|$), M DENOTES THE NUMBER OF TARGETS, AND α DENOTES THE NUMBER OF FAILURES

	Mean RMSE		Peak RMSE	
	NonRes	RAIN	NonRes	RAIN
$n = 5,\ M = 10$				
$\alpha = 1$	0.28	0.19	9.62	2.09
$\alpha = 2$	1.47	0.68	26.07	15.71
$\alpha = 4$	10.67	4.9	225.47	103.82
$n = 10,\ M = 5$				
$\alpha = 2$	0.35	0.14	57.65	1.87
$\alpha = 4$	0.39	0.28	6.66	3.17
$\alpha = 6$	2.07	0.65	93.27	15.63
$n = 10,\ M = 10$				
$\alpha = 2$	0.13	0.08	1.4	1.32
$\alpha = 4$	0.24	0.23	4.19	2.66
$\alpha = 6$	4.39	1.2	69.77	26.4

Note: Two performance metrics are used: mean RMSE and peak RMSE, both per target, and averaged over the robots in the team.

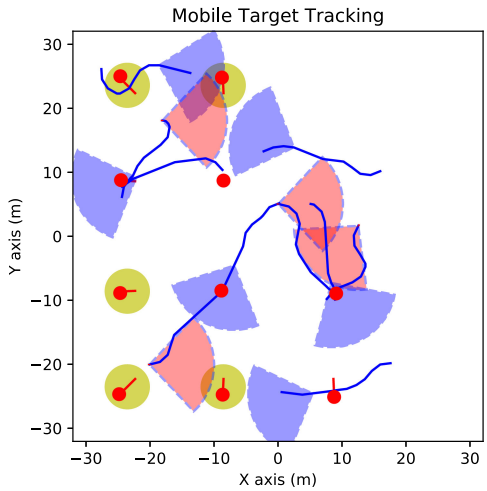


Fig. 2. *Resilient multitarget tracking scenario*. Ten robots are depicted tracking ten targets, whereas four of the robots are be attacked (causing their sensing capabilities to be, at least temporarily, disabled). The robots are depicted with their conic-shaped field of view, colored light blue for nonattacked robots and light red for attacked robots. The targets are depicted with red disks. Planned robot trajectories are shown as solid blue lines. Predicted target trajectories are shown as solid red lines. Each light-green ellipse represents the covariance of the target's location estimate.

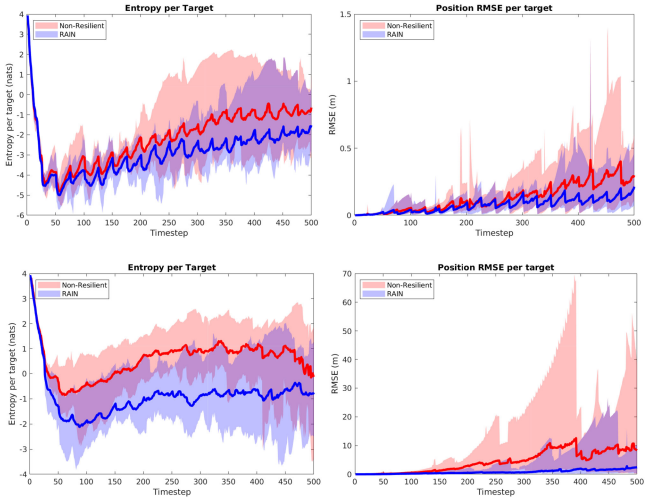


Fig. 3. *Resilient multitarget tracking results.* Performance comparison of RAIN with coordinate descent (noted as NonResilient in the plots), for two configurations and two performance metrics: top row considers ten robots, ten targets, and two attacks; bottom row considers the same number of robots and targets but six attacks. The left column depicts mean entropy per target, averaged over the robots; the right column depicts the position RMSE per target, also averaged over the robots.

Algorithm 2 is most prominent when the number of attacks is large relative to the size of the robot team.

B. Resilient Occupancy Grid Mapping

We show how (P-RAIN)'s framework for RAIN can be adapted to explore an environment when the map and objective are defined via occupancy grids. In this section, we also assess RAIN's sensitivity against nonworst-case attacks, in particular, random.

1) Experimental Setup: We specify the used *robot dynamics*, *target process*, *sensor model*, *information acquisition objective function*, and *algorithm for solving the optimization problem in* RTP*'s line 2*.

a) Robot dynamics: The robots' dynamics are as in the multitarget tracking application (see Section VI-A).

b) Target process: We define the target process y_t, which we will denote henceforth as M for consistency with common references on occupancy grid mapping [39], where we also drop the time subscript since the process y_t does not evolve in time. The occupancy grid M is a two-dimensional grid with n rows and m columns, discretized into cells $M = \{C_1, \ldots, C_n m\}$, which are binary variables that are either occupied or free, with some probability. Cell occupancy is assumed to be independent, so that the probability mass function can be factored as $\mathbb{P}(M = m) = \prod_{i=1}^{nm} \mathbb{P}(C_i = c_i)$, where $c_i \in \{0, 1\}$, and where $m \in \{0, 1\}^{nm}$ is a particular realization of the map.

c) Sensor model: We express the sensor model as a series of B beams, such that $z_t = [z_t^1, \ldots, z_t^B]$, where z_k^b is the random variable of the distance that a beam b travels to intersect an object in the environment. We next define the distribution for a single beam, determined by the true distance d to the first obstacle that the beam intersects

$$p(z_k^b = z|d) = \begin{cases} \mathcal{N}(z - 0, \sigma^2), & d < z_{\min} \\ \mathcal{N}(z - z_{\max}, \sigma^2), & d > z_{\max} \\ \mathcal{N}(z - d, \sigma^2), & \text{otherwise} \end{cases} \tag{13}$$

where $z_{\min}$ and $z_{\max}$ are the minimum and maximum sensing ranges, respectively.

d) Information acquisition objective function: The used information objective function is the Cauchy–Schwarz quadratic mutual information (CSQMI), which is shown in the literature to be computationally efficient, as well as sufficiently accurate for occupancy mapping [19]. We denote the CSQMI gathered at time t by $I_{CS}(m;\ z_{\mathcal{V},t})$, given the measurements collected at t by the robots in $\mathcal{V}$. Then

$$J_{\mathcal{V},1:T_{TASK}} \triangleq \frac{1}{T_{TASK}} \sum_{t=1}^{T_{TASK}} I_{CS}(m;\ z_{\mathcal{V},t}).$$

Remark 3 (Evaluation of CSQMI): Details on the evaluation of the CSQMI objective are beyond the scope of this article; we refer the reader to Charrow et al. [19]. We note that it relies on a ray-tracing operation for each beam, computed over the current occupancy grid map belief to determine which cells each beam from the LIDAR will observe when the sensor visits a given pose. CSQMI is approximated by assuming the information computed over single beams is additive, but not before pruning approximately independent beams. This removal of approximately independent beams encourages robots to explore areas where their beams will not overlap. In coordinate descent, once a set of prior robots has planned trajectories, future robots must check their beams to see if they are approximately independent from the fixed beams, before their individual beam contributions may be added to the joint CSQMI objective.

e) Algorithm for solving optimization problem in RTP*'s line 2:* The single-robot motion planning is performed via a full forward search over a short planning horizon $T_{PLAN} = 4$, since scaling beyond short horizons is challenging in occupancy mapping problems; for details, cf., [19]. We remark that the performance guarantees do not explicitly hold for the single-robot planner since the measurement model is highly nonlinear and the cost function depends on the realization of the measurements, so open-loop planning is not optimal as it is with the Gaussian case [3]. Nonetheless, approaches similar to what we adopt here have been successfully used for (attack-free) occupancy grid mapping [5], [7].

2) Simulated Scenarios: To evaluate the performance of the resilient occupancy mapping algorithm, we compare the results with different attack types. Namely, we consider an attack model where the attacks on robots may be uniformly random, rather than the worst-case attack assumption of the previous section and the algorithm itself. In the experiment, the robots choose trajectories composed of $T_{PLAN} = 4$ steps of duration $\tau = 1$ s, with motion primitives $\mathcal{U} = \{(v, \omega) : v \in \{0, 1, 2\}\text{m/s},\ \omega \in \{0, \pm 1, \pm 2\}\text{rad/s}\}$. The maximum sensor range r_{sense} is 1.5m, with a noise standard deviation of $\sigma = .01$m. The experiment considers a team of six robots, subject to two attack models,

