Distributed Sweep Coverage Algorithm of Multi-agent Systems Using Workload Memory

Chao Zhai, Gaoxi Xiao and Michael Z. Q. Chen *

November 18, 2018

Abstract

This paper addresses the sweep coverage problem of multi-agent systems in the uncertain environment. A novel formulation of distributed sweep coverage is proposed for multiple agents to cooperatively complete the workload in the coverage region. To save the sweep time, each agent takes part in partitioning the whole region using its partition bar while sweeping its own subregion at a constant rate. The trajectories of partition bars form the boundaries between adjacent subregions. Essentially, the partition operation is carried out by means of workload memory in order to balance the workload in each subregion. In particular, it is proved that the dynamics of multi-agent system is input-to-state stable. Theoretical analysis is conducted to obtain the upper bound of the error between the actual sweep time and the optimal sweep time. Moreover, a sufficient condition is provided to avoid the collision of partition bars during the partition. Finally, numerical simulations demonstrate the effectiveness of the proposed approach.

Keywords: sweep coverage, multi-agent systems, distributed control, workload memory

1 Introduction

The great progress in communication technologies makes it easy and low-cost to share mutual information and coordinate the joint actions among multiple agents. Thus, cooperative control of

*Chao Zhai and Gaoxi Xiao are with Institute of Catastrophe Risk Management and School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. They are also with Future Resilient Systems, Singapore-ETH Centre, 1 Create Way, CREATE Tower, Singapore 138602. Email: zhaichao@amss.ac.cn, egxxiao@ntu.edu.sg. Michael Z. Q. Chen is with the School of Automation, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, P. R. China. Email: mzqchen@outlook.com. Corresponding author: Gaoxi Xiao

multi-agent systems has attracted much interest of researchers in various fields in the past decade. The coordination of multiple agents contributes to improving the efficiency and robustness of carrying out complicated tasks, such as leader tracking [1, 2, 3], flocking behavior [4, 5], boundary patrolling [6], persistent monitoring [7, 8] and region coverage [9], to name just a few.

As a type of coordination tasks, cooperative coverage of multi-agent systems refers to the path planning of a robot team to visit every point in the environment or the optimal deployment of sensor networks according to certain performance indexes. The approach of divide-and-conquer is widely applied in the region coverage of multi-agent systems [10, 11, 12]. Specifically, [11] presents a gradient descent algorithm to optimize a class of utility functions in the coverage region, where the centroidal Voronoi partition is adopted to allocate a subregion for each mobile sensor. [12] extends the above work by proposing a distributed, adaptive coverage algorithm for nonholonomic mobile sensors. In multi-robot coordination, the coverage problem falls into three categories: blanket coverage, barrier coverage and sweep coverage [13]. Blanket coverage aims at deploying multiple agents in the given coverage region to maximize the probability of identifying the target [14]. Barrier coverage is used to protect the target in the given region and meanwhile maximize the detection rate of invaders overpassing the barrier that is formed by the agents [15]. As an important type of multi-robot coverage, sweep coverage can be used in many tasks, such as maintenance inspection [13], border patrolling [16] and environmental monitoring [17]. It is largely unexplored to investigate multi-agent sweep coverage in the uncertain environment, where the exact information on the coverage region is unknown in advance. Thus, the distributed control algorithm is developed to deal with the uncertainty in multi-agent sweep coverage. Sweep coverage can actually be regarded as a moving barrier, and it focuses on sweeping or monitoring the given region by arriving at every point. For example, [16] investigates the sweep coverage of mobile sensors in a corridor environment without taking into account the workload distribution. To address this issue, [9] develops a sweep coverage algorithm by dividing the whole region into a series of stripes and then cooperatively completing the workload in each stripe in sequence. Nevertheless, it is not a fully distributed algorithm because agents have to shift stripes under the centralized command. Moreover, the partition error of workload accumulates rapidly as the number of stripes increases.

In this paper, we investigate the sweep coverage problem of multi-agent systems in the uncertain environment, where the workload distribution is unknown in advance. The agents cooperatively partition the whole region using the trajectories of their partition bars while sweeping their respective subregions in a distributed manner. Compared with existing work, our approach achieves the fully distributed sweep coverage of multi-agent system in the uncertain environment by capitalizing on the workload memory, and the sweeping task can be fulfilled with guaranteed complete time in theory. Specifically, the main contributions of this work are listed as follows.

1. A distributed sweep coverage formulation of multi-agent systems is developed with workload memory, and an upper bound for the error between the actual sweeping time and the optimal time is estimated.
2. It is demonstrated that the multi-agent system is input-to-state stable with respect to the vertical speed of partition bars.
3. A sufficient condition is derived to avoid the collision of partition bars in the process of partitioning the coverage region.

The remainder of this paper is organized as follows. Section 2 formulates the problem of distributed sweep coverage and proposes the sweep coverage algorithm of multi-agent systems. Sections 3 and 4 present theoretical results on the proposed sweep coverage algorithm, followed by numerical simulations in Section 5. Finally, we draw the conclusion and discuss the future direction in Section 6.

2 Problem Formulation

This section formulates the sweep coverage problem of multi-agent systems in the uncertain environment, where the distribution of workload (e.g., dust in the room, crops in the farmland, leaking oil in the sea, etc) is unknown to the agents in advance. First of all, we present the formal description of multi-agent sweep coverage in the uncertain environment. Then the goal of multi-agent sweep coverage is provided and the multi-agent dynamics is designed mathematically. Finally, a distributed sweep coverage algorithm of multi-agent systems is proposed to fulfil the sweeping task in the uncertain region by means of divide and conquer.

The problem of multi-agent sweep coverage in the uncertain environment is described as follows: Consider the two dimensional coverage region Ω (see Fig. 1) enclosed by two parallel horizontal lines with the distance l and two smooth boundaries described by $x = g_a(y)$ and $x = g_b(y)$, respectively. The distribution density function of workload is given by $\rho(x, y)$, satisfying $\rho(x, y) \in [\underline{\rho}, \bar{\rho}]$, where $\underline{\rho}$ and $\bar{\rho}$ are the lower and upper bounds of $\rho(x, y)$, respectively. To be precise, the uncertain environment is defined as the 2-tuple $(\Omega, \rho(x, y))$ with $\rho(x, y) \in [\underline{\rho}, \bar{\rho}]$,

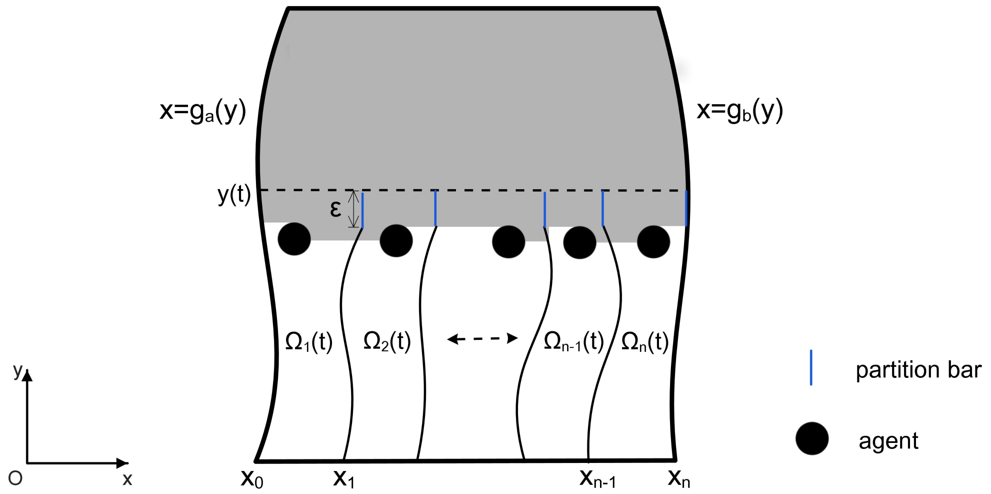

Figure 1: Sweep coverage of the uncertain region Ω with smooth boundaries. The left and right boundaries of coverage region are described by $x = g_a(y)$ and $x = g_b(y)$, respectively. The blue line segments with the length ϵ denote the partition bars of agents, and their trajectories divide the partition region into n subregions (i.e., $\Omega_i(t), i \in \mathcal{I}_n$ at time t). The uncompleted part of the region Ω is marked in gray, while the completed part is marked in white.

$\forall (x, y) \in \Omega$. There are n mobile agents responsible for sweeping the workload in the region Ω. Each agent can only detect the workload in its neighborhood and communicate with its neighbors to share the workload information. In addition, each agent is assumed to complete the workload on the coverage region at the constant sweeping rate of σ. In practice, σ depends on the sweeping performance of agents (e.g., sweeping robots, robot vacuums), and it is determined by the completed workload per unit time. In brief, the problem is how to design the multi-agent dynamics and allocate the workload for each agent so that the workload in the coverage region can be completed as soon as possible. This is a challenging problem because the global information on the workload distribution is unknown to the agents, which makes it impossible to equally divide the workload for each agent in advance. As a result, an online algorithm of multi-agent systems has to be developed to achieve the workload allocation for each agent by using the local workload information.

Suppose an online algorithm manages to divide the whole region Ω into n subregions (i.e., $\Omega = \bigcup_{i=1}^{n} \Omega_i$). Let T^* represent the optimal sweep time of multi-agent systems, and it can be achieved if the whole region is divided into n subregions with equal workload. In this way, all the agents can complete the workload in their own subregions simultaneously. Thus, the optimal sweep time T^* is given by $T^* = \bar{m}/\sigma$, where

$$\bar{m} = \frac{1}{n}\iint_{\Omega} \rho(x,y)dxdy = \frac{1}{n}\sum_{i=1}^{n}\iint_{\Omega_i} \rho(x,y)dxdy = \frac{1}{n}\sum_{i=1}^{n} m_i$$

denotes the average workload in each subregion, and m_i refers to the workload in the i-th subregion Ω_i. In fact, T^* is normally unavailable due to the local information on workload distribution, and it is only regarded as the benchmark to assess the performance of a sweep coverage algorithm. Considering that an online algorithm normally leads to the unequal allocation of workload in subregions, the actual sweep time T mainly depends on the maximum workload in subregions. And it can be computed as:

$$T = \frac{1}{\sigma}\max_{i\in\mathcal{I}_n}\iint_{\Omega_i} \rho(x,y)dxdy = \frac{1}{\sigma}\max_{i\in\mathcal{I}_n} m_i, \quad \mathcal{I}_n = \{1,2,...,n\},$$

Therefore, the error between the actual sweep time and the optimal sweep time ΔT is given by:

$$\Delta T = T - T^* = \frac{1}{\sigma}\left(\max_{i\in I_n} m_i - \bar{m}\right).$$

The goal of this paper is to design a real-time sweep coverage algorithm of multi-agent systems with local workload information to complete the workload in the region Ω and estimate the upper bound of time error ΔT.

Next, we propose the partition dynamics of multi-agent systems and then present the distributed sweep coverage algorithm in real time. To achieve the real-time workload partition, each agent is equipped with a partition bar with the length ϵ to allocate the workload in the whole region (see Fig.1). Let (x_i, y_i), $i \in \mathcal{I}_n$ refer to the coordinate position (i.e., the lower terminal point) of partition bar for agent i, then the dynamics of partition bars is designed as

$$\begin{cases} \dot{x}_i = \kappa \sum_{j\in\mathcal{N}_i} (m_j(t) - m_i(t)) \\ \dot{y}_i = v, \quad i \in \mathcal{I}_n \end{cases} \tag{1}$$

where κ refers to a positive constant, and $\mathcal{N}_i$ represents the set of neighbors of agent i. In addition, v denotes the vertical speed of agents along the y-axis.

$$m_i(t) = \iint_{\Omega_i(t)} \rho(x,y)dxdy$$

represents the workload in the subregion $\Omega_i(t)$ at time t. The idea behind the dynamics (1) is that the partition bars move at the same speed v along the y-axis. Meanwhile, each agent communicates with its neighbors nearby to share the workload information of their respective subregions, so that their partition bars also move along the x-axis to equally partition the workload in the region Ω (For Agent n, we have $\dot{x}_n = g_b^{'}(y_n)\dot{y}_n = g_b^{'}(y_n)v$, since its partition bar moves along the boundary $x = g_b(y)$ at the speed v along the y-axis). Eventually, the trajectories of partition bars form the boundaries between adjacent subregions, and thus the region Ω is divided into n subregions (i.e., $\Omega_i, i \in \mathcal{I}_n$). To save the sweep time, each agent is required to complete the workload in their respective subregions at the constant sweeping rate σ, while partitioning the workload in the region with the dynamics (1).

Remark 2.1. *At the beginning, all agents are located at the bottom of the region Ω (i.e., $y = 0$). When it starts the sweeping process, the partition bars of agents move towards the top of the region Ω (i.e., $y = l$) with the dynamics (1). In this way, the trajectories of partition bars succeed in dividing the coverage region Ω into subregions.*

For simplicity, the distributed sweep coverage algorithm (DSCA) of multi-agent systems is summarized in Table 1. First of all, it is necessary to set the parameters v, σ and ϵ using constant values and compute the complete time of region partition $T_p = (l - \epsilon)/v$ and the initial error $e_i(0) = m_i(0), i \in \mathcal{I}_n$. Then each agent (say, agent i) starts to sweep its own subregion at the sweeping rate σ and meanwhile cooperates with its neighbors for the equal workload partition using workload memory. After the region partition is completed at time $t = T_p$, each agent focuses on sweeping the workload in its own subregion. Finally, the sweeping task is fulfilled after all the agents have cleaned up the workload in their respective subregions.

Remark 2.2. *The DSCA is expected to sweep the workload in the general connected and bounded regions via coordinate transformation. Specifically, a homeomorphism $f : X \longrightarrow Y$ with the condition $\int_X \rho(x,y)dxdy = \int_Y \hat{\rho}(x,y)dxdy$ can be constructed from the original bounded region X to the regular bounded region Y (e.g., a rectangular region). By implementing the DSCA in the regular region Y, the trajectories of partition bars (i.e., the boundaries between subregions) are mapped onto the original region X via the inverse function $f^{-1} : Y \longrightarrow X$, which allows to achieve the sweep coverage of the original region X.*

3 Key Lemmas

To quantify the partition performance of multi-agent system with the dynamics (1), we introduce the mismatch vector at time t:

$$\Delta m(t) = m(t) - \bar{m}(t) \cdot \mathbf{1}_n, \tag{2}$$

Table 1: Distributed Sweep Coverage Algorithm.

```
**Input:** $x_i(0), i \in \mathcal{I}_n$    **Output:** $m_i(t)$, $t \in [0, T_p]$
Set parameters $v$, $\sigma$ and $\epsilon$
Compute $T_p$ and $e_i(0) = m_i(0), i \in \mathcal{I}_n$
**while** $(e_i(t) > 0)$
    **if** $(t \leq T_p)$
        Compute the workload $m_i(t)$ in $\Omega_i(t)$
        Obtain the workload $m_j(t)$ in $\Omega_j(t)$, $j \in \mathcal{N}_i$
        Partition the region $\Omega$ with the dynamics (1)
    **end if**
    Sweep the workload in $\Omega_i(t)$ with the rate $\sigma$
    Compute residual workload $e_i(t) = m_i(t) - \sigma t$
**end while**
```

where $m(t) = (m_1(t), m_2(t), ..., m_n(t))^T \in R^n$ and $\mathbf{1}_n = (1, 1, ..., 1)^T \in R^n$. $\bar{m}(t)$ denotes the average workload on each subregion at time t, and it is computed as:

$$\bar{m}(t) = \frac{1}{n} \iint_{\Omega(t)} \rho(x, y) dx dy$$

with the partition region $\Omega(t) = \bigcup_{i=1}^{n} \Omega_i(t)$ at time t. Clearly, the workload is equally divided for each subregion when $\|\Delta m(t)\| = 0$ with the 2-norm $\| \cdot \|$. Next, three lemmas are provided to facilitate the derivation of main theoretical results in this paper. The logical path is presented as follows: Lemma 3.1 contributes to establishing the inequality in Lemma 3.3, which serves to demonstrate the input-to-sate stability of Multi-agent dynamics (1) in Theorem 4.1. Lemma 3.2 and Lemma 3.3 are used to estimate the upper bound of time error ΔT in Theorem 4.2. In addition, Lemma 3.3 helps to derive the sufficient condition of collision avoidance among subregion boundaries in Theorem 4.3.

The first lemma uncovers the relationship between the partition error of workload in subregions and the average workload in each subregion.

Lemma 3.1.

$$\|\Delta m(t)\| \leq \sqrt{n(n-1)} \cdot \bar{m}(t).$$

Proof. See Appendix A. □

Then we present the second lemma, which provides an estimation for the maximum partition error of workload in subregions, compared to the total partition error.

Lemma 3.2.

$$\|\Delta m(t)\|_\infty \leq \sqrt{(n-1)/n} \cdot \|\Delta m(t)\|.$$

Proof. See Appendix B. □

Remark 3.1. *For $\Delta m(t) \in R^n$, there exists the relationship between the infinity norm and the Euclidean norm with the properties of norms as follows $\|\Delta m(t)\|_\infty \leq \|\Delta m(t)\|$. The relationship in Lemma 3.2 is less conservative according to $\|\Delta m(t)\|_\infty \leq \sqrt{(n-1)/n}\|\Delta m(t)\| \leq \|\Delta m(t)\|$. This is because the elements of $\Delta m(t)$ satisfy the constraint $\sum_{i=1}^n \Delta m_i(t) = 0$.*

The partition error of workload in subregions evolves as the DSCA in Table 1 is implemented. The third lemma aims to estimate the partition error of workload with respect to time t, speed v, communication network and other parameters of the coverage region.

Lemma 3.3.

$$\|\Delta m(t)\| \leq \|\Delta m(t_0)\| e^{-\frac{\kappa\lambda_2(t-t_0)}{2}} + \frac{1}{2}\int_{t_0}^{t} e^{\frac{\kappa\lambda_2(\tau-t)}{2}} \zeta(\tau) d\tau, \tag{3}$$

where λ_2 is the second smallest eigenvalue of a symmetric matrix and

$$\zeta(\tau) = 2v\bar{\rho}\left[\epsilon\sqrt{g_a'(y)^2 + g_b'(y)^2} + (g_b(y) - g_a(y))\sqrt{(n-1)/n}\right]. \tag{4}$$

Proof. See Appendix C. □

Remark 3.2. *λ_2 is the second smallest eigenvalue of a symmetric matrix that depends on both communication network of multi-agent systems and the workload distribution on the coverage region. Mathematically, it can be expressed as*

$$\lambda_2 = \inf_{x_i \in \mathcal{C}} \lambda_2\left(\frac{G(x)L_{n-1} + L_{n-1}^T G(x)^T}{2}\right)$$

with $x = (x_1, x_2, ..., x_{n-1})$, $\mathcal{C} = [\inf_{0\leq y\leq l} g_a(y), \sup_{0\leq y\leq l} g_b(y)]$ and $G(x) = -J_{-1,n}^T \cdot E_{n-1} \cdot \Lambda(\epsilon, x)$. Here, $J_{-1,n}$ denotes the n-dimensional Jordan matrix with the diagonal elements being -1, and E_{n-1} is composed of the first (n-1) columns in the n-dimensional unit matrix. In addition, $\Lambda(\epsilon, x)$ refers to the (n-1) dimensional diagonal matrix with the i-th diagonal element being $\int_{y-\epsilon}^{y} \rho(x_i, y)dy$, $i \in \mathcal{I}_{n-1}$. Finally, $L_{n-1} \in R^{(n-1)\times n}$ is composed of the first (n-1) rows *of Laplacian matrix that represents the communication network of multi-agent systems. To guarantee the partition performance, the topology structure of community network has to be predetermined such that $\lambda_2 > 0$ holds.*

4 Main Results

This section provides main theoretical results for the DSCA in Table 1 with the aid of lemmas in Section 3. First of all, it is proved that multi-agent system (1) is input-to-state stable. Then the upper bound of ΔT is estimated for completing the workload in the coverage region Ω, followed by a sufficient condition to guarantee the collision avoidance of subregion boundaries. Finally, some discussions are presented in the remarks.

By treating the speed v of partition bars along the y-axis as the external input, we demonstrate the input-to-state stability of multi-agent system (1) as follows.

Theorem 4.1. *Multi-agent system (1) is input-to-state stable.*

Proof. By treating $\Delta m(t)$ as the state vector of multi-agent system (1), Inequality (3) in Lemma 3.3 can be rewritten as

$$\begin{aligned}\|\Delta m(t)\| &\leq \|\Delta m(t_0)\| e^{-\frac{\kappa\lambda_2(t-t_0)}{2}} + \frac{1}{2}\int_{t_0}^{t} e^{\frac{\kappa\lambda_2(\tau-t)}{2}}\zeta(\tau)d\tau \\ &\leq \|\Delta m(t_0)\| e^{-\frac{\kappa\lambda_2(t-t_0)}{2}} + \frac{\bar{\zeta}}{\kappa\lambda_2},\end{aligned}$$

where $\bar{\zeta} = 2\bar{\rho}\|v\|_{[t_0,t]}\sup_{y\in[0,l]}\left[\epsilon\sqrt{g_a'(y)^2+g_b'(y)^2}+(g_b(y)-g_a(y))\sqrt{(n-1)/n}\right]$ and $\|v\|_{[t_0,t]} = \sup_{\tau\in[t_0,t]}\|v(\tau)\|$. Clearly, $\bar{\zeta}$ is a continuous increasing function with respect to the speed v, and it satisfies $\bar{\zeta} = 0$ when $v = 0$. Therefore, it follows from the definition in [18] that multi-agent system (1) is input-to-state stable when the speed v is regarded as the external input. □

For the DSCA described in Table 1, we provide a quantitative estimation for the upper bound of time error ΔT, which allows us to assess the sweep coverage performance of multi-agent systems.

Theorem 4.2. *With the DSCA in Table 1, the time error ΔT for sweeping the coverage region Ω is bounded by*

$$\Delta T \leq \frac{1}{\sigma}\sqrt{(n-1)/n}\left(\|\Delta m(0)\| e^{-\frac{\kappa\lambda_2 T_p}{2}} + \frac{1}{2}\int_{0}^{T_p} e^{\frac{\kappa\lambda_2(\tau-T_p)}{2}}\zeta(\tau)d\tau\right). \tag{5}$$

where $\zeta(\tau)$ is given by Equation (4).

Proof. Since the sweeping rate σ is the same for all agents, ΔT depends on the subregion with the maximum workload in subregions. Thus, it can be computed by

$$\Delta T = \frac{1}{\sigma}\left(\max_{i\in I_n} m_i(T_p) - \bar{m}(T_p)\right) \leq \frac{1}{\sigma}\max_{i\in\mathcal{I}_n}|m_i(T_p) - \bar{m}(T_p)| = \frac{1}{\sigma}\|\Delta m(T_p)\|_\infty.$$

From Lemma 3.2 and Lemma 3.3 with $t_0 = 0$, we have

$$\begin{aligned}\Delta T &\leq \frac{1}{\sigma}\sqrt{(n-1)/n}\cdot\|\Delta m(T_p)\|\\ &\leq \frac{1}{\sigma}\sqrt{(n-1)/n}\left(\|\Delta m(0)\|e^{-\frac{\kappa\lambda_2 T_p}{2}} + \frac{1}{2}\int_0^{T_p} e^{\frac{\kappa\lambda_2(\tau - T_p)}{2}}\zeta(\tau)d\tau\right).\end{aligned}$$

The proof is thus completed. □

Remark 4.1. *Compared with existing results in [9], the proposed sweep coverage algorithm in this work can be implemented in a distributed manner to complete the workload without the centralized control command. When the vertical speed v is relatively small and ϵ is less than the stripe width d, Theorem 4.2 in this work can provide a tighter estimation on the upper bound of time error ΔT, compared to Theorem 4.1 in [9].*

Remark 4.2. *A tighter upper bound for time error ΔT can be obtained when ϵ is sufficiently small. This is due to $\lim_{\epsilon\to 0}\|\Delta m(0)\| = 0$, which leads to a tighter upper bound as follows*

$$\Delta T \leq \frac{\bar{\rho}v(n-1)}{\sigma n}\int_0^{T_p} e^{\frac{\kappa\lambda_2(\tau - T_p)}{2}}\left[g_b(v\tau) - g_a(v\tau)\right]d\tau,$$

according to Equation (5) in Theorem 4.2. Nevertheless, if the parameter ϵ is equal to 0, the partition bar degrades into a point and fails to effectively balance the workload in subregions during the sweeping process. Thus, ϵ should be positive and sufficiently small for a tight upper bound of time error ΔT.

Remark 4.3. *To facilitate the estimation of time error ΔT, the time interval $[0, T_p]$ can be divided into a series of consecutive subintervals $c_i = [(i-1)T_p/q, iT_p/q]$, $1 \leq i \leq q$ and $q \in Z^+$. Considering that $e^{\frac{\kappa\lambda_2(\tau - T_p)}{2}}$, $0 \leq \tau \leq T_p$ is an increasing function with respect to τ, we have*

$$\int_0^{T_p} e^{\frac{\kappa\lambda_2(\tau - T_p)}{2n}}\zeta(\tau)d\tau \leq \sum_{i=1}^{q} e^{\frac{\kappa\lambda_2(t_i - T_p)}{2}}\int_{t_{i-1}}^{t_i}\zeta(\tau)d\tau = \sum_{i=1}^{q} e^{-\frac{\kappa(q-i)\lambda_2 T_p}{2q}}\int_{(i-1)T_p/q}^{iT_p/q}\zeta(\tau)d\tau.$$

which leads to

$$\Delta T \leq \frac{1}{\sigma}\sqrt{(n-1)/n}\left(\|\Delta m(0)\|e^{-\frac{\kappa\lambda_2 T_p}{2}} + \frac{T_p}{2q}\sum_{i=1}^{q} e^{-\frac{\kappa(q-i)\lambda_2 T_p}{2q}}\sup_{\tau\in c_i}\zeta(\tau)\right). \tag{6}$$

The upper bound of time error ΔT in Inequality (6) converges to that in Inequality (5) as the parameter q goes to the positive infinity. This implies that a tighter upper bound of time error ΔT can be achieved when the parameter q is relatively large. In practice, the parameter q can be increased gradually in order to obtain the desired estimation for the upper bound of ΔT.

While partitioning the coverage region Ω, the partition bars are not allowed to intersect with each other for achieving the disjoint subregions. Thus a sufficient condition is provided to avoid the collision of partition bars.

Theorem 4.3. *Collision avoidance of partition bars is guaranteed if the following inequality*

$$x_{i+1}(0) > x_i(0) + \kappa \|(e_{i+1} - e_i)^T L\| T_p \left(\|\Delta m(0)\| + \frac{1}{\kappa \lambda_2} \sup_{0 \le s \le T_p} \zeta(s) \right), \quad \forall i \in \mathcal{I}_{n-2}$$

holds, where $\zeta(\tau)$ is given by Equation (4) and e_i denotes the i-th column vector in the n dimensional unit matrix. In addition, L represents the Laplacian matrix of the multi-agent communication network.

Proof. Let $\Delta x_i(t) = x_{i+1}(t) - x_i(t),\ i \in \mathcal{I}_{n-2}$ denote the distance between two adjacent partition bars at time t. Then we have $\frac{d\Delta x_i(t)}{dt} = \dot{x}_{i+1}(t) - \dot{x}_i(t) = \kappa e_i^T L m(t) - \kappa e_{i+1}^T L m(t) = \kappa(e_i - e_{i+1})^T L m(t)$. It follows from $Lm(t) = L\Delta m(t)$ that $\frac{d\Delta x_i(t)}{dt} = -\kappa(e_{i+1} - e_i)^T L \Delta m(t)$. Therefore, we have

$$\begin{aligned}
\Delta x_i(t) &= \Delta x_i(0) - \kappa \int_0^t (e_{i+1} - e_i)^T L \Delta m(\tau) d\tau \\
&= \Delta x_i(0) - \kappa (e_{i+1} - e_i)^T L \int_0^t \Delta m(\tau) d\tau \\
&\ge \Delta x_i(0) - \kappa \|(e_{i+1} - e_i)^T L\| \cdot \left\| \int_0^t \Delta m(\tau) d\tau \right\|
\end{aligned}$$

which leads to

$$\begin{aligned}
\Delta x_i(t) &\ge \Delta x_i(0) - \kappa \|(e_{i+1} - e_i)^T L\| \cdot \int_0^t \|\Delta m(\tau)\| d\tau \\
&\ge \Delta x_i(0) - \kappa \|(e_{i+1} - e_i)^T L\| \cdot \int_0^t \left(\|\Delta m(0)\| e^{-\frac{\kappa \lambda_2 \tau}{2}} + \frac{1}{2} \int_0^\tau e^{\frac{\kappa \lambda_2 (s-\tau)}{2}} \zeta(s) ds \right) d\tau \\
&\ge \Delta x_i(0) - \kappa \|(e_{i+1} - e_i)^T L\| \cdot \int_0^t \left(\|\Delta m(0)\| + \frac{1}{\kappa \lambda_2} \sup_{0 \le s \le \tau} \zeta(s) \right) d\tau \\
&\ge \Delta x_i(0) - \kappa \|(e_{i+1} - e_i)^T L\| T_p \left(\|\Delta m(0)\| + \frac{1}{\kappa \lambda_2} \sup_{0 \le s \le T_p} \zeta(s) \right)
\end{aligned}$$

according to $\left\|\int_0^t \Delta m(\tau)d\tau\right\| \leq \int_0^t \|\Delta m(\tau)\|d\tau$ and Lemma 3.3. To avoid the collision of partition bars, we need to ensure $\Delta x_i(t) > 0, \forall t \in [0, T_p],\ i \in \mathcal{I}_{n-2}$, which can be achieved by the following inequality

$$\Delta x_i(0) - \kappa\|(e_{i+1} - e_i)^T L\|T_p \left(\|\Delta m(0)\| + \frac{1}{\kappa\lambda_2} \sup_{0\leq s\leq T_p} \zeta(s)\right) > 0, \quad \forall i \in \mathcal{I}_{n-2}.$$

The proof is thus completed. □

Remark 4.4. *During the partition, the outmost two partition bars (i.e., x_1 and x_{n-1}) have to avoid the collision against the boundaries $g_a(y)$ and $g_b(y)$, respectively. Then the sufficient conditions for avoiding the boundary collision are given by*

$$x_1(0) > \sup_{0\leq y\leq l} g_a(y) + \kappa\|e_1^T L\|T_p \left(\|\Delta m(0)\| + \frac{1}{\kappa\lambda_2} \sup_{0\leq s\leq T_p} \zeta(s)\right)$$

and

$$\inf_{0\leq y\leq l} g_b(y) > x_{n-1}(0) + \kappa\|e_{n-1}^T L\|T_p \left(\|\Delta m(0)\| + \frac{1}{\kappa\lambda_2} \sup_{0\leq s\leq T_p} \zeta(s)\right),$$

respectively.

Remark 4.5. *For the rectangular region of length l_α and width l_β, when the length of partition bars ϵ goes to zero, the upper bound of ΔT can be estimated by*

$$\Delta T \leq \frac{2l_\beta\bar{\rho}v}{\kappa\sigma\lambda_2} \cdot \frac{n-1}{n} \cdot \left(1 - e^{-\kappa\lambda_2 T_p/2}\right) \leq \frac{2l_\beta\bar{\rho}v}{\kappa\sigma\lambda_2} \cdot \frac{n-1}{n}, \quad \epsilon \to 0$$

Furthermore, if the speed v approaches zero as well, we have

$$0 \leq \Delta T \leq \frac{2l_\beta\bar{\rho}v}{\kappa\sigma\lambda_2} \cdot \frac{n-1}{n} \to 0, \quad v \to 0$$

This implies that the actual sweeping time T approaches the optimal sweeping time T^* as both the speed v and the length of partition bars ϵ go to zero.

Remark 4.6. *During the sweeping process, it is assumed that the inequality $\sigma t \leq m_i(t)$ holds for $t \in [0, T_p]$, $i \in \mathcal{I}_n$, which implies that the sweeping operation lags that of region partition, and the agents do not complete the workload in their respective subregion before the region partition comes to an end.*

Remark 4.7. *It is possible to extend the proposed sweep coverage approach to the surface in the three dimensional Euclidean space. The dynamics of partition bars in the first two dimensions (i.e., xy-plane) can be the same as Equation (1), and it determines the dynamics of partition* *bars in the third dimension (i.e., z-axis) due to the surface constraint. In addition, we can estimate the time error between the actual sweep time and the optimal time for sweeping the surface in the same way.*

5 Numerical Simulations

This section provides numerical examples to validates the DSCA in Table 1 and theoretical results on the time error ΔT. Specifically, 5 mobile agents are connected in a directed chain-like communication network (from the 5-th agent to the 1st agent) to cooperatively sweep the region Ω, which is enclosed by two curves:

$$x_0 = g_a(y) = 0.2 \sin \frac{\pi(y-4)}{3} + 1, \quad x_5 = g_b(y) = 0.2 \sin \frac{\pi(y-4)}{3} + 6$$

and two straight lines: $y = 0$ and $y = 10$. In addition, the distribution density function of workload is described by

$$\rho(x, y) = \frac{3}{2} + \frac{1}{2} \sin \frac{x^2 + y^2}{5}$$

with the upper bound $\bar{\rho} = 2$ and the lower bound $\underline{\rho} = 1$. Other parameters are given as follows: $\kappa = 1$, $\epsilon = 0.01$, $v = 8$, $\sigma = 6$, $q = 10$ and $\lambda_2 = 0.0011$. For simplicity, the Euler method is employed to implement the partition dynamics (1) with a step size of 0.001. Figure 2 presents the cooperative sweep process of 5 mobile agents in the coverage region, where the color bar indicates the workload density ranging from light yellow to dark red. At the initial time $t = 0$, all the partition bars are located at the bottom of the region. Then multi-agent system starts partitioning the whole region and meanwhile the agents are sweeping their own subregions. At $t = 0.5$, part of the region has been partitioned by the trajectories of partition pars, and each agent completes the same workload $\sigma t = 3$ in its own subregion. The completed parts are marked in white. The task of region partition is finished at $t = 1.25$ when all the partition bars arrive at the top of coverage region. Next, the mobile agents continue to sweep their respective subregions. Finally, the sweeping process comes to an end at $t = 2.72$ after cleaning up the workload in the whole region. Since the optimal sweeping time T^* is equal to 2.54, the time error is $\Delta T = 0.18$. According to Theorem 4.2 in this work, the upper bound of time error ΔT is 0.88. In comparison, the upper bound of time error ΔT is 13.29 using Theorem 4.1 in [9]. This demonstrates that the theoretical results in this work can provide a tighter upper bound of time error ΔT in comparison with existing results [9].

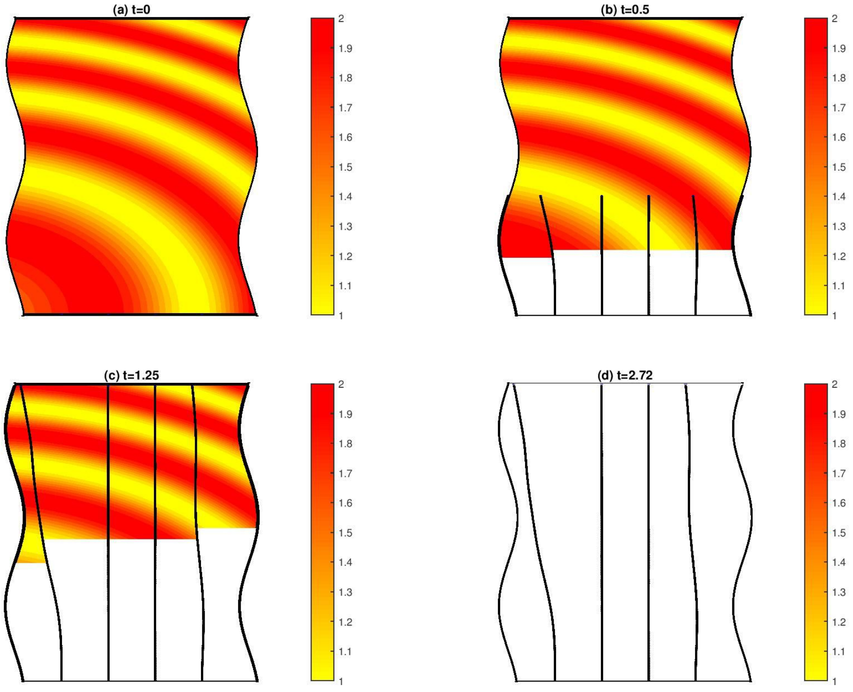

Figure 2: Numerical example on distributed sweep coverage of 5 mobile agents.

6 Conclusions

In this paper, we developed a novel formulation for the sweep coverage of multi-agent systems in the uncertain environment. It has been proved that the dynamics of multi-agent systems is input-to-state stable. Moreover, we obtained the upper bound for the error between the actual sweep time and the optimal sweep time. A sufficient condition was derived to avoid the collision of partition bars in the process of workload partition. Numerical simulations demonstrated the effectiveness of the proposed sweep approach. It is expected that the proposed sweep coverage algorithm can complete the workload in the general region via the coordinate transformation. Future work may include the sweep coverage of an uncertain region with obstacles and the kinematics of multi-agent system with nonholonomic constraints.

Acknowledgment

The authors wish to thank the anonymous reviewers for their constructive comments. This work is partially supported by the Future Resilience System Project at the Singapore-ETH Centre (SEC), which is funded by the National Research Foundation of Singapore (NRF) under its

Campus for Research Excellence and Technological Enterprise (CREATE) program. It is also supported by Ministry of Education of Singapore under Contract MOE2016-T2-1-119.

Appendix A: Proof of Lemma 3.1

It follows from

$$\begin{aligned}\|\Delta m(t)\|^2 &= \|m(t)\|^2 + n\bar{m}(t)^2 - 2\bar{m}(t)\sum_{i=1}^{n} m_i(t)\\ &\le \left(\sum_{i=1}^{n} m_i(t)\right)^2 - n\bar{m}(t)^2 = n(n-1)\bar{m}(t)^2\end{aligned}$$

that $\|\Delta m(t)\| \le \sqrt{n(n-1)} \cdot \bar{m}(t)$, which completes the proof.

Appendix B: Proof of Lemma 3.2

First of all, we split $\|\Delta m(t)\|^2$ into two terms as follows

$$\|\Delta m(t)\|^2 = [\Delta m_k(t)]^2 + \sum_{i=1, i\neq k}^{n} [\Delta m_i(t)]^2, \quad \forall k \in \mathcal{I}_n. \tag{7}$$

For simplicity, define $z_i(t) = \Delta m_i(t)$, and we formulate the constrained optimization problem

$$\min \sum_{i=1, i\neq k}^{n} z_i(t)^2, \tag{8}$$

which is subject to $\sum_{i=1,i\neq k}^{n} z_i(t) = -z_k(t)$, since we have $\sum_{i=1}^{n} z_i(t) = 0$. To solve the optimization problem (8), we introduce the Lagrange function

$$\mathcal{L}(z_1, z_2, ..., z_n, c) = \sum_{i=1, i\neq k}^{n} z_i(t)^2 - c\sum_{i=1}^{n} z_i(t).$$

By solving the system of equations

$$\frac{\partial \mathcal{L}}{\partial c} = \sum_{i=1}^{n} z_i(t) = 0, \quad \frac{\partial \mathcal{L}}{\partial z_i} = 0, \quad i \in \mathcal{I}_n, \quad i \neq k,$$

we obtain $z_i(t) = -\frac{z_k(t)}{n-1}, i \in \mathcal{I}_n, i \neq k$ and the minimum of optimization problem (8)

$$\min \sum_{i=1, i\neq k}^{n+1} z_i(t)^2 = \frac{z_k(t)^2}{n-1},$$

which implies

$$\sum_{i=1,i\neq k}^{n} \left[\Delta m_i(t)\right]^2 \geq \frac{1}{n-1}\left[\Delta m_k(t)\right]^2.$$

Then it follows from Equation (7) that

$$\|\Delta m(t)\|^2 \geq \left[\Delta m_k(t)\right]^2 + \frac{1}{n-1}\left[\Delta m_k(t)\right]^2 = \frac{n}{n-1}\left[\Delta m_k(t)\right]^2, \quad \forall k \in \mathcal{I}_n$$

which indicates

$$|\Delta m_k(t)| \leq \left(\frac{n-1}{n}\|\Delta m(t)\|^2\right)^{\frac{1}{2}}, \quad \forall k \in \mathcal{I}_n.$$

Therefore, we conclude

$$\|\Delta m(t)\|_\infty = \max_{k\in\mathcal{I}_n} |\Delta m_k(t)| \leq \sqrt{(n-1)/n}\cdot\|\Delta m(t)\|.$$

The proof is thus completed.

Appendix C: Proof of Lemma 3.3

The time derivative of $\|\Delta m(t)\|^2$ with respect to the trajectory of multi-agent dynamics (1) is given by

$$\begin{aligned}\frac{d\|\Delta m(t)\|^2}{dt} &= 2\sum_{i=1}^{n}\Delta m_i(t)\left(\dot{x}_i\int_{y-\epsilon}^{y}\rho(x_i,y)dy - \dot{x}_{i-1}\int_{y-\epsilon}^{y}\rho(x_{i-1},y)dy\right)\\ &+ 2v\sum_{i=1}^{n}\Delta m_i(t)\left(\int_{x_{i-1}(y)}^{x_i(y)}\rho(x,y)dx - \frac{1}{n}\int_{g_a(y)}^{g_b(y)}\rho(x,y)dx\right).\end{aligned} \tag{9}$$

The first term in Equation (9) can be further expressed as

$$\begin{aligned}&\sum_{i=1}^{n}\Delta m_i(t)\left(\dot{x}_i\int_{y-\epsilon}^{y}\rho(x_i,y)dy - \dot{x}_{i-1}\int_{y-\epsilon}^{y}\rho(x_{i-1},y)dy\right)\\ &= -\kappa\Delta m(t)^T G(x)L_{n-1}m(t) - v\Delta m(t)^T P(y)\end{aligned}$$

where

$$P(y) = \left(g_a'(y)\int_{y-\epsilon}^{y}\rho(g_a(y),y)dy, 0, ..., 0, -g_b'(y)\int_{y-\epsilon}^{y}\rho(g_b(y),y)dy\right)^T \in R^n$$

and $L_{n-1} \in R^{(n-1)\times n}$ is composed of the first $(n-1)$ rows of Laplacian matrix that represents the communication network of multi-agent systems. In addition, $G(x) \in R^{n\times(n-1)}$ is given by

$$G(x) = \begin{pmatrix} \int_{y-\epsilon}^{y} \rho(x_1, y)dy & 0 & . & . & . \\ -\int_{y-\epsilon}^{y} \rho(x_1, y)dy & \int_{y-\epsilon}^{y} \rho(x_2, y)dy & . & . & . \\ 0 & -\int_{y-\epsilon}^{y} \rho(x_2, y)dy & . & . & . \\ . & . & . & . & . \\ . & . & . & -\int_{y-\epsilon}^{y} \rho(x_{n-2}, y)dy & \int_{y-\epsilon}^{y} \rho(x_{n-1}, y)dy \\ . & . & . & 0 & -\int_{y-\epsilon}^{y} \rho(x_{n-1}, y)dy \end{pmatrix},$$

and it can be rewritten as $G(x) = -J_{-1,n}^T \cdot E_{n-1} \cdot \Lambda(\epsilon, x)$, where $J_{-1,n}$ denotes the n-dimensional Jordan matrix with the diagonal elements being -1, and E_{n-1} is composed of the first $(n-1)$ columns in the n-dimensional unit matrix. Note that $\Lambda(\epsilon, x)$ refers to the $(n-1)$ dimensional diagonal matrix with the i-th diagonal element being $\int_{y-\epsilon}^{y} \rho(x_i, y)dy$, $i \in \mathcal{I}_{n-1}$.

Because $\Delta m(t)^T G(x) L_{n-1} m(t) = \Delta m(t)^T G(x) L_{n-1} \Delta m(t)$, the first term in Equation (9) can be estimated as

$$\begin{aligned} &\sum_{i=1}^{n} \Delta m_i(t) \left(\dot{x}_i \int_{y-\epsilon}^{y} \rho(x_i, y)dy - \dot{x}_{i-1} \int_{y-\epsilon}^{y} \rho(x_{i-1}, y)dy \right) \\ &\leq -\kappa \Delta m(t)^T G(x) L_{n-1} \Delta m(t) - v \Delta m(t)^T P(y) \\ &= -\kappa \lambda_2 \|\Delta m(t)\|^2 - v \Delta m(t)^T P(y) \\ &\leq -\kappa \lambda_2 \|\Delta m(t)\|^2 + v \|\Delta m(t)\| \cdot \|P(y)\| \\ &\leq -\kappa \lambda_2 \|\Delta m(t)\|^2 + v \epsilon \bar{\rho} \sqrt{g_a'(y)^2 + g_b'(y)^2} \cdot \|\Delta m(t)\| \end{aligned}$$

where λ_2 denotes the second smallest eigenvalue of the matrix $\left(G(x)L_{n-1} + L_{n-1}^T G(x)^T\right)/2$ with $x = (x_1, x_2, ..., x_{n-1})$ and $x_i \in [\inf_{0\leq y\leq l} g_a(y), \sup_{0\leq y \leq l} g_b(y)]$, $i \in \mathcal{I}_{n-1}$. For the second term in Equation (9), it follows from Cauchy-Schwarz inequality and Lemma 3.1 that

$$\begin{aligned} &\sum_{i=1}^{n} \Delta m_i(t) \left(\int_{x_{i-1}(y)}^{x_i(y)} \rho(x, y)dx - \frac{1}{n} \int_{g_a(y)}^{g_b(y)} \rho(x, y)dx \right) \\ &\leq \|\Delta m(t)\| \left[\sum_{i=1}^{n} \left(\int_{x_{i-1}(y)}^{x_i(y)} \rho(x, y)dx - \frac{1}{n} \int_{g_a(y)}^{g_b(y)} \rho(x, y)dx \right)^2 \right]^{\frac{1}{2}} \\ &\leq \|\Delta m(t)\| \cdot \sqrt{(n-1)/n} \cdot \bar{\rho} \left[g_b(y) - g_a(y) \right]. \end{aligned}$$

Therefore, the time derivative of $\|\Delta m(t)\|^2$ can be estimated as follows

$$\begin{aligned}\frac{d\|\Delta m(t)\|^2}{dt} &\leq -\kappa\lambda_2\|\Delta m(t)\|^2 + 2v\bar{\rho}\epsilon\sqrt{g_a'(y)^2 + g_b'(y)^2}\cdot\|\Delta m(t)\| \\ &+ 2v\bar{\rho}\sqrt{(n-1)/n}\cdot[g_b(y) - g_a(y)]\cdot\|\Delta m(t)\| \\ &= -\kappa\lambda_2\|\Delta m(t)\|^2 + \zeta(t)\|\Delta m(t)\|.\end{aligned} \tag{10}$$

with $\zeta(t) = 2v\bar{\rho}\left[\epsilon\sqrt{g_a'(y)^2 + g_b'(y)^2} + (g_b(y) - g_a(y))\sqrt{(n-1)/n}\right]$. Solving the above differential inequality yields

$$\|\Delta m(t)\| \leq \|\Delta m(t_0)\|e^{-\frac{\kappa\lambda_2(t-t_0)}{2}} + \frac{1}{2}\int_{t_0}^{t} e^{\frac{\kappa\lambda_2(\tau-t)}{2}}\zeta(\tau)d\tau.$$

The proof is thus completed.

References

[1] W. Ni, and D. Cheng, Leader-following consensus of multi-agent systems under fixed and switching topologies. *Systems & Control Letters*, 59(3-4), 209-217, 2010.

[2] Y. Hong, J. Hu, and L. Gao, Tracking control for multi-agent consensus with an active leader and variable topology, *Automatica*, 42(7): 1177-1182, 2006.

[3] G. Wen, G. Hu, W. Yu, J. Cao, and G. Chen, Consensus tracking for higher-order multi-agent systems with switching directed topologies and occasionally missing control inputs, *Systems & Control Letters*, 62(12), 1151-1158, 2013.

[4] H. T. Zhang, C. Zhai, and Z. Chen, A general alignment repulsion algorithm for flocking of multi-agent systems. *IEEE Transactions on Automatic Control*, 56(2): 430-435, 2011.

[5] H. Su, X. Wang, and Z. Lin. Flocking of multi-agents with a virtual leader, *IEEE Transactions on Automatic Control*, 54(2): 293-307, 2009.

[6] R. Alberton, R. Carli, A. Cenedese, and L. Schenato, Multi-agent perimeter patrolling subject to mobility constraints, *Proceedings of American Control Conference*, pp. 4498-4503, 2012.

[7] I. I. Hussein and D. M. Stipanovic, Effective coverage control for mobile sensor networks with guaranteed collision avoidance, *IEEE Transactions on Control Systems Technology*, 15(4): 642-657, 2007.

[8] C. Song, L. Liu, G. Feng, and S. Xu, Optimal control for multi-agent persistent monitoring, *Automatica*, 50(6): 1663-1668, 2014.

[9] C. Zhai and Y. Hong, Decentralized sweep coverage algorithm for multi-agent systems with workload uncertainties, *Automatica*, 49(7): 2154-2159, 2013.

[10] H. Choset, Coverage for robotics: A survey of recent results, *Annals of Mathematics and Artifical Intelligence*, 31(1): 113-126, 2001.

[11] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, Coverage control for mobile sensing network, *IEEE Trans. Robotics and Automation*, 20(2): 243-255, 2004.

[12] J. Luna, R. Fierro, C. Abdallah, and J. Wood, An adaptive coverage control algorithm for deployment of nonholonomic mobile sensors, *Proceedings of IEEE Conference on Decision and Control*, 1250-1256, Atlanta, GA, USA, 2010

[13] D. W. Gage, Command control for many-robot systems, *Proceedings of Annual AUVS Teachnical Symposium*, pp. 22-24, 1992.

[14] A. Ghosh and S. K. Das, Coverage and connectivity issues in wireless sensor networks: A survey, *Pervasive and Mobile Computing*, 4(3): 303-334, 2008.

[15] S. Kumar, T. H. Lai, and A. Arora, Barrier coverage with wireless sensors, *Wireless Networks*, 13(6): 817-834, 2007.

[16] T. M. Cheng and A. V. Savkin, Decentralized coordinated control of a vehicle network for deployment in sweep coverage, *Proceedings of IEEE International Conference on Control and Automation*, pp. 275-279, 2009.

[17] A. Jeremic and A. Nehorai, Design of chemical sensor arrays for monitoring disposal sites on the ocean floor, *IEEE Journal of Oceanic Engineering*, 23(4): 334-343, 1998.

[18] H. K. Khalil, Nonlinear Systems. Prentice-Hall, New Jersey, 1996.