Distributed PDOP Coverage Control: Providing Large-scale Positioning Service using a Multi-robot System

Liang Zhang[1], Zexu Zhang[1*], Roland Siegwart[2] and Jen Jen Chung[2]

***Abstract*—This manuscript addresses the active positioning service using a multi-robot system (MRS) for providing large-scale coverage and scalability in terms of MRS size. Inspired by the coverage control problems from Wireless Sensor Network (WSN) literature, we propose a gradient-based control method where a *position dilution of precision* (PDOP) field is applied to indicate the performance of the positioning service. We derive the gradient of the PDOP field through rigorous derivation, which can be distributively computed by each robot using only the local information from itself and its neighbors. Extensive simulation results show that our proposed method is robust to initial positions, scalable to MRS size, and can extend the coverage area of the positioning service by leveraging the mobility and coordination within the MRS.**

***Index Terms*—Multi-Robot Systems; Optimization and Optimal Control; Distributed Robot Systems; PDOP field Coverage; Active Positioning Service.**

I. INTRODUCTION

ACCURATE and easy-to-access positioning systems are gradually becoming essential for people's daily life and society in general, such as GNSS (Global Navigation Satellite Systems). However, GNSS suffers from attenuation, distortion, multi-path effects etc. in critical environments, for example, indoors, in tunnels, underwater, or outer space [1]. Therefore, research into large-scale positioning services in GNSS-limited areas is getting growing attention for various possible applications in monitoring and exploration, search and rescue, surveillance, or base construction on the Moon or Mars, and so on [2].

Positioning systems for GNSS-limited areas have been widely developed over the past decades and can be generally categorized according to its coverage size: 1) regional systems, such as the LORAN [3]; 2) local systems, like the Locata [4]; 3) indoor systems, for example, the Vicon [5] and Cricket [6].

However, most existing systems assume that the anchors remain either stationary or keep moving along a regular orbit or a predefined trajectory. Therefore, the underlying positioning system can only provide a passive service, where the users can be positioned only when they are inside the coverage areas. Besides, the quality of the positioning service also cannot be adjusted on-the-fly. Lastly, the possible mission space is constrained by the size of the positioning systems. As a result, state-of-art systems tend to deploy redundant anchors for maintaining an efficient mission space.

In this paper, we put the anchors on mobile robots and present a distributed control solution for dynamically deploying the MRS in GNSS-limited areas to provide active and optimal positioning service. The designed control strategy is formulated and solved by the coverage control scheme from WSN [7], where the positioning performance is indicated as the distribution of the PDOP field over a specified hotspot area. Theoretical derivations show that the gradient of the PDOP field can be computed locally, which results in a gradient-descent-based motion strategy for each robot. Therefore, the MRS can be autonomously maneuvered to an optimal geometric configuration over the given hotspot area.

Innovations of the proposed control method are threefold, 1) greatly extending the coverage by leveraging the mobility and coordination within MRS, 2) adjusting positioning performance on-the-fly and 3) scalability to MRS size.

The main contributions of this paper are: 1) The reformulation of active positioning service into a coverage control problem over a PDOP field; 2) The rigorous derivation of the PDOP gradient both within the coverage dominance and on the boundary over a single robot's position.

A. Related work

Positioning systems: Existing regional and local systems mainly attempt to either enhance GNSS by applying pseudolites or provide additional location information by incorporating new measurements. A pseudolite is a generator and transmitter that can receive and augment GNSS signals on either ground-based [4], [8], [9] or airborne-based [10], [11] platforms. Consequently, the signal from a pseudolite can be orders of magnitude stronger than from a satellite and hence it can overcome the attenuation caused by obstacles in the

Manuscript received: Sep. 30th, 2020; Revised Dec. 2nd, 2020; Accepted Feb, 3th, 2021.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Natural Science Foundation of China under Granted (61374213,61573247) and the Open Foundation from Shanxi Key Laboratory of Integrated and Intelligent Navigation (SKLIIN-20180208). The authors gratefully acknowledge financial support from China Scholarship Council.

