

Received 27 November 2023, accepted 27 December 2023, date of publication 2 January 2024, date of current version 10 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349319

## RESEARCH ARTICLE

# A Two-Layer Control Framework for Persistent Monitoring of a Large Area With a Robotic Sensor Network

**MADE WIDHI SURYA ATMAN**<sup>1</sup>, (Member, IEEE),  
**MUHAMAD RAUSYAN FIKRI**<sup>1</sup>, (Member, IEEE),  
**KARLISA PRIANDANA**<sup>2</sup>, (Senior Member, IEEE),  
**AND AZWIRMAN GUSRIALDI**<sup>1</sup>, (Member, IEEE)

<sup>1</sup>Faculty of Engineering and Natural Sciences, Tampere University, 33720 Tampere, Finland

<sup>2</sup>Department of Computer Science, IPB University, Bogor 16680, Indonesia

Corresponding author: Made Widhi Surya Atman (widhi.atman@tuni.fi)

The work of Made Widhi Surya Atman and Azwirman Gusrialdi was supported by the Research Council of Finland under Academy Project under Grant 330073. The work of Azwirman Gusrialdi was supported by the Research Council of Finland under the Profiling Action on System-on-Chip and Wireless Technology for Intelligent Machines at Tampere University. The work of Karlisa Priandana was supported by the Ministry of Education, Culture, Research and Technology Republic of Indonesia Grant through the Directorate of Research and Innovation, IPB University, under Contract 001/E5/PG.02.00.PL/2023 jo 15853/IT3.D10/PT.01.02/PT/2023.

**ABSTRACT** The deployment of a group of robots equipped with sensors for monitoring, also known as a robotic sensor network, is a promising technological solution to solve time-critical societal and environmental issues. This paper considers the problem of deploying a robotic sensor network to persistently and effectively monitor multiple locations of interest in a large field represented by grids. To this end, we propose a novel two-layer control framework where the first layer (i.e., task allocation strategy) encapsulates the targeted grids into a set of tasks (small regions of interest) followed by optimally allocating the robots to each task based on their initial position (location of their base stations) and sensing capabilities. The second layer (i.e., persistent monitoring algorithm) generates each robot's motion control input to ensure persistent monitoring over the designated region of interest. The proposed framework is demonstrated and evaluated via numerical simulations. It is shown that the proposed control framework improves real-time monitoring in terms of both the coverage performance and travel distance of the robots.

**INDEX TERMS** Automation, coverage control, mobile sensors, persistent monitoring, task allocation.

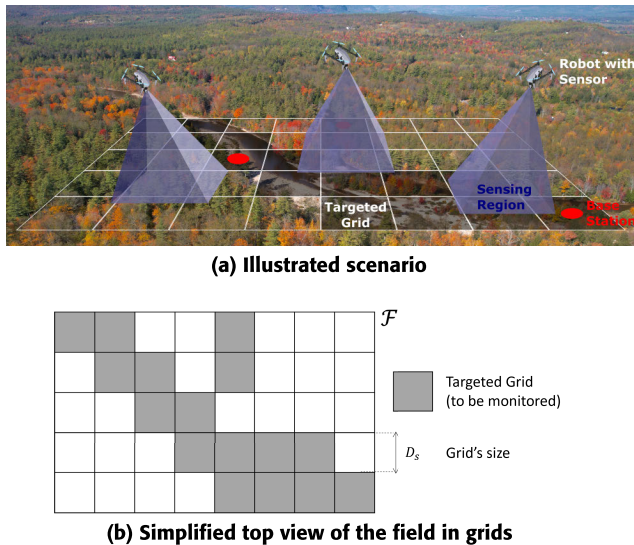
## I. INTRODUCTION

The deployment of a group of robots equipped with sensors for monitoring, also known as a robotic sensor network, is a promising technological solution to solve time-critical societal and environmental issues, such as infrastructure survey, environmental monitoring, and search-and-rescue mission [1], [2], [3], [4], [5], [6]. Notably, a robotic sensor network promises a multitude of benefits, such as the capability to cover multiple places at the same time, robustness to individual robot failure, and flexibility in

The associate editor coordinating the review of this manuscript and approving it for publication was Renato Ferrero<sup>1</sup>.

completing a complex task. To take the most of these benefits in ensuring a successful monitoring mission, a robotic sensor network requires a coordinated control scheme to effectively cover the targeted areas of interest.

The focus of this paper is on the deployment of a robotic sensor network to persistently and effectively monitor multiple locations spread across a large field, as depicted in Fig. 1a. To this end, coverage control has been demonstrated as one of the most promising approaches for coordinating a robotic sensor network [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Briefly speaking, coverage control is a distributed control strategy for deploying a robotic sensor network by maximizing a cost function that represents the information



**FIGURE 1.** The scenario of a field divided into a collection of grids monitored by a robotic sensor network.

gained over a given domain. Here, the importance of each location of interest is emphasized by a spectrum of values denoted as the density function. Coverage control has been studied for various sensors' footprints [10], [11] and has also been demonstrated on ground robots [10], flying robots [12], [13], and a combination of both [14], [15]. Originally, the coverage control algorithm was designed to ensure a stationary configuration of the optimal robots' position. However, when the region to be monitored is much larger than the total sensing capability (footprint) of the robotic sensor network, having a stationary sensor configuration may result in some important areas being left uncovered.

An extension of the coverage control to ensure the whole region to be covered at least one time has often been categorized as dynamic coverage control [12], [16], [17], [18], [19], [20]. The idea is to introduce an exogenous input for the robotic sensor network given by a time-varying density function. Specifically, the density function on a sub-region is updated, marking it less important, once a robot senses it. Built on top of this idea, the persistent coverage control scheme [15], [20], [21] introduces the concept of information decay that rebounds the density function after the robot leaves the sub-region for a certain time. This addition generates a persistent patrolling motion over a given region of interest. The literature has often demonstrated persistent coverage control via full-domain exploration, where the whole domain initially has equal importance. Note that specifications for monitoring vary depending on the application scenarios. In some practical cases, the full-domain patrolling motion may not be required. Instead, patrolling only over several locations within the domain can be sufficient. To this end, the discussion on persistent coverage control which focuses on only a spread of locations in the large region of interest remains lacking from the literature. This particular case is especially important as a poorly designed controller may

render the robots stuck in local optimal as shown in [22] which also makes the robots waste their energy to monitor unimportant areas.

A potential strategy to remove the unimportant areas is by restructuring the domain into a number of smaller regions, where each region represents a cluster of important areas to be persistently covered. To this end, one needs to solve the following problems: i) how to effectively divide the domain and ii) how to optimally allocate the robotic sensor network to the cluster of areas, which is a combinatorial optimization problem. Swarm intelligence is a class of meta-heuristics approach which is able to find good solutions (but not necessarily optimal) of combinatorial optimization problems within a reasonable amount of time. Particularly, the Ant-Colony Optimization (ACO) algorithm [23] is a well-known swarm intelligent algorithm for solving the Traveling Salesman Problem, that has also been adopted to the Task Allocation problem in [24], [25], [26], [27], and [28]. Regardless, a thorough and extensive study is required to explore the potential use of ACO in persistent monitoring using a robotic sensor network.

In this paper, we propose a novel two-layer control framework to persistently monitor the targeted locations in a large field, here formulated as a set of targeted grids, to improve the coverage performance and reduce the robots' travel distance. Specifically, the first layer (task allocation strategy) encapsulates the targeted grids as several smaller regions via K-means clustering [29], [30] followed by assigning the robots to monitor the regions accordingly via a modified ant colony optimization (M-ACO) [31]. Then, the second layer (persistent monitoring algorithm) models the importance of the area in each smaller region as a density function and further generates the control input for each robot based on a persistent coverage control algorithm [20], [21]. In particular, the main contributions of this paper are listed as follows:

- We propose a modified ant colony optimization (M-ACO) to find the most suitable agents (robots) to cover the tasks (regions), where the earlier version is presented in [31]. The proposed M-ACO in [31] prioritizes the compatibility of the task area with the total capability of the agent(s) allocated to that particular task, resulting in a limitation where task(s) with a small area may not be covered by any agent. In this work, modifications to the M-ACO are introduced to guarantee that each task is covered by at least one agent.
- A novel design of the density function is introduced to model the area of importance within the region. The design is also a novel feature of our coverage algorithm, as most of the literature on coverage control assumes that the density function is given. Furthermore, compared to our previous framework in [31], we improve the encapsulation of the targeted grids as a convex hull (formerly a box rectangle) which can further improve the coverage performance. Collision avoidance between the robots is also explicitly considered by imposing

additional constraints using the control barrier function approach [6], [32], [33], [34].

- We present an extensive simulation for the proposed two-layer control framework. It highlights the benefits of dividing the targeted grids into several smaller regions that help improve coverage performance and reduce the overall travel distance.

The rest of this article is organized as follows. In Section II, we provide the problem settings considered in this paper. We present our two-layer control framework in Section III, followed by a more detailed formulation of the first layer (Task Allocation Strategy) in Section IV and the second layer (Persistent Monitoring Algorithm) in Section V. Section VI presents the numerical simulation and discussion of the proposed framework. Finally, Section VII presents a concluding remark.

## II. PROBLEM STATEMENT

In this paper, we consider a scenario of monitoring a specified *field*, formally denoted as a convex polytope-shaped region (compact set)  $\mathcal{F} \subset \mathbb{R}^2$ . The field  $\mathcal{F}$  is further discretized into square grids with equal width  $D_s$  where a number of grids of interest need to be monitored (also referred to as *targeted grids*), as depicted in Fig. 1. Let the set  $\mathcal{M}$  denote a collection of indices of  $m$  number of targeted grids in the field  $\mathcal{F}$  whose center location is represented by  $h_i \in \mathcal{F}$ ,  $i \in \mathcal{M}$ . The targeted grids over the field  $\mathcal{F}$  can then be defined as

$$\mathcal{H} = \bigcup_{i \in \mathcal{M}} \{q \in \mathcal{F} \mid \|q - h_i\|_\infty \leq D_s/2\}.$$

To monitor the targeted grids  $\mathcal{H}$ , we consider a solution with  $n$  number of autonomous robots equipped with sensors working in cooperation, often referred to as robotic sensor networks.

Let the set of identifiers for the robots is denoted by  $\mathcal{I} = \{1, \dots, n\}$ , and let the term  $p_i(t) = [x_i(t) \ y_i(t)]^T \in \mathcal{F}$  describes the position of the robot  $i \in \mathcal{I}$  at the time  $t$ . Here, we assume that each robot starts from its own base station  $p_{i_b}$ , i.e.,  $p_i(0) = p_{i_b}$ . Furthermore, it is assumed that the position of each robot is updated according to the kinematic model<sup>1</sup>:

$$\dot{p}_i = u_i. \quad (1)$$

where  $u_i \subseteq \mathbb{R}^2$  denotes the velocity input to be designed.

Each robot  $i$ 's sensor is assumed to be able to sense a subset of the area in the field within a fixed sensing radius  $R_i > 0$ , with the sensing region further defined as

$$\mathcal{B}_i(p_i) = \{q \in \mathcal{F} \mid \|q - p_i\| \leq R_i\}.$$

Within this paper, let us define the size of a given region  $\mathcal{R}$  as  $A(\mathcal{R})$ . We can then quantify the sensing capability of each robot as the size of its sensing region, i.e.,  $A(\mathcal{B}_i) := \pi R_i^2$ .

To this end, the objective of this paper is to design a cooperative control framework for the robotic sensor network

<sup>1</sup>Despite its simplicity, the single-integrator model (1) can still be effectively used to control various mobile robots. For example, by having designed a good trajectory tracking controller, the kinematic model (1) can be used to generate velocity references for mobile robots such as quadrotors.

to maximize coverage over the targeted grids, i.e.,  $\mathcal{H}$ , at all times. Note that in most cases, the energy consumption of the robot is proportional to its travel distance [33]. Thus, to ensure the long duration of the monitoring mission, the proposed solution needs to specifically account the distribution of the targeted grids to reduce the required traveling distance from the base station as well as during the monitoring mission.

An example of the considered scenario is the monitoring of a field, where the grids correspond to areas that each static sensor needs to cover [31]. However, coverage holes (unmonitored areas) may appear, e.g., due to malfunction or lack of sensors [35], [36]. As a complementary solution, a group of quadrotors equipped with sensors (the robotic sensor network) can then be deployed for monitoring the coverage holes (the targeted grids), resulting in a hybrid monitoring system that comprises of static and mobile sensors [37], [38], [39].

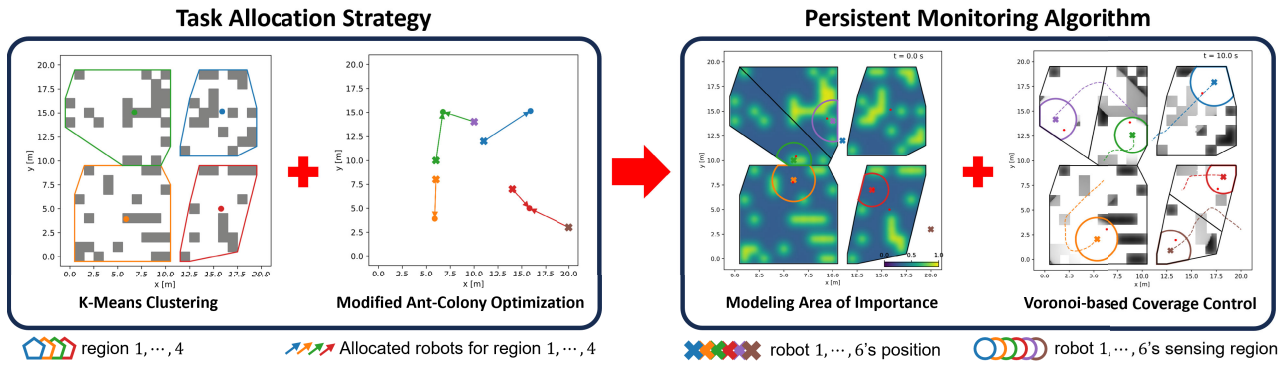
## III. PROPOSED TWO-LAYER CONTROL FRAMEWORK

Given the presented problem statement and objectives, we formulate our framework for solving the following problems:

- 1) to encapsulate the targeted grids  $\mathcal{H}$  to monitor as a set of tasks and provide an optimal allocation based on the robots' initial position and sensing capabilities, and
- 2) to design a control input  $u_i$  in (1) for each robot  $i \in \mathcal{I}$  to ensure persistent monitoring over the designated tasks' region.

The separation of these two problems allows us to approach the solution in two separate layers: the first layer (i.e., a higher layer) which divides the original targeted grids into smaller tasks followed by allocating each robot to one of the tasks while accounting for the expected distance each robot needs to take, and the second layer (i.e., a lower layer) which directly governs the motion of the robot to ensure persistent monitoring. Throughout this paper, we refer to these higher and lower layers respectively as *task allocation strategy* and *persistent monitoring algorithm*. The Fig. 2 illustrates our proposed two-layer control framework.

The task allocation strategy comprises dividing the targeted grid  $\mathcal{H}$  into  $\ell$  smaller regions via K-means clustering and then assigning the  $n$  number of robots accordingly via a modified ant colony optimization (M-ACO). The process of dividing the targeted grids  $\mathcal{H}$  into  $\ell$  smaller regions reduces the total area to be covered and, at the same time, localizes the area that each robot needs to monitor. Here, the K-means clustering algorithm is chosen since it is a straightforward clustering method that can decide whether an object belongs to one cluster or not [29], [30]. Since K-means clustering aims to minimize the pairwise squared deviations of points (grids) in the same regions (clusters), this will help to reduce the travel distance of the drone for monitoring all the grids within its region. Moreover, the requirement to define an explicit number of the generated clusters allows a thorough study of how these region divisions impact the coverage performance, as will be presented later



**FIGURE 2.** An overview of the proposed two-layer control framework. The targeted grids are divided into 4 regions using K-means clustering. The Modified Ant-Colony Optimization algorithm assigns 6 robots to monitor the 4 regions. The importance of area in each region is modeled as a density function, where the coverage hole has a high value (yellow). Finally, the Voronoi-based coverage control provides the control input for each robot to maximize the coverage of the region.

in Section VI. We further impose that  $\ell \leq n$  and each region need to be monitored by at least 1 robot and all  $n$  robots need to be allocated. The Modified Ant Colony Optimization (M-ACO) is then used to allocate the robots to regions by considering the size of each region, the sensing capability of the robot, and the distance between the robot's initial position and the region. Our proposed M-ACO method is a modified version of the ACO algorithms that aims at finding the most suitable agents (robots) to cover the tasks (regions). Here, some modifications are introduced to fit the intended scenario and the aimed objective.

Finally, the persistent monitoring algorithm comprises modeling the importance of area on each smaller region as a density function and deploying the robots based on a time-varying Voronoi-based coverage controller [20], [21]. Here, we introduce a novel approach to construct a density function for fields in grids. Then, the control input  $u_i$  for each robot is computed via a gradient ascent algorithm to maximize the coverage within each robot's sensing area. The density value across each region varies over time, i.e., the value at a certain position decreases when a robot passes through it and increases again once it is outside any robot's sensing area. These time-varying properties of the density function along with the gradient ascent algorithm ensure a continuous motion to travel to all the targeted grids. Additionally, obstacle avoidance via control barrier function [6], [34] is implemented to ensure no collision between robots. The detailed descriptions of our approaches are described in more detail in the subsequent sections.

#### IV. TASK ALLOCATION STRATEGY

In this section, we detail our proposed approach for a higher layer, which introduces  $\ell$  smaller regions and then accordingly allocates all  $n$  numbers of robots into the regions. Each smaller region is identified as  $Q^j$ ,  $j \in \mathcal{C} := \{1, \dots, \ell\}$  with its centroid  $c^j \in \mathbb{R}^2$ ,  $j \in \mathcal{C}$ . The output of the modified ant colony optimization (M-ACO) is the sets of robots  $\mathcal{I}^j \subseteq \mathcal{I}$  to monitor each region  $j \in \mathcal{C}$ .

#### A. K-MEANS CLUSTERING

The K-Means clustering requires the explicit number of clusters to be generated and the data to be clustered, i.e.,  $\ell$  and  $h_i$ ,  $\forall i \in \mathcal{M}$ . In practice,  $\ell$  is chosen by considering the number of available robots, the distribution of the targeted grids, and the size of the field  $\mathcal{F}$ . The summarized procedure of K-Means clustering is given as follows:

- 1) First, initialize  $\ell$  number of random points inside  $\mathcal{F}$  as the initial centroids of the cluster.
- 2) Each point located at  $h_i$ ,  $\forall i \in \mathcal{M}$  is then assigned to the closest centroid, by calculating the Euclidean distances between the point and the centroids.
- 3) The centroid positions are then updated by taking the average of all points that were assigned to them.
- 4) Steps 2 and 3 are repeated until the centroid positions converge, i.e., do not change anymore.

We direct the interested readers to [29] for more details on the algorithm.

The output from K-Means clustering is  $\ell$  groups of targeted grids. For each group  $j \in \mathcal{C}$ , the cluster of the targeted grids is denoted as  $\mathcal{M}^j \subseteq \mathcal{M}$ . Then, the region to be monitored in each group  $j$  is constructed as a convex hull (conv) that contains each grid  $i \in \mathcal{M}^j$ , i.e.,

$$Q^j = \text{conv} \left( \bigcup_{i \in \mathcal{M}^j} \{q \in \mathcal{F} \mid \|q - h_i\|_\infty \leq D_s/2\} \right),$$

with the region's centroid as  $c^j$ . To this end, the combination of the regions overlaps the targeted grids  $\mathcal{H}$ , i.e.,  $\bigcup_{j \in \mathcal{C}} Q^j \supseteq \mathcal{H}$ . The resulting  $\ell$  regions from this procedure are illustrated in Fig. 2, where each region for  $\ell = 4$  is denoted by the colored polygons with its centroid is denoted by the colored circle.

#### B. MODIFIED ANT COLONY OPTIMIZATION

In this subsection, we describe our proposed modified ant colony optimization (M-ACO) method. The goal is to obtain an equal distribution of  $n$  robots in covering the  $\ell$  regions,



where  $\ell \leq n$ . Referring to the terminology of “agents” and “tasks” as commonly used in ACO [23], in this work, the agents are the robots and the tasks are the regions with targeted grids. To this end, both the robots (agents) and the regions (tasks) are represented as nodes, while the ants are the computational units to find the optimum coalition of robots. Given an  $N$  number of ants and  $M$  number of iterations, the general procedure of the M-ACO method is summarized as follows:

- 1) An ant  $k \in \{1, \dots, N\}$  randomly chooses the pairing of region and robot based on a given probability function  $p_{ij}^k(t)$ . Here,  $p_{ij}^k(t)$  defines the probability of ant  $k$  in pairing robot  $i$  with region  $j$  based on the pheromone concentration and heuristic function at the time  $t$ .
- 2) Ant  $k$  repeats step 1 until all regions are allocated with at least 1 robot and one of the following conditions is satisfied: each robot has been allocated to a region, or all regions has been covered with enough robots.
- 3) Repeat steps 1-2 for all  $N$  ants.
- 4) Updates the pheromones concentration based on the chosen pairings from all  $N$  ants.
- 5) Repeat steps 1-4 for  $M$  number of iterations.

In this work, we define the probability for ant  $k$  in pairing robot  $i$  with region  $j$  as

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}^k(t)]^\beta}{\sum_{a,b \in allowed_k} [\tau_{ab}(t)]^\alpha [\eta_{ab}^k(t)]^\beta}; & i, j \in allowed_k \\ 0; & otherwise \end{cases} \quad (2)$$

where  $\tau_{ij}$  denotes the pheromone concentration of pair  $ij$ ,  $\eta_{ij}^k$  denotes the heuristic function of pair  $ij$  for ant  $k$ , and  $\alpha$  and  $\beta$  denote the importance of the pheromone and the heuristic value respectively. Here,  $allowed_k$  is the set of robots and regions that can be chosen by ant  $k$ . In our previous work [31],  $allowed_k$  is defined as the robots that have not been allocated to any region and the regions that still have an “uncovered” area. Here, we modify the definition of allowed regions in  $allowed_k$  into two conditions: 1) If there are still regions that have not been paired with any robot, then the regions that can be selected are the regions that have not been assigned with any robot; 2) If all regions have been allocated with at least one robot, then the regions that can be selected are the regions that have not been fully covered.

The heuristic function  $\eta_{ij}^k$  is defined as

$$\eta_{ij}^k = \begin{cases} \frac{A(Q^j) - \sum_{l \in \mathcal{I}^j} A(B_l)}{\|p_{i_b} - c^j\|}; & A(Q^j) > \sum_{l \in \mathcal{I}^j} A(B_l) \\ 0; & otherwise \end{cases} \quad (3)$$

where  $\mathcal{I}^j \subset \mathcal{I}$  is the set of robots already selected for the region  $j \in \mathcal{C}$ . The heuristic function in (3) considers the following terms:

- 1) size of each region,  $A(Q^j)$ ,  $j \in \mathcal{C}$ ;
- 2) sensing capability of each robot,  $A(B_i)$ ,  $i \in \mathcal{I}$ ;

- 3) distance between the robot’s initial position (the base station) and the centroid of the region,  $\|p_{i_b} - c^j\|$ ,  $i \in \mathcal{I}$ ,  $j \in \mathcal{C}$ ;

such that the ants tend to pair the region that has not been fully covered yet and then pair it with the closest robot. Note that due to the change of definition in  $allowed_k$  which regulates how the ants pair a region with a robot, the heuristic function introduced in [31] may give a negative value when the size of the region is smaller than the sensing capability of the robot assigned to it. This will result in negative probability (see (2)). Thus, an additional term is added to avoid a negative value in the heuristic function (see (3)).

One iteration of pairing the robots and regions for a single ant (Step 1–2) is defined as one tour. At the end of one tour, each ant records the changes it imposes to the pheromone in each pair  $ij$  (pairing of robot  $i$  to region  $j$ ) as

$$\Delta \tau_{ij}^k = \begin{cases} P\epsilon^k; & \text{if ant } k \text{ select agent } i \text{ for task } j \\ 0; & \text{otherwise} \end{cases} \quad (4)$$

where  $P$  is the pheromone strength, and  $\epsilon^k$  is the efficiency factor of ant  $k$  in one tour, defined as

$$\epsilon^k = \frac{\sum_{j \in \mathcal{C}} (A(Q^j) - |\sum_{k \in \mathcal{I}^j} A(B_k) - A(Q^j)|)}{\sum_{j \in \mathcal{C}} (\sum_{k \in \mathcal{I}^j} d_{kj})}$$

Once all ants finish their pairings (step 3), the pheromones for each pair  $ij$  are updated according to

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \sum_N \Delta \tau_{ij}^k(t),$$

where  $\rho$  is the pheromone evaporation coefficient. Here, the pheromone concentration is iteratively updated by minimizing the following two costs: (i) the difference between the size of the region and the total sensing capability of the allocated robots; (ii) the distance between the paired robots and regions. At the end of  $M$  iterations, the chosen solution is the ant tour which yields the maximum efficiency factor  $\epsilon^k$ .

The output of the M-ACO algorithm is the set of indexes  $\mathcal{I}^j \subseteq \mathcal{I}$ ,  $j \in \mathcal{C}$  which ensures a distinct selection of robots to each given region, i.e.,  $\mathcal{I}^j \cap \mathcal{I}^k = \emptyset$  for  $j \neq k$ . The information of each region  $Q^j$ ,  $j \in \mathcal{C}$ , the location of targeted grids within (i.e.,  $h_i$ ,  $\forall i \in \mathcal{M}^j$ ), and the selected robots (i.e.,  $\mathcal{I}^j$ ) will be used to generate each robot’s control input  $u_i$  as described in the next section.

## V. PERSISTENT MONITORING ALGORITHM

In this section, we describe our proposed approach for the lower layer, that formulates each robot’s control input to ensure all the assigned robots persistently cover their designated region. For simplicity, in the remain of the discussion we focus on a single region of  $Q^j$  and the robots in the set  $\mathcal{I}^j$  where  $j \in \mathcal{C}$ .

### A. MODELING AREA OF IMPORTANCE

In this subsection, we first describe the procedure of defining the degree of importance for all locations within the region

$\mathcal{Q}^j$ . Here, we define a range of value  $[0, 1]$  where 1 specifies the location with the highest importance to monitor and 0 as the lowest. Thus, the locations nearby  $h_i, \forall i \in \mathcal{M}^j$  (targeted grids within  $\mathcal{Q}^j$ ) need to be represented with values close to 1. To achieve that, we design a density function  $\hat{\phi}^j : \mathcal{Q}^j \rightarrow [0, 1]$  which is formulated using a mixture of Gaussian functions:

$$\hat{\phi}^j(q) := c_f + k \sum_{i \in \mathcal{M}^j} \phi_i^j(q)$$

$$\phi_i^j(q) = \exp \left[ -\frac{1}{2}(q - h_i)^T \Sigma^{-1}(q - h_i) \right], \quad (5)$$

where  $\Sigma \in \mathbb{R}^{2 \times 2}$  is the covariance matrix,  $c_f \geq 0$  is a fixed minimum value of the density function, and  $k \in [0, 1]$  is a fixed gain to adjust the maximum density value. In this paper, we consider a covariance matrix in the shape of  $\Sigma = \sigma^2 \mathbf{I}_2$  with  $\sigma > 0$ . Note that each selection of  $c_f, k$ , and  $\sigma$  provides unique characteristics in highlighting the contrast of importance between the targeted grids and the remaining area in  $\mathcal{Q}^j$ , as will be shown later. Additionally, the parameters need to be selected such that the maximum density value over the field  $\mathcal{Q}^j$  does not exceed 1.

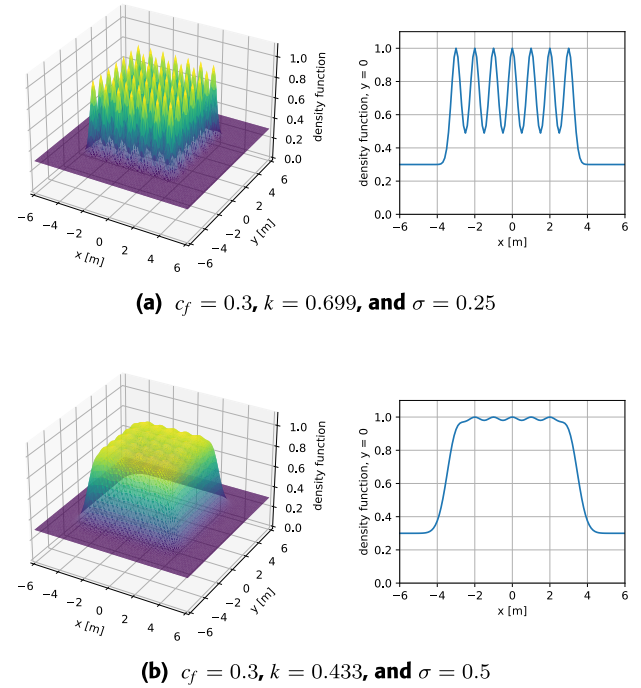
Given the formulation in (5), notice that any targeted grid  $i \in \mathcal{M}^j$  increases the density values over its surrounding grid's center  $h_i$ . Note that this influence diminishes as distance from  $h_i$  increases. The term  $\phi_i^j(q)$  can be written as  $\exp \left[ -\frac{(d_i^j(q))^2}{2\sigma^2} \right]$  with  $d_i^j(q) = \|q - h_i\|_2$ . Then, the computed values of  $\phi_i^j(q)$  for any points  $q$  around  $h_i$  whose  $d_i^j$  equal to  $\sigma, 2\sigma, 3\sigma$  and  $4\sigma$  are 0.607, 0.135, 0.011 and 0.0003, respectively. These series of values serve as a basic benchmark in guiding the selection of appropriate parameters for  $\sigma$  as well as  $k$  for a given field with grid distance  $D_s$ .

To illustrate this further, we present two examples for a field with 49 targeted grids (a  $7m \times 7m$  with  $D_s = 1m$ ) as shown in Fig. 3. Both examples share the same value of  $c_f = 0.3$  with two different sets of  $k$  and  $\sigma$ . Fig. 3a presents a case where  $\sigma = 0.25$  ( $D_s = 4\sigma$ ) which results in almost non-existent influence from  $\phi_i^j(q)$  to the adjacent grid. Hence, the selection of  $k$  can be set close to the remaining 0.7 values. On the other hand, Fig. 3b presents a case with  $\sigma = 0.5$  ( $D_s = 2\sigma$ ) that provides significant influence from  $\phi_i^j(q)$  to the adjacent grid, hence resulting in a smaller range of permissible values for  $k$ .

In both examples in Fig. 3, the max value of  $k$  which ensures  $\hat{\phi}^j(q) \leq 1$  for any  $q \in \mathcal{Q}^j$  can be approximated by considering the influence from direct adjacent neighboring targeted grids, such as solving

$$c_f + k \left( 1 + 4 \exp(-D_s^2/2\sigma^2) + 4 \exp(-2D_s^2/2\sigma^2) \right) < 1.$$

This results in  $k < 0.6991$  and  $k < 0.4335$  for the case in Fig. 3a and Fig. 3b, respectively. Note that with  $\sigma \leq 0.5 D_s$  the influence from the targeted grid further than  $\sqrt{2}D_s$  are significantly smaller, and hence it is sufficient to choose a slightly smaller value than the approximated  $k$ , e.g., 0.433 for Fig. 3b. However, with a larger value of  $\sigma$ , the designer needs



**FIGURE 3.** Examples of density functions for a field with 49 broken sensors ( $7 \times 7m^2$  with  $D_s = 1m$ ). (Left) The 3D plot of the density functions. (Right) A vertical cut of the density function for  $y = 0$ .

to assess the degree of influence from each Gaussian function  $\phi_i^j(q)$  component when deciding the appropriate value of  $k$ .

Finally, let us focus our discussion on the selection of  $c_f$ . Consider a case where  $\mathcal{M}^j$  comprises multiple clusters of targeted grids that are apart from each other, e.g., when the number of available drones is smaller than the number of clusters of targeted grids. An example of this is shown in Fig. 4. Here, the gaps between the groups may result in an area with  $\hat{\phi}^j(q) = c_f$ . Providing the  $c_f = 0$  can potentially hinder the robot from crossing from one area to another, as often observed for voronoi-based coverage controllers [22] (described later). Thus, a positive  $c_f$ , i.e.,  $c_f > 0$ , highlights a small importance of these regions where  $\hat{\phi}^j(q) = c_f$  and promotes the exploration that potentially results in the robot traversing to all targeted grids. Note that a high value of  $c_f$  further promotes the exploration, but with a drawback of reducing the relative importance of the targeted grids.

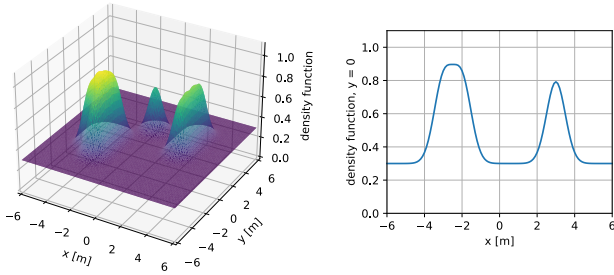
### B. VORONOI-BASED COVERAGE CONTROL WITH OBSTACLE AVOIDANCE

Finally, in this subsection, we present the computation for each robot's control input  $u_i$  based on the provided density map. First, let us consider the collocations of  $p_i$  for all  $i \in \mathcal{I}^j$  as  $p^j$ . Then, the Voronoi partition of each region  $\mathcal{Q}^j$  [7], namely the collection of the sets  $\{\mathcal{V}_i(p^j)\}_{i \in \mathcal{I}^j}$ , is defined as

$$\mathcal{V}_i(p^j) = \{q \in \mathcal{Q}^j \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \in \mathcal{I}^j \setminus \{i\}\}.$$

Moreover, let us define the feasible sensing area  $\mathcal{S}_i(p^j)$  as

$$\mathcal{S}_i(p^j) := \mathcal{B}_i(p_i) \cap \mathcal{V}_i(p^j).$$



**FIGURE 4.** An example scenario where the area of importance are distanced to each other. The value of  $c_f = 0.3$  highlights the small importance of the remaining regions, hence promoting exploration.

It is shown in [7] that the set  $\mathcal{S}_i(p^j)$  depends solely on the position of robots that lie within a radius  $2R_i$  from  $p_i$ . Therefore,  $\mathcal{S}_i(p^j)$  can be computed in a distributed fashion, e.g., by allowing robots to exchange position information within  $2R_i$  radius.

In the remaining discussion, let us consider a practical case when the total sensing capabilities of the assigned robots is much less than the total area to be covered, which requires the robots to periodically visit the broken sensors locations. To that end, we consider the following time-varying density function  $\phi^j : \mathcal{Q}^j \times \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ , updated according to

$$\frac{d\phi^j(q, t)}{dt} = \begin{cases} -\delta\phi^j(q, t), & \text{if } q \in \cup_{i \in \mathcal{I}^j} \mathcal{S}_i \\ \bar{\delta}(\hat{\phi}^j(q) - \phi^j(q, t)) & \text{otherwise} \end{cases} \quad (6)$$

with  $\bar{\delta}, \delta > 0$  and  $\hat{\phi}^j(q)$  refer to the density map designed via (5). The update rule (6) implies that the importance of a point being monitored by a robot is decreasing with rate  $\delta$ , and will then be increased with rate  $\bar{\delta}$  if it is left unmonitored and thus requiring the robot to revisit that point to maintain persistent monitoring of the targeted grids. Note that the parameters  $\bar{\delta}, \delta$  are assigned by the designer by considering the characteristics of the sensor attached to the robot. For example, if the sensor requires more time to take measurements, then  $\delta$  can be set to be small.

The control input  $u_i$  for all  $i \in \mathcal{I}^j$  can then be computed based on the gradient ascent algorithm to maximize the following objective function

$$J(p^j, t) := - \sum_{i \in \mathcal{I}^j} \int_{\mathcal{S}_i} \|q - p_i\|^2 \phi^j(q, t) dq.$$

To that end, let us then consider a (partially) distributed computation given by

$$\frac{\partial J(p^j, t)}{\partial p_i} = 2\text{mass}(\mathcal{S}_i(p^j))(\text{cent}(\mathcal{S}_i(p^j)) - p_i) \quad (7)$$

where

$$\begin{aligned} \text{mass}(\mathcal{S}_i(p^j)) &:= \int_{\mathcal{S}_i} \phi^j(q, t) dq \\ \text{cent}(\mathcal{S}_i(p^j)) &:= \frac{1}{\text{mass}(\mathcal{S}_i(p^j))} \int_{\mathcal{S}_i} q \phi^j(q, t) dq \end{aligned} \quad (8)$$

During deployment, the initial position of each robot may be outside the designated region  $\mathcal{Q}^j$ . Thus, proportional control is introduced for the robot to navigate towards the centroid of  $\mathcal{Q}^j$  (i.e.,  $c^j$ ) until it enters the region  $\mathcal{Q}^j$ . To summarize, the computation of  $u_i^{\text{cov}}$  for  $i \in \mathcal{I}^j$  is given by

$$u_i^{\text{cov}} = \begin{cases} \frac{\partial J(p^j, t)}{\partial p_i}, & \text{if } p_i \in \mathcal{Q}^j \\ \gamma(c^j - p_i) & \text{otherwise} \end{cases}$$

where  $\gamma > 0$  is the proportional gain.

Additionally, the collision avoidance and actuator limit can be explicitly incorporated by utilizing quadratic programming with control barrier function [6], [33], [34] and additional constraint as follows

$$\begin{aligned} u_i &= \arg \min_{u_i} \|u_i^{\text{cov}} - u_i\|^2 \\ \text{s.t. } &\left( \frac{\partial h_o(p_i, p_j)}{\partial x_i} \right)^T u_i \geq -\gamma h_o(p_i, p_j), \quad \forall j \in \mathcal{N}_i^s \\ &u_i^{\text{lb}} \leq u_i \leq u_i^{\text{ub}}. \end{aligned}$$

given that

$$\begin{aligned} h_o(p_i, p_j) &:= \|p_i - p_j\|^2 - D^2 \geq 0, \\ \mathcal{N}_i^s &:= \{j, j \neq i \mid \|x_i - x_j\|^2 \leq (2R_i)^2\}. \end{aligned}$$

Here,  $D$  is the considered collision distance,  $\gamma > 0$  is the obstacle avoidance gain, and  $u_i^{\text{lb}}$  and  $u_i^{\text{ub}}$  are the lower- and upper-bound of the actuator range.

Finally, in practice, the computation of the density update in (6) will be performed by a central system for each region  $j$ , since each robot hardly knows if other robots have visited each  $q \in \mathcal{Q}^j$ . On the other hand, provided the information of  $\mathcal{Q}^j$ ,  $c^j$ , and  $\phi^j(q, t)$  in  $\mathcal{S}_i$  from region  $j$ 's central system, each robot  $i \in \mathcal{I}^j$  can distributively compute  $u_i$ .

## VI. NUMERICAL SIMULATIONS AND DISCUSSION

In this subsection, we demonstrate and evaluate the proposed two-layer control framework in simulations. Particularly, we are interested in investigating the effects of selecting the number of regions to be generated on the overall performance of the monitoring mission.

### A. SIMULATION SETUP

Throughout the simulation, we consider a field  $\mathcal{F}$  of size  $[-0.5, 19.5] \text{ m} \times [-0.5, 19.5] \text{ m}$  discretized into 400 grids with grid size  $D_s = 1 \text{ m}$ . Here, we consider 20% of the grids are targeted grids, i.e.,  $m = 80$ , and we prepare 2 cases of targeted grid distribution, which are 1) uniform distribution throughout  $\mathcal{F}$  denoted by  $\mathcal{H}^u$ , and 2) clustered distribution towards 4 corners of the field  $\mathcal{F}$  denoted by  $\mathcal{H}^c$ . Then, we consider that the number of robots to be deployed is  $n = 6$  with 2 different sets of sensing capabilities (homogeneous and heterogeneous) as shown in Table 1.

In total, we investigate these 3 scenarios:

A.  $\mathcal{H}^u$  (uniform distribution) with homogeneous robots

**TABLE 1.** The sets of homogeneous and heterogeneous sensing capabilities of the robots.

$i \in \mathcal{I}$	1	2	3	4	5	6
Homogeneous						
$R_i$ [m]	2.11	2.11	2.11	2.11	2.11	2.11
$A(\mathcal{B}_i)$ [m <sup>2</sup> ]	14	14	14	14	14	14
Heterogeneous						
$R_i$ [m]	2.39	2.19	1.78	1.69	2.11	2.03
$A(\mathcal{B}_i)$ [m <sup>2</sup> ]	18	15	10	9	14	13

**TABLE 2.** The simulation parameter values.

Parameter for M-ACO	Symbol	Value
Number of ants	$N$	10
Number of iterations (tours)	$M$	1000
Initial pheromone value / strength	$P(0)$	1
Pheromone evaporation coefficient	$\rho$	0.2
Importance of pheromone value	$\alpha$	1
Importance of heuristic value	$\beta$	1
Parameter for Density Map & Controller	Symbol	Value
Minimum value for density map	$c_f$	0.3
Gaussian function's gain	$k$	0.433
Gaussian function's covariance matrix	$\Sigma$	$(0.5)^2 \mathbf{I}_2$
Information sensing rate	$\underline{\delta}$	2.5
Information decay rate	$\bar{\delta}$	0.06
Collision distance (CBF)	$D$	0.15m
Obstacle avoidance gain (CBF)	$\gamma$	10
actuator limit	$-u_i^{lb} = u_i^{ub}$	1m/s

- B.  $\mathcal{H}^c$  (clustered distribution) with homogeneous robots  
 C.  $\mathcal{H}^c$  (clustered distribution) with heterogeneous robots

Additionally, we prepare 50 sets of random initial positions for the 6 robots within the field  $\mathcal{F}$ . Given that the number of the generated region should not exceed the number of robots, i.e.,  $\ell \leq n$ , we simulate each scenario for all  $\ell \in \{1, \dots, 6\}$ . Thus, we generate and analyze 300 simulation data for each scenario.

The parameter values for both the task allocation and persistent monitoring algorithms are summarized in Table 2. For the M-ACO algorithm, we utilized these values based on [24]. The number of ants corresponds to the number of computational units to try some possible region-robot pairing based on the pheromone and heuristic values (see (2)). Here, we utilized 10 ants where in every iteration, each ant paired all regions with at least one robot. At the end of each iteration, the pheromone value is updated according to (4) before the next iteration begins. When the iterations reach 1000, the best region-robot pairing is then decided. Note that the initial pheromone value / strength was set to 1, which means that all region-robot pairs were given the same pheromone strength of 1, before being updated in each iteration. Pairs that are not chosen by any ants will evaporate with a pheromone evaporation coefficient of 0.2. In this work, we give the same importance to the pheromone value and heuristic value. Thus, both parameter values are set to 1. For the modelling area of importance, we use the same parameter as the example in Fig. 3b as this parameter generates steady slopes for the gaps between targeted grids (as compared to Fig. 3a).

**TABLE 3.** Sizes of partitioned coverage holes (in m<sup>2</sup>) from K-means clustering.

$\mathcal{H}^u$ (Uniform Distribution)							
	1	2	3	4	5	6	Total
$\ell = 1$	377.						377.
$\ell = 2$	187.	165.5					352.5
$\ell = 3$	117.5	87.5	119.				324.
$\ell = 4$	62.	97.5	95.5	59.			314.
$\ell = 5$	51.	42.5	62.	76.5	52.5		284.5
$\ell = 6$	36.	44.	50.5	64.5	44.5	37.	276.5
$\mathcal{H}^c$ (Clustered Distribution)							
	1	2	3	4	5	6	Total
$\ell = 1$	372.						372.
$\ell = 2$	151.	110.5					261.5
$\ell = 3$	35.	110.5	46.				191.5
$\ell = 4$	35.	37.	46.	34.5			152.5
$\ell = 5$	39.	19.5	35.	34.5	23.5		151.5
$\ell = 6$	37.	35.	24.5	21.	11.5	9.	138.

**Remark:** For the deployment in other scenarios, Table 2 can be used as a reference for the selection of the parameter values. For the M-ACO algorithm, the same sets of parameter can be used irrespective of the number of robots (including their sensing range) and regions (clusters). The modelling area of importance can use the same  $c_f$  and  $k$  while the  $\Sigma$  can be scaled linearly to a different grid size. Otherwise, section V-A provides guidelines to compute these parameters. For the persistent monitoring algorithm, the  $\underline{\delta}$  and  $\bar{\delta}$  are dictated by the specification on how long the robot needs to stay in order to gather the information and how often the information needs to be revisited. The  $D$ ,  $u_i^{lb}$  and  $u_i^{ub}$  rely on the robot's physical features, while the  $\gamma$  can be used as it is or adjusted if more reactive response is needed by making the gain smaller.

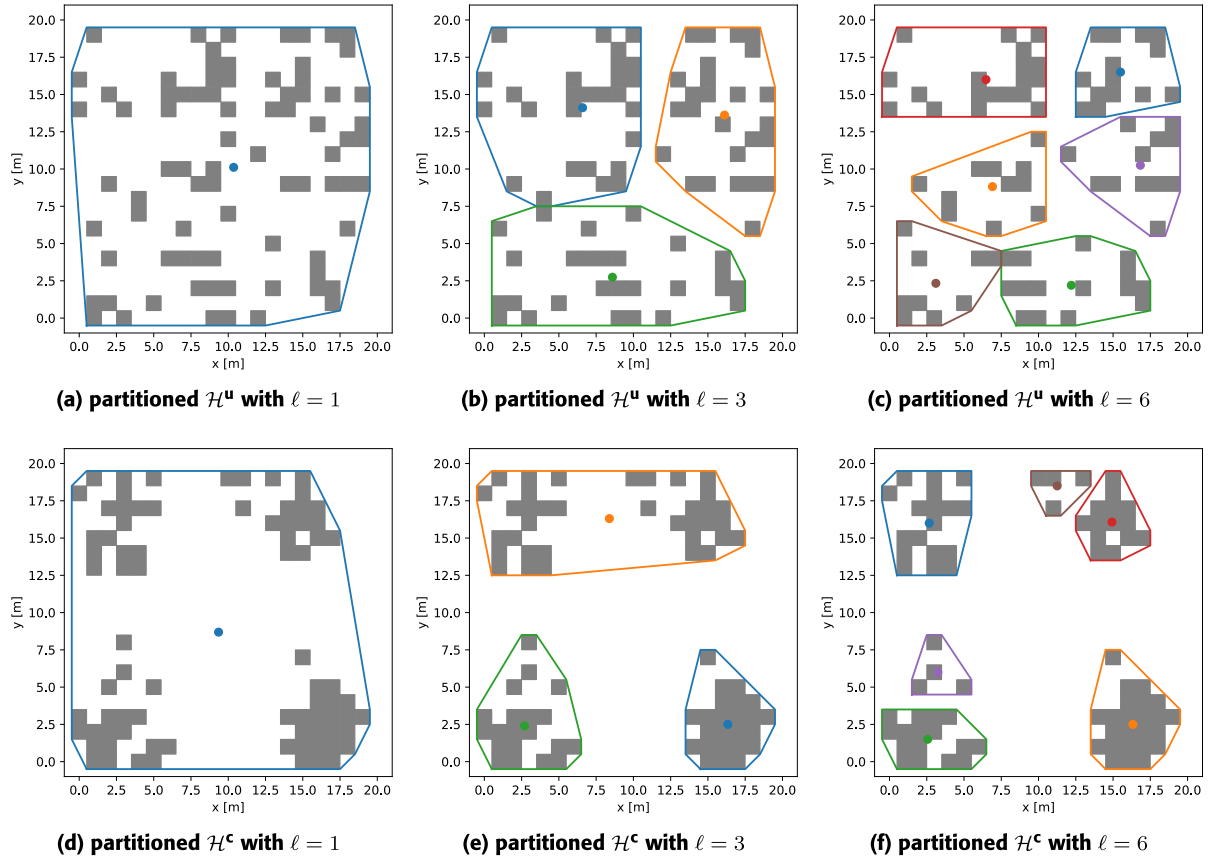
## B. EVALUATION OF TASK ALLOCATION STRATEGY

In this subsection, we first evaluate each step in task allocation strategy, namely K-means clustering and Modified Ant-Colony Optimization (M-ACO) algorithm, for 300 simulation data at each of the 3 scenarios.

### 1) DIVIDED REGIONS FROM K-MEANS CLUSTERING

As described in subsection IV-A, the procedure of K-means clustering requires the input of the desired cluster, i.e.,  $\ell$ , and the initial value of the centroids of the cluster. Within our test, we generate the initial values of the centroids randomly with a fixed seed. The fixed seed minimizes the variation from K-means clustering and results in a consistent shape of clustered regions for each  $\ell$  value. Examples of the resulting clustered regions are shown in Fig. 5 for both  $\mathcal{H}^u$  (uniform distribution) and  $\mathcal{H}^c$  (clustered distribution). Note that the K-means clustering does not take account of the robots' initial position and sensing capabilities, and thus there is no difference in the results for homogeneous or heterogeneous robots. The detailed values of the area for each region after K-means clustering are summarized in Table 3.





**FIGURE 5.** Results of K-mean clustering for targeted grids that is generated with uniform distribution ( $\mathcal{H}^u$ ) and clustered distribution ( $\mathcal{H}^c$ ). The colored polygons and circles denote the resulting clustered regions and its centroid, respectively. The color indexes for region 1 to 6 are blue, orange, green, red, purple, and brown.

From Table 3, we observe that with the increase of  $\ell$ , the reduction of the total area of the generated region is more significant on  $\mathcal{H}^c$  compared to  $\mathcal{H}^u$ . Moreover, as  $\mathcal{H}^c$  is intentionally generated based on 4 clustered distributions, we can observe that the sum of the regions size is significantly reduced towards  $\ell = 4$  and with reduced improvement for  $\ell = 5$  onward. Thus, the K-means clustering helps in condensing the area to be monitored, but has the most impact on targeted grids whose distribution is more clustered.

## 2) TASK ALLOCATION PERFORMANCE FROM M-ACO ALGORITHM

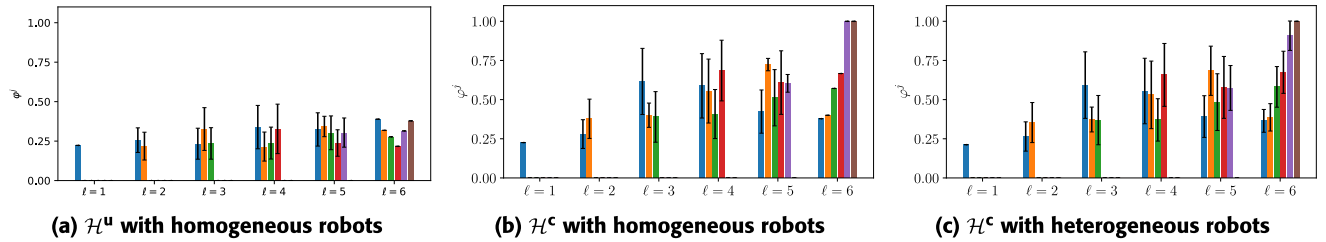
Next, we evaluate the task allocation performance of the M-ACO algorithm. Here, we define the metric to evaluate the task allocation as the occupancy ratio between the summation of the capabilities from the allocated robots versus the region  $j$ 's size as

$$\varphi^j = \min \left\{ 1, \frac{\sum_{k \in \mathcal{J}} A(\mathcal{B}_k)}{A(Q^j)} \right\}$$

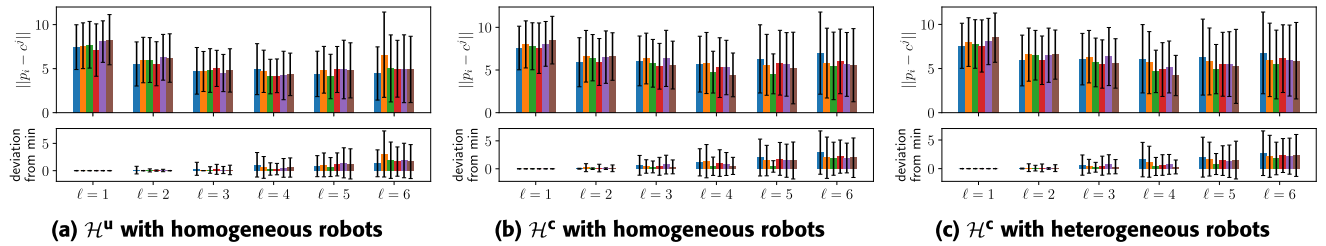
The case where  $\sum_{k \in \mathcal{J}} A(\mathcal{B}_k)/A(Q^j) > 1$  denotes the condition where the sum of the allocated robots' capabilities exceeds the area to be monitored. Thus, we capped the value

to 1 using the minimization function. For each scenario at each value of  $\ell$ , we compute the mean  $\pm$  variance of  $\varphi^j$  for all 50 simulations with different robots' initial positions. The resulting graph for the 3 scenarios is shown in Fig. 6. Additionally, we evaluate the distances of each robot to the allocated region's centroid. The mean  $\pm$  variance of the initial distances and also the deviation from the true minimum, i.e., the shortest distance of each robot to any region's centroid, are shown in Fig. 7.

Note that the generated regions from K-means clustering is consistent for each  $\ell$  value in each scenario. Thus, the resulting variations of  $\varphi^j$  on Fig. 6 are solely the results from the M-ACO algorithms when allocating the robots to each region. Observing the result on scenario  $\mathcal{H}^u$  (uniform distribution) with homogeneous robots in Fig. 6a, we can observe that the ratios  $\varphi^j$  are relatively consistent for all regions on each different value of  $\ell$ . Specific to  $\ell = 6$ , the M-ACO algorithm is reduced to one-to-one allocation of robot-region based on the shortest pairing distance. Due to the homogeneity of the sensing capabilities, the allocation ratio is consistent, i.e., the variance value is 0. A similar observation can be seen for the scenario  $\mathcal{H}^c$  (clustered distribution) with homogeneous robots in Fig. 6b. In comparison to Fig. 6a, the value of the ratio is slightly larger for  $\ell \geq 3$  due to the smaller



**FIGURE 6.** Occupancy ratios (mean  $\pm$  variance) between the allocated robot's sensing capabilities versus the region's size, i.e.  $\phi^j$ , for 50 simulations on each scenario. A larger value of  $\phi^j$  (with a maximum of one) means better occupancy to region  $j$ . The color indexes for region  $j = 1$  to 6 are blue, orange, green, red, purple, and brown.



**FIGURE 7.** The upper figure describes the distance (mean  $\pm$  variance) between the robot's initial position to the allocated region's centroid, i.e.  $\|p_i - c^j\|$ ,  $i \in \mathcal{D}^i$ ,  $j \in \mathcal{C}$ . The lower figure describes the deviation (mean  $\pm$  variance) of  $\|p_i - c^j\|$  to the minimum distance of robot  $i$  to any region's centroid. The figures are results from 50 simulations on each scenario where the color indexes for robot  $i = 1$  to 6 are blue, orange, green, red, purple, and brown.

sizes of region after K-means clustering in  $\mathcal{H}^c$  compared to  $\mathcal{H}^u$ . Finally, in the case that the sensing capability of the robots is heterogeneous as shown in Fig. 6c, the resulting means of the ratio  $\phi^j$  follows closely to the Fig. 6b with slightly larger variance.

From Fig. 7, in each scenario, we can observe a consistent distribution of mean and variance for each robot's distance to the allocated region's centroid, i.e.,  $\|p_i - c^j\|$ ,  $i \in \mathcal{D}^i$ ,  $j \in \mathcal{C}$ , suggesting optimal distribution in terms of distances. Additionally, we can also observe the deviation from the minimum distance pairings of a robot to any region's centroid. With  $\ell = 1$ , the only possible pairings are toward a single region, thus no deviation occurs. A larger deviation is introduced along the increase of  $\ell$ . This illustrates the trade-off in distance to ensure each region is allocated to at least one robot and also in a balanced allocation, as also illustrated by the coverage ratio  $\phi^j$ . Thus, with these observations, we can conclude that the M-ACO algorithms manage to provide an optimal allocation based on the robot's initial position and sensing capabilities.

### C. EVALUATION OF PERSISTENT MONITORING ALGORITHM

Finally, in this subsection, we evaluate the persistent monitoring algorithm after the resulting allocation from the task allocation strategy in the previous subsection. Here, we evaluate the monitoring performance throughout 120s of simulation duration, and we execute the simulation with a time sampling of 20ms. An example of the persistent monitoring algorithm in action is shown in Fig. 6 for the first 45 seconds.

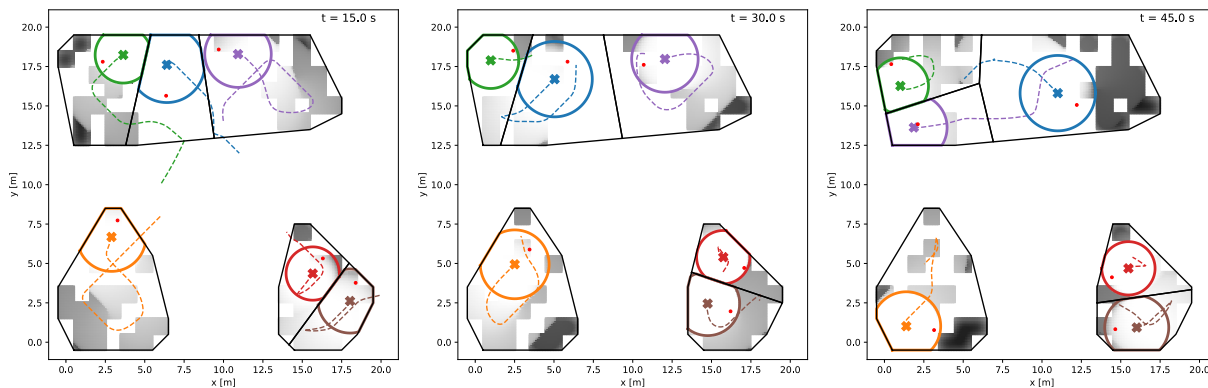
#### 1) COVERAGE PERFORMANCE

Here, we define the ratio of total density value at the time  $t$  compared to the modeled density value as

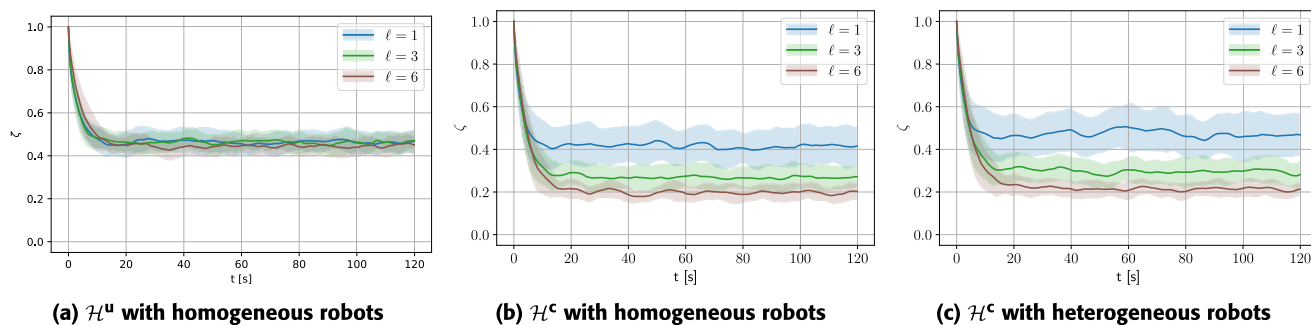
$$\zeta(t) := \frac{\int_{\mathcal{H}} \phi^j(q, t) dq}{\int_{\mathcal{H}} \hat{\phi}^j(q) dq}.$$

This value roughly denotes the performance of the coverage control algorithm over time, where the smaller the coverage ratio  $\zeta$  denotes the more often the overall targeted grids are being covered by any of the robots. For each scenario at each value of  $\ell$ , we compute the mean  $\pm$  variance of the coverage ratio  $\zeta$  over time for all 50 simulations with different robot initial positions. The resulting graph for the 3 scenarios and for  $\ell = \{1, 3, 6\}$  are shown in Fig. 9.

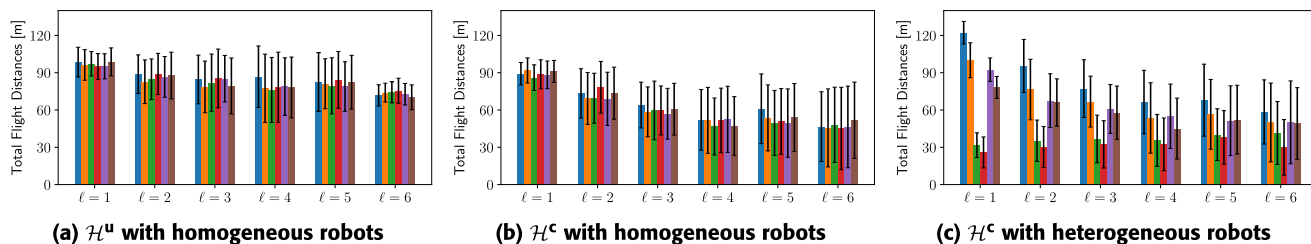
Through Fig. 9, we can observe that the coverage ratio is significantly reduced within the first 20s, and then the value remains oscillating within a bounded region, where we refer to the middle value of the bounded region as a steady value. Observing the result on scenario  $\mathcal{H}^u$  (uniform distribution) with homogeneous robots in Fig. 9a, the steady value remains similar throughout the various  $\ell$  values. On the other hand, for the scenario  $\mathcal{H}^c$  (clustered distribution) with homogeneous robots in Fig. 9b, the steady value is reduced with the increase of the  $\ell$  values. Then, for the case of heterogeneous robots in Fig. 9c, the results do not vary much from the homogeneous counterpart in Fig. 9b. Note that in the previous subsection, we observe that the K-means clustering provides a significant reduction of the sum of regions' size with the increase of  $\ell$  for the  $\mathcal{H}^c$  compared to  $\mathcal{H}^u$ . Thus, we conclude that the steady value is inversely proportional to the sum of the regions' size,



**FIGURE 8.** An example of the robots’ trajectory (dashed lines) and the changes in density map for the first 45s of the simulation for the scenario  $\mathcal{H}^c$  (clustered distribution) with heterogeneous robots and  $\ell = 3$ .



**FIGURE 9.** Coverage ratio (mean  $\pm$  variance) over the coverage hole  $\mathcal{H}$ , i.e.  $\zeta(t)$ , for 50 simulations on each scenario. A smaller value of  $\zeta$  indicates a better coverage performance.



**FIGURE 10.** Total travel distance (mean  $\pm$  variance) for each robot for 50 simulations on each scenario. The color indexes for robots 1 to 6 are blue, orange, green, red, purple, and brown.

meaning that the coverage performance is increased as the region’s size is smaller.

2) TOTAL TRAVELING DISTANCES

Last, we investigate the total travel distance for each robot. For each scenario at each value of  $\ell$ , we compute the mean total travel distance for all 50 simulations with different robot initial positions. The resulting graph for the 3 scenarios is shown in Fig. 10. Here, we observe one more time that the increase of  $\ell$  results in a larger decrease in robots’ travel distance for  $\mathcal{H}^c$  (Fig. 10b) compared to  $\mathcal{H}^u$  (Fig. 10a). Note that despite the similar steady value of coverage ratio throughout various  $\ell$  value  $\mathcal{H}^u$ , we can observe a small reduction of the final travel distances as  $\ell$  increases. Finally,

by comparing the travel distance for the heterogeneous case (Fig. 10c) versus the homogeneous case (Fig. 10b), we can observe a larger variation of the final travel distance in the heterogeneous case. Here, the robot with a larger sensing radius results in a larger movement gain (mass) as depicted in the equation (7) and (8), thus resulting with further final travel distance. However, the general trend that the increase of the  $\ell$  results in a decrease in robots’ travel distance remains.

VII. CONCLUSION AND FUTURE WORK

This paper presents a novel two-layer control framework for a robotic sensor network to persistently monitor targeted grids in a given field with an improved coverage performance and reduced traveling distance of the robots. The first layer

(task allocation strategy) encapsulates the targeted grids into a set of tasks and optimally allocates the robots based on their initial position and sensing capabilities, while the second layer (persistent monitoring algorithm) generates each robot's control input to ensure persistent monitoring over the designated task area. Throughout our numerical simulation, we can conclude that dividing the originally targeted grids into several smaller regions helps in improving persistent monitoring in terms of coverage performance and travel distance. Moreover, this benefit has more impact on targeted grids where the distribution is more clustered than the uniform case. However, as evident from the analysis, dividing into too many clusters will increase the travel distance from/to the base station as the number of robots that can be allocated to a cluster decreases. Hence, there is a trade-off between the system's performances, which needs to be considered in choosing the number of clusters.

To this end, several limitations persist in the proposed framework. The K-means clustering method requires an explicit number of regions to be generated. Moreover, the proposed framework neglects the issue of energy limitation and is not fully adaptive to environmental changes in the monitoring regions, as the task allocation strategy is limited to offline computation pre-deployment of the monitoring mission. Nonetheless, our proposed framework is modular in the sense that each approach can be independently replaced if improvements to the mission performance are needed. To alleviate these limitations, our future works will investigate: a clustering method that optimizes some metrics to remove the need to specify the number of clusters; an online task allocation algorithm that allows adaptation to changes in the monitored environment; energy management that accommodates charging scheduling of the robots; and implementation of the proposed control framework in a real world experiment (e.g., via crazyflie nano-quadcopters in our networked robotic laboratory) to investigate the possible deviations from the current simulation results.

## REFERENCES

- [1] J. J. Roldán, J. D. Cerro, and A. Barrientos, "Should we compete or should we cooperate? Applying game theory to task allocation in drone swarms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5366–5371.
- [2] L. García, L. Parra, J. M. Jimenez, J. Lloret, and P. Lorenz, "IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture," *Sensors*, vol. 20, no. 4, p. 1042, Feb. 2020.
- [3] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present, and future [point of view]," *Proc. IEEE*, vol. 109, no. 7, pp. 1152–1165, Jul. 2021.
- [4] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 626–645, Feb. 2022.
- [5] A. M. Annaswamy, K. H. Johansson, and G. J. Pappas, Eds., *Control for Societal-Scale Challenges: Road Map 2030*. Orlando, FL, USA: IEEE Control Systems Society Publication, 2023.
- [6] M. Egerstedt, *Robot Ecology: Constraint-Based Design for Long-Duration Autonomy*, 1st ed. Princeton, NJ, USA: Princeton Univ. Press, 2021.
- [7] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM, Control, Optim. Calculus Variat.*, vol. 11, no. 4, pp. 691–719, Oct. 2005.
- [8] J. Cortes and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE J. Control, Meas., Syst. Integr.*, vol. 10, no. 6, pp. 495–503, Nov. 2017.
- [9] H. Mahboubi and A. G. Aghdam, "Distributed deployment algorithms for coverage improvement in a network of wireless mobile sensors: Relocation by virtual force," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 4, pp. 736–748, Dec. 2017.
- [10] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 919–925, Apr. 2018.
- [11] J. Chen and P. Dames, "Distributed multi-target tracking for heterogeneous mobile sensing networks with limited field of views," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 9058–9064.
- [12] W. Bentz, T. Hoang, E. Bayasgalan, and D. Panagou, "Complete 3-D dynamic coverage in energy-constrained multi-UAUV sensor networks," *Auto. Robots*, vol. 42, no. 4, pp. 825–851, Apr. 2018.
- [13] R. Funada, M. Santos, T. Gencho, J. Yamauchi, M. Fujita, and M. Egerstedt, "Visual coverage maintenance for quadcopters using nonsmooth barrier functions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3255–3261.
- [14] M. Rudolph, S. Wilson, and M. Egerstedt, "Range limited coverage control using air-ground multi-robot teams," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 3525–3530.
- [15] J. G. Martin, T. Hatanaka, J. M. Maestre, and E. F. Camacho, "Persistent coverage in non-convex environments with heterogeneous robotic networks: Constraint-based approach," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 10708–10714, 2023.
- [16] D. Panagou, D. M. Stipanovic, and P. G. Voulgaris, "Distributed dynamic coverage and avoidance control under anisotropic sensing," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 4, pp. 850–862, Dec. 2017.
- [17] Y. Diaz-Mercado, S. G. Lee, and M. Egerstedt, "Human-swarm interactions via coverage of time-varying densities," in *Trends Control and Decision-Making for Human-Robot Collaboration Systems*, Y. Wang and F. Zhang, Eds. Cham, Switzerland: Springer, 2017, pp. 357–385.
- [18] B. Haydon, K. D. Mishra, P. Keyantuo, D. Panagou, F. Chow, S. Moura, and C. Vermillion, "Dynamic coverage meets regret: Unifying two control performance measures for mobile agents in spatiotemporally varying environments," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 521–526.
- [19] M. Suenaga, T. Shimizu, T. Hatanaka, K. Uto, M. Mammarella, and F. Dabbene, "Experimental study on angle-aware coverage control with application to 3-D visual map reconstruction," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2022, pp. 327–333.
- [20] K. Mizuta, Y. Hirohata, J. Yamauchi, and M. Fujita, "Safe persistent coverage control with control barrier functions based on sparse Bayesian learning," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2022, pp. 311–318.
- [21] H. Dan, T. Hatanaka, J. Yamauchi, T. Shimizu, and M. Fujita, "Persistent object search and surveillance control with safety certificates for drone networks based on control barrier functions," *Frontiers Robot. AI*, vol. 8, Oct. 2021, Art. no. 740460.
- [22] A. Gusrialdi and C. Yu, "Exploiting the use of information to improve coverage performance of robotic sensor networks," *IET Control Theory Appl.*, vol. 8, no. 13, pp. 1270–1283, Sep. 2014.
- [23] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook Metaheuristics* (International Series in Operations Research & Management Science), M. Gendreau and J.-Y. Potvin, Eds. Cham, Switzerland: Springer, 2019, pp. 311–351.
- [24] L. Wang, Z. Wang, S. Hu, and L. Liu, "Ant colony optimization for task allocation in multi-agent systems," *China Commun.*, vol. 10, no. 3, pp. 125–132, Mar. 2013.
- [25] F. Zitouni, S. Harous, and R. Maamri, "A distributed approach to the multi-robot task allocation problem using the consensus-based bundle algorithm and ant colony system," *IEEE Access*, vol. 8, pp. 27479–27494, 2020.
- [26] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *J. Ambient Intell. Humanized Comput.*, pp. 1–12, Oct. 2020.
- [27] N. Yi, J. Xu, L. Yan, and L. Huang, "Task optimization and scheduling of distributed cyber-physical system based on improved ant colony algorithm," *Future Gener. Comput. Syst.*, vol. 109, pp. 134–148, Aug. 2020.



- [28] L. Chen, W.-L. Liu, and J. Zhong, "An efficient multi-objective ant colony optimization for task allocation of heterogeneous unmanned aerial vehicles," *J. Comput. Sci.*, vol. 58, Feb. 2022, Art. no. 101545.
- [29] Z. Wu and Z. Wu, "An enhanced regularized k-means type clustering algorithm with adaptive weights," *IEEE Access*, vol. 8, pp. 31171–31179, 2020.
- [30] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Inf. Sci.*, vol. 622, pp. 178–210, Apr. 2023.
- [31] K. Priandana, M. W. S. Atman, and A. Gusrialdi, "Persistent monitoring for indoor farming using static and mobile sensors," in *Proc. IFAC-PapersOnLine*, 2022, pp. 129–134.
- [32] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 3420–3431.
- [33] G. Notomista and M. Egerstedt, "Persistification of robotic tasks," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 756–767, Mar. 2021.
- [34] T. Ibuki, T. Hirano, R. Funada, and M. Sampei, "Optimization-based distributed safety control with applications to collision avoidance for mobile robotic networks," *Adv. Robot.*, vol. 37, nos. 1–2, pp. 87–98, Jan. 2023.
- [35] P. Singh and Y.-C. Chen, "Sensing coverage hole identification and coverage hole healing methods for wireless sensor networks," *Wireless Netw.*, vol. 26, no. 3, pp. 2223–2239, Apr. 2020.
- [36] F. Wang and H. Hu, "Coverage hole detection method of wireless sensor network based on clustering algorithm," *Measurement*, vol. 179, Jul. 2021, Art. no. 109449.
- [37] H. Huang, A. V. Savkin, M. Ding, and C. Huang, "Mobile robots in wireless sensor networks: A survey on tasks," *Comput. Netw.*, vol. 148, pp. 1–19, Jan. 2019.
- [38] A. Pang, F. Chao, H. Zhou, and J. Zhang, "The method of data collection based on multiple mobile nodes for wireless sensor network," *IEEE Access*, vol. 8, pp. 14704–14713, 2020.
- [39] N. Temene, C. Sergiou, C. Georgiou, and V. Vassiliou, "A survey on mobility in wireless sensor networks," *Ad Hoc Netw.*, vol. 125, Feb. 2022, Art. no. 102726.



MADE WIDHI SURYA ATMAN (Member, IEEE) received the B.Eng. and M.Sc. degrees in electrical engineering from Institut Teknologi Bandung, Indonesia, in 2011 and 2014, respectively, and the M.Eng. degree in mechanical and control engineering and the D.Eng. degree in systems and control engineering from the Tokyo Institute of Technology, in 2017 and 2020, respectively. Since 2020, he has been a Postdoctoral Research Fellow of automation technology and mechanical engineering with Tampere University. His research interests include human-swarm interaction, passivity-based control, and distributed control of networked systems.



MUHAMAD RAUSYAN FIKRI (Member, IEEE) received the B.Sc. degree in electronics and instrumentation from Universitas Gadjah Mada, Indonesia, in 2015, and the M.Eng. degree in systems and control engineering from the Tokyo Institute of Technology, in 2018. He is currently pursuing the Ph.D. degree in automation technology and mechanical engineering with Tampere University. His research interests include networked control systems, distributed optimization, and robotics.



KARLISA PRIANDANA (Senior Member, IEEE) was born in Bandung, Indonesia, in 1985. She received the bachelor's degree from the Department of Electrical Engineering, Bandung Institute of Technology, Indonesia, in 2008, the M.Eng. degree from the Department of International Development Engineering, Tokyo Institute of Technology, Japan, in 2010, and the Ph.D. degree from the Department of Electrical Engineering, Universitas Indonesia, in 2017.

She is an Associate Professor with the Department of Computer Science, IPB University, Indonesia, where she has been a Faculty Member, since 2012. She was the Secretary of the Department of Computer Science, from 2017 to 2021. Currently, she is the Assistant Director of research strategy and development with IPB University. Her research interests include robotics and artificial intelligence.

Dr. Priandana received the Best Student Award from the Bandung Institute of Technology, in 2006. She received full scholarships for a short-term research program with TU Delft, The Netherlands, in 2008; for the master's study from the Tokyo Institute of Technology, from 2008 to 2010; and for the doctoral study from Universitas Indonesia, from 2014 to 2017. She was the Best Graduate of the Bandung Institute of Technology, in 2008; and the Best Doctoral Graduate of Universitas Indonesia, in 2017. She has been appointed as a Deputy Treasurer, a Treasurer, and the Deputy Secretary of IEEE Indonesia Section, in 2017, 2018, and 2021–2022, respectively.



AZWIRMAN GUSRIALDI (Member, IEEE) received the B.Eng. and M.Eng. degrees in mechanical and control engineering from the Tokyo Institute of Technology, Japan, in 2006 and 2008, respectively, and the Dr.-Ing. degree in control engineering from Technische Universität München, Germany, in 2012. He was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of Central Florida (UCF), Orlando. Since 2019, he has been an Assistant Professor with the Automation Technology and Mechanical Engineering Unit, Tampere University, Finland, and leading the Intelligent Networked Systems Group. His research interests include the design of resilient networked systems, cooperative control, and distributed optimization for networked cyber-physical systems.

...