Optimal Coupled Sensor Placement and Path Planning in Unknown Time-Varying Environments

Prakash Poudel* and Raghvendra V. Cowlagi†
Worcester Polytechnic Institute, Worcester, Massachusetts 01609

https://doi.org/10.2514/1.G009081

We address path planning for a mobile agent to navigate in an unknown environment with minimum exposure to a spatially and temporally varying threat field. The threat field is estimated using pointwise noisy measurements from a mobile sensor network. For this problem, we present a new information gain measure for optimal sensor placement that quantifies reduction in uncertainty in the path cost rather than the environment state. This measure, which we call the context-relevant mutual information (CRMI), couples the sensor placement and path-planning problem. We propose an iterative coupled sensor configuration and path-planning (CSCP) algorithm. At each iteration, this algorithm places sensors to maximize CRMI, updates the threat estimate using new measurements, and recalculates the path with minimum expected exposure to the threat. The iterations converge when the path cost variance, which is an indicator of risk, reduces below a desired threshold. We show that CRMI is submodular, and therefore greedy optimization provides near-optimal sensor placements while maintaining computational efficiency of the CSCP algorithm. Distance-based sensor reconfiguration costs are introduced in a modified CRMI measure, which we also show to be submodular. Through numerical simulations, we demonstrate that the principal advantage of this algorithm is that near-optimal low-variance paths are achieved using far fewer sensor measurements as compared to a standard sensor placement method.

Nomenclature

C	=	measurement model
$c(\boldsymbol{x}, t)$	=	threat intensity
H	=	entropy
$I^{\text{mod}}(\boldsymbol{q})$	=	modified context-relevant mutual information
$I(J; \boldsymbol{z}(\boldsymbol{q}))$	=	context-relevant mutual information
$I(\boldsymbol{\Theta}; \boldsymbol{z}(\boldsymbol{q}))$	=	standard mutual information
$J(\boldsymbol{v}), \hat{J}(\boldsymbol{v})$	=	true and estimated path cost
$\mathcal{K}$	=	path-relevant set
N_P	=	threat state dimension
N_S	=	number of sensors
N_G	=	number of grid points
$P_{\boldsymbol{\Theta}\boldsymbol{\Theta}}, P_{\boldsymbol{\Theta}z}, P_{zz}, P_{JJ}, P_{Jz}$	=	various estimation error covariances and cross-covariances
$\boldsymbol{q}$	=	sensor configuration
$\mathcal{R}_n^{\text{sup}}$	=	region of significant support
$\boldsymbol{v}$	=	path in grid topological graph
$\mathcal{W}$	=	compact 2D workspace
$\boldsymbol{x}$	=	Cartesian position coordinates
$\bar{\boldsymbol{x}}, a_n$	=	constants used in spatial basis functions
$\boldsymbol{z}$	=	measurement
δ	=	grid spacing
ε	=	coupled sensor configuration and path-planning termination threshold
$\boldsymbol{\eta}, R$	=	measurement noise and error covariance
$\boldsymbol{\Theta}(t), \hat{\boldsymbol{\Theta}}(t)$	=	true and estimated threat states
$\boldsymbol{\Phi}(\boldsymbol{x})$	=	spatial basis function
$\boldsymbol{\omega}, Q$	=	process noise and error covariance

Presented as Paper 2025-2069 at the 2025 AIAA SciTech Forum and Exposition, Orlando, FL, January 6–10, 2005; received 24 January 2025; accepted for publication 8 August 2025; published online 24 October 2025.

*Graduate Research Assistant, Aerospace Engineering Department, 100 Institute Road; ppoudel@wpi.edu. Student Member AIAA.

†Associate Professor, Aerospace Engineering Department, 100 Institute Road; rvcowlagi@wpi.edu. Senior Member AIAA (Corresponding Author).

I. Introduction

WE CONSIDER scenarios where a mobile agent navigating in an unknown environment can leverage measurements collected by a network of spatially distributed sensors. The unknown environment may include various adverse attributes, which we abstractly represent by a spatiotemporally varying scalar field and refer to as the *threat field*. The threat field represents unfavorable areas such as those associated with various natural or artificial phenomena, such as wildfires, harmful gases in the atmosphere, or the perceived risk of adversarial attack.

Figure 1 provides a motivating example of a time-varying flood map based on data collected during Hurricane Harvey at spatially distributed gauge stations over 15-min time intervals [1]. Initially, there is low water discharge with no signs of flooding in Fig. 1a. Significant flooding is visible in Fig. 1c at $t = 1500$ min, which subsequently recedes. The changing flood levels may be considered as a spatiotemporally varying "threat" to which exposure of, say, an emergency first-response vehicle, should be minimized.

We address the problem of path planning with minimum threat exposure in such an environment. Because the environment is unknown, an important task is to place the sensors in appropriate locations, which is called the *sensor placement* problem, or more generally, the sensor *configuration* problem. If we had at our disposal an abundance of sensors and computational resources to process large amounts of sensor data, then the placement/configuration problem would be trivial. We would simply place sensors to ensure maximum area coverage.

In practical applications, however, sensor networks may be constrained by size as well as energy usage. In the flood evolution example, a sensor network of unmanned aerial vehicles (UAVs) may be deployed for real-time surveillance of flooding. Due to cost and battery limitations, it may not be possible to achieve full area coverage quickly enough to inform the actions of a ground robot to safely navigate the environment. This situation exemplifies the broader problem of path planning with a *minimal* number of sensor measurements and, in turn, highlights the need for optimal sensor configuration *in the context* of path planning. This problem lies at the intersection of several research areas, including estimation, path planning, and sensor placement, which we briefly review next. We note that the problem of interest here is quite different from the simultaneous localization and mapping (SLAM) problem, where the path planning and sensing are coupled due to the assumption of fully

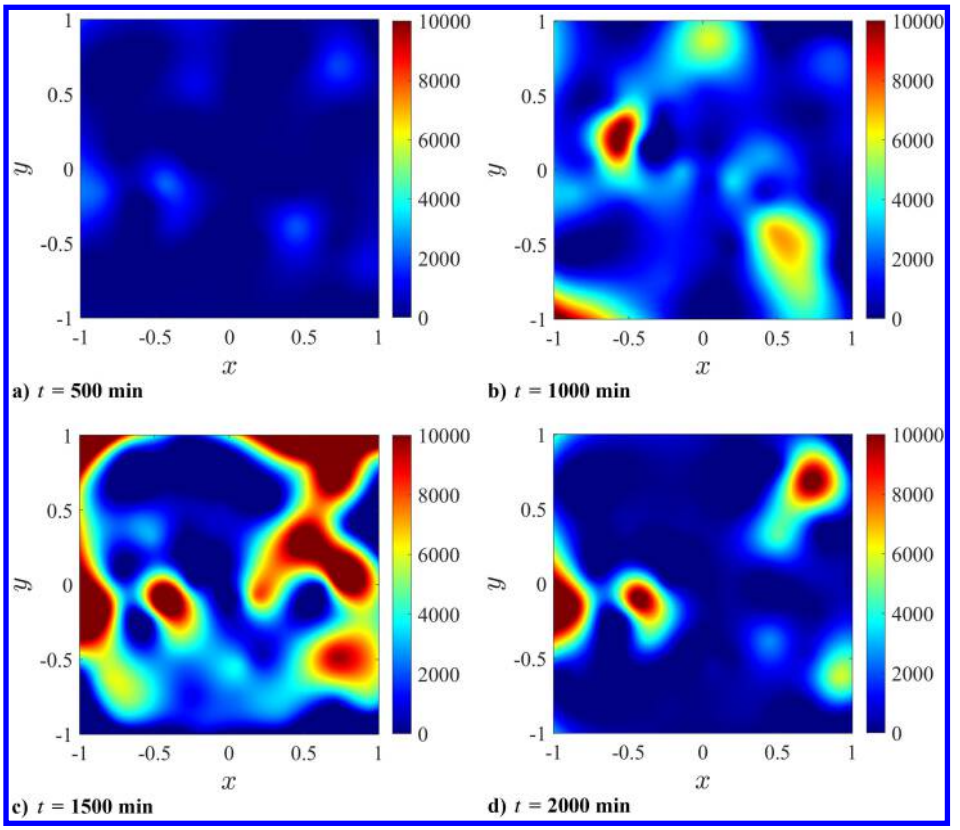

Fig. 1 Visualization of the flood map at different times (minutes).

onboard sensing. By contrast, we consider a distributed sensor network separate from the mobile agent.

Of these areas, perhaps estimation is the most mature [2]. The literature on estimation involves different techniques, including the Kalman filter, the maximum likelihood estimator [2], and the Bayesian filter [3]. Application of the extended Kalman filter (EKF), the unscented Kalman filter (UKF) [4], or the particle filter [5] is common for nonlinear dynamic systems. Several data-driven estimation techniques have been employed for spatiotemporal modeling of hazardous regions, particularly in robotics, environmental monitoring, and surveillance. These include a supervised learning approach using Gaussian process regression for threat field estimation [6], a statistical generalized additive model for landslide hazard estimation [7], and a combined approach using Bayesian inference and random forest for the spatial prediction of wildfires [8].

Path planning and motion planning are similarly mature areas of research. Generally, path planning under uncertainty involves finding paths that minimize the expected cost. Classical approaches to path planning include cell decomposition, probabilistic roadmaps, and artificial potential field techniques [9,10]. Dijkstra's algorithm, A*, and its variants are branch-and-bound optimization algorithms that leverage heuristics to effectively steer the path search toward the goal. While classical path-planning methods are powerful, they are inherently limited by the accuracy of the environment's available information. An accurate representation of the environment is difficult if the environment's states or dynamics are unknown. Modern approaches to path planning leverage advanced methodologies such as adaptive informative path planning [11], coverage path planning [12], and informed sampling-based path planning [13]. More recently, learning-based techniques, particularly deep reinforcement learning [14–17] and fuzzy logic [18], are reported for addressing environmental uncertainty. In risk-aware path planning, the objective is not only to find a feasible or optimal path but also to minimize exposure to uncertain or hazardous regions [19]. Recent risk-aware path-planning techniques integrate reinforcement learning with failure mode and effect analysis to ensure safe and complete coverage in hazardous environments [20].

Different sensor placement approaches have been employed depending on the type of application and parameters that need to be measured. Greedy approaches based on information-based metrics are presented in [21–23]. Machine-learning-based sensor placement techniques are reported for efficient sensing with a minimal number of sensors and measurements as possible [24,25]. Information-theory-based sensor placement techniques utilize performance metrics such as the Fisher information matrix (FIM) [26], entropy [27], Kullback–Leibler (KL) divergence [28], mutual information [29], and frame potential [21] to maximize the amount of valuable information gathered from the surrounding environment. Similarly, in [30], the authors utilize two information measures: one associated with mutual information based on object detection and another with mutual information based on classification of the detected objects. With all these performance metrics, the intention is to maximally reduce some quantification of the uncertainty. More recently, a sensing and path-planning method based on reinforcement learning has been reported, which evaluates performance using a technique called proximal policy optimization (PPO) [31]. In robotics and aerospace applications, *active sensing* plays a crucial role in linking perception and action, enabling systems to gather meaningful information that supports intelligent decision-making. Some examples of active sensing include information-driven or cooperative active sensing [32–34], uncertainty-aware active sensing [35], and active sensing using machine learning [36,37].

The problem of minimizing sensor reconfiguration costs, such as the distance traveled by mobile sensors, is commonly studied in the mobile sensor network literature, but less so in the context of sensor placement. Reconfiguration becomes an important issue when multiple iterations of sensor configuration and estimation are conducted, which in turn may be necessary when the number of sensors is small. Some examples include consideration of the reconfiguration cost of the sensor network topology [38] or the total energy consumption of the sensor network [39,40].

In this paper we consider the problem of optimal sensor placement coupled with path planning in an unknown dynamic environment. Specifically, we are interested in sensor placement to collect information of most relevance to the path-planning problem. The

objective is to find a near-optimal path with high confidence, i.e., low variance in the path cost, with a minimal number of sensor measurements. We aim to compare such a coupled sensor configuration and path-planning (CSCP) method against decoupled methods, where sensor configuration is achieved by optimizing a metric that does not consider the path-planning problem in any way. This is a relatively new research problem. Prior works by the second author and coworkers address this problem for static (i.e., time-invariant) environments. A heuristic task-driven sensor placement approach called the interactive planning and sensing (IPAS) for static environments is reported in [41]. The IPAS method outperforms several decoupled sensor placement methods in terms of the total number of measurements needed to achieve near-optimal paths. Sensor configuration for location and field-of-view is reported in [6], also for static fields. Sensor placement for multi-agent path planning based on entropy reduction is presented in [42].

The novelty of this work is that we consider a time-varying threat field and provide a new sensor placement method. Specifically, we develop the so-called context-relevant mutual information (CRMI), which quantifies the amount of information in configuration-dependent sensor measurements in the context of reducing uncertainty in path cost, rather than the environment state estimation error (as a decoupled method typically does). We develop an iterative algorithm for CSCP. At each iteration, a threat estimate is first computed using sensor measurements. Next, a path-planning algorithm finds a path of minimum expected threat. Next, optimal sensor placements are computed to maximize the path-dependent CRMI, and the iterations repeat. We compare this CSCP-CRMI method to a decoupled method that finds optimal sensor placement by maximizing the "standard" mutual information (SMI). The metric of comparison is based on the number of measurements needed to achieve a path cost with variance no greater than a user-specified threshold. The results of this comparison show that the CSCP-CRMI method significantly outperforms the decoupled method. We further extend the CSCP method to incorporate sensor reconfiguration costs into the cost function. The objective is to collectively maximize the path-dependent CRMI and minimize sensor movement. A comparison is performed between the results of the CSCP methods, one ignoring and the other incorporating the sensor reconfiguration cost.

Preliminary results from this work were recently discussed in conference papers [43,44]. In this paper we further extend the conference paper works by introducing an approximation algorithm based on the greedy placement of sensors. Additionally, we show the submodularity property of both the CRMI and the modified CRMI and demonstrate that the greedy placement of sensors guarantees a near-optimal solution. A proof of convergence of the CSCP algorithm is also discussed.

The rest of the paper is organized as follows: In Sec. II, we introduce the elements of the problem. In Sec. III, we present and analyze the new CRMI measure and the CSCP-CRMI algorithm, followed by a discussion on reconfiguration costs and greedy optimization for near-optimal sensor placement. We present numerical simulation results in Sec. IV and conclude the paper in Sec. V.

