

# Distributed persistent coverage control and performance evaluation of multi-agent system

S. Lee 

lsw7169@snu.ac.kr

Y. Kim 

ydkim@snu.ac.kr

Seoul National University

Department of Mechanical and Aerospace Engineering

Institute of Advanced Aerospace Technology

Seoul, Republic of Korea

## ABSTRACT

The persistent coverage control problem is formulated based on cell discretisation of two-dimensional mission space and time-increasing cell ages. A new performance function is defined to represent the coverage level of the mission space, and time behaviour is evaluated by the probabilistic method based on the detection model of agents. For comparison, persistent coverage controllers are designed by a target-based approach and a reactive approach. Both controllers are designed in a distributed manner using Voronoi tessellation and Delaunay graph-based local information sharing. Numerical simulation is performed to analyse the evaluated mean age of cells and evaluated coverage level over time for the designed persistent coverage controllers. The differences between the evaluation model and simulation situation are discussed.

**Keywords:** Coverage control, Multi-agent system, Distributed system, Performance evaluation

## NOMENCLATURE

$a_i$	age of cell $i$
$a_s$	average age of cells in steady state
$\hat{a}_i^j$	evaluated age of cell $i$ predicted by agent $j$
$\bar{a}(t)$	evaluated average age of cells
$\mathcal{A}$	age vector

---

Received 23 April 2018; revised 2 November 2018; accepted 31 January 2019.

A version of this paper was presented at the 31<sup>st</sup> ICAS Congress of the International Council of the Aeronautical Sciences in Belo Horizonte, Brazil in September 2018.

$A_j$	evaluated age vector predicted by agent $j$
$c_i$	$i$ -th cell
$C$	performance function
$C_s$	performance function value in steady state
$C_j$	center of mass of $V_j$
$\mathcal{D}$	set of cells inside of the sensing ranges of the agents
$\mathcal{D}_j$	set of cells inside of the sensing ranges of the agent $j$
$\mathbb{E}(\cdot)$	expected value
$\mathcal{E}$	edges of Delaunay graph
$\mathcal{G}$	Delaunay graph
$L_j$	first moment of $V_j$
$M_j$	mass of $V_j$
$N(=n^2)$	number of cells in mission space
$P_{c_i}$	position of cell $i$
$P_j = [x_j, y_j]^T$	position of agent $j$
$r_j$	sensing radius of agent $j$
$t_j$	target cell of agent $j$
$T_j$	position of target cell of agent $j$
$u_j$	control input of agent $j$
$V$	set of Voronoi region
$V_j$	Voronoi region of agent $j$
$\mathcal{V}$	vertices of Delaunay graph
$\mathcal{V}_j$	set of neighbour agents of agent $j$

## Greek Symbols

$\Omega$	mission space
$\mu$	speed of agent
$\mu_c$	speed of agent with respect to the number of cells
$\rho$	scaling constant
$\theta_j$	heading angle of agent $j$

## 1.0 INTRODUCTION

Various guidance and control algorithms for multi-agent systems have been developed according to technological advances in hardware and software. In a multi-agent system, the agents, which are unmanned aerial vehicles (UAVs) or autonomous ground mobile robots, conduct a given mission while maintaining a certain distance from each other. Therefore, a centralised headquarters is widely considered for commanding individual agents; alternatively, each agent decides its own action based on local information. The former is often called a centralised system, whereas the latter is called a distributed system<sup>(1)</sup>. In a centralised system, the computation cost is heavy because the state information of the whole agents should be considered simultaneously when determining the actions, and practical issues, including communication between the agent and headquarters, exist. By contrast, in a distributed system, much less information is used than in a centralised system because the distributed system utilises only the local information of neighbouring agents. The practical issue in a distributed system is that high performance of the on-board computer of each agent is required to run various decision algorithms.

The coverage control problem is a typical and important problem in the multi-agent system, which may be designed in a centralised or distributed manner<sup>(2,3)</sup>. The coverage control problem can be understood as deciding the positions (nodes) of the robot network for monitoring the space of interest. The coverage control problem is often referred to as the active sampling problem or static coverage problem. In the coverage control problem, the positions of the monitoring robots are usually determined to maximise the information level that can be achieved and minimise the communication costs between robots and the energy consumption. Therefore, the coverage control problem is often formulated as an optimal control problem.

For the mission space that cannot be fully covered by stationary robot networks, i.e. traditional coverage control methods, the persistent coverage control problem has been formulated and solved<sup>(4,5)</sup>. A persistent coverage control problem is considered if a blind spot or dead zone should not exist in the mission space and the agents repeatedly patrol around the mission space. The persistent coverage problem can also be understood as an extended version of the dynamic coverage problem. In contrast to the coverage control problem, agents sweep the entire mission space without leaving any blind spots in the dynamic coverage problem. The dynamic coverage control scheme becomes a persistent coverage control scheme when perpetually conducted<sup>(6)</sup>. However, it is not necessary for every persistent coverage controller to be designed in this form.

The coverage control problem for networked robots has been extensively studied for decades<sup>(7)</sup>. In particular, decentralised and distributed algorithms have been developed that focus on utilising local information without depending on a centralised system<sup>(8,9)</sup>. The Voronoi partition and Delaunay graph are common tools for utilising the local information-sharing network of space-covering agents<sup>(3,10)</sup>. Based on the distributed method, the coverage control approach has been explored, and an adaptive coverage control approach has been developed<sup>(10,11)</sup>. In the adaptive coverage control approach, the distributed agents typically estimate the distributed function over the mission space based on linear combinations of the basis kernel function. By contrast, the dynamic coverage control problem considers a time-varying distribution function<sup>(12)</sup>. The receding horizon control approach has also been utilised to solve the optimal coverage control problem<sup>(3)</sup>.

The persistent coverage control problem usually considers the uncertainty level or decaying information, which is distributed in the mission space and represented by a dynamic function<sup>(13–15)</sup>. The dynamic equation of the uncertainty level in the persistent coverage control problem is typically modelled by using an age model that increases at a constant rate. Therefore, the persistent coverage control problem can be formulated as an optimal control problem<sup>(13,16)</sup>. In [14], a local gradient-based controller is designed for sustaining the desired uncertainty level of the mission space. Various persistent coverage control approaches have been developed<sup>(1)</sup>; however, performance evaluation for general persistent coverage controllers has not been conducted. Performance evaluation is to evaluate the performance of given persistent coverage controller based on the parameters of the algorithm. Only the optimal coverage performance in a limited problem formulation has been examined<sup>(17)</sup>.

In this study, the persistent coverage control problem is modelled, and distributed controllers are designed. The performance of the persistent coverage controllers is evaluated by probabilistic expected value analysis. It is shown that the proposed method for evaluating the performance of the persistent coverage control is valid for general persistent coverage control algorithms based on the age uncertainty model and the binary deterministic detection model. The monitoring space is discretised into cells having uncertainty, where the uncertainty is modelled based on the age model. The binary deterministic detection model is used for the detection model of monitoring agents to evaluate the coverage performance function. The

persistent coverage controller utilises only local information when determining its control input command, where the distributed controller is designed based on the Voronoi tessellation and Delaunay graph connecting the agents. Voronoi regions are calculated for the agents in the mission space, and local information sharing is conducted using the Delaunay graph network. Two distributed controllers, a target-based controller and a reactive controller, are designed in this study. The target cell is selected in each Voronoi region of the agents in the target-based controller. In the reactive controller, the control direction is computed based on the age of the cells inside the agent's sensing range.

The rest of this paper is organised as follows. Section 2.0 summarises the preliminaries for the formulation of the persistent coverage control problem. The agent model, mission space model, and performance function are also defined. The performance evaluation is performed in Section 3.0. Persistent coverage controllers are designed in Section 4.0, and the numerical simulation is performed in Section 5.0. Differences between the evaluation model and simulation situations are discussed and quantified. Section 6.0 presents the conclusions.

## 2.0 PRELIMINARIES

In this study, the monitoring space is discretised into cells, and the agent determines its action based on the local information in a distributed manner.

### 2.1 Agent model and mission space

The agent dynamics are described as follows:

$$\dot{x}_j = \mu \cos \theta_j \quad \dots (1)$$

$$\dot{y}_j = \mu \sin \theta_j \quad \dots (2)$$

$$\dot{\theta}_j = u_j \quad \dots (3)$$

where  $x_j, y_j$  are elements of agent position  $P_j = [x_j, y_j]^T$ ,  $\theta_j$  is a heading angle, and the heading angular acceleration  $u_j$  is the control input. It is assumed that the speed  $\mu$  of each agent is constant.

In this study, a uniformly discretised two-dimensional square-shaped mission space is considered, i.e.  $\Omega = [0, 1]_d^2 \subset \mathbb{R}_d^2$ . Subscript  $(\cdot)_d$  indicates the discretised space, i.e. the cells, and the number of cells in one side of the square is  $n$ . Thus, the total number of cells is  $N = n^2$  in  $\Omega$ . Each cell  $c_i$  has age  $a_i$ , which increases with time if the cell  $c_i$  is not detected by any agent.

The age  $a_i$  can be represented as

$$a_i(t) = \begin{cases} t - t_i & \text{if } c_i \notin \mathcal{D} \\ 0 & \text{if } c_i \in \mathcal{D} \end{cases} \quad \forall i = 1, \dots, N \quad \dots (4)$$

where  $t_i$  is the time when the cell  $c_i$  belongs to  $\mathcal{D}$ , and  $\mathcal{D}$  is the set of cells inside the sensing ranges of the agents. That is,

$$\mathcal{D} = \bigcup_{j=1}^M \mathcal{D}_j \quad \dots (5)$$

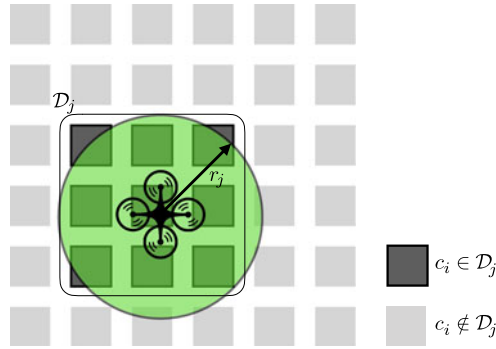


Figure 1. Sensing range description.

where  $\mathcal{D}_j$  is the set of cells inside the sensing range of agent  $j$ , and  $M$  is the number of agents in the mission space, i.e.

$$\mathcal{D}_j = \{c_i \mid d(P_{c_i}, P_j) \leq r_j\}, \quad i \in \Omega \quad \dots (6)$$

Note that  $d(\cdot, \cdot)$  is a distance operator, and Euclidean distance  $d(a, b) = \sqrt{(a - b)^T(a - b)}$  is used in this study.  $P_{c_i}$  and  $P_j$  are the positions of cell  $i$  and agent  $j$ , respectively, and  $r_j$  is the sensing radius of agent  $j$ . Figure 1 shows the sensing range of agent and set  $\mathcal{D}_j$ .

In this study, the deterministic binary detection model is adopted for the agent detection model. Utilising the binary detection model, the age  $a_i$  of a cell  $c_i \in \mathcal{D}$  is instantly initialised to zero. Although various coverage control algorithms adopt probabilistic detection models and incorporate the probabilistic detection model into uncertainty dynamics for the sake of convenience in deriving the Jacobian, it can be argued that adopting the binary detection model is reasonable because sensors usually acquire information very rapidly relative to the uncertainty or agent dynamics.

### 2.2 Voronoi tessellation and Delaunay graph

The Voronoi region  $V = \{V_1, \dots, V_M\}$  can be generated with agent positions  $P = \{P_1, \dots, P_M\}$ . The Voronoi region can be defined as follows:

$$V_j = \{s \in \Omega \mid d(s, p_j) \leq d(s, p_k), \forall k \neq j\} \quad \dots (7)$$

The Voronoi tessellation and Delaunay graph are common tools for designing distributed communication models of coverage control. Figure 2 shows the concept of the Voronoi tessellation and Delaunay graph. The Delaunay graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is generated by connecting the agents with adjacent Voronoi regions. The vertices of the graph are allocated to each agent  $\mathcal{V} = \{1, \dots, M\}$ , and  $\mathcal{E} \subset [\mathcal{V} \times \mathcal{V}]$  is the set of edges. When the agent pair  $(j, k) = (k, j) \in \mathcal{E}$  is the edge of the Delaunay graph  $\mathcal{G}$ , agents  $j$  and  $k$  share their local information, including agent position  $P_j, P_k$  and predicted age vector  $\mathcal{A}_j, \mathcal{A}_k$ , with each other.

Each agent should know the information of Voronoi neighbours, which are the agents in the adjacent Voronoi region, to compute the Voronoi region of the agent itself. By means of the communication between Voronoi neighbours, a distributed coverage algorithm can be designed. The detailed algorithm about the communication between Voronoi neighbours can be found in [18].

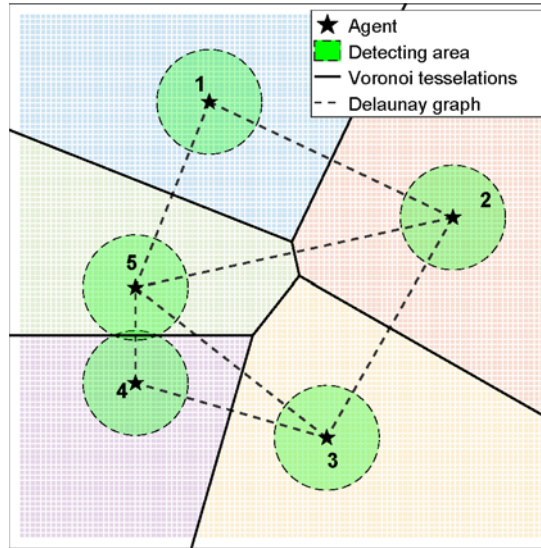


Figure 2. Concept of the Voronoi diagram and Delaunay graph.

The Voronoi tessellation-based space dividing is motivated from the approaches in the distributed coverage control problem. Despite that the agents share their local information only with other agents in the adjacent Voronoi region, the agents can acquire up-to-date information of the entire space. Therefore, a distributed algorithm, in the aspect of information sharing between agents, can be designed using the Voronoi tessellation. It is beneficial to utilise the Voronoi tessellation not only because a distributed coverage algorithm can be designed but also the collision between agents can be avoided<sup>(14)</sup>. The collision between agents in the monitoring space can be avoided because every agent moves toward a cell inside its Voronoi region, which is a convex set.

The neighbour set of agent  $j$  is defined as a set of connected agents denoted as  $\mathcal{V}_j \subset \mathcal{V}$ . By means of local information sharing using the Delaunay graph, agent  $j$  can compute its Voronoi region  $V_j$  according to the distributed method. In addition, collision avoidance between the agents can be achieved based on the Voronoi tessellation because the agent  $j$  determines its action (target position, velocity direction) inside its own Voronoi region  $V_j$ .

Using the increasing dynamics of  $a_i$ , each agent can predict the age  $a_i$  of each cell  $c_i$  and generate the predicted age vector  $\mathcal{A}_j = [\hat{a}_1^j(t), \dots, \hat{a}_N^j(t)]^T \in \mathbb{R}^N$ . Note that  $\hat{a}_i^j$  is the predicted age of cell  $i$  by agent  $j$ . Because agent  $j$  can acquire only local information, the real age vector  $\mathcal{A} = [a_1(t), \dots, a_N(t)]^T$  may differ from the predicted age vector of the individual agent.

### 2.3 Performance function

The objective of persistent coverage control with the binary detection model is to continuously monitor the mission space without any blind spots or dead areas. That is, every cell in the mission space should be detected by agents in finite time. To evaluate the performance of the persistent coverage control, the following performance function is defined.

$$C(t) = 1 - \frac{\sum_{i=1}^N \tanh(\rho a_i(t))}{N} \quad \dots (8)$$

where  $\rho \in \mathbb{R}^+$  is a positive scaling constant. In this study, a hyperbolic tangent function is used in the numerator, but one can use any function  $f(a_i(t))$  that can bound the age as  $f(a_i(t)) \in [0, 1]$ ,  $a_i(t) \geq 0$  to define the performance function in the range  $C(t) \in [0, 1]$ . If there does not exist any agent available in the mission space, then the numerator in the performance function converges to  $N$ , and  $C$  becomes zero, indicating a zero coverage situation. By contrast,  $C = 1$  implies the full coverage situation in which every cell in the mission space is detected by agents, i.e.  $c_i \in \mathcal{D}$ ,  $\forall i \in \{1, \dots, N\}$ .

### 3.0 ANALYSIS FOR PERFORMANCE EVALUATION

In this section, the time behaviour of the performance function is analysed based on the predefined age dynamics and agent detection model defined in Section 2.0 to evaluate the performance. Performance evaluation means that the performance of given persistent coverage controller is evaluated based on the parameters and specifications of the monitoring space and the sensors (agents), which include the discretisation of the space, speed, and sensing radius of the agents.

The performance evaluation is conducted based on the following assumptions.

- 1 There is no loss of detecting area.
- 2 The number of detected cells per time is constant.
- 3 Every cell in the mission space is evenly detected.

The following analysis is performed based on the above assumptions. The first assumption states that the number of cells inside the sensing range is consistent. The second assumption is related to the modelling of the monitoring space and the sensor. The third assumption is adopted for the probabilistic analysis of the performance prediction. Based on the third assumption, it can be assumed that the statistical probability of detection for a cell is equivalent to all cells inside the mission space.

#### 3.1 Steady-state analysis

It is assumed that the performance function converges to a steady-state value after the transition phase because the total increment of age is constant and there is a bound (limit) for the decrement amount. The total increment of ages over the mission space in time interval  $\Delta t$  can be written as

$$\Delta \mathbf{a}_\uparrow = (N - m)\Delta t \quad \dots (9)$$

where  $m (< N)$  is the number of cells in  $\mathcal{D} \subset \Omega$ , and  $m$  is assumed as constant in Assumption 1. In the steady state, the average age,  $a_s$ , of the cells being detected is constant. In addition, because the agents keep moving with constant speed  $\mu$  and continuously detect new cells, the score decrement of ages can be represented as follows:

$$\Delta \mathbf{a}_\downarrow = -a_s \mu_c m \Delta t \quad \dots (10)$$

where  $\mu_c$  is the speed of the agent with respect to the number of cells, not to the distance. That is,  $\mu_c$  is the number of cells passing by the agent moving with constant speed  $\mu$ . Because the mission space is discretised into square cells, the number of cells passing by may not be consistent with respect to the velocity direction even though  $\mu$  is constant. A hexagonal world

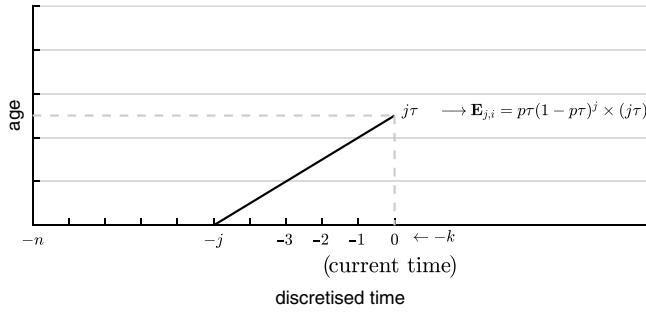


Figure 3. Span of time with timestep  $\tau$ .

representation may be helpful to alleviate this issue<sup>(19)</sup>. To satisfy Assumption 2,  $\mu_c$  should be constant. In this paper,  $\mu_c$  is approximated as follows:

$$\mu_c \approx \mu \sqrt{n} \tag{11}$$

The sum of the age increment and decrement becomes zero in the steady state because the performance function is steady. The average age of the cells in the steady state  $a_s$  can be computed by the following simple equation:

$$\Delta \mathbf{a}_\uparrow + \Delta \mathbf{a}_\downarrow = \Delta t((N - m) - a_s \mu_c m) \equiv 0 \tag{12}$$

$$a_s = \frac{1}{\mu_c} \left( \frac{N}{m} - 1 \right) \tag{13}$$

Substituting  $a_s$  into the performance function gives the predicted steady-state performance  $C_s$  as

$$\begin{aligned} C_s &= 1 - \frac{1}{N} \sum_{i \notin \mathcal{D}} \tanh(\rho a_s) \\ &= 1 - \left( 1 - \frac{m}{N} \right) \tanh \left( \frac{\rho}{\mu_c} \left( \frac{N}{m} - 1 \right) \right) \end{aligned} \tag{14}$$

### 3.2 Transient region analysis

For transient region analysis, a discretised span of time is considered. It is assumed that persistent coverage is guaranteed by monitoring the whole mission space  $\Omega$  without leaving any blind spots. It is also assumed that the coverage algorithm can permit the agent to evenly detect all cells unless there exist some preferred cells. At each timestep, let  $p$  be the probability of an individual cell that has been detected during the past one second. That is,  $i \in \mathcal{D}$  at least once during the arbitrary time interval  $[t, t + 1]$  when  $p = 1$ . Note that  $p$  is a constant based on Assumption 3. If the trial is conducted for every discretised time interval  $\tau$ , then the probability of each trial is  $p\tau$ .

The evaluated age of an individual cell  $\bar{a}_i(t) = \mathbb{E}(a_i)$  can be calculated by adding up the expected values of each trial. Figure 3 describes the discretised span of time and timesteps.



The expected value  $\mathbb{E}(a_i)$  at time  $t$  is the summation of the expected values  $\mathbb{E}_{k,i}$  at every  $k \in \{0, 1, \dots\}$  timestep, which can be represented as

$$\mathbb{E}(a_i) = \lim_{n \rightarrow \infty} \sum_{k=0}^n \mathbb{E}_{k,i} \quad \dots (15)$$

where

$$\begin{aligned} \mathbb{E}_{0,i} &= p\tau \times 0 \\ \mathbb{E}_{1,i} &= p\tau(1 - p\tau) \times \tau \\ \mathbb{E}_{2,i} &= p\tau(1 - p\tau)^2 \times 2\tau \\ &\vdots \\ \mathbb{E}_{n,i} &= p\tau(1 - p\tau)^n \times n\tau \end{aligned}$$

with

$$\tau = t/n. \quad \dots (16)$$

Note that  $\mathbb{E}_{k,i}$  is the expected value when the cell  $i \in \mathcal{D}$  at the  $k$ -th timestep and  $i \notin \mathcal{D}$  until the current timestep,  $k = 0$ . Simplifying the geometric series and taking the extreme value  $n \rightarrow \infty$ , we have

$$\begin{aligned} \bar{a}(t) = \mathbb{E}(a_i) &= \lim_{n \rightarrow \infty} \sum_{k=0}^n p\tau(1 - p\tau)^k k\tau \\ &= \lim_{n \rightarrow \infty} \left[ \frac{1}{p} \left( 1 - p\frac{t}{n} \right) \left( 1 - \left( 1 - p\frac{t}{n} \right)^n \right) - \left( 1 - p\frac{t}{n} \right)^{n+1} \right] \\ &= \frac{1}{p} - e^{-pt} \left( \frac{1}{p} + t \right) \quad \dots (17) \end{aligned}$$

Note that the subscript  $(\cdot)_i$  is omitted for the evaluated age  $\bar{a}(t)$  because it is assumed that the average value of age is considered not a certain age of the cell  $c_i$ . Now,  $p$  and  $\bar{a}_i(t)$  can be obtained using the steady-state value  $a_s$  as

$$a_s = \lim_{t \rightarrow \infty} \bar{a}(t) = \frac{1}{p} \quad \dots (18)$$

where

$$p = \frac{\mu_c m}{N - m} \quad \dots (19)$$

$$\bar{a}(t) = \frac{1}{\mu_c} \left( \frac{N}{m} - 1 \right) - e^{\mu_c m t / (N - m)} \left( \frac{1}{\mu_c} \left( \frac{N}{m} - 1 \right) + t \right) \quad \dots (20)$$

Substituting  $\bar{a}_i(t)$  into the performance function yields the predicted performance  $\bar{C}(t)$  as

$$\bar{C}(t) = 1 - \left(1 - \frac{m}{N}\right) \tanh(\rho \bar{a}(t)) \quad \dots (21)$$

Note that the predicted age and predicted performance are calculated based on the assumption that  $m$  is constant. The overlapping of detection ranges between the agents as well as the loss of detection ranges when detecting the boundary of the mission space is not considered in this analysis. That is,  $\mathcal{D}_j(t) \cap \mathcal{D}_k(t) = \emptyset$  and  $\mathcal{D}_j(t) \subset \Omega, \forall (j, k) \in \{1, \dots, N\}, \forall t$ . The predicted mean age and predicted performance are calculated as functions of agent speed, detection range combined with the number of agents, and the number of cells in the mission space.

## 4.0 PERSISTENT COVERAGE CONTROL APPROACH

In this study, two different persistent coverage control algorithms are considered to compare the performance of the algorithms using the predicted performance. Snapshots generated by conducting two designed persistent coverage control algorithms are shown in Figs. 4 and 5.

Note that the optimal persistent coverage control problem is an NP-hard problem, which is difficult to solve and time consuming. The target-based approach is one possible and easy heuristics to deal with this problem.

Meanwhile, the reactive approach is one of widely used methods in the coverage control problems. In the coverage control problem, the reactive controller calculates the gradient of a given weight function defined in the monitoring space. There is a difference between the coverage control problem and the persistent coverage control problem; that is, the weight function to calculate the gradient changes continuously. Therefore, the reactive controller in the persistent coverage problem makes the agent move persistently and does not converge to a particular position.

Distributed persistent coverage control algorithms are designed so that the individual agent uses only local information to achieve persistent coverage of the entire mission space. The local information for agent  $j$  consists of  $(P_j, \mathcal{A}_j)$ , and  $(P_k, \mathcal{A}_k)$  of neighbouring agents  $k \in \mathcal{V}_j$ . The information update rule for agent  $j$  is as follows:

$$\hat{a}_i^j \leftarrow \min(\hat{a}_i^j, \hat{a}_i^k), \quad \forall k \in \mathcal{V}_j, \quad \forall i \in \{1, \dots, N\} \quad \dots (22)$$

The minimum value of the predicted age is the most correct because the age is monotonically increasing; therefore, the cell with a smaller age represents the cell detected by the agent more recently.

### 4.1 Target-based approach

The target-based control approach guides agent  $j$  to a certain target cell,  $t_j$ . The cell with the maximum age inside the Voronoi region is determined as a target cell, and the agent is guided to position  $T_j$  of the target cell  $t_j$ .

$$t_j = \operatorname{argmax}_{c_i \in \mathcal{V}_j} (\hat{a}_i^j), \quad \hat{a}_i^j \in \mathcal{A}_j \quad \dots (23)$$

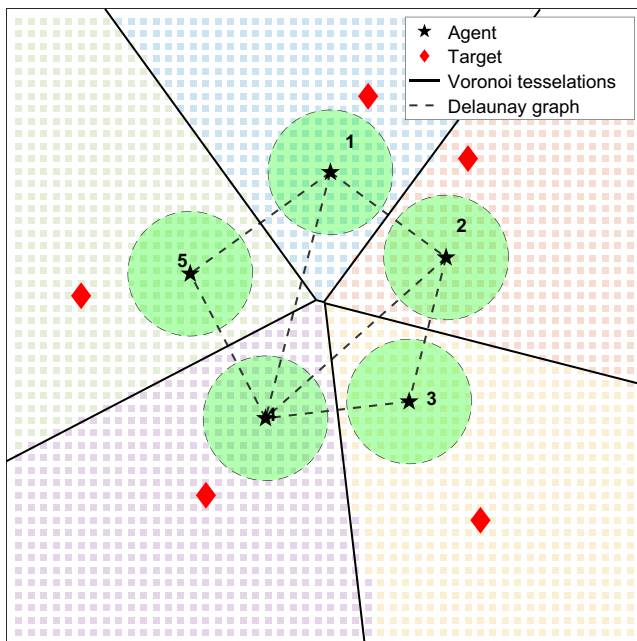


Figure 4. Snapshot of the target-based approach.

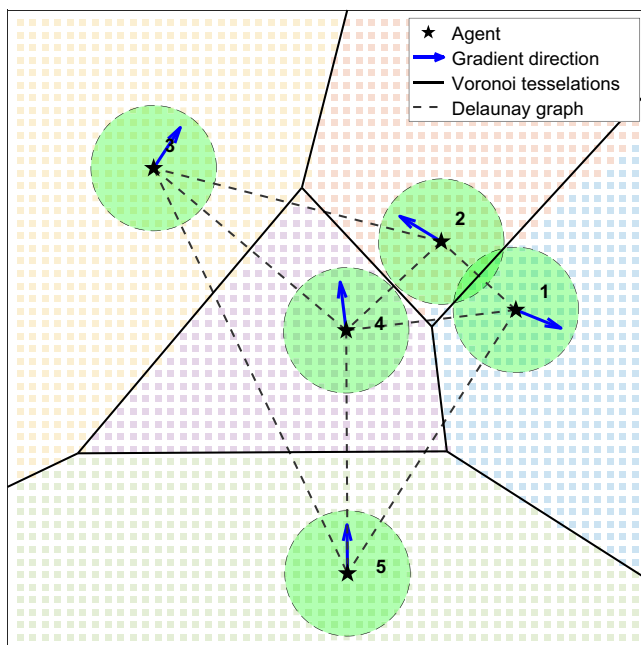


Figure 5. Snapshot of the gradient-based approach.

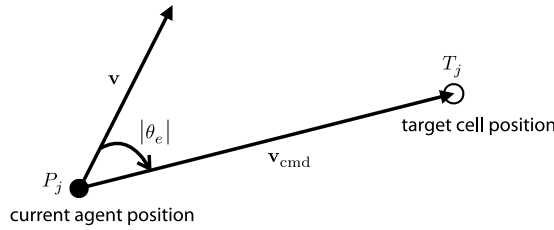


Figure 6. Geometry between the agent position and target position.

Figure 6 shows the geometry between  $P_j$  and  $T_j$ . The angular error  $\theta_e$  is defined as follows:

$$\theta_e = |\theta_e|e \quad \dots (24)$$

where

$$|\theta_e| = \cos^{-1}(\mathbf{v} \cdot \mathbf{v}_{cmd}) \quad \dots (25)$$

$$e = -\text{sign}(\mathbf{v}_n(3)) \quad \dots (26)$$

$$\mathbf{v} = \begin{bmatrix} \mu \cos \theta_j \\ \mu \sin \theta_j \\ 0 \end{bmatrix}, \quad \mathbf{v}_{cmd} = \begin{bmatrix} T_j(1) - P_j(1) \\ T_j(2) - P_j(2) \\ 0 \end{bmatrix}, \quad \mathbf{v}_n = \mathbf{v} \times \mathbf{v}_{cmd} \quad \dots (27)$$

Now, the error dynamic can be designed as follows:

$$\dot{\theta}_e = \frac{d}{dt}(\theta_{cmd} - \theta_j) = -\dot{\theta}_j = -k\theta_e \quad \dots (28)$$

where  $k$  is a positive scalar gain. The target-based control input  $u_j$  that makes the error exponentially converge to zero can be designed as follows:

$$u_i^0 = \dot{\theta}_j = k|\theta_e|e \quad \dots (29)$$

Considering the maximum limit of the control input, we have

$$u_i = \begin{cases} u_{lim} & \text{if } u_i^0 \geq u_{lim} \\ u_i^0 & \text{Otherwise} \\ -u_{lim} & \text{if } u_i^0 \leq -u_{lim} \end{cases} \quad \dots (30)$$

### 4.2 Reactive approach

The reactive control approach guides the agent to the centre of mass direction of the age distribution in the mission space. The individual agent  $j$  computes the direction using the evaluated age vector  $\mathcal{A}_j$ , which is local information. The heading command is computed based on the age of the cells detected by the agent. Because the age of the detected cells is initialised

to zero, the agent computes the direction before initialising the predicted age of the newly detected cells. The center of mass  $C_j$  is computed as follows:

$$C_j = \frac{L_j}{M_j} \in \Omega \quad \dots (31)$$

where  $M_j$  and  $L_j$  are mass and first moment of the detection range  $\mathcal{D}_j$ .

$$M_j = \int_{s \in \mathcal{D}_j} \hat{\alpha}_i^j ds, \quad L_j = \int_{s \in \mathcal{D}_j} (s \cdot \hat{\alpha}_i^j) ds \quad \dots (32)$$

The same guidance law is applied to the reactive control approach, but the command vector  $\mathbf{v}_{\text{cmd}}$  is defined as follows:

$$\mathbf{v}_{\text{cmd}} = \begin{bmatrix} C_j(1) - P_j(1) \\ C_j(2) - P_j(2) \\ 0 \end{bmatrix} \quad \dots (33)$$

## 5.0 NUMERICAL SIMULATION

### 5.1 Evaluation and simulation

A numerical simulation is performed to demonstrate the usefulness of the proposed method. The difference between the simulation result and the evaluated value is the result of the assumptions, which differ from those in the simulation situation. The first factor is the approximation of  $\mu_c$  in Equation (11), which is related to Assumption 2. The actual  $\mu_c$  varies along the velocity direction of the agent because the mission space is discretised into square cells. For instance,  $\mu_c = \mu_n$  when the velocity direction is  $[0, 1]$  or  $[1, 0]$ . Because this may be a low estimate of the real velocity with respect to the cells, the predicted performance of the coverage is lower than that of the simulation result. Note that the square discretisation of the mission space results in directional symmetry in four directions ( $0, \pi/2, \pi$  and  $3\pi/2$  directions). Hexagonal discretisation<sup>(19)</sup> of the mission space would bring about six directional symmetry and may lower the approximation discrepancy of  $\mu_c$ .

The second factor is a redundant detection, which is related to Assumption 3. Redundant detection occurs when a certain cell is detected again before its age increases to the predicted mean age due to the physical location of the cell and agent. That is, cells with low age may be on the path of the agent moving to its target cell, and therefore it will be detected redundantly. Redundant detection may lower the performance of coverage compared to the evaluated performance.

The third factor is the assumption of constant  $m$ . Detection ranges may overlap between agents. In addition, loss of detection range would occur when the agent is close to the boundary of the mission space. This factor is related to Assumption 1.

In this study, four simulation cases are considered. Simulations of Case A and Case B are conducted to demonstrate the properties of the evaluation model using proper simulation parameter settings. Different numbers of agents and agent speeds are considered in Case A and Case B, respectively. The steady-state evaluation and simulation results are compared with respect to each other. Simulations of Case C and Case D are performed to illustrate how

**Table 1**  
**Simulation settings of Case A (agent number variation)**

Parameter	Value
$n$	50
$\mu$	0.5
sensing radius	0.1
$\rho$	$1/\sqrt{n}$
$k$	10
$u_{lim}$	10

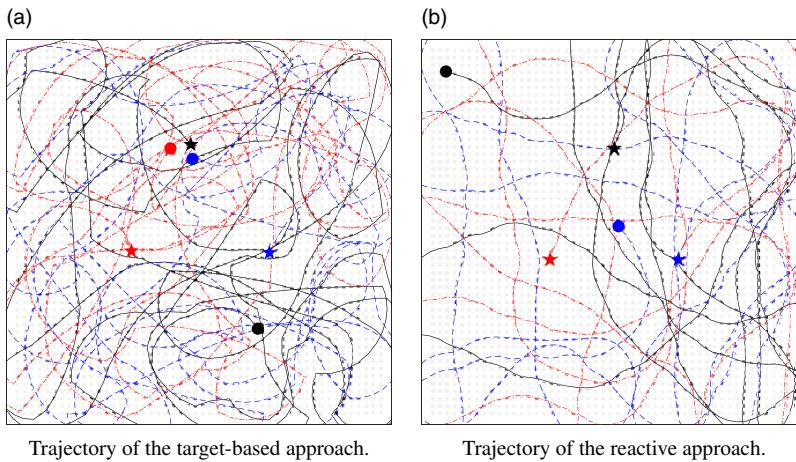


Figure 7. Trajectory of agents.

the evaluation performance is affected by discrepancies between the assumptions and actual simulation situations. The sensing range (sensing radius of the agent) and space discretisation level  $n$  are varied in Case C and Case D, respectively. The values of  $m$  and  $\mu_c$  are computed during the mission, and the average values are compared with the approximated value used in the evaluation model.

## 5.2 Case A. Number of agents

In this section, a numerical simulation is performed to demonstrate the performance of the designed persistent coverage controllers. The performance of the persistent coverage controllers is compared with the predicted performance described in Section 3.0. The simulation conditions are summarised in Table 1.

Adequate simulation parameters are set empirically. The constant  $m$  is computed from the relative extent of the sensing range (area) with respect to that of the mission space  $\Omega$ .

Figure 7 shows the trajectories of the agents for the target-based approach (TBA) and gradient-based approach (GBA). The pentagrams and filled circles are the initial positions and final positions of the agents, respectively. The lines in the figure show the paths of the agents. Figures 8 to 11 are the simulation results and predicted values for various numbers of

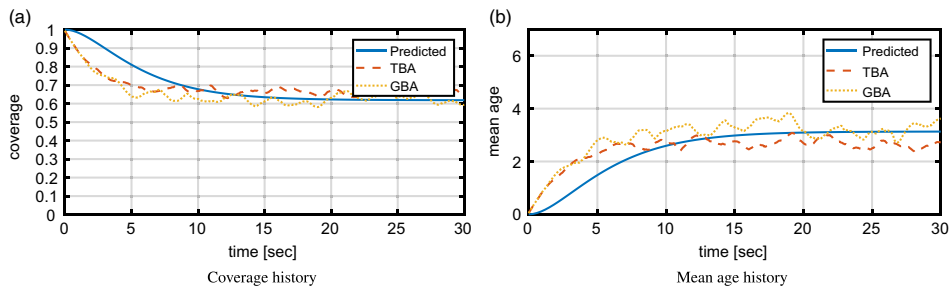


Figure 8. Coverage history and mean age history (three agents).

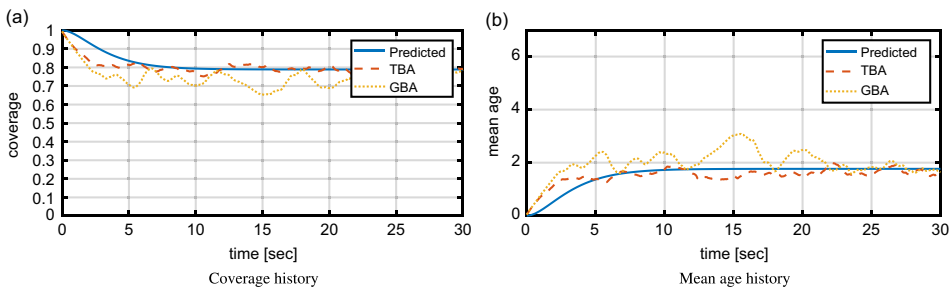


Figure 9. Coverage history and mean age history (five agents).

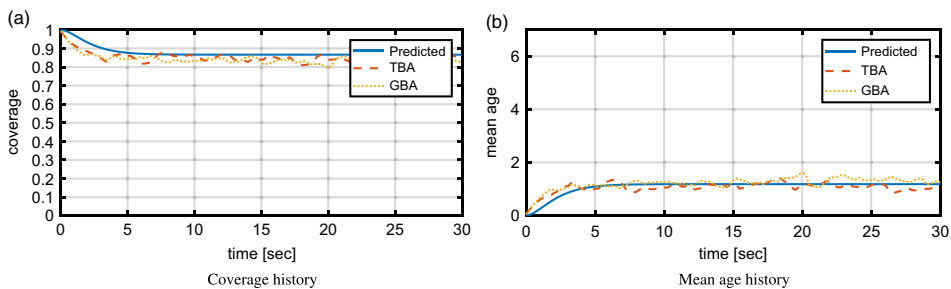


Figure 10. Coverage history and mean age history (seven agents).

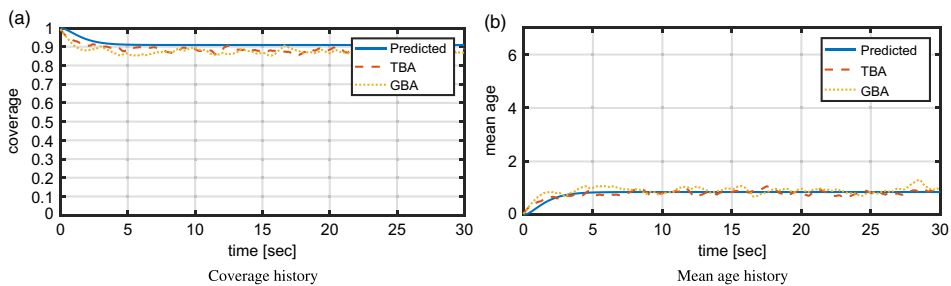


Figure 11. Coverage history and mean age history (nine agents).

**Table 2**  
**Simulation settings of Case B (agent speed variation)**

Parameter	Value
$n$	50
sensing radius	0.1
$M$	3
$\rho$	$1/\sqrt{n}$
$k$	10
$u_{\text{lim}}$	10

**Table 3**  
**Steady-state evaluation performance in Cases A and B**

	Evaluation (%)	Evaluation–TBA  (%p)	Evaluation–GBA  (%p)
3 agents	61.85	5.30	1.28
5 agents	78.90	0.03	4.84
7 agents	86.65	1.66	3.33
9 agents	90.95	2.40	3.66
$\mu = 0.3$	43.27	12.24	7.11
$\mu = 0.5$	61.85	4.88	3.35
$\mu = 0.7$	71.90	2.24	1.43
$\mu = 0.9$	77.87	0.73	4.38

agents. Both the target-based control approach and the reactive control approach show convergence to the steady-state value of mean age with some fluctuations. The predicted mean age and predicted coverage, denoted as a solid line, show decent evaluation of the mean age and coverage values.

The initial age of every cell is zero, and therefore the mean age and coverage start from 0 and 1, respectively. Note that as the number of agents increases, the transient phase is short, and the evaluated age and performance coverage converge to the steady-state values quickly. It is natural that the steady-state mean age is lower as the number of agents increases, and the performance of coverage increases as well.

### 5.3 Case B. Speed of agents

The speed of the agent is also a major factor affecting steady-state performance. The simulation settings are summarised in Table 2. Figures 12 to 15 show the coverage performance and mean age history as  $\mu$  increases. A faster agent speed gives faster convergence to the steady-state performance and higher steady-state performance value. The evaluated performance in the steady state increases and the transient phase becomes shorter as the speed of the agent increases. These results are due to the denominator  $\mu_c$  and exponential term in Equation (20), respectively.

The simulation results for Case A and Case B are summarised in Table 3. The Evaluation column in the table represents the steady-state coverage performance as a percentage. The other columns represent the difference between the evaluated value and the simulation results of TBA and GBA as percentages. The differences are less than 10%p for every simulation conducted except for one case. Note that the same simulation conditions are considered for



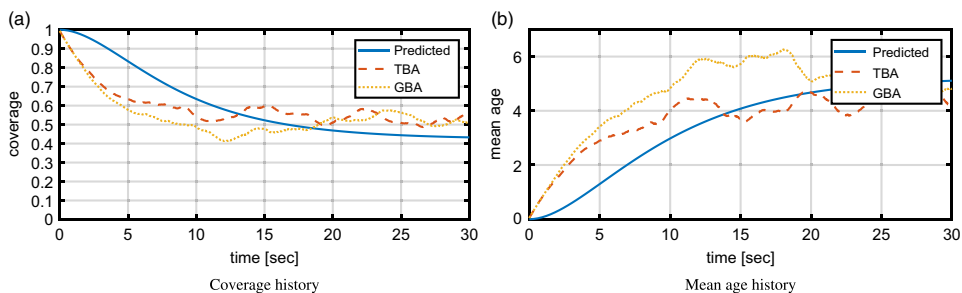


Figure 12. Coverage history and mean age history ( $\mu = 0.3$ ).

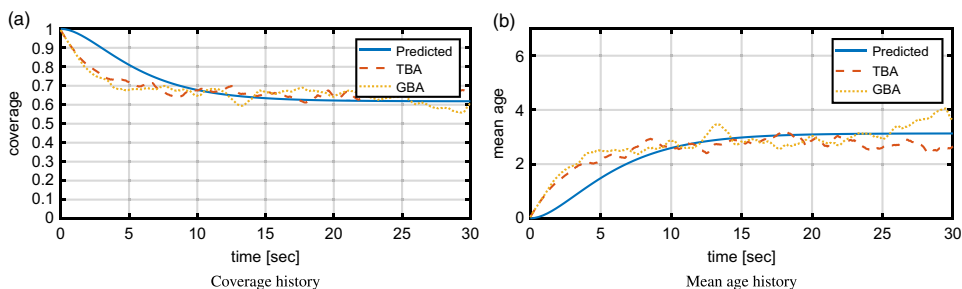


Figure 13. Coverage history and mean age history ( $\mu = 0.5$ ).

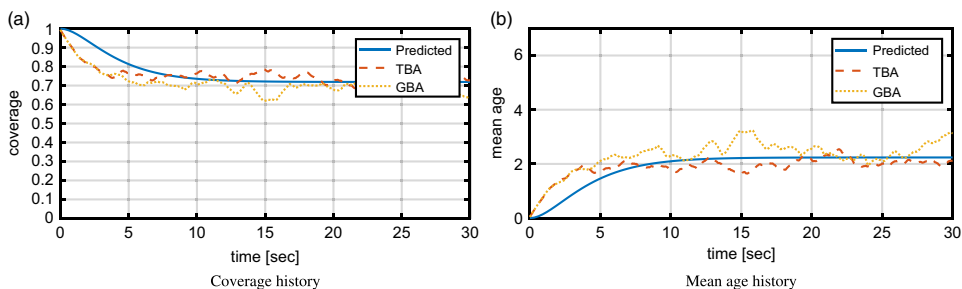


Figure 14. Coverage history and mean age history ( $\mu = 0.7$ ).

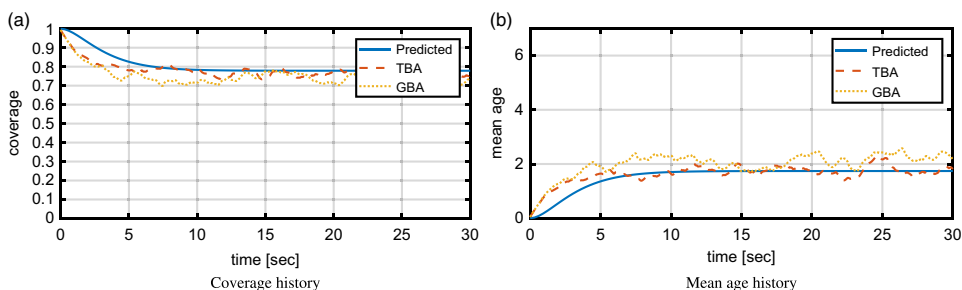


Figure 15. Coverage history and mean age history ( $\mu = 0.9$ ).

**Table 4**  
**Simulation settings of Case C. (sensing radius variation)**

Parameter	Value
$n$	50
$\mu$	0.5
$M$	3
$\rho$	$1/\sqrt{n}$
$k$	10
$u_{\text{lim}}$	10

**Table 5**  
**Simulation settings of Case D (space discretisation level variation)**

Parameter	Value
$\mu$	0.5
sensing radius	0.1
$M$	3
$\rho$	$1/\sqrt{n}$
$k$	10
$u_{\text{lim}}$	10

the first row (Case A) and sixth row (Case B), but the numerical results are slightly different due to the random initial values of each agent's predicted age vector.

#### 5.4 Case C. Sensing range

Simulations of Case C and Case D are conducted to illustrate the effectiveness of the discrepancy between the assumptions and the actual simulation situations. In Case C, the effect of the sensing radius of the agent is considered. A larger relative size of the sensing range with respect to the size of the mission space may affect the simulation performance because it may incur more frequent overlapping of the sensing range between agents and loss of detection range near the mission space boundary. The simulation settings are summarised in Table 4.

Agents with a larger sensing range can detect more cells. However, redundant detections are induced more frequently, and the assumption of constant  $m$  is violated more often during the simulation because of the overlapping and loss of detection ranges between agents. Figures 16 to 19 show the coverage and mean age history for various sensing radii.

#### 5.5 Case D. Level of space discretisation

In the simulation of Case D, different discretisation levels  $n$  are considered. The simulation settings are summarised in Table 5. Because the discretised space is adopted to construct an age model approximating the real continuous space, it is natural to expect that a dense discretisation would result in a more realistic evaluation than sparse discretisation.

Figures 20 to 23 show the coverage history and mean age history to compare the evaluated performance and simulation results with respect to various discretisation levels,  $n$ . The trend of the difference between the evaluated performance and the simulation result is analogous to that of Case C.

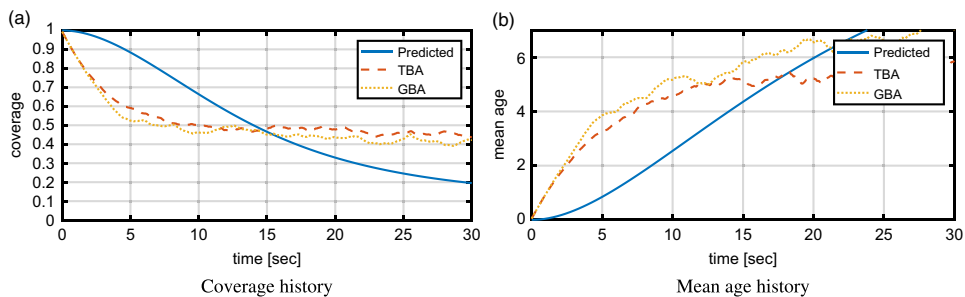


Figure 16. Coverage history and mean age history (sensing radius = 0.05).

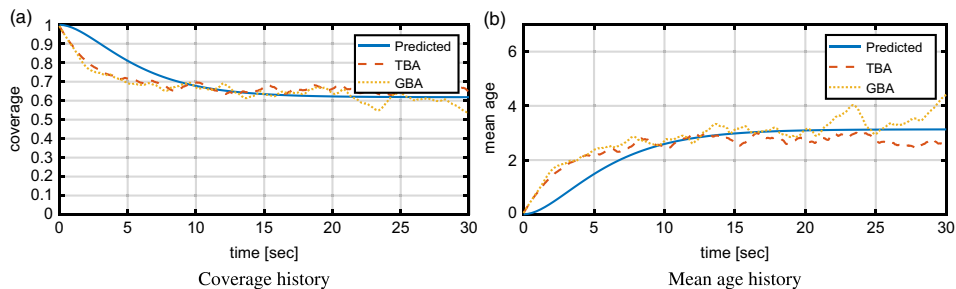


Figure 17. Coverage history and mean age history (sensing radius = 0.10).

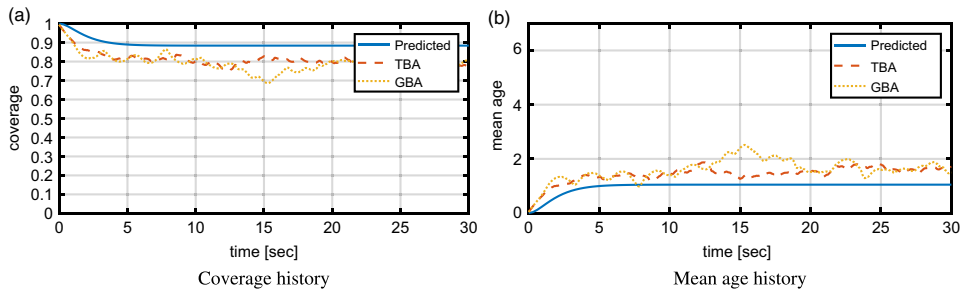


Figure 18. Coverage history and mean age history (sensing radius = 0.15).

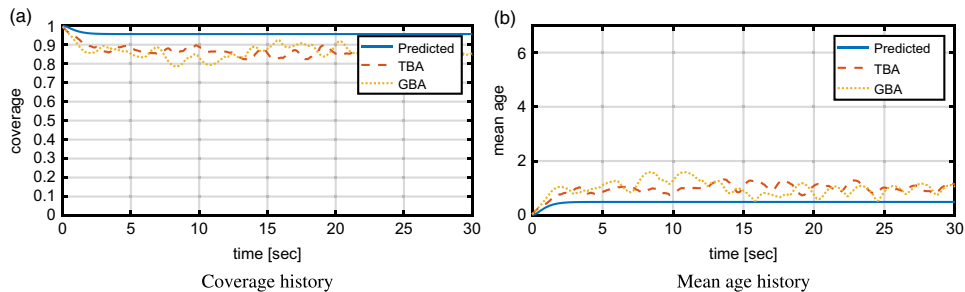


Figure 19. Coverage history and mean age history (sensing radius = 0.20).

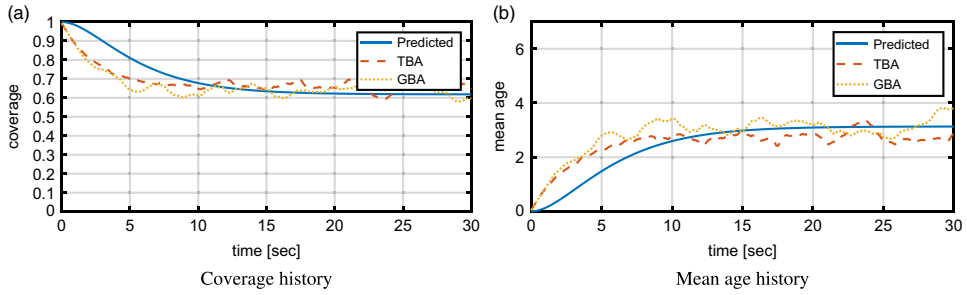


Figure 20. Coverage history and mean age history ( $n = 50$ ).

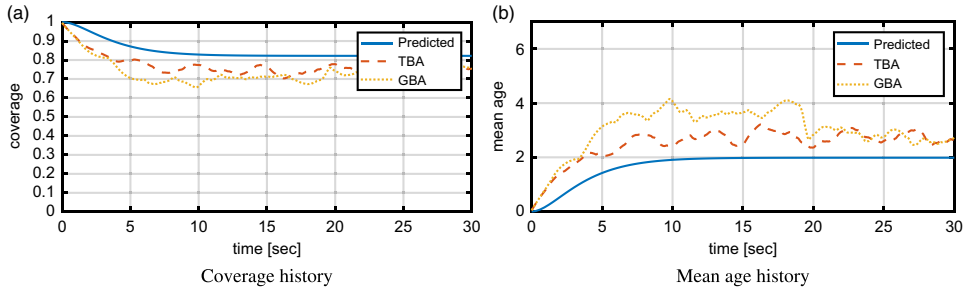


Figure 21. Coverage history and mean age history ( $n = 100$ ).

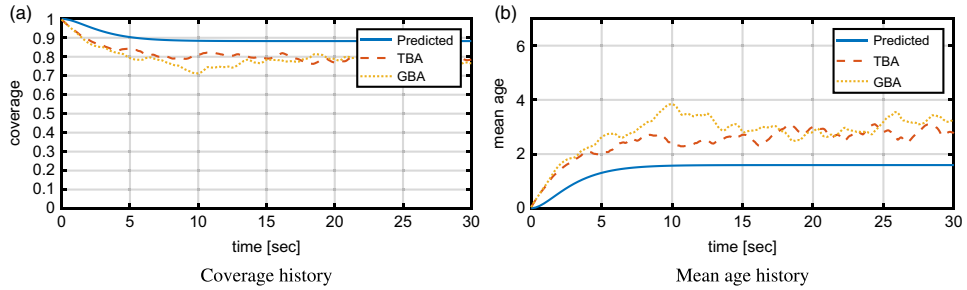


Figure 22. Coverage history and mean age history ( $n = 150$ ).

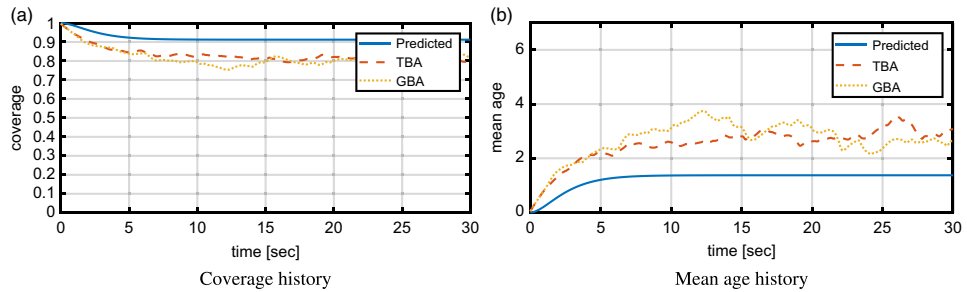


Figure 23. Coverage history and mean age history ( $n = 200$ ).

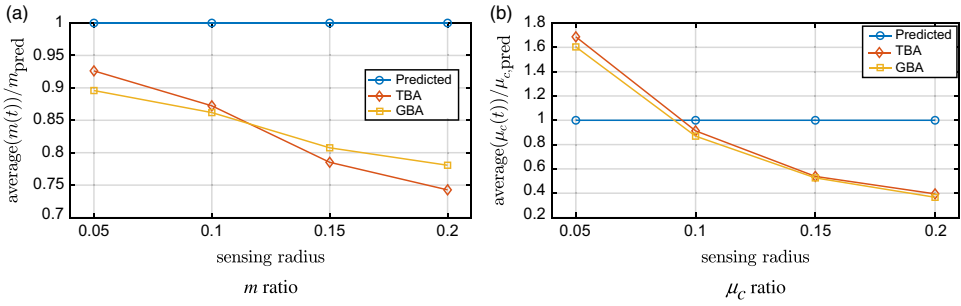


Figure 24. Difference between the evaluation and simulation with respect to the sensing radius.

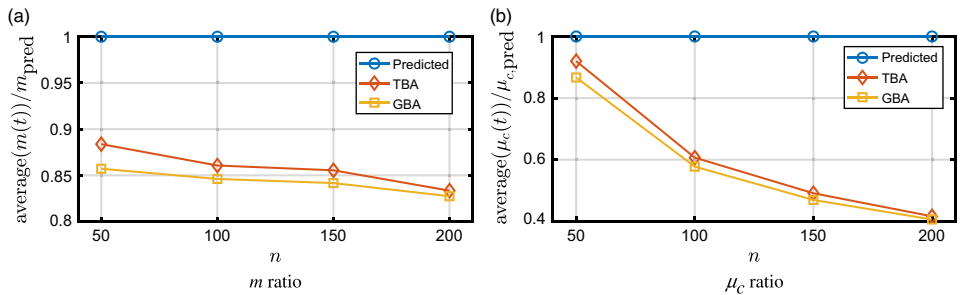


Figure 25. Difference between evaluation and simulation with respect to discretisation level.

### 5.6 Comparative analysis

Figure 24 shows the differences between the evaluation and simulation. The constant  $m$  and  $\mu_c$  assumptions are violated to different extents. In Fig. 24(a), the average  $m$  is divided by  $m_{\text{pred}}$ , which is used in the evaluation. In Fig. 24(b), the average  $\mu_c$  is divided by  $\mu_{c,\text{pred}}$ . Because the overlapping and loss of sensing range increase along with the sensing radius, the relative ratios of the average  $m$  and average  $\mu_c$  (in the simulation) to the assumed  $m_{\text{pred}}$  and  $\mu_{c,\text{pred}} (= \mu\sqrt{n})$  decrease. As a result, whereas the steady-state evaluation of coverage is lower than the simulation result in Fig. 16, it becomes higher than the simulation result as the sensing radius increases (Figs 17 to 19).

Figure 25 shows the differences between the evaluation and simulation situation with respect to  $n$ . As shown in Fig. 25(a), the discretisation level does not have much of an effect on the change in  $m$ . However, as shown in Fig. 25(b), making the discretisation level dense has an adverse effect upon the  $\mu_c$  assumption. That is, the approximation in Equation (11) becomes invalid. As a result, the steady-state coverage performance is overestimated in dense discretisation as shown in Fig. 23. Note that the basic reason for the overestimation is the inadequate  $\mu_c$  approximation, rather than the dense discretisation level.

## 6.0 CONCLUSION

The persistent coverage control problem was modelled with the uncertainty model based on age and the binary deterministic detection model of monitoring agents. Two controllers, the target-based controller and reactive controller, were designed using local information based

on the Voronoi diagram and information sharing via the Delaunay graph. The performance of the general persistent coverage control algorithm was evaluated and calculated as a function of the mission space parameters and the speed of the monitoring agent. The performance evaluated by the proposed method was compared with the performance of two different persistent coverage controllers.

Differences were observed between the evaluated performance and the simulation results. The differences in the evaluation are induced by the assumptions, which differ from the actual simulation situations, i.e. the constant speed approximation, redundant detection, and constant number of cells in the sensing region assumption. The factors responsible for the differences were computed by numerical simulation, and the interrelations between the factors and mission space-describing parameters were discussed. The mission space-describing parameters are the relative size of the sensing range with respect to the mission space and the discretisation level of the mission space. Based on this knowledge, one may select adequate assumptions and approximations of parameters for accurate performance evaluation. The evaluated mean age and performance function values provide useful and meaningful evaluation of the coverage control performance by setting proper simulation parameters, including the discretisation level.

## REFERENCES

1. NIGAM, N. The multiple unmanned air vehicle persistent surveillance problem: A review, *Machines*, January 2014, **2**, (1), pp 13–72.
2. MOHSENI, F., DOUSTMOHAMMADI, A. and MENHAJ, M.B. Centralized receding horizon coverage control for mobile sensory networks. *International Conference on Intelligent Systems Modelling and Simulation (ISMS)*, 8–10 February, 2012, Kota Kinabalu, Malaysiapp, pp 588–593.
3. MOHSENI, F., DOUSTMOHAMMADI, A. and MENHAJ, M.B. Distributed receding horizon coverage control for multiple mobile robots, *IEEE Systems Journal*, March 2016, **10**, (1), pp 198–207.
4. LIN, X. and CASSANDRAS, C.G. An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces, *IEEE Transactions on Automatic Control*, June 2015, **60**, (6), pp 1659–1664.
5. NIGAM, N., BIENIAWSKI, S., KROO, I. and VIAN, J. Control of multiple UAVs for persistent surveillance: Algorithm and flight test results, *IEEE Transactions on Control Systems Technology*, September 2012, **20**, (5), pp 1236–1251.
6. SOLTERO, D.E., SCHWAGER, M. and RUS, D. Decentralized path planning for coverage tasks using gradient descent adaptive control, *The International Journal of Robotics Research*, March 2014, **33**, (3), pp 401–425.
7. GALCERAN, E. and CARRERAS, M. A survey on coverage path planning for robotics, *Robotics and Autonomous Systems*, December 2013, **61**, (12), pp 1258–1276.
8. SCHWAGER, M., RUS, D. and SLOTINE, J.-J. Decentralized, adaptive coverage control for networked robots, *The International Journal of Robotics Research*, March 2009, **28**, (3), pp 357–375.
9. WALLAR, A., PLAKU, E. and SOFGE, D. A. Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas, *IEEE Transactions on Automation Science and Engineering*, July 2015, **12**, (3), pp 969–980.
10. SCHWAGER, M., VITUS, M.P., POWERS, S., RUS, D. and TOMLIN, C.J. Robust adaptive coverage control for robotic sensor networks, *IEEE Transactions on Control of Network Systems*, September 2017, **4**, (3), pp 462–476.
11. WANG, Y., WU, S., CHEN, Z., GAO, X. and CHEN, G. Coverage problem with uncertain properties in wireless sensor networks: A survey, *Computer Networks*, August 2017, **123**, pp 200–232.
12. ZUO, L., SHI, Y. and YAN, W. Dynamic coverage control in a time-varying environment using Bayesian prediction, *IEEE Transactions on Cybernetics*, December 2017, pp 1–9. (On-line publication)

13. LIN, X. and CASSANDRAS, C.G. An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces, *IEEE Transactions on Automatic Control*, June 2015, **60**, (6), pp 1659–1664.
14. PALACIOS-GASOS, MANUEL, J. and SAGÜÉS, C. Distributed coverage estimation and control for multirobot persistent tasks, *IEEE Transactions on Robotics*, **32**, December 2016, (6), pp 1444–1460.
15. RAMASAMY, M. and GHOSE, D. Learning-based preferential surveillance algorithm for persistent surveillance by unmanned aerial vehicles, *International Conference on Unmanned Aircraft Systems (ICUAS)*, 7–10 June, 2016, Arlington, VA, pp 1032–1040.
16. CASSANDRAS, C.G., LIN, X. and DING, X. An optimal control approach to the multi-agent persistent monitoring problem, *IEEE Transactions on Automatic Control*, April 2013, **58**, (4), pp 947–961.
17. MARTINOLI, A., PALACIOS-GASOS, M., TALEBPOUR, Z., MONTIJANO, E. and SAG, C. Optimal path planning and coverage control for multi-robot persistent coverage in environments with obstacles. *IEEE International Conference on Robotics and Automation (ICRA)*, 29 May–3 June, 2017, Singapore, Singapore, pp 1321–1327.
18. CORTES, J., MARTINEZ, S., KARATAS, T. and BULLO, F. Coverage control for mobile sensing networks, *IEEE Transactions on Robotics and Automation*, April 2004, **20**, (2), pp 243–255.
19. QUIJANO, H.J. and GARRIDO, L. Improving cooperative robot exploration using an hexagonal world representation, *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, 25–28 September, 2007, Morelos, Mexico, pp 450–455.