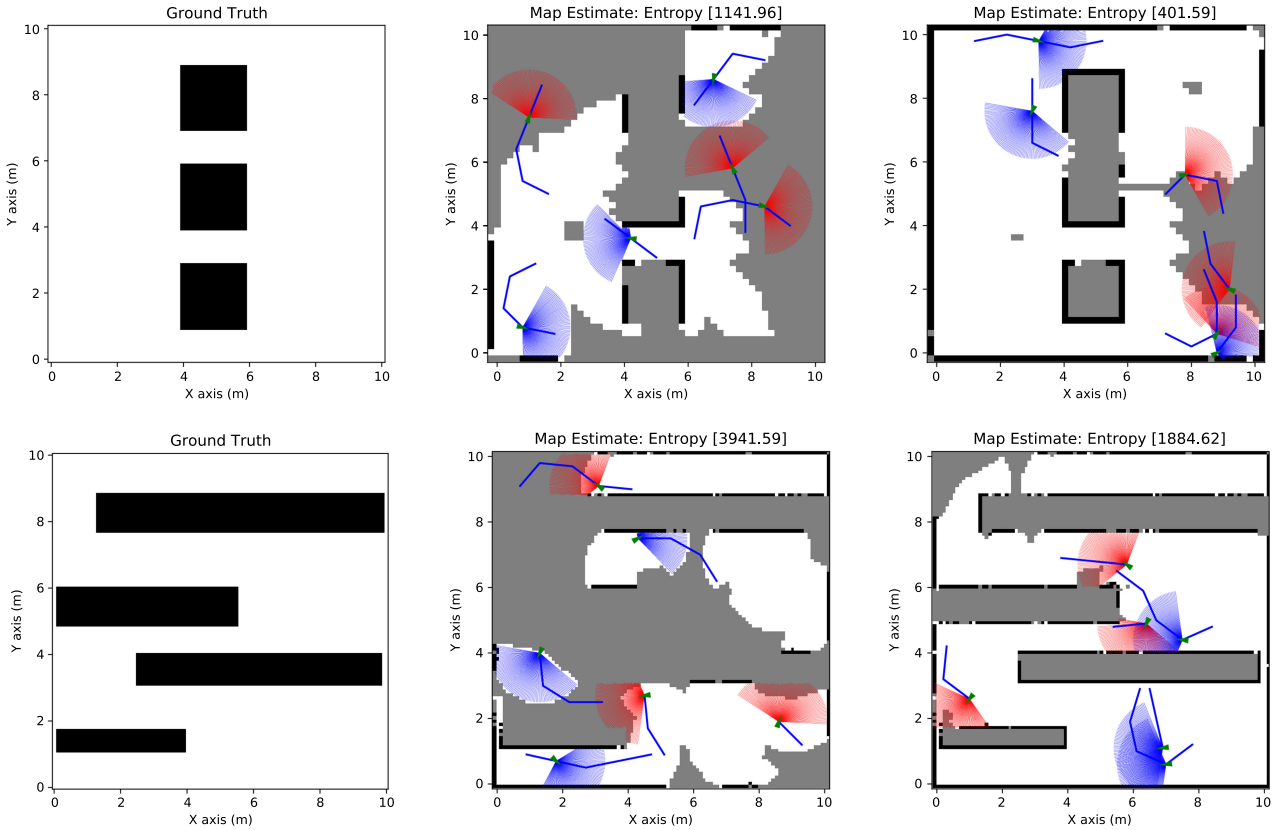


Fig. 4. *Resilient occupancy grid mapping scenarios*. Two scenarios are considered: a square obstacle map (top row) and a corridor map (bottom row), where free space is colored white, occupied space is colored black, and unexplored/unknown space is colored gray. The nonattacked robots are shown with their field-of-view colored blue, whereas the attacked robots are shown with field-of-view colored red. The left-most column shows the considered ground truth maps; the middle column shows the map estimate halfway through the task horizon; the right-most column shows the map estimate near completion.

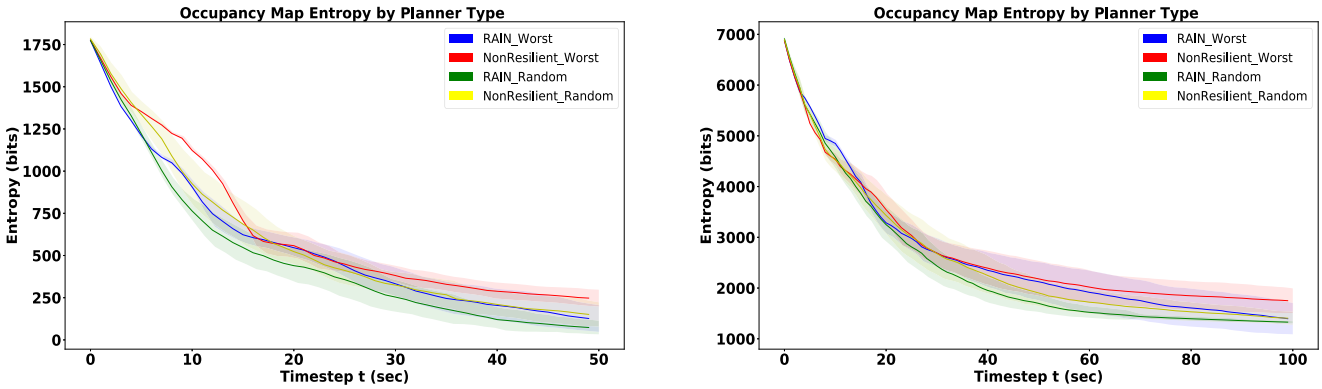


Fig. 5. *Resilient occupancy mapping results.* Comparison of achieved entropy by RAIN against coordinate descent (noted as NonResilient in the plots) for increasing time and for two types of attacks, worst case and random: (left plot) result for the square obstacles map (top row of Fig. 4); (right plot) result for the corridor map (bottom row of Fig. 4).

described in the following paragraph. The attacked set of robots is recomputed at the end of each planning duration. We evaluate the performance on two map environments, which we will refer to as the *square obstacles map* (see Fig 4, top) and the *corridor map* (see Fig 4, bottom). We use $T_{TASK} = 50$ and $T_{TASK} = 100$ for the squares and corridor map, respectively, and $T_{REPLAN} = T_{TASK}$.

3) Compared Attack Models: We test RAIN's ability to be effective even against *non*worst-case failures. To this end, beyond considering the worst-case attack model prescribed by (P-RAIN)'s problem formulation (cf., red and blue curves in Fig. 5), we also consider random attacks, chosen with uniformly random assignment among the robots in $\mathcal{V}$, given the attacks number α (cf., green and yellow curves Fig. 5).

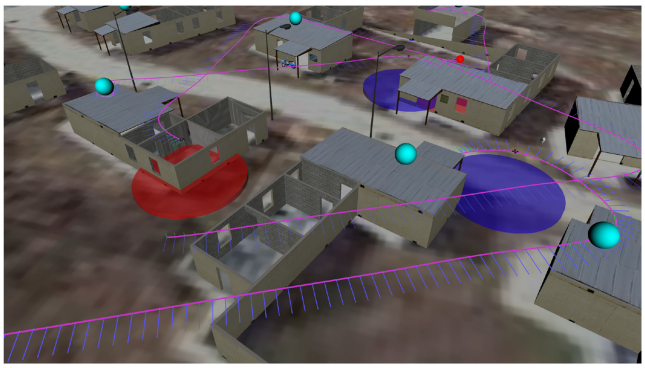


Fig. 6. *Resilient persistent surveillance scenario.* Camp Lejeune 3-D environment. The robots' trajectories are shown in pink. The blue lines along the trajectories indicate the velocity profile generated along the trajectory. The blue discs show the field of view of the nonattacked robots. A red colored field of view indicates an attacked robot. The cyan spheres represent the relative uncertainty on the landmarks' locations. The landmarks are depicted as the red spheres.

4) Results: The results, averaged across 50 Monte Carlo runs, are shown in Fig. 5. The plots indicate RAIN always improves performance. Specifically, RAIN improves performance both when worst-case attacks are present (cf., blue and red curves in Fig. 5), and when random attacks are present (cf., green and yellow curves in Fig. 5); in both cases, RAIN attains lesser map entropy against coordinate descent (noted as NonResilient in the plots). Both the square obstacles map (left plot in Fig. 5) and the corridor map (right plot in Fig. 5) support this conclusion. Moreover, Fig. 5 supports the intuition that since RAIN is designed to withstand the worst-case attacks, RAIN's performance will improve when instead only random failures are present (cf., blue and green curves in Fig. 5).

C. Resilient Persistent Surveillance

In *resilient persistent surveillance*, the robots' objective is to revisit a series of static and known landmarks while the robots are under attack. The landmarks represent points of interest in the environment. For example, a team of robots may be tasked to monitor the entrances to buildings for intruders [40]; the task becomes especially interesting as the number of entrances becomes more than the number of robots. In this section, we choose the landmarks to be a set of buildings in an outdoor camp environment (see Fig. 6). We use the simulated scenarios to determine the effect of the replanning rate on RAIN's performance (cf., Footnote 5).

1) Experimental Setup: The environment used is a 3-D environment provided by ARL DCIST (see Fig. 6). It contains a set of outdoor buildings, over which we place landmarks to encourage visitation (one landmark per building). To have the robots (re)visit the landmarks, we add artificial uncertainty to the location of each landmark by proportionally increasing the uncertainty with time passed since the last observation. The software simulation stack used is based on the robot operating system; the back-end physics are based on unity. In all experiments, the map is assumed to be known. Localization is provided by the simulator. The simulations here run on a 3.4-GHz CPU with 16 GB of RAM. We next specify the used *robot dynamics*, *target process*, *sensor model*, and *information acquisition objective function.*

a) Robot dynamics: The robot motion model is adapted from the 2-D in (12) to the following 3-D:

$$\begin{pmatrix} x^1_{t+1} \\ x^2_{t+1} \\ x^3_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x^1_t \\ x^2_t \\ x^3_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} \nu\,\mathrm{sinc}(\frac{\omega\tau}{2})\cos(\theta_t + \frac{\omega\tau}{2}) \\ \nu\,\mathrm{sinc}(\frac{\omega\tau}{2})\sin(\theta_t + \frac{\omega\tau}{2}) \\ 0 \\ \tau\omega \end{pmatrix}.$$

That is, we assume the quadrotors to fly at a fixed height over the environment ($x^3_{t+1} = x^3_t$ always).

b) Target process: The targets are assumed static in location, but corrupted with uncertainty that increases over time to encourage (re)visitation by the robots, according to a noise covariance matrix $qk_{t,m}I_3$, where q is the rate of uncertainty increase, and $k_{t,m}$ denotes the number of time steps since target m was last visited

$$y_{t+1,m} = y_{t,m} + w_t, \quad w_t \sim \mathsf{N}(0, qk_{t,m}I_3).$$

c) Sensor model: We assume that the robots operate a 360° field of view downward facing sensor. In particular, we assume a range sensing model that records information as long as the robots are within some radius r_{sense} from a landmark; otherwise, no information is granted to the robot. The range-based model for detecting the buildings is as follows:

$$z_{t,m} = h(x_t, y_{t,m}) + v_t, \quad v_t \sim \mathsf{N}(0, V(x_t, y_{t,m}))$$

$$h(x, y_m) = \left[r(x, y_m)\right] \triangleq \sqrt{\sum\nolimits_{i=1}^{3}(y^1 - x^1)^2}.$$

d) Information acquisition objective function: We use the same information acquisition objective function as in the *multi-target tracking* scenarios (see Section VI-A).