* Corresponding author, zexuzhang@hit.edu.cn

[1] Liang Zhang and Zexu Zhang are with the Deep Space Exploration and Research Center, School of Astronautics, Harbin Institute of Technology, Harbin 150001, China. Liang Zhang is also a visiting student in Autonomous System Lab, ETH Zürich 8092, Switzerland when developing part of this paper. Zexu Zhang is also with the Shaanxi Key Laboratory of Integrated and Intelligent Navigation. liangzhangprc@gmail.com

[2] Roland Siegwart and Jen Jen Chung are with the Autonomous Systems Lab, ETH Zürich, Zürich 8092, Switzerland. {rsiegwart;chungj}@ethz.ch

Digital Object Identifier (DOI): see top of this page.

environment. Regarding indoor systems, different techniques have been developed over the past decades. The employed signal technologies involve radio frequency devices [12] (e.g. WLAN, WIFI, Bluetooth, ultrawideband and RFID), ultrasound [13], infrared [14], IMU, vision-based systems, magnetic fields and visible light [15]. As a consequence, the physical characteristics include internodal range, signal frequency and strength, fingerprint, acceleration, angle and so on. Besides, optimal deployments for stationary pseudolites have been investigated in [16]–[18]. However the final solutions are derived case-by-case with respect to the number of pseudolites. Such deployment strategies are usually centralized onto a ground station or a central node, and hence are not robust and not scalable.

PDOP in GNSS: DOP is a concept that is tightly related to GNSS, but extendable to any range-based positioning solution [16], [19]. The most commonly used one is the Geometric DOP (GDOP), which states how errors in the pseudorange measurement will affect the final state estimation [20], [21]. Mathematically, GDOP is composed of both PDOP and time DOP (TDOP). While PDOP reveals the scaling effect of 3d relative positions, the TDOP accounts for how the time synchronization matters in state estimation.

In practice, the positioning accuracy of range-based systems is affected by many factors, such as the atmosphere refraction, multi-path effects, geometry, clock offset, etc. While others depend on either environment or devices, only the geometry (indicated by PDOP) is fully controllable. Therefore, seeking a lower PDOP over hotspot areas becomes the primary motivation of the active positioning service.

Coverage control: Each sensor in a WSN aims at maximizing the probability of detecting an event happening within the mission space. While finding the optimal global coverage configuration is NP hard [22], recent works focus on local solutions by utilizing greedy gradient descent along a proper coverage objective [7], [23]. The prevailing objectives are the expected sensing cost of event locations or the joint event detection probability of sensors [7], [24]–[26]. As most works assume that every sensor detects the event independently, the derivation of gradient over each sensor's location can be also independently extracted from the objectives.

However, the PDOP reflects the relative formation of the MRS with respect to users and hence the robot positions are no longer independent. Therefore, the computation of PDOP involves several complicated matrix operations and hence is more difficult for analysis. Besides, most existing coverage control problems mainly focus on 2D applications with either a fixed or unlimited covering radius. We extend the benchmark by deploying the robots into 3D mission space. Though the coverage area is still on a 2D plane, the covering radius varies with the altitude of the flying robots.

II. Problem Formulation

As shown in Fig. 1, we consider a large-scale *mission space* $\boldsymbol{F} \subset \mathbb{R}^3$ in 3D space as a vertical prism generated by a convex polygon on the ground plane $\boldsymbol{\Omega} \subset \mathbb{R}^2$ i.e. the x-y plane (gray area in Fig. 1). In order to provide a positioning service for possible clients on the ground plane, a MRS is deployed. The position of the i-th robot is thereby a 3D vector which is denoted as $\boldsymbol{S}_i = [s_{ix}, s_{iy}, s_{iz}]^T, \boldsymbol{S}_i \in \boldsymbol{F}$. When the number of mobile robots is N, their joint location vector is defined as $\boldsymbol{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_N]^T$.

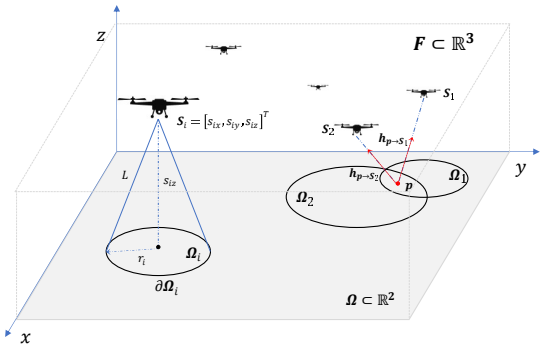


Fig. 1. