Active Environmental Monitoring and Anomaly Search System for Space Habitat With Markov Decision Process and Active Sensing

YANJIE GUO, ZHAOYI XU, AND JOSEPH HOMER SALEH, (Senior Member, IEEE)
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
Corresponding author: Yanjie Guo (guoyanjie@gatech.edu)

This work was supported by the Space Technology Research Institute through the National Aeronautics and Space Administration’s (NASA’s) Space Technology Research Grants Program.

ABSTRACT For future crewed missions that could last years with limited ground support, the environmental control and life support system (ECLSS) will likely evolve to meet new, more stringent reliability and autonomy requirements. In this work, we focus on improving the performance of the environmental monitoring and anomaly detection systems using Markov decision process and active sensing. We exploit actively moving sensors to develop a novel sensing architecture and supporting analytics, termed Active environmental Monitoring and Anomaly Search System (AMASS). We design a Dynamic Value Iteration policy to solve the path planning problem for the moving sensors in a dynamic environment. To test and validate AMASS, we developed a series of computational experiments for fire search, and we assessed the performance against three metrics: (1) anomaly detection time lag, (2) source location uncertainty, and (3) state estimation error. The results demonstrate that: AMASS provides $10 \sim 15$ times better performance than the traditional fixed sensor monitoring and detection strategy; ventilation in the monitored environment affects the performance by $6 \sim 40$ times for any monitoring architecture with fixed or moving sensors; the monitoring performance cannot be fully reflected in a monolithic single metric, but should include different metrics for the timeliness and spatial resolution of the detection function.

INDEX TERMS Anomaly detection, environmental monitoring, Markov decision process, space habitat.

I. INTRODUCTION

The Environmental Control and Life Support System (ECLSS) is a core element for human space exploration missions. It consists of subsystems that provide and control necessary elements for human survival, including atmosphere monitoring and revitalization, fire detection and suppression, water recycle and recovery, and waste management. The ECLSS has significantly evolved since its inception as mission duration has increased, from non-regenerative systems in the 1960s, to the distributed-controlled regenerative system currently onboard the International Space Station (ISS) [1]–[3]. For future missions that could last years with limited ground support, the ECLSS will likely further evolve to meet new, more stringent reliability and autonomy requirements [4]. Due to the life-threatening consequences that potential anomalies (e.g., leaks, fires) may cause in a future deep space habitat [5], it is essential to have a smart environmental monitoring system for the ECLSS that can search and detect anomalies autonomously and in a timely manner [6]. This work addresses in part these issues by developing an active monitoring and anomaly search system using Markov decision process (MDP) and moving sensors.

Environmental monitoring systems typically consist of sensors installed at fixed locations for the duration of their deployment or service life of the system within which they are embedded. While this fixed sensor (FS) strategy is easy to deploy, it has some drawbacks [7]. For example, FS lacks fine-grid measurements at arbitrary locations, which can result in inadequate coverage of the whole space and large uncertainty about the anomaly source if the sensor density is limited. Furthermore, the feasible locations to attach sensors are usually restricted. For instance, most sensors need to be attached on walls because of fixation and connection requirement. It is challenging to take measurements in an open area without a foothold. These issues can
lead to significant state uncertainty and large detection time lag should anomalies develop. In short, FS strategy for environmental monitoring may limit the observability and accuracy of state estimation within the monitored environment.

In this work, we focus on improving the performance of the environmental monitoring and anomaly detection systems to address more stringent requirements for future space exploration missions. Our initial motivation for undertaking this work was to explore the possibility of leveraging advances in robotics and reinforcement learning to devise a novel sensing architecture that can outperform the traditional environmental monitoring provided by the ECLSS with fixed sensors.

As studies in robotics have significantly advanced over the past decades, it is now practical to use sensors mounted on moving platforms to explore an unknown environment or to monitor a known one. Some examples include Spheres and Astrobee currently operating on the International Space Station (ISS) [8], [9]. We propose in this work to exploit actively moving sensors (MS) to monitor the environment within a space habitat or any enclosed environment, and to search for anomalies. More specifically, we develop a novel framework and supporting analytics, termed Active environmental Monitoring and Anomaly Search System (AMASS), to improve the detection and monitoring performance in a habitat. We utilize Markov decision process (MDP)-based active sensing techniques to devise an efficient policy for autonomous anomaly search and detection. Our objectives are threefold: (1) to reduce the detection time lag of potential anomalies, (2) to locate the source of the anomaly, and (3) to improve the overall state estimation accuracy of the monitored environment.

While AMASS is developed for general environmental monitoring and anomaly search purposes, in order to expand on the proposed framework and demonstrate its performance, we restrict the present work to temperature monitoring and fire search in a micro-gravity 3D environment as the main application. We develop a series of computational experiments to validate our framework and assess its performance against three metrics: (1) the anomaly detection time lag, (2) the source location uncertainty, and (3) the state estimation error. We first investigate the performance of AMASS with several policies, including our customized Dynamic Value Iteration (DVI) policy for the moving sensors. We then leverage the feasible best-in-class policy to benchmark the detection performance of our moving sensors against the traditional fixed sensor strategy. We identify and examine the effects of critical parameters on the performance of both approaches (FS and AMASS), including varying sensor density and ventilation speed in the monitored environment; both will be shown to have significant impact on the monitoring and anomaly detection performance.

The reminder of the article is organized as follows. A brief literature review and our scheme on active sensing and reinforcement learning is discussed in Section II. The detail of the AMASS architecture is provided in Section III. The computational experiment setup is introduced in Section IV. The results and the comparative analysis are presented and examined in Section V. Finally, Section VI concludes this work.

II. ACTIVE SENSING AND REINFORCEMENT LEARNING: A BRIEF LITERATURE REVIEW AND OUR SCHEME

Sensors are often viewed as passive in that they only receive signals passively without reacting or modifying their “posture” based on the collected data and given their sensing objectives. This view is limited considering the fact that research on biological systems indicates that most sensory processes in nature are active, with specific goals and dynamic sensor manipulation [10]. A popular application of active sensing in robotics is the use of mobile sensors. Some examples of a single moving sensor include autonomous observation on the International Space Station [11], and learning dynamic spatiotemporal fields [12]. Multiple moving sensors can achieve more complex tasks, such as target tracking [13], map exploration [14], environmental monitoring [7], and fire searches [15]. Laport-López et al. [16] consider multi-agent systems (MAS) as an ideal solution to create large-scale, multi-device, and multi-purpose mobile sensing systems to obtain information from heterogeneous devices, open sources, and social networks.

One critical problem for mobile sensor networks is motion planning, i.e., how to make decisions for the next actions to achieve their sensing objectives [17]. A smart policy can significantly improve the sensing performance, while a poor one may lead to worse results than with (sparse) fixed sensors. For simple applications, some prior works used variant of the traveling salesman problem (TSP) [7], A∗ [15], and linear regression [13] to devise a motion policy and determine how to move the sensors within the environment to achieve their objectives. Using reinforcement learning (RL) techniques is another popular approach to solve more complex problems. The active sensing problem can be formulated as a variant of partially observable Markov decision process (POMDP), and the multi-agent policy can be solved using decentralized approaches [18]–[20]. With recent improvement in computational power, some works also use deep reinforcement learning techniques to tackle the multi-robot collision avoidance problem for navigation in complex scenarios, such as working in a dense human crowd and fighting forest fire [21], [22]. Although reinforcement learning (RL) is a powerful tool to solve complex robotic problems, it requires a delicately designed reward function to correctly reflect the desired goal of the tasks and to guarantee a successful deployment of the algorithms. An inappropriate reward function may result in degraded performance or even unintended and harmful behavior [23]. In practice, it can be surprisingly difficult to define a good reward function [24], especially for complex problems such as our application. There are two approaches to solve this reward function design problem. The first approach is inverse reinforcement learning (IRL), where the algorithm learns a suitable reward function from expert demonstration [25]. If the expert’s knowledge is not
available for a novel application like ours, we can leverage the second approach to parameterize the reward function and tune it according to the performance.

Our work leverages active sensing for anomaly search and detection in a dynamic environment using a multi-agent system. Specifically, the task is to monitor the temperature in a space habitat and search for potential fires based on the measurements using multiple moving temperature sensors. The task has three objectives (1) to reduce the detection time lag, (2) to locate the source of the anomaly, and (3) to improve the overall state estimation accuracy of the monitored environment.

There are significant differences between our application and the aforementioned works. First, we use a dynamic environment with evolving states (temperature), unlike most map exploration or goal searching problems which assume a static environment with binary states indicating if a place has been visited or not. Second, we combine the monitoring and anomaly search process, rather than only state monitoring which focuses more on exploration, or only anomaly search which focuses more on exploiting the collected information. Third, for the detection process, we do not directly have access to the information of whether we have an anomaly or not. Instead, we need to infer it from the measurements (here temperature). These differences add more complexity to the problem but also make it more realistic.

These differences between our application and prior art require that we devise a different approach to fulfill the tasks. The prior works formulated the active sensing problem as variants of partially observable Markov decision process (POMDP), where the sensing agents can only make observations of the states and have to derive belief of the true states. However, in practice, it is often computationally intractable to solve a POMDP exactly despite the well-defined mathematical model [26]. Thus, instead of formulating our problem as a POMDP, we divide it into two smaller (and easier to solve) subproblems: an estimation process and a Markov decision process (MDP). In the estimation process, we focus on extracting useful information from the raw measurements to improve our understanding of the whole environment. In the MDP, we focus on path planning for the moving sensors based on the extracted information. The two subproblems depend on each other while maintaining minimum overlap of their objectives. By dividing the problem into two subproblems, we reduce the complexity of the original task, and we acquire explainable intermediate information that may be buried in a POMDP problem for humans to understand. The detailed discussion of our framework is provided in the following section.

III. THE ACTIVE ENVIRONMENTAL MONITORING AND ANOMALY SEARCH SYSTEM (AMASS): ARCHITECTURE AND ANALYTICS

In this section, we first provide a high-level overview of the AMASS architecture. We then discuss its technical details, including the moving sensors, the state analyzer, and the policymaker.

A. OVERVIEW OF THE AMASS ARCHITECTURE

The following discussion is a cursory overview of the temperature monitoring and fire search process in AMASS. The technical details are provided in the next subsections.

The AMASS framework incorporates three main elements: moving sensors, a state analyzer for measurement analysis, and a policymaker for path planning, as shown in Fig. 1. Each element addresses a particular subproblem discussed earlier. At each time step, the moving sensors take raw measurements while patrolling the environment/habitat. The analyzer then processes these measurements and extracts meaningful information, including full state estimation of the habitat, estimation uncertainty level, and the risk or likelihood of anomaly at particular locations. This information represents a high-level understanding of the environment and can be shared with external systems such as the ground station, a higher-level supervisory system, or the astronauts onboard if they are present. In AMASS, this information is used for anomaly detection and decision-making for path planning of the sensors. If the state estimation or likelihood of anomaly breaches a predefined threshold, an alarm will be triggered. Otherwise, the MDP-based policymaker determines the next optimal moves for the sensors. How this next move is determined is a critical element of RL in general and AMASS specifically. The policymaker solves this decision-making problem by a Dynamic Value Iteration method (DVI) we developed at each time step, following which either the alarm is triggered or the next optimal moves for the sensors is determined. The technical details of each element in AMASS are discussed next.