2) Simulation Setup: The admissible control inputs are $\mathcal{U} = \{(\nu, \omega) : \nu \in \{1, 3\}m/s, \omega \in \{0, 1, 2\}\}$. r_{sense} is 10 m, the task duration is $T_{TASK} = 300$ steps, and the planning horizon is $T_{PLAN} = 10$ steps. Specifically, each timestep has duration $\tau =$ 1s. The noise parameter is $q = 0.01$.

3) Performance Metric: We measure RAIN's performance by computing the average number of timesteps that a building goes unobserved for. For example, if a landmark is observed at timestep k, and not observed again until timestep $k + l$, we record l as the number of timesteps the landmark was unobserved. Particularly, we average these durations across all targets and timesteps for a given experimental trial.

4) Results: The results, averaged across 50 Monte Carlo runs, are shown in Fig. 7. In Fig. 7, we observe even for the highest replanning rate ($T_{REPLAN} = 1$), RAIN offers a performance gain of $\simeq 24\%$ in comparison to coordinate descent (noted as NonResilient in Fig. 7). The gain increases on average, the lower the replanning rate becomes, as expected (cf., Footnote 5). More broadly, Fig. 7 supports the intuition that a higher replanning rate allows even a nonresilient algorithm, such as coordinate descent, to respond to attacks rapidly, and thus perform well.

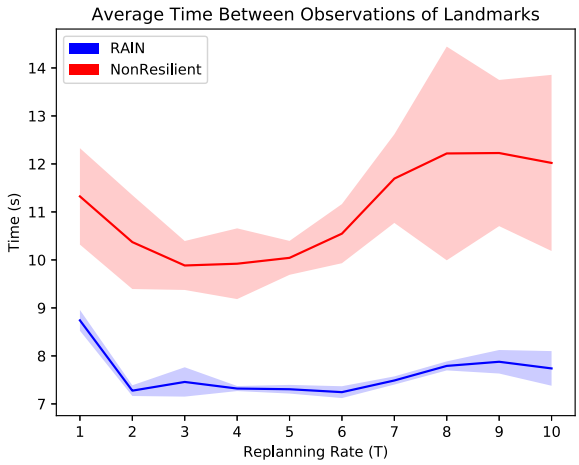


Fig. 7. *Resilient persistent surveillance results.* Comparison of average time between consecutive observations of the same landmarks by RAIN and coordinate descent (noted as NonResilient in the plot) for increasing replanning values T_{REPLAN}.

Still, in Fig. 7, RAINdominates coordinate descent across all possible replanning rate values.

VII. Conclusion

We introduced the first receding-horizon framework for *resilient multirobot path planning* against attacks that disable robots' sensors during information acquisition tasks [cf., (P-RAIN)]. We proposed RAIN, a robust and adaptive multirobot planner against any number of attacks. RAIN calls, in an online fashion, RTP, a subroutine that plans attack-robust control inputs over a look-ahead planning horizon. We quantified RTP's performance by bounding its suboptimality, using notions of curvature for set function optimization. We demonstrated the necessity for *resilient multirobot path planning*, as well as RAIN's effectiveness, in information acquisition scenarios of *multitarget tracking*, *occupancy grid mapping*, and *persistent surveillance*. In all simulations, RAIN was observed to run in real time, and exhibited superior performance against a state-of-the-art baseline, (nonresilient) *coordinate descent* [6]. Across the three scenarios, RAIN's exhibited robustness and superiority even in the presence of a high number of attacks, against varying models of attacks, and high replanning rates. Future work includes extending the proposed framework and algorithms to distributed settings [7], [16].

Appendix

In the appendixes that follow, we prove Theorem 1 (see Appendix C) and Proposition 1 (see Appendix D). To this end, we first present supporting lemmas (see Appendix A) and the algorithm *coordinate descent* [6] (see Appendix B). We also use the notation.

Notation: Consider a finite set $\mathcal{V}$ and a set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$. Then, for any set $\mathcal{X} \subseteq \mathcal{V}$ and any set $\mathcal{X}' \subseteq \mathcal{V}$, the symbol $f(\mathcal{X}|\mathcal{X}')$ denotes the marginal value $f(\mathcal{X} \cup \mathcal{X}') - f(\mathcal{X}')$. We also introduce notation emphasizing that subsets of robots may use different algorithms to compute their control inputs: we let $J(u^a_{\mathcal{A},\,1:T}, u^b_{\mathcal{B},\,1:T}) \triangleq J_{\mathcal{A}^a \cup \mathcal{B}^b,\,1:T}$ indicate that the robots in $\mathcal{A}$ contribute their measurements to J and their control inputs are chosen with algorithm a (*e.g.*, coordinate descent), whereas robots in $\mathcal{B}$ also contribute their measurements to J but their inputs are chosen with another algorithm b. We also occasionally drop the subscript for time indices, since all time indices in the appendixes are identical, (namely, $t+1:t+T_{PLAN}$). Similarly, when only the set of robots is important, we use the notation $J(\mathcal{A}) \triangleq J_{\mathcal{A},\,t+1:t+T_{PLAN}}$, for any $\mathcal{A} \subseteq \mathcal{V}$. Finally, the notation $J(\emptyset)$ refers to the information measure evaluated without any measurements from the robot set.

Appendix A
Preliminary Lemmas

The proof of the lemmas is also found in [27] and [41].

Lemma 1: Consider a finite set $\mathcal{V}$ and a nondecreasing and submodular set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that f is nonnegative and $f(\emptyset) = 0$. For any $\mathcal{A} \subseteq \mathcal{V}$

$$f(\mathcal{A}) \geq (1 - \kappa_f) \sum_{a \in \mathcal{A}} f(a).$$

Proof of Lemma 1: Let $\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$. We prove Lemma 3 by proving the following two inequalities:

$$f(\mathcal{A}) \geq \sum_{i=1}^{|\mathcal{A}|} f(a_i|\mathcal{V} \setminus \{a_i\}) \tag{14}$$

$$\sum_{i=1}^{|\mathcal{A}|} f(a_i|\mathcal{V} \setminus \{a_i\}) \geq (1 - \kappa_f) \sum_{i=1}^{|\mathcal{A}|} f(a_i). \tag{15}$$

We begin with the proof of inequality (14)

$$f(\mathcal{A}) = f(\mathcal{A}|\emptyset) \tag{16}$$

$$\geq f(\mathcal{A}|\mathcal{V} \setminus \mathcal{A}) \tag{17}$$

$$= \sum_{i=1}^{|\mathcal{A}|} f(a_i|\mathcal{V} \setminus \{a_i, a_{i+1}, \ldots, a_{|\mathcal{A}|}\}) \tag{18}$$

$$\geq \sum_{i=1}^{|\mathcal{A}|} f(a_i|\mathcal{V} \setminus \{a_i\}) \tag{19}$$

where inequalities (17)–(19) hold for the following reasons: inequality (17) is implied by (16) because f is submodular and $\emptyset \subseteq \mathcal{V} \setminus \mathcal{A}$; (18) holds since for any sets $\mathcal{X} \subseteq \mathcal{V}$ and $\mathcal{Y} \subseteq \mathcal{V}$, we have $f(\mathcal{X}|\mathcal{Y}) = f(\mathcal{X} \cup \mathcal{Y}) - f(\mathcal{Y})$, and also $\{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$ denotes the set $\mathcal{A}$; and inequality (19) holds since f is submodular and $\mathcal{V} \setminus \{a_i, a_{i+1}, \ldots, a_\mu\} \subseteq \mathcal{V} \setminus \{a_i\}$. These observations complete the proof of inequality (14).

We now prove inequality (15) using the Definition 4 of κ_f, as follows: since $\kappa_f = 1 - \min_{v \in \mathcal{V}} \frac{f(v|\mathcal{V}\setminus\{v\})}{f(v)}$, it is implied that for all elements $v \in \mathcal{V}$, it is $f(v|\mathcal{V} \setminus \{v\}) \geq (1 - \kappa_f) f(v)$. Therefore, adding the latter inequality across all elements $a \in \mathcal{A}$ completes the proof of inequality (15).

