HYBRID SENSOR NETWORKS FOR ACTIVE MONITORING:

COLLABORATION, OPTIMIZATION, AND RESILIENCE

A Dissertation
Presented to
The Academic Faculty

by

Yanjie Guo

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2022

HYBRID SENSOR NETWORKS FOR ACTIVE MONITORING:

COLLABORATION, OPTIMIZATION, AND RESILIENCE

Approved by:

Dr. Evangelos Theodorou, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Christopher Carr
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Brian C. Gunter
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Raghvendra V. Cowlagi
Aerospace Engineering Department
Worcester Polytechnic Institute

Dr. Pranay Seshadri
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: Dec 6th, 2022

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

DVI	Dynamic value iteration
HSN	Hybrid sensor networks
IoT	Internet of Thing
IRS	Inverse Reward Shaping
KF	Kalman filter
MDP	Markov decision process
MI	Mutual information
MPC	Model predictive control
MSPP	Mobile sensor path planning
R-HSN	Resilient hybrid sensor networks
SMAC	Spacecraft Maximum Allowable Concentrations
SSA	Static sensor allocation
SSN	Static sensor networks
UAV	Unmanned aerial vehicles
VI	Value iteration
WGC	Weighted Gaussian coverage
c	CO concentration
C	Cost budget
E_t	Exploration index in the reward function
G	Gaussian kernel
ker	Scaling kernel
K	Coverage index
L	A finite set of all possible sensor locations

n_a	Number of mobile sensors
n_f	Number of static sensors
p_f	Failure probability of static sensors
p_a	Failure probability of mobile sensors
Q	Q-value function
R	Reward function
s	Choice of static sensor locations
T	Temperature
U	Sensor coverage or utility metric
U_t	Estimation uncertainty
V	Value function
$\vec{V}$	Ventilation velocity field
w	Anomaly risk distribution
α	Thermal diffusivity
β	Sensor portfolio mix
γ	Cost ratio of a mobile sensor to a static sensor
Φ_t	Anomaly risk
Φ	Dummy variable for sensor deployment method

SUMMARY

Hybrid sensor networks (HSN) consist of both static and mobile sensors deployed to fulfill a common monitoring task. The hybrid structure generalizes the network's design problem and offers a rich set of possibilities for a host of environmental monitoring and anomaly detection applications. HSN also raise a new set of research questions. Their deployment and optimization provide unique opportunities to improve the network's monitoring performance and resilience. This thesis addresses three challenges associated with HSN related to the *collaboration*, *optimization*, and *resilience* aspects of the network. Broadly speaking, these challenges revolve around the following questions: (1) how to collaboratively allocate the static sensors and devise the path planning of the mobile sensors to improve the monitoring performance? (2) how to select and optimize the sensor portfolio (the mix of each type of sensors) under given cost constraints? And (3) how to embed resilience in a HSN to sustain the monitoring performance in the face of sensor failures and disruptions?

In part I, *collaboration*, this thesis develops a novel deployment strategy for HSN. The strategy solves the static sensor allocation problem, the mobile sensor path planning problem, and most importantly, the collaboration between these two types of sensors. Previous research in this area has addressed these problems separately in simplified environments. In this thesis, a collaborative deployment strategy of HSN is developed to improve the ultimate monitoring performance in complex environments with obstacles and non-uniform risk distribution.

In part II, *optimization*, this thesis addresses the HSN sensor portfolio selection problem. It investigates the tradeoff between the static and mobile sensors to achieve the optimal monitoring performance under different cost constraints. Previous research in this area has studied the optimization problem for networks with a single type of sensor. In this thesis, a general optimization problem is formulated for HSN with static and mobile sensors and solved to identify the optimal portfolio mix and its main drivers.

In part III, *resilience*, this thesis identifies monitoring resilience as a key feature enabled by HSN. This part focuses on the performance degradation of HSN in the presence of sensor failures and disruptions, and it identifies the means to embed resilience in a HSN to mitigate this performance degradation. Monitoring resilience is achieved by accounting for potential sensor failures in the deployment strategy of both static and mobile sensors through a novel, carefully designed probability sum technique. Previous research in this area has examined the reliability problem from a coverage point of view. This thesis extends the scope of investigation of HSN from reliability to resilience, and it shifts the focus from coverage considerations to the actual monitoring performance (e.g., detection time lag) and its resilience in the face of disruptions.

To demonstrate and validate this novel perspective on HSN and the associated technical developments, this thesis focused on two examples of fire detection in a multi-room apartment using temperature sensors and CO leak detection in a 3D space station module with ventilation system. Three metrics are adopted as the ultimate monitoring performance, namely the detection time lag, the anomaly source localization uncertainty, and the state estimation error. A simulation environment based on the advection-conduction heat propagation model is developed for the computational experiments. The

results (1) demonstrate that the optimal collaborative deployment strategy allocates the static sensors at high-risk locations and directs the mobile sensors to patrol the rest of the low-risk areas; (2) identify a set of conditions under which HSN significantly outperform purely static and purely mobile sensor networks across the three performance metrics here considered; and (3) establish that while sensor failures can considerably degrade the monitoring performance of traditional static sensor networks, the resilient deployment of HSN drastically reduces the performance degradation.

CHAPTER 1. INTRODUCTION

1.1 Backgrounds and Motivations

Sensor networks are the focus of an increasingly vigorous research area with rich applications in environmental monitoring and anomaly detection [1,2], maritime search and rescue [3], target tracking [4], precision agriculture [5], geoscience systems [6–9], and within the broad context of the Internet of Thing (IoT) [10–13]. For example, Hart and Martinez [6] summarized several large-scale applications of environment sensor networks including global seismographic network with seismometer accelerometer, the Georgia automated environmental monitoring network with meteorological data, and web based hot spot modelling using geos multispectral imaging. Nittel [7] discussed geo-sensor networks applications including continuous agricultural observation systems for plant health and growth circumstances, and a volcano network to detect and measure tremor events. Inoue et al. [8] used an ocean bottom pressure sensor network to estimate tsunami sources. Barnes et al. [9] studied Earth-ocean process using a network of heterogeneous sensors including seismic, tsunami, and chemical sensors.

With the recent development in low-cost robots, more applications consider the adoption of hybrid sensor networks (HSNs) which exploit mobile sensors in conjunction with static sensors smartly distributed across the environment to be monitored [14–20]. Traditional static sensors are low-cost but cannot be relocated once deployed. Recent mobile sensors are more expensive, but their mobility enables them to have a dynamic coverage area and to smartly adjust their monitoring and search patterns as new risks or priorities emerge. When carefully designed, HSNs can outperform purely static sensor

networks by a collaboration between the two types of sensors. This observation prompts some general questions, for example, how should the static sensors be allocated and the path planning of the mobile sensors formulated to achieve a collaborative optimal monitoring performance? What is the optimal sensor portfolio mix (of static and mobile sensors) and what is it contingent upon? And how to embed resilience in a HSN to sustain the monitoring performance in the face of sensor failures and disruptions? These are some of the questions addressed in this work. A brief review of the literature on hybrid sensor network deployment is presented next to provide context for the present work.

The design of HSNs can be naturally decomposed into two parts: a static sensor allocation part and a mobile sensor path planning part. The aforementioned questions alluded to the need for a collaborative combination of these two parts. Two state-of-the-art information theoretic metrics for sensor placement are entropy and mutual information (MI). There are a lot of recent sensor placement applications using these two metrics [21–29]. For example, Ariga et al. [21] proposed a MI-based sensor placement approach for spatial sound field recording; Bhattacharyya and Beck [22] studied structural sensor placement for a 50-story shear building model using MI; Pei et al. [25] examined beam and bridge structural sensor placement using entropy; Turko et al. [26] studied temperature and salinity fields around the Barents Sea and the Svalbard group of islands using entropy-based sensor placement; Wang et al. [27] studied sensor placement for obstacle detection based on relative entropy; and Hu et al. [28] used entropy-based sensor placement for water distribution systems with sensor failures. These two metrics optimize sensor placement based on a given or estimated covariance matrix for a given process. They are suitable for estimation process but not for fast anomaly detection. Later results and analysis also show

that they are not suitable for non-uniform risk distribution. For path planning, one state-of-the-art algorithm is model predictive control (MPC). The core idea of MPC is to make predictions for a time horizon in the future based on which the control or decision making is optimized. Some recent examples include autonomous driving with obstacles and path planning for UAVs [30–32].

Most previous works considered the collaboration between mobile sensors only [33,34], or just used mobile sensors to assist a group of pre-allocated static sensor [35,36]. For example, Popescu et al. [33] examined the collaboration between unmanned aerial vehicles (UAVs). Sun et al. [36] focused on the collaboration between mobile sensors for spatiotemporal coverage. Freitas et al. [35] adopted UAVs to check some area of interest after an alarm triggered by the static sensors. While these works make important contributions, they do not consider simultaneously the joint static sensor allocation and mobile sensor path planning problems, or the potential collaboration between them.

For sensor systems with a long mission span and limited maintenance opportunity such as those for future deep space habitats, sensor failures can be a critical issue. Previous works that focused on the reliability of sensor networks addressed important topics such as network connectivity and coverage [37]. These aspects of reliability can be affected by several factors including hardware choices and network deployment strategies. For example, Dong et al. [38] optimized the network lifetime and transport delay with reliability constraints. AboElFotoh et al. [39] examined the reliable message delay problem for wireless distributed sensor networks. Dagdeviren et al. [40] designed a depth-first search algorithm with connected dominating set to efficiently find the critical nodes that break the connectivity of the network when they fail. Deif and Gadallah [41] proposed a

reliability metric based on coverage and connectivity, subject to four types of component failures. Kabadurmus and Smith [42] proposed a network reliability metric considering the link capacity and rerouting options. Chakraborty et al. [43] proposed a different reliability metric based on the minimal path for multistate nodes. In a recent survey, Yue and He [44] summarized the recent literature on the reliability of mobile wireless sensor networks, mainly from the communication perspective. The mobility of the sensor or sink nodes are used in these works to improve the network's connectivity. The aforementioned works have made important contributions to the reliability evaluation of sensor network from a communication and coverage perspectives. A more direct performance assessment in terms of the actual monitoring objectives, such as detection time lag, is missing in the literature.

Another important aspect that has not received proper attention in the literature is the environment models. Previous works generally assumed a simple environment and focused on the sensor networks. For example, ref. [45,46] investigated the search and localization problem of a stationary target with static measurement field. Research in [14,15,47,48] assumed a static environment with uniform event occurrence probability distribution. The work reported in [15,16,47,49,50] adopted the commonly used disk-shaped Boolean or probabilistic sensing model [51]. However, these simplified environment models may not reflect the dynamics in real applications. For instance, the temperature field is usually unsteady in an environment [52]; the seismic hazard map is highly non-uniform [53]; and the measurements of gas sensors are affected by dynamic airflow [54]. Recent studies [55,56] have pointed out the need to examine more complex environments with time-varying environmental fields, non-uniform point-of-interest or risk maps, and physical sensing models to meaningfully assess the performance of monitoring sensor networks.

1.2 Current Challenges and Research Objectives

There are three challenges in the current research on HSN. First, the deployment strategy of HSN does not fully consider the collaboration between the static and mobile sensors, and no optimization has yet been investigated for the portfolio mix of the two types of sensors under cost constraints. Second, research in this area has examined the network reliability problem from a communication and coverage point of view. Different metrics were adopted, such as communication efficiency and coverage percentage, to evaluate the network performance. More direct assessments are missing in terms of the ultimate monitoring performance for specific applications, such as anomaly detection time. Third, the environment models adopted are relatively simple with static fields, uniform risk distribution, and a disk-shaped sensing model. Real monitoring applications have more complex environments, which rarely conform to these assumptions. More advanced deployment strategies are needed to cope with realistic environments for useful applications.

This thesis focuses on the collaboration, optimization, and resilience of hybrid sensor networks (HSNs) in realistic environments. This work addresses the aforementioned open issues for general sensor networks. More specifically, a collaborative deployment strategy of HSN is developed to improve the ultimate monitoring performance in complex environments with obstacles and non-uniform risk distribution. A general optimization problem is formulated for HSN with static and mobile sensors and solved to identify the optimal portfolio mix and its main drivers. The scope of investigation of HSN is extended from reliability to resilience, and this work shifts the focus from coverage considerations to the actual monitoring performance (e.g., detection time lag) and its resilience in the face

of disruptions. A resilient upgrade of the previous deployment strategy is developed to mitigate potential performance degradation when the network sensors are disrupted. More specific research objectives are described in the corresponding chapters. To demonstrate and validate the novel perspective on HSN and the associated technical developments, this thesis focused on two examples of fire detection in a multi-room apartment using temperature sensors and CO leak detection in a 3D space station module with ventilation system.

1.3 Outline of The Thesis

The remainder of this thesis is organized as follows. Chapter 2 introduces the collaborative deployment strategy for hybrid sensor networks. This chapter provides the problem formulation, an overview of the HSN architecture, and its technical details. It also provides the simulation environment and performance metrics to evaluate the HSN, as well as the example result of a specific scenario. Chapter 3 focuses on the optimization on the sensor portfolio mix between the two types of sensors. This chapter provides the optimization problem statement and detailed results of cost-performance tradeoff analysis and the effect of mobile sensor speed on the optimal sensor portfolio. Chapter 4 presents the resilience aspect and consideration of HSN. This chapter provides the resilience problem statement, the upgraded resilient deployment strategy, and the detailed results that quantifies the resilience improvement in terms of the ultimate monitoring performance. Chapter 5 focuses on the conclusion, the contribution of this thesis, and future work.

CHAPTER 2. COLLABORATION

The objective of this chapter can be succinctly expressed as follows: to design a hybrid sensor network (HSN) deployment strategy by collaborative static sensor allocation and mobile sensor path planning (O1). The design of the deployment strategy can be formatted as an optimization problem:

$$\max_{\Phi} \quad \text{performance}\left(\Phi\left(n_f, n_a\right)\right) \tag{1}$$

where n_f and n_a are the number of static and mobile sensors; Φ is a dummy variable for the deployment method which includes the static sensor allocation and mobile sensor path planning; the performance is a general objective function to be optimized. For example, it can be minimizing the detection time lag for anomaly detection or maximizing the localization accuracy for target search. Detailed discussion of our performance metrics will be provided in section 2.4.

Note that the focus of this chapter is on the first research objective (O1). It is assumed that the number of sensors is given and the deployment strategy Φ is the design target. The aim is to optimize the network performance through collaboration and shared measurements between the static and mobile sensors. A high-level overview of the deployment method is provided next, and its technical details are discussed in the following subsections.

2.1 Overview

The HSN deployment method consists of two parts, the static sensor allocation (SSA) and the mobile sensor path planning (MSPP). The high-level HSN architecture is shown in Figure 1.

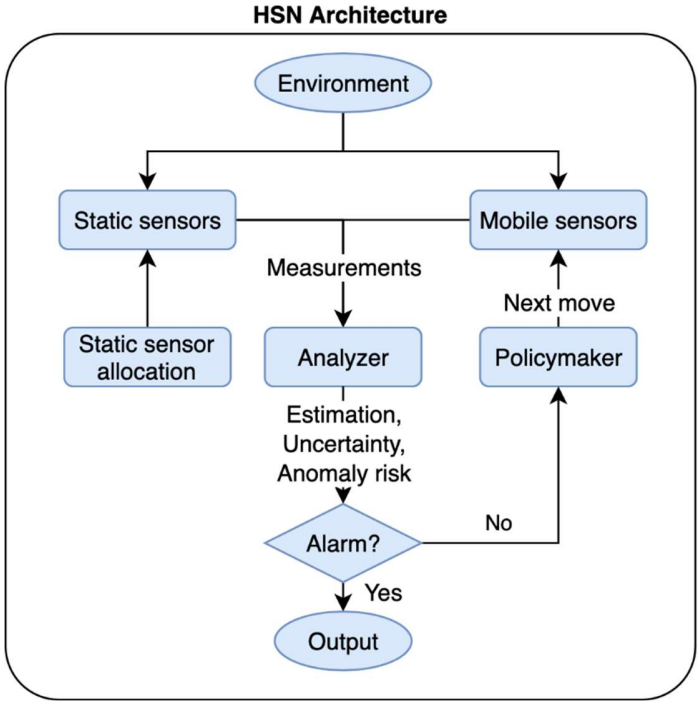


Figure 1. The architecture of hybrid sensor network deployment method.

Most realistic environments have a non-uniform risk map. For example, cooking in the kitchen, electric heaters in the bedrooms, and dryers (wherever they are placed) exhibit a higher risk of fire. With this observation, we achieve the collaboration between static and mobile sensors by allocating the static ones at high-risk locations and utilizing the mobile ones to patrol the rest of the area. In the absence of a risk map or for an unknown environment, we default to a uniform risk map, which is a special case of our more general

solution. The following discussion addresses HSN deployment problem with a uniform and non-uniform risk map.

First, we design a weighted Gaussian coverage (WGC) metric to solve the SSA problem. The WGC metric incorporates considerations of the geometry and risk distribution of the environment. Second, the raw measurements from the static and mobile sensors are used to solve the MSPP problem online. There are two main components in the MSPP process, an analyzer and a policymaker. At each time step, the analyzer processes the raw measurements and extracts meaningful information, including full state estimation of the environment, estimation uncertainty level, and the risk or likelihood of anomaly at particular locations. This information represents a high-level understanding of the environment and is used for anomaly detection and MSPP. If the state estimation or likelihood of anomaly breaches a predefined threshold, an alarm is triggered. Otherwise, the policymaker determines the next optimal moves for the mobile sensors based on a Markov decision process (MDP) model. The policymaker formulates a parameterized reward function based on the extracted information from the analyzer. A Dynamic Value Iteration (DVI) method is developed to solve the MDP. Next, we introduce the technical details of each component.

2.2 Static Sensor Allocation with Weighted Gaussian Coverage

In this subsection, we focus on the first step of our deployment method, the static sensors allocation. Previous works studied the uniform sensor allocation problem [57,58]. In our method, we consider a more general problem of distributing sensors to cover high-interest areas according to a non-uniform risk map, should it be available, and a default

uniform map if not. The general problem can be formulated as a discrete optimization problem given by Eq. 2:

$$\max_{s} U(s)$$
$$s = \{s_1, s_2, \dots, s_n\}, s_i \in L, i = 1,2, \dots, n < |L| \tag{2}$$

Let L be a finite set of all possible sensor locations (through discretizing the environment into grids for example). We need to choose n locations from L to maximize a metric of sensor utility U. We denote the choice of these n locations as $s = \{s_1, s_2, \dots, s_n\}$. Note that $n < |L|$, and this can be guaranteed by discretizing the environment into smaller grids.

We first need to define sensor utility U such that it reflects the ultimate monitoring performance. To address the drawbacks of the entropy and mutual information approaches, we propose the weighted Gaussian coverage (WGC) as our metric, as shown in Eq. 3-5:

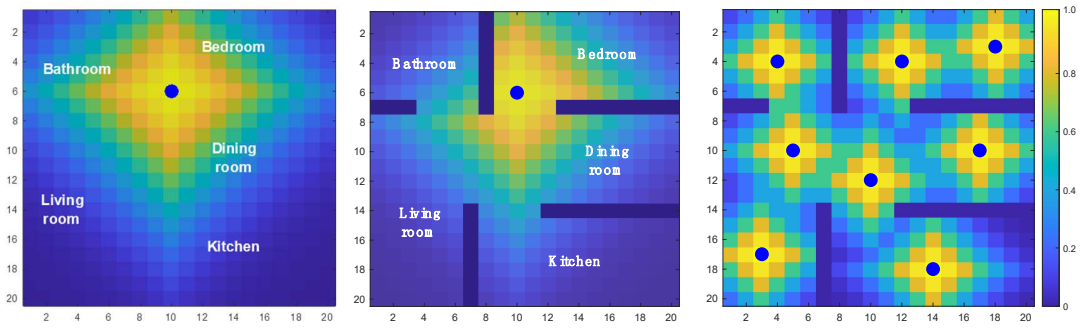
$$G(l; s_i, \sigma) = exp\left(-\frac{d(l, s_i)^2}{2\sigma^2}\right), \qquad l, s_i \in L \tag{3}$$

$$K(l) = \max_{s_i \in s} G(l; s_i, \sigma) \tag{4}$$

$$U(s) = \sum_{l \in L} w(l) \cdot K(l) \tag{5}$$

First, at each sensor location s_i, we place a Gaussian kernel $G(l; s_i, \sigma)$ with standard derivation σ to represent its capture area, as shown in Eq. 3. This kernel G generally

defines how well another location l can be sensed by a sensor located at s_i. Note that we use a distance function $d(l, s_i)$ to represent the travel distance between these two locations accounting for potential obstacles in the environment. The travel distance can be calculated by path search algorithms such as A* [59]. Also note that the travel distance is calculated in L_1 norm because it is assumed that the mobile sensors, discussed later in section 2.4, can only move towards the four neighbor grids. Other norms such as L_2 might be better depending on the moving mechanism and the specific environment dynamics. Fig. 2a-2b shows an example kernel in an environment with and without obstacles. This choice of using travel distance in the kernel reflects in part some monitoring performance metrics such as detection time lag and distance between sensor and anomaly to be detected. Second, at each location l, we take the maximum kernel value among all sensors as an index of coverage $K(l)$ to indicate how well this location can be sensed by all sensors, as shown in Eq. 4. Finally, we use the weighted sum of the coverage index over the whole space according to a predefined spatial risk probability distribution $w(l)$, as shown in Eq. 5. Notice that the risk distribution is assumed to be available and correct. This can be done through expert estimation of the environment or from history data. When the environment is unknown or has a uniform risk map, we set a uniform weight. If the risk distribution is incorrect or does not match the true distribution, the performance of WGC will be negatively affected. For a non-uniform risk map, our WGC method will allocate static sensors closer to the high-risk locations. Fig. 2c shows an example allocation of 8 sensors in the same environment. Note that the WGC metric is also submodular as the entropy and MI metrics. The proof and a greedy version of WGC is given in the Appendix A.



(a) (b) (c)

Figure 2. Example kernel and coverage index functions for sensors in a $20 \times 20\ m^2$ apartment. The blue dots represent sensor locations. Note that in (a-b) the kernel function in the bathroom is different with and without the wall. (c) Example coverage index function with 8 sensors

2.3 Mobile Sensor Path Planning

After the static sensors are allocated, a Markov decision process model is developed to solve the MSPP problem. The next optimal moves for the mobile sensors are determined by maximizing the cumulative reward given by a parameterized function formulated with extracted information from the raw measurements. The technical details of the two main components, the analyzer and the policymaker, are discussed next.

2.3.1 Analyzer

The measurements taken by the moving sensors are processed by the analyzer to derive more instructive information, namely the full state estimation of the habitat for the sensed quantity (here temperature at every point in the habitat), the uncertainty analysis in state estimation, and the anomaly risk (here the likelihood of fire). A flowchart of the analyzer is shown in Fig. 3. We use Kalman Filter (KF) to generate the first two pieces of

information, and the cumulative probability for the last. The details are provided in the following subsections.

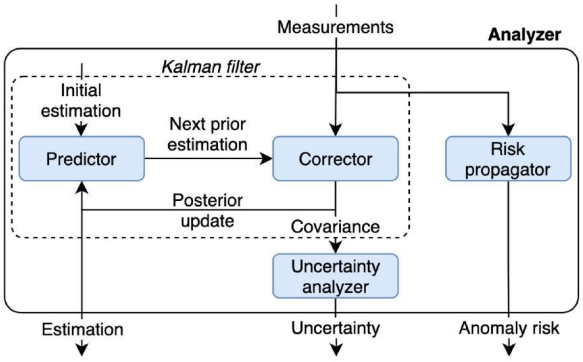


Figure 3. The analyzer flowchart within the HSN architecture. The analyzer takes the raw measurements as input and extracts three pieces of information, the full state estimation, the uncertainty of the estimation, and the probability of having anomalies.

2.3.1.1 Full State Estimation and Uncertainty Analysis

Kalman filter, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time with statistical noise to produce estimation and the associated covariance of the future trend of the temporal signal. The KF has numerous applications in research areas, such as guidance and navigation and control theory [60,61]. We use KF in the analyzer to obtain the full state estimation and the associated uncertainty level. Here, we provide a brief introduction of KF, and more details are referred to [62].

The KF is commonly used for state estimation and prediction for a discrete linear time-invariant (LTI) system with state disturbance and measurement noise. A typical LTI system can be represented as follows:

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + v_t \tag{6}$$

where x is the state variable, u is the control input, w is the state disturbance, y is the measurement, and v is the measurement noise. A, B, C are the system matrices for the state transition, control, and measurement. The noise is assumed with a centered Gaussian distribution with fixed covariance R and Q, as follows:

$$E(v) = E(w) = 0,$$
$$R = E(vv^T), Q = E(ww^T) \tag{7}$$

The purpose of KF is to minimize the error of the posterior state estimation $e = \hat{x} - x$, where $\hat{x}$ denotes the estimated state, and provide the covariance of the error $P = E(ee^T)$, based on the system parameters and measurements. The posterior state estimation consists of five equations, Eq. 8-12, which can be divided into two parts, prediction and correction. The prediction process propagates the current estimation to the next time step, as shown in:

$$\hat{x}'_{t+1} = A\hat{x}_t + Bu_t \tag{8}$$

$$P'_{t+1} = AP_tA^T + Q \tag{9}$$

where the $\hat{x}'$ and P' denote measurement predictions of the state variable and error covariance, respectively. After taking the next measurement, the correction process updates the prediction, as shown in:

$$\hat{x}_t = \hat{x}'_t + K_t(y_t - C\hat{x}'_t) \tag{10}$$

$$P_t = (I - K_t C)P'_t(I - K_t C)^T + K_t R K_t^T \tag{11}$$

where K_t, the Kalman gain, is given by:

$$K_t = P'_{t-1}C^T(CP'_{t-1}C^T + R)^{-1} \tag{12}$$

The diagonal of the covariance matrix P is the variance of the estimation error at each state. Assuming a Gaussian process, we can compute the 95% uncertainty U_t with:

$$U_t = 2\sqrt{diag(P_t)} \tag{13}$$

With the Kalman filter, we obtain the full state estimation $\hat{x}_t$ and the associated uncertainty U_t as the first two outputs of the analyzer. With this information, we have an overall understanding of the whole environment. For example, a high temperature estimation can indicate a potential fire or heat source, while a high uncertainty level at a particular location can indicate the need for further investigation of that location. The policymaker discussed shortly will leverage this information to determine efficient next moves for the sensors.

2.3.1.2 Anomaly Risk

The last output of the analyzer, the anomaly risk, is defined here as the cumulative probability of having anomalies (here a fire) at any particular location in the habitat. We obtain this information Φ_t using:

$$\Phi_t = 1 - (1 - \phi_A)^{t - t_x} \tag{14}$$

where t and t_x are the current time and the most recent time of visit at location x, and ϕ_A is the probability of having anomalies over a unit time period. This information keeps track of how long each location has not been visited (for $t - t_x$). The longer it is, the higher risk we have at this location of having an anomaly.

Note that the anomaly risk may seem somewhat similar but is intrinsically different uncertainty compared with the measurement uncertainty. The former comes from the uncertainty of the environment dynamics, whereas the latter comes from the estimation error and measurement noise. Keeping both uncertainties not only provides more information of the system as output for external uses, but also benefits the policymaker to generate better policy for the moving sensors.

So far, in the analyzer, we have reconstructed the full state estimation $\hat{x}_t$ and the estimation uncertainty U_t from the raw measurements, and we derived the anomaly risk Φ_t from the visiting history of the moving sensors. This information represents a better level understanding of the environment compared with the raw measurements. It can be shared, as noted previously, with external systems such as the ground station, a higher-level supervisory system, or the astronauts onboard if they are present. In HSN, this information is passed to the policymaker for path planning of the moving sensors, as discussed next.

2.3.2 Policymaker

Based on the three outputs from the analyzer, namely the state estimation, the estimation uncertainty, and the anomaly risk, the policymaker determines the next optimal move that will minimize system uncertainty and result in shorter detection time lag if there is an anomaly. We first introduce the decision-making process for a single agent, and then expand it for a multi-agent system using decentralized approaches.

A flowchart of the policymaker is shown in Fig. 4. The problem is formulated as an optimal Markov Decision Process (MDP). It is realized by first formulating a reward function based on the three outputs, and second by solving the optimization problem that maximizes the reward using a Dynamic Value Iteration (DVI) method. We also append an Inverse Reward Shaping (IRS) process to the policymaker to adjust the reward function parameters in light of the system performance metrics. The details are provided next.

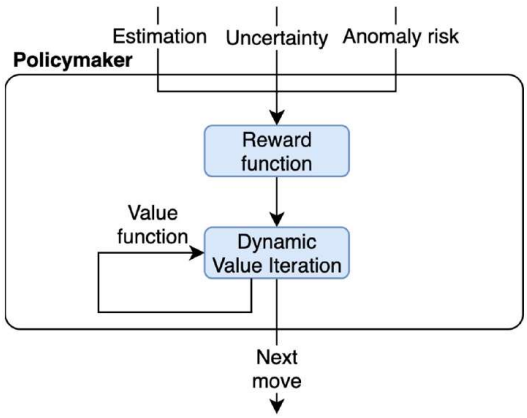


Figure 4. The policymaker flowchart within the HSN architecture. The policymaker determines the next move for the moving sensors by maximizing the reward using our proposed Dynamic Value Iteration (DVI) method.

2.3.2.1 Reward Function

The reward function is the core of the policymaker. It must be carefully designed to reflect the objectives assigned to the moving sensors, which are to minimize state estimation uncertainty and detect anomalies within the shortest time. To devise a proper reward function, a value needs to be assigned at each location to represent how much reward there is for a sensor to visit that location. This should be done at each time step given the three outputs of the analyzer. The design of a reward function needs to balance a tradeoff between exploitation and exploration. On the one hand the sensors should leverage information or state estimation for locations with high probability of anomalies, on the other hand the sensors should also explore and take measurements at locations with high state uncertainty.

First, we design an index for exploitation to be included in the reward function based on the estimated state. We approximate the current conditional probability of having anomalies at each location by a function of the state estimation, $E_t = E(\hat{x}_t)$. For simplicity, we posit that this is a linear function, $E_t = (\hat{x}_t - x_0)/\Delta x$. This is reasonable since, for example, the higher measurement of temperature, CO_2 concentration, or smoke (soot) density, the higher probability of a fire event.

Second, we design an index for exploration to be included in the reward function based on the estimation uncertainty U_t, together with the anomaly risk Φ_t. As noted previously, these two quantities are intrinsically different. The estimation uncertainty U_t comes from the measurement noise and system disturbance, whereas the anomaly risk Φ_t comes from the randomness of having anomalies. Exploring locations with higher

estimation uncertainty or anomaly risk can improve the overall monitoring and understanding of the environment.

The final reward function is computed with these two indices, as shown in:

$$R_t = E_t + \alpha_t U_t + \beta_t \Phi_t \tag{15}$$

The two parameters α_t and β_t can be time-varying, which control the exploration rate. Now that we have developed the reward function for the policymaker to consider, the next critical problem to solve is the determination of the optimal next move for the sensor given this reward function. Traditionally, this problem can be solved by the Value Iteration (VI) method [63], or by solving a variant of the travelling salesman problem [64]. We discuss in the next subsection why these methods are not suitable for our application and how we propose to tackle this problem.

2.3.2.2 Dynamic Value Iteration (DVI)

We use a dynamic version of the Value Iteration (VI) method to successively solve for the optimal move at each time step. Note that this optimal move is based on the previous reward function we develop, and it is supposed to fulfill the goal of minimizing state estimation uncertainty and detecting anomalies within the shortest time possible. We first introduce the traditional VI, and then discuss the modification we made to address our problem.

Value function is an important concept within the context of Markov decision process (MDP). It is defined as the optimal total reward one can obtain starting from a

particular location. Traditional VI computes the value function through dynamic programming, as shown in:

$$V_t^0(s) = 0$$

$$Q_t^{k+1}(s,a) = \sum_{s'} P(s'|s,a)\left(R_t(s,a,s') + \gamma V_t^k(s')\right), k \geq 0 \tag{16}$$

$$V_t^k(s) = \max_a Q_t^k(s,a), k > 0$$

where V is the value function at each location s, P is the transition probability from one location s to the next location s' by taking an action a, R is the reward function, and γ is the decaying parameter ensuring convergence. The subscript t indicates the time step, and the superscript indicates the iteration number. At each time step, the algorithm starts with an initial guess of zero, and iteratively updates the value function until convergence or maximum iterations for infinite or finite horizon respectively.

For a deterministic problem with a reward function that depends only on the next location, Eq. 16 can be simplified to:

$$V_t^0(s) = 0$$

$$Q_t^{k+1}(s,a) = R_t(s') + \gamma V_t^k(s'), k \geq 0 \tag{17}$$

$$V_t^k(s) = \max_a Q_t^k(s,a), k > 0$$

The problem of traditional VI, as well as some other techniques noted previously, is that they assume the reward is constant in time, which is clearly not the case in our

environmental monitoring and anomaly search problem. The value function solved at each time step is only valid at that specific time step. In our problem, as new measurements are taken, the reward function expressed in Eq. 15 also changes. Computing the value function at the next time step with a different reward function requires solving the problem again, which might incur unnecessary computational cost for real-time decision making.

We address this problem by considering the dynamic nature of the environment. Since the decision-making timescale is much shorter than the state changes in the environment given the underlying anomaly generation process (convection and conduction in the case of fire), we assume the value function at one time step is proximal with that at the previous step. Therefore, we reuse the value function at the previous time step as the initial guess for the next one, as shown in:

$$V_{t+1}^{0} = V_t \tag{18}$$

Moreover, we truncate the VI with a finite horizon at each time step. Because the environment is constantly evolving, the true value function at current time step will be largely different from that in the far future. Therefore, it is enough to iterate the value function up to an appropriate finite horizon with little impact on the optimality of the policy.

These two operations together save significant computational cost and speed up the decision-making process compared with the traditional VI with infinite horizon. By designing the reward function in Eq. 15 and the DVI method to solve it, we have developed a policy for a single moving sensor. We expand this decision-making process for a multi-agent system next.

2.3.2.3 Multi-Agent System

For multiple moving sensors, a critical problem is to avoid them crowding at one location. This will result in system inefficiency and lead to potential collision between them. Solving for the global optimum for a multi-agent system has exponential order of complexity with respect to the agent number [65]. This makes the computational cost prohibitive very quickly as the number of agents increases.

A more practical way is to find a local sub-optimum that is feasible with the available computational power. This is usually done by decentralized approaches. In our work, we realize this by scaling down the reward around the other agents while making decision for each one of them. We use a Gaussian kernel for the scaling effect, as given in:

$$ker_i(s; s_i, \sigma) = 1 - exp\left(-\frac{d(s - s_i)^2}{2\sigma^2}\right) \tag{19}$$

where s_i is the location where agent i is, σ is a parameter controlling the kernel size. We set σ proportional to the average agent distance with a factor p, $\sigma = p \cdot sensor\ density^{-1/3} = p\left(\frac{\#\ of\ agents}{total\ volume}\right)^{-1/3}$. The adjusted reward function for agent i can be derived by applying this kernel to all the other agents, as shown in:

$$R_{t,i}(s) = R_t(s) \cdot \min_{j \neq i} ker_j(s; s_j, \sigma) \tag{20}$$

We use the DVI to solve for the next move for each agent based on this adjusted reward function. This can be done either in the central system with parallel computing or be distributed to each agent to compute locally. To recap, we have developed a

decentralized policymaker for a multi-agent system using our dynamic value iteration (DVI) method based on the reward function expressed in Eq. 15. To improve the monitoring and detection performance, we conduct an Inverse Reward Shaping (IRS) process to tune the reward function parameters, as discussed in the next subsection.

2.3.2.4 Inverse Reward Shaping (IRS)

To make the reward function more accurately relate to our objectives, we append an Inverse Reward Shaping (IRS) process to the policymaker to adjust the reward function in light of the performance metrics. This is essentially a hyperparameter tuning process, which can be formulated as an optimization problem. For simplicity, we focus only on the detection time lag next. The IRS process is defined as finding the reward function parameter (α, β) to minimize the average detection time lag of n random computational experiments, as formulated in:

$$\min_{\alpha,\beta} \frac{1}{n} \sum_{i=1}^{n} lag_i(\alpha, \beta) \tag{21}$$

where $lag_i(\alpha, \beta)$ is the detection time lag of the i-th experiment, with the reward function parameter (α, β). This problem can be solved by any appropriate optimization methods, such as ADAM, stochastic gradient descent, or Nelder-Mead simplex algorithm [66–68]. We choose the Nelder-Mead simplex algorithm as provided in MATLAB *fminsearch* function [69].

This completes the description of the whole architecture of HSN and its analytical constituents. In the next section, we discuss the computational experiments we conduct to validate and benchmark the monitoring performance.

2.4 Application-Specific Simulation Environment and Performance Metrics

In Section 2, the problem statement is defined for a general monitoring task with an arbitrary performance metric as the objective function. In this section, the discussion is narrowed down to two specific applications in dynamic simulation environments: a fire search example in a 2D apartment with temperature measurements and a CO leak detection example in a 3D space station module with ventilation system. The simulation environments and the specific performance metrics used hereafter are introduced next.

2.4.1 Fire Search in a 2D Apartment

The first application is for temperature monitoring and fire detection in a 2D multi-room apartment. The floorplan of the apartment is shown in Fig. 5. A $20 \times 20\, m^2$ apartment is discretized into $1\, m^2$ grids for potential sensor locations. The mobile sensors can move to the four neighboring grids or stay at each time step. Note that although the current environment is a regular apartment, the deployment method and the network monitoring performance is scalable to the environment size as long as the sensor density is the same. For example, 10 sensors in an $10 \times 10\, m^2$ area will have a similar performance to 40 sensors in a $20 \times 20\, m^2$ area.

Figure 5. Example environment of a $20 \times 20\ \mathrm{m}^2$ apartment for temperature monitoring and fire search and detection. The red crosses indicate locations with higher fire risks (R1–R6). The blue circles indicate 50 sampled fire locations for later simulations.

There are three key characteristics of this application. First, the temperature field is dynamic. The purpose of building the simulation environment is to obtain a dynamic temperature field inside a room when a fire event occurs. A conduction heat propagation process is adopted to simulate realistic indoor conditions [70]:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T + \frac{Q}{\rho c_v} \tag{22}$$

where t stands for time, T the temperature field, α the thermal diffusivity, ρ the air density, c_v the constant volume specific heat constant, and Q the external volumetric heat release rate. The boundary condition is set to be adiabatic. It is assumed that there is no explicit process noise, and the sensor measurement noise is 0.1 ℃. But the unknown fire

source can be seen as an implicit process noise. Although a specific simulation environment has been developed for this study, other options are possible and users can adopt other models or high-fidelity simulation software, for example the Fire Dynamics Simulator (FDS) developed by the National Institute of Standards and Technology (NIST) [71], to obtain a suitable dynamic field for their specific applications.

The simulation parameters are selected to reflect a realistic situation. Air properties are taken at $300K$ (27 ℃), and the heat release rate (HRR) of the fire source is set to be $5000\ W$ [72,73]. Notice that this is a relatively small fire whose HRR does not increase with time. A real fire can rapidly double its HRR within a short time. The probability of fires occurring at high-risk locations is set to be ten times as higher than the other locations. The complete simulation parameters are provided in Table 1.

Table 1. Simulation parameters for fire search example

Parameters	Descriptions	Value
c_v	Specific heat capacity at constant volume	$0.718\ [kJ \cdot kg^{-1}K^{-1}]$ at $300K$
ρ	Density	$1.161\ [kg \cdot m^{-3}]$ at $300K$
α	Thermal diffusivity	$22.63 \times 10^{-6}\ [m^2 \cdot s^{-1}]$ at $300K$
HRR	Heat release rate	$5000\ [W]$
Q	External volumetric HRR	$5000\ [W/m^3]$

Second, fire risk is non-uniform in this room. This conforms with a previous observation that some locations have higher risks of fire. For example, cooking in the kitchen and electric heaters in the bedrooms exhibit a higher fire risk. The high-risk locations are marked with red crosses in Fig. 5. Third, a physical sensing model for the fire detection process is adopted instead of the commonly used disk-shaped model with a predefined sensing radius. A fire event is detected according to the measured temperature. The sensor selected for the computational experiments is a pointwise temperature sensor

with alarm threshold at 47 ℃ [17]. The sensors trigger a fire alarm when the measured temperature breaches this threshold.

2.4.2 CO Leak Detection in a 3D Space Station Module

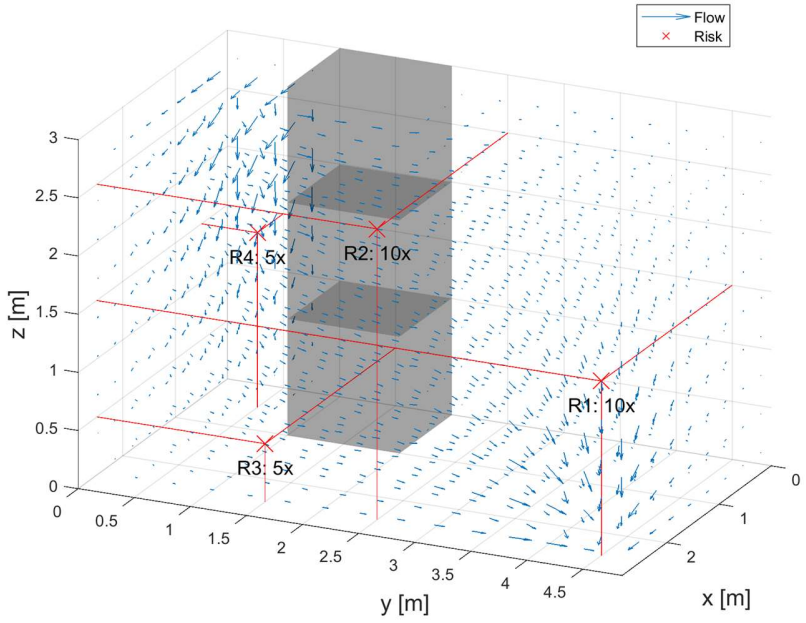


Figure 6. CO leak in a space station module with venting system. The blue arrows indicate the flow field, and the red crosses indicate locations with higher leak risks (R1–R4).

The second application is for CO leak detection in a 3D space station module with ventilation system. The environment is shown in Fig. 6. CO leaks in a $3 \times 5 \times 3\ m^3$ space station module is simulated. To add complexity to the environment dynamics, an $1 \times 1 \times 3\ m^3$ obstacle is presented as some scientific equipment. A pair of air inlet/outlet are placed at $(0,0,3)$ top-left corner and $(3,5,0)$ bottom-right corner blowing/sucking air with a flow speed of $5\ cm/s$. In a 3D environment, CO leaks can only come from anomalies behind the panels (walls) through some gaps on them. For the CO propagation

inside the module, it is equivalent to treat these gaps as the CO sources. It is assumed that there are four high risk locations at R1-R4 on the wall with 10- and 5-times higher risk of CO leak than the rest of the wall.

The CO propagation process is a computational fluid dynamic problem, which is solved with the Navier-Stokes (NS) equations. An incompressible viscous flow is assumed because of the low flow velocity simulated. The flow field is first solved by

$$\nabla \cdot \vec{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{23}$$

$$\frac{D\vec{V}}{Dt} = \frac{\partial \vec{V}}{\partial t} + \left(\vec{V} \cdot \nabla\right)\vec{V} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \vec{V} \tag{24}$$

where x, y, z are the coordinates; $\vec{V} = [u, v, w]$ is the flow velocity; ρ, p, ν are the density, pressure, and kinematic viscosity of the flow. The boundary conditions are set as constant flow velocity at the air inlet/outlet and no perpendicular velocity for the other (wall and obstacle) surfaces. With the flow field given, the CO propagation is solved by

$$\frac{Dc}{Dt} = \frac{\partial c}{\partial t} + \vec{V} \cdot \nabla c = \alpha \nabla^2 c \tag{25}$$

where c is the CO concentration and α is the CO diffusion coefficient in air. A no flux boundary condition is assumed for the CO propagation, except a constant 500 ppm concentration at the CO source.

The environment is discretized into grids to computationally solve the flow field and CO propagation with a spatial step of $\Delta x = \Delta y = \Delta z = 1/3\ m$ and a temporal step of

$\Delta t = 0.1\ s$. An example CO leak in this environment is demonstrated in the attached mp4 file. A total of 39 CO leaks at all 1 meter grid on the wall is simulated and will be used to test the sensor network performance.

Table 2. Simulation parameters for CO leak example

Parameters	Value
Environment size	$3 \times 5 \times 3\ m^3$
Vent inlet/outlet flow speed	$5\ cm/s$
kinematic viscosity ν at 20℃	$15.06 \times 10^{-6}\ m^2/s$
CO diffusion coefficient in air α at 20℃	$2.08 \times 10^{-5}\ m^2/s$
CO leak source concentration	$500\ ppm$
Spatial discretization step $\Delta x, \Delta y, \Delta z$	$1/3\ m$
Temporal discretization step Δt	$0.1\ s$
CO sensor alarm threshold	$10\ ppm$

Similar to the CO leak locations, static sensors need to be located on the wall and cannot be placed at an open space, whereas the mobile sensors (e.g., sensor on AstroBee [74]) are allowed to move freely. The alarm threshold for the sensors is set to be 10 ppm according to National Ambient Air Quality Standards 8-hour time weighted average CO exposure[1]. Note that the 7-day Spacecraft Maximum Allowable Concentrations (SMAC) for CO is 55 ppm[2]. All environment parameters are summarized in Table 2.

2.4.3 Performance Metrics

As noted in the introduction, previous sensor network deployment methods focused on improving sensor coverage, network connectivity, and energy efficiency. However, for a given application in a complex environment, the aforementioned metrics do not

[1] https://www.epa.gov/criteria-air-pollutants/naaqs-table

[2] https://www.nasa.gov/sites/default/files/atoms/files/jsc_20584_signed.pdf

necessarily reflect the ultimate aim of the monitoring tasks, such as detection time and state estimation error. In this work, the three performance metrics listed in Table 3 are considered as the ultimate objective for the fire detection application [16].

Table 3. Performance metrics in this application.

Performance Metrics	Descriptions
Detection time lag	The time to detect an anomaly after it occurs
Alarm distance	The distance between the alarmed sensor and the anomaly source
Estimation error	The median error between the estimated state value and the true measurement

First, detection time lag represents the temporal sensitivity of the sensor network. Early detections grant people more reaction time. Second, alarm distance indicates the spatial uncertainty of the anomaly source when it is detected. The reasoning behind using this metric is that when an alarm is triggered, the source of the anomaly is often uncertain. For example, it took several months to localize the source of a leak on-board the International Space Station after its detection [44]. It will be easier and faster to perform the subsequent search and intervention if this spatial uncertainty is reduced. Third, the estimation error reflects the general monitoring performance over the whole space. The median error is adopted because the estimation can present a significant error in a small region around the anomaly source. This deviance will affect both the mean and the maximum estimation error, whereas the median error is less biased and more robust. Therefore, the entire distribution of the state estimation error is more ambitious and computationally challenging. It is left as a potential avenue for future work.

Note that calculating these metrics requires the ground truth provided by the simulation environment. These metrics serve as an evaluation of the sensor network for the testing purpose.

Next, these three metrics are used as the objective functions to evaluate the performance of the sensor networks.

2.5 Comparison Between WGC, Entropy, and MI Approaches

Several groups of comparison between WGC, entropy, and MI are conducted for different number of static sensors. To have better performance for the entropy and MI approaches, the ground truth covariance for the simulated data is used. Note that this provided the best possible performance for these two approaches and makes the comparison harder.

2.5.1 Fire Search Example

Table 4. Mean detection time and standard deviation of 6 sensors for 50 fires with different approaches.

Methods	Mean detection time [hour]	Detection time std [hour]
WGC	**3.15**	1.58
WGC - Greedy	3.21	1.62
Entropy	3.58	2.27
Entropy - Greedy	3.49	2.36
MI	4.83	2.24
MI - Greedy	4.73	2.12

The ground truth covariance of 50 fires is used for the entropy and MI approaches. Fig. 7 shows the sensor placement of 6 sensors. The reason for choosing this number of sensors is that it corresponds to the number of high-risk locations, about which we have most intuition. And the mean detection performance is shown in Table 4. Note that due to

the environment setup of pure conduction heat propagation, which is very slow without advection, the detection time is also very long for several hours.

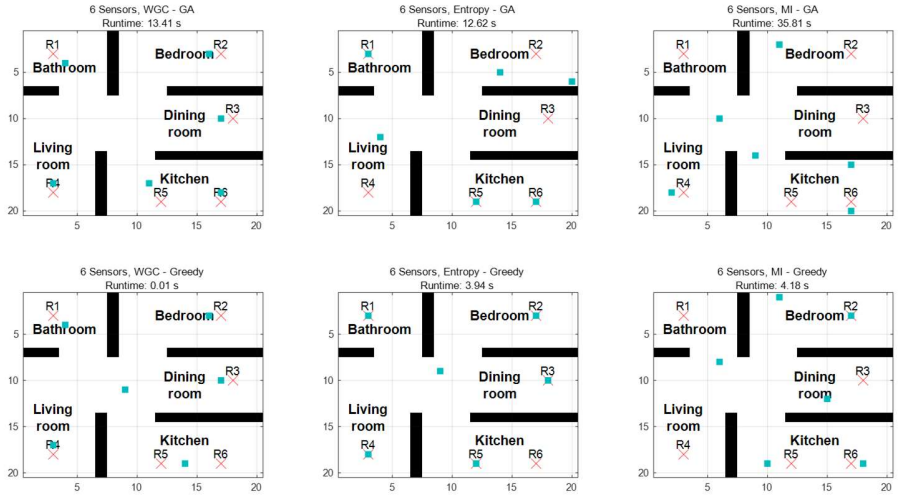


Figure 7. Sensor placement results of 6 sensors for WGC, entropy, and MI approaches.

The five placements are for WGC, entropy, and MI with genetic algorithm (GA), and the greedy version of the latter two. From the performance perspective, the most salient observation is that WGC provides the best performance, followed by entropy approaches and MI approaches. Intuitively, the six sensors should be placed around the six high-risk locations. It can be observed that this is the WGC deployment. The sensors are placed a little bit closer to the central area to cover more space while maintaining good knowledge of the high-risk locations. The entropy approaches, especially the greedy version, favor high-variance locations and place sensors at high-risk locations and around the corner. These approaches are more suitable for environments with a non-uniform risk distribution. The MI approaches try to balance the entropy between the selected sensor locations and the rest of the area. While they are good for placing sensors more uniformly in an

environment with uniform risk, they fail to cover the high-risk locations well enough, which result in the poorest detection performance.

From the computational cost perspective, the most salient observation is that the greedy algorithms provide the fastest solution. Since the entropy and MI metrics involve the inversion of a high-dimensional covariance matrix, their evaluations are slow which makes the GA optimization insufficient with limited time and results in poor performance. In contrast, the WGC metric is easy to evaluate, and thus very suitable for GA optimization. WGC requires a longer but acceptable time to obtain a better solution.

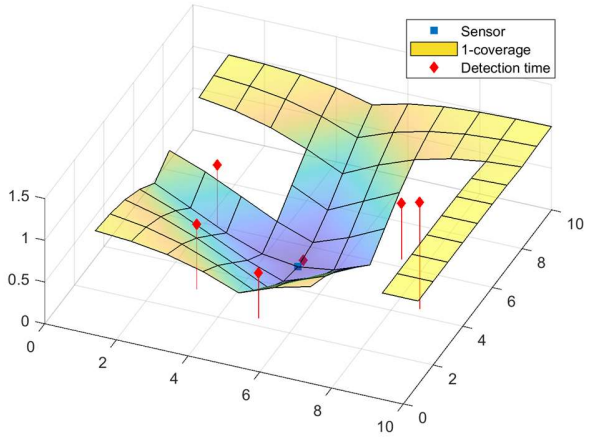


Figure 8. Compare the coverage function in WGC with the detection time.

The core reason for WGC to perform better is that it is a more direct metric to reflect detection time than entropy and MI which reflect more about uncertainty. Fig. 8 shows one example of the comparison between the coverage function used in WGC and the actual detection time derived from the simulations. The surface plot shows $1 -$ coverage function for a sensor located at $(5, 5)$. The time for this sensor to detect its surrounding fires are marked with red diamonds. Note that the detection time is scaled for better comparison.

One major observation is that the shape of $1-$ coverage function corresponds to the detection time. Therefore, maximizing the WGC is equivalent to minimizing the overall detection time for all potential fires in the environment according to the risk distribution.

The results and analysis apply to more sensors and the qualitative tendencies can be seen more clearly as shown in Table 5 for results with 3, 10, and 20 sensors.

Table 5. Mean detection time of 3-20 sensors for 50 fires with different approaches.

Methods	Mean detection time [hour]			
	3 Sensors	6 Sensors	10 Sensors	20 Sensors
WGC	**4.59**	**3.15**	**1.81**	0.84
WGC -Greedy	4.76	3.21	1.89	**0.82**
Entropy	4.99	3.58	3.60	1.88
Entropy - Greedy	4.62	3.49	2.99	2.26
MI	5.36	4.83	3.76	2.35
MI - Greedy	5.46	4.73	3.75	2.40

2.5.2 CO Leak Example

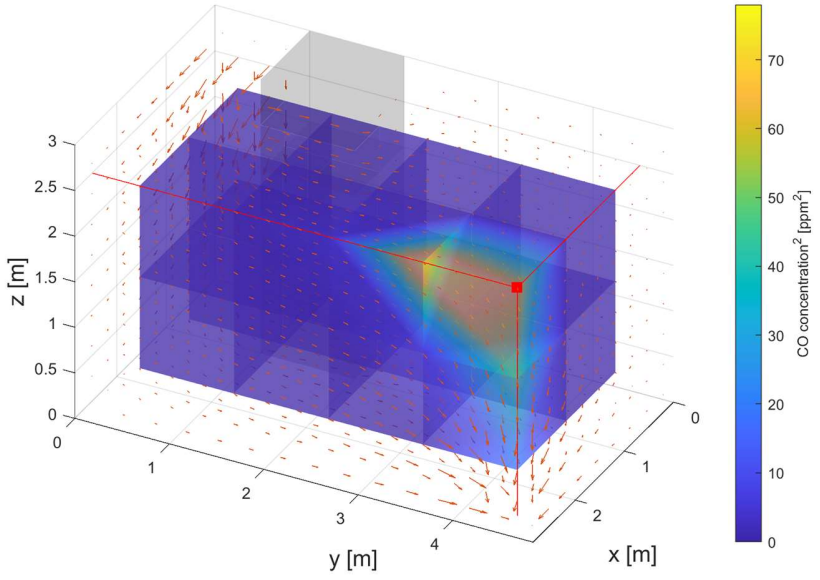


Figure 9. Covariance map of a sensor placed at $(3, 5, 3)$.

The ground truth covariance of 39 CO leaks is used for the entropy and MI approaches. Fig. 9 shows the covariance map of a sensor placed at (3,5,3). It can be observed that the covariance is higher along the flow but lower along the perpendicular direction.

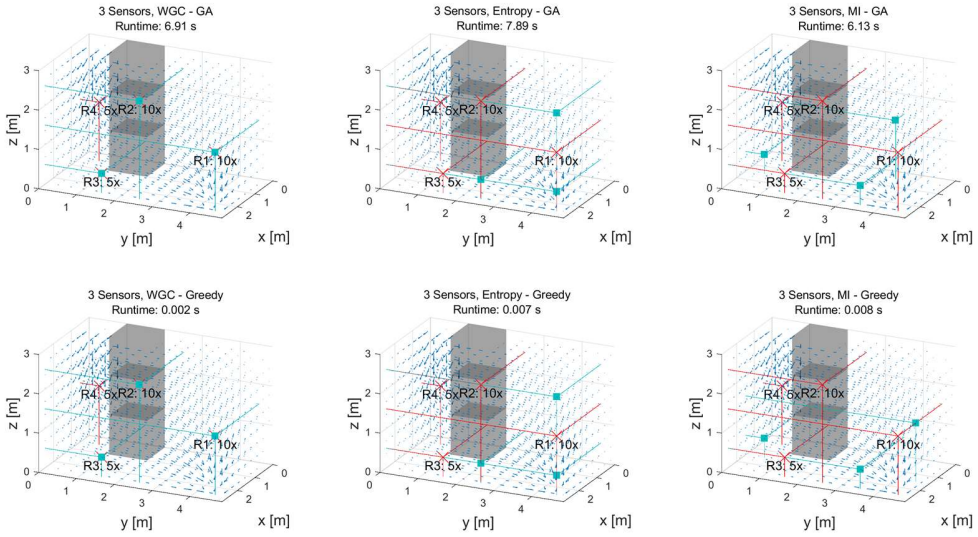


Figure 10. Sensor placement results of 3 sensors for WGC, entropy, MI, and their greedy approaches.

Table 6. Mean detection time and runtime of 3 sensors for 39 CO leaks with different approaches.

Methods	Mean detection time [s]	Detection time std [s]	Runtime [s]
WGC	130.58	39.78	6.02
WGC - Greedy	**130.58**	**39.78**	**0.009**
Entropy	157.72	51.89	7.44
Entropy - Greedy	157.72	51.89	0.015
MI	202.08	87.57	5.94
MI - Greedy	199.91	85.82	0.019

Fig. 10 shows the sensor placement of 3 sensors. And the mean detection performance and the computational runtime is shown in Table 6. The reason for choosing

this number of sensors is that it is less than the number of high-risk locations, and the priority of the high-risk locations can be observed from the sensor placement results.

From the performance perspective, the most salient observation is that WGC provides the best performance, followed by entropy approaches and MI approaches. In addition, due to the small number of sensors, the greedy solutions are almost identical to the global optimized solution. When the sensor number is less than the number of high-risk locations, WGC places sensors at higher risk-locations first at R1 and R2. For R3 and R4, WGC chooses R3 for a better coverage over the whole space. The entropy approach favors high-variance locations and place sensors at the downstream outlet vent corner. While given enough time, a sensor at the outlet vent can detect leaks at any locations, it is not the fastest solution. The MI approaches try to balance the entropy between the selected sensor locations and the rest of the area. While they are good for placing sensors more uniformly in an environment with uniform risk, they fail to cover the high-risk locations well enough, which result in the poorest detection performance. From the computational cost perspective, the most salient observation is that the greedy algorithms provide the fastest solution. When the sensor number is small, the performance drop of the greedy solutions is also small. Since the entropy and MI metrics involve the inversion of a high-dimensional covariance matrix, their greedy evaluations are slower than the WGC metric.

The results and analysis apply to more sensors and the qualitative tendencies can be seen more clearly as shown in Table 7 for results with 3, 6, and 9 sensors. It is worth pointing out that the greedy WGC approach provides the best and fastest performance.

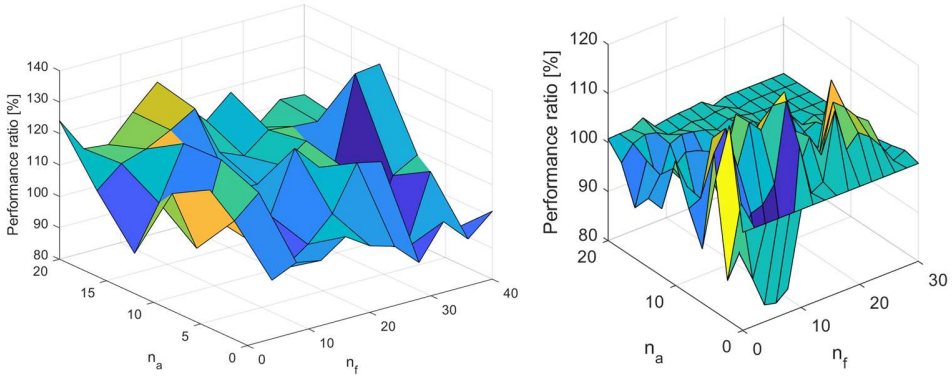
Table 7. Mean detection time of 3-9 sensors for 39 fires with different approaches.

Methods	Mean detection time [s]		
	3 Sensors	6 Sensors	9 Sensors
WGC	**130.58**	93.92	81.85
WGC - Greedy	**130.58**	**93.84**	**75.68**
Entropy	157.72	106.22	84.40
Entropy - Greedy	157.72	106.22	84.40
MI	202.08	146.87	128.79
MI - Greedy	199.91	144.86	128.79

In conclusion, the results show that WGC is a better metric for the SSA problem with non-uniform risk. It has a uniform distribution around the low-risk area while maintaining good coverage around the high-risk locations. Entropy approaches favor high-variance locations at the high-risk locations and corners, but they fail to have a good coverage over the low-risk area. In contrast, MI approaches have a more uniform distribution to balance the entropy on the selected sensor locations and the rest of the area, but they fail to cover the high-risk locations well enough.

2.6 Comparison Between DVI and MPC Approaches

To apply model predictive control and compare with it, there needs some update based on the original MDP model. Since the moving mechanism for the mobile sensors are relatively simple, the major predictions are for the environment dynamics. The technical details of the MPC application are discussed in Appendix C. A series of experiments are conducted with different combinations of static and mobile sensors. For MPC, a prediction step of 5 is used. The performance ratio between MPC and DVI is shown in Fig. 11. The average runtime of the two methods are summarized in Table 8.



(a) Fire search example (b) CO leak example

Figure 11. Performance ratio of detection time between MPC and DVI for the two environments.

Table 8. Mean runtime of DVI and MPC for the two environments.

Methods	Mean Runtime [s]	
	Fire Search	CO Leak
DVI	23.27	0.0976
MPC	131.82	0.4566

First, we can observe that the runtime of MPC is approximately 5 times that of DVI. This is expected as MPC takes 5 prediction steps for each move. Second, we can observe that the performance is inconsistent for both environments. Applying MPC can both improve and deteriorate the detection time for different scenarios. But there is no significant and consistent improvement for MPC. This might be counter-intuitive in the first place because making predictions should help the decision making. It should be the case if the predictions are accurate. The prediction of the mobile sensor trajectories is very easy with this simple moving mechanism assumed in this work. However, for anomaly detection tasks with unknown source, such as in these two examples, making accurate prediction for the environment dynamics can be challenging. The unknown anomaly source can bring huge uncertainty increasing with time and accurate predictions will be very hard

without more measurements. Eventually, the poor predictions make the decision making very unpredictable. Overall speaking, the original DVI method is sufficient for these applications, and it is more computationally efficient.

2.7 Example Simulation Result

The 3D CO leak example proves that the HSN deployment strategy can be extended to more complex environments such as ocean monitoring. But the 2D fire search and temperature monitoring example is still general enough for a lot of applications such as forest fire search, precise agriculture, and weather monitoring. For the ease of demonstration purposes, the 2D example will be used for the following study and examination.

In the next example, we deploy 6 static sensors and 2 mobile sensors with moving speed of $6.67\ cm/s$ for illustrative purposes. At the start of each simulation, an initial fire location is randomly selected according to the fire risk map. We run the simulation until the fire is detected by one of the sensors. For the robustness of the result, we repeat the simulation for one hundred times with different initial fire locations. We use the weighted mean performance according to the fire risks for our comparison. Here, we include a snapshot taken at the end of the simulation, as shown in Fig. 12.

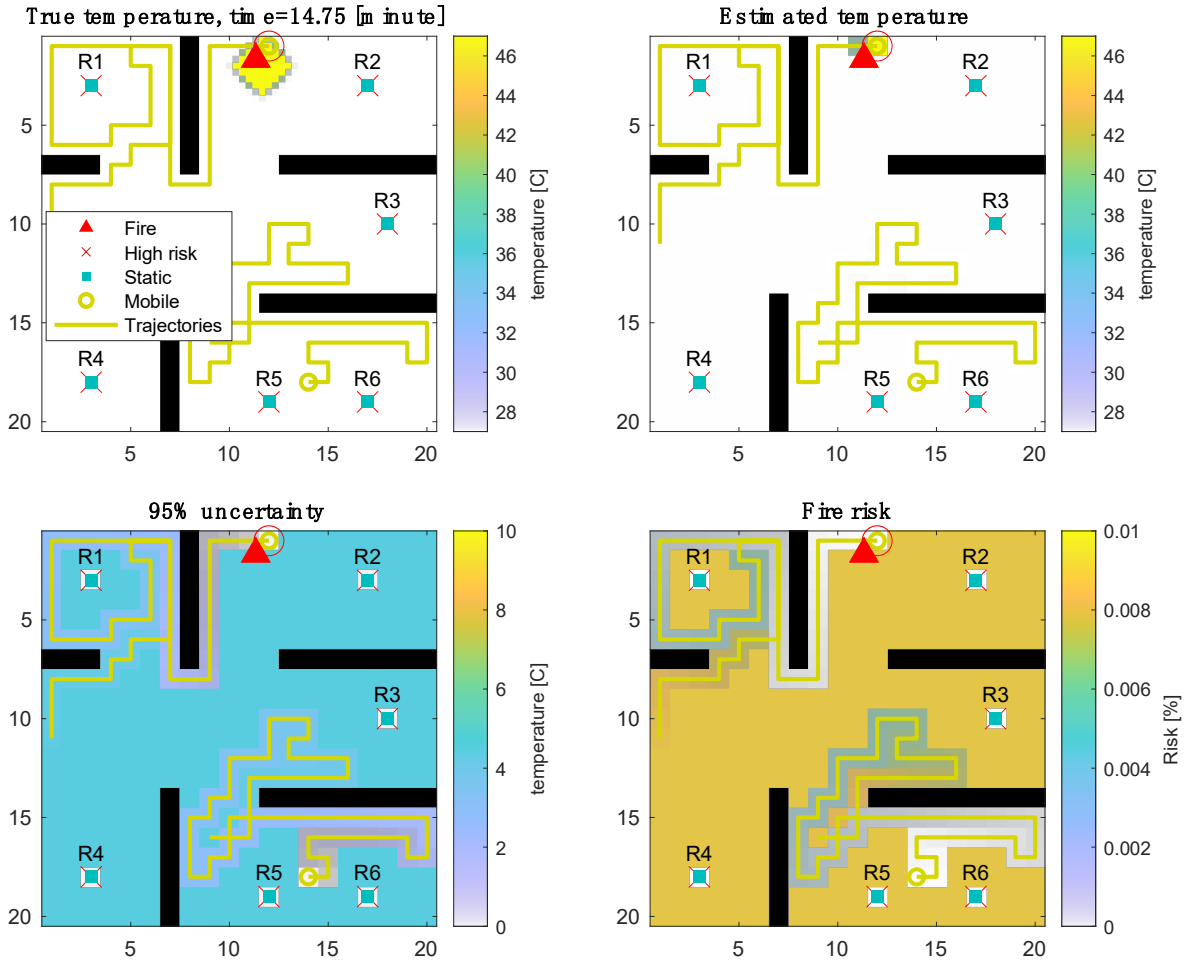


Figure 12. The final snapshot of one example simulation of 6 static sensors and 2 mobile sensors. The four maps show the true temperature (top-left panel), the estimated temperature (top-right panel), the estimated fire risk (bottom-right panel), and the 95% uncertainty of the estimation (bottom-left panel).

We observe in this example that all six static sensors are allocated at the high-risk locations. The sensors are allocated this way to achieve the highest weighted Gaussian coverage (WGC). The two mobile sensors avoid each other as well as the static sensors to explore the environment more efficiently. This simple result illustrates how the static sensors can be used to monitor fixed high-risk locations, and the mobile sensors can patrol

a larger area with dynamic risk uncertainty. The fire is detected after 14 minutes because the heat transport is relatively slow under our defined conditions.

CHAPTER 3. OPTIMIZATION

Chapter 2 provides a collaborative deployment strategy for hybrid sensor networks when the number of sensors is given. A more general problem is to also optimize how many sensors of each kind should be deployed when the total cost of the network is constrained. It is referred to as the sensor portfolio optimization problem for hybrid sensor networks. In this chapter, a general optimization problem is first formulated and then solved to identify the optimal sensor portfolio mix and its main drivers under different cost constraints.

3.1 Problem Statements

Before we elaborate on our specific research objectives, we first introduce the main design parameters considered (Table 9) and assumptions to facilitate the subsequent discussion.

Table 9. Design parameters for the hybrid sensor network problem

Design parameters	Descriptions
C	Total budget for the sensor network
$\gamma \in (1, \infty)$	Cost ratio of a mobile sensor over a static sensor
n_f	Number of static sensors
n_a	Number of mobile sensors
$N_{tot} \in \left[\frac{C}{\gamma}, C\right]$	Total number of sensors (both static and mobile sensors)
$\beta \in [0,1]$	The proportion of mobile sensors in the network. Referred to as the portfolio mix

First, we consider the cost parameters. We set C as the total budget for the sensor network. We define the cost of a static sensor to be one as the unit cost and that of a mobile sensor to be γ, which stands for the cost ratio between them. Since the focus in this work is not on the detailed cost structure, e.g., design cost, manufacturing cost, integration,

testing, and operational costs, we use a constant total cost to include all of them. Note that while some cost might seem to be time-varying (e.g., running cost for mobile sensors), we can work around this matter by setting a fixed expected working life for them (e.g., 50 thousand hours for a mobile sensor) and calculating an approximate constant total cost. We assume that each mobile sensor consists of a static sensor mounted on a moving platform. Thus, the cost ratio is larger than one, $\gamma > 1$. Note that in practice, the mobile element usually consists of a larger suite of sensing devices. In this work, we restrict that both sensors have the same sensing device such that we can study the mobility effect only.

Second, we consider the sensor portfolio parameters. The number of static and mobile sensors is defined to be n_f and n_a respectively. The total number of sensors in the network is $N_{tot} = n_f + n_a$, and the sensor network total cost C_{tot} can be calculated by:

$$\begin{aligned} C_{tot} &= n_f \cdot (\text{cost of a static sensor}) + n_a \cdot (\text{cost of a mobile sensor}) \\ &= n_f + \gamma n_a \end{aligned} \tag{26}$$

Note that for environments with an arbitrary size, sensor density is a more appropriate parameter to describe the quantity of sensors. Since we focus here on one environment with fixed size, sensor density is equivalent to sensor number, and we use them interchangeably in our later discussion. Given the linearity of Eq. 23, the iso-cost curve in the n_f-n_a space is represented by a straight line. This observation will be useful later in our analysis in section 3.2. Since we only consider two types of sensors, the sensor portfolio can be characterized by the proportion of either one type. Without loss of generality, we choose the proportion of the mobile sensors among all sensors $\beta =$

$n_a/N_{tot} \in [0,1]$, and refer to this ratio as the portfolio mix hereafter. For example, $\beta = 0$ indicates a purely static sensor network; $\beta = 1$ a purely mobile one; and $\beta \in (0,1)$ a hybrid sensor network. The sensor network cost C_{tot} can be rewritten using β in:

$$C_{tot} = (1-\beta)N_{tot} + \gamma\beta N_{tot} = (1-\beta+\gamma\beta)N_{tot} \tag{27}$$

Given these design parameters, we can now present the problem formulation. Previous works have generally formulated the design of sensor networks as optimization problems [18,55,75]. In this work, we formulate our problem as a cost-constrained optimization problem, as shown in:

$$\begin{aligned} \max_{n_f, n_a} \quad & \text{performance}\left(\Phi\left(n_f, n_a\right)\right) \\ \text{subject to} \quad & C_{tot} = n_f + \gamma n_a \leq C \end{aligned} \tag{28}$$

We seek to find a sensor portfolio (n_f, n_a) with the collaborative sensor deployment method Φ in chapter 2 to maximize some performance metrics and guarantee the total cost is within the budget $C_{tot} \leq C$. Note the following remarks on Eq. 25. First, the performance is still kept as a general objective function to be optimized. It can be easily replaced by a target performance metrics as described in section 2.4.2. Second, we exclude the energy constraints on the sensors because compared with typical single-cycle battery runtime of several hours or more, the total runtime for our example monitoring task is significantly shorter, as will be illustrated later.

Next, we simplify Eq. 25 to an equivalent optimization problem. Since the optimal performance is achieved by using the maximum budget C, the inequality constraint is

always active. If the sensor portfolio mix β is given, we can use Eq. 24 to compute the total number of sensors by:

$$N_{tot} = \left\lfloor \frac{C}{1 - \beta + \gamma\beta} \right\rfloor \tag{29}$$

where the floor function $\lfloor \cdot \rfloor$ gives the greatest integer less than or equal to its argument. Note that the denominator in Eq. 26 does not vanish and this expression is well-defined. In other words, given the cost parameters (C, γ), we can calculate the total number of sensors N_{tot} by finding the optimal portfolio mix β. Since the sensor portfolio (n_f, n_a) can be obtained from (N_{tot}, β), we switch the design variables to β. We then simplify Eq. 25 to an equivalent problem as shown in:

$$\max_{\beta \in [0,1]} \quad \text{performance}\big(\Phi(\beta; C, \gamma)\big) \tag{30}$$

The optimal portfolio mix β should be determined given the cost parameters (C, γ). Note that for practical use with one given cost condition, only a line search algorithm along the corresponding iso-cost curve is needed. By varying the cost parameters, a Pareto front is obtained for the network cost-performance multi-objective optimization problem. This problem is solved in a divide-and-conquer manner. With the deployment method Φ, simulations are conducted for different sensor portfolio β under different cost conditions to identify the optimal sensor portfolios.

There are two research objectives in this chapter. First, to identify the optimal sensor portfolio β for different cost parameters (C, γ) (O2). Most previous works only consider

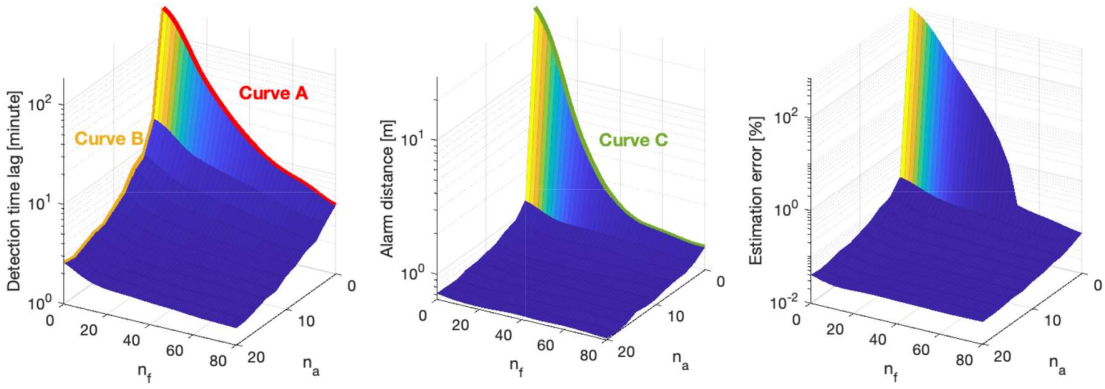
HSNs with pre-set (not modifiable) sensor portfolio [10,38]. In this work, a cost-performance analysis is conducted for different sensor portfolios, and the optimal portfolio mix is determined under various cost conditions. Second, to investigate the effect of the mobile sensor speed on the optimal portfolio mix (O3). To address these questions, the HSN results are presented and discussed in the next section.

3.2 Computational Experiments: Results and Discussion

In this section, the results of the computational experiments are discussed in light of the two research objectives O2 and O3. The same computational environment as in chapter 2 is used. To identify the optimal sensor portfolio (O2), the network performance is evaluated for multiple portfolio candidates under various cost conditions. Additional experiments are conducted to examine the impact of the mobile sensor on the network performance and the optimal sensor portfolio (O3).

3.2.1 Cost-Performance Tradeoff Analysis and Sensor Portfolio

The optimization of the sensor portfolio is a central problem for HSNs with cost constraint. It is important to determine the benefit of adding each type of sensor and to balance the tradeoffs between them. A performance saturation point can be reached when adding sensors of the same type beyond a certain point. The performance of a large combination of different sensor numbers (n_f, n_a) is analyzed to understand this performance landscape. Additionally, the marginal improvement of different sensor portfolios is investigated. The moving speed for the mobile sensors is set to be $5cm/s$, and the sensor numbers are varied for $n_f \in [0,80]$ and $n_a \in [0,20]$. The results are provided in Fig. 13.



(a) (b) (c)

Figure 13. Weighted mean performance for different sensor configurations: (a) Detection time lag; (b) Alarm distance; (c) Estimation error.

Several important results are displayed in Fig. 13. The most salient are the following:

i. Adding more sensors of any type will improve all performance metrics. However, this improvement is highly non-linear, and it indicates a decreasing marginal benefit with higher sensor density. Consider curve B (mobile sensors) in Fig. 13a for example, whereby a proverbial knee in the (performance) curve exists, and it occurs around approximately 3 mobile sensors. The incremental advantage of having a higher sensor density decreases beyond this point. The benefit of an increase from 1 to 3 mobile sensors is significant; in contrast, the benefit of an increase from 18 to 20 is negligible. This asymptotic behavior displayed in Fig. 13a reflects that it approaches the point of network performance saturation.

ii. The marginal improvement is different for static and mobile sensors (along the n_f-axis versus the n_s-axis). Compare, for example, curve A (static sensors) and

curve B (mobile sensors) in Fig. 13a whereby the detection time lag can be improved from over 100 min to 5 min with 80 static sensors (curve A), while the same level of improvement can be achieved by as few as 8 mobile sensors (curve B). The iso-cost performance of different sensor portfolios will be compared shortly.

iii. The same observations apply to all three performance metrics in Fig. 13b-c. The results display similar trends, although the exact shapes of the surfaces are different. For example, when comparing the static sensors when $n_a = 0$ in Fig. 13a (curve A) and 7b (curve C), the knee in the curve occurs at a smaller sensor density for alarm distance (around 45 static sensors) than detection time lag (around 60 static sensors). This indicates that despite having 45 static sensors, adding more can still improve the detection time lag considerably, although this does not improve the alarm distance in any meaningful way.

The previous results compare the performance of different HSNs and provide some insights into their performance landscape across (n_f, n_a). Next, the cost constraint is considered, and the optimal sensor portfolio mix β^* is identified for instances where the total budget is limited. An example contour plot of detection time lag is shown in Fig. 14, to illustrate how to identify the optimal sensor portfolios.