II. Problem Formulation

Let $\mathbb{R}, \mathbb{N}$ denote the sets of real and natural numbers, respectively. We denote by $\{N\}$ the set $\{1, 2, \ldots, N\}$ and by $\boldsymbol{I}_N$ the identity matrix of size N, for any $N \in \mathbb{N}$.

Consider a closed square region denoted by $\mathcal{W} \subset \mathbb{R}^2$ and referred to as the *workspace*, within which the mobile agent (called the *actor*) and a network of mobile sensors operate. In this workspace, consider a grid consisting of N_G uniformly spaced points. The coordinates of these points in a prespecified Cartesian coordinate axis system are denoted by $\boldsymbol{x}_i = (x_i, y_i)$, for each $i \in N_G$. The distance between the adjacent grid points is denoted by δ. The mobile agent traverses grid points according to the "4-way adjacency rule," such that the adjacent points are top, down, left, and right. Furthermore, we assume a constant speed such that the actor's transitions to adjacent grid points occur in a constant time step Δt.

We formulate the path-planning problem for the actor as a graph search problem on a graph, $\mathcal{G} = (V, E)$, with $V = \{N_G\}$ such that each vertex in V is uniquely associated with a grid point. The set of edges E in this graph consists of pairs of grid points that are geometrically adjacent to each other.

A *threat field*, denoted as $c : \mathcal{W} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$, is a time-varying scalar field that takes strictly positive values, indicating regions with higher intensity that are potentially hazardous and unfavorable. A path between two prespecified initial and goal vertices, $i_s, i_g \in V$, is defined as a finite sequence $\boldsymbol{v} = \{v_0, v_1, \ldots, v_L\}$ of successively adjacent vertices. This sequence starts at the initial vertex $v_0 = i_s$ and ends at the goal vertex $v_L = i_g$, where $L \in \mathbb{N}$ represents the number of vertices in the sequence. The edge transition costs, which account for the expenses incurred when an actor moves between vertices in a graph, are determined by a scalar function $g : E \to \mathbb{R}_{>0}$. This function assigns a value to each edge in the graph, representing the associated cost or effort required for traversal, and is defined as

$$g((i,j),t) = c(\boldsymbol{x}_j, t), \quad \text{for} \quad i, j \in \{N_G\}, \quad (i,j) \in E \tag{1}$$

The cost $J(\boldsymbol{v}) \in \mathbb{R}_{>0}$ indicates the total threat exposure for an actor on its traversal along a path $\boldsymbol{v}$ and is defined as the sum of edge transition costs, $J(\boldsymbol{v}) = \delta \sum_{\ell=1}^{L} g((v_{\ell-1}, v_\ell), \ell\Delta t)$. The main problem of interest is to find a path $\boldsymbol{v}^*$ with minimum cost. Because the threat field is unknown and time-varying, its estimation is essential. A network of N_S sensors, where $N_S \ll N_G$, can be used to measure the intensity of threat. These sensor measurements are denoted $\boldsymbol{z}(\boldsymbol{x}, t; \boldsymbol{q}) = \{z_1(\boldsymbol{x}, t; q_1), z_2(\boldsymbol{x}, t; q_2), \ldots, z_{N_s}(\boldsymbol{x}, t; q_{N_s})\}$. Sensors are placed at distinct grid points, and the set of these grid points is called the *sensor configuration*, $\boldsymbol{q} = \{q_1, q_2, \ldots, q_{N_s}\} \subset \{N_G\}$.

The threat field is considered to be a stochastic quantity with a predictive model involving uncertainty. Specifically, we consider a finite parameterization $c(\boldsymbol{x}, t) \coloneqq 1 + \sum_{n=1}^{N_P} \theta_n(t)\phi_n(\boldsymbol{x}) = 1 + \boldsymbol{\Phi}^\top(\boldsymbol{x})\boldsymbol{\Theta}(t)$, with $\boldsymbol{\Phi}(\boldsymbol{x}) \coloneqq [\phi_1(\boldsymbol{x}) \ldots \phi_{N_P}(\boldsymbol{x})]^\top$, and $\phi_n(\boldsymbol{x}) \coloneqq \exp(-(\boldsymbol{x} - \bar{\boldsymbol{x}}_n)^\top(\boldsymbol{x} - \bar{\boldsymbol{x}}_n)/2a_n)$ representing the basis functions for each $n \in \{N_P\}$. Here, N_P denotes the number of parameters (or bases), representing the *threat state*, and a_n and $\bar{\boldsymbol{x}}_n$ are constants. The locations of the basis functions $\boldsymbol{\Phi}$ are fixed and remain unchanged throughout the entire sensing and path-planning process. Although the actor has prior knowledge of the functional forms of these functions, the threat parameter $\boldsymbol{\Theta}(t) \coloneqq [\theta_1(t) \ldots \theta_{N_P}(t)]^\top$ is unknown and must be estimated. The values of the constants $a_n \in \mathbb{R}_{>0}$ and $\bar{\boldsymbol{x}}_n \in \mathcal{W}$ are prespecified and chosen in such a manner that the combined interiors of the significant support regions cover $\mathcal{W}$. The parameter a_n is chosen to minimize the overlap between the basis functions, ensuring better distinction and independence among them [41]. In general, the basis functions should be chosen to approximate threat field data, e.g., the flood map data in Fig. 1.

The temporal evolution of the threat is modeled by $\dot{\boldsymbol{\Theta}}(t) = A_c\boldsymbol{\Theta}(t) + \boldsymbol{\omega}(t)$, where $\boldsymbol{\omega}(t) \sim \mathcal{N}(0, Q_c)$ is white process noise with $Q_c \coloneqq \sigma_P^2 \boldsymbol{I}_{N_P}$. Such a model may be available either from an underlying physical model of the threat, or it may be derived from data, or a combination of both. As an illustrative example, the solution to a heat diffusion equation, $(\partial c/\partial t) = \alpha[(\partial^2 c/\partial x^2) + (\partial^2 c/\partial y^2)]$ can be approximated by $c(\boldsymbol{x}, t) = 1 + \boldsymbol{\Phi}^\top(\boldsymbol{x})\boldsymbol{\Theta}(t)$ such that $\boldsymbol{\Theta}(t)$ satisfies $\dot{\boldsymbol{\Theta}}(t) = \alpha(\boldsymbol{\Phi}^\top/|\boldsymbol{\Phi}|^2)\nabla^2\boldsymbol{\Phi}\boldsymbol{\Theta}(t)$, where $A_c \coloneqq \alpha(\boldsymbol{\Phi}^\top/|\boldsymbol{\Phi}|^2)\nabla^2\boldsymbol{\Phi}$.

We restrict the scope of this paper to linear threat field dynamics for the purpose of establishing a proof of convergence of the proposed method. However, the method itself is not limited to linear dynamics. We implement the unscented Kalman filter in our current threat field estimator in anticipation of this future extension. Our current CSCP implementation will work as-is for nonlinear threat dynamics, but the CRMI calculations will be approximations. Other works in the literature, e.g., [45], have studied MI calculations for nonlinear systems, which we can easily adapt to CRMI in the future.

Discretization in time of this model is easily accomplished by the series expansion $A \coloneqq \boldsymbol{I}_{N_P} + A_c\Delta t + [(A_c)^2(\Delta t)^2/2!] + \ldots$ and $Q \coloneqq Q_c\Delta t + [(A_cQ_c + Q_cA_c^\top)(\Delta t)^2/2!] + \ldots$ terminated at a desired order [2]. The discretized system dynamics are

$$\boldsymbol{\Theta}_k = A\boldsymbol{\Theta}_{k-1} + \boldsymbol{\omega}_{k-1} \tag{2}$$

where $\boldsymbol{\omega}_{k-1} \sim \mathcal{N}(0, Q)$ for each $k \in \mathbb{N}$.

The measurements obtained from each sensor are modeled as

$$z_k \coloneqq c(\boldsymbol{x}_{q_k}, t) + \boldsymbol{\eta}_k = C_k(\boldsymbol{q})\boldsymbol{\Theta}_k + \boldsymbol{\eta}_k \tag{3}$$

$$\text{where} \quad C_k(\boldsymbol{q}) \coloneqq [\boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{q}_{k,1}}) \quad \boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{q}_{k,2}}) \quad \dots \quad \boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{q}_{k,N_S}})]^\top \tag{4}$$

and $\boldsymbol{\eta}_k \sim \mathcal{N}(0, R)$ is zero-mean measurement noise with covariance $R \succ 0$.

We generate stochastic estimates of the threat state with mean value $\hat{\boldsymbol{\Theta}}_k$ and estimation error covariance P. For any path, $\boldsymbol{v} = \{v_0, v_1, \dots, v_L\}$ in $\mathcal{G}$, the cost of the path is

$$J(\boldsymbol{v}) \coloneqq \delta \sum_{\ell=1}^{L} c(\boldsymbol{x}_{\boldsymbol{v}_\ell}, t) = \delta\left(L + \sum_{\ell=1}^{L} \boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})\boldsymbol{\Theta}_\ell\right) \tag{5}$$

Here, $\boldsymbol{\Theta}_\ell$ represent the threats at different iterations (or time steps) equivalent to the iterations required for an actor to follow the path $\boldsymbol{v}$. The cost J becomes a random variable with distribution dependent on $\boldsymbol{\Theta}$. If $\boldsymbol{\Theta}$ is Gaussian, then J is also Gaussian because it is linearly dependent on $\boldsymbol{\Theta}$. Let $\mathcal{V} \subset \mathcal{W}$ be the set of grid point locations in the workspace associated with each v_ℓ in the path $\boldsymbol{v}$. Note that while $\theta_n(t)$ and ϕ_n represent the total number of threat parameters and the basis functions involved in the threat field generation, $\theta_m(t)$ and ϕ_m denote the subset of threat parameters and associated basis functions that lie within the path and are relevant for computing the path cost $J(\boldsymbol{v})$. The set $\mathcal{K} \coloneqq \{m \in \{N_P\} : \mathcal{V} \cap \mathcal{R}_m^{\text{sup}} \neq \varnothing\}$ is defined as the path-relevant set of threat states for any path $\boldsymbol{v}$. In other words, $\mathcal{K}$ consists of the set of indices m, representing the threat parameters θ_m or their corresponding basis functions ϕ_m, such that the path lies within the total region of significant support $\mathcal{R}_m^{\text{sup}}$ defined by ϕ_m.

An important characteristic associated with the convergence of the path-planning algorithm is the *risk* of the path denoted by $\rho(\boldsymbol{v})$. For a Gaussian J, the risk of the path $\boldsymbol{v}$ is defined as $\rho(\boldsymbol{v}) \coloneqq \hat{J}(\boldsymbol{v}) + \sqrt{\text{Var}[J(\boldsymbol{v})]}$ [46]. Since $N_S \ll N_G$, it is not possible to obtain good estimates with only one set of measurements, and it is required to take measurements repeatedly over a finite number of iterations.

Problem 1: For a prespecified termination threshold $\varepsilon > 0$ and some finite iterations $k = 0, 1, \dots, M$, find a sequence of sensor configurations $\boldsymbol{q}_k^*$ and a path $\boldsymbol{v}^*$ with minimum expected cost $\hat{J}(\boldsymbol{v}^*)$ and such that $\text{Var}[J(\boldsymbol{v}^*)] \leqslant \varepsilon$.

Alternatively, one may consider a requirement that the path risk $\rho(\boldsymbol{v}^*)$ be lower than a prespecified threshold.

III. Coupled Sensing and Planning

Coupled sensor configuration and path planning (CSCP) is our proposed iterative approach to solve Problem 1. At each iteration, a sensor configuration is determined, and measurements of the threat field are collected. The optimal sensor configuration is found by maximizing an information measure that we call *context-relevant mutual information* (CRMI). These sensor measurements are used to update the threat field estimates in an estimator. For future compatibility with nonlinear threat dynamics, an unscented Kalman filter (UKF) is used for estimating the parameter $\boldsymbol{\Theta}$. The interested reader is referred to Appendix B for a brief description of the UKF. Next, the path plan is modified based on the new threat field estimate, and this process continues until the path cost variance is reduced below a prespecified threshold ε.

In this paper, we assume that the actor is a *planning agent* that does not move before the CSCP algorithm is complete. In other words, we may think of this implementation of CSCP as occurring in a virtual world for planning the future movements of the actor and sensors. The reason for this restriction in scope is to be able to provide a proof of convergence of the CSCP method as a baseline for future applications. In our recent work [47], we implement a version of CSCP where the actor moves simultaneously with the CSCP iterations, and the sensors need a finite nonzero duration to relocate. The simulations in [47] demonstrate a successful implementation and provide sensor resource benefits similar to those described in the present manuscript, pending a formal proof of convergence.

In what follows, we provide details of this iterative process, analysis, and an illustrative example.

A. Context-Relevant Mutual Information (CRMI)

For any time step k, the *mutual information (MI)* between the state $\boldsymbol{\Theta}_k$ and measurement z_k random variables is defined as [48]

$$I(\boldsymbol{\Theta}_k; z_k) \coloneqq \iint p(\boldsymbol{\Theta}_k, z_k) \log\left(\frac{p(\boldsymbol{\Theta}_k, z_k)}{p(\boldsymbol{\Theta}_k)p(z_k)}\right) \mathrm{d}\boldsymbol{\Theta}_k \, \mathrm{d}z_k$$

where $p(\boldsymbol{\Theta}_k)$, $p(z_k)$, and $p(\boldsymbol{\Theta}_k, z_k)$ represent the probability density functions (PDFs) of state, measurement, and a joint PDF of state and measurement, respectively. The joint PDF $p(\boldsymbol{\Theta}_k, z_k)$ is a multivariate normal distribution with mean $(\hat{\boldsymbol{\Theta}}_{k|k-1}, \hat{z}_k)$ and covariance

$$\begin{bmatrix} P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}} & P_{\boldsymbol{\Theta}z_{k|k-1}} \\ P_{\boldsymbol{\Theta}z_{k|k-1}}^\top & P_{zz_{k|k-1}} \end{bmatrix}$$

Here, $P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}$ is obtained from the UKF algorithm. The covariance of the measurement random vector $P_{zz_{k|k-1}}$ and cross covariance between the state and measurement random vectors $P_{\boldsymbol{\Theta}z_{k|k-1}}$ depend on the sensor configuration $\boldsymbol{q}$. At each grid point, these covariances are determined as

$$\begin{aligned} P_{zz_{k|k-1}} &\coloneqq \mathbb{E}\big[(z_k - \hat{z}_k)(z_k - \hat{z}_k)^\top\big] \\ &= \mathbb{E}\big[\big(C_k(\boldsymbol{q})(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1}) + (\boldsymbol{\eta}_k - \hat{\boldsymbol{\eta}}_k)\big)\big(C_k(\boldsymbol{q})(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1}) \\ &\quad + (\boldsymbol{\eta}_k - \hat{\boldsymbol{\eta}}_k)\big)^\top\big] \\ &= C_k(\boldsymbol{q})\mathbb{E}\big[(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1})(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1})^\top\big]C_k^\top(\boldsymbol{q}) \\ &\quad + \mathbb{E}\big[(\boldsymbol{\eta}_k - \hat{\boldsymbol{\eta}}_k)(\boldsymbol{\eta}_k - \hat{\boldsymbol{\eta}}_k)^\top\big] \\ &= C_k(\boldsymbol{q})P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}C_k^\top(\boldsymbol{q}) + R_k \end{aligned} \tag{6}$$

$$\begin{aligned} P_{\boldsymbol{\Theta}z_{k|k-1}} &\coloneqq \mathbb{E}\big[(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1})(z_k - \hat{z}_k)^\top\big] \\ &\coloneqq \mathbb{E}\big[(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1})\big(C_k(\boldsymbol{q})(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1}) + (\boldsymbol{\eta}_k - \hat{\boldsymbol{\eta}}_k)\big)^\top\big] \\ &= \mathbb{E}\big[(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1})(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1})^\top\big]C_k^\top(\boldsymbol{q}) \\ &\quad + \mathbb{E}\big[(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_{k|k-1})(\boldsymbol{\eta}_k - \hat{\boldsymbol{\eta}}_k)^\top\big] \\ &= P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}C_k^\top(\boldsymbol{q}) \end{aligned} \tag{7}$$

The mutual information between the state $\boldsymbol{\Theta}_k$ and measurement $z_k(\boldsymbol{q})$ is then written as [45]:

$$I(\boldsymbol{\Theta}_k; z_k(\boldsymbol{q})) = \frac{1}{2}\log\left(\frac{|P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}|}{|P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}} - P_{\boldsymbol{\Theta}z_{k|k-1}}P_{zz_{k|k-1}}^{-1}P_{\boldsymbol{\Theta}z_{k|k-1}}^\top|}\right) \tag{8}$$

The MI $I(\boldsymbol{\Theta}_k; z_k(\boldsymbol{q}))$ depends on the sensor configuration $\boldsymbol{q}$, and in that sense it quantifies "informativeness" of a configuration $\boldsymbol{q}$. A canonical method of finding the optimal sensor configuration is to maximize the MI (I) over $\boldsymbol{q}$. Note, however, that this MI $I(\boldsymbol{\Theta}_k; z_k(\boldsymbol{q}))$ is entirely decoupled from the path-planning problem, in that it is in no way contextualized by the optimal path to be found. We introduce a new information measure called the context-relevant mutual information (CRMI), which couples the sensor configuration (placement) and path-planning problems.

We define CRMI as the mutual information between the path cost and the measurements. CRMI may be thought of as a formalization of the vague notion "sensors 'near' the planned path are more informative than those farther away." To this end, consider that, for any path $\boldsymbol{v}$, the expected (mean) cost is $\hat{J}(\boldsymbol{v}) \coloneqq \delta(L + \sum_{\ell=1}^{L} \boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})\hat{\boldsymbol{\Theta}}_\ell)$. The joint PDF $\boldsymbol{p}(J_k, z_k)$ between the path cost and measurement variables is

$$\boldsymbol{p}(J_k, z_k) = \mathcal{N}\left(\begin{bmatrix} J_k \\ z_k \end{bmatrix} : \begin{bmatrix} \hat{J}_{k|k-1} \\ \hat{z}_k \end{bmatrix}, \begin{bmatrix} P_{JJ_{k|k-1}} & P_{Jz_{k|k-1}} \\ P_{Jz_{k|k-1}}^\top & P_{zz_{k|k-1}} \end{bmatrix}\right)$$

The variance of the path cost is

$$P_{JJ_{k|k-1}} \coloneqq \mathbb{E}\left[(J(\boldsymbol{v}) - \hat{J}(\boldsymbol{v}))^2\right] = \mathbb{E}\left[\left(\delta \sum_{\ell=1}^{L} \boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)\right)^2\right]$$

Using the formula for square of sums, $\left(\sum_i a_i\right)^2 = \sum_i a_i^2 + 2\sum_{i<j} a_i a_j$,

$$\begin{aligned} P_{JJ_{k|k-1}} &= \delta^2 \Bigg(\mathbb{E}\Bigg[\sum_{\ell=1}^{L} (\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell))^2 \\ &\quad + 2 \sum_{\substack{\ell<m \\ \ell,m\in\{L\}}}^{L} (\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_m})(\boldsymbol{\Theta}_m - \hat{\boldsymbol{\Theta}}_m)) \Bigg] \Bigg) \\ &= \delta^2 \Bigg(\mathbb{E}\Bigg[\sum_{\ell=1}^{L} (\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell))^2 \\ &\quad + 2 \sum_{\substack{\ell<m \\ \ell,m\in\{L\}}}^{L} (\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)(\boldsymbol{\Theta}_m - \hat{\boldsymbol{\Theta}}_m)^\top \boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_m})) \Bigg] \Bigg) \end{aligned}$$

The first term $\mathbb{E}\left[\sum_{\ell=1}^{L} (\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell))^2\right]$ can be further expressed as

$$\begin{aligned} P_{JJ_{k|k-1}} &= \sum_{\ell=1}^{L} \mathbb{E}\left[\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)^\top \boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_\ell})\right] \\ &= \sum_{\ell=1}^{L} \boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell}) \mathbb{E}\left[(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)^\top\right] \boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_\ell}) \\ &= \sum_{\ell=1}^{L} \boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell}) P_{k_\ell} \boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_\ell}) \end{aligned}$$

Therefore,

$$\begin{aligned} P_{JJ_{k|k-1}} &= \delta^2 \sum_{\ell=1}^{L} (\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell}) P_{k_\ell} \Phi(\boldsymbol{x}_{\boldsymbol{v}_\ell})) \\ &\quad + 2\delta^2 \sum_{\substack{\ell<m \\ \ell,m\in\{L\}}}^{L} (\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell}) P_{k_{\ell m}} \Phi(\boldsymbol{x}_{\boldsymbol{v}_m})) \end{aligned} \tag{9}$$

The calculation of $P_{JJ_{k|k-1}}$ requires the determination of $\boldsymbol{\Phi}$ and the error covariance P for every grid point $\boldsymbol{v}_l$ lying on the path. P_{k_ℓ} and $P_{k_{\ell m}}$ are determined by propagating the UKF prediction steps for iterations equivalent to the path length. The covariance of the measurement and the cross covariance between the path cost and the measurement random vector are formulated as

$$\begin{aligned} P_{Jz_{k|k-1}} &= \mathbb{E}[(J(\boldsymbol{v}) - \hat{J}(\boldsymbol{v}))(z_k - \hat{z}_k)^\top] \\ &= \mathbb{E}\Bigg[\Bigg(\delta \sum_{\ell=1}^{L} \boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)\Bigg)(C_k(\boldsymbol{q})(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_k) \\ &\quad + (\boldsymbol{\eta}_k - \hat{\boldsymbol{\eta}}_k))^\top\Bigg] \\ &= \delta\mathbb{E}\left[\left(\sum_{\ell=1}^{L} \boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell})(\boldsymbol{\Theta}_\ell - \hat{\boldsymbol{\Theta}}_\ell)(\boldsymbol{\Theta}_k - \hat{\boldsymbol{\Theta}}_k)^\top C_k^\top(\boldsymbol{q})\right)\right] \\ &= \delta \sum_{\ell=1}^{L} (\boldsymbol{\Phi}^\top(\boldsymbol{x}_{\boldsymbol{v}_\ell}) P_{k_\ell}) C_k^\top(\boldsymbol{q}) \end{aligned} \tag{10}$$

$$P_{zz_{k|k-1}} = \mathbb{E}[(z_k - \hat{z}_k)(z_k - \hat{z}_k)^\top] = C_k(\boldsymbol{q}) P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}} C_k^\top(\boldsymbol{q}) + R_k \tag{11}$$

Finally, the CRMI is calculated as

$$I(J_k; z_k(\boldsymbol{q})) = \frac{1}{2}\log\left(\frac{|P_{JJ_{k|k-1}}|}{|P_{JJ_{k|k-1}} - P_{Jz_{k|k-1}} P_{zz_{k|k-1}}^{-1} P_{Jz_{k|k-1}}^\top|}\right) \tag{12}$$

This definition of the CRMI is the critical step in the proposed coupled sensor placement and path planning (CSCP) algorithm, described next. Similar to MI, CRMI is the difference between entropy and conditional entropy given sensor measurements, namely

$$I(J; z(\boldsymbol{q})) = H(J) - H(J|z(\boldsymbol{q})) \tag{13}$$

where $H(J)$ and $H(J|z(\boldsymbol{q}))$ denote the entropy of J and its entropy conditional given sensor measurements $z(\boldsymbol{q})$.

B. CSCP Algorithm

The CSCP algorithm described in Algorithm 1 initializes with $\hat{\boldsymbol{\Theta}}_0 = \boldsymbol{0}$ and $P_0 = \chi \boldsymbol{I}_{N_p}$, where χ is a large arbitrary number. The initial sensor placement $\boldsymbol{q}_0$ is arbitrary. At the initial iteration, an optimal path $\boldsymbol{v}_0^*$ of minimum expected cost $\mathbb{E}[J(\boldsymbol{v}_0^*)]$ is calculated. This threat estimate initialization implies that until a measurement is taken, the algorithm assumes all threat states to be zero. As a result, the CSCP planner is "optimistic" in that it plans paths through regions associated with threat states that were not previously measured or estimated. The CRMI-based sensor placement then ensures sensors are placed close to this path, thereby ensuring that the sensors explore the entire workspace.

The description in Algorithm 1 is quite general, and its various steps can be implemented using different methods of the user's choice. At each iteration k, the algorithm calculates the variance $\mathrm{Var}[J(\boldsymbol{v}_k^*)]$ of the cost of the path $\boldsymbol{v}_k^*$ per Eq. (9). The algorithm terminates whenever the variance of the path cost reduces below a prespecified threshold $\varepsilon > 0$. The method of computation of the optimal path $\boldsymbol{v}_k^*$ is the user's choice: for most practical applications, Dijkstra's algorithm (our choice for implementation) or the A* algorithm will suffice.

The optimal sensor configuration in line 6 can be calculated by optimizing some measure of information gain. In a *decoupled* approach, we may optimize the standard MI in Eq. (8). In the proposed CSCP method, we optimize the CRMI in Eq. (12). The method of optimization is left to the user, and it is not the focus of this paper. For a small number of grid points, we can determine $\boldsymbol{q}_k^*$ by mere enumeration. In prior works, e.g., [6], we have found success in implementing evolutionary global optimization methods for sensor configuration, albeit using a different reward function. It is possible to apply such methods for CRMI maximization as well.

With an optimal sensor placement, a new set of measurements is recorded, which is then used to update the state estimate for $\boldsymbol{\Theta}$. Yet again, the specific method of estimation is the user's choice. We choose the linearization-free UKF method for future applications to nonlinear threat dynamics, briefly described in the Appendix. This iterative process continues until the termination criteria $\mathrm{Var}[(\hat{J}(\boldsymbol{v})] \leqslant \varepsilon$ is satisfied.

Algorithm 1: Coupled sensor configuration and planning (CSCP)

Set $k = 0, \hat{\boldsymbol{\Theta}}_0 = \mathbf{0}$, and $P_0 = \chi \boldsymbol{I}_{N_P}$;
Initialize sensor placement $\boldsymbol{q}_0 \subset \{N_G\}$;
Find $\boldsymbol{v}_0^* = \arg\min(\hat{J}_0(\boldsymbol{v}))$;
while $\mathrm{Var}[J(\boldsymbol{v}_k^*)] > \varepsilon$ **do**
  Determine $I(J_k; z_k(\boldsymbol{q}))$ per Eq. (12);
  Find optimal sensor configuration $\boldsymbol{q}_k^* \coloneqq \arg\max_{\boldsymbol{q}} I(J_k; z_k(\boldsymbol{q}))$;
  Obtain new sensor measurements $z_k(\boldsymbol{q}_k^*)$;
  Update $\hat{\boldsymbol{\Theta}}_k, P_k$;
  Find $\boldsymbol{v}_k^* \coloneqq \arg\min(\hat{J}_k(\boldsymbol{v}))$;
  Increment iteration counter $k = k + 1$.

C. CRMI Optimization

Finding the optimal sensor configuration by maximizing CRMI is challenging because the number of feasible sensor configurations suffers combinatorial explosion with an increasing number of sensors N_S. To resolve this issue, we execute greedy optimization of one sensor at a time, which enormously reduces the computation time. When the objective function is submodular, the suboptimality due to greedy optimization remains bounded. Therefore, we consider submodularity of the proposed CRMI.

A brief definition of submodularity is as follows: Consider a finite set Ω and a real-valued function $f: 2^{\Omega} \to \mathbb{R}$. For any two subsets $\mathcal{X}, \mathcal{Y} \subseteq \Omega$ such that $\mathcal{X} \subseteq \mathcal{Y}$, the function f is said to be submodular if

$$f(\mathcal{X} \cup \{x\}) - f(\mathcal{X}) \geqslant f(\mathcal{Y} \cup x) - f(\mathcal{Y}) \tag{14}$$

for each $x \in \Omega \setminus \mathcal{Y}$. The inequality (14) expresses the property of diminishing returns, i.e., the increase in f due to the introduction of a new element in a set diminishes with the size of that set. In the context of sensor placement, this means that if an information gain measure is submodular, then the information gain due to the placement of a new sensor diminishes with the number of sensors already placed.

Proposition 1: The CRMI $I(J; z(\boldsymbol{q}))$ is submodular.

Proof: Refer to Appendix A. □

Algorithm 2: Greedy CRMI optimization

Set $\boldsymbol{q}_{\mathrm{gr}} = \varnothing$;
for $i = 1$ to N_S **do**
  Set $\boldsymbol{q} \coloneqq \{N_G\} \setminus \boldsymbol{q}_{\mathrm{gr}}$;
  Calculate $q_i^* \coloneqq \arg\max_{q \in \boldsymbol{q}} I(J; z(\boldsymbol{q}))$;
  $\boldsymbol{q}_{\mathrm{gr}} = \boldsymbol{q}_{\mathrm{gr}} \cup q_i^*$.

D. Greedy Sensor Placement

We implement a greedy sensor placement algorithm, as shown in Algorithm 2, in which the sensor locations are chosen in sequence such that the choice of the next sensor maximizes the CRMI. This approach aims to select a set of sensors that collectively provides the maximum information relevant to the path planning. As described in Algorithm 2, greedy optimization initializes the empty configuration $\boldsymbol{q}_{\mathrm{gr}} = \varnothing$ and iterations are carried out until N_S sensors are selected. At each iteration, the greedy optimal configuration is the scalar $q^* \in \{N_G\} \setminus \boldsymbol{q}_{\mathrm{gr}}$ that maximizes CRMI $I(J; z(q))$. The configuration $\boldsymbol{q}_{\mathrm{gr}}$ is then updated to include q^*.

Theorem 1: [49] The greedy placement algorithm for any monotone submodular function provides a performance guarantee of $[1 - (1/e)]$ times the optimal value.

We denote the maximum $I(J; z(\boldsymbol{q}^*))$ as an optimal value, and $I(J; z(\boldsymbol{q}_{\mathrm{gr}}))$ as the approximate mutual information value. For any $k = N_S$ sensor elements chosen by the greedy algorithm, it follows from Theorem 1 that

$$I(J; z(\boldsymbol{q}_{\mathrm{gr}})) \geqslant \left(1 - \left(1 - \frac{1}{k}\right)^k\right) I(J; z(\boldsymbol{q}^*)) \geqslant \left(1 - \frac{1}{e}\right) I(J; z(\boldsymbol{q}^*))$$

E. Sensor Reconfiguration Cost

Sensor reconfiguration cost refers to the cost of moving sensors from one location to another in the grid space. We consider a sensor reconfiguration cost based on the Euclidean distance between the new and previous sensor locations, as illustrated in Fig. 2. Here, d_i^j is the Euclidean distance between the ith grid point and the location of the jth sensor.

Informally, at the kth iteration of the CSCP algorithm, the objective is to find new sensor locations $\boldsymbol{q}_{k+1}^*$ that maximize the CRMI while minimizing the cost of sensor reconfiguration from the current configuration $\boldsymbol{q}_k^* = \{q_1^{k^*}, \ldots, q_{N_S}^{k^*}\}$. To this end we define

$$I^{\mathrm{mod}}(\boldsymbol{q}) \coloneqq I(J_k; z_k(\boldsymbol{q})) + \alpha_1 - \alpha_2 \min_{j,\ell \in \{N_S\} \times \{N_S\}} \| q_\ell - q_j^{k^*} \| \tag{15}$$

where α_1, α_2 are constants. We choose α_1 to be a relatively large value, e.g., proportional to the size of the overall workspace, whereas α_2 is chosen based on the user's preference for reducing the reconfiguration cost.

Proposition 2: The modified CRMI $I^{\mathrm{mod}}(\boldsymbol{q})$ is submodular.

Proof: Refer to Appendix A. □

F. Convergence of the Proposed CSCP Algorithm

In this section, we show that the proposed CSCP algorithm converges in a finite number of iterations. To this end, first consider the following result.

Proposition 3: If A is Schur, then the CSCP algorithm terminates in a finite number of iterations.

Proof: If all modes of A are stable, then the pair $(A, C_k(\boldsymbol{q}))$ is uniformly detectable for any $\boldsymbol{q}$. Furthermore, because $Q = \sigma_P^2 \boldsymbol{I}_{N_P}$, the pair (A, Q) is controllable. By [50], it follows that the estimation error is exponentially stable. Consequently, for any $\varepsilon > 0$ there is a finite iteration number $M \in \mathbb{N}$ such that $\mathrm{tr}(P_k) \leqslant \varepsilon$, after which the CSCP algorithm terminates. □

Remark 1: Whereas Proposition 3 is sufficient, it is not necessary for the convergence of the CSCP algorithm. A less restrictive criterion for convergence, which does not assume A to be Schur, is discussed below.

At the kth iteration of the CSCP method, the predicted error covariance is defined as $P_{k|k-1} \coloneqq AP_{k-1|k-1}A^{\top} + Q_{k-1}$. Here, the term $AP_{k-1|k-1}A^{\top}$ grows the uncertainty from the previous step's posterior covariance based on the A, and Q_{k-1} accounts for the uncertainty introduced due to process noise. The measurement update is defined as $P_{k|k} \coloneqq P_{k|k-1} - L_k C_k P_{k|k-1}$. The uncertainty growth in the system is $\{P_{k|k-1} - P_{k-1|k-1} = AP_{k-1|k-1}A^{\top} - P_{k-1|k-1} + Q_{k-1}\}$.

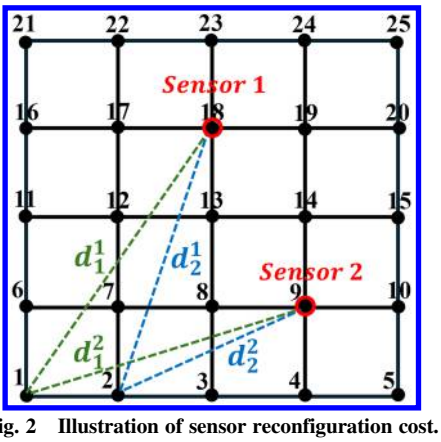

Fig. 2 Illustration of sensor reconfiguration cost.

Similarly, the reduction in the uncertainty of the system after the measurement update is $\{P_{k|k-1} - P_{k|k} = L_k C_k P_{k|k-1}\}$. The convergence of the CSCP method is guaranteed if the following criterion is satisfied for every time steps.

$$\mathrm{tr}(L_k C_k P_{k|k-1}) \geqslant \mathrm{tr}(A P_{k-1|k-1} A^\top - P_{k-1|k-1} + Q_{k-1})$$

Note that this criterion cannot be verified *a priori*, as it depends on the estimation error covariances computed during the algorithm's execution.

IV. Results and Discussion

In this section, we first provide an illustrative example of the proposed CSCP-CRMI method. Second, we compare the proposed method against a decoupled method that finds optimal sensor placement using the standard (path-independent) MI; for brevity we call this decoupled method CSCP-SMI. Third, we study the effects of varying numbers of sensors, threat parameters, and grid points on the CSCP-CRMI method. Fourth, we perform a comparative study between the CSCP-CRMI and greedy sensor placements and observe the equivalency as well as differences between the two approaches. Finally, we conduct a comparative study between the CSCP method with and without the sensor reconfiguration cost. All simulations are performed within a square workspace $\mathcal{W} = [-1,1] \times [-1,1]$ using nondimensional units in a Cartesian coordinate axes system.

A MATLAB®-based implementation of the CSCP method used for producing these results is available at this repository: https://github.com/prakashpoudel2014/CSCP_time_varying.

A. Illustrative Example

The implementation of the CSCP-CRMI algorithm on an illustrative example is shown in Fig. 3. The number of threat parameters, grid points, and sensors are $N_P = 25$, $N_G = 49$, and $N_S = 2$, respectively. The threat parameters N_P, indicated by the black dots and numbered from 1 to 25, are uniformly spaced in the workspace. The white dots represent the grid points. The initial and the goal points are represented by the bottom left and the top right grid points in the map. The evolution of the threat field estimate $\hat{c}$ for different time steps, namely $k = 1, 5, 11$, and 15 is shown by a color map. The path $\boldsymbol{v}_k^*$ of minimum estimated cost is indicated by red circles, and the sensor placement $\boldsymbol{q}_k$ is shown by white circles, as illustrated in Figs. 3a–3d. For a specified threshold $\varepsilon = 0.1$, the algorithm terminates at $k = 15$ iterations, and the optimal path $\boldsymbol{v}^*$ is achieved. Figures 4a–4d show the absolute error between the true threat field c and the estimated field $\hat{c}$, computed as $|c - \hat{c}|$, for the respective iterations $k = 1, 5, 11, 15$. These plots illustrate how the estimation accuracy improves as the algorithm progresses.

The comparison between the true and estimated path cost is illustrated in Fig. 5a. Upon termination, the true and estimated path costs are nearly identical, with $J(\boldsymbol{v}_k^*) = 16.46$ and $\hat{J}(\boldsymbol{v}_k^*) = 16.77$, respectively. Figure 5b shows the convergence of the proposed CSCP-CRMI algorithm. The path cost variance $\mathrm{Var}[(\hat{J}(\boldsymbol{v}_k^*)]$ decreases with time, and the algorithm terminates in 15 iterations as the path cost variance decreases below 0.1.

B. Comparison of CSCP-CRMI and CSCP-SMI

For comparison, now consider the execution of CSCP-SMI on the same example as discussed above.

Figure 6 shows the estimation error covariance P at the final iteration, mapped to the spatial regions of the environment using the centers of spatial basis functions Φ. In a slight departure from convention, Fig. 6 shows the *logarithms* of the diagonal values of P, which explains the negative values despite P being a symmetric positive definite matrix. The reason for this choice of logarithmic values is to clearly show the orders of magnitude difference in the estimation error covariance in different regions of the environment.

The white regions in Fig. 6a with high error covariance represent areas where few, if any, sensors are placed throughout the

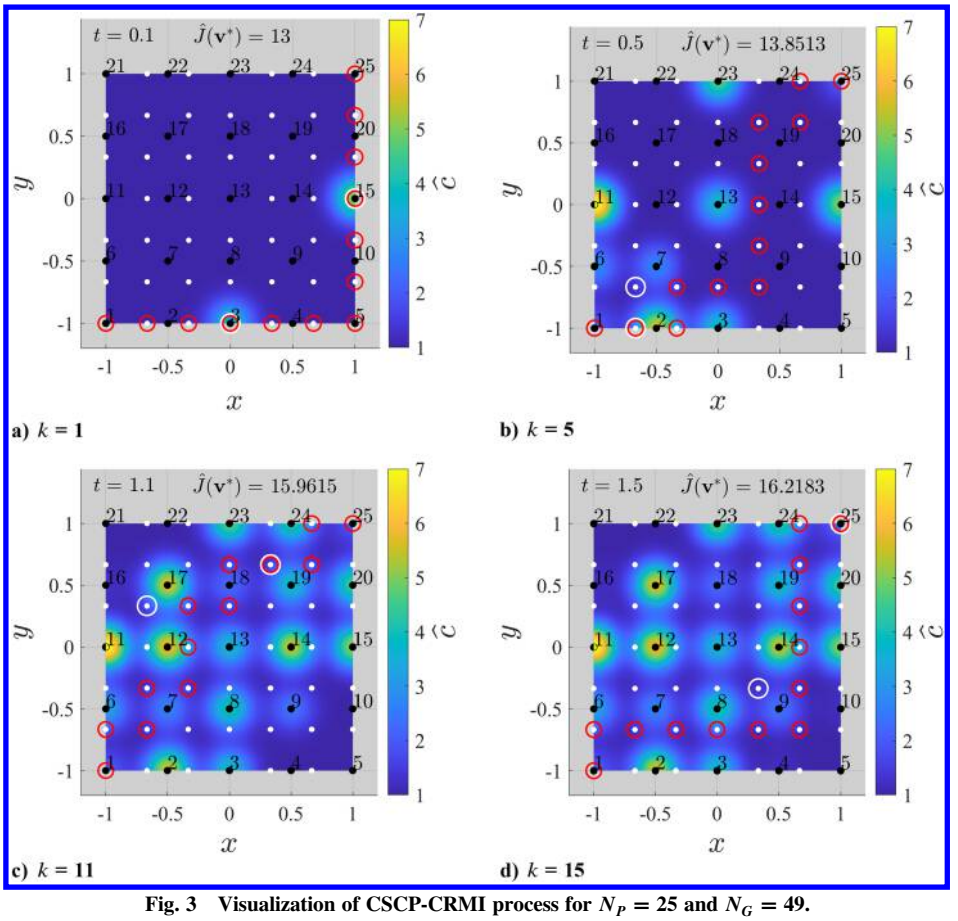

Fig. 3 Visualization of CSCP-CRMI process for $N_P = 25$ and $N_G = 49$.

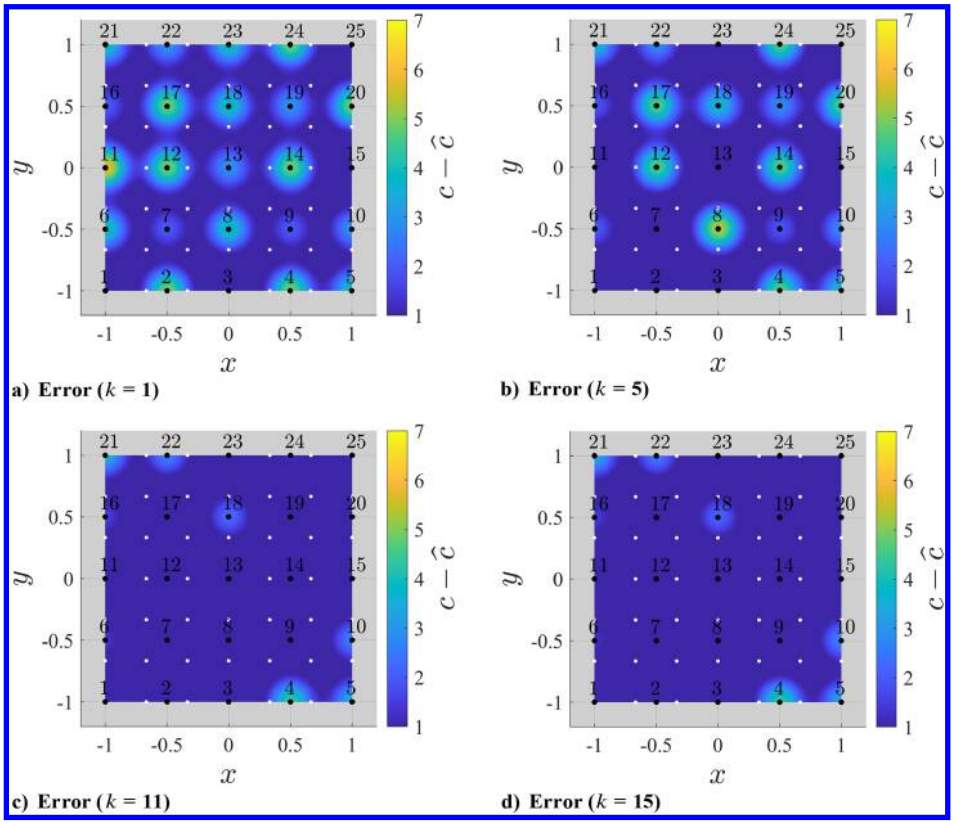

Fig. 4 Error between the mean estimate and ground truth during CSCP-CRMI, for $N_P = 25$ and $N_G = 49$.

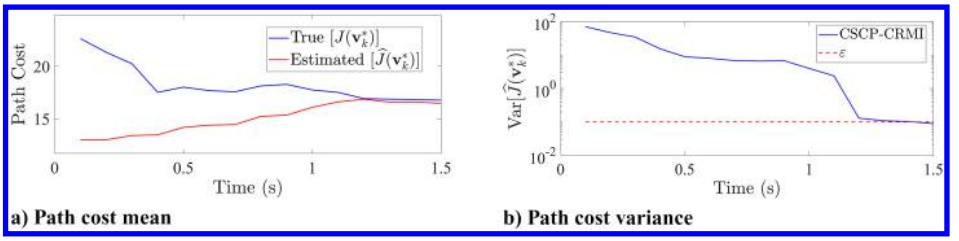

Fig. 5 Convergence of CSCP-CRMI algorithm.

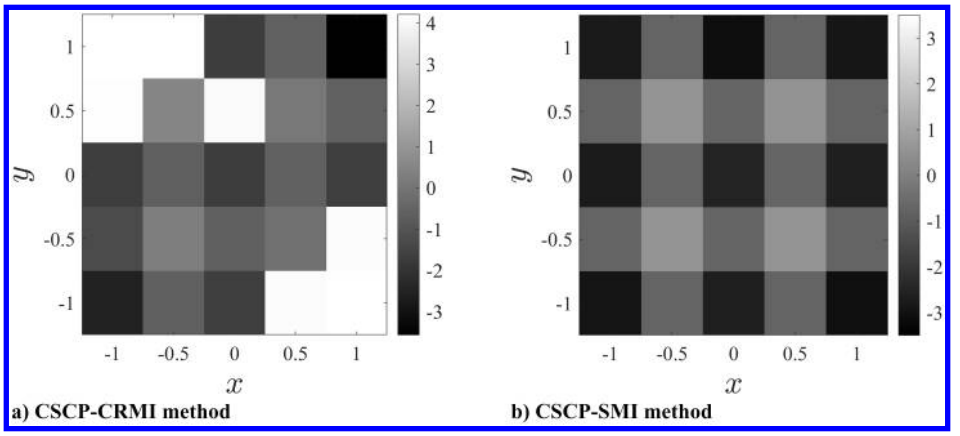

Fig. 6 Diagonal log values of error covariance at final iteration.

execution of CSCP-CRMI. The darker regions represent areas where sensors were placed to reduce the estimation error covariance values orders of magnitude below those of the white-colored regions. Compare Fig. 6 to the optimal path found in Fig. 3d, and we find that the CSCP-CRMI sensor placement is such that areas around the optimal path generally have lower estimation error covariance. Note that the path does not exclusively follow regions of minimal covariance. The planning objective is to minimize cumulative threat exposure; the optimal trajectory occasionally favors areas with slightly higher estimation uncertainty if the expected threat intensity in those areas is substantially lower. Note, crucially, the novelty of this approach: *the optimal path is at first unknown*, and the sensor placement and path planning are performed iteratively to arrive at these results.

Figure 6b shows the estimation error covariance P of the CSCP-SMI method. By contrast to Fig. 6a for CSCP-CRMI, we note here

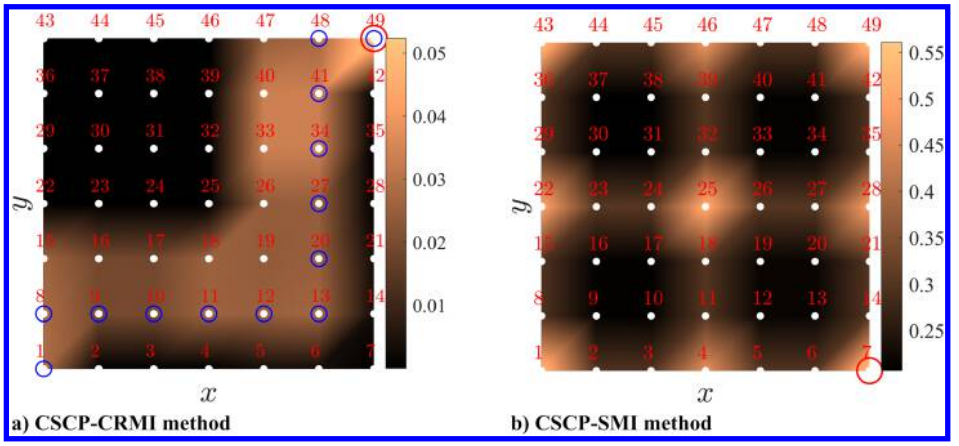

Fig. 7 Mutual information intensity map.

spatially uniform covariance values. This means that in CSCP-SMI, the sensors are placed in such a way that the error covariances in *all* regions of the environment are low compared to CSCP-CRMI. Whereas this would be of benefit if we were merely trying to map the threat in the environment, this uniformly low error covariance is indicative of wasteful sensor placement in the context of path planning. In the case of CSCP-CRMI, although there are some regions that are not explored, the outcome of the path-planning algorithm is still near-optimal.

An intensity map showing the mutual information values for each grid point from an illustrative example is shown in Fig. 7. Notably, these values are derived with consideration for only a single sensor. The brown regions represent the areas with higher CRMI values. It can be observed in Fig. 7a that the CRMI regions are more visible around the vicinity of the path obtained in Fig. 3d. The optimal path, indicated by blue circles, is overlaid on the CRMI intensity map for clarity. The sensor is placed at the grid point, here number 49, with a maximum CRMI value. Similarly, Fig. 7b shows the SMI map representing the mutual information values between the state and measurement variables. The higher CRMI regions are observed around the location of the threat parameter, and for this example, the sensor is placed at the 7 numbered grid point.

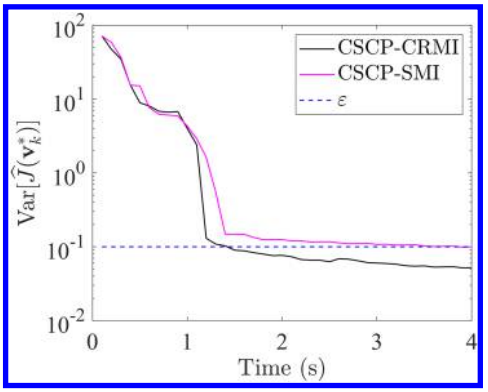

Fig. 8 Comparison of path cost variance between CSCP-CRMI and CSCP-SMI with $N_S = 2$.

Figure 8 shows a comparison between the path cost variance $\mathrm{Var}[J(\boldsymbol{v}_k^*)]$ of the two methods. Note that, for $\varepsilon = 0.1$, the CSCP-SMI algorithm requires 39 iterations to converge, whereas the CSCP-CRMI algorithm requires only 15 iterations. This indicates that the number of iterations for the SMI-based method is *160% larger* than CSCP. Figure 9 shows similarly large differences in convergence rate for different numbers of sensors.

Figure 10a shows the variation of path cost variance with varying the number of sensors. For a specified number of threat parameters, $N_P = 49$ and the grid points $N_G = 49$, better convergence is achieved with more number of sensors. Using a single sensor will require a relatively large number of iterations for convergence.

The convergence of the CSCP-CRMI algorithm for different numbers of threat parameters N_P is shown in Fig. 10b. For $N_S = 2$ and $N_G = 81$, CSCP converges faster for fewer threat states (e.g., $N_P = 9$ or 16) compared to, say, $N_P = 64$ or 81.

We also performed comparative analysis for a specific number of sensors and parameters with varying numbers of grid points. It is observed that the path cost and the path cost variance remain unchanged for different numbers of grid points. This result is a consequence of proper scaling in the path cost formulation, namely, the scaling of the cost by the grid spacing δ.

C. Analysis of Greedy Optimization

A comparative example of sensor placements for $N_P = 25$, $N_G = 25$, and $N_S = 4$ using greedy and nongreedy (true optimal) is shown in Fig. 11a. Red circles in the figure indicate sensor positions with greedy placement, whereas the black circles indicate true optimal. Note that the two sensor locations in the grid space are

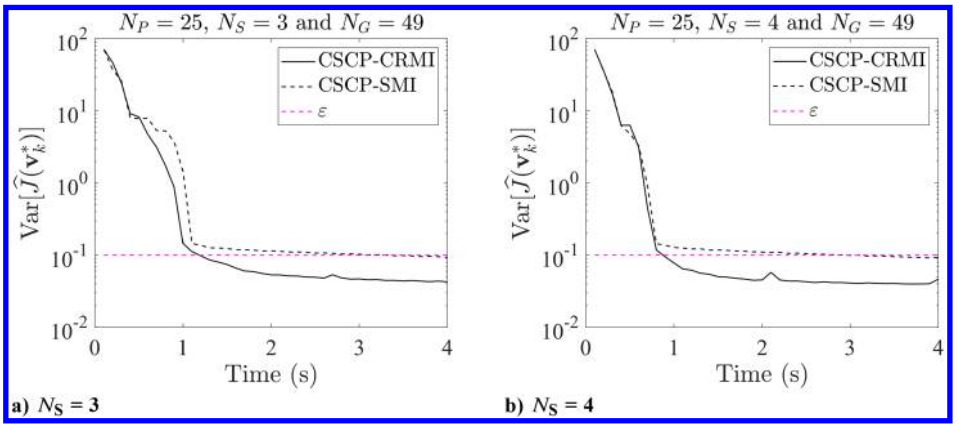

Fig. 9 Additional comparison of path cost variance between CSCP-CRMI and CSCP-SMI.

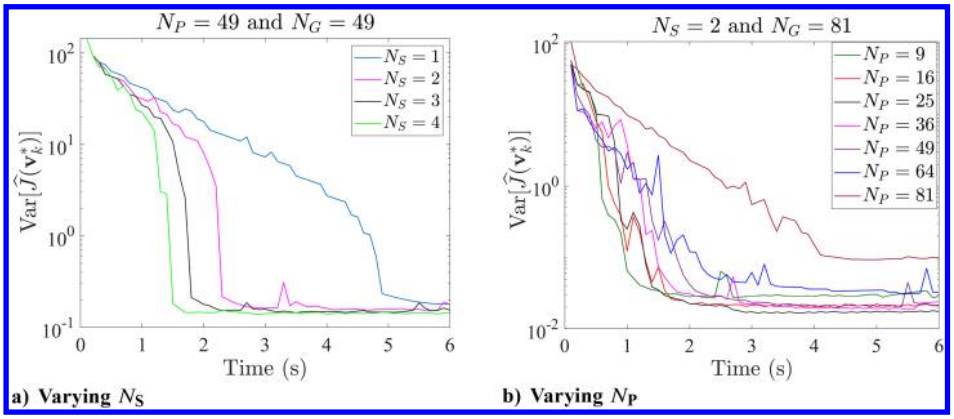

Fig. 10 Convergence of CSCP-CRMI algorithm for different number of sensors and parameters.

common for both criteria. This is a specific example where greedy optimization results in the true optimal sensor configuration. For a submodular function, greedy optimization results in near-optimal configurations.

Figure 11b compares the time required to compute the optimal sensor configuration using the greedy and nongreedy approaches. The relative computing time is plotted for different combinations of grid points and number of sensors. As the number of sensors increases, the relative computation time rapidly decreases, indicating the computational advantages of the greedy method. A sharp decrease in the relative computation time is observed, especially as the number of grid points increases.

A comparison between the CRMI values for greedy and nongreedy methods is shown in Fig. 11c. The suboptimality bounds described in Theorem 1 are also shown. As expected, Fig. 11c indicates that the greedy optimization-based CRMI values lie within these bounds. Finally, a comparison of the path cost variance is shown in Fig. 11d. For this specific example and $\varepsilon = 0.1$, the nongreedy (true optimal) method results in 12 CSCP iterations for convergence, whereas the greedy method requires 16 CSCP iterations.

D. Analysis of Sensor Reconfiguration Cost

The number of threat parameters, sensors, and grid points used for the analysis are $N_P = 49$, $N_S = 3$, and $N_G = 121$, respectively. Figure 12a shows the comparison between the true and estimated mean path costs. Initially, both estimated mean path costs are much lower than the true path cost. This is because there is not much information about the environment, and the estimator relies heavily on the "optimistic" prior, which results in threat estimates with small values. The estimated path costs for both with and without sensor reconfiguration costs are similar. The CSCP method with a sensor reconfiguration cost converges in 21 iterations, whereas the method without considering the sensor reconfiguration cost converges in 20 iterations. The convergence of both CSCP methods is shown by a path cost variance plot in Fig. 12b. As the iterative process continues, the path cost variance $\mathrm{Var}[(\hat{J}(\boldsymbol{v}_k^*)]$ decreases, and the algorithm terminates when $\mathrm{Var}[(\hat{J}(\boldsymbol{v}_k^*)]$ falls below $\varepsilon = 0.1$. Figure 12c shows the sensor reconfiguration cost values at different time steps. For the computation of sensor reconfiguration cost, we choose $\alpha_1 = \sqrt{8}$ (diagonal distance across the workspace) and $\alpha_2 = 0.01$. For

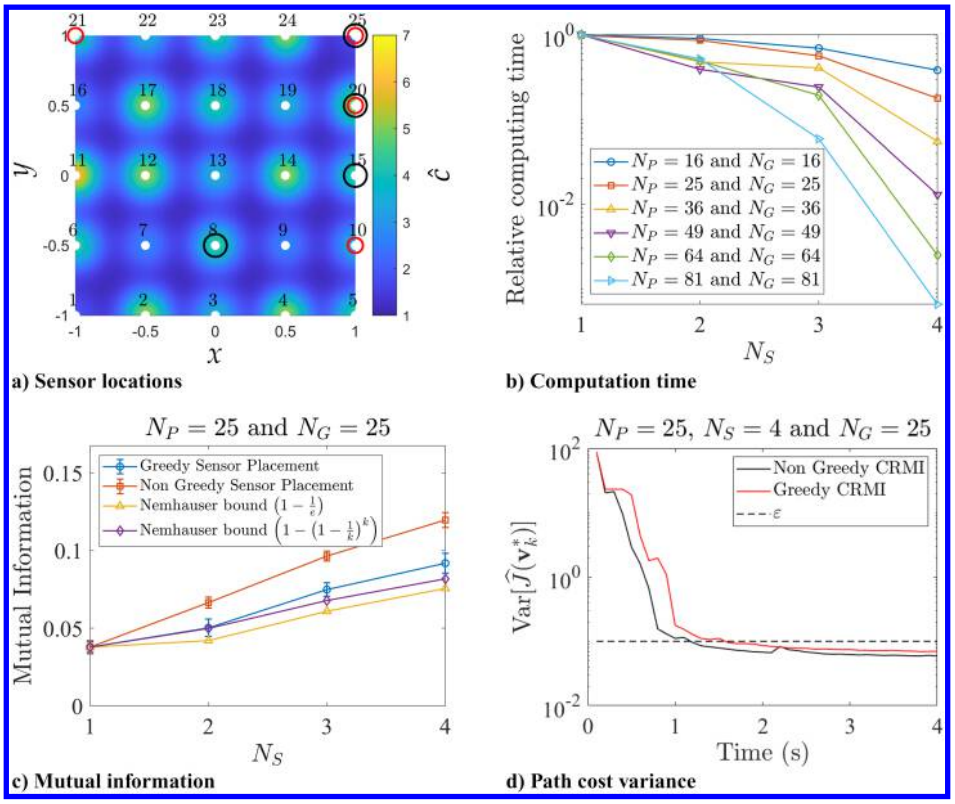

Fig. 11 Comparison between greedy and nongreedy CSCP-CRMI.

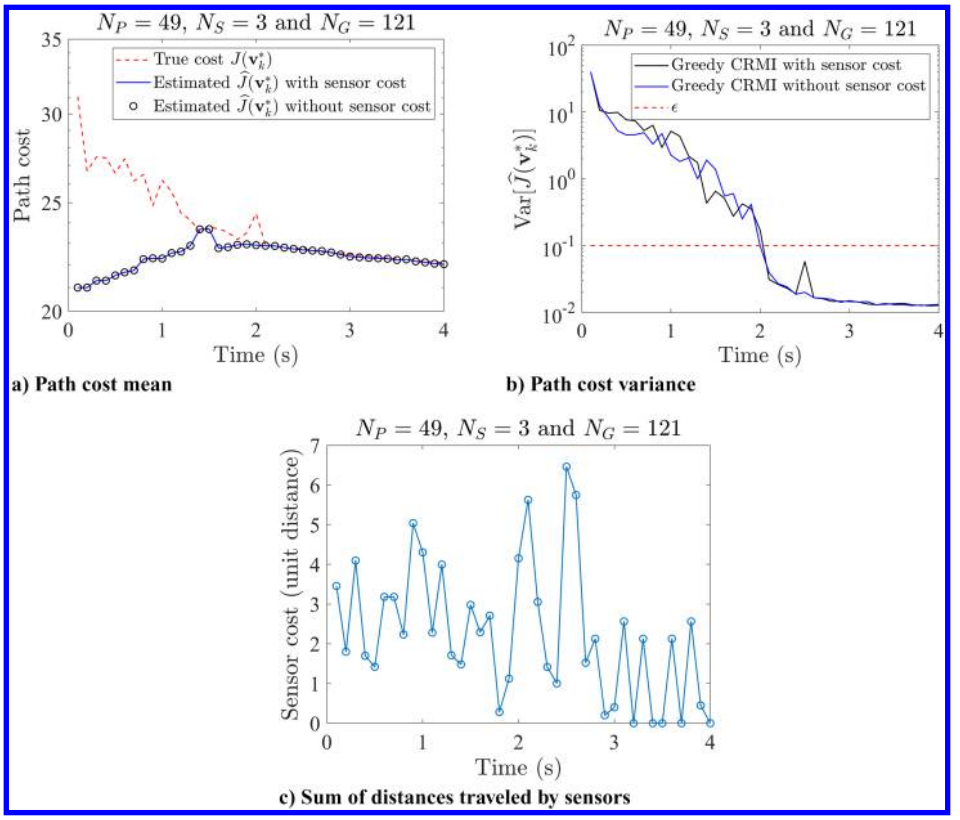

Fig. 12 Path cost mean, variance over CSCP iterations with and without sensor reconfiguration cost.

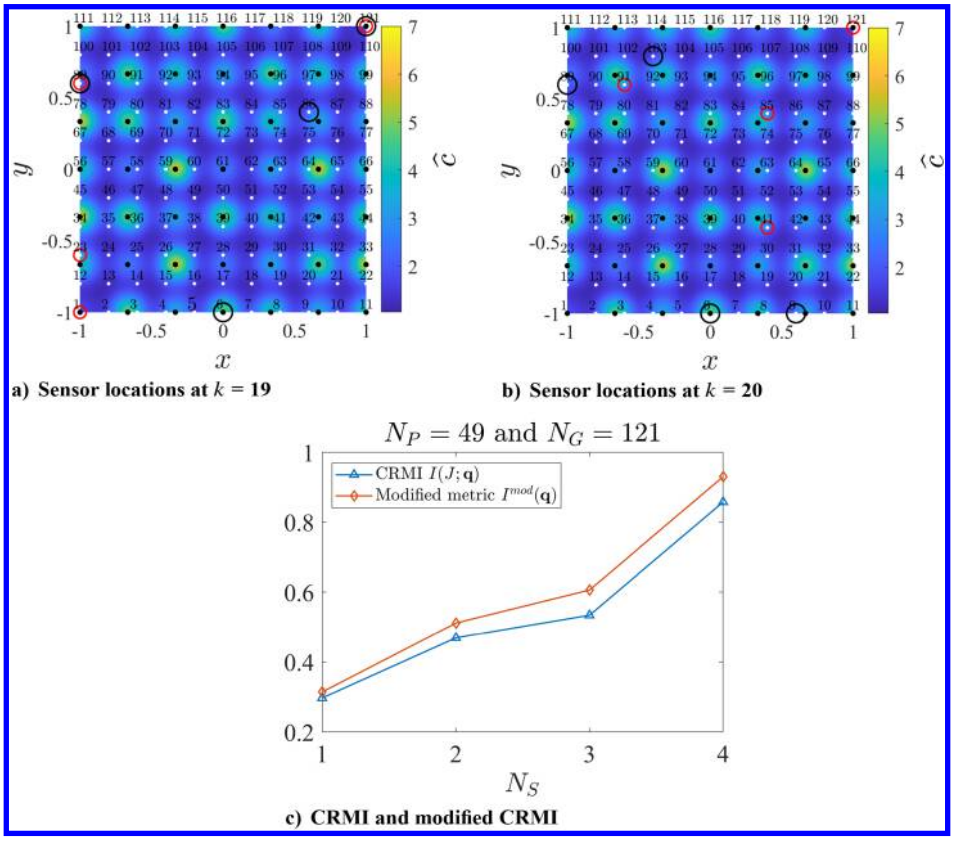

Fig. 13 Comparison between the CSCP method with and without sensor reconfiguration cost.

$N_S = 3$, the sensor cost is the cumulative sum of the distance traveled by the three sensors at each iteration.

A comparative example of sensor placements for $N_P = 49$, $N_G = 121$, and $N_S = 4$ using CSCP with and without consideration of the sensor configuration cost is shown in Figs. 13a and 13b. Red circles in the figure indicate sensor positions based on maximizing CRMI, whereas the black circles indicate sensor locations based on maximizing the modified CRMI. At $k = 19$, it can be observed that

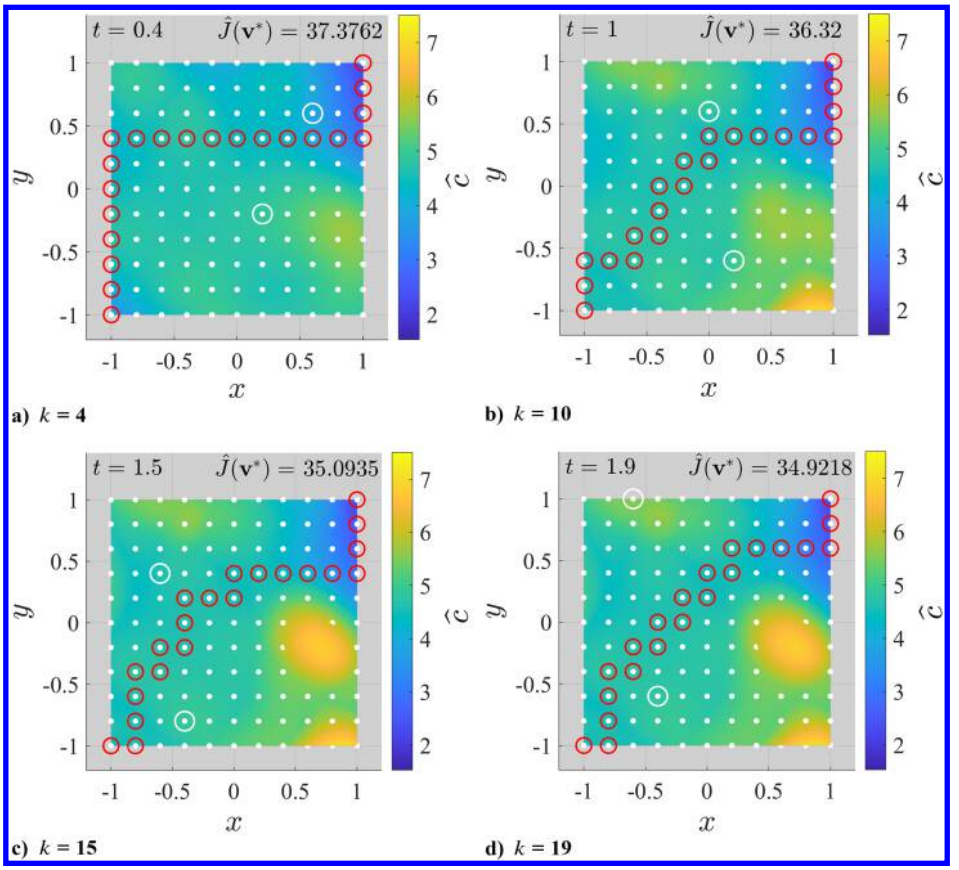

Fig. 14 CSCP iterations for nonuniformly spaced basis functions, with $N_P = 49$, $N_G = 121$, and $N_S = 2$.

the two sensor locations in the grid space are common for methods. As expected, when sensor reconfiguration cost is considered, the new sensor locations (black circles) at $k = 20$ are relatively closer to the previous locations at $k = 19$ as compared to sensor positions denoted by the red circles across the two iterations. Figure 13c shows the values of the CRMI and the modified CRMI for different numbers of sensors.

Finally, although the basis functions used in the previous results had uniformly spaced centers and nonsignificant intersections in regions of support, the proposed CSCP method is not restricted by these assumptions. For instance, Fig. 14 shows different CSCP iterations in a threat field constructed using overlapping basis functions and nonuniformly spaced centers.

E. Computational Complexity

The computational complexity of Algorithm 1 depends on the complexity of CRMI optimization in line 5 and the number of calls performed in Dijkstra's algorithm in line 9. CRMI calculation and maximization involves determinant and inverse computations, thus has a time complexity of $\mathcal{O}(N_S^3)$. By comparison, the CRMI calculation for the greedy optimization method has a complexity of $\mathcal{O}(N_S)$. The worst-case time complexity of Dijkstra's algorithm implemented with a Fibonacci heap is $\mathcal{O}(N_G + N_G \log(N_G))$ [41]. The complexity of CRMI optimization depends on the specific algorithm chosen. CSCP addresses scenarios where sensor resources are limited. Scaling up should then be considered in the context of the grid size, which may be achieved by relaxation to a continuous workspace and a suitable optimization method, e.g., evolutionary global optimization methods, as previously mentioned. Further computational efficiency may be achieved via decentralized estimation, as we had reported earlier in [51], albeit with a naïve sensor placement method.

V. Conclusions

In this paper, we discussed a new measure of information gain for optimal sensor placement to capture the coupling of the sensor placement problem with a path-planning problem. This measure, which we call the context-relevant mutual information (CRMI), addresses the reduction in uncertainty in the path cost, rather than the environment state. We presented a coupled sensor placement and path-planning algorithm that iteratively places sensors based on CRMI maximization, updates the environment threat estimate, and then plans paths with minimum expected cost. Crucially, we showed CRMI to be a submodular function, due to which we can apply greedy optimization to arrive at near-optimal results while maintaining computational efficiency. We performed a comparative study between CSCP-CRMI and a decoupled CSCP-SMI method. The CSCP-SMI method places sensors by maximizing the standard (path-independent) mutual information of the measurements and threat state. We showed via numerical simulation examples that the CSCP-CRMI algorithm converges in less than half as many iterations compared to the CSCP-SMI algorithm, which indicates a significant reduction in the number of sensor observations needed to find near-optimal paths. We also introduced and analyzed a modified cost function that addresses costs associated with distances traveled by sensors (i.e., sensor reconfiguration cost) after each iteration of placement. For future work, we will consider other real-world sensor constraints such as communication limitations, energy consumption, and sensor reliability issues.

Appendix A: Technical Proofs

Proof of Proposition 1: By the definition of path-relevant set $\mathcal{K}$, for the path $\boldsymbol{v}_k^* = \{v_0, v_1, \ldots, v_L\}$, the values of $\phi_m(\boldsymbol{x}_{v_\ell}) \approx 0$ for each $\ell = (0, 1, \ldots, L)$ and $m \notin \mathcal{K}$. Therefore, for all basis functions $m \notin \mathcal{K}$, $\boldsymbol{\Phi}(\boldsymbol{x}_{v_\ell}) = \boldsymbol{0}$, and by Eq. (10), $P_{Jz_{k|k-1}} \approx 0$. By Eq. (12), $I(J_k; z_k(\boldsymbol{q})) \approx 0$ at locations in the workspace where the basis functions ϕ_m do not overlap with the path $\boldsymbol{v}$.

Per the proposed sensor reconfiguration policy, $\boldsymbol{q}_k^* = \max_{\boldsymbol{q}} I(J_k; z_k(\boldsymbol{q}))$, sensors are necessarily placed at locations within the regions of significant support $\mathcal{R}_m^{\text{sup}}$ defined by ϕ_m, for each $m \in \mathcal{K}$, i.e., within the basis support regions of the basis functions contained within a path-relevant set. Given the path cost $J(\boldsymbol{v})$ is known, the

sensor measurements $\boldsymbol{z}(\boldsymbol{q})$ within the path-relevant set $\mathcal{K}$ fully capture the information about the $J(\boldsymbol{v})$. This shows that the sensor measurements are conditionally independent given the path cost. i.e., $p(z_1, z_2, \dots z_{N_s}|J) = p(z_1|J)p(z_2|J)\dots p(z_{N_s}|J)$. Next, we show that this conditional independence implies submodularity.

Consider two subsets of the sensor configurations, $\mathcal{A}$ and $\mathcal{B}$, such that $\mathcal{A} \subseteq \mathcal{B} \subseteq \boldsymbol{q}$, where $\boldsymbol{q} \subset \{N_G\}$. By Eq. (13), we write $I(J;\mathcal{A}) = H(\mathcal{A}) - H(\mathcal{A}|J)$ and $I(J;\mathcal{B}) = H(\mathcal{B}) - H(\mathcal{B}|J)$. For any $x \in \boldsymbol{q} \setminus \mathcal{B}$, the marginal CRMI gain due to a sensor placed at x to in addition to those in sets $\mathcal{A}$ and $\mathcal{B}$ is

$$\begin{aligned}\Delta_{\mathcal{A}}(x) &= I(J;\mathcal{A}\cup x) - I(J;\mathcal{A})\\ &= H(\mathcal{A}\cup x) - H(\mathcal{A}\cup x|J) - H(\mathcal{A}) + H(\mathcal{A}|J)\\ \Delta_{\mathcal{B}}(x) &= I(J;\mathcal{B}\cup x) - I(J;\mathcal{B})\\ &= H(\mathcal{B}\cup x) - H(\mathcal{B}\cup x|J) - H(\mathcal{B}) + H(\mathcal{B}|J)\end{aligned}$$

By the chain rule of conditional entropy, $H(\mathcal{A}\cup x|J) = H(\mathcal{A}|J) + H(x|\mathcal{A},J)$, and $H(\mathcal{B}\cup x|J) = H(\mathcal{B}|J) + H(x|\mathcal{B},J)$. Also, the conditional independence of $\boldsymbol{z}(\boldsymbol{q})$ given J implies $H(x|\mathcal{A},J) = H(x|\mathcal{B},J) = H(x|J)$. It follows that

$$\begin{aligned}\Delta_{\mathcal{A}}(x) &= H(\mathcal{A}\cup x) - H(x|J) - H(\mathcal{A}),\\ \Delta_{\mathcal{B}}(x) &= H(\mathcal{B}\cup x) - H(x|J) - H(\mathcal{B})\end{aligned}$$

By the chain rule again, $H(\mathcal{A}\cup x) = H(\mathcal{A}) + H(x|\mathcal{A})$ and $H(\mathcal{B}\cup x) = H(\mathcal{B}) + H(x|\mathcal{B})$ from the previous expressions, we obtain $\Delta_{\mathcal{A}}(x) = H(x|\mathcal{A}) - H(x|J)$, and $\Delta_{\mathcal{B}}(x) = H(x|\mathcal{B}) - H(x|J)$. Therefore, the difference in marginal gains is $\Delta_{\mathcal{A}}(x) - \Delta_{\mathcal{B}}(x) = H(x|\mathcal{A}) - H(x|\mathcal{B})$, which must be nonnegative because $\mathcal{A} \subseteq \mathcal{B}$. Consequently, $I(J;\mathcal{A}\cup x) - I(J;\mathcal{A}) \geqslant I(J;\mathcal{B}\cup x) - I(J;\mathcal{B})$, which satisfies the submodularity criterion. □

Proof of Proposition 2: The sensor reconfiguration cost function can be represented as a set function:

$$f(\mathcal{S}) \coloneqq \alpha_1 - \alpha_2 \min_{j\in\{N_S\},\ell\in\mathcal{S}}\{\|q_\ell - q_j^{k^*}\|\}$$

where $\mathcal{S} \subseteq \{N_S\}$ is a subset of sensor indices. Consider two subsets $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \subseteq \mathcal{B} \subseteq \{N_S\}$ and $x \in \{N_S\} \setminus \mathcal{B}$. In the rest of this proof, the symbol j is an index over $\{N_S\}$, i.e., $j \in \{N_S\}$, which we avoid writing explicitly for notational convenience. Without loss of generality, we assume $\alpha_2 = 1$.

To calculate the marginal costs due to the inclusion of the index x in either $\mathcal{A}$ or $\mathcal{B}$, we note

$$\begin{aligned}f(\mathcal{A}\cup x) &= \alpha_1 - \min_{\ell\in\{\mathcal{A}\cup x\}}\|q_\ell - q_j^{k^*}\|\\ f(\mathcal{B}\cup x) &= \alpha_1 - \min_{\ell\in\{\mathcal{B}\cup x\}}\|q_\ell - q_j^{k^*}\|\\ \Rightarrow f(\mathcal{A}\cup x) - f(\mathcal{A}) &= -\min_{\ell\in\{\mathcal{A}\cup x\}}\|q_\ell - q_j^{k^*}\| + \min_{\ell\in\mathcal{A}}\|q_\ell - q_j^{k^*}\|\\ \Rightarrow f(\mathcal{B}\cup x) - f(\mathcal{B}) &= -\min_{\ell\in\{\mathcal{B}\cup x\}}\|q_\ell - q_j^{k^*}\| + \min_{\ell\in\mathcal{B}}\|q_\ell - q_j^{k^*}\|\end{aligned}$$

Since $\mathcal{A} \subseteq \mathcal{B}$, $\min_{\ell\in\mathcal{A}}\|q_\ell - q_j^{k^*}\| \geqslant \min_{\ell\in\mathcal{B}}\|q_\ell - q_j^{k^*}\|$. It follows that, for sufficiently large α_1, $f(\mathcal{A}) \leqslant f(\mathcal{B})$. Then we consider the following three cases:

$$\|q_x - q_j^{k^*}\| \leqslant \min_{\ell\in\mathcal{B}}\|q_\ell - q_j^{k^*}\| \leqslant \min_{\ell\in\mathcal{A}}\|q_\ell - q_j^{k^*}\| \quad \text{(i)}$$

$$\min_{\ell\in\mathcal{B}}\|q_\ell - q_j^{k^*}\| \leqslant \|q_x - q_j^{k^*}\| \leqslant \min_{\ell\in\mathcal{A}}\|q_\ell - q_j^{k^*}\| \quad \text{(ii)}$$

$$\min_{\ell\in\mathcal{B}}\|q_\ell - q_j^{k^*}\| \leqslant \min_{\ell\in\mathcal{A}}\|q_\ell - q_j^{k^*}\| \leqslant \|q_x - q_j^{k^*}\| \quad \text{(iii)}$$

For case (i) we note

$$\begin{aligned}f(\mathcal{A}\cup x) - f(\mathcal{A}) &= \min_{\ell\in\mathcal{A}}\|q_\ell - q_j^{k^*}\| - \|q_x - q_j^{k^*}\|,\\ f(\mathcal{B}\cup x) - f(\mathcal{B}) &= \min_{\ell\in\mathcal{B}}\|q_\ell - q_j^{k^*}\| - \|q_x - q_j^{k^*}\|\end{aligned}$$

It follows that $f(\mathcal{A}\cup x) - f(\mathcal{A}) > f(\mathcal{B}\cup x) - f(\mathcal{B})$. For case (ii) we note

$$\begin{aligned}f(\mathcal{A}\cup x) - f(\mathcal{A}) &= \min_{\ell\in\mathcal{A}}\|q_\ell - q_j^{k^*}\| - \|q_x - q_j^{k^*}\|,\\ f(\mathcal{B}\cup x) - f(\mathcal{B}) &= \min_{\ell\in\mathcal{B}}\|q_\ell - q_j^{k^*}\| - \min_{\ell\in\mathcal{B}}\|q_\ell - q_j^{k^*}\| = 0\end{aligned}$$

Again, it follows that $f(\mathcal{A}\cup x) - f(\mathcal{A}) > f(\mathcal{B}\cup x) - f(\mathcal{B})$. Finally, for case (iii) we note

$$f(\mathcal{A}\cup x) - f(\mathcal{A}) = f(\mathcal{B}\cup x) - f(\mathcal{B}) = 0$$

Therefore, the inequality $f(\mathcal{A}\cup x) - f(\mathcal{A}) \geqslant f(\mathcal{B}\cup x) - f(\mathcal{B})$ is always true and f is submodular. By Proposition 1 and the additive property of submodular functions [49], $I^{\text{mod}}(\boldsymbol{q})$ is submodular. □

Appendix B: Unscented Kalman Filter for Threat Estimation

The estimated state parameters and the error covariance of the system are calculated by an unscented Kalman filter (UKF) [4]. Although the scope of this paper is limited to linear threat field dynamics, we would like to ensure generality for nonlinear threat models to be considered in the future. For the reader's convenience we provide a brief overview of the UKF here and refer to [4] for further details.

An augmented state variable $\boldsymbol{\Theta}^a_{k-1} \in \mathbb{R}^{N_{\boldsymbol{\Theta}}+N_{\boldsymbol{\omega}}+N_{\boldsymbol{\eta}}}$ is defined as $\boldsymbol{\Theta}^a_{k-1} = [\boldsymbol{\Theta}^T_{k-1} \quad \boldsymbol{\omega}^T_{k-1} \quad \boldsymbol{\eta}^T_{k-1}]^T$ and $2N+1$ sigma vectors are initialized using augmented estimated states $\hat{\boldsymbol{\Theta}}^a_{k-1}$ and error covariance P^a_{k-1} as

$$\boldsymbol{\sigma}^a_{k-1} \coloneqq [\hat{\boldsymbol{\Theta}}^a_{k-1} \ \cdots \ \hat{\boldsymbol{\Theta}}^a_{k-1}] + \begin{bmatrix}\boldsymbol{0} & \sqrt{(N+\lambda)P^a_{k-1}} & -\sqrt{(N+\lambda)P^a_{k-1}}\end{bmatrix}$$

The generated sigma vectors are propagated through the process model as $\boldsymbol{\sigma}^{\boldsymbol{\Theta}}_{k|k-1} = \boldsymbol{A}\boldsymbol{\sigma}^{\boldsymbol{\Theta}}_{k-1} + \boldsymbol{\sigma}^{\boldsymbol{\omega}}_{k-1}$, where $\boldsymbol{\sigma}^{\boldsymbol{\Theta}}_{k-1}$ and $\boldsymbol{\sigma}^{\boldsymbol{\omega}}_{k-1}$ are the components of $\boldsymbol{\sigma}^a$ that correspond to the state variables and process noise, respectively. At time instants where measurements are not available, the estimated state and error covariance are predicted as

$$\begin{aligned}\hat{\boldsymbol{\Theta}}^-_k &= \sum_i^{2N} W_i^{(m)}\boldsymbol{\sigma}^{\boldsymbol{\Theta}}_{i,k|k-1},\\ P^-_k &= \sum_i^{2N} W_i^{(c)}(\boldsymbol{\sigma}^{\boldsymbol{\Theta}}_{i,k|k-1} - \hat{\boldsymbol{\Theta}}^-_k)(\boldsymbol{\sigma}^{\boldsymbol{\Theta}}_{i,k|k-1} - \hat{\boldsymbol{\Theta}}^-_k)^\top\end{aligned}$$

where $W_i^{(m)}$ and $W_i^{(c)}$ are weights associated with sigma vectors. After the sensor measurements are taken, the measurement model is propagated using $\boldsymbol{\gamma}_{k|k-1} = C_k(\boldsymbol{q})\boldsymbol{\sigma}^{\boldsymbol{\Theta}}_{k|k-1} + \boldsymbol{\sigma}^{\boldsymbol{\eta}}_{k-1}$. Next, the mean and covariance of the measurement and the cross-covariance of the state and measurement are calculated as

$$\begin{aligned}\hat{\boldsymbol{z}}^-_k &= \sum_i^{2N} W_i^{(m)}\boldsymbol{\gamma}_{i,k|k-1},\\ P_{\boldsymbol{z}\boldsymbol{z}_k} &= \sum_i^{2N} W_i^{(c)}(\boldsymbol{\gamma}_{i,k|k-1} - \hat{\boldsymbol{z}}^-_k)(\boldsymbol{\gamma}_{i,k|k-1} - \hat{\boldsymbol{z}}^-_k)^\top,\\ P_{\boldsymbol{\Theta}\boldsymbol{z}_k} &= \sum_i^{2N} W_i^{(c)}(\boldsymbol{\sigma}^{\boldsymbol{\Theta}}_{i,k|k-1} - \hat{\boldsymbol{\Theta}}^-_k)(\boldsymbol{\gamma}_{i,k|k-1} - \hat{\boldsymbol{z}}^-_k)^\top\end{aligned}$$

Finally, the filter gain is calculated as $L_k = P_{\Theta z_k} P_{z z_k}^{-1}$ and the estimated states $\hat{\Theta}_k$ and covariance P_k are updated as $\hat{\Theta}_k = \hat{\Theta}_k^- + L_k(z_k - \hat{z}_k^-)$ and $P_k = P_k^- - L_k P_{z z_k} L_k^\top$.

Acknowledgment

This work is funded in part by the NSF Dynamics, Control, and Systems Diagnostics program grant #2126818.

References

[1] Arctur, D., "U.S. Geological Survey (USGS)—Harvey Gaged Streamflow Timeseries," HydroShare, 2023. https://doi.org/10.4211/hs.4f4397545503c4ff4959c7e3703121940

[2] Lewis, F. L., Xie, L., and Popa, D., *Optimal and Robust Estimation: With an Introduction to Stochastic Control*, CRC Press, Boca Raton, FL, 2017, pp. 19–22.

[3] Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2006.

[4] Julier, S., and Uhlmann, J., "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422. https://doi.org/10.1109/JPROC.2003.823141

[5] Doucet, A., and Johansen, A. M., "A Tutorial on Particle Filtering and Smoothing : Fiteen Years Later," *The Oxford Handbook of Nonlinear Filtering*, edited by D. Crisan, and B. Rozovskii, Oxford Univ. Press, Oxford, NY, 2011, pp. 656–705, http://webcat.warwick.ac.uk/record=b2490036.

[6] St. Laurent, C., and Cowlagi, R. V., "Near-Optimal Task-Driven Sensor Network Configuration," *Automatica*, Vol. 152, June 2023, Paper 110966. https://doi.org/10.1016/j.automatica.2023.110966

[7] Fang, Z., Wang, Y., van Westen, C., and Lombardo, L., "Landslide Hazard Spatiotemporal Prediction Based on Data-Driven Models: Estimating Where, When and How Large Landslide May Be," *International Journal of Applied Earth Observation and Geoinformation*, Vol. 126, Feb. 2024, Paper 103631. https://doi.org/10.1016/j.jag.2023.103631

[8] Cisneros, D., Gong, Y., Yadav, R., Hazra, A., and Huser, R., "A Combined Statistical and Machine Learning Approach for Spatial Prediction of Extreme Wildfire Frequencies and Sizes," *Extremes*, Vol. 26, No. 2, 2023, pp. 301–330. https://doi.org/10.1007/s10687-022-00460-8

[9] LaValle, S. M., *Planning Algorithms*, Cambridge Univ. Press, Cambridge, MA, 2006.

[10] Patle, B., Pandey, A., Parhi, D., and Jagadeesh, A., "A Review: On Path Planning Strategies for Navigation of Mobile Robot," *Defence Technology*, Vol. 15, No. 4, 2019, pp. 582–606. https://doi.org/10.1016/j.dt.2019.04.011

[11] Popović, M., Ott, J., Rückin, J., and Kochenderfer, M. J., "Learning-Based Methods for Adaptive Informative Path Planning," *Robotics and Autonomous Systems*, Vol. 179, Sep. 2024, Paper 104727. https://doi.org/10.1016/j.robot.2024.104727

[12] Chen, J., Du, C., Zhang, Y., Han, P., and Wei, W., "A Clustering-Based Coverage Path Planning Method for Autonomous Heterogeneous UAVs," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 12, 2022, pp. 25,546–25,556. https://doi.org/10.1109/TITS.2021.3066240

[13] Chintam, P., Lei, T., Osmanoglu, B., Wang, Y., and Luo, C., "Informed Sampling Space Driven Robot Informative Path Planning," *Robotics and Autonomous Systems*, Vol. 175, May 2024, Paper 104656. https://doi.org/10.1016/j.robot.2024.104656

[14] Rückin, J., Jin, L., and Popović, M., "Adaptive Informative Path Planning Using Deep Reinforcement Learning for UAV-Based Active Sensing," *International Conference on Robotics and Automation (ICRA)*, Inst. of Electrical and Electronics Engineers, New York, 2022, pp. 4473–4479. https://doi.org/10.1109/ICRA46639.2022.9812025

[15] Qin, Y., Zhang, Z., Li, X., Huangfu, W., and Zhang, H., "Deep Reinforcement Learning Based Resource Allocation and Trajectory Planning in Integrated Sensing and Communications UAV Network," *IEEE Transactions on Wireless Communications*, Vol. 22, No. 11, 2023, pp. 8158–8169. https://doi.org/10.1109/TWC.2023.3260304

[16] Wen, T., Wang, X., Zheng, Z., and Sun, Z., "A DRL-Based Path Planning Method for Wheeled Mobile Robots in Unknown Environments," *Computers and Electrical Engineering*, Vol. 118, Sept. 2024, Paper 109425. https://doi.org/10.1016/j.compeleceng.2024.109425

[17] Xue, J., Zhang, S., Lu, Y., Yan, X., and Zheng, Y., "Bidirectional Obstacle Avoidance Enhancement-Deep Deterministic Policy Gradient: A Novel Algorithm for Mobile-Robot Path Planning in Unknown Dynamic Environments," *Advanced Intelligent Systems*, Vol. 6, No. 4, 2024, Paper 2300444. https://doi.org/10.1002/aisy.202300444

[18] Kamil, F., and and, M. Y. M., "Multilayer Decision-Based Fuzzy Logic Model to Navigate Mobile Robot in Unknown Dynamic Environments," *Fuzzy Information and Engineering*, Vol. 14, No. 1, 2022, pp. 51–73. https://doi.org/10.1080/16168658.2021.2019432

[19] Cai, X., Ancha, S., Sharma, L., Osteen, P. R., Bucher, B., Phillips, S., Wang, J., Everett, M., Roy, N., and How, J. P., "EVORA: Deep Evidential Traversability Learning for Risk-Aware Off-Road Autonomy," *IEEE Transactions on Robotics*, Vol. 40, July 2024, pp. 3756–3777. https://doi.org/10.1109/TRO.2024.3431828

[20] Wijegunawardana, I. D., Samarakoon, S. M. B. P., Muthugala, M. A. V. J., and Elara, M. R., "Risk-Aware Complete Coverage Path Planning Using Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 55, No. 4, 2025, pp. 2476–2488. https://doi.org/10.1109/TSMC.2024.3524158

[21] Ranieri, J., Chebira, A., and Vetterli, M., "Near-Optimal Sensor Placement for Linear Inverse Problems," *IEEE Transactions on Signal Processing*, Vol. 62, No. 5, 2014, pp. 1135–1146. https://doi.org/10.1109/TSP.2014.2299518

[22] Kreucher, C., Kastella, K., and Hero, A., III., "Information-Based Sensor Management for Multitarget Tracking," *Signal and Data Processing of Small Targets 2003*, Vol. 5204, 2003, pp. 480–489. https://doi.org/10.1117/12.502699

[23] Soderlund, A. A., and Kumar, M., "Optimization of Multitarget Tracking Within a Sensor Network via Information-Guided Clustering," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 2, 2019, pp. 317–334. https://doi.org/10.2514/1.G003656

[24] Kasper, K., Mathelin, L., and Abou-Kandil, H., "A Machine Learning Approach for Constrained Sensor Placement," *American Control Conference (ACC)*, Inst. of Electrical and Electronics Engineers, New York, 2015, pp. 4479–4484. https://doi.org/10.1109/ACC.2015.7172034

[25] Wang, Z., Li, H.-X., and Chen, C., "Reinforcement Learning-Based Optimal Sensor Placement for Spatiotemporal Modeling," *IEEE Transactions on Cybernetics*, Vol. 50, No. 6, 2020, pp. 2861–2871. https://doi.org/10.1109/TCYB.2019.2901897

[26] Tian, K., and Zhu, G., "Sensor Management Based on Fisher Information Gain," *Journal of Systems Engineering and Electronics*, Vol. 17, No. 3, 2006, pp. 531–534. https://doi.org/10.1016/S1004-4132(06)60091-1

[27] Wang, H., Yao, K., Pottie, G., and Estrin, D., "Entropy-Based Sensor Selection Heuristic for Target Localization," *3rd International Symposium on Information Processing in Sensor Networks*, Assoc. for Computing Machinery, New York, 2004, pp. 36–45. https://doi.org/10.1145/984622.984628

[28] Blasch, E. P., Maupin, P., and Jousselme, A.-L., "Sensor-Based Allocation for Path Planning and Area Coverage Using UGSs," *Proceedings of the IEEE. 2010 National Aerospace & Electronics Conference*, Inst. of Electrical and Electronics Engineers, New York, 2010, pp. 361–368. https://doi.org/10.1109/NAECON.2010.5712978

[29] Krause, A., Singh, A., and Guestrin, C., "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies," *Journal of Machine Learning Research*, Vol. 9, No. 2, 2008, pp. 235–284.

[30] Robbiano, C., Azimi-Sadjadi, M. R., and Chong, E. K. P., "Information-Theoretic Interactive Sensing and Inference for Autonomous Systems," *IEEE Transactions on Signal Processing*, Vol. 69, March 2021, pp. 5627–5637. https://doi.org/10.1109/TSP.2021.3067476

[31] Hoffmann, F., Charlish, A., Ritchie, M., and Griffiths, H., "Sensor Path Planning Using Reinforcement Learning," *IEEE 23rd International Conference on Information Fusion (FUSION)*, Inst. of Electrical and Electronics Engineers, New York, 2020, pp. 1–8. https://doi.org/10.23919/FUSION45008.2020.9190242

[32] Jang, D., Yoo, J., Son, C. Y., Kim, D., and Kim, H. J., "Multi-Robot Active Sensing and Environmental Model Learning With Distributed Gaussian Process," *IEEE Robotics and Automation Letters*, Vol. 5, No. 4, 2020, pp. 5905–5912. https://doi.org/10.1109/LRA.2020.3010456

[33] Park, M., and Oh, H., "Cooperative Information-Driven Source Search and Estimation for Multiple Agents," *Information Fusion*, Vol. 54,

Feb. 2020, pp. 72–84.
https://doi.org/10.1016/j.inffus.2019.07.007

[34] La, H. M., Sheng, W., and Chen, J., "Cooperative and Active Sensing in Mobile Sensor Networks for Scalar Field Mapping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 45, No. 1, 2015, pp. 1–12.
https://doi.org/10.1109/TSMC.2014.2318282

[35] MacDonald, R. A., and Smith, S. L., "Active Sensing for Motion Planning in Uncertain Environments via Mutual Information Policies," *The International Journal of Robotics Research*, Vol. 38, Nos. 2–3, 2019, pp. 146–161.
https://doi.org/10.1177/0278364918772024

[36] Li, T., Wang, C., Meng, M. Q.-H., and de Silva, C. W., "Attention-Driven Active Sensing With Hybrid Neural Network for Environmental Field Mapping," *IEEE Transactions on Automation Science and Engineering*, Vol. 19, No. 3, 2022, pp. 2135–2152.
https://doi.org/10.1109/TASE.2021.3077689

[37] Wu, K., Wang, H., Esfahani, M. A., and Yuan, S., "Achieving Real-Time Path Planning in Unknown Environments Through Deep Neural Networks," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 3, 2022, pp. 2093–2102.
https://doi.org/10.1109/TITS.2020.3031962

[38] Leong, A. S., Quevedo, D. E., Ahlén, A., and Johansson, K. H., "Network Topology Reconfiguration for State Estimation over Sensor Networks with Correlated Packet Drops," *IFAC Proceedings*, Vol. 47, No. 3, 2014, pp. 5532–5537.
https://doi.org/10.3182/20140824-6-ZA-1003.01846

[39] Ramachandran, G. S., Daniels, W., Matthys, N., Huygens, C., Michiels, S., Joosen, W., Meneghello, J., Lee, K., Cañete, E., Rodriguez, M. D., and Hughes, D., "Measuring and Modeling the Energy Cost of Reconfiguration in Sensor Networks," *IEEE Sensors Journal*, Vol. 15, No. 6, 2015, pp. 3381–3389.
https://doi.org/10.1109/JSEN.2015.2388857

[40] Grichi, H., Mosbahi, O., Khalgui, M., and Li, Z., "New Power-Oriented Methodology for Dynamic Resizing and Mobility of Reconfigurable Wireless Sensor Networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 48, No. 7, 2018, pp. 1120–1130.
https://doi.org/10.1109/TSMC.2016.2645401

[41] Cooper, B. S., and Cowlagi, R. V., "Interactive Planning and Sensing in Unknown Static Environments with Task-Driven Sensor Placement," *Automatica*, Vol. 105, July 2019, pp. 391–398.
https://doi.org/10.1016/j.automatica.2019.04.014

[42] Fang, J., Zhang, H., and Cowlagi, R. V., "Interactive Route-Planning and Mobile Sensing with a Team of Robotic Vehicles in an Unknown Environment," *AIAA Scitech 2021 Forum*, AIAA Paper 2021-0865, 2021.
https://doi.org/10.2514/6.2021-0865

[43] Poudel, P., and Cowlagi, R. V., "Coupled Sensor Configuration and Planning in Unknown Dynamic Environments with Context-Relevant Mutual Information-Based Sensor Placement," *2024 American Control Conference (ACC)*, IEEE Publ., Piscataway, NJ, 2024, pp. 306–311.
https://doi.org/10.23919/ACC60939.2024.10644304

[44] Poudel, P., and Cowlagi, R. V., "Reconfiguration Costs in Coupled Sensor Configuration and Path-Planning for Dynamic Environments," *AIAA SCITECH 2025 Forum*, AIAA Paper 2025-2069, 2025.
https://doi.org/10.2514/6.2025-2069

[45] Adurthi, N., Singla, P., and Majji, M., "Mutual Information Based Sensor Tasking with Applications to Space Situational Awareness," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 4, 2020, pp. 767–789.
https://doi.org/10.2514/1.G004399

[46] Rockafellar, R. T., and Uryasev, S., "Optimization of Conditional Value-at-Risk," *Journal of Risk*, Vol. 2, No. 3, 2000, pp. 21–42.
https://doi.org/10.21314/JOR.2000.038

[47] Poudel, P., DesRoches, J., and Cowlagi, R. V., "Actively Coupled Sensor Configuration and Planning in Unknown Dynamic Environments," *2025 American Control Conference (ACC)*, Inst. of Electrical and Electronics Engineers, New York, 2025, pp. 3067–3072.
https://doi.org/10.23919/ACC63710.2025.11107730

[48] Cover, T., and Thomas, J., *Elements of Information Theory*, Wiley-Interscience, Hoboken, NJ, 1991, p. 231.

[49] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L., "An Analysis of Approximations for Maximizing Submodular Set Functions—I," *Mathematical Programming*, Vol. 14, No. 1, 1978, pp. 265–294.
https://doi.org/10.1007/BF01588971

[50] Anderson, B. D. O., and Moore, J. B., "Detectability and Stabilizability of Time-Varying Discrete-Time Linear Systems," *SIAM Journal on Control and Optimization*, Vol. 19, No. 1, 1981, pp. 20–32.
https://doi.org/10.1137/0319002

[51] Cooper, B., and Cowlagi, R. V., "Decentralized Interactive Planning and Sensing in an Unknown Spatiotemporal Threat Field," *2019 Sixth Indian Control Conference (ICC)*, Inst. of Electrical and Electronics Engineers, New York, 2019, pp. 110–115.
https://doi.org/10.1109/ICC47138.2019.9123221

V. Cichella
Associate Editor