Lemma 2: Consider any finite set $\mathcal{V}$, a nondecreasing and submodular $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$, and nonempty sets $\mathcal{Y}, \mathcal{P} \subseteq \mathcal{V}$ such that

for all $y \in \mathcal{Y}$ and all $p \in \mathcal{P}$ $f(y) \geq f(p)$. Then

$$f(\mathcal{P}|\mathcal{Y}) \leq |\mathcal{P}| f(\mathcal{Y}).$$

Proof of Lemma 2: Consider any $y \in \mathcal{Y}$ (such an element exists since Lemma 2 considers that $\mathcal{Y}$ is nonempty), then

$$\begin{aligned} f(\mathcal{P}|\mathcal{Y}) &= f(\mathcal{P} \cup \mathcal{Y}) - f(\mathcal{Y}) && (20)\\ &\leq f(\mathcal{P}) + f(\mathcal{Y}) - f(\mathcal{Y}) && (21)\\ &= f(\mathcal{P}) \\ &\leq \sum_{p \in \mathcal{P}} f(p) && (22)\\ &\leq |\mathcal{P}| \max_{p \in \mathcal{P}} f(p) \\ &\leq |\mathcal{P}| f(y) && (23)\\ &\leq |\mathcal{P}| f(\mathcal{Y}) && (24) \end{aligned}$$

where (20) to inequality (24) hold for the following reasons: (20) holds since for any sets $\mathcal{X} \subseteq \mathcal{V}$ and $\mathcal{Y} \subseteq \mathcal{V}$, $f(\mathcal{X}|\mathcal{Y}) = f(\mathcal{X} \cup \mathcal{Y}) - f(\mathcal{Y})$; inequality (21) holds since f is submodular and, as a result, the submodularity Definition 3 implies that for any set $\mathcal{A} \subseteq \mathcal{V}$ and $\mathcal{A}' \subseteq \mathcal{V}$, $f(\mathcal{A} \cup \mathcal{A}') \leq f(\mathcal{A}) + f(\mathcal{A}')$; inequality (22) holds for the same reason as inequality (21); inequality (23) holds since for all elements $y \in \mathcal{Y}$ and all elements $p \in \mathcal{P}$ $f(y) \geq f(p)$; finally, inequality (24) holds because f is monotone and $y \in \mathcal{Y}$.

Lemma 3: Consider a finite set $\mathcal{V}$ and a nondecreasing $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that f is nonnegative and $f(\emptyset) = 0$. For any set $\mathcal{A} \subseteq \mathcal{V}$ and any set $\mathcal{B} \subseteq \mathcal{V}$ such that $\mathcal{A} \cap \mathcal{B} = \emptyset$

$$f(\mathcal{A} \cup \mathcal{B}) \geq (1 - c_f) \left(f(\mathcal{A}) + \sum_{b \in \mathcal{B}} f(b) \right).$$

Proof of Lemma 3: Let $\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}$. Then

$$f(\mathcal{A} \cup \mathcal{B}) = f(\mathcal{A}) + \sum_{i=1}^{|\mathcal{B}|} f(b_i | \mathcal{A} \cup \{b_1, b_2, \ldots, b_{i-1}\}). \quad (25)$$

In addition, Definition 5 of total curvature implies

$$\begin{aligned} f(b_i | \mathcal{A} \cup \{b_1, b_2, \ldots, b_{i-1}\}) &\geq (1 - c_f) f(b_i | \emptyset) \\ &= (1 - c_f) f(b_i) \quad (26) \end{aligned}$$

where the latter equation holds since $f(\emptyset) = 0$. The proof is completed by substituting (26) in (25) and then taking into account that $f(\mathcal{A}) \geq (1 - c_f) f(\mathcal{A})$ since $0 \leq c_f \leq 1$.

Lemma 4: Consider a finite set $\mathcal{V}$ and a nondecreasing $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that f is nonnegative and $f(\emptyset) = 0$. For any $\mathcal{A} \subseteq \mathcal{V}$ and any $\mathcal{B} \subseteq \mathcal{V}$ such that $\mathcal{A} \setminus \mathcal{B} \neq \emptyset$

$$f(\mathcal{A}) + (1 - c_f) f(\mathcal{B}) \geq (1 - c_f) f(\mathcal{A} \cup \mathcal{B}) + f(\mathcal{A} \cap \mathcal{B}).$$

Proof of Lemma 4: Let $\mathcal{A} \setminus \mathcal{B} = \{i_1, i_2, \ldots, i_r\}$, where $r = |\mathcal{A} - \mathcal{B}|$. From Definition 5 of total curvature c_f, for any $i = 1, 2, \ldots, r$, it is $f(i_j | \mathcal{A} \cap \mathcal{B} \cup \{i_1, i_2, \ldots, i_{j-1}\}) \geq (1 - c_f) f(i_j | \mathcal{B} \cup \{i_1, i_2, \ldots, i_{j-1}\})$. Summing these r inequalities

$$f(\mathcal{A}) - f(\mathcal{A} \cap \mathcal{B}) \geq (1 - c_f) \left(f(\mathcal{A} \cup \mathcal{B}) - f(\mathcal{B}) \right)$$

which implies the lemma.

Corollary 1: Consider a finite set $\mathcal{V}$ and a nondecreasing $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that f is nonnegative and $f(\emptyset) = 0$. For any $\mathcal{A} \subseteq \mathcal{V}$ and any $\mathcal{B} \subseteq \mathcal{V}$ such that $\mathcal{A} \cap \mathcal{B} = \emptyset$

$$f(\mathcal{A}) + \sum_{b \in \mathcal{B}} f(b) \geq (1 - c_f) f(\mathcal{A} \cup \mathcal{B}).$$

Proof of Corollary 1: Let $\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}$

$$\begin{aligned} f(\mathcal{A}) + \sum_{i=1}^{|\mathcal{B}|} f(b_i) &\geq (1 - c_f) f(\mathcal{A}) + \sum_{i=1}^{|\mathcal{B}|} f(b_i)) \\ &\geq (1 - c_f) f(\mathcal{A} \cup \{b_1\}) + \sum_{i=2}^{|\mathcal{B}|} f(b_i) \\ &\geq (1 - c_f) f(\mathcal{A} \cup \{b_1, b_2\}) + \sum_{i=3}^{|\mathcal{B}|} f(b_i) \\ &\vdots \\ &\geq (1 - c_f) f(\mathcal{A} \cup \mathcal{B}) \quad (27) \end{aligned}$$

where (27) holds since $0 \leq c_f \leq 1$, and the rest due to Lemma 4 since $\mathcal{A} \cap \mathcal{B} = \emptyset$ implies $\mathcal{A} \setminus \{b_1\} \neq \emptyset$, $\mathcal{A} \cup \{b_1\} \setminus \{b_2\} \neq \emptyset$, $\ldots$, $\mathcal{A} \cup \{b_1, b_2, \ldots, b_{|\mathcal{B}|-1}\} \setminus \{b_{|\mathcal{B}|}\} \neq \emptyset$.

APPENDIX B
COORDINATE DESCENT

We describe *coordinate descent* [6, Sec. IV], and generalize the proof in [6] that *coordinate descent* guarantees an approximation performance up to a multiplicative factor $1/2$ the optimal when the information objective function is the mutual information. In particular, we extend the proof to *any* nondecreasing and possibly submodular information objective function; the result will support the proof of Proposition 1.

The algorithm coordinate descent works as follows: consider an *arbitrary* ordering of the robots in $\mathcal{V}$, such that $\mathcal{V} \equiv \{1, 2, \ldots, n\}$, and suppose that robot 1 first chooses its controls, without considering the other robots; in other words, robot 1 solves the single robot version of Problem 2, *i.e.*, $J_{\{1\}, t+1:t+T_{PLAN}} := J(u_1)$, to obtain controls $u_{\{1\}}$ such that

$$u^{cd}_{1,\, t+1:t+T_{PLAN}} = \arg \max_{\substack{u_{i,t'} \in \mathcal{U}_{i,t'} \\ t' = t+1 \,:\, t+T_{PLAN}}} J(u_{i,\, t+1:t+T_{PLAN}}). \quad (28)$$

Afterward, robot 1 communicates its chosen control sequence to robot 2, and robot 2, given the control sequence of robot 1, computes its control input as follows, assuming the control inputs for robot 1 are fixed:

$$\begin{aligned} &u^{cd}_{2,\, t+1:t+T_{PLAN}} \\ &= \arg \max_{\substack{u_{i,t'} \in \mathcal{U}_{i,t'} \\ t' = t+1 \,:\, t+T_{PLAN}}} J(u^{cd}_1, u_{i,\, t+1:t+T_{PLAN}}). \quad (29) \end{aligned}$$

This continues such that robot i solves a single robot problem, given the control inputs from the robots $1, 2, \ldots, i-1$

$$u^{cd}_{i,\,t+1:t+T_{PLAN}} = \arg \max_{\substack{u_{i,t'} \in \mathcal{U}_{i,t'} \\ t' = t+1 : t+T_{PLAN}}} J(u^{cd}_{1:i-1}, u_{i,\,t+1:t+T_{PLAN}}). \quad (30)$$

Notably, if we let u^*_i be the control inputs for the ith robot resulting from the optimal solution to the n robot problem, then from the coordinate descent algorithm, we have

$$J(u^{cd}_{1:i-1}, u^\star_i) \leq J(u^{cd}_{1:i}). \quad (31)$$

Lemma 5 (Approximation Performance of Coordinate Descent): Consider a set of robots $\mathcal{V}$, and an instance of problem $(P-RTP)$. Denote the optimal control inputs for problem $(P-RTP)$, across all robots and all times, by $u^\star_{\mathcal{V},t+1\,:\,t+T_{PLAN}}$. The coordinate descent algorithm returns control inputs $u^{cd}_{\mathcal{V},\,t+1\,:\,t+T_{PLAN}}$, across all robots and all times, such that the following hold.