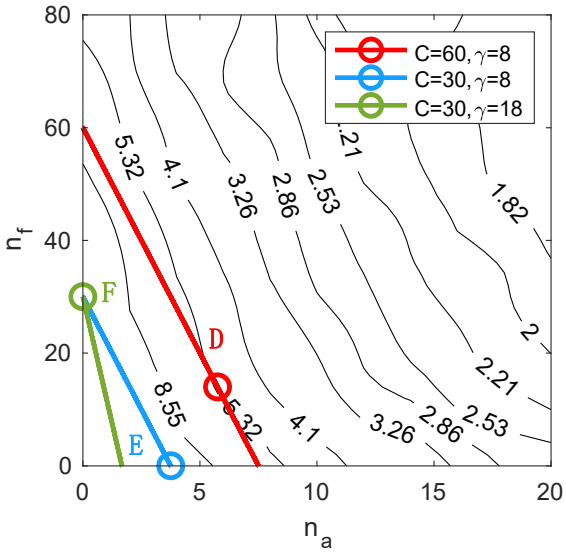


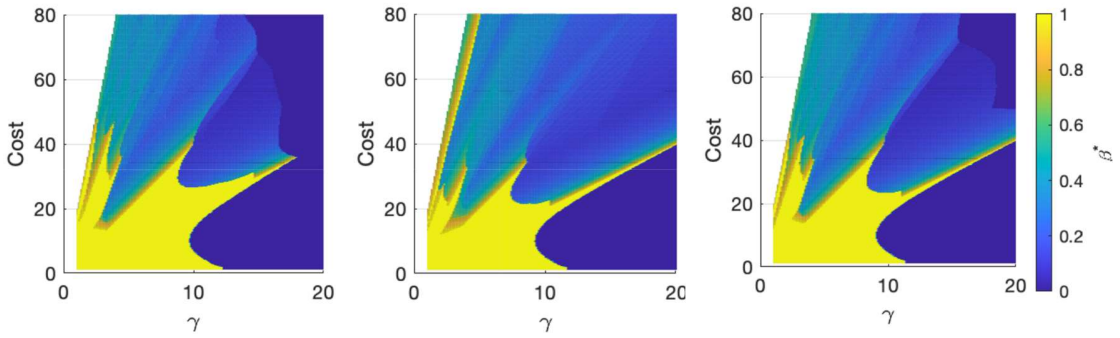
Figure 14. Performance contour of detection time lag with some constant-cost lines. The circles indicate the optimal sensor portfolio with the best iso-cost performance.

As noted in section 3.1, the cost constraints are represented by straight lines in the n_f-n_a space according to Eq. 23 (which is copied here for the readers' convenience):

$$C_{tot} = n_f + \gamma n_a \tag{26}$$

The intercept (on the n_f axis) indicates the sensor cost C, and the slope indicates the cost ratio γ. For example, line D in Fig. 14 has a sensor cost of 60 and cost ratio of 8. The optimal iso-cost sensor portfolios, as indicated by the circles in Fig. 14, are the points that provide the best performance on a given constant-cost line. Given the necessary condition of constrained optimization [45], the best sensor portfolio is obtained either at the tangent point (if there exists one) between a constant-cost line and a contour level (line D), or on one of the axes (line E and F). Note that points on the axes indicate either static or mobile sensor networks. The optimal portfolio mix β^* can be easily computed from the best portfolio points $\left(n_f^*, n_a^*\right)$. Note that for practical use with one given cost condition, the

optimal sensor portfolio can be found by a line search algorithm along the corresponding iso-cost line. The results, illustrated in Fig. 13, are generalized to a broad range of cost parameters (C, γ) to assess their effect on the optimal sensor portfolio. The results are provided in Fig. 15.



(a) (b) (c)

Figure 15. Optimal mobile sensor portfolio mix β^* for different cost parameters: (a) Detection time lag; (b) Alarm distance; (c) Estimation error. The mobile sensor speed is 5 cm/s.

The most important result observable in Fig. 15 is the existence of certain regions in the C-γ space, where hybrid sensor networks ($0 < \beta^* < 1$) provide the best monitoring performance. The other salient results in Fig. 15 are as follows:

i. When γ is large, mobile sensors are more expensive than static sensors. As a result, the optimal portfolio tends to include only static sensor networks with $\beta^* = 0$ (dark blue areas in Fig. 15).

ii. When γ is small, mobile sensors are more economical. For a limited range of total cost C, the optimal configuration tends to include only mobile sensor networks with $\beta^* = 1$ (yellow areas in Fig. 15).

iii. Outside these two conditions, for small γ and larger C than in (ii), the results in Fig. 15 indicate that hybrid sensor networks become the optimal configuration. One way of conceptualizing this result is that with larger C, the network density is higher, and as result, either purely static or mobile sensor networks approaches their saturation points. Thus, the incremental improvement from adding the same sensors in the network decreases dramatically. However, adding a different sensor still improves the performance, resulting in $0 < \beta^* < 1$ or a hybrid sensor network as the optimal portfolio mix.

Note that the results shown in Fig. 15 is conditioned on the specific simulation environment. The sharp jump of β^* from 1 to 0 when the cost is limited is because the sensor numbers are also limited with limited total cost. Under this circumstance, the saturation effect of adding more sensors of the same kind is negligible. Therefore, a purely static or mobile sensor network can have the optimal performance when the corresponding sensor is more economic depending on the cost ratio γ. Notice that the specific shape of the boundary between these two regions will be different for other environments, whereas the trend can be generalized to other environments.

Given the optimal portfolio mix, the performance–cost Pareto fronts can also be derived. The detailed results are discussed in Appendix B.

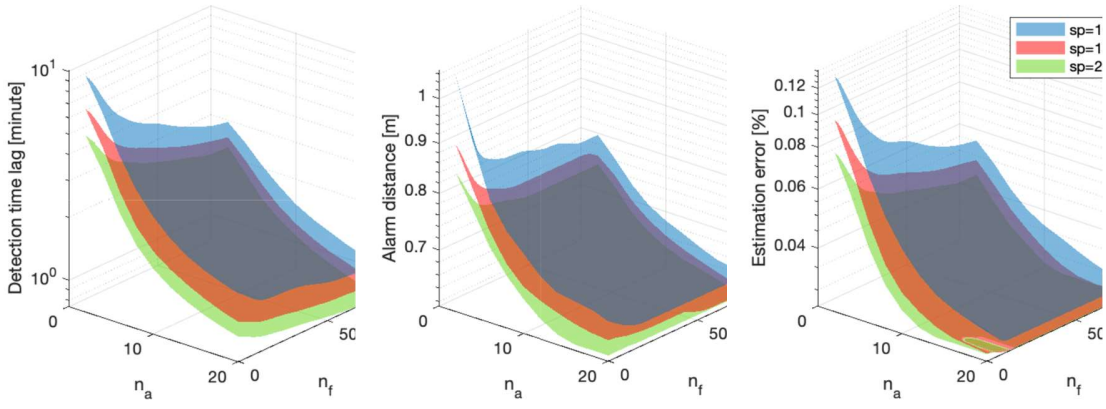
This series of experiments demonstrates that hybrid sensor networks (HSNs) outperform purely static and mobile sensor networks under specific cost conditions. The sensor portfolio mix β is a critical parameter that needs to be carefully assessed and optimized to achieve the ideal iso-cost performance. Next, another critical parameter, the

mobile sensor speed, is examined, and its impact on the network performance and the optimal portfolio mix is assessed.

3.2.2 Investigating the Impact of Mobile Sensor Speed on The Monitoring Performance

The speed of the mobile sensors is an important parameter that affects the network monitoring performance. The extent of this effect, however, is not negligible and will depend on the sensor density and portfolio mix in a complex manner. In the previous experiments, the mobile sensor speed was held constant at $5\ cm/s$. In the next series of experiments, the speed is set at $(10, 15, \text{and } 20)\ cm/s$ respectively, and the changes in monitoring performance for different sensor network portfolios are investigated.

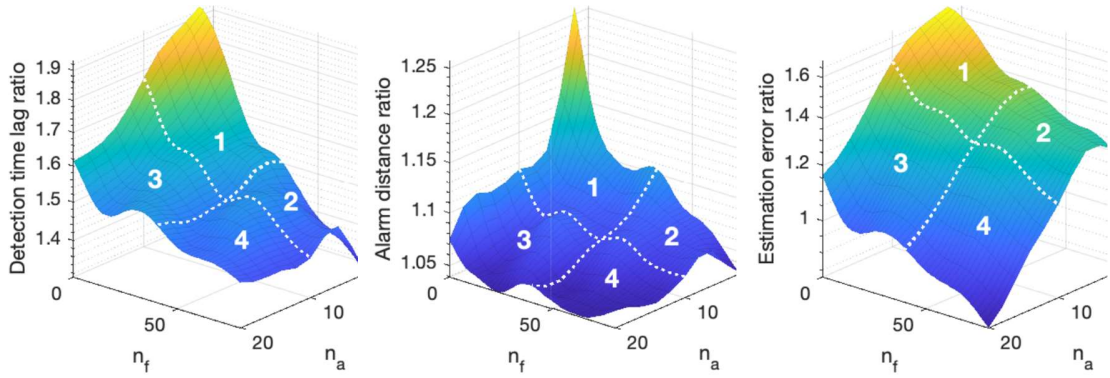
The performance surfaces for different mobile sensor speeds are provided in Fig. 16. Note that the purely static sensor networks when $n_a = 0$ are excluded since the moving speed does not affect their performance.



(a) (b) (c)

Figure 16. Performance comparison of mobile sensor speed for different sensor portfolios: (a) Detection time lag; (b) Alarm distance; (c) Estimation error. The view angle is different for this plot to show the three surfaces.

A similar general trend is observed for different mobile sensor speeds, such that higher moving speeds generally result in better monitoring performance across the three metrics. To better visualize this performance differential for different speeds, the performance ratio, for example, between 10 cm/s and 20 cm/s is computed. The results are shown in Fig. 17. The performance ratios between the other speeds share a similar trend. The n_f-n_a space is divided into four regions based on the sensor density to facilitate the discussion.



(a) (b) (c)

Figure 17. Performance ratio between 10 cm/s and 20 cm/s for different sensor portfolios: (a) Detection time lag; (b) Alarm distance; (c) Estimation error. The n_f-n_a plane is divided into four regions (1–4).

Several observations can be made based on the results in Fig. 17. The most salient ones are the following:

i. In region 1, where both static and mobile sensor densities are limited, the performance ratio is the highest among the four regions when the speed is increased from 10 cm/s to 20 cm/s. This is the greatest improvement achieved by increasing the moving speed. Consider the detection time lag ratio for example in Fig. 17a. The

highest ratio or performance differential is around 2 when the sensor density is small. This means that by doubling the mobile sensor speed, the detection time lag in this region is almost halved. This verifies the assumption that for the same environment, it takes half the time to search for an anomaly with a doubled speed.

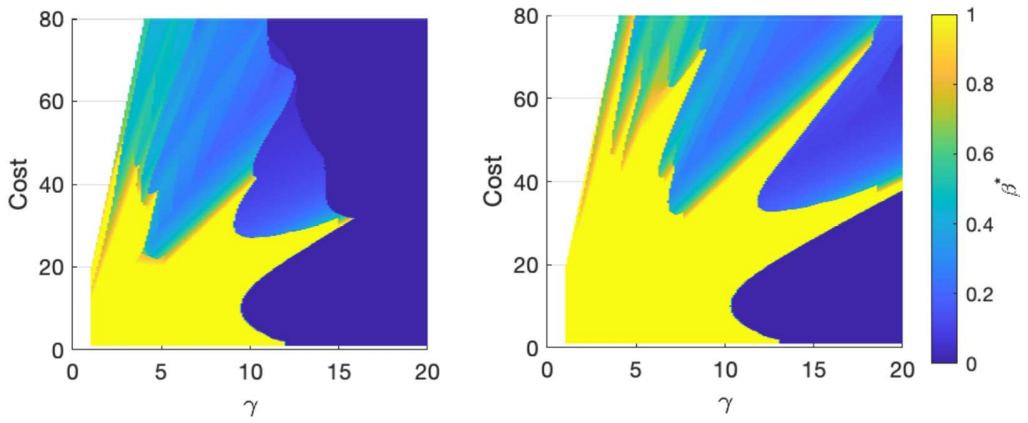
ii. When comparing region 2 with region 1, the static sensor density is higher. The detection time ratio decreases to around 1.4 because the mobile sensor proportion β is small and the majority of fires are detected by the static sensors. As a result, the mobile sensor speed does not affect the monitoring performance as drastically as in region 1.

iii. When comparing region 3 with region 1, the mobile sensor density is found to be higher. The detection time ratio decreases to around 1.6 because of the saturation effect. As discussed in section 3.2.1, after a saturation point, the incremental improvement obtained from additional sensors decreases. For a similar reason, the improvement achieved by increasing the mobile sensor speed decreases when the mobile sensor density is high.

iv. In region 4, both mobile sensor density and static sensor density are high. The performance ratio gradually plateaus, around and below 1.4.

In short, higher mobile sensor speed results in better monitoring performance, and that speed has a nonlinear effect on performance, which varies with sensor density and portfolio mix β. The optimal portfolio mix β^* also changes with the speed of mobile sensor. "Asymptotic thinking" helps to develop ideas about this observation. To illustrate this, assume first that the mobile sensor speed tends to 0, in which case, they provide a similar

performance of the static sensors but at a higher cost ($\gamma > 1$). They are therefore discarded in the optimization process, and the resulting optimal monitoring network will consist of purely static sensors ($\beta^* = 0$). On the other extreme, assume that the mobile sensor speed tends to infinity. In this hypothetical case, a single mobile sensor can be present at any location at any time and fulfill the function of an infinitely large static sensor network. Once the given budget allows for the acquisition of a single mobile sensor, the optimization process will switch to a purely mobile sensor network ($\beta^* = 1$). The reality will be somewhere between these two extremes, and the optimal portfolio mix β^* will change in relation to the speed of the mobile sensor. The comparative results of β^* in the C-γ space for mobile sensor speeds of 5 and 20 cm/s are provided in Fig. 18. Only the detection time lag is included to avoid visual clutter.



(a) (b)

Figure 18. Optimal mobile sensor portfolio mix β^* for different speeds: (a) $5cm/s$; (b) $20cm/s$.

The most significant result in Fig. 18 is that the region in C-γ space where mobile sensor networks are optimal ($\beta^* = 1$) is significantly larger when the sensor speed is

higher. It further expands/shrinks as the mobile sensor speed increases/decreases in accordance with the asymptotic thinking described previously.

Beyond these detailed results, it is useful to reflect on the main high-level findings this section demonstrated: (1) that a simulation-based analysis of sensor networks provides a more direct assessment of the ultimate objectives of specific monitoring and anomaly search tasks. It is myopic to restrict the analysis to indirect measures of the performance, such as area coverage; (2) that given specific cost parameters, hybrid sensor networks can provide significant and robust advantages over purely static or mobile sensor networks; (3) that identifying the optimal sensor portfolio mix requires careful analysis and is driven by cost parameters and mobile sensors speed.

CHAPTER 4. RESILIENCE

In this chapter, a new aspect of hybrid sensor network is considered, the resilience. The previous two chapters assume the sensors are robust and do not fail during the anomaly search and detection. When the sensor networks are deployed for a long time, the sensor failure probability also increases exponentially to a value that cannot be ignored. This chapter first introduced the new problem statement. It then presents the upgraded sensor deployment strategy to embed resilience consideration into HSN. Finally, it evaluates and compares the monitoring performance between a resilient HSN and its previous version.

4.1 Problem Formulation

In a given environment, there can be an initial anomaly starting at a random location according to a given distribution. We deploy n_f static sensors and n_a mobile sensors for the purpose of monitoring and anomaly search. These sensors have some failure probabilities, and some of them will fail before this environmental anomaly occurs. The deployment strategy includes the static sensor allocation and the mobile sensor path planning. The objective for the sensor network is to detect the anomaly in the shortest time possible. The performance of the sensor network is evaluated by the mean detection time of a series of computational simulations. We seek to assess and reduce the detection performance degradation in the presence of sensor failures.

To further investigate the problem, we make the following assumptions. First, we define the failure probability of the static and mobile sensors to be p_f and p_a respectively. Since the moving platform of the mobile sensors can also fail, mobile sensors have a higher

failure probability, $p_a > p_f$. Second, we assume the sensor failures are permanent and the failed sensors are removed from the network. Lastly, since the mean time to failure (MTTF) of the sensors (millions of hours) are much larger than the time it takes for the sensor network to detect the anomalies (typically less than one hour in our application), we assume the sensor failures only happen at the beginning of each test with a predefined failure probability. This can be conceptualized as that the failure occurs a certain amount of time after the deployment of the system that results in some proportion of the sensors failed.

We focus on two research objectives. First, we seek to develop a resilient static sensor allocation method that can increase the network robustness to sensor failures (O4). Second, we seek to develop a resilient mobile sensor path planning method to enable network recoverability from sensor failures (O5). We propose to combine these two methods and create a resilient collaborative hybrid sensor network. To address these questions, we introduce in the next section our proposed sensor deployment strategy.

4.2 A Resilient Hybrid Sensor Network (R-HSN) Deployment Strategy

The R-HSN shares common aspects with HSN, with resilient upgrades of its elements. Before we step into the technical details, it is helpful to discuss (qualitatively) why the performance of the original HSN degrades when sensor failures occur.

There are two main reasons for the performance degradation in the presence of sensor failures. First, in a realistic environment with a non-uniform risk distribution, the traditional static sensor allocation does not account for differential spatial risks, and it places the same number of sensors at high-risk and low-risk locations. This is an optimal allocation when the sensors are always operational—it implicitly assumes that they are perfectly reliable.

Should the sensors fail at high-risk locations, coverage is lost at these important spots. For anomalies in this region, it takes long for their associated signals (e.g., heat and smoke for fires) to propagate to farther sensors to be detected. This results in slower detection and leads to performance degradation. Second, although the original mobile sensors can autonomously adapt to the situation when sensors fail, they have to wait till the uncertainty increases at those locations previously covered by the failed sensors such that they can obtain more reward exploring their locations. The multi-agent system prevents the mobile sensors from being too close to the other sensors before they fail. In other words, current mobile sensors can only remedy the situation after failures occur.

We introduce next the resilient static sensor allocation and mobile sensor path planning methods for a resilient hybrid sensor network (R-HSN).

4.2.1 Resilient Static Sensor Allocation Method

In Eq. 4, we took the maximum kernel value because we assume the sensors are perfectly reliable. However, the sensor that provides the best kernel value can now fail. It is important to consider how likely this will happen and take into account the other kernel values when this happens. In other words, **the expectation of the kernel values should be used as the new coverage index** $K_r(l)$ to account for potential random failures. We develop a probability sum technique to calculate this expectation.

At each location l, there are n_f kernel values $G(l; s_i, \sigma)$, $i = 1 \sim n_f$, for the n_f static sensors. We reorder the sensor indices by their kernel value at l in the descending order such that $G(l; s_1, \sigma) \geq \cdots \geq G\left(l; s_{n_f}, \sigma\right)$. For simplicity, we denote the sorted kernel

values as $G_i = G(l; s_i, \sigma), i = 1 \sim n_f$. We denote the hardware reliability and failure probability of sensor i as Re_i and $P_i = 1 - Re_i$. Note that the order of the n_f kernels depends on the location l. We partition the circumstances by the conditional best kernel values as given in Table 10:

Table 10. Circumstances partitioned by the best kernel values

Conditional best	G_1	G_2	…	G_j
Sensor Circumstance	s_1 working	s_1 failed, s_2 working	…	$s_1 \sim s_{j-1}$ failed, s_j working
Probability	Re_1	$P_1 Re_2$	…	$Re_j \prod_{k=1}^{j-1} P_k$

Our new coverage index $K_r(l)$ is given by the probability sum:

$$Re_{tot,j} = Re_j \prod_{k=1}^{j-1} P_k \tag{31}$$

$$K_r(l) = \sum_{j=1}^{n} G_j Re_{tot,j} \tag{32}$$

The new weighted Gaussian coverage metric M_r is given by:

$$M_r(s) = \sum_{l \in L} w(l) \cdot K_r(l) \tag{33}$$

Does this result in a different static sensor allocation? As the value of hardware reliability Re_i decreases, the sensors are more likely to fail. As an outcome of this upgrade, more static sensors are allocated closer to the high-risk locations to ensure a reliable

coverage. We discuss our specific application in the next section (fire detection in a multi-room apartment), but for the time being we simply note in Fig. 19 the differences in static sensor allocation when their failure probability is accounted for. In the presence of a non-uniform risk map of the environment to be monitored, accounting for sensor failure probability leads to different sensor allocations (note, for example, the differences around the R1–R6 in Fig. 19). In section 4.3, we examine how the upgraded sensor placement improves the monitoring resilience of the sensor network with sensor failures.

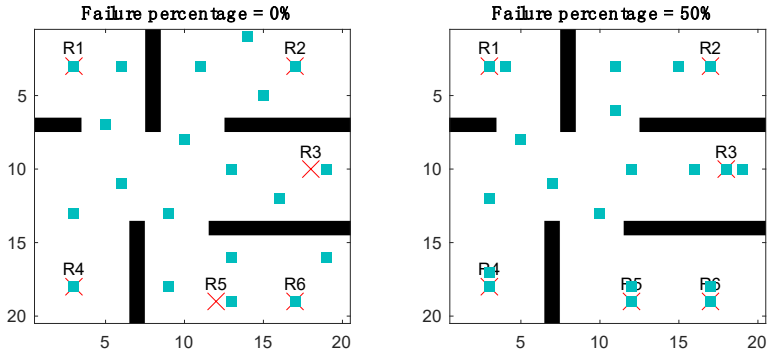


Figure 19. Example static sensor allocation with different sensor reliability. The crosses indicate the high-risk locations, and the squares indicate the sensor allocation.

4.2.2 Resilient Mobile Sensor Path Planning Method

The upgrade to the path planning is based on a similar probability sum calculation as in the resilient static sensor allocation. There are two modifications, which together provide resilience in the face of sensor failures. First, for the multi-agent system, we formerly took the minimum ker_j value around the other sensors as the reward scaling down effect to avoid clustering, as in Eq. 20. If a sensor that provides the minimum kernel values over some area fails, scaling down the reward in this area will wrongly direct the other mobile sensors away from this area, which results in longer response time. Therefore, the

minimum value should be replaced by the expectation of the reward scaling process to reflect the sensor failure probability. We use a similar probability sum technique as described in the resilient static sensor allocation method. The difference is to replace the coverage kernel G_i by the reward scaling kernel ker_j and reverse the kernel order in Table 10 because now the conditional "best" for a minimization process should start with the smallest kernel value. The probability sum of the reward scaling process has the same form as in Eq. 28, but the order of the probabilities is different. The new adjusted reward is given by:

$$R_{r,t,i}(l) = R_{r,t}(l) \cdot \sum_{j=1,j\neq i}^{n} ker_j\left(l; s_j, \sigma\right) Re_{tot,j} \tag{34}$$

Second, the original HSN considers three terms representing the state estimation, the estimation uncertainty level, and the cumulative anomaly risk in the reward function given by Eq. 15. We now add the anomaly risk distribution $w(l)$ as the fourth term in the upgraded reward function $R_{r,t}$ representing the reliability at each location. Note that the anomaly risk distribution $w(l)$ is different from the cumulative anomaly risk $\Phi_t(l)$. The former is a static probability of having an anomaly during a unit time step, whereas the latter is a dynamic cumulative probability since the last time of visit at this location l. The upgraded reward function $R_{r,t}$ is given by:

$$R_{r,t} = E_t + \alpha_t U_t + \beta_t \Phi_t + \epsilon_t w \tag{35}$$

where ϵ_t controls the balance between the exploitation, exploration, and reliability considerations. Note that the anomaly risk distribution w does not change with time as the other three terms.

After adjusted by the reward scaling process, which is time varying according to the mobile sensor locations, the fourth term serves as a dynamic indicator of how reliably covered each location is. This reward will be higher around high-risk locations, when the sensors are far away, or the failure probabilities are high, which will attract the mobile sensors to visit. Note that the anomaly risk increases faster for locations with high anomaly risk probability. Therefore, the fourth term can be seen as a generalized derivative of the third term. It predicts the increase of the anomaly risk and make the mobile sensors move to the high-risk locations sooner if they are not reliably covered as a means of precaution depending on the value of Re_i. It also makes the exploration over the low-risk area more efficiently when the high-risk locations are already reliably covered.

This upgrade improves the resilience of the mobile sensors by providing precautions to potential sensors failure before they occur such that their response time can be reduced. Next, we will see how this idea of "planning ahead" benefit the autonomous recovery when failures occur.

4.3 Computational Experiments

To provide a direct assessment of the network resilience in terms of the monitoring performance, we reuse the same simulation environment for our application. At the start of each simulation, an initial fire location is randomly selected according to the fire anomaly risk distribution $w(l)$. Some static sensors and mobile sensors are randomly failed

according to their failure probabilities p_f and p_a. We run the simulation until the fire is detected by one of the sensors and use the detection time as the performance metric. For the robustness of the result, we conduct the experiment for 50 independent initial fire locations as shown in Fig. 20. We repeat each one for several hundred times to account for the randomness in sensor failures. We examine the weighted mean detection time for robustness of the results. This weighted mean performance is equivalent to the mean performance of a large number of simulations with initial fires sampled according to the risk map. Note that the method can also be applied to other metrics such as the worst-case performance or variance of the performance with proper parameter adjustment.

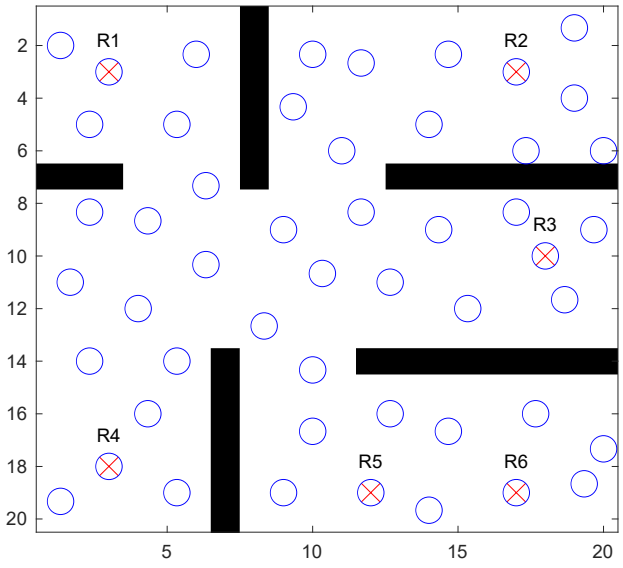


Figure 20. 50 independent initial fire locations indicated by blue circles. The ones coinciding with the 6 red crosses (R1-R6) have 10 times higher fire risk.

To investigate performance degradation given different sensor failure probabilities, we conduct a series of computational experiments. We vary the static sensor failure probability p_f between $[0, 2.5, 0.5, 7.5, 10]\%$. Since the mobile sensor failure probability

p_a should be higher because the additional moving platform can also fail, we set $p_a = 2p_f$ for simplicity. This can be easily adjusted in the future for other applications and user preferences.

To assess the effect of the resilient upgrades, we first conduct the experiments with a traditional static sensor network as the baseline. We then benchmark the performance of a resilient static sensor network and a resilient hybrid sensor network against this baseline performance. The sensor numbers are given in Table 11, and they are carefully chosen for fair comparison. Because of the moving platform, mobile sensors presumably cost several times more than static sensors. These cost parameters are chosen as an example to illustrate the effect of including mobile sensors on the network resilience. For a more comprehensive cost-performance tradeoff analysis on the sensor numbers, the readers are referred to [76].

Table 11. Sensor numbers in the networks.

Sensor network	(Resilient) static	Resilient hybrid
Static sensors n_f	40	10
Mobile sensors n_a	0	6

Next, we present and discuss the results of the network resilience analysis and improvement.

4.4 Results and Discussion

We divide the results into two parts for the static and hybrid sensor networks. They correspond to the two research objectives set forth in the Introduction. We first compare the performance of the traditional static sensor network (baseline) and the resilient one to

evaluate the improvement of the new static sensor allocation strategy provides (O4). We then examine the resilient mobile sensor path planning deployment and assess the performance of the resilient hybrid sensor network (O5).

4.4.1 Static Sensor Network

The detection time performance of the static sensor network is shown in Fig. 21. We denote the original and resilient deployment of the static sensor network as SSN and R-SSN respectively. The SSN performance serves as the baseline performance for our later resilience comparisons of deployment strategies.

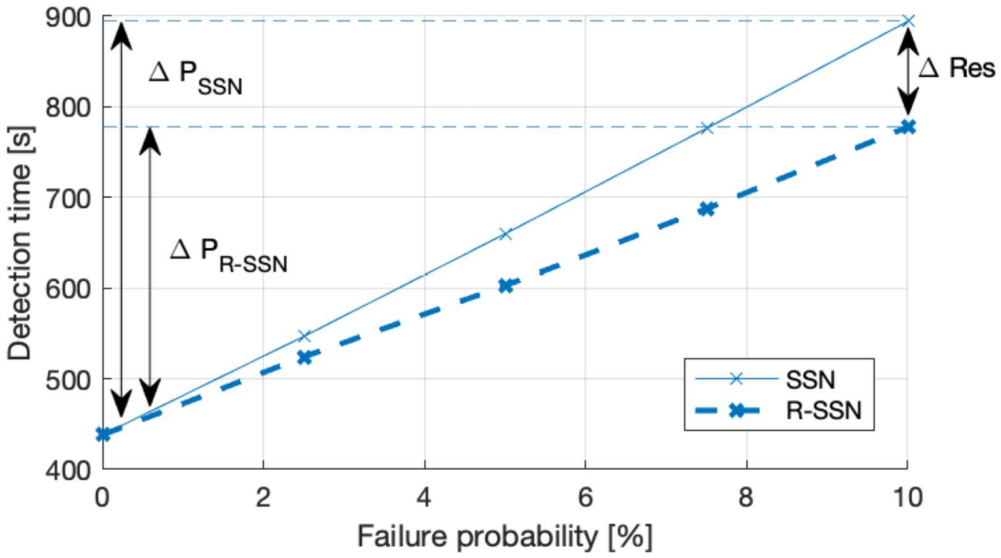


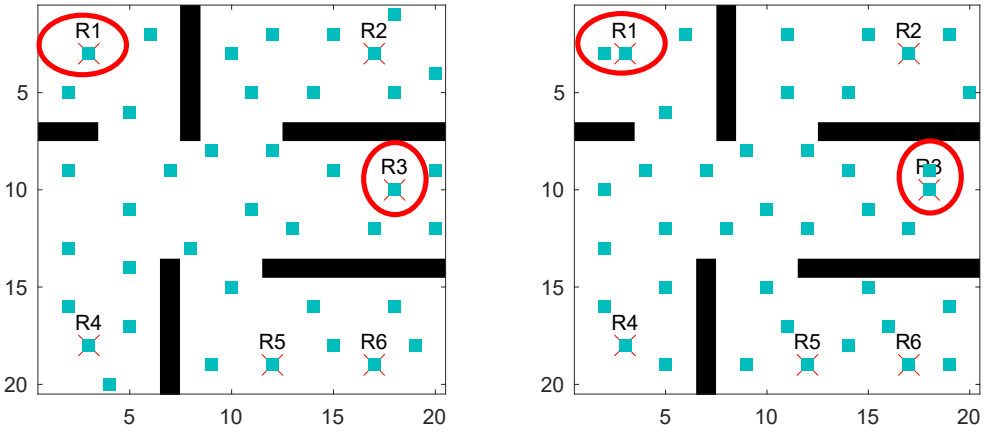
Figure 21. Detection time for the static sensor network with different failure probabilities. The original and resilient deployments are indicated by the solid and thick dashed lines.

We discussed previously that accounting for the sensor failure probabilities upfront in the static sensor allocation phase results in different sensor placement. The results in Fig. 21 clearly demonstrates that this embedded resilience provides performance advantage, or

more accurately, a reduction in monitoring performance degradation given sensor failure probability. Note that the x-axis in Fig. 21 represents the failure probability of an individual sensor and the y-axis the monitoring performance of the network, here the detection time.

We observe that the detection time increases with increasing sensor failure probability for both the SSN and R-SSN. This is expected as higher sensor failure probabilities will further degrade the detection performance. But the extent of performance degradation is the important aspect for our purposes. Note, for example, that for a 10% sensor failure probability, these can be due to random failures or deliberate attacks, the performance degradation with the standard sensor placement results on average in $\Delta P_{SSN} = 456\,s$, whereas with the resilient sensor placement the performance degradation is $\Delta P_{R-SSN} = 339\,s$. The difference between these two performance degradations is one aspect of resilience, in this example, $\Delta Res = \Delta P_{SSN} - \Delta P_{R-SSN} = 117\,s$. In other words, accounting for resilience in the static sensor allocation provides a 26% reduction in performance degradation over the standard sensor allocation when 10% of the sensors in the network fail. This is a meaningful difference, and it is forfeited in traditional static sensor allocation.

To better understand why the R-SSN deployment mitigates the performance degradation, we examine the sensor allocation of both methods shown in Fig. 22.

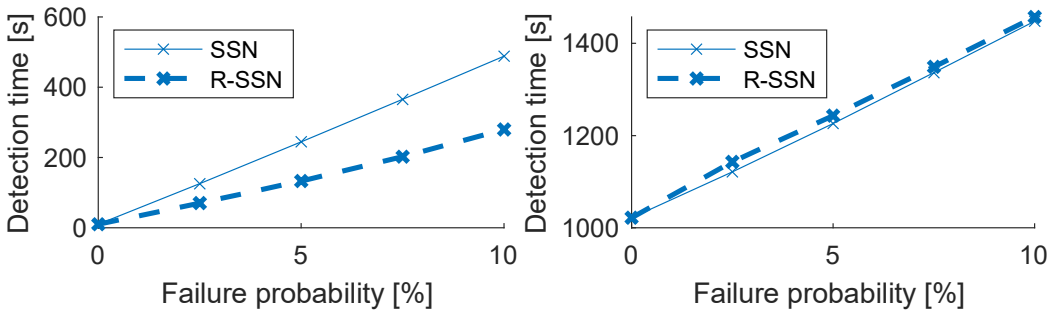


(a) SSN deployment (b) R-SSN deployment with $p_f = 10\%$

Figure 22. Static sensor allocation (indicated by squares) for $n_f = 40$ with the original and resilient deployment strategies.

We observe that both methods allocate sensors at the six high-risk locations. The SSN has the remaining sensors spread out more evenly over the rest of the area, whereas the R-SSN places redundant sensors around some high-risk locations (R1 and R3). The reason for this different placement is that when accounting for failure probabilities, more sensors closer to the high-risk locations increase the expected coverage index in Eq. 29. The exact number of redundant sensors and how close they should be to which high-risk locations are optimally selected by the R-SSN deployment to maximize the overall coverage metric in Eq. 30. This resultant (pseudo-)redundancy from the optimization process reduces the probability that all static sensors closer to one particular high-risk location fail, thereby mitigating the performance degradation at these high-risk locations. Note that at the same time, this redundancy inevitably sacrifices the detection performance for the rest of the area. The tradeoff is optimally balanced. This is illustrated by comparing the mean detection time for fires at the high-risk and low-risk locations separately, as

shown in Fig. 23. The performance at the high-risk locations with 10% sensor failures degrades less with R-SSN (200 seconds shorter detection time) with only a little sacrifice at the low-risk locations (9 seconds longer detection time).



(a) High-risk locations (b) Low-risk locations

Figure 23. Detection time of SSN and R-SSN for fires at high-risk and low-risk locations.

Note that this different sensor placement depends on the anomaly risk distribution. On the one hand, for an unknown or uniform distribution, R-SSN results in the same sensor placement as the SSN, because there is not a location with a higher risk to place redundant sensors. Since most realistic environments have non-uniform distributions, this observation reflects the importance of obtaining the risk distribution beforehand to improve the monitoring performance. On the other hand, if the risk distribution has a bigger probability difference, R-SSN results in more sensors closer to the high-risk locations.

The main finding in this subsection is that considerations of sensor failure probabilities in the network design phase have significant impacts on the sensor placement and monitoring performance. Accounting for sensor failure probabilities upfront can

dramatically improve the network resilience through more judicious, failure-informed allocation.

Next, we evaluate the detection performance for the resilient hybrid sensor network and compare it with the static sensor network.

4.4.2 Hybrid Sensor Network

The detection time performance of the hybrid sensor network is shown in Fig. 24. We denote the resilient deployment of the hybrid sensor network as R-HSN.

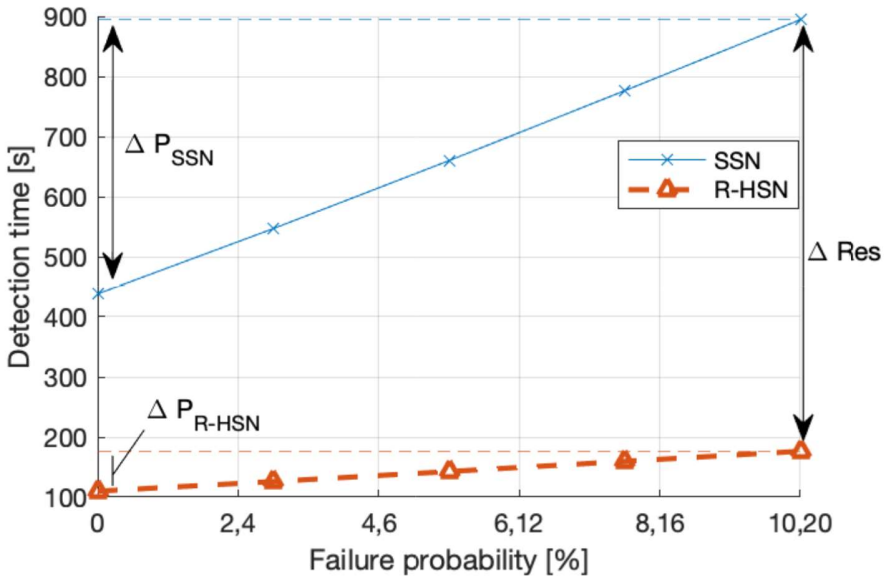


Figure 24. Detection time for the hybrid sensor network with different failure probabilities. The original SSN and the resilient HSN are indicated by the solid and dashed lines. The two numbers on the x-axis are for static and mobile sensor failure probabilities p_f and p_a respectively.

Fig. 24 demonstrates a significant resilience advantage obtained when mobile sensors are introduced into the monitoring network. First, note that even before considering resilience, a HSN provides significant performance advantage over a static sensor network

(the detection time at 0% failure probability, for example, is roughly 100s for the HSN and about 450s for the static sensor network). This monitoring performance dominance is maintained by the increased resilience of the HSN across all sensor failure probabilities. More specifically, when the static and mobile sensor failure probabilities are $p_f = 10\%$ and $p_a = 20\%$, the monitoring performance degradation is $\Delta P_{SSN} = 456\ s$ for the SSN, and $\Delta P_{R-HSN} = 66\ s$ for the R-HSN. This reduction in performance degradation, 85% in this example, is a reflection and a result of the increased resilience provided by the HSN and its deployment strategy (R-HSN).

Although cost considerations are not examined in this work, an extensive cost-performance analysis of HSN can be found in [76]. The particular networks selected here (static and hybrid) and shown in Table 11 were chosen because they are iso-cost under some assumptions discussed in [35].

To better understand how the mobile sensors contribute to the monitoring performance and network resilience, we examine one example simulation shown in Fig. 25.

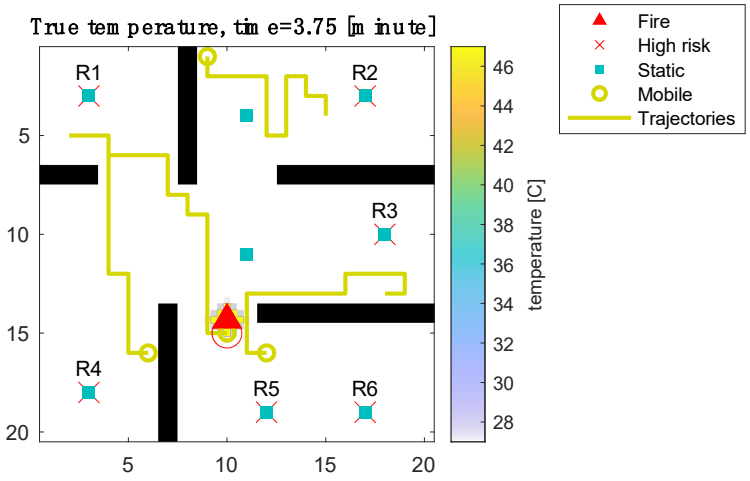


Figure 25. One example experiment for the resilient hybrid sensor network. The experiment stops when the fire is detected by one of the sensors (circled in red).

In this experiment, two static sensors and two mobile sensors have failed. The remaining 8 static sensors and 4 mobile sensors are operational. The high-risk locations are covered by the static sensors and the mobile sensors search the remaining areas. The advantage of mobile sensors is demonstrated through their path flexibility. When sensors fail, the static sensors cannot adjust to the resulting situation. The redundant sensors placed around the high-risk locations (if they exist) are wasted if the failed sensors are not in these regions. In contrast, the mobile sensors can adapt to the situation depending on whichever sensor fail (in any location). The fourth term of the reward function in Eq. 32 provides the information of the risk distribution for the mobile sensors such that they are constantly aware of whether each location is robustly covered or not. They react autonomously to recover the monitoring performance by visiting the area previously covered by the failed sensors. This path planning flexibility provides resilience for hybrid sensor networks.

The combination of the resilient static sensor allocation and mobile sensor path planning provides a general deployment strategy for arbitrary hybrid sensor networks and

provides resilient monitoring performance. Beyond these detailed analyses, it is useful to recap the main, high-level findings this section demonstrated: (1) that sensor failures can have a significant effect on the monitoring performance of sensor networks; (2) that sensor network deployment should be carefully designed to mitigate performance degradation following sensor failures. Mobile sensors play an important role in improving resilience, and hybrid sensor networks can provide significant resilience advantages over traditional static sensor networks; (3) incorporating resilience in the deployment of sensor networks is important to cope with sensor failures. It requires careful analysis and can provide significant advantages.

CHAPTER 5. CONCLUSION

5.1 Summary

Hybrid sensor networks (HSN) consist of both static and mobile sensors deployed to fulfill a common monitoring task. The hybrid structure generalizes the network's design problem and offers a rich set of possibilities for a host of environmental monitoring and anomaly detection applications. HSN also raise a new set of research questions. Their deployment and optimization provide unique opportunities to improve the network's monitoring performance and resilience. This thesis addressed three challenges associated with HSN related to the *collaboration*, *optimization*, and *resilience* aspects of the network. Broadly speaking, these challenges revolve around the following questions: (1) how to collaboratively allocate the static sensors and devise the path planning of the mobile sensors to improve the monitoring performance? (2) how to select and optimize the sensor portfolio (the mix of each type of sensors) under given cost constraints? And (3) how to embed resilience in a HSN to sustain the monitoring performance in the face of sensor failures and disruptions?

In Chapter II, *collaboration*, this thesis developed a novel deployment strategy for HSN. The strategy solved the static sensor allocation problem, the mobile sensor path planning problem, and most importantly, the collaboration between these two types of sensors. Previous research in this area has addressed these problems separately in simplified environments. In this thesis, a collaborative deployment strategy of HSN was developed to improve the ultimate monitoring performance in complex environments with obstacles and non-uniform risk distribution.

In Chapter III, *optimization*, this thesis addressed the HSN sensor portfolio selection problem. It investigated the tradeoff between the static and mobile sensors to achieve the optimal monitoring performance under different cost constraints. Previous research in this area has studied the optimization problem for networks with a single type of sensor. In this thesis, a general optimization problem was formulated for HSN with static and mobile sensors and solved to identify the optimal portfolio mix and its main drivers.

In Chapter IV, *resilience*, this thesis identified monitoring resilience as a key feature enabled by HSN. This part focused on the performance degradation of HSN in the presence of sensor failures and disruptions, and it identified the means to embed resilience in a HSN to mitigate this performance degradation. Monitoring resilience was achieved by accounting for potential sensor failures in the deployment strategy of both static and mobile sensors through a novel, carefully designed probability sum technique. Previous research in this area has examined the reliability problem from a coverage point of view. This thesis extended the scope of investigation of HSN from reliability to resilience, and it shifted the focus from coverage considerations to the actual monitoring performance (e.g., detection time lag) and its resilience in the face of disruptions.

To demonstrate and validate this novel perspective on HSN and the associated technical developments, this thesis focused on two examples of fire detection in a multi-room apartment using temperature sensors and CO leak detection in a 3D space station module with ventilation system. Three metrics are adopted as the ultimate monitoring performance, namely the detection time lag, the anomaly source localization uncertainty, and the state estimation error. Two simulation environments based on the advection-conduction propagation model were developed for the computational experiments. The

results (1) demonstrated that the optimal collaborative deployment strategy allocates the static sensors at high-risk locations and directs the mobile sensors to patrol the rest of the low-risk areas; (2) identified a set of conditions under which HSN significantly outperform purely static and purely mobile sensor networks across the three performance metrics here considered; and (3) established that while sensor failures can considerably degrade the monitoring performance of traditional static sensor networks, the resilient deployment of HSN drastically reduces the performance degradation.

5.2 Contributions

The contributions of this thesis are briefly summarized next. They consist in the following:

- Development of the weighted Gaussian coverage (WGC) metric for the static sensor allocation problem. Proof of submodularity of the WGC metric and the development of a greedy WGC approach. The WGC approach are more suitable for environments with non-uniform fire risk distribution compared with the entropy and mutual information approaches. The WGC provides a better monitoring and anomaly detection performance. The WGC scales linearly with respect to the sensor numbers and are much faster than the benchmarked approaches.
- Development of the Markov Decision Process based policymaker for the mobile sensor path planning problem. The dynamic value iteration (DVI) approach iteratively solves for the next optimal moves based on a dynamic and parameterized reward function formulated by the estimation and uncertainty

information and tuned directly using the ultimate monitoring performance. A multi-agent system is developed by scaling down the reward function around the other sensors while solving the next move for each mobile sensor.

- Development of the sensor portfolio optimization problem for hybrid sensor networks with cost constraints. This thesis identifies the existence of certain cost conditions where a hybrid sensor network outperforms purely static or mobile sensor networks. This thesis demonstrates the saturation effect of deploying multiple sensors of the same kind. It investigates the effect of mobile sensor speed on the optimal sensor portfolio. The results demonstrate that higher mobile sensor speed expands the region where the optimal network structure is purely mobile.
- Development of the resilience upgrade of the hybrid sensor network deployment strategy. This thesis demonstrates that traditional static sensor networks without resilience consideration can suffer from a huge performance degradation with sensor failures. The resilience upgrade involves a probability sum technique to compute the expectation of the sensor utility function and the reward function. The resilience strategy places redundant sensors around high-risk locations and directs mobile sensors toward high-risk locations as a precaution for sensor failures. The resilient deployment significantly mitigates the performance degradation with sensor failures. The results also demonstrate that hybrid sensor networks are more resilient to sensor failures and have smaller performance degradation.

- Development of two simulation environments for fire search and temperature monitoring in a multi-room 2D apartment and CO leak detection in a 3D space station module with ventilation system. The HSN deployment strategy is evaluated and benchmarked using these two simulation environments. They also serve as potential benchmark platforms for other sensor network deployment strategies.

5.3 Future Work

The current HSN problem setup can be expanded and upgraded to include considerations for more general applications. Four directions are proposed for example.

5.3.1 New Applications

The current work applies HSN to fire detection in an apartment with temperature measurements and CO leak in a space station module with ventilation system. The second simulation environment provides a platform for many other applications whose dynamics are also based on advection and conduction processes. With parameter changes, the environment can be used to test the effect of different geometries, target variables, ventilation speed, mobile sensor speed, and sampling frequency of the sensors (e.g., scanning speed). HSN can be applied to other environmental monitoring problems such as smoke, CO2, VOCs, and AQI. For example, Georgescu et al. [77] examines the CO2 accumulation effect in astronaut's crew quarters on the International Space Station. The placement of CO2 sensors is critical to early detection of excess CO2 concentrations. The HSN can also be applied to large-scale geoscience monitoring tasks such as ocean monitoring, agriculture monitoring, and weather monitoring. These detection targets share

similar dynamics as heat propagation, whereby the current HSN deployment strategy can be suitably migrated.

5.3.2 Detection Kernel for WGC

The current kernel used in the WGC metric is Gaussian. It can be a good approximation for detection time for specific environment dynamics, i.e., pure conduction. However, for more complex environments, such as one with ventilation in the CO leak example, a Gaussian kernel may not be the optimal approximation for detection time. It is faster to detect information for a sensor downstream the anomaly and not the other way. Because the air flow will bring related information (e.g., smoke concentration for fire detection) toward a sensor downstream and away from a sensor upstream. One direction is proposed to address this issue with an adjusted "detection kernel" (DK) $K_D(l; s_i)$. It can be derived from the covariance matrix either computed directly from the training data or a learned one $K_D(l; s_i) = Cov_l(l, s_i)$, where Cov_l is the covariance of an individual CO leak at location l. Note that the detection kernel K_D is intrinsically different from the covariance kernel Cov_l of each single CO leak. The former reflects the covariance between anomalies at each location and the sensor at s_i, whereas the latter reflects the covariance between a fixed anomaly and the rest of the space. Essentially, the detection kernel describes how easy it is for a sensor at s_i to detect anomalies from the rest of the space.

Preliminary results of WGC using the adjusted detection kernel is shown in Fig. 26 and Table 12. It can be observed that the WGC-DK approach merges the WGC and variance approaches. For 3 sensors, it places a sensor at the downstream outlet vent to cover R1 and R3 simultaneously and the other two sensors upstream at R2 and R4. It balances

the considerations that high-risk locations should be covered and that downstream locations can detect anomalies from a larger space.

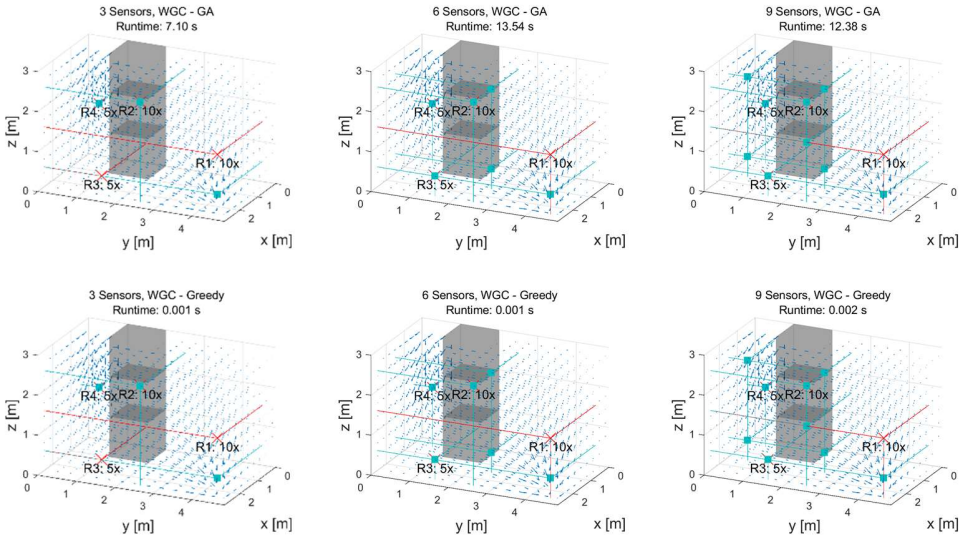


Figure 26. Sensor placement results of 3-9 sensors for WGC - DK and its greedy approaches.

Table 12. Mean detection time of 3-9 sensors for 39 fires with different approaches.

Methods	Mean detection time [s]		
	3 Sensors	6 Sensors	9 Sensors
WGC - DK	108.33	76.33	63.85
WGC - DK - Greedy	**108.33**	**76.33**	**63.85**
WGC	130.58	93.92	81.85
WGC - Greedy	130.58	93.84	75.68

5.3.3 Continuous Space and Environment Dynamics

The current environment is modeled as a grid world to reduce computational burden in the simulation. Ideally, both the static and mobile sensors should be placed or move in a continuous space. This consideration involves potential upgrades from three aspects. First, for the sensors to take measurements in a continuous space, the simulation model needs to provide a continuous state for the measured variable. Since usually the environment

dynamics does not have an exact solution, this can only be done via interpolation in a discretized simulation. Second, the static sensor allocation (SSA) algorithm needs to be upgraded. Since the weighted Gaussian coverage (WGC) metric is continuous intrinsically, only the optimization process needs to be replaced by a continuous one. Third, the mobile sensor path planning needs to be upgraded. This requires the reward function, the mobile sensor moving mechanism, and the planning algorithm all to be continuous. The reward function can be continuous using a combination of interpolation and Gaussian process regression from the discretized model. The other two parts requires a proper continuous motion model and motion control which is not the focus of this work.

5.3.4 Multi-type-Sensor-Multi-Anomaly

Besides these considerations and upgrades within the current problem setup, the HSN problem can be expanded to a more ambitious multi-type-sensor-multi-anomaly (MSMA) version. The mobile platforms usually have a heterogeneous suite of sensors that can take multiple kinds of measurements to detect one or more kinds of anomaly. The examination of SSA and MSPP in this new situation with multi-objective optimization can be performed in future studies.

APPENDIX A. SUBMODULARITY OF THE WEIGHTED GAUSSIAN COVERAGE METRIC AND ITS GREEDY VERSION

A set function is submodular if for all $A, B \subseteq L$, a finite set of all possible sensor locations, it holds that $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$. Essentially, for sensor selection, submodulariry means that adding a sensor has diminishing improvement when the selected sensor set is bigger. The proof of submodular of the WGC metric is provided below.

Recall that the WGC is a weighted sum of the coverage index $K(l)$ at each location l. The coverage index is the maximum of some Gaussian kernel G placed at each sensor location s_i. The first step is to prove that the coverage index function K is submodular at each location l.

$$G(l; s_i, \sigma) = exp\left(-\frac{d(l, s_i)^2}{2\sigma^2}\right), \qquad l, s_i \in L$$

$$K(l; s) = \max_{s_i \in S} G(l; s_i, \sigma), \qquad s = \{s_1, \dots, s_n\}$$

$$U(s) = \sum_{l \in L} w(l) \cdot K(l; s)$$

At each location l, let $K_A = K(A) = K(l; A) = \max_{s_i \in A} G(l; s_i, \sigma)$, then

$$K_{A \cup B} = K(l; A \cup B) = \max_{s_i \in A \cup B} G(l; s_i, \sigma)$$

$$= \max\left(\max_{s_i \in A} G(l; s_i, \sigma), \max_{s_i \in B} G(l; s_i, \sigma)\right) = \max(K_A, K_B).$$

Let $C = A \cap B, A' = A \backslash C, B' = B \backslash C$, then $A = A' \cup C, B = B' \cup C$,

$$L = K(A) + K(B) = K_{A' \cup C} + K_{B' \cup C} = \max(K_{A'}, K_C) + \max(K_{B'}, K_C)$$

$$R = K(A \cup B) + K(A \cap B) = \max(K_{A'}, K_{B'}, K_C) + \max(K_C)$$

Because A', B' are symmetric in the equation, without loss of generality, let $K_{A'} \geq K_{B'}$ and only the magnitude of K_C needs to be considered:

Case 1: $K_{A'} \geq K_{B'} \geq K_C$

$$L = K_{A'} + K_{B'}, R = K_{A'} + K_C, L \geq R$$

Case 2: $K_{A'} \geq K_C \geq K_{B'}$

$$L = K_{A'} + K_C, R = K_{A'} + K_C, L \geq R$$

Case 3: $K_C \geq K_{A'} \geq K_{B'}$

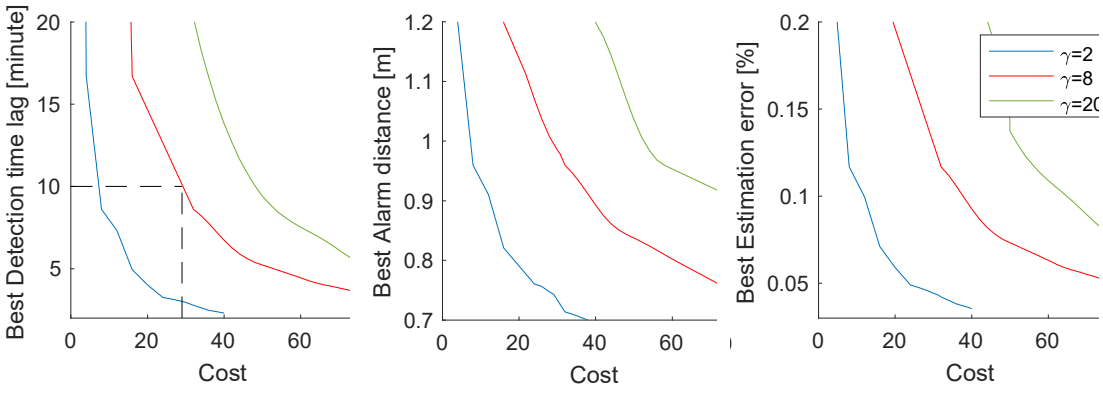
$$L = K_C + K_C, R = K_C + K_C, L \geq R$$

Therefore, $L \geq R$ for all cases and the coverage index function K is submodular at each location l. Since submodular functions are closed under linear combinations, the WGC, which is a linear combination of the coverage index function K, is also submodular.

With the submodularity of WGC, its greedy solution is guaranteed to have a $1 - 1/e$ performance. The greedy algorithm places the next sensor s_i at the location where the overall coverage is maximized, $s_i = \underset{s'}{argmax}\big(WGC(s_1, \ldots, s_{i-1}, s')\big)$.

APPENDIX B. PERFORMANCE-COST PARETO FRONTS FOR HYBRID SENSOR NETWORKS

The previous results presented in Fig. 15 identify the optimal sensor portfolio mix under various cost conditions in the C-γ space. However, what happens in instances where the optimal portfolio mix is no longer the primary focus, and the interest lies instead in identifying the best monitoring performance that can be achieved for a given budget C? This performance–cost multi-objective optimization problem is considered next, and the performance–cost Pareto fronts are identified. This analysis can help to inform several decisions, for example, to determine the required cost to meet a minimum performance requirement (e.g., a detection time lag of less than 10 min); or to assess whether a small increase in C can help "buy" a significant improvement in monitoring performance or not. To help address these decision problems, the performance–cost Pareto fronts for different γ are provided in Fig. 27.



(a) (b) (c)

Figure 27. Performance-cost Pareto front of different γ: (a) Detection time lag; (b) Alarm distance; (c) Estimation error. The n_f-n_a plane is divided into four regions (1–4). Larger γ represents current technology (expensive moving platforms). Smaller γ reflects the situation when moving platforms become more affordable. Note that the optimal portfolio mix β^* varies along these Pareto fronts.

Fig. 27 illustrates that the Pareto front with smaller γ dominates those with a larger γ Pareto front. Cheaper mobile sensors are favorable for their mobility and flexibility, and they can provide better performance. For the same cost, better monitoring performance is achieved with smaller γ. This reflects the changes in the optimal portfolio mix with the addition of mobile sensors. Moreover, it can be directly read from the Pareto front the network cost required to meet some performance requirement. For example, when $\gamma = 8$, a detection time lag of less than 10 min requires a cost of around 30 (red curve and dashed lines in Fig. 27a). Furthermore, the Pareto front reveals whether it is worth increasing the budget for the sensor network. For example, when $\gamma = 8$, an increase in C from 20 to 25 can lead to considerable performance improvements for the detection time lag (and other performance metrics), whereas an increase from 60 to 65 will result in a minor improvement.

