Coupled Sensor Configuration and Planning in Unknown Dynamic Environments with Context-Relevant Mutual Information-based Sensor Placement

Prakash Poudel* and Raghvendra V. Cowlagi†

***Abstract*— 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 sensor network separate from the mobile agent. For this problem, we present a new metric for optimal sensor placement that quantifies reduction in uncertainty in the path cost, rather than the environment state. This metric, which we call the context-relevant mutual information (CRMI), couples the sensor placement and path-planning problem. We propose also an iterative coupled sensor configuration and path planning (CSCP) algorithm. At each iteration, the 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. 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 decoupled method.**

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 within the workspace. It may be associated with various natural or artificial phenomena, such as wildfires, harmful gases in the atmosphere [1], or the perceived risk of adversarial attack.

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 and computational efforts would be directed at the *estimation* problem.

In practical applications, however, sensor networks may be constrained by size as well as energy usage. Consider, for example, a sensor network of unmanned aerial vehicles (UAVs) for surveillance of a threat field like wildfire in a large area. 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 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 is coupled due to the assumption of *proprioceptive* (fully onboard) sensing, as opposed to *exteroceptive* sensing considered here.

Of these areas, perhaps estimation is the most mature [2]. The literature on estimation involves different techniques including Kalman filter, maximum likelihood estimator [2], and 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 dynamical systems.

Path- 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 [6], [7]. Dijkstra's algorithm, A*, and their variants are branch-and-bound optimization algorithms that leverage heuristics to effectively steer the path search towards the goal. More recently, techniques based on reinforcement learning [8] and fuzzy logic [9] are reported.

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 [10]–[12] are studied. Machine learning-based sensor placement techniques are reported for efficient sensing with a minimal number of sensors and measurements as possible [13], [14]. Information-theoretic based sensor placement techniques utilize performance metrics that include Fisher information matrix (FIM) [15], entropy [16], Kullback-Leibler (KL) divergence [17], mutual information [18], and frame potential [10], for maximizing the amount of valuable information gathered from the surrounding environment. Another example of information-theoretic metric [19] utilizes two metrics, one associated with mutual information based on objection 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, sensing and path-planning method based on reinforcement learning [20] that evaluates the performance of a called Proximal Policy Optimization

(PPO) is reported.

In this paper we consider the problem of optimal sensor configuration coupled with path-planningin an unknown dynamic environment. Stated differently, 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 independent of the path-planning problem.

This is a relatively new research problem. Prior works by the second author and co-workers address this problem for static environments. A heuristic task-driven sensor placement approach called the interactive planning and sensing (IPAS) for static environments is reported in [21]. The IPAS method is shown to outperform 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 [22], also for static fields. Sensor placement for multi-agent path-planning based on entropy reduction is reported in [23].

The novelty of this work is that we consider a time-varying threat field and provide a new sensor placement metric for CSCP. Specifically, we develop the so-called context-relevant mutual information (CRMI) metric. Informally, this metric 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 would do). 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. We show that the CSCP-CRMI method significantly outperforms the decoupled method.

II. Problem Formulation

Let $\mathbb{R}$ represent the set of real numbers, and $\mathbb{N}$ the set of natural numbers. For any $N \in \mathbb{N}$, we denote by $[N]$ the set $\{1, 2, \ldots, N\}$, and by $\boldsymbol{I}_{(N)}$ the identity matrix of size N.

Consider a closed square region denoted by $\mathcal{W} \subset \mathbb{R}^2$ and referred as the *workspace*, within which the mobile agent and 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 $\boldsymbol{x}_i$, for each $i \in N_\mathrm{g}$. 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. We formulate the path planning problem for a actor as a graph search problem on a graph, $\mathcal{G} = (V, E)$ with $V = [N_\mathrm{g}]$ 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 $c : \mathcal{W} \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{>0}$, 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,

$$g((i,j),t) = c(\boldsymbol{x}_j, t), \text{ for } \quad i,j \in [N_\mathrm{g}], \quad (i,j) \in E \quad (1)$$

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, $J(\boldsymbol{v}) = \sum_{\ell=1}^{L} g((v_{\ell-1}, v_\ell), \ell\Delta t_s, \delta)$. The main problem of interest is to find a path with a minimum cost, $\boldsymbol{v}^*$. Since the threat field is unknown and is changing dynamically, estimation of the threat field in the environment is essential. A network of N_s sensors, where $N_\mathrm{s} \ll N_\mathrm{g}$, can be used to measure the intensity of threat. These sensor measurements are denoted $\boldsymbol{z}(\boldsymbol{x}, t; \boldsymbol{q}) = \{z_1(\boldsymbol{x}, t; \boldsymbol{q}), z_2(\boldsymbol{x}, t; \boldsymbol{q}), \ldots, z_{N_s}(\boldsymbol{x}, t; \boldsymbol{q})\}$ will be used to define the filter required for estimating the state of the dynamic system. Sensors are placed at distinct grid points, and the set of this grid points is called the *sensor configuration*, $\boldsymbol{q} = \{q_1, q_2, \ldots, q_{N_s}\} \subset [N_\mathrm{g}]$.

The threat field is modeled in parametric form as $c(\boldsymbol{x}, t) := 1 + \sum_{n=1}^{N_P} \theta_n(t)\phi_n(\boldsymbol{x}) = 1 + \boldsymbol{\Phi}(\boldsymbol{x})^\intercal \boldsymbol{\Theta}(t)$, with $\boldsymbol{\Phi}(\boldsymbol{x}) := [\phi_1(\boldsymbol{x}) \ldots \phi_{N_P}(\boldsymbol{x})]^\intercal$, and $\phi_n(\boldsymbol{x}) := \exp(-(\boldsymbol{x} - \overline{\boldsymbol{x}}_n)^\intercal(\boldsymbol{x} - \overline{\boldsymbol{x}}_n)/2a_n)$ representing the basis functions for each $n \in [N_P]$. Here, N_P represents the number of threat parameters involved to define the threat field. The values of the constants $a_n \in \mathbb{R}_{>0}$ and $\overline{\boldsymbol{x}}_n \in \mathcal{W}$ are prespecified and chosen in such a manner that the combined interiors of the significant support regions cover the entire workspace [21]. The parameter $\boldsymbol{\Theta}(t) := [\theta_1(t) \ldots \theta_{N_P}(t)]^\intercal$ is to be estimated.

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 a white process noise with $Q_c := \sigma_P \boldsymbol{I}_{(N_P)}$. The matrix A_c represents the evolution of threat parameters $\boldsymbol{\Theta}(t)$. This evolution model may be derived from an underlying physical model of the threat. For example, the solution to a heat diffusion equation, $\frac{\partial c}{\partial t} = \alpha(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2})$ can be approximated by $c(\boldsymbol{x}, t) = 1 + \boldsymbol{\Phi}(\boldsymbol{x})^\intercal\boldsymbol{\Theta}(t)$. It can be shown that the parameters $\boldsymbol{\Theta}(t)$ satisfy $\dot{\boldsymbol{\Theta}}(t) = \alpha\frac{\boldsymbol{\Phi}^\intercal}{|\boldsymbol{\Phi}|^2}\nabla^2\boldsymbol{\Phi}\boldsymbol{\Theta}(t)$, such that $A_c = \alpha\frac{\boldsymbol{\Phi}^\intercal}{|\boldsymbol{\Phi}|^2}\nabla^2\boldsymbol{\Phi}$.

To discretize the system, the continuous state transition matrix A_c and noise covariance matrix Q_c can be used to obtain the infinite series expansion of the sampled matrices as, $A := \boldsymbol{I}_{(N_P)} + A_c\Delta t_f + \frac{(A_c)^2(\Delta t_f)^2}{2!} + \ldots$ and

$Q := Q_c\Delta t_f + \frac{(A_cQ_c+Q_cA_c^\intercal)(\Delta t_f)^2}{2!} + \dots$ [2]. The time difference Δt_f is considered to be sufficiently small, thus the terms involving $(\Delta t_f)^2$ are neglected. 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 by $\boldsymbol{z}_k := c(\boldsymbol{x}_{q_k}, t) + \boldsymbol{\eta}_k = H_k(\boldsymbol{q})\boldsymbol{\Theta}_k + \boldsymbol{\eta}_k$, where

$$H_k(\boldsymbol{q}) = \begin{bmatrix}\boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{q}_{k,1}}) & \boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{q}_{k,2}}) \dots \boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{q}_{k,N_\mathrm{s}}})\end{bmatrix}^\intercal,$$

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

The threat parameters $\boldsymbol{\Theta}(t)$ are unknown quantities, and therefore we generate stochastic estimates with mean value $\widehat{\boldsymbol{\Theta}}(t)$ 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

$$J(\boldsymbol{v}) := L + \delta\textstyle\sum_{l=1}^{L} \boldsymbol{\Phi}(\boldsymbol{x}_l)^\intercal\boldsymbol{\Theta}(t).$$

The cost J becomes a random variable with distribution dependent on the estimate $\boldsymbol{\Theta}$.

An important characteristic associated with the convergence of the path-planning algorithm is the *risk* of the path. If J is Gaussian, then the risk of the path $\boldsymbol{v}$ is defined as $\rho(\boldsymbol{v}) := \widehat{J}(\boldsymbol{v}) + \sqrt{\mathrm{Var}[J(\boldsymbol{v})]}$ [24]. Since $N_\mathrm{s} \ll N_\mathrm{g}$, repeated iterative measurements are required.

Problem 1. *For a prespecified termination threshold, $\epsilon > 0$ 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 $\widehat{J}(\boldsymbol{v}^*)$ and such that Var$[J(\boldsymbol{v}^*)] \leqslant \epsilon$.*

III. Coupled Sensing and Planning

Coupled sensor configuration and path-planning (CSCP) is an 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 a new information metric that we call *context-relevant mutual information* (CRMI). Next, these sensor measurements are used to update the threat field estimates in an estimator. 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 what follows, we provide details of this iterative process, analysis, and an illustrative example.

Any estimator may be used to estimate the state parameters. We implement an Unscented Kalman Filter (UKF, [4]) for generality, although the examples provided in this paper are restricted to linear threat evolution models.

A. Mutual Information

1) Mutual Information of State and Measurement: For any time step k, the mutual information (MI) between the state $\boldsymbol{\Theta}_k$ and measurement $\boldsymbol{z}_k$ is defined by [25]

$$I(\boldsymbol{\Theta}_k; \boldsymbol{z}_k) := \int\int p(\boldsymbol{\Theta}_k, \boldsymbol{z}_k) \log\left(\frac{p(\boldsymbol{\Theta}_k, \boldsymbol{z}_k)}{p(\boldsymbol{\Theta}_k)p(\boldsymbol{z}_k)}\right) d\boldsymbol{\Theta}_k\, d\boldsymbol{z}_k,$$

where $p(\boldsymbol{\Theta}_k)$, $p(\boldsymbol{z}_k)$ and $p(\boldsymbol{\Theta}_k, \boldsymbol{z}_k)$ represent the probability density functions (PDFs) of state, measurement and a joint PDF of state and measurement, respectively. Considering the Gaussian PDFs, the joint PDF $p(\boldsymbol{\Theta}_k, \boldsymbol{z}_k)$ can be represented using the estimated states, estimated measurements and joint error covariance as

$$p(\boldsymbol{\Theta}_k, \boldsymbol{z}_k) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\Theta}_k \\ \boldsymbol{z}_k\end{bmatrix} : \begin{bmatrix}\widehat{\boldsymbol{\Theta}}_{k|k-1} \\ \widehat{\boldsymbol{z}}_k\end{bmatrix}, \begin{bmatrix}P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}} & P_{\boldsymbol{\Theta}\boldsymbol{z}_{k|k-1}} \\ P_{\boldsymbol{\Theta}\boldsymbol{z}_{k|k-1}}^\intercal & P_{\boldsymbol{z}\boldsymbol{z}_{k|k-1}}\end{bmatrix}\right).$$

Here, $P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}$ is the same as P_k^- in (4). The covariance of the measurement random vector $P_{\boldsymbol{z}\boldsymbol{z}_{k|k-1}}$ and cross covariance between the state and measurement random vectors $P_{\boldsymbol{\Theta}\boldsymbol{z}_{k|k-1}}$ depend on the sensor configuration $\boldsymbol{q}$. At each grid point, these covariances are determined as

$$P_{\boldsymbol{z}\boldsymbol{z}_{k|k-1}} = H_k(\boldsymbol{q})P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}H_k^\intercal(\boldsymbol{q}) + R_k, \tag{3}$$

$$P_{\boldsymbol{\Theta}\boldsymbol{z}_{k|k-1}} = P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}H_k^\intercal(\boldsymbol{q}). \tag{4}$$

An expression for the mutual information between the state and measurement variables can then be written as [26]:

$$I(\boldsymbol{\Theta}_k; \boldsymbol{z}_k(\boldsymbol{q})) = \frac{1}{2}\log\left(\frac{|P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}|}{|P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}} - P_{\boldsymbol{\Theta}\boldsymbol{z}_{k|k-1}}P_{\boldsymbol{z}\boldsymbol{z}_{k|k-1}}^{-1}P_{\boldsymbol{\Theta}\boldsymbol{z}_{k|k-1}}^\intercal|}\right). \tag{5}$$

2) Mutual Information between Path Cost and Measurement (CRMI): The mutual information between the path cost and the measurements, which we call CRMI is an information metric that provides most relevant information to a concurrent path-planning problem. Informally, the CRMI takes into account the locations that lies within the vicinity of the path planning, and the locations that lies far from the concurrent path are not considered.

For any path $\boldsymbol{v}$, the expected cost is $\widehat{J}(\boldsymbol{v}) := L + \delta\sum_{l=1}^{L}\boldsymbol{\Phi}(\boldsymbol{x}_l)^\intercal\widehat{\boldsymbol{\Theta}}(t)$. The joint PDF $p(J_k, \boldsymbol{z}_k)$ between the path cost and measurement variables is

$$p(J_k, \boldsymbol{z}_k) = \mathcal{N}\left(\begin{bmatrix}J_k \\ \boldsymbol{z}_k\end{bmatrix} : \begin{bmatrix}\widehat{J}_{k|k-1} \\ \widehat{\boldsymbol{z}}_k\end{bmatrix}, \begin{bmatrix}P_{JJ_{k|k-1}} & P_{J\boldsymbol{z}_{k|k-1}} \\ P_{J\boldsymbol{z}_{k|k-1}}^\intercal & P_{\boldsymbol{z}\boldsymbol{z}_{k|k-1}}\end{bmatrix}\right).$$

The variance of the path cost is

$$\begin{aligned}P_{JJ_{k|k-1}} &:= \mathbb{E}\left[\left(J(\boldsymbol{v}) - \widehat{J}(\boldsymbol{v})\right)^2\right] \\ &= \mathbb{E}\left[\left(\delta\sum_{l=1}^{L}\boldsymbol{\Phi}^\intercal(\boldsymbol{x}_{\boldsymbol{v}_l})\left(\boldsymbol{\Theta}(t) - \widehat{\boldsymbol{\Theta}}(t)\right)\right)^2\right], \\ &= \delta^2\sum_{l=1}^{L}\left(\boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_l})^\intercal P_{k_l}\boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_l})\right) \\ &\quad + 2\delta^2\sum_{l<m,\ \ l,m\in[L]}^{L}\left(\boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_l})^\intercal P_{k_{lm}}\boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_m})\right).\end{aligned} \tag{6}$$

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 $P_{k_{lm}}$ are determined by propagating the UKF prediction steps for a time steps for traversing between grid points. The covariance of the measurement and the

cross covariance between the path cost and the measurement random vector are formulated as:

$$P_{J\boldsymbol{z}_{k|k-1}} = \mathbb{E}\left[(\boldsymbol{z}-\widehat{\boldsymbol{z}})\left(J(\boldsymbol{v})-\widehat{J}(\boldsymbol{v})\right)\right]$$
$$= \delta\sum_{l=1}^{L}\left(\boldsymbol{\Phi}(\boldsymbol{x}_{\boldsymbol{v}_l})^{\intercal}P_{k_l}\right)H_k^{\intercal}(\boldsymbol{q}), \quad (7)$$

$$P_{\boldsymbol{z}\boldsymbol{z}_{k|k-1}} = H_k(\boldsymbol{q})P_{\boldsymbol{\Theta}\boldsymbol{\Theta}_{k|k-1}}H_k^{\intercal}(\boldsymbol{q}) + R_k. \quad (8)$$

Finally, the CRMI is calculated as

$$I(J_k;\boldsymbol{z}_k(\boldsymbol{q})) = \frac{1}{2}\log\left(\frac{|P_{JJ_{k|k-1}}|}{|P_{JJ_{k|k-1}} - P_{J\boldsymbol{z}_{k|k-1}}P_{\boldsymbol{z}\boldsymbol{z}_{k|k-1}}^{-1}P_{J\boldsymbol{z}_{k|k-1}}^{\intercal}|}\right). \quad (9)$$

B. CSCP Algorithm

The coupled sensing and planning algorithm described in Algorithm 1 initializes with $\widehat{\boldsymbol{\Theta}}_0 = \mathbf{0}$ 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.

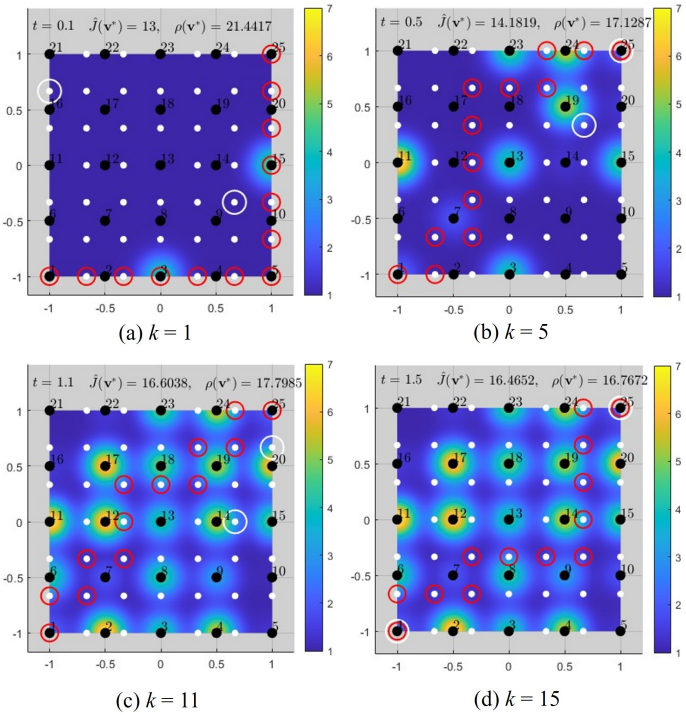
Algorithm 1: CSCP Algorithm

1 Set $k=0, \widehat{\boldsymbol{\Theta}}_0 = \mathbf{0}$, and $P_0 = \chi\boldsymbol{I}_{(N_P)}$
2 Initialize sensor placement $\boldsymbol{q}_0 \subset [N_\mathrm{g}]$
3 Find $\boldsymbol{v}_0^* = \arg\min(\widehat{J}_0(\boldsymbol{v}))$
4 **while** $Var[(J(\boldsymbol{v}_k^*)] \leqslant \epsilon$ **do**
5 Determine $I(J_k;\boldsymbol{z}_k(\boldsymbol{q}))$ per (5) or (9)
6 Find optimal sensor configuration $\boldsymbol{q}_k^* = \arg\max_{\boldsymbol{q}} I(J_k;\boldsymbol{z}_k(\boldsymbol{q}))$
7 Obtain new sensor measurements $\boldsymbol{z}_k(\boldsymbol{q}_k^*)$
8 Update $\widehat{\boldsymbol{\Theta}}_k, P_k$
9 Find $\boldsymbol{v}_k^* := \arg\min(\widehat{J}_k(\boldsymbol{v}))$
10 Increment iteration counter $k = k+1$
11 **end**

The description in Algorithm 1 is quite general, and its various steps can be implemented using different methods of choice by the user.

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 (6). The algorithm terminates whenever the variance of the path cost reduces below a prespecified threshold $\epsilon > 0$. 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 A* algorithm will suffice.

The optimal sensor configuration in Line 6 can be calculated by optimizing the metric chosen. In a decoupled method, we may optimize the standard MI in (5). In the proposed coupled method, we optimize the CRMI in (9). The method of optimization is left to the user, and is not the focus of this paper. We note that SMI is known to be submodular [18], and therefore, the proposed CRMI is also submodular. Greedy optimization algorithms provide near-optimal results for submodular functions. For a small number of grid points, we can determine $\boldsymbol{q}_k^*$ by mere enumeration.



(a) $k = 1$ (b) $k = 5$ (c) $k = 11$ (d) $k = 15$

Fig. 1. Visualization of CSCP-CRMI process for $N_P = 25$ and $N_\mathrm{g} = 49$.

With an optimal sensor placement, a new set of measurements is recorded, which are then used to update the state estimate for $\boldsymbol{\Theta}$. Yet again, the specific method of estimation is the user's choice. This iterative process continues until the termination criteria $\mathrm{Var}[(\widehat{J}(\boldsymbol{v})] \leqslant \epsilon$ is satisfied. The computational complexity of the CSCP algorithm mainly depends on the complexity of CRMI optimization, which further depends on the type of optimization method chosen.

IV. Results and Discussion

This section first provides an illustrative example of the proposed CSCP-CRMI method. Next, 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. Finally, we study the effects of varying numbers of sensors, threat parameters, and grid points on the CSCP-CRMI method. All numerical simulations are performed within a square workspace $\mathcal{W} = [-1,1]\times[-1,1]$ using non-dimensional units.

A. Illustrative Example

The implementation of CSCP-CRMI algorithm on an illustrative example is shown in Fig. 1. The number of threat parameters, grid points, and sensors are $N_P = 25$, $N_\mathrm{g} = 49$, and $N_\mathrm{s} = 2$, respectively. The initial and the goal points are represented by the bottom left and the top right grid points in the map. The threat parameters N_P, indicated by the black dots are uniformly spaced in the workspace and the white dots represent the grid points, N_g. The locations of the threat parameters, which are the center of the basis function $\boldsymbol{\Phi}$, do not change with time. The evolution of the threat

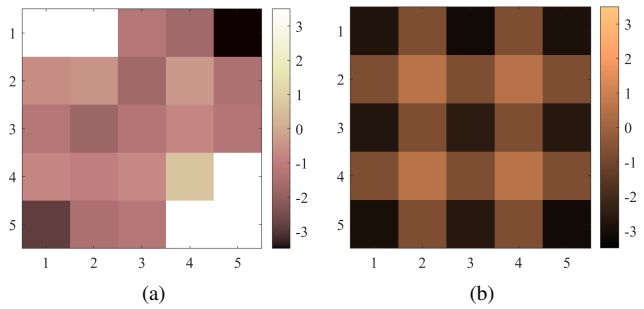


Fig. 2. log diagonal values of error covariance at final iteration with CSCP-CRMI (a) and with CSCP-SMI (b).

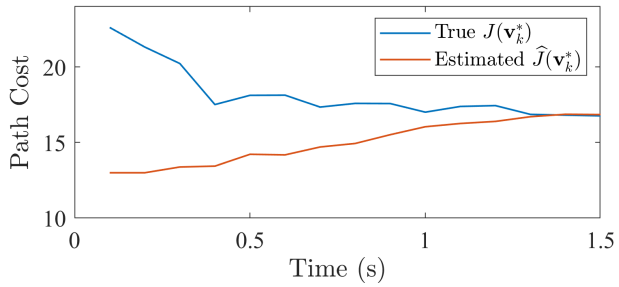


Fig. 3. Path cost of CSCP-CRMI method.

field estimate $\widehat{c}$ 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 the Fig. 1(a)-(d). For a specified threshold $\epsilon = 0.1$, the algorithm terminates at $k = 15$ iterations.

Figure 2(a) 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. 2 shows the *logarithms* of the diagonal values of P, which explains the negative values despite P being symmetric positive definite.

The white regions in Fig. 2(a) with high error covariance represents the areas where few if any sensors are placed throughout the execution of CSCP-CRMI. By contrast, the brown and black regions represent areas where sensors were placed to reduce the estimation error covariance values orders of magnitude below the white-colored regions. Compare Fig. 2 to the optimal path found in Fig. 1(d), and we find that the CSCP-CRMI sensor placement is such that areas close to the optimal path are those with low estimation error covariance. Note, crucially, the novelty of this approach: the optimal path is at first unknown, and that the sensor placement and path-planning are performed iteratively to arrive at these results.

The comparison between the true and estimated path cost is illustrated in Fig. 3. Upon termination, the true and estimated path costs are nearly identical, with $J(\boldsymbol{v}_k^*) = 16.46$ and $\widehat{J}(\boldsymbol{v}_k^*) = 16.77$, respectively. Figure 4 shows the convergence of the CSCP-CRMI algorithm. The path cost variance $\mathrm{Var}[(\widehat{J}(\boldsymbol{v}_k^*)]$ iteratively decreases and the algorithm terminates when $\mathrm{Var}[(\widehat{J}(\boldsymbol{v}_k^*)]$ falls below $\epsilon = 0.1$.

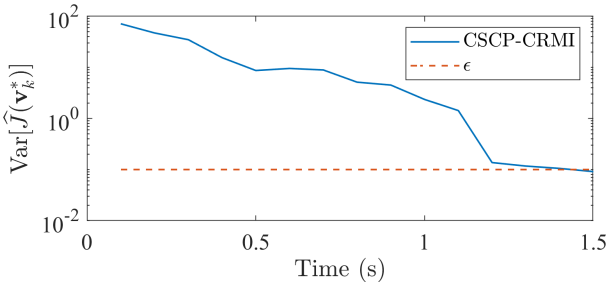


Fig. 4. Convergence of CSCP-CRMI algorithm.

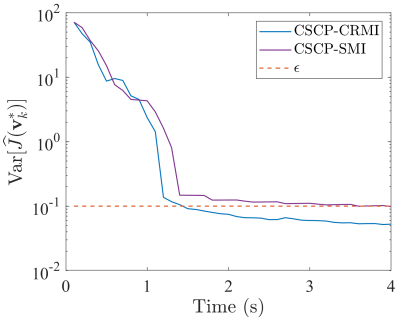
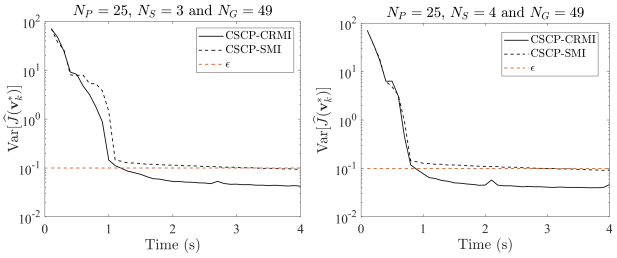


Fig. 5. Comparison of path cost variance for $N_\mathrm{s} = 2$..

B. Comparative Study

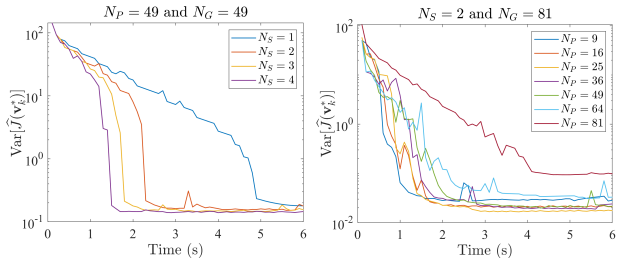
For comparison, now consider the execution of CSCP-SMI on the same example as discussed above. Figure 2(b) shows the estimation error covariance P of CSCP-SMI method. By contrast to Fig. 2(a) for CSCP-CRMI, we note here spatially uniform covariance values. This means that in CSCP-SMI, the sensors are 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, whereas some regions are not explored, the outcome of the path-planning algorithm is near-optimal.

Figure 5 shows a comparison between the path cost variance $\mathrm{Var}[J(\boldsymbol{v}_k^*)]$ of the two methods. Note that for $\epsilon = 0.1$, the CSCP-SMI algorithm requires 39 iterations for convergence, which is **160% larger** than the number of iterations required for CSCP-CRMI. Figure 6 shows similarly large differences in convergence rates with different numbers



(a) $N_\mathrm{s} = 3$. (b) $N_\mathrm{s} = 4$.

Fig. 6. Further comparisons of path cost variance.



(a) (b)

Fig. 7. Convergence of CSCP-CRMI algorithm for different numbers of sensors (a) and parameters (b).

of sensors.

Figure 7(a) shows the variation of path cost variance with varying number of sensors. For a specified number of threat parameters $N_P = 49$ and the grid points $N_\mathrm{g} = 49$, a 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 number of threat parameters N_P is shown in Fig. 7(b). For $N_\mathrm{s} = 2$ and $N_\mathrm{g} = 81$, a threat with $N_P = 9$ or 16 will lead to faster convergence relative to threats with $N_P = 64$ or 81. We also performed comparative analysis for specific number of sensors and parameters with varying number of grid points. It is observed that the path cost and the path cost variance remain unchanged for different number of grid points, which is due to appropriate scaling in the path cost formulation.

V. Conclusions

In this paper, we discussed a new metric of information for optimal sensor placement to capture coupling of the sensor placement problem with a path-planning problem. This metric, which we call the context-relevant mutual information (CRMI), quantifies the reduction in uncertainty in the path cost, rather than the environment state as is standard practice. We also presented a coupled sensor placement and path-planning algorithm that iteratively places sensors based on CRMI-maximization, updates the environment threat estimate, and plans paths with minimum expected cost. A comparative study between CSCP-CRMI and a decoupled CSCP-SMI method was performed. We showed via numerical simulation examples that the CSCP-CRMI algorithm converges in less than half as many iterations compared to CSCP-SMI algorithm.

Acknowledgments

This work is funded in part by the US NSF-DCSD-CMMI grant #2126818.

References

[1] M. A. Demetriou, N. A. Gatsonis, and J. R. Court, "Coupled controls-computational fluids approach for the estimation of the concentration from a moving gaseous source in a 2-D domain with a lyapunov-guided sensing aerial vehicle," *IEEE Transactions on Control Systems Technology*, vol. 22, pp. 853–867, May 2014.

[2] F. L. Lewis, L. Xie, and D. Popa, *Optimal and robust estimation: with an introduction to stochastic control theory*. CRC press, 2017.

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

[4] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, vol. 92, pp. 401–422, Mar. 2004.

[5] A. Doucet, A. M. Johansen, *et al.*, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[6] S. M. Lavalle, *Planning Algorithms*. Cambridge University Press, 2006.

[7] B. K. Patle, G. B. L, A. Pandey, D. R. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, pp. 582–606, August 2019.

[8] J. Rückin, L. Jin, and M. Popović, "Adaptive informative path planning using deep reinforcement learning for uav-based active sensing," in *International Conference on Robotics and Automation (ICRA)*, pp. 4473–4479, 2022.

[9] F. Kamil and M. Y. Moghrabiah, "Multilayer decision-based fuzzy logic model to navigate mobile robot in unknown dynamic environments," *Fuzzy Information and Engineering*, vol. 14, no. 1, pp. 51–73, 2022.

[10] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Transactions on Signal Processing*, vol. 62, pp. 1135–1146, Mar. 2014.

[11] C. M. Kreucher, K. D. Kastella, and A. O. H. III, "Information-based sensor management for multitarget tracking," in *Proceedings of SPIE*, vol. 5204, pp. 480–489, Dec. 2003.

[12] A. A. Soderlund and M. Kumar, "Optimization of multitarget tracking within a sensor network via information-guided clustering," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 317–334, Feb. 2019.

[13] K. Kasper, L. Mathelin, and H. Abou-Kandil, "A machine learning approach for constrained sensor placement," in *American Control Conference (ACC)*, pp. 4479–4484, Jul. 2015.

[14] Z. Wang, H. X. Li, and C. Chen, "Reinforcement learning-based optimal sensor placement for spatiotemporal modeling," *IEEE Transactions on Cybernetics*, vol. 50, pp. 2861–2871, Jun. 2020.

[15] T. Kangsheng and Z. Guangxi, "Sensor management based on fisher information gain," *Journal of Systems Engineering and Electronics*, vol. 17, pp. 531–534, 2006.

[16] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *3rd Int. Symp. Information Processing in Sensor Networks*, pp. 36–45, Apr. 2004.

[17] E. P. Blasch, P. Maupin, and A.-L. Jousselme, "Sensor-based allocation for path planning and area coverage using ugss," in *Proceedings of the IEEE 2010 National Aerospace & Electronics Conference*, pp. 361–368, 2010.

[18] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[19] C. Robbiano, M. R. Azimi-Sadjadi, and E. K. Chong, "Information-theoretic interactive sensing and inference for autonomous systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5627–5637, 2021.

[20] F. Hoffmann, A. Charlish, M. Ritchie, and H. Griffiths, "Sensor path planning using reinforcement learning," in *IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–8, 2020.

[21] B. S. Cooper and R. V. Cowlagi, "Interactive planning and sensing in unknown static environments with task-driven sensor placement," *Automatica*, vol. 105, pp. 391–398, Jul. 2019.

[22] C. S. Laurent and R. V. Cowlagi, "Near-optimal task-driven sensor network configuration," *Automatica*, vol. 152, Jun. 2023.

[23] J. Fang, H. Zhang, and R. V. Cowlagi, "Interactive route-planning and mobile sensing with a team of robotic vehicles in an unknown environment," in *AIAA Scitech 2021 Forum*, p. 0865, 2021.

[24] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of risk*, vol. 2, pp. 21–42, 2000.

[25] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.

[26] N. Adurthi, P. Singla, and M. Majji, "Mutual information based sensor tasking with applications to space situational awareness," *Journal of Guidance, Control, and Dynamics*, vol. 43, pp. 767–789, Apr. 2020.