1) If the objective function J is nondecreasing submodular in the active robot set, and (without loss of generality) J is nonnegative and $J(\emptyset)=0$, then
$$\frac{J(u^{cd}_{1:n})}{J(u^\star_{1:n})} \geq \frac{1}{2}. \quad (32)$$
2) If the objective function J is nondecreasing in the active robot set, and (without loss of generality) J is nonnegative and $J(\emptyset)=0$, then
$$\frac{J(u^{cd}_{1:n})}{J(u^\star_{1:n})} \geq \frac{1-c_J}{2}. \quad (33)$$

Proof of Lemma 5:

1) If the objective function J is nondecreasing and submodular in the active robot set, and (without loss of generality) J is nonnegative and $J(\emptyset)=0$, then
$$3J(u^\star_{1:n}) \leq J(u^\star_{1:n}) + \sum_{i=1}^{n}[J(u^{cd}_{1:i}, u^\star_{i+1:n}) - J(u^{cd}_{1:i-1}, u^\star_{i+1:n})] \quad (34)$$
$$= J(u^{cd}_{1:n}) + \sum_{i=1}^{n}[J(u^{cd}_{1:i-1}, u^\star_{i:n}) - J(u^{cd}_{1:i-1}, u^\star_{i+1:n})] \quad (35)$$
$$= J(u^{cd}_{1:n}) + \sum_{i=1}^{n} J(u^\star_i|\{u^{cd}_{1:i-1}, u^\star_{i+1,n}\}) \quad (36)$$
$$\leq J(u^{cd}_{1:n}) + \sum_{i=1}^{n} J(u^\star_i|u^{cd}_{1:i-1}) \quad (37)$$
$$\leq J(u^{cd}_{1:n}) + \sum_{i=1}^{n} J(u^{cd}_i|u^{cd}_{1:i-1}) \quad (38)$$
$$\leq J(u^{cd}_{1:n}) + J(u^{cd}_{1:n}) \quad (39)$$
$$\leq 2J(u^{cd}_{1:n}) \quad (40)$$

where inequality (34) holds due to monotonicity of J; (35) is a shift in indexes of the first term in the sum; (36) is an expression of the sum as a sum of marginal gains; inequality (37) holds due to submodularity; inequality (38) holds by the coordinate-descent policy [per (31)]; (39) holds due to the definition of the marginal gain symbol $J(u^\star_i|u^{cd}_{1:i-1})$ (for any $i=1,2,\ldots,n$) as $J(u^\star_i, u^{cd}_{1:i-1}) - J(u^{cd}_{1:i-1})$; finally, a rearrangement of the terms in (40) gives $J(u^{cd}_{1:n})/J(u^*_{1:n}) \geq 1/2$.

2) If J is nondecreasing in the active robot set, and (without loss of generality) J is nonnegative and $J(\emptyset)=0$, then multiplying both sides of (36) (which holds for any nondecreasing J) with $(1-c_J)$, we have
$$(1-c_J)J(u^\star_{1:n}) = (1-c_J)J(u^{cd}_{1:n}) + (1-c_J)\sum_{i=1}^{n} J(u^\star_i|\{u^{cd}_{1:i-1}, u^\star_{i+1,n}\})$$
$$\leq J(u^{cd}_{1:n}) + (1-c_J)\sum_{i=1}^{n} J(u^\star_i|\{u^{cd}_{1:i-1}, u^\star_{i+1,n}\}) \quad (41)$$
$$\leq J(u^{cd}_{1:n}) + \sum_{i=1}^{n} J(u^\star_i|u^{cd}_{1:i-1}) \quad (42)$$
$$\leq J(u^{cd}_{1:n}) + \sum_{i=1}^{n} J(u^{cd}_i|u^{cd}_{1:i-1}) \quad (43)$$
$$= J(u^{cd}_{1:n}) + J(u^{cd}_{1:n}) \quad (44)$$
$$\leq 2J(u^{cd}_{1:n}) \quad (45)$$

where inequality (41) holds since $0 \leq c_J \leq 1$; inequality (42) holds since J is nondecreasing in the set of active robots, and Definition 5 of total curvature implies that for any nondecreasing set function $g: 2^{\mathcal{V}} \mapsto \mathbb{R}$, for any element $v \in \mathcal{V}$, and for any set $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V} \setminus \{v\}$

$$(1-c_g)g(v|\mathcal{B}) \leq g(\{v\}|\mathcal{A}). \quad (46)$$

Inequality (43) holds by the coordinate-descent algorithm; (44) holds due to the definition of the marginal gain symbol $J(u^\star_i|u^{cd}_{1:i-1})$ (for any $i=1,2,\ldots,n$) as $J(u^\star_i, u^{cd}_{1:i-1}) - J(u^{cd}_{1:i-1})$; finally, a rearrangement of terms gives $J(u^{cd}_{1:n})/J(u^*_{1:n}) \geq (1-c_J)/2$.

APPENDIX C
PROOF OF THEOREM 1

We first prove Theorem 1's part 1 (approximation performance), and then, Theorem 1's part 2 (running time).

A. Proof of Theorem 1's Part 1 (Approximation Performance)

The proof follows the steps of the proof of [27, Th. 1] and [41, Th. 1]. We first prove (9), then (8).

To the aforementioned ends, we use the following notation (along with the notation introduced in Theorem 1 and in Appendix 1): given that using Algorithm 2, the robots in $\mathcal{V}$ select control inputs $u_{\mathcal{V},t+1:t+T_{PLAN}}$, then, for notational simplicity, the following hold.

1) Let $\mathcal{A}^\star \triangleq \mathcal{A}^\star(u_{\mathcal{V},\, t+1:t+T_{PLAN}})$.
2) Let $\mathcal{L}^+ \triangleq \mathcal{L} \setminus \mathcal{A}^\star$, *i.e.*, $\mathcal{S}_1$ be the remaining robots in $\mathcal{L}$ after the removal of the robots in $\mathcal{A}^\star$.
3) Let $(\mathcal{V} \setminus \mathcal{L})^+ \triangleq (\mathcal{V} \setminus \mathcal{L}) \setminus \mathcal{A}^\star$, *i.e.*, $\mathcal{S}_2$ be the remaining robots in $\mathcal{V} \setminus \mathcal{L}$ after the removal of the robots in $\mathcal{A}^\star$.

Proof of inequality (9): The proof follows the steps of the proof of [27, Th. 1]. Consider that the objective function J is nondecreasing and submodular in the active robot set, such that (without loss of generality) J is nonnegative and $J(\emptyset) = 0$. We first prove the part $1 - \kappa_J$ of the bound in the right-hand side of inequality (9), and then, the part $h(|\mathcal{V}|, \alpha)$ of the bound in the right-hand side of inequality (9).

To prove the part $1 - \kappa_J$ of the bound in the right-hand side of inequality (9), we follow the steps of the proof of [27, Th. 1], and make the following observations:

$$
\begin{aligned}
&J(\mathcal{V} \setminus \mathcal{A}^\star) \\
&\quad = J(\mathcal{L}^+ \cup (\mathcal{V} \setminus \mathcal{L})^+) && (47)\\
&\quad \geq (1 - \kappa_J) \sum_{v \in \mathcal{L}^+ \cup (\mathcal{V} \setminus \mathcal{L})^+} J(v) && (48)\\
&\quad \geq (1 - \kappa_J) \left(\sum_{v \in (\mathcal{V} \setminus \mathcal{L}) \setminus (\mathcal{V} \setminus \mathcal{L})^+} J(v) + \sum_{v \in (\mathcal{V} \setminus \mathcal{L})^+} J(v) \right) && (49)\\
&\quad \geq (1 - \kappa_J) J\{[(\mathcal{V} \setminus \mathcal{L}) \setminus (\mathcal{V} \setminus \mathcal{L})^+] \cup (\mathcal{V} \setminus \mathcal{L})^+\} && (50)\\
&\quad = (1 - \kappa_J) J(\mathcal{V} \setminus \mathcal{L}) && (51)
\end{aligned}
$$

where (47)–(51) hold for the following reasons: (47) follows from the definitions of the sets $\mathcal{L}^+$ and $(\mathcal{V} \setminus \mathcal{L})^+$; inequality (48) follows from inequality (47) due to Lemma 1; inequality (49) follows from inequality (48) because for all elements $v \in \mathcal{L}^+$ and all elements $v' \in (\mathcal{V} \setminus \mathcal{L}) \setminus (\mathcal{V} \setminus \mathcal{L})^+$, we have $J(v) \geq J(v')$ (note that due to the definitions of the sets $\mathcal{L}^+$ and $(\mathcal{V} \setminus \mathcal{L})^+$, $|\mathcal{L}^+| = |(\mathcal{V} \setminus \mathcal{L}) \setminus (\mathcal{V} \setminus \mathcal{L})^+|$, that is, the number of nonremoved elements in $\mathcal{L}$ is equal to the number of removed elements in $\mathcal{V} \setminus \mathcal{L}$); finally, inequality (50) follows from inequality (49) because the set function J is submodular and, as a result, the submodularity Definition 3 implies that for any sets $\mathcal{S} \subseteq \mathcal{V}$ and $\mathcal{S}' \subseteq \mathcal{V}$, $J(\mathcal{S}) + J(\mathcal{S}') \geq J(\mathcal{S} \cup \mathcal{S}')$ [24, Proposition 2.1]. We now complete the proof of the part $1 - \kappa_J$ of the bound in the right-hand side of inequality (9) by proving that in inequality (51)

$$
J(\mathcal{V} \setminus \mathcal{L}) \geq J^\star \tag{52}
$$

when the robots in $\mathcal{V}$ optimally solve the problems in Algorithm 2's step 8, per the statement of Theorem 1. In particular, if for any active robot set $\mathcal{R} \subseteq \mathcal{V}$, we let $\bar{u}_{\mathcal{R}} \triangleq \{\bar{u}_{i,t'} : \bar{u}_{i,t'} \in \mathcal{U}_{i,t'}, i \in \mathcal{R}, t' = t+1, \ldots, t+T_{PLAN}\}$ denote a collection of control inputs to the robots in $\mathcal{R}$, then

$$
\begin{aligned}
J(\mathcal{V} \setminus \mathcal{L}) \equiv & \max_{\substack{\bar{u}_{i,t} \in \mathcal{U}_{i,t}, i \in \mathcal{V}, \\ t' = t+1 : t+T_{PLAN}}} J(\bar{u}_{\mathcal{V} \setminus \mathcal{L}, t+1:t+T_{PLAN}}) && (53)\\
\geq & \min_{\substack{\bar{\mathcal{L}} \subseteq \mathcal{V}, \\ |\bar{\mathcal{L}}| \leq \alpha}} \max_{\substack{\bar{u}_{i,t} \in \mathcal{U}_{i,t}, i \in \mathcal{V}, \\ t' = t+1 : t+T_{PLAN}}} J(\bar{u}_{\mathcal{V} \setminus \bar{\mathcal{L}}, t+1:t+T_{PLAN}}) && (54)\\
\geq & \max_{\substack{\bar{u}_{i,t} \in \mathcal{U}_{i,t}, i \in \mathcal{V}, \\ t = t+1 : t+T_{PLAN}}} \min_{\substack{\bar{\mathcal{L}} \subseteq \mathcal{V}, \\ |\bar{\mathcal{L}}| \leq \alpha}} J(\bar{u}_{\mathcal{V} \setminus \bar{\mathcal{L}}, t+1:t+T_{PLAN}}) && (55)\\
\equiv & J^\star && (56)
\end{aligned}
$$

where (53)–(56) hold true since: the equivalence in (53) holds since the robots in $\mathcal{V}$ solve optimally the problems in Algorithm 2's step 8, per the statement of Theorem 1; (54) holds since we minimize over the set $\mathcal{L}$; (55) holds because for any set $\hat{\mathcal{L}} \subseteq \mathcal{V}$ and any control inputs $\hat{u}_{\mathcal{R},t+1:t+T_{PLAN}} \triangleq \{\hat{u}_{i,t} : \hat{u}_{i,t} \in \mathcal{U}_{i,t}, i \in \mathcal{R}, t' = t+1, \ldots, t+T_{PLAN}\}$

$$
\begin{aligned}
&\max_{\substack{\bar{u}_{i,t} \in \mathcal{U}_{i,t}, i \in \mathcal{V}, \\ t' = t+1 : t+T_{PLAN}}} J(\bar{u}_{\mathcal{V} \setminus \bar{\mathcal{L}}, t+1:t+T_{PLAN}}) \\
&\quad \geq J(\hat{u}_{\mathcal{V} \setminus \hat{\mathcal{L}}, t+1:t+T_{PLAN}})
\end{aligned}
$$

which implies

$$
\begin{aligned}
&\min_{\substack{\bar{\mathcal{L}} \subseteq \mathcal{V}, \\ |\bar{\mathcal{L}}| \leq \alpha}} \max_{\substack{\bar{u}_{i,t} \in \mathcal{U}_{i,t}, i \in \mathcal{V}, \\ t' = t+1 : t+T_{PLAN}}} J(\bar{u}_{\mathcal{V} \setminus \bar{\mathcal{L}}, t+1:t+T_{PLAN}}) \\
&\qquad \geq \min_{\substack{\bar{\mathcal{L}} \subseteq \mathcal{V}, \\ |\bar{\mathcal{L}}| \leq \alpha}} J(\hat{u}_{\mathcal{V} \setminus \bar{\mathcal{L}}, t+1:t+T_{PLAN}}) \Rightarrow \\
&\min_{\substack{\bar{\mathcal{L}} \subseteq \mathcal{V}, \\ |\bar{\mathcal{L}}| \leq \alpha}} \max_{\substack{\bar{u}_{i,t} \in \mathcal{U}_{i,t}, i \in \mathcal{V}, \\ t' = t+1 : t+T_{PLAN}}} J(\bar{u}_{\mathcal{V} \setminus \bar{\mathcal{L}}, t+1:t+T_{PLAN}}) \\
&\quad \geq \max_{\substack{\bar{u}_{i,t} \in \mathcal{U}_{i,t}, i \in \mathcal{V}, \\ t' = t+1 : t+T_{PLAN}}} \min_{\substack{\bar{\mathcal{L}} \subseteq \mathcal{V}, \\ |\bar{\mathcal{L}}| \leq \alpha}} J(\bar{u}_{\mathcal{V} \setminus \bar{\mathcal{L}}, t+1:t+T_{PLAN}})
\end{aligned}
$$

where the last one is (55); finally, the equivalence in (56) holds since $J^\star$ (per the statement of Theorem 1) denotes the optimal value to Problem 2. Overall, we proved that inequality (56) proves inequality (52); and, now, the combination of inequalities (51) and (52) proves the part $1 - \kappa_J$ of the bound in the right-hand side of inequality (9).

We finally prove the part $1/(1+\alpha)$ of the bound in the right-hand side of inequality (9), and complete this way the proof of Theorem 1. To this end, we follow the steps of the proof of [27, Th. 1], and use the notation introduced in Fig. 8, along with the following notation:

$$
\eta \triangleq \frac{J(\mathcal{A}_2^\star | \mathcal{V} \setminus \mathcal{A}^\star)}{J(\mathcal{V} \setminus \mathcal{L})}. \tag{57}
$$

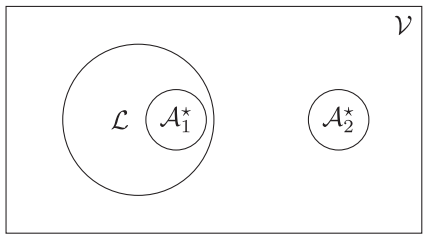


Fig. 8. Venn diagram, where the set $\mathcal{L}$ is the robot set defined in step 2 of Algorithm 2, and the set $\mathcal{A}_1^\star$ and the set $\mathcal{A}_2^\star$ are such that $\mathcal{A}_1^\star = \mathcal{A}^\star \cap \mathcal{L}$ and $\mathcal{A}_2^\star = \mathcal{A}^\star \cap (\mathcal{V} \setminus \mathcal{L})$ (observe that these definitions imply $\mathcal{A}_1^\star \cap \mathcal{A}_2^\star = \emptyset$ and $\mathcal{A}^\star = \mathcal{A}_1^\star \cup \mathcal{A}_2^\star$).

Later in this proof, we prove $0 \leq \eta \leq 1$. We first observe that

$$J(\mathcal{V} \setminus \mathcal{A}^\star) \geq \max\{J(\mathcal{V} \setminus \mathcal{A}^\star), J(\mathcal{L}^+)\} \tag{58}$$

in the following paragraphs, we prove the three inequalities

$$J(\mathcal{V} \setminus \mathcal{A}^\star) \geq (1-\eta) J(\mathcal{V} \setminus \mathcal{L}) \tag{59}$$

$$J(\mathcal{L}^+) \geq \eta \frac{1}{\alpha} J(\mathcal{V} \setminus \mathcal{L}) \tag{60}$$

$$\max\{(1-\eta), \eta\frac{1}{\alpha}\} \geq \frac{1}{\alpha+1}. \tag{61}$$

Then, if we substitute inequalities (59), (60), and (61) to inequality (58), and take into account that $J(\mathcal{V} \setminus \mathcal{L}) \geq 0$, then

$$J(\mathcal{V} \setminus \mathcal{A}^\star) \geq \frac{1}{\alpha+1} J(\mathcal{V} \setminus \mathcal{L})$$

which implies the part $1/(1+\alpha)$ of the bound in the right-hand side of inequality (9), after taking into account inequality (52).

We next complete the proof of the part $1/(1+\alpha)$ of the bound in the right-hand side of inequality (9) by proving $0 \leq \eta \leq 1$, inequalities (59), (60), and (61).

Proof of inequality $0 \leq \eta \leq 1$: We first prove $\eta \geq 0$, and then $\eta \leq 1$: $\eta \geq 0$, since $\eta \equiv J(\mathcal{A}_2^\star|\mathcal{V} \setminus \mathcal{A}^\star)/J(\mathcal{V} \setminus \mathcal{L})$, and J is nonnegative; and $\eta \leq 1$, since $J(\mathcal{V} \setminus \mathcal{L}) \geq J(\mathcal{A}_2^\star)$, due to monotonicity of J and that $\mathcal{A}_2^\star \subseteq \mathcal{V} \setminus \mathcal{L}$, and $J(\mathcal{A}_2^\star) \geq J(\mathcal{A}_2^\star|\mathcal{V} \setminus \mathcal{A}^\star)$, due to submodularity of J and that $\emptyset \subseteq \mathcal{V} \setminus \mathcal{A}^\star$.

Proof of inequality (59): We complete the proof of inequality (59) in two steps. First, it can be verified that

$$\begin{aligned} &f(\mathcal{V} \setminus \mathcal{A}^\star) = f(\mathcal{V} \setminus \mathcal{L}) \\ &- J(\mathcal{A}_2^\star|\mathcal{V} \setminus \mathcal{A}^\star) + J(\mathcal{L}|\mathcal{V} \setminus \mathcal{L}) - J(\mathcal{A}_1^\star|\mathcal{V} \setminus \mathcal{A}_1^\star) \end{aligned} \tag{62}$$

since for any sets $\mathcal{X} \subseteq \mathcal{V}$ and $\mathcal{Y} \subseteq \mathcal{V}$, $J(\mathcal{X}|\mathcal{Y}) \equiv J(\mathcal{X} \cup \mathcal{Y}) - J(\mathcal{Y})$. Second, (62) implies inequality (59), since $J(\mathcal{A}_2^\star|\mathcal{V} \setminus \mathcal{A}^\star) = \eta J(\mathcal{V} \setminus \mathcal{L})$ and $J(\mathcal{L}|\mathcal{V} \setminus \mathcal{L}) - J(\mathcal{A}_1^\star|\mathcal{V} \setminus \mathcal{A}_1^\star) \geq 0$; the latter is true due to the following two observations: $J(\mathcal{L}|\mathcal{V} \setminus \mathcal{L}) \geq J(\mathcal{A}_1^\star|\mathcal{V} \setminus \mathcal{L})$, since J is monotone and $\mathcal{A}_1^\star \subseteq \mathcal{L}$; and $J(\mathcal{A}_1^\star|\mathcal{V} \setminus \mathcal{L}) \geq J(\mathcal{A}_1^\star|\mathcal{V} \setminus \mathcal{A}_1^\star)$, since J is submodular and $\mathcal{V} \setminus \mathcal{L} \subseteq \mathcal{V} \setminus \mathcal{A}_1^\star$ (see also Fig. 8).

Proof of inequality (60): To prove inequality (60), since $\mathcal{A}_2^\star \neq \emptyset$ (and, as a result, also $\mathcal{L}^+ \neq \emptyset$), and for all elements $a \in \mathcal{L}^+$ and all elements $b \in \mathcal{A}_2^\star$, $J(a) \geq J(b)$, from Lemma 2, we have

$$\begin{aligned} J(\mathcal{A}_2^\star|\mathcal{L}^+) &\leq |\mathcal{A}_2^\star| J(\mathcal{L}^+) \\ &\leq \alpha J(\mathcal{L}^+) \end{aligned} \tag{63}$$

since $|\mathcal{A}_2^\star| \leq \alpha$. Overall

$$J(\mathcal{L}^+) \geq \frac{1}{\alpha} J(\mathcal{A}_2^\star|\mathcal{L}^+) \tag{64}$$

$$\geq \frac{1}{\alpha} J(\mathcal{A}_2^\star|\mathcal{L}^+ \cup (\mathcal{V} \setminus \mathcal{L})^+) \tag{65}$$

$$= \frac{1}{\alpha} J(\mathcal{A}_2^\star|\mathcal{V} \setminus \mathcal{A}^\star) \tag{66}$$

$$= \eta \frac{1}{\alpha} J(\mathcal{V} \setminus \mathcal{L}) \tag{67}$$

where inequality (64) to (67) hold for the following reasons: inequality (64) follows from inequality (63); inequality (65) holds since J is submodular and $\mathcal{L}^+ \subseteq \mathcal{L}^+ \cup (\mathcal{V} \setminus \mathcal{L})^+$; (66) holds due to the definitions of the sets $\mathcal{L}^+$, $(\mathcal{V} \setminus \mathcal{L})^+$, and $\mathcal{A}^\star$; finally, (67) holds due to the definition of η. Overall, the latter derivation concludes the proof of inequality (60).

Proof of inequality (61): Let $b = 1/\alpha$. We complete the proof first for the case where $(1-\eta) \geq \eta b$, and then for the case $(1-\eta) < \eta b$: First, when $(1-\eta) \geq \eta b$, $\max\{(1-\eta), \eta b\} = 1-\eta$ and $\eta \leq 1/(1+b)$. Due to the latter, $1-\eta \geq b/(1+b) = 1/(\alpha+1)$ and, as a result, (61) holds. Second, when $(1-\eta) < \eta b$, $\max\{(1-\eta), \eta b\} = \eta b$ and $\eta > 1/(1+b)$. Due to the latter, $\eta b > b/(1+b)$ and, as a result, (61) holds.

We completed the proof of $0 \leq \eta \leq 1$, and of inequalities (59)–(61). Thus, we also completed the proof of the part $1/(1+\alpha)$ of the bound in the right-hand side of inequality (9), and, in sum, the proof of inequality (9).

Proof of inequality (8): Consider that the objective function J is nondecreasing in the active robot set, such that (without loss of generality) J is nonnegative and $J(\emptyset) = 0$.

The proof follows the steps of the proof of [41, Th. 1], by making the following observations:

$$J(\mathcal{V} \setminus \mathcal{A}^\star)$$

$$= J(\mathcal{L}^+ \cup (\mathcal{V} \setminus \mathcal{L})^+) \tag{68}$$

$$\geq (1 - c_J) \sum_{v \in \mathcal{L}^+ \cup (\mathcal{V} \setminus \mathcal{L})^+} J(v) \tag{69}$$

$$\geq (1 - c_J) \left(\sum_{v \in (\mathcal{V} \setminus \mathcal{L}) \setminus (\mathcal{V} \setminus \mathcal{L})^+} J(v) + \sum_{v \in (\mathcal{V} \setminus \mathcal{L})^+} J(v) \right) \tag{70}$$

$$\geq (1 - c_J)^2 J\{[(\mathcal{V} \setminus \mathcal{L}) \setminus (\mathcal{V} \setminus \mathcal{L})^+] \cup (\mathcal{V} \setminus \mathcal{L})^+\} \tag{71}$$

$$= (1 - c_J)^2 J(\mathcal{V} \setminus \mathcal{L}) \tag{72}$$

where (68)–(72) hold for the following reasons: (68) follows from the definitions of the sets $\mathcal{L}^+$ and $(\mathcal{V} \setminus \mathcal{L})^+$; inequality (69) follows from inequality (68) due to Lemma 3; inequality (70) follows from inequality (69) because for all elements $v \in \mathcal{L}^+$ and all elements $v' \in (\mathcal{V} \setminus \mathcal{L}) \setminus (\mathcal{V} \setminus \mathcal{L})^+$, we have $J(v) \geq J(v')$ (note that due to the definitions of the sets $\mathcal{L}^+$ and $(\mathcal{V} \setminus \mathcal{L})^+$ it is $|\mathcal{L}^+| = |(\mathcal{V} \setminus \mathcal{L}) \setminus (\mathcal{V} \setminus \mathcal{L})^+|$, that is, the number of nonremoved elements in $\mathcal{L}$ is equal to the number of removed elements in $\mathcal{V} \setminus \mathcal{L}$); finally, inequality (71) follows from inequality (70) because the set function J is nondecreasing and Corollary 1 applies. Overall, the combination of inequalities (72) and (52)

(observe that inequality (52) still holds if the objective function J is merely nondecreasing) proves inequality (8).

B. Proof of Theorem 1's Part 2 (Running Time)

RTP's running time is found by adding the running time of lines 1–3, *i.e.*, $|\mathcal{V}|\rho$, line 4, *i.e.*, $|\mathcal{V}|\log(|\mathcal{V}|)$ (using, *e.g.*, the merge sort algorithm), lines 5–7, whose running time can be ignored since the optimization problems in line 6 have already been solved in lines 1–3, and line 8, *i.e.*, ρ. The total is $|\mathcal{V}|(\rho+1)+|\mathcal{V}|\log(|\mathcal{V}|)=O(|\mathcal{V}|\rho)$.

Appendix D
Proof of Proposition 1

We first prove Proposition 1's part 1 (approximation bounds), and then, Proposition 1's part 2 (running time).

A. Proof of Proposition 1's Part 1 (Approximation Bounds)

The proof follows the steps of the proof of Theorem 1; hence, we describe here only the steps where the proof differs.

We first prove inequality (11), then, we prove inequality (10).

Proof of inequality (11): Consider that the objective function J is nondecreasing and submodular in the active robot set, such that (without loss of generality) J is nonnegative and $J(\emptyset)=0$. Since, per Proposition 1, Algorithm 2 calls the coordinate descent algorithm in step 4, the equivalence in (53) is now invalid, and, in particular, using Lemma 5, the following inequality holds instead:

$$J(\mathcal{V}\setminus\mathcal{L})\geq\frac{1}{2}\max_{\substack{\bar{u}_{i,t}\in\mathcal{U}_{i,t},i\in\mathcal{V},\\ t'=t+1:t+T_{PLAN}}}J(\bar{u}_{\mathcal{V}\setminus\mathcal{L},t+1:t+T_{PLAN}}). \tag{73}$$

Using inequality (73), and following the same steps as in (53)–(56), we conclude

$$J(\mathcal{V}\setminus\mathcal{L})\geq\frac{1}{2}J^{\star}. \tag{74}$$

Using inequality (74) the same way that inequality (52) was used in the proof of Theorem 1's part 1, inequality (10) is proved.