B. MOVING SENSORS

As a core element in AMASS, the moving sensors take measurements of the environment while patrolling the habitat. The notion of a moving sensor is essentially a type...
of robot or mobile platform with an emphasis on sensing and observation, as opposed to action and intervention. It consists of two parts: a suite of sensors and a moving mechanism.

The choice of sensors depends on the requirements and tasks to be performed. What is sensed should reflect what is searched for. In this work, we consider fire search and detection application to illustrate one particular use of AMASS. For fire detection, different sensors can be used, for example smoke (soot), CO, or temperature sensors. Note that these are “pointwise” sensors that take measurements at a point or within a small region. More advanced sensors include 1D/2D/3D Lidar, 2D acoustic (noise), or 2D infrared sensors. These types of sensors take higher dimensional measurements along their field of view. They can be used to detect other anomalies such as leaks [27]. For simplicity, the sensor we choose for our computational experiments is a pointwise temperature sensor with alarm threshold at 47°C [28].

The purpose of the moving mechanism is to accommodate and mobilize the sensors for a wider monitoring range and a more flexible anomaly detection. The moving mechanism is not the focus of this work. Some existing platforms that can be used as the potential moving mechanism include Spheres and Astrobot currently onboard the International Space Station [8], [9]. The moving directions are along the six coordinates, up, down, left, right, forward, and backward. At each time step, the moving platform can move to the six neighboring blocks or stay in position.

C. 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. 2. 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.

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 [29], [30]. 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 [31].

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:

$$\dot{x}_t = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + v_t$$

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:

$$E(v) = E(w) = 0, \quad R = E( vv^T ), \quad Q = E( ww^T )$$

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. 3-7, which can be divided into two parts, the prior prediction, and the correction update. The prediction process propagates the current estimation to the next time step, as shown in Eq. 3 and Eq. 4:

$$\hat{x}_{t+1}' = A\hat{x}_t + Bu_t$$
$$P_t' = AP_tA^T + Q$$

where the $\hat{x}'$ and $P'$ denote prior measurement predictions of the state variable and error covariance, respectively.

After taking the next measurement, the correction process updates the prior prediction, as shown in Eq. 5 and Eq. 6:

$$\hat{x}_t = \hat{x}_t' + K_t ( y_t − C\hat{x}_t' )$$
$$P_t = ( I − K_tC ) P_t' ( I − K_tC)^T + K_tRK_t^T$$

where $K_t$, the Kalman gain, is given by Eq. 7:

$$K_t = P_{t-1}'C^T ( CP_{t-1}'C^T + R )^{-1}$$

![FIGURE 2. The analyzer flowchart within the AMASS 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.](image-url)
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 Eq. (8):

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

(8)

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 in subsection III.D, will leverage this information to determine efficient next moves for the sensors.

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 $\Phi_t$ using Eq. (9):

$$\Phi_t = 1 - (1 - \phi_A)^{t-t_s}$$  

(9)

where $t$ and $t_s$ are the current time and the most recent time of visit at location $x$, and $\phi_A$ 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 $t - t_s$). 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 $\Phi_t$ 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 AMASS, this information is passed to the policymaker for path planning of the moving sensors, as discussed next.

D. 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. 3. 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.

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 $\Phi_t$. 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 $\Phi_t$ 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 Eq. 10:

$$ R_t = E_t + \alpha_t U_t + \beta_t \Phi_t $$

(10)

The two parameters $\alpha_t$ and $\beta_t$ 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 [32], or by solving a variant of the travelling salesman problem [7]. We discuss in the next subsection why these methods are not suitable for our application and how we propose to tackle this problem.

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 Eq. 11:

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

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

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

(11)

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 action $a$, $R$ is the reward function, and $\gamma$ 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. 11 can be simplified to Eq. 12:

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

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

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

(12)

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. 10 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 Eq. 13:

$$ V_{t+1}^0 = V_t $$

(13)

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. 10 and the DVI method to solve it, we have developed a policy for a single moving sensor. We discuss three benchmarked policies in the following subsection for comparative purpose.

3) BENCHMARKED POLICIES

To benchmark the performance of our dynamic value iteration (DVI) policy, we develop additional policies for comparative analysis. In the simulations discussed in sections IV and V, we examine the performance of the moving sensors when operating under each one of the four policies considered in this work. The feasible best-in-class practical policy will be chosen for AMASS to conduct further analysis and comparisons with the fixed sensors.

First, we consider the random walk moving policy ($P_1$), which randomly chooses the next move (or stay) for the sensor. This is evidently not a good policy since it does not leverage any of the measurements and state estimation the sensor provides. It is, however, an important policy to
examine since it provides us with the “lower bound” or baseline performance that any other policy should outperform or be rejected. This random walk policy also illustrates the importance of having an appropriate, carefully designed policy for the moving sensors.

Second, we consider the local greedy search policy (P2), which has a narrow view of the habitat and simply chooses to move to the neighboring location with the highest reward. This policy borrows the idea of the hill climbing algorithm in optimization. It is easy and fast to deploy, which is an important consideration for real-time decision making. The local greedy search policy is a reasonable and straightforward heuristic approach even though it may get trapped in a local reward maximum.

Lastly, we consider the jump policy (P3), which directly jumps to the location with the global highest reward. This policy is evidently not realistic since it is asymptotically equivalent to a speed so large that the sensor jumps/teleports to the best location. It is, however, important to consider since it serves as the “upper bound” or (unrealizable) best-in-class performance of any conceivable policy. How close will the DVI or local greedy search policy be to this best-in-class policy performance is examined in section V.

These three policies offer us clear landmarks in both directions for the moving sensor on the performance spectrum. P1 provides the worst baseline performance, P2 is a reasonable heuristic approach, and P3 serves as the unrealizable upper bound for any conceivable policy. We use them to benchmark the performance of our DVI policy and determine its location on this performance spectrum.

So far, we have developed a DVI policy and three benchmarked policies aiming at the path planning of a single sensor. We expand this decision-making process for a multi-agent system next.

4) 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 [33]. 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 Eq. 14:

\[
\text{ker}_i(s; s_j, \sigma) = 1 - \exp \left( -\frac{(s - s_j)^2}{2\sigma^2} \right) \quad (14)
\]

where \(s_j\) is the location where agent \(i\) is, \(\sigma\) is a parameter controlling the kernel size. We set \(\sigma\) proportional to the average agent distance with a factor \(p, \sigma = p \cdot \text{sensor density}^{-1/3} = p \left( \frac{\# \text{ of agents}}{\text{total volume}} \right)^{-1/3} \).

The adjusted reward function for agent \(i\) can be derived by applying this kernel to all the other agents, as shown in Eq. 15:

\[
R_{t,j}(s) = R_t(s) \cdot \min_{j \neq i} \text{ker}_j(s; s_j, \sigma) \quad (15)
\]

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. 10. To improve the monitoring and detection performance of AMASS, we conduct an Inverse Reward Shaping (IRS) process to tune the reward function parameters, as discussed in the next subsection.

5) 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 of AMASS. 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 \((\alpha, \beta)\) to minimize the average detection time lag of \(n\) random computational experiments, as formulated in Eq. 16:

\[
\min_{\alpha, \beta} \frac{1}{n} \sum_{i=1}^{n} \text{lag}_i(\alpha, \beta) \quad (16)
\]

where \(\text{lag}_i(\alpha, \beta)\) is the detection time lag of the \(i\)-th experiment, with the reward function parameter \((\alpha, \beta)\).

This problem can be solved by any appropriate optimization methods, such as ADAM, stochastic gradient descent, or Nelder-Mead simplex algorithm [34]–[36]. We choose the Nelder-Mead simplex algorithm as provided in MATLAB fminsearch function [37].

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

IV. COMPUTATIONAL EXPERIMENTS

We devise a series of computational experiments of fire search in a 3D environment (details in IV.A) for two purposes: first to assess the performance of AMASS with the different search policies discussed previously; and second to use the best-in-class feasible policy to benchmark the performance of moving sensors with AMASS against that of the traditional fixed sensor (FS) strategy for environmental monitoring and fire detection. We focus on three performance metrics: detection time lag, source localization uncertainty, and state estimation error (details in IV.B). We investigate the
performance of both AMASS and FS strategies with respect to two critical factors, sensor density and ventilation effect in the habitat (details in IV.III).

A. SIMULATION ENVIRONMENT

The purpose of building the simulation environment is to obtain the simulated temperature map of a fire event inside a space habitat with ventilation under micro-gravity conditions. The simulator consists of two core components: a habitat model, and a heat propagation model (including ventilation).

For simplicity, we use a 5-meter cubic module with 10% random obstacles inside, as shown in Fig. 4a. We select some locations on the boundary (walls) as the air inlet/outlet to model the ventilation effect.

The heat propagation process is a computational fluid dynamic problem, which is solved with the Navier-Stokes (NS) equations. We assume an incompressible viscous flow because of the low flow velocity simulated. With this assumption, the full NS equations can be simplified to the forms shown in Eq. 17-19 [38]:

\[
\nabla \cdot \vec{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{17}
\]

\[
\frac{D\vec{V}}{Dt} = \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla) \vec{V} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{V} \tag{18}
\]

\[
\frac{DT}{Dt} = \frac{\partial T}{\partial t} + \vec{V} \cdot \nabla T = \frac{Q}{\rho c_v} + \alpha \nabla^2 T \tag{19}
\]

where \(x, y, z\) are the coordinates; \(\vec{V} = [u, v, w]\) is the flow velocity; \(\rho, p, T\) are the density, pressure, and temperature of the flow; \(\nu, \alpha\) are the kinematic viscosity and thermal diffusivity of the flow; \(Q\) is the external volumetric heat rate; and \(c_v\) is the constant volume specific heat constant. The three equations are also known as the continuity, momentum, and energy equations.

We solve this system by decoupling the continuity and momentum equations from the energy equation. Because the dominant driving force of the flow is ventilation, we can neglect the influence of temperature on the flow velocity. We first calculate the steady state flow velocity \(\vec{V}\) using the continuity and momentum equations, Eq. 17-18. We set the boundary conditions at the air inlet/outlet to be constant flow velocity. For the other (wall and obstacle) surfaces, the boundary condition is the absence of a perpendicular velocity. Finally, we keep the flow velocity \(\vec{V}\) constant and substitute it in the energy equation, Eq. 19, to solve for the temperature, with an adiabatic boundary condition.

In the computational experiments, we take air properties at 300\(K\)(27\(°\)C). The heat release rate (HRR) of the fire source is chosen to be relatively small, as that of several candles [39], [40]. The ventilation speed is varied and will be shown shortly to be a critical parameter affecting the performance of both AMASS and FS. The complete simulation parameters are provided in Appendix A. An example of the solved flow velocity and temperature map is shown in Fig. 4b.

B. PERFORMANCE METRICS

The three performance metrics considered in this work are listed in Table 1. They are chosen to reflect different aspects of the performance of an environmental monitoring and anomaly detection system.

First, detection time lag is a commonly used performance metric [41]. It is defined as the time it takes for the sensors to trigger an alarm after the start of the fire. This metric represents the temporal sensitivity of the anomaly detection system.

Second, the source localization uncertainty performance metric is defined as the distance between the sensor that triggers the alarm and the anomaly (fire) source. This metric indicates the spatial uncertainty of the anomaly source when
the alarm is triggered. The use of this metric is motivated by the fact that, even when an anomaly is detected, the exact location of its source is often unknown. For example, a low or decreasing atmosphere pressure indicates the presence of a leak in the habitat. But further investigation is required to pinpoint the exact location of the leak. As a timely example, a leak was recently detected onboard the International Space Station (2020), but it took several months after its detection to localize its source [42]. The importance of the source localization uncertainty performance metric is related to the fact that, if this spatial uncertainty at detection time is reduced, the subsequent search and intervention for dealing with the anomaly will be made easier and faster.

Third, the state estimation error performance metric is defined as the median value of the temperature difference between the full state estimation of the entire habitat and the true state value. This metric reflects an overall monitoring performance and the understanding of the whole space. A more ambitious (and computationally challenging) metric would consider the entire distribution of the state estimation error. This is left as a fruitful venue for future work.

We restrict ourselves hereafter to the median error because the estimation error can be significant in a small area around the fire source location. Both the mean and the maximum estimation error will be affected by the deviance in this small region, whereas the median is more robust and makes for a less biased metric.

**C. CRITICAL PARAMETERS IN OUR COMPUTATIONAL EXPERIMENTS**

The parameters used in our computational experiments are provided in Table 2.

To examine the effect of two critical parameters on the performance of the monitoring systems, we conduct a series of computational experiments with varying sensor density (in both AMASS and FS) and ventilation speed. The sensor density is a critical parameter. Having higher sensor density means a better coverage of the whole monitored environment. We vary it between 1 ~ 80 per 125m³ for fixed sensors, and 1 ~ 20 per 125m³ for moving sensors. The ventilation speed is another important parameter which affects heat propagation. Stronger ventilation accelerates the transport of heat, particles, and air constituents. We vary the maximum ventilation speed within a reasonable range of 5 ~ 20 cm/s [43]. Other parameters that can influence the monitoring or anomaly search process include the moving sensor speed, the alarm threshold, and the sensor sampling rate. They are held constant in this work for simplicity and in order to emphasize the use and performance of AMASS. The moving sensor speed is 33 cm/s, which is roughly similar to that of the AstroBee currently in operation on the International Space Station (50 cm/s) [44]. The temperature sensor we choose has an alarm threshold at 47 °C and a sampling frequency of 5 Hz.

The computational experiments we conduct consists of 100 simulations for every scenario (for a given sensor density and ventilation speed) to derive the average performance metrics in Table 1. In each simulation, the fire starts at the beginning of the run at a new random location. Although we set a uniform fire probability in the habitat, this can be easily amended in practice to include a fire risk map should one be available with different likelihoods of fire at different locations.

**V. RESULTS AND DISCUSSION**

In this section, we first present the performance results of the different moving policies discussed previously. We then select the feasible best-in-class policy to incorporate in AMASS for the subsequent computational experiments. These are devised for the comparative performance analysis of both fixed sensors and moving sensors architecture for monitoring and anomaly detection within the habitat. The details are provided next.

**A. COMPARATIVE ANALYSIS OF DETECTION PERFORMANCE WITH DIFFERENT MOVING POLICIES**

As noted previously, we run 100 simulations with a fire occurring at a new random location for each run. Four separate 100 simulation runs are carried out for each of the four moving policies within AMASS, namely the random walk, the local greedy search, our dynamic value iteration (DVI) and the jump policy. These first sets of simulations are carried out without ventilation. The results of the detection time lag are provided in Fig. 5, the other performance metric statistics display similar trends. We conducted these experiments with multiple moving sensors as well, and the results were identical albeit more visually cluttered than those in Fig. 5. We chose to include in this subsection the results for the single
moving sensor for clarity purposes, and the multiple sensors with ventilation effect in the next subsection.

Recall that the lower-bound performance is provided by the random walk policy and the unrealistic upper-bound performance by the jump policy. We observe that the local greedy search has a shorter detection time lag with smaller variability than the random walk policy. Our DVI policy is more robust to the randomness of initial fire location than the local greedy search (smaller spread, no outliers), with a shorter median detection time and smaller interquartile range. The performance of our DVI approaches that of the jump policy, which recall serves as the supremum performance.

DVI is better than the local policy mainly because it makes moving decision based on the global estimation information as discussed in subsection III.C. DVI can direct the sensors away from a local reward to check for further locations where the reward is larger, which benefits the overall performance.

Since the DVI is the best practical policy among the benchmarked ones, we adopt it as the moving policy in AMASS for the subsequent experiments.

B. DETECTION PERFORMANCE WITH MULTIPLE MOVING SENSORS: EFFECT OF SENSOR DENSITY

In the next series of experiments, we study the effect of varying sensor density (FS and AMASS) on the detection performance. It is evident that increasing sensor density will reduce detection time and source localization uncertainty. This intuition, however, is not sufficient for design purposes, and it is important to quantify the extent to which this increase in sensor density (marginally) improves the monitoring and detection performance. We conduct first a series of experiments with multiple sensors and no ventilation. The ventilation effects are examined in the next subsection.

The results of this performance comparison between fixed sensors (FS) and moving sensors in AMASS with different sensor density are provided in Fig. 6. The three plots display the sensor density required to achieve some performance level. Consider, for example, Fig. 6a in which the x axis is the detection time lag, and the y axis the required sensor density: if mission requirements mandate the fire detection time lag to be less than 50 minutes, it can be directly determined from the figure that the needed density for fixed sensors (FS, the red curve in Fig. 6a) should be larger than 7 per 125 m$^3$. The same reading grid applies to the other panels in Fig. 6. These plots can be particularly useful from a design and mission requirements perspectives.