APPENDIX C. APPLICATION OF MODEL PREDICTIVE CONTROL TO THE ORIGINAL MOBILE SENSOR PATH PLANNING IN HSN

The current mobile sensor path planning is solved by optimizing the reward function at each time step using a dynamic value iteration (DVI) method. It is optimal spatially at each time step but "greedy" in time because the reward function is fixed. One way to improve the path planning is to introduce the idea of model predictive control (MPC) into the algorithm. Since the moving mechanism is relatively simple in our case, the major prediction is about the environment dynamics. Specially, the reward function is currently formulated by three terms accounting for state estimation, uncertainty level, and anomaly risk. These terms change with time as the mobile sensors move around. Incorporating the idea of MPC needs to predict and update these terms during the DVI in the planning process. The prediction of the three terms can be done using the aforementioned estimators.

Let p be the prediction horizon, then at each time t, we can derive the predicted reward function based on the three terms for p steps, $R_t^k(s)$, $k = 1,2,\dots,p$. Here the subscript denotes the current time t for computing the next optimal moves (at $t+1$), and the superscript denotes the prediction steps k at this current time t. A predictive value iteration (PVI) approach is developed to solve for the value function with reward predictions. The basic idea is to first find the value function at the last prediction step V_t^p, and then update the value function backwards in time with the reward prediction. The detailed procedure is as follows.

First, the value function at the last prediction V_t^p can be computed using a traditional value iteration approach with the last predicted reward $R_{t,p}$:

$$V_t^p = VI(R_{t,p}, V_t^0)$$

where V_t^0 is the initial guess, VI stands for the traditional value iteration algorithm. The physical meaning behind this step is to assume that the reward stays constant after the last prediction. The last value function for this time step V_t^p can be reused as the initial guess for the next time step, $V_{t+1}^0 = V_t^p$.

Next, given V_t^p, the value function at current time t can be computed using a backward Bellman equation:

$$Q_t^k(s,a) = R_t^k(s') + \gamma V_t^{k+1}(s'),\ k = p-1, \dots, 2, 1$$

$$V_t^k(s) = \max_a Q_t^k(s,a)$$

where s' is the next location arrived by taking an action a and γ is the decaying parameter. Starting from the last prediction step, we iteratively update the value function for $p-1$ steps till the current time. The optimal moves for this time step t can be found using the Q-value function at the current step t, $a^*(s) = \underset{a}{argmax}\, Q_t^1(s,a)$.

Another consideration that can be facilitated by incorporating MPC is to include running cost for the HSN tradeoff analysis. The current cost for mobile sensors is treated as a total constant cost. A more general condition is to consider the energy constraints for each run time of the mobile sensors and take into account a time-varying running cost. This

consideration can be incorporated by penalizing the running cost in the reward function as an additional term. This provides another research direction for future work.

REFERENCES

[1] Mankad J, Natarajan B, Srinivasan B. Integrated approach for optimal sensor placement and state estimation: A case study on water distribution networks. ISA Trans 2021. https://doi.org/10.1016/j.isatra.2021.06.004.

[2] Khoa V Van, Takayama S. Wireless sensor network in landslide monitoring system with remote data management. Measurement (Lond) 2018;118:214–29. https://doi.org/10.1016/j.measurement.2018.01.002.

[3] Wu H, Mei X, Chen X, Li J, Wang J, Mohapatra P. A novel cooperative localization algorithm using enhanced particle filter technique in maritime search and rescue wireless sensor network. ISA Trans 2018;78:39–46. https://doi.org/10.1016/j.isatra.2017.09.013.

[4] Munjani J, Joshi M. A non-conventional lightweight Auto Regressive Neural Network for accurate and energy efficient target tracking in Wireless Sensor Network. ISA Trans 2021;115:12–31. https://doi.org/10.1016/j.isatra.2021.01.021.

[5] Huerta MK, Garcia-Cedeno A, Guillermo JC, Clotet R. Wireless Sensor Networks Applied to Precision Agriculture: A worldwide literature review with emphasis on Latin America. IEEE Geosci Remote Sens Mag 2021;9:209–22. https://doi.org/10.1109/MGRS.2020.3044235.

[6] Hart JK, Martinez K. Environmental Sensor Networks: A revolution in the earth system science? Earth Sci Rev 2006;78:177–91. https://doi.org/10.1016/j.earscirev.2006.05.001.

[7] Nittel S. A Survey of Geosensor Networks: Advances in Dynamic Environmental Monitoring. Sensors 2009;9:5664–78. https://doi.org/10.3390/s90705664.

[8] Inoue M, Tanioka Y, Yamanaka Y. Method for Near-Real Time Estimation of Tsunami Sources Using Ocean Bottom Pressure Sensor Network (S-Net). Geosciences (Basel) 2019;9:310. https://doi.org/10.3390/geosciences9070310.

[9] Barnes Christopher ; Best M; PL; PB. Understanding Earth–: Ocean Processes using Real-time Data from NEPTUNE, Canada's Widely Distributed Sensor Networks, Northeast Pacific. Geoscience Canada 2011;38:21–30.

[10] Duan J, Gao D, Yang D, Foh CH, Chen HH. An energy-aware trust derivation scheme with game theoretic approach in wireless sensor networks for IoT applications. IEEE Internet Things J 2014;1:58–69. https://doi.org/10.1109/JIOT.2014.2314132.

[11] Wu F, Xu L, Kumari S, Li X, Shen J, Choo KKR, et al. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. Journal of Network and Computer Applications 2017;89:72–85. https://doi.org/10.1016/j.jnca.2016.12.008.

[12] Ullo SL, Sinha GR. Advances in smart environment monitoring systems using iot and sensors. Sensors (Switzerland) 2020;20. https://doi.org/10.3390/s20113113.

[13] Jiang X, Lin Z, Li S, Ji Y, Luan Y, Ma S. Dynamical attitude configuration with wearable wireless body sensor networks through beetle antennae search strategy. Measurement (Lond) 2021;167:108128. https://doi.org/10.1016/j.measurement.2020.108128.

[14] Zygowski C, Jaekel A. Optimal path planning strategies for monitoring coverage holes in Wireless Sensor Networks. Ad Hoc Networks 2020;96:101990. https://doi.org/10.1016/j.adhoc.2019.101990.

[15] Lambrou TP. Optimized cooperative dynamic coverage in mixed sensor networks. ACM Trans Sens Netw 2015;11:46. https://doi.org/10.1145/2700260.

[16] Debnath S, Hossain A, Chowdhury SM, Singh AK. Effective sensing radius (ESR) and performance analysis of static and mobile sensor networks. Telecommun Syst 2018;68:115–27. https://doi.org/10.1007/s11235-017-0379-z.

[17] Chakraborty S, Goyal NK, Mahapatra S, Soh S. A Monte-Carlo Markov chain approach for coverage-area reliability of mobile wireless sensor networks with multistate nodes. Reliab Eng Syst Saf 2020;193:106662. https://doi.org/10.1016/j.ress.2019.106662.

[18] Di Francesco M, Das SK, Anastasi G. Data collection in wireless sensor networks with mobile elements: A survey. ACM Trans Sens Netw 2011;8. https://doi.org/10.1145/1993042.1993049.

[19] Keung GY, Li B, Zhang Q. The intrusion detection in mobile sensor network. IEEE/ACM Transactions on Networking 2012;20:1152–61. https://doi.org/10.1109/TNET.2012.2186151.

[20] Yalçın S, Erdem E. BTA-MM: Burst traffic awareness-based adaptive mobility model with mobile sinks for heterogeneous wireless sensor networks. ISA Trans 2021. https://doi.org/10.1016/j.isatra.2021.06.027.

[21] Ariga K, Nishida T, Koyama S, Ueno N, Saruwatari H. Mutual-Information-Based Sensor Placement for Spatial Sound Field Recording. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE; 2020, p. 166–70. https://doi.org/10.1109/ICASSP40776.2020.9053715.

[22] Bhattacharyya P, Beck J. Exploiting convexification for Bayesian optimal sensor placement by maximization of mutual information. Struct Control Health Monit 2020;27. https://doi.org/10.1002/stc.2605.

[23] Adurthi N, Singla P, Majji M. Mutual Information Based Sensor Tasking with Applications to Space Situational Awareness. Journal of Guidance, Control, and Dynamics 2020;43:767–89. https://doi.org/10.2514/1.G004399.

[24] Schmidt K, Smith RC, Hite J, Mattingly J, Azmy Y, Rajan D, et al. Sequential optimal positioning of mobile sensors using mutual information. Statistical Analysis and Data Mining: The ASA Data Science Journal 2019;12:465–78. https://doi.org/10.1002/sam.11431.

[25] Pei X-Y, Yi T-H, Qu C-X, Li H-N. Conditional information entropy based sensor placement method considering separated model error and measurement noise. J Sound Vib 2019;449:389–404. https://doi.org/10.1016/j.jsv.2019.02.035.

[26] Turko N, Lobashev A, Ushakov K, Kaurkin M, Ibrayev R. Information Entropy Initialized Concrete Autoencoder for Optimal Sensor Placement and Reconstruction of Geophysical Fields 2022.

[27] Wang Y, Zhang L, Chen G. Optimal Sensor Placement for Obstacle Detection of Manipulator Based on Relative Entropy. 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE; 2019, p. 702–7. https://doi.org/10.1109/ICIEA.2019.8833986.

[28] Hu Z, Chen W, Chen B, Tan D, Zhang Y, Shen D. Robust Hierarchical Sensor Optimization Placement Method for Leak Detection in Water Distribution System. Water Resources Management 2021;35:3995–4008. https://doi.org/10.1007/s11269-021-02922-3.

[29] Krause A, Singh A, Guestrin C. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. Journal of Machine Learning Research 2008;9:235–84.

[30] Jayaweera HMPC, Hanoun S. UAV Path Planning for Reconnaissance and Look-Ahead Coverage Support for Mobile Ground Vehicles. Sensors 2021;21:4595. https://doi.org/10.3390/s21134595.

[31] Pae D-S, Kim G-H, Kang T-K, Lim M-T. Path Planning Based on Obstacle-Dependent Gaussian Model Predictive Control for Autonomous Driving. Applied Sciences 2021;11:3703. https://doi.org/10.3390/app11083703.

[32] Liu C, Lee S, Varnhagen S, Tseng HE. Path planning for autonomous vehicles using model predictive control. 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE; 2017, p. 174–9. https://doi.org/10.1109/IVS.2017.7995716.

[33] Popescu D, Stoican F, Stamatescu G, Chenaru O, Ichim L. A survey of collaborative UAV-WSN systems for efficient monitoring. Sensors (Switzerland) 2019;19:1–40. https://doi.org/10.3390/s19214690.

[34] Nazlibilek S. Autonomous navigation of robotic units in mobile sensor network. Measurement (Lond) 2012;45:938–49. https://doi.org/10.1016/j.measurement.2012.01.050.

[35] de Freitas EP, Heimfarth T, Vinel A, Wagner FR, Pereira CE, Larsson T. Cooperation among wirelessly connected static and mobile sensor nodes for surveillance applications. Sensors (Switzerland) 2013;13:12903–28. https://doi.org/10.3390/s131012903.

[36] Sun G, Zhou R, Di B, Dong Z, Wang Y. A novel cooperative path planning for multi-robot persistent coverage with obstacles and coverage period constraints. Sensors (Switzerland) 2019;19. https://doi.org/10.3390/s19091994.

[37] Venkatesan L, Shanmugavel S, Subramaniam C. A Survey on Modeling and Enhancing Reliability of Wireless Sensor Network. Wireless Sensor Network 2013;05:41–51. https://doi.org/10.4236/wsn.2013.53006.

[38] Dong M, Ota K, Liu A, Guo M. Joint Optimization of Lifetime and Transport Delay under Reliability Constraint Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 2016;27:225–36. https://doi.org/10.1109/TPDS.2015.2388482.

[39] AboElFotoh HMF, Iyengar SS, Chakrabarty K. Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures. IEEE Trans Reliab 2005;54:145–55. https://doi.org/10.1109/TR.2004.842540.

[40] Dagdeviren O, Akram VK, Tavli B. Design and Evaluation of Algorithms for Energy Efficient and Complete Determination of Critical Nodes for Wireless Sensor Network Reliability. IEEE Trans Reliab 2019;68:280–90. https://doi.org/10.1109/TR.2018.2873917.

[41] Deif D, Gadallah Y. A comprehensive wireless sensor network reliability metric for critical Internet of Things applications. EURASIP J Wirel Commun Netw 2017;2017. https://doi.org/10.1186/s13638-017-0930-3.

[42] Kabadurmus O, Smith AE. Evaluating Reliability/Survivability of Capacitated Wireless Networks. IEEE Trans Reliab 2018;67:26–40. https://doi.org/10.1109/TR.2017.2712667.

[43] Chakraborty S, Goyal NK, Mahapatra S, Soh S. Minimal Path-Based Reliability Model for Wireless Sensor Networks with Multistate Nodes. IEEE Trans Reliab 2020;69:382–400. https://doi.org/10.1109/TR.2019.2954894.

[44] Yue YG, He P. A comprehensive survey on the reliability of mobile wireless sensor networks: Taxonomy, challenges, and future directions. Information Fusion 2018;44:188–204. https://doi.org/10.1016/j.inffus.2018.03.005.

[45] Wu W, Zhang F, Wardi Y. Target localization: Energy-information trade-offs using mobile sensor networks. Proceedings of the IEEE Conference on Decision and Control 2014;2015-Febru:2944–9. https://doi.org/10.1109/CDC.2014.7039842.

[46] Hoffmann GM, Tomlin CJ. Mobile sensor network control using mutual information methods and particle filters. IEEE Trans Automat Contr 2010;55:32–47. https://doi.org/10.1109/TAC.2009.2034206.

[47] Lambrou TP, Panayiotou CG. Collaborative path planning for event search and exploration in mixed sensor networks. International Journal of Robotics Research 2013;32:1424–37. https://doi.org/10.1177/0278364913498909.

[48] Balaji S, Anitha M, Rekha D, Arivudainambi D. Energy efficient target coverage for a wireless sensor network. Measurement (Lond) 2020;165:108167. https://doi.org/10.1016/j.measurement.2020.108167.

[49] Njoya AN, Ari AAA, Nana Awa M, Titouna C, Labraoui N, Effa JY, et al. Hybrid Wireless Sensors Deployment Scheme with Connectivity and Coverage Maintaining

in Wireless Sensor Networks. Wirel Pers Commun 2020;112:1893–917. https://doi.org/10.1007/s11277-020-07132-5.

[50] Liu B, Dousse O, Nain P, Towsley D. Dynamic coverage of mobile sensor networks. IEEE Transactions on Parallel and Distributed Systems 2013;24:301–11. https://doi.org/10.1109/TPDS.2012.141.

[51] Akyildiz IF, Can Vuran M. Wireless Sensor Networks. Hoboken, New Jersey: John Wiley & Sons, Ltd; 2010. https://doi.org/10.1002/9780470515181.

[52] Baralis E, Cerquitelli T. Selecting representatives in a sensor network. SEBD 2006 - Proceedings of the 14th Italian Symposium on Advanced Databases Systems 2006:351–60.

[53] Gomberg J, Schweig E. Earthquake Hazard in the Heart of the Homeland. Eeriorg 2007;1974:4.

[54] Boubrima A, Knightly EW. Robust mission planning of UAV networks for environmental sensing. Proceedings of the 6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, DroNet 2020 2020. https://doi.org/10.1145/3396864.3399698.

[55] Huang H, Savkin A V., Ding M, Huang C. Mobile robots in wireless sensor networks: A survey on tasks. Computer Networks 2019;148:1–19. https://doi.org/10.1016/j.comnet.2018.10.018.

[56] Abdollahzadeh S, Navimipour NJ. Deployment strategies in the wireless sensor network: A comprehensive review. Comput Commun 2016;91–92:1–16. https://doi.org/10.1016/j.comcom.2016.06.003.

[57] Katanforoush A, Shahshahani M. Distributing points on the sphere, I. Exp Math 2003;12:199–209. https://doi.org/10.1080/10586458.2003.10504492.

[58] Dhillon SS, Chakrabarty K. Sensor placement for effective coverage and surveillance in distributed sensor networks. IEEE Wireless Communications and Networking Conference, WCNC 2003;3:1609–14. https://doi.org/10.1109/WCNC.2003.1200627.

[59] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 1968;4:100–7. https://doi.org/10.1109/TSSC.1968.300136.

[60] Aghili F, Su CY. Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU. IEEE/ASME Transactions on Mechatronics 2016;21:2015–26. https://doi.org/10.1109/TMECH.2016.2547905.

[61] Moore T, Stouch D. A generalized extended Kalman filter implementation for the robot operating system. In: Menegatti E, Michael N, Berns K, Yamaguchi H, editors. Advances in Intelligent Systems and Computing, vol. 302, Cham: Springer International Publishing; 2016, p. 335–48. https://doi.org/10.1007/978-3-319-08338-4_25.

[62] Lacey A, Thacker N. Tutorial: The Kalman filter. 1998.

[63] Bellman R. A Markovian Decision Process. Indiana University Mathematics Journal 1957;6:679–84. https://doi.org/10.1512/iumj.1957.6.56038.

[64] Wang R, Veloso M, Seshan S. Active sensing data collection with autonomous mobile robots. Proc IEEE Int Conf Robot Autom, vol. 2016- June, Stockholm, Sweden: IEEE; 2016, p. 2583–8. https://doi.org/10.1109/ICRA.2016.7487415.

[65] Bernstein DS, Givan R, Immerman N, Zilberstein S. The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research 2002;27:819–40. https://doi.org/10.1287/moor.27.4.819.297.

[66] Bottou L. On-line Learning and Stochastic Approximations. On-Line Learning in Neural Networks 2010;17:9–42. https://doi.org/10.1017/cbo9780511569920.003.

[67] Kingma DP, Ba JL. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings 2015.

[68] Nelder JA, Mead R. A Simplex Method for Function Minimization. Comput J 1965;7:308–13. https://doi.org/10.1093/comjnl/7.4.308.

[69] MATLAB. fminsearch Algorithm n.d. https://www.mathworks.com/help/matlab/math/optimizing-nonlinear-functions.html#bsgpq6p-11 (accessed March 25, 2021).

[70] Wendt JF, Anderson JD, Degroote J, Degrez G, Dick E, Grundmann R, et al. Computational fluid dynamics: An introduction. 2009. https://doi.org/10.1007/978-3-540-85056-4.

[71] McGrattan K, Hostikka S, McDermott R, Floyd J, Vanella M. Fire Dynamics Simulator User's Guide. NIST Special Publication 1019 Sixth Edition 2019:347.

[72] Hamins A, Bundy M, Dillon SE. Characterization of candle flames. Journal of Fire Protection Engineering 2005;15:265–85. https://doi.org/10.1177/1042391505053163.

[73] Kim HJ, Lilley DG. Heat release rates of burning items in fires. 38th Aerospace Sciences Meeting and Exhibit 2000. https://doi.org/10.2514/6.2000-722.

[74] Bualat MG, Smith T, Smith EE, Fong T, Wheeler D. Astrobee: A New Tool for ISS Operations. 2018 SpaceOps Conference, Reston, Virginia: American Institute of Aeronautics and Astronautics; 2018. https://doi.org/10.2514/6.2018-2517.

[75] Singh SK, Kumar P. A comprehensive survey on trajectory schemes for data collection using mobile elements in WSNs. J Ambient Intell Humaniz Comput 2020;11:291–312. https://doi.org/10.1007/s12652-019-01268-4.

[76] Guo Y, Xu Z, Saleh J. Collaborative Allocation and Optimization of Path Planning for Static and Mobile Sensors in Hybrid Sensor Networks for Environment Monitoring and Anomaly Search. Sensors 2021;21:7867. https://doi.org/10.3390/s21237867.

[77] Georgescu MR, Meslem A, Nastase I. Accumulation and spatial distribution of CO2 in the astronaut's crew quarters on the International Space Station. Build Environ 2020;185:107278. https://doi.org/10.1016/j.buildenv.2020.107278.