Proof of inequality (10): Consider that the objective function J is nondecreasing in the active robot set, such that (without loss of generality) J is nonnegative and $J(\emptyset)=0$. Similarly with the observations we made in the proof of inequality (11), since, per Proposition 1, Algorithm 2 calls the coordinate descent algorithm in step 4, the equivalence in (53) is now invalid, and, in particular, using Lemma 5, the following inequality holds instead:

$$J(\mathcal{V}\setminus\mathcal{L})\geq\frac{1-c_J}{2}\max_{\substack{\bar{u}_{i,t}\in\mathcal{U}_{i,t},i\in\mathcal{V},\\ t'=t+1:t+T_{PLAN}}}J(\bar{u}_{\mathcal{V}\setminus\mathcal{L},t+1:t+T_{PLAN}}). \tag{75}$$

Using inequality (75), and following the same steps as in (53)–(56), we conclude

$$J(\mathcal{V}\setminus\mathcal{L})\geq\frac{1-c_J}{2}J^{\star}. \tag{76}$$

Using inequality (76) the same way that inequality (52) was used in the proof of Theorem 1's part 1, inequality (10) is proved.

B. Proof of Proposition 1's Part 2 (Communication Rounds)

Per Appendix B, coordinate descent needs to solve $|\mathcal{V}|$ times an optimization problem of same complexity as line 2 of RTP. RTP, instead, needs to solve $|\mathcal{V}|$ times the optimization problem in line 2 (per the "for" loop in lines 1–3), and the optimization problem in line 8, using coordinate descent. Since the running time of line 4 and lines 5–7 is negligible (cf., Appendix C-B) and can be ignored, the total running time of RTP is at most $2\rho_{CD}=O(\rho_{CD})$.

References

[1] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, no. 8, pp. 966–1005, Aug. 1988.

[2] A. O. Hero III and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors J.*, vol. 11, no. 12, pp. 3064–3075, Dec. 2011.

[3] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6447–6454.

[4] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multi-robot active information gathering," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1025–1032, Apr. 2018.

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

[6] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 4775–4782.

[7] M. Corah and N. Michael, "Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice," *Auton. Robots*, vol. 43, no. 2, pp. 485–501, 2019.

[8] D. Thakur, G. Loianno, W. Liu, and V. Kumar, "Nuclear environments inspection with micro aerial vehicles: Algorithms and experiments," in *Proc. Int. Symp. Exp. Robot.*, 2018, pp. 191–200.

[9] A. Hilal, "An intelligent sensor management framework for pervasive surveillance," Ph.D. dissertation, department of electrical and computer engineering, Univ. Waterloo, Waterloo, ON, Canada, 2013.

[10] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 410–426, Apr. 2012.

[11] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Comput.*, vol. 3, no. 4, pp. 24–33, Oct.–Dec. 2004.

[12] B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Maximum information bounds for planning active sensing trajectories," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4913–4920.

[13] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1271–1287, 2014.

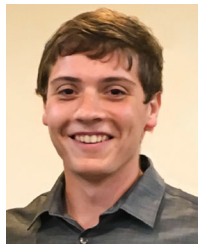
[14] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering." in *Proc. Robot.: Sci. Syst.*, 2019, pp. 22–26.

[15] M. Schwager, P. Dames, D. Rus, and V. Kumar, "A multi-robot control policy for information gathering in the presence of unknown hazards," in *Proc. Robot. Res.*, 2017, pp. 455–472.

[16] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Distributed attack-robust submodular maximization for multi-robot planning," in *2020 IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2020, pp. 2479–2485.

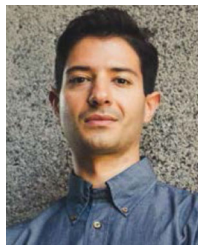
[17] J. L. Ny and G. J. Pappas, "On trajectory optimization for active sensing in Gaussian process models," in *Proc. 48th IEEE Conf. Decis. Control Held Jointly 28th Chin. Control Conf.*, 2009, pp. 6286–6292.

[18] W. Tabib, K. Goel, J. Yao, M. Dabhi, C. Boirum, and N. Michael, "Real-time information-theoretic exploration with Gaussian mixture model maps," in *Proc. Robot.: Sci. Syst.*, 2019, pp. 2–4.
[19] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic mapping using Cauchy-Schwarz quadratic mutual information," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 4791–4798.
[20] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 32–47, Jan. 2010.
[21] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar, "Resilient flocking for mobile robot teams," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1039–1046, Apr. 2017.
[22] A. Mitra, J. A. Richards, S. Bagchi, and S. Sundaram, "Resilient distributed state estimation with mobile agents: Overcoming byzantine adversaries, communication losses, and intermittent measurements," *Auton. Robots*, vol. 43, no. 3, pp. 743–768, 2019.
[23] A. Prorok, "Robust assignment using redundant robots on transport networks with uncertain travel time," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 2025–2037, Oct. 2020.
[24] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
[25] A. Krause and D. Golovin, "Submodular function maximization," in *Tractability: Practical Approaches to Hard Problems*, vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 2012, p. 19.
[26] J. B. Orlin, A. S. Schulz, and R. Udwani, "Robust monotone submodular function maximization," *Math. Programming*, vol. 172, no. 1, pp. 505–537, 2018.
[27] V. Tzoumas, K. Gatsis, A. Jadbabaie, and G. J. Pappas, "Resilient monotone submodular function maximization," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 1362–1367.
[28] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Resilient non-submodular maximization over matroid constraints," 2018, *arXiv:1804.01013*.
[29] M. Conforti and G. Cornuéjols, "Submodular set functions, matroids and the greedy algorithm," *Discrete Appl. Math.*, vol. 7, no. 3, pp. 251–274, 1984.
[30] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," *Math. Oper. Res.*, vol. 42, no. 4, pp. 1197–1218, 2017.
[31] B. Schlotfeldt, V. Tzoumas, D. Thakur, and G. J. Pappas, "Resilient active information gathering with mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4309–4316.
[32] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[33] U. Feige, "A threshold of $ln(n)$ for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.
[34] D. Sharma, A. Kapoor, and A. Deshpande, "On greedy maximization of entropy," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1330–1338.
[35] V. Tzoumas, N. A. Atanasov, A. Jadbabaie, and G. J. Pappas, "Scheduling nonlinear sensors for stochastic process estimation," in *Proc. Amer. Control Conf.*, 2017, pp. 580–585.
[36] L. F. Chamon, G. J. Pappas, and A. Ribeiro, "The mean square error in Kalman filtering sensor selection is approximately supermodular," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 343–350.
[37] J. B. Orlin, A. S. Schulz, and R. Udwani, "Robust monotone submodular function maximization," in *Proc. Int. Conf. Integer Program. Combinatorial Optim.*, 2016, pp. 312–324.
[38] S. T. Jawaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.
[39] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.
[40] N. Michael, E. Stump, and K. Mohta, "Persistent surveillance with a team of MAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 2708–2714.
[41] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Resilient monotone sequential maximization," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 7261–7268.



Brent Schlotfeldt (Student Member, IEEE) received the Bachelor of Science degree in electrical engineering and computer science from the University of Maryland, College Park, MD, USA, in 2016, and the Master of Science degree in robotics in 2017 from the University of Pennsylvania, Philadelphia, PA, USA, where he is currently working toward the Ph.D. degree in electrical and systems engineering.

His research interests include planning, control, and state estimation for robotic systems, with applications to autonomous driving and active sensing.



Vasileios Tzoumas (Member, IEEE) received the Diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2012, the Master of Arts degree in statistics from the Wharton School of Business, University of Pennsylvania, Philadelphia, PA, USA, in 2016, and the Master of Science degree in electrical engineering and the Ph.D. degree in electrical and systems engineering from the University of Pennsylvania, in 2016 and 2018, respectively.

He is currently an Assistant Professor with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA. He was with the Department of Aeronautics and Astronautics and with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, were he was a Research Scientist from 2019 to 2020 and a Postdoctoral Associate from 2018 to 2019. He was a Visiting Ph.D. Student with the Institute for Data, Systems, and Society, MIT, during 2017. He works on control, learning, and perception, as well as combinatorial and distributed optimization, with applications to robotics, cyber-physical systems, and self-reconfigurable aerospace systems. He aims for trustworthy collaborative autonomy. His work includes foundational results on robust and adaptive combinatorial optimization, with applications to multirobot information gathering for resiliency against robot failures and adversarial removals.

Dr. Tzoumas was a recipient of the Best Paper Award in Robot Vision at the 2020 IEEE International Conference on Robotics and Automation, and was a Best Student Paper Award Finalist at the 2017 IEEE Conference in Decision and Control.

George J. Pappas (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998.

He is currently the Joseph Moore Professor and Chair with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. He also holds a secondary appointment with the Department of Computer and Information Sciences and the Department of Mechanical Engineering and Applied Mechanics. He is a Member of the GRASP Lab and the PRECISE Center. He had served as the Deputy Dean for Research with the School of Engineering and Applied Science. His research interests include control theory and, in particular, hybrid systems, embedded systems, cyberphysical systems, and hierarchical and distributed control systems, with applications to unmanned aerial vehicles, distributed robotics, green buildings, and biomolecular networks.

Dr. Pappas was the recipient of various awards, such as the Antonio Ruberti Young Researcher Prize, the George S. Axelby Award, the Hugo Schuck Best Paper Award, the George H. Heilmeier Award, the National Science Foundation PECASE Award, and numerous Best Student Papers Awards.