Since the trends in the results are similar for the three metrics, we discuss only the detection time lag next. The most salient results are the following:

1) When comparing FS with AMASS, the results in Fig. 6a indicate that a significantly smaller moving sensor (MS) density is required to achieve the same level of detection time with fixed sensors, as viewed along a vertical slice in Fig. 6a. Alternatively, at iso-sensor density, i.e., when viewed along a horizontal slice in Fig. 6a, the moving sensor strategy outperforms the fixed sensor monitoring strategy with 10~15 times smaller detection time lag. In short, AMASS robustly outperforms the FS monitoring strategy under the no ventilation condition in the space habitat, and across all performance metrics.

2) When considering the FS strategy alone, the results in Fig. 6a clearly indicate a decreasing marginal benefit (in terms of detection time lag) from having a higher sensor density. A proverbial knee in the (performance) curve exists, and it occurs roughly around 10 per 125 m$^3$. The incremental advantage of having higher sensor density decreases past this point. For example, the benefit of an increase in density from 1 to 10 fixed sensors per 125 m$^3$ is enormous; in contrast the benefit of an increase in density from 20 to 30 per 125 m$^3$ is close to insignificant. This asymptotic behavior displayed in Fig. 6a reflects the approaching point of saturation of sensor coverage in the habitat.

3) The same observations in 2) apply to the moving sensors as well.

A cost-benefit analysis is worth undertaking for both monitoring strategies (FS and AMASS) to delineate the entirety of the trade-space, not just detection performance, and to identify whether tipping points in favor of one or the other or Pareto-optimal architecture exist. This is an important topic, but it is beyond the scope of the present work and is left as a fruitful venue for future work.

The advantages of AMASS displayed in Fig. 6 are in part due to the no ventilation condition examined here. Heat transfer in micro-gravity with no ventilation is slow, and this confers a significant advantage to moving sensors in environmental monitoring and searching for anomalies (fire). How this situation changes with ventilation effect is examined in the next subsection.

C. DETECTION PERFORMANCE WITH MULTIPLE MOVING SENSORS: EFFECT OF VENTILATION SPEED

In this subsection, we vary the ventilation speed from 5 to 10 and 20 cm/s, and we examine the detection performance...
of both monitoring strategies (AMASS and FS) at different sensor densities. The results are provided in Fig. 7.

Fig. 7 displays several important results. In this ventilation case, it is no longer appropriate to restrict the discussion to a single performance metric, as we did previously, since clearly the performance metrics are affected differently by ventilation (compare, for example, Fig. 7a and 7b, detection time lag and source localization uncertainty respectively). The most salient results are the following:

1) The most important result in Fig. 7 is that ventilation significantly affects the detection performance of a monitoring system by 6~40 times, whether involving fixed or moving sensors. For illustration, consider the detection time for fixed sensor density of 20 per 125 m³ without ventilation (Fig. 6) and with a 5 cm/s ventilation (Fig. 7): the detection time drops from roughly 20 minutes to about 1 minute. The same effect occurs with moving sensors albeit to a less dramatic extent.

2) One related result displayed in Fig. 7 is the dose-response like effect of ventilation on detection time lag: the faster the ventilation, the shorter the detection time lag. The performance improvement is more substantial with fixed sensors than with moving sensors.

3) One interesting result that is not as clearly visible as the previous ones in Fig. 7 is the fact that sensor density saturation—when the marginal reduction in detection time lag with increasing sensor density is no longer meaningful—decreases with increasing ventilation speed. For example, with a 5 m/s ventilation, there is no further improvement in detection time lag past 40 fixed sensors per 125 m³. With a 20 m/s ventilation, no meaningful reduction occurs past 20 fixed sensors per 125 m³. This result can be explained by the fact that the ventilation speeds up the heat transfer and leads to faster detection for both strategies. Ventilation reduces the demand for more dense sensors. We infer from this observation that higher ventilation speed makes a reasonable moving sensor relatively less mobile with respect to the environment (air) it is monitoring. As a result, the performance advantage of moving sensors over fixed sensors in terms of detection time lag diminishes with increasing ventilation speed.

4) The second most important result can be gleaned by comparing Fig. 7a and 7b. Ventilation affects the
two detection performance metrics, detection time lag and source localization uncertainty in opposite ways: while detection time lag improves with ventilation, source localization uncertainty worsens. For example, with the previously noted fixed sensor density of 20 per 125 m$^3$, the source localization uncertainty without ventilation is roughly 1.5 m; it degrades to about 2.5 m with a 5 m/s ventilation. The same effect is observed with the moving sensors. However, the moving sensors maintain their significant performance advantage over fixed sensors in terms of source localization uncertainty with increasing ventilation speed. As a side note, the source localization uncertainty is less sensitive to changes in ventilation speed than the detection time lag.

5) The last salient result concerns the state estimation error in Fig. 7c. We observe that the estimation error shares similar trend with the detection time lag at smaller density. Faster heat transfer reduces the spatial variance of the temperature, thus yielding a more uniform temperature distribution. As a result, fewer sensors are needed to derive the same level of full state estimation accuracy. However, at higher density, the limiting performance is similar for different ventilation speed. The reason is that with higher sensor density, both fixed and moving sensors have a better coverage of the whole space. They can achieve accurate estimation regardless of the environment dynamics. Consequently, the moving sensors achieve limited improvement in full state estimation at high density.

Beyond these detailed results, this subsection demonstrates three main, high-level findings: (1) that AMASS provides significant and robust advantages of 10−15 times improvement over the traditional fixed sensor monitoring and detection strategy across the three performance metrics here considered; (2) that ventilation has significant effects on the monitoring performance of any architecture by 6−40 times—not accounting for ventilation in analyzing the performance of an environmental monitoring architecture is fundamentally myopic; and (3) that the monitoring performance cannot be fully reflected in a monolithic, single performance metric, but should include different metrics for the timeliness and spatial resolution of the detection function. The importance of this last point is reflected in the fact that ventilation affects differently these two performance aspects. As a result, if both aspects are important in a particular context, as is the case in the fire detection application considered here, it is myopic to restrict the analysis to a single performance metric such as the detection time lag.

VI. CONCLUSION AND FUTURE WORK

For future crewed space missions that could last years with limited ground support, the environmental control and life support system (ECLSS) will have to evolve to meet new, more stringent reliability and autonomy requirements. In this work, we focused on improving the performance of the environmental monitoring and anomaly detection systems using Markov decision process (MDP) and active sensing. We exploited actively moving sensors to tackle the current monitoring challenges and address the new requirements. We developed a novel sensing architecture and supporting analytics, termed Active environmental Monitoring and Anomaly Search System (AMASS). We designed a dynamic value iteration (DVI) policy to solve the path planning problem for the moving sensors in a dynamic environment. Although developed in the context of a space habitat and for the purpose of improving the ECLSS environmental monitoring performance, AMASS is also relevant beyond this particular context, and some of its foundational ideas and analytical developments (e.g., its analyzer, and policymaker, including the DVI) can be useful for ground-based civilian and military applications. We examine these in an upcoming publication.

To test and validate AMASS in a micro-gravity environment, we developed a series of computational experiments of environmental (temperature) monitoring and fire search, and we assessed the performance of our monitoring architecture against three metrics: (1) the anomaly detection time lag, (2) the source location uncertainty, and (3) the state estimation error. We first compared our DVI policy with three additional policies within AMASS and found that it provides the realistic best-in-class performance of all the benchmarked policies. We then used the DVI policy to benchmark the performance of moving sensors with AMASS against that of the traditional fixed sensor (FS) strategy for environmental monitoring and fire detection. We investigated the monitoring performance with respect to two critical factors: sensor density and ventilation effect in the habitat. The results demonstrate three most salient high-level findings: first, that AMASS provides significant and robust advantages of 10−15 times improvement over the traditional FS strategy across the three performance metrics here considered; second, that ventilation in the monitored environment has significant effects on the performance of any monitoring architecture by 6−40 times whether involving fixed or moving sensors; third, that the monitoring performance cannot be fully reflected in a monolithic, single metric, but should include different metrics for the timeliness and spatial resolution of the detection function.

This work addressed some research questions, but it also raised other important questions, which pave the way to several fruitful venues for future work. For example, we compared in this work two distinct monitoring architectures, with either fixed sensors or moving sensors, but not both. The examination of hybrid monitoring architectures that include both fixed and moving sensors is an important next step, and it will likely provide significantly rich opportunities for analytical developments and novel insights (for design, operation, and monitoring performance). We also noted in the text another important future research direction: a cost-benefit analysis for the monitoring strategies (FS, AMASS, and hybrid monitoring architectures) to delineate the entirety of the trade-space, not just detection performance, and to identify whether tipping points in favor of one or the other or
Pareto-optimal architecture exist. Finally, we compared in this work our DVI policy with additional heuristic policies. In future work, we propose to extend this comparison with more deep reinforcement learning (DRL) moving policies.

**APPENDIX A**

**FULL SIMULATION PARAMETERS**

See Table 3.

**REFERENCES**

[1] P. Eckart, *Spaceflight Life Support and Biospherics*, Dordrecht, The Netherlands: Springer, 1996.

[2] P. O. Wieland, *Design for Human Presence in Space: An Introduction to Environental Control and Life Support Systems—Update*, Huntsville, AL, USA: NASA Marshall Space Flight Center, 2005.

[3] G. Jones, H. Hodgson, E. Kliss, and M. Gentry, “How do lessons learned on the international space station (ISS) help plan life support for mars?” in *Proc. 46th Int. Conf. Environ. Syst.*, Jun. 2016, pp. 1–14.

[4] M. Jernigan, R. Gatens, J. Joshi, and J. Perry, “The next steps for environmental control and life support systems development for deep space exploration,” in *Proc. 48th Int. Conf. Environ. Syst.*, Jul. 2018, pp. 1–8.

[5] S. Eshima and J. Nabity, “Failure mode and effects analysis for environmental control and life support system self-awareness,” in *Proc. Int. Conf. Environ. Syst.*, 2020, pp. 1–13.

[6] D. Jan and R. Newton, “Environmental monitoring as part of life support: Deep space exploration,” in *Proc. 42nd Int. Conf. Environ. Syst.*, Jul. 2012, p. 3433.

[7] R. Wang, M. Veloso, and S. Seshan, “Active sensing data collection with autonomous mobile robots,” in *Proc. IEEE Int. Conf. Robot. Auton.*, (ICRA), May 2016, pp. 2583–2588.

[8] M. Bualat, J. Barlow, T. Fong, C. Provencher, and T. Smith, “Astrobee: Developing a free-flying robot for the international space station,” in *Proc. AIAA SPACE Conf. Expo.*, Aug. 2015, pp. 1–10.

[9] D. Miller, A. Saenz-Otero, J. Wertz, and A. Chen, “SPHERES: A testbed for long duration satellite formation flying in micro-gravity conditions,” in *Proc. AAS/AIAA Space Flight Mech. Meeting*, vol. 105, Jun. 2000, pp. 167–179.

[10] C. E. Schroeder, D. A. Wilson, T. Radman, H. Scharffman, and P. Lakatos, “Dynamics of active sensing and perceptual selection,” *Current Opinion Neurobiol.*, vol. 20, no. 2, pp. 172–176, Apr. 2010.

[11] D. Kent and S. Chernova, “Human-centric active perception for autonomous observation,” 2020, arXiv:2006.00037. [Online]. Available: http://arxiv.org/abs/2006.00037

[12] X. Lan and M. Schwager, “Learning a dynamical system model for a spatiotemporal field using a mobile sensing robot,” in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 170–175.

[13] R. Petrolo, Y. Lin, and E. Knightly, “ASTRO: Autonomous, sensing, and tetherless networKed dOnEs,” in *Proc. 4th ACM Workshop Micro Vehicl Micro Netw., Syst., Appl.*, Jun. 2018, pp. 1–4.

[14] T. Kollar and N. Roy, “Trajectory optimization using reinforcement learning for map exploration,” *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 175–196, 2008.

[15] A. Marjovi, J. G. Nunes, L. Marques, and A. da Almeida, “Multi-robot fire searching in unknown environment,” in *Springer Tracts in Advanced Robotics*, vol. 62, B. Siciliano, O. Khatib, F. Groen, A. Howard, K. Iagnemma, and A. Kelly, Eds, Berlin, Germany: Springer, 2010, pp. 341–351.

[16] F. Laport-López, E. Serrano, J. Bajo, and A. T. Campbell, “A review of mobile sensing systems, applications, and opportunities,” *Knowl. Inf. Syst.*, vol. 62, no. 1, pp. 145–174, Jan. 2020.

[17] L. Mihaylova, T. Lefebvre, H. Bruynincks, K. Gadeyne, and J. De Schutter, “Active sensing for robotics—A survey,” in *Proc. Conf. Robot.*, 2002, pp. 1–8.

[18] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, “Decentralized time-varying formation control for multi-robot systems,” *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 1029–1043, Jun. 2014.

[19] J. Capitan, M. T. J. Spaan, L. Merino, and A. Ollero, “Decentralized multi-robot cooperation with auctioned POMDPs,” in *Proc. IEEE Int. Conf. Robot. Auton.*, May 2012, pp. 515–518.

[20] S. Omidsfahie, A. Agha–Mohammadi, C. Amato, S. Liu, J. P. How, and J. Vian, “Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions,” *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 231–258, Feb. 2017.

[21] T. Fan, P. Long, W. Liu, and J. Pan, “Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios,” *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 856–892, Jun. 2020.

[22] R. N. Haksr and M. Schwager, “Distributed deep reinforcement learning for fighting forest fires with a network of aerial robots,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1067–1074.

[23] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, “Concrete problems in AI safety,” 2016, arXiv:1606.06565. [Online]. Available: http://arxiv.org/abs/1606.06565

[24] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” in *Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1238–1247, Sep. 2013.

[25] S. Arora and P. Doshi, “A survey of inverse reinforcement learning: Challenges, methods and progress,” 2018, arXiv:1806.06877. [Online]. Available: http://arxiv.org/abs/1806.06877

[26] M. Hauskrecht, “Value-function approximations for partially observable Markov decision processes,” *J. Artif. Intell. Res.*, vol. 13, pp. 33–94, Aug. 2000.

[27] J. Wang, X. Wang, and M. Hong, “Gas leak location detection based on data fusion with time difference of arrival and energy decay using an ultrasonic sensor array,” *Sensors*, vol. 19, no. 8, p. 2985, Sep. 2018.

[28] L. Qiang, “Estimation of fire detection time,” *Proceeda Eng.*, vol. 11, pp. 233–241, Dec. 2011.

[29] F. Aghil and C.-Y. Su, “Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU,” *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 2015–2026, Aug. 2016.

[30] T. Moore and D. Stouch, “A generalized extended Kalman filter implementation for the robot operating system,” in *Advances in Intelligent Systems and Computing*, vol. 302, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds. Cham, Switzerland: Springer, 2016, pp. 335–348.

[31] N. Thacker and A. Lacey, “Tutorial: The Kalman filter,” *Imag. Sci.* vol. 13, pp. 33–94, 1957.

[32] J. Artif. Intell. Res. vol. 6, no. 4, p. 679–684, 1996.

[33] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *On-line learning and stochastic approximations*.

[34] L. Bottou, “Online learning and stochastic approximations,” *On-Line Learning. Neural Netw.*, vol. 17, no. 9, pp. 9–42, 2010.

[35] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, 2015, pp. 1–5

[36] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965.
ZHAOYI XU received the B.S. degree in aerospace engineering from Northwestern Polytechnical University, China, in 2015, and the M.S. degree in aerospace engineering from the Georgia Institute of Technology, where he is currently pursuing the Ph.D. degree with the Guggenheim School of Aerospace Engineering. His work on supercritical airfoil optimization, using the genetic algorithm (GA) with ML surrogate was recently published in the AIAA Journal. His research interests include machine learning (ML) in engineering applications, with an emphasis on ML for reliability and system safety, and design optimization. He is also proficient in computational fluid dynamics (CFD). He received an Outstanding Undergraduate Thesis Award from Northwestern Polytechnical University, based on his work on transonic unsteady flow.

JOSEPH HOMER SALEH (Senior Member, IEEE) received the master’s degree from Harvard University and the Ph.D. degree from MIT, in 2002. Prior to joining Tech, he served as the Executive Director for the Ford-MIT Research Partnership. He is currently an Associate Professor with the School of Aerospace Engineering, Georgia Tech. He has authored two books and 78 journal articles. His research interests include spacecraft reliability and multi-state analysis, analytical systems engineering with a focus on its temporal dimension, and multidisciplinary accident causation and system safety. He is an Associate Fellow of AIAA. He has received several awards at Georgia Tech, including the Outstanding Faculty Award, the Lockheed Dean’s Excellence in Teaching Award, and the Class of 1940 Roanne Beard Outstanding Teaching Award. He served as an Associate Editor for Reliability Engineering and System Safety and the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS.