KYBERNETIKA — VOLUME *50* (2014), NUMBER 1, PAGES 19–31

SWEEP COVERAGE OF DISCRETE TIME MULTI-ROBOT NETWORKS WITH GENERAL TOPOLOGIES

Chao Zhai

This paper addresses a sweep coverage problem of multi-robot networks with general topologies. To deal with environmental uncertainties, we present discrete time sweep coverage algorithms to guarantee the complete coverage of the given region by sweeping in parallel with workload partition. Moreover, the error between actual coverage time and the optimal time is estimated with the aid of continuous time results. Finally, numerical simulation is conducted to verify the theoretical results.

Keywords: sweep coverage, multi-robot networks, discrete time, general topologies

Classification: 93E12, 62A10

1. INTRODUCTION

Multi-robot networks and control have received increasingly attention in a wide variety of applications, which play an important role in distributed collection and control information. Distributed design with advantages such as low cost, reliability, and flexibility provides a feasible way to deploy a large number of networked robots over a region of interest to achieve desired collective tasks. In practice, robots are usually equipped with various sensors. Since a single robot may be difficult to complete the task due to its limited capacities, a group of robots (or viewed as a mobile sensor network) are usually teamed up to complete the tasks by communicating and coordinating their actions through network. Various coordination tasks and algorithms for multiple robots have been reported [6, 10, 15, 17].

Cooperative coverage problems of multiple robots have drawn much attention to the researchers in recent years, particularly for robotic networks [8, 11]. There are various dynamic coverage types including barrier coverage, sweep coverage, and blanket coverage [4]. Different from the set containment to drive the agents to a give set [13, 14], the coverage problem is to design multi-robot motion within a given region. Compared with the static coverage [1, 3], where the Voronoi partition was adopted with the locational cost function as an index to optimize sensor locations, dynamic coverage is considered to allow robots move around to cover the region of interest. Sweep coverage as a dynamic coverage problem is to make a group of robots with sensing capability move across the

DOI: 10.14736/kyb-2014-1-0019

given region to detect targets of interest or complete workload on the area. It is difficult in that all the robots have to cooperate in order to optimize the operation time. [2] proposed a formation-based sweep coverage strategy. Additionally, [16] proposed a sweep coverage algorithm for the rectangular region based on the nearest neighbor's information. However, many problems of the sweep coverage with environmental uncertainties in any bounded region remain to be solved.

The objective of this paper is to investigate the practical implementation of sweep coverage algorithm in the uncertain environment for multi-robot networks with general communication topologies. Although the parameter uncertainty in the environment was studied based on adaptive control to learn environment information online in [11, 12], we consider bounded uncertainties and cannot apply adaptive technique here. Because of those uncertainties, we cannot give a fixed formation-based coverage strategy in advance. Instead, we have to deal with the uncertainty when we carry out the coverage control. Thus, we propose a decentralized sweep coverage algorithm with two combined operations: partition (to handle the uncertainties) and sweep (to complete the coverage). Here a decentralized control technique is adopted to deal with the unknown dynamical environment. Although it is impossible to achieve sweep coverage of the given region with optimal operation time due to the uncertainty, we give the estimation on the difference between the actual coverage time and the optimal time.

The rest of the paper is organized as follows. A discrete time formulation of sweep coverage in uncertain environment is presented in section 2. Then, a decentralized sweep coverage algorithm and the estimation of the extra time are shown in section 3, followed by simulation results in section 4. Finally, conclusions are given in the last section.

2. PROBLEM FORMULATION

In this paper, we consider the practical implementation of sweep coverage algorithm for multi-robot networks in uncertain environment.

To increase the effectiveness in the coverage control, the whole region is divided into several subregions, and each robot is responsible for completing workload (such as dust on the floor) on its own subregion. If there is no uncertainty and workload distribution can be known for all the robots in advance, the optimal strategy to complete the coverage can be carried out by partitioning the whole region into subregions with equal workload for each robot, where the sweeping task can be completed in the shortest time without regard to the shape of subregions. Unfortunately, limited sensing range of each robot and unknown workload distribution make it impossible to get average workload on the whole region for each robot at each moment.

Consider a rectangular region D with width l_a and length l_b (see Figure 1, and only 3 robots are deployed in the plot.). All the robots line up at the left boundary of D, and they will move to the right boundary by sweeping all the rectangular region. Notice that the actuation range of each robot is d, so when it sweeps when it moves, it will clean up a stripe with width d. Thus, the whole region is partitioned into stripes with length l_a and width d, and the robots sweep these stripes one by one to complete the whole region coverage. For convenience, we assume $l_b = qd$ for some integer q. To shorten the whole time for coverage, each stripe is partitioned into sub-stripes for each robot so that each robot has the same workload. Then each robot sweeps its own sub-stripe and

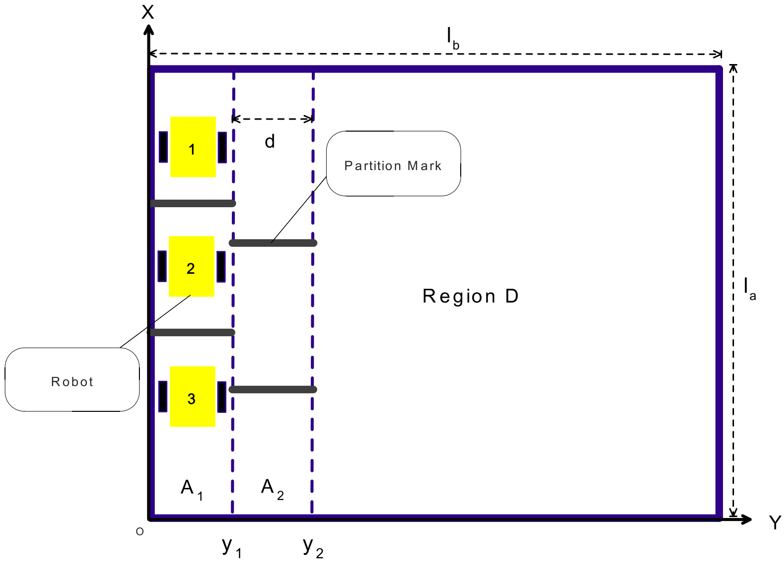

Fig. 1. Sweep coverage of multiple robots in region D.

simultaneously partitions the next stripes for sweep.

The workload distribution on D is denoted by $\rho(x, y)$, which is continuous and always larger than 0 on D. With respect to $\rho(x, y)$, we have the following assumption [16].

Assumption 1. There exist positive constants $\underline{\rho}$ and $\bar{\rho}$ such that

$$\underline{\rho} \leq \rho(x, y) \leq \bar{\rho}.$$

For simplicity, suppose the first stripe has been partitioned into sub-stripes with equal workload for each agent. Moreover, each robot has the same sweeping rate ν. Obviously, the completion time of sweeping each sub-stripe only depends on ν and workload on the sub-stripe. In addition, each robot detects the workload around its position and gets the workload information of its neighbors on the next stripe. When each robot sweeps its own sub-stripe, the partition on the next stripe is conducted according to the proposed partition algorithm. Once all the robots finish sweeping their own sub-stripes, partition operation stops. Then, all the robots move to the newly partitioned stripes and repeat previous operations until the whole sweeping task is completed.

Naturally, the workload on the ith sub-stripe of a stripe (say A_k) is given by

$$m_i^k = \int_{x_{i-1}^k}^{x_i^k} \omega_k(\tau)\, \mathrm{d}\tau$$

where $\omega_k(\tau) = \int_{y_{k-1}}^{y_k} \rho(\tau, y)\, \mathrm{d}y$ with $y_0 = 0$, $y_q = l_b$ and $y_k = y_{k-1} + d$, $1 \leq k \leq q$. Clearly,

x_{i-1}^k and x_i^k are vertical positions of partition marks $i-1$ and i on A_k, respectively (where $i \in E_n = \{1, 2, \ldots, n\}$, $x_0^k = 0$ and $x_n^k = l_a$).

The discrete time dynamics on A_k for partition mark i is given as follows

$$\frac{x_i^k(zT_s + T_s) - x_i^k(zT_s)}{T_s} = u_i(zT_s), \quad i = 1, \ldots, n-1 \tag{1}$$

where T_s denotes the sampling period, $z \in N^+$, and $u_i(zT_s)$ is the partition control input as follows

$$u_i(zT_s) = \kappa \sum_{j \in \mathcal{N}_i} (m_j^k(zT_s) - m_i^k(zT_s)). \tag{2}$$

Here, $\kappa > 0$ is a given constant and $\mathcal{N}_i$ represents the neighbor set of robot i. For simplicity of notation, T_s in brackets is omitted when no confusion is caused. Moreover, Figure 2 describes the implementation of discrete time sweep coverage algorithm for ith robot.

3. MAIN RESULTS

In this section, we give the theoretical analysis for the proposed sweep coverage algorithm. The basic idea is to estimate the extra time for the discrete time sweep coverage algorithm with the help of the continuous time model and some numerical methods since it is difficult to investigate the discrete time model directly (see Figure 3). Here, the extra time denotes the error between actual coverage time and optimal coverage time. The discrete time model can be obtained by discretizing the continuous time model with numerical methods such as Euler method. Moreover, the truncation error caused by the discretization can be easily estimated. Then we can get the upper bound of the extra time for the discrete time model by using the error and results in continuous time model. First of all, the workload partition algorithm for robot i on each stripe is presented as follows: each robot collects the workload information on neighbors'sub-stripes; and all the robots update the position of partition mark i with (1) and (2). Combining (1) and (2), we get the following dynamics of the partition marks:

$$\frac{x_i^k(z+1) - x_i^k(z)}{T_s} = \kappa \sum_{j \in \mathcal{N}_i} (m_j^k(z) - m_i^k(z)), \quad i = 1, \ldots, n-1. \tag{3}$$

Clearly, as the sampling period T_s tends to 0, we get the continuous time partition dynamics of mark i

$$\dot{x}_i^k = \kappa \sum_{j \in \mathcal{N}_i} (m_j^k - m_i^k), \quad i = 1, \ldots, n-1. \tag{4}$$

Notice that the time derivative of m_i^k along (4) can be expressed as

$$\dot{m}_i^k = \omega_k(x_i^k)\dot{x}_i^k - \omega_k(x_{i-1}^k)\dot{x}_{i-1}^k, \quad i \in E_n. \tag{5}$$

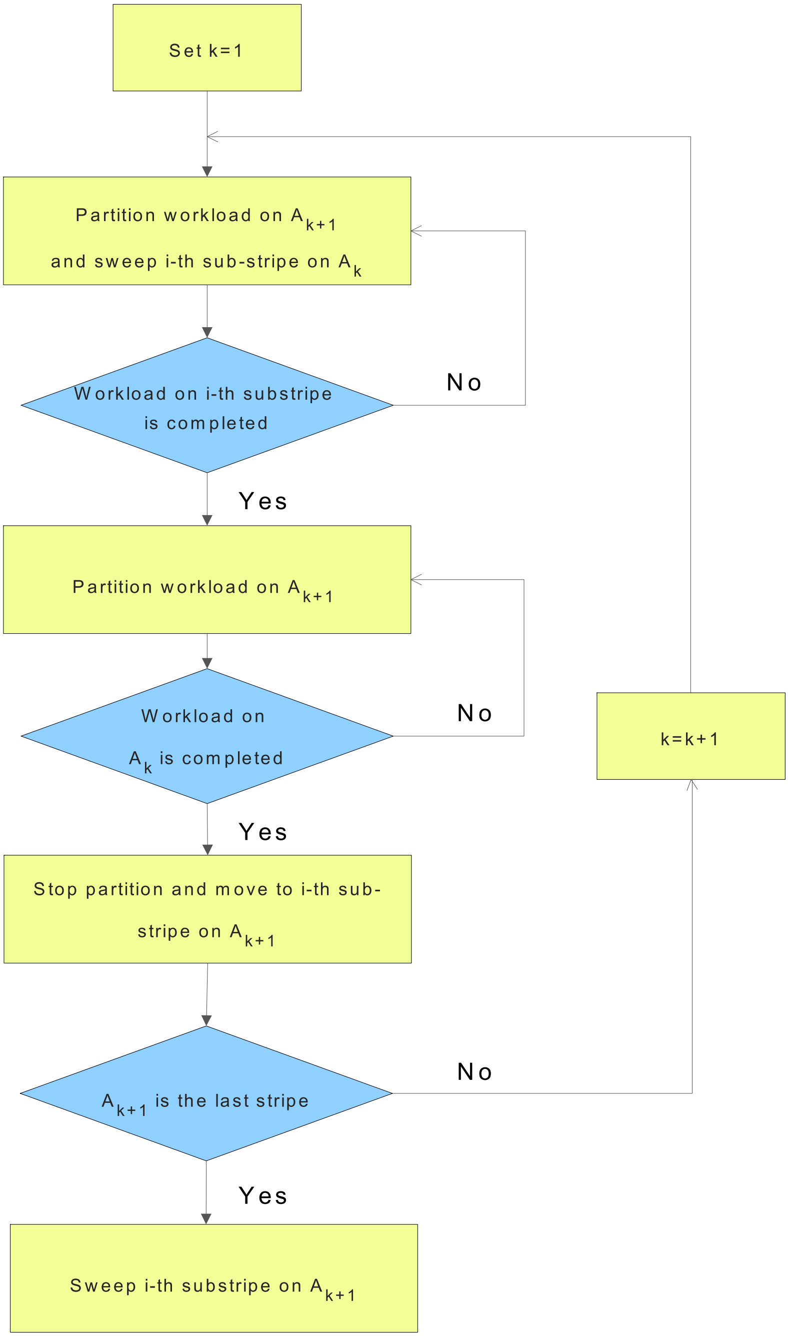

Fig. 2. Flow diagram of discrete time sweep coverage algorithm.

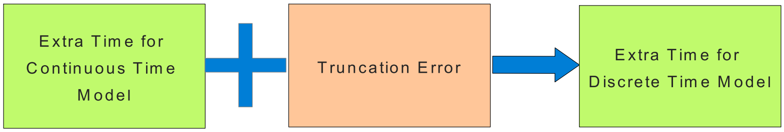

Fig. 3. Schematic illustration on the acquisition of extra time for the discrete time coverage algorithm.

Rewrite the above set of equations in matrix form as follows.

$$\dot{m}^k = \begin{pmatrix} \sigma_1 & 0 & . & . & 0 & 0 \\ -\sigma_1 & \sigma_2 & . & . & 0 & 0 \\ 0 & -\sigma_2 & . & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & -\sigma_{n-2} & \sigma_{n-1} \\ 0 & 0 & . & . & 0 & -\sigma_{n-1} \end{pmatrix} \dot{x}^k$$

where $x^k = (x_1^k, x_2^k, \ldots, x_{n-1}^k)^T$, $m^k = (m_1^k, m_2^k, \ldots, m_n^k)^T$ and $\sigma_i = \omega_k(x_i^k) > 0$.

According to (4), the dynamics (5) can be converted to the following "linear" system with the state m^k as follows.

$$\begin{aligned} \dot{m}_1^k &= -\kappa s_1 \sigma_1 m_1^k + \kappa \sigma_1 \sum_{j \in \mathcal{N}_1} m_j^k \\ \dot{m}_2^k &= -\kappa s_2 \sigma_2 m_2^k + \kappa s_1 \sigma_1 m_1^k - \kappa \sigma_1 \sum_{j \in \mathcal{N}_1} m_j^k + \kappa \sigma_2 \sum_{j \in \mathcal{N}_2} m_j^k \\ &\vdots \\ \dot{m}_{n-1}^k &= -\kappa s_{n-1} \sigma_{n-1} m_{n-1}^k + \kappa s_{n-2} \sigma_{n-2} m_{n-2}^k - \kappa \sigma_{n-2} \sum_{j \in \mathcal{N}_{n-2}} m_j^k + \kappa \sigma_{n-1} \sum_{j \in \mathcal{N}_{n-1}} m_j^k \\ \dot{m}_n^k &= \kappa s_{n-1} \sigma_{n-1} m_{n-1}^k - \kappa \sigma_{n-1} \sum_{j \in \mathcal{N}_{n-1}} m_j^k \end{aligned} \tag{6}$$

where s_i denotes the number of robots in the neighbor set of robot i. Similarly, we rewrite the above equations in a compact form.

$$\dot{m}^k = -\kappa P m^k \tag{7}$$

where P is time-varying matrix based on x^k. Therefore, we will focus on the consensus problem of the time-varying system (7).

Clearly, the dynamical equation (7) can be viewed as a linear time-varying system since the elements of P depending on linear combination of σ_i that is related to x_i^k are time-varying. As for P, we have the following result.

Lemma 3.1.

$$P\mathbf{1} = P^T\mathbf{1} = \mathbf{0}$$

where $\mathbf{1} = (1, 1, \dots, 1)^T \in R^n$ and $\mathbf{0}$ denotes the zero vector with n elements.

Proof. Take $m^k = \mathbf{1}$ and it follows from (6) and (7) that $-\kappa P\mathbf{1} = \mathbf{0}$. Since $\kappa > 0$, $P\mathbf{1} = \mathbf{0}$. Note that $P^T\mathbf{1} = \mathbf{0}$ means the sum of elements in each column of P is 0. Without loss of generality, we consider the sum of elements in the jth column of $P = [p_{ij}]$. Let $\mathcal{N}_{\vec{j}}$ represent the set of robots, which have the common neighbor, robot j. According to (6), we have

$$\sum_{i=1}^{n} p_{ij} = -s_j\sigma_j + s_j\sigma_j + \sum_{i\in\mathcal{N}_{\vec{j}}} (\sigma_i - \sigma_i) = 0$$

for $1 \leq j < n$, and

$$\sum_{i=1}^{n} p_{ij} = \sum_{i\in\mathcal{N}_{\vec{j}}} (\sigma_i - \sigma_i) = 0$$

for $j = n$. Thus, the proof is completed. □

Denote $P_u = \frac{P+P^T}{2}$, and 0 is the eigenvalue of the symmetric matrix P_u with the associated eigenvector $\mathbf{1}$, since $P_u\mathbf{1} = \frac{1}{2}(P\mathbf{1} + P^T\mathbf{1}) = \mathbf{0}$. Then we label n eigenvalues of P_u according to increasing order:

$$0 = \lambda_1(P_u) \leq \lambda_2(P_u) \leq \ldots \leq \lambda_{n-1}(P_u) \leq \lambda_n(P_u).$$

Here we propose another assumption:

Assumption 2. $\lambda_2(P_u) > 0$.

Remark 3.2. The above assumption is not so restrictive, which is related to the connectivity (depending on the workload). If we select $u_i = \kappa(m_{i+1}^k - m_i^k)$ as the control input, the assumption holds immediately. Actually, The interconnection graph of robots in [16] is a directed chain, and the corresponding matrix P_u can be expressed as

$$\begin{pmatrix} \sigma_1 & -\sigma_1 & . & . & 0 & 0 \\ -\sigma_1 & \sigma_1+\sigma_2 & . & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & \sigma_{n-2}+\sigma_{n-1} & -\sigma_{n-1} \\ 0 & 0 & . & . & -\sigma_{n-1} & \sigma_{n-1} \end{pmatrix}.$$

Clearly, P_u has only one zero eigenvalue with the eigenvector $\mathbf{1}$, since rank(P_u)$= n-1$. Furthermore, all other eigenvalues of P_u expect for $\lambda_1(P_u) = 0$ are larger than zero according to Geršgorin disk theorem [7]. The assumption on $\lambda_2(P_u) > 0$ is thus satisfied.

Define

$$\lambda_2 = \min_{x^k \in \Delta^{n-1}} \lambda_2(P_u), \quad \Delta = [0, l_a].$$

Next, we give the result related to workload partition in the uncertain region. The following lemma provides the convergence analysis of workload partition algorithm on the stripe A_k.

Lemma 3.3. Under Assumptions 1 and 2, the plane $\{m^k \in R^n \mid m_1^k = m_2^k = \ldots = m_n^k\}$ of the dynamics (1) with control input (2) is exponentially stable.

Proof. To show the stability, we consider the change of variables $\delta_i = m_i^k - \bar{m}^k, i \in E_n$ and the following lower bounded function

$$H_k = \frac{1}{2}\sum_{i=1}^{n}(m_i^k - \bar{m}^k)^2, \quad \bar{m}^k = \frac{1}{n}\int_0^{l_a} \omega_k(\tau)\,\mathrm{d}\tau.$$

Thus, we have

$$H_k = \frac{1}{2}\sum_{i=1}^{n}\delta_i^2 = \frac{1}{2} \parallel \delta \parallel^2$$

where $\delta = (\delta_1, \delta_2, \ldots, \delta_n)^T$ and $\parallel \cdot \parallel$ denotes the Euclidean norm. Since $\delta = m^k - \bar{m}^k\mathbf{1}$, $P\delta = Pm^k - \bar{m}^k P\mathbf{1} = Pm^k$ according to Lemma 3.1. Moreover, $\dot{\delta} = \dot{m}^k = -\kappa P m^k = -\kappa P\delta$. The time derivative of H_k along the trajectories of (7) is given by

$$\dot{H}_k = \delta^T\dot{\delta} = -\kappa\delta^T P\delta.$$

Due to

$$\delta^T P\delta = \delta^T P^T\delta = \frac{1}{2}\delta^T(P + P^T)\delta = \delta^T P_u\delta,$$

we get

$$\dot{H}_k = -\kappa\delta^T P_u\delta.$$

Further, $\delta^T P_u\delta$ has the following property [7]

$$\min_{\delta\perp\mathbf{1},\delta\neq\mathbf{0}} \frac{\delta^T P_u\delta}{\parallel \delta \parallel^2} = \lambda_2(P_u) \geq \lambda_2.$$

Hence,

$$\dot{H}_k = -\kappa\delta^T P_u\delta \leq -\kappa\lambda_2 \parallel \delta \parallel^2 \leq -2\kappa\lambda_2 H_k.$$

Solving the above differential equation yields

$$H_k(t) \leq H_k(t_0)e^{-2\kappa\lambda_2(t-t_0)}, \tag{8}$$

which implies the conclusion. □

H_k defined in Lemma 3.3 is used to measure the uniformity of workload partition on A_k. Obviously, the smaller H_k is, the more well-proportioned workload partition we will get. In addition, H_k^a and H_k^b denote the uniformity of workload partition at initial and final partition positions on A_k, respectively.

Note that the boundaries between sub-stripes on A_k are exactly the final positions of the partition marks on the stripe. Clearly, the initial positions of partition marks on A_{k+1} is the final partition position, that is, the boundaries between sub-stripes on A_k, respectively. Recalling Lemma 3.3 in [16], we can easily get the following lemma, and the proof is thus omitted here.

Lemma 3.4. H_{k+1}^a and H_k^b satisfy the following inequality

$$H_{k+1}^a \leq \alpha^2 H_k^b + \beta \frac{d^2 l_a^2}{n}, \tag{9}$$

where $\alpha = \frac{\bar{\rho}}{\underline{\rho}}$ and $\beta = \frac{\bar{\rho}^4}{\underline{\rho}^2} - \underline{\rho}^2$.

Let $x^k(t_z)$ and $x^k(z)$ denote the position vector of partition marks on stripe k ($1 \leq k \leq q$) at time t_z in the continuous time formulation and discrete computation, respectively. Then we have the following result.

Lemma 3.5. The global truncation error $e_z^k = x^k(t_z) - x^k(z)$ on stripe k satisfies

$$\|e_z^k\| \leq \frac{(n+2)\kappa\bar{\rho} d l_a}{8}(\gamma - 1)T_s + \gamma\|e_0^k\|$$

where $\gamma = e^{\frac{4(n-1)\kappa d^2 \bar{\rho}^2 l_a}{\nu}}$.

P r o o f. In numerical computation, we adopt Euler method to implement the continuous time partition algorithm, which can be rewritten as follows

$$\begin{aligned}\dot{x}_i^k = \kappa \sum_{j \in N_i}(m_j^k - m_i^k) = \kappa \sum_{j \in N_i}\left(\int_{x_{j-1}^k}^{x_j^k} \omega_k(\tau)\,\mathrm{d}\tau - \int_{x_{i-1}^k}^{x_i^k} \omega_k(\tau)\,\mathrm{d}\tau\right) \\ = f_i(t, x^k), \quad i = 1, 2, \ldots, n-1\end{aligned}$$

with $x^k = (x_1^k, x_2^k, \ldots, x_n^k)^T$. From the mean value theorem, we have

$$f_i(t, x^k) - f_i(t, x^{k^*}) = \sum_{j=1}^n \frac{\partial f_i}{\partial x_j^k}(\xi_j)(x_j^k - x_j^{k^*}).$$

Therefore,

$$\begin{aligned}\left|f_i(t, x^k) - f_i(t, x^{k^*})\right| &= \left|\sum_{j=1}^n \frac{\partial f_i}{\partial x_j^k}(\xi_j)(x_j^k - x_j^{k^*})\right| \\ &\leq \sum_{j=1}^n \left|\frac{\partial f_i}{\partial x_j^k}(\xi_j)\right| \cdot \left|x_j^k - x_j^{k^*}\right|\end{aligned}$$

Define $F^k = (F^k_{ij})_{(n-1)\times n}$ and $F^k_{ij} = \frac{\partial f_i}{\partial x^k_j}(\xi_j)$. Then we get

$$\|f(t,x^k) - f(t,x^{k^*})\|_\infty = \|F^k(\xi)(x^k - x^{k^*})\|_\infty \le \|F^k\|_\infty \cdot \|x^k - x^{k^*}\|_\infty \le L\|x^k - x^{k^*}\|_\infty$$

where

$$f(t,x^k) = (f_1(t,x^k), f_2(t,x^k), \ldots, f_{n-1}(t,x^k))^T$$

and

$$L = \max_\xi \|F^k(\xi)\|_\infty = \max_\xi \left\{ \max_i \sum_{l=1}^{n} \left| \frac{\partial f_i}{\partial x^k_l}(\xi_l) \right| \right\} = 4(n-1)\kappa \max_\xi \omega_k(\bar\xi) = 4(n-1)\kappa d\bar\rho.$$

Then the local truncation error of x^k_i is expressed as

$$d^i_z(T_s) = \frac{1}{2}T_s^2 \frac{\mathrm{d}f_i(t,x^k(t))}{\mathrm{d}t} + o(T_s^3) = \frac{1}{2}T_s^2 \frac{\mathrm{d}f_i(t,x^k(t))}{\mathrm{d}t}|_{t=t^*}$$

where

$$\begin{aligned}\frac{\mathrm{d}f_i(t,x^k(t))}{\mathrm{d}t} &= \kappa \sum_{j\in N_i} (\dot m^k_j - \dot m^k_i)\\ &= \kappa \sum_{j\in N_i} (\omega_k(x^k_j)\dot x^k_j - \omega_k(x^k_{j-1})\dot x^k_{j-1} - \omega_k(x^k_i)\dot x^k_i + \omega_k(x^k_{i-1})\dot x^k_{i-1})\\ &\le (n-1)(n+2)\kappa^2\bar\rho^2 d^2 l_a.\end{aligned}$$

Hence,

$$\|d_z(T_s)\|_\infty \le \frac{(n-1)(n+2)\kappa^2\bar\rho^2 d^2 l_a}{2} T_s^2$$

where $d_z(T_s) = (d^1_z(T_s), d^2_z(T_s), \ldots, d^{n-1}_z(T_s))^T$. Let $D = \frac{(n-1)(n+2)\kappa^2\bar\rho^2 d^2 l_a}{2}$. Then, from Theorem 4.4 in [5], we can conclude that the global truncation error $e^k_z = x^k(t_z) - x^k(z)$ on stripe k satisfies

$$\|e^k_z\| \le \frac{(n+2)\kappa\bar\rho d l_a}{8}(\gamma - 1)T_s + \gamma\|e^k_0\|$$

where $\gamma = e^{\frac{4(n-1)\kappa d^2\bar\rho^2 l_a}{\nu}}$. □

Remark 3.6. In the above lemma, we consider general network topologies, and the bound holds for any communication interconnection between robots. For the certain interconnection network, we can get the tighter bound.

Then, we give the key theoretical result in this paper.

Theorem 3.7. Suppose the Assumptions 1 and 2 hold and the initial stripe is partitioned with equal workload. With the discrete time sweep coverage algorithm, the extra time to sweep D spent more than the optimal coverage time is bounded by

$$\Delta T \le \frac{d l_a}{\nu}\sqrt{\frac{\beta}{n}} \sum_{j=1}^{q-1} j\alpha^{q-1-j} e^{-\lambda_2\kappa(q-j)\underline{t}} + \frac{(n+2)\kappa d^2\bar\rho^2 l_a}{4\nu}\left(\frac{1-\gamma^q}{1-\gamma} - q\right)T_s \tag{10}$$

with $\underline{t} = \frac{d\underline{\rho} l_a}{n\nu}$.

P r o o f. Since robots sweep their respective sub-stripes simultaneously, the time of completing sweeping the stripe depends on the sub-stripe with the most workload. Obviously, optimal partition of the stripe leads to sub-stripes with equal workload. We first estimate the difference between the most workload and average workload on the subregion of each stripe. Then the extra time to sweep each stripe compared to the optimal partition is calculated. Moreover, the error caused by the discretization is also considered. Finally, we obtain the upper bound of the extra time of sweeping D with the proposed discrete time sweep coverage algorithm. From (8) in Lemma 3.3, we have

$$H_k(t) \le H_k^a e^{-2\lambda_2 \kappa t}.$$

The time used to partition the stripe A_{k+1} is exactly the time to sweep the stripe A_k, which has the lower bound $\underline{t} = \frac{d\rho l_a}{n\nu}$. Therefore, we have

$$H_{k+1}^b \le H_{k+1}(\underline{t}) \le H_{k+1}^a e^{-2\lambda_2 \kappa \underline{t}}.$$

Substituting (9) in Lemma 3.4 into the above inequality, we get

$$H_{k+1}^b \le (\alpha^2 H_k^b + \beta \frac{d^2 l_a^2}{n}) e^{-2\lambda_2 \kappa \underline{t}}. \tag{11}$$

Thus,

$$\sqrt{H_{k+1}^b} \le (\alpha \sqrt{H_k^b} + dl_a \sqrt{\frac{\beta}{n}}) e^{-\lambda_2 \kappa \underline{t}}.$$

In addition, since there is no partition operation for the first stripe, $\|e_z^1\| = 0$. Moreover, we have $e_0^{k+1} = e_z^k$ from $x_{k+1}(t_0) = x_k(t_z)$ and $x_{k+1}(0) = x_k(z)$. Therefore, by Lemma 3.5 we get

$$\|e_z^2\| \le \frac{(n+2)\kappa\bar{\rho}dl_a}{8}(\gamma - 1)T_s,$$

$$\|e_z^3\| \le \frac{(n+2)\kappa\bar{\rho}dl_a}{8}(\gamma - 1)T_s + \gamma\|e_0^3\| \le \frac{(n+2)\kappa\bar{\rho}dl_a}{8}(\gamma^2 - 1)T_s.$$

Similarly, we obtain the partition error caused by discretization on strip k as follows

$$\|e_z^k\| \le \frac{(n+2)\kappa\bar{\rho}dl_a}{8}(\gamma^{k-1} - 1)T_s, \quad k = 1, 2, \ldots, q.$$

Since the first stripe is partitioned with equal workload, $H_1^b = 0$ and the extra time to sweep A_1 is 0. Hence, the extra time of sweeping the whole region D is bounded by

$$\begin{aligned}
\Delta T &\le \frac{1}{\nu} \sum_{k=1}^{q} (\sqrt{H_k^b} + 2\|e_z^i\| d\bar{\rho}) \\
&\le \frac{dl_a}{\nu} \sqrt{\frac{\beta}{n}} \sum_{j=1}^{q-1} j\alpha^{q-1-j} e^{-\lambda_2 \kappa (q-j)\underline{t}} + \frac{(n+2)\kappa d^2 \bar{\rho}^2 l_a}{4\nu} (\frac{1-\gamma^q}{1-\gamma} - q) T_s
\end{aligned}$$

which completes the proof of the theorem. □

Remark 3.8. The above sweep coverage algorithm is also applied to the general region in the absence of parallel boundaries. Nevertheless, we need to ensure that partition marks do not intersect the bilateral boundaries of the stripe when the algorithm is implemented. Moreover, for the complex coverage region, we can divide it into several subregions with relatively simple profiles and then complete sweeping each subregion one by one.

Remark 3.9. For the directed chain $\mathcal{G}_m$, the tuning parameter κ should be chosen with the constraint $\kappa < \frac{1}{2d\bar{\rho}T_s}$ to avoid collision between partition marks.

4. SIMULATIONS

In this section, a numerical example has been conducted to verify the above algorithm using Matlab. For simplicity, we consider 4 robots with the interconnection graph $\mathcal{G}_m$ in the rectangular region D. Parameters of D are selected as follows: $l_a = 4$, $l_b = 6$, $d = 1$ (that is $q = 6$), $\kappa = 5$, $\underline{\rho} = 1$, $\bar{\rho} = 2$ and $\rho(x, y) = 0.5\sin(x + y) + 1.5$. In addition, 4 robots with $\nu = 60$ are used to sweep the region, and sampling period T_s is sufficiently small. Initially, the first stripe has been divided into 4 sub-stripes with equal workload. Then sweeping and partition operations are carried out simultaneously. Colored parts denote the region that has been swept, and sub-stripes of the same color are covered by the same robot. Finally, the complete coverage of D is finished with the time 0.15, and the optimal coverage time is 0.14. Hence, the extra time to sweep D by multiple robots with interconnection graph $\mathcal{G}_m$ is 0.01, which is less than the upper bound 0.09, given in the theoretical results (10).

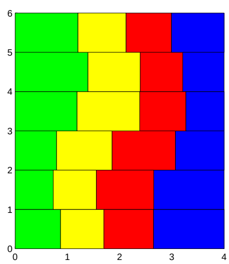

Fig. 4. Sweep process of 4 robots in the region D.

5. CONCLUSIONS

In this paper, a discrete time formulation was proposed to deal with the sweep coverage problem of multiple robots with general communication topologies in an uncertain region. The theoretical analysis was given to estimate the upper bound of the coverage time spent more than the optimal time. Moreover, the numerical result demonstrated the effectiveness of the discrete time sweep coverage algorithm. However, many problems remain to be solved, including the extension of the sweep coverage algorithm to nonholonomic dynamics and tighter estimation of coverage time.

(Received July 18, 2013)

REFERENCES

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

[2] T. M. Cheng and A. V. Savkin: Decentralized coordinated control of a vehicle network for deployment in sweep coverage. In: Proc. IEEE Internat. Conference on Control and Automation, Christchurch 2009.

[3] Q. Du, V. Faber, and M. Gunzburger: Centroidal Voronoi tesseuations: applications and algorithms. SIAM Rev. *41* (1999), 637–676.

[4] D. W. Gage: Command control for many-robot systems. In: Proc. 19th Annual AUVS Teachnical Symposium, Huntsville 1992.

[5] C. W. Gear: Numerical Initial Value Problems for Ordinary Differential Equations. Prentice-Hall, Englewood Cliffs, New Jersey 1971.

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

[7] R. A. Horn and C. R. Johnson: Matrix Analysis. Cambridge University Press, Cambridge 1987.

[8] A. Howard, L. E. Parker, and G. Sukhatme: Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. Internat. J. Robotics Research *25* (2006), 431–447.

[9] J. Hu and G. Feng: Distributed tracking control of leader-follower multi-agent systems under noisy measurement. Automatica *46* (2010), 1382–1387.

[10] W. Ren and R. Beard: Distributed Consensus in Multi-vehicle Cooperative Control. Springer-Verlag, London 2008.

[11] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. Kosmatopoulos: Adaptive-based distributed cooperative multi-robot coverage. In: Proc. American Control Conference, San Francisco 2011.

[12] M. Schwager, D. Rus, and J. J. E. Slotine: Decentralized, adaptive control for coverage with networked robots. Internat. J. Robotics Research *28* (2009), 357–375.

[13] G. Shi and Y. Hong: Global target aggregation and state agreement of nonlinear multi-agent systems with switching topologies. Automatica *45* (2009), 1165–1175.

[14] G. Shi, Y. Hong, and K. Johansson: Connectivity and set tracking of multi-agent systems guided by multiple moving leaders. IEEE Trans. Automat. Control *57* (2012), 663–676.

[15] X. Wang and F. Han: Robust coordination control of switching multi-agent systems via output regulation approach. Kybernetika *47* (2012), 755–772.

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

[17] H. Zhang, C. Zhai, and Z. Chen: A general alignment repulsion algorithm for flocking of multi-agent systems. IEEE Trans. Autom. Control *56* (2011), 430–435.

Chao Zhai, Department of Engineering Mathematics, University of Bristol, Bristol, BS8 1TR. United Kingdom.

e-mail: cz13803@bristol.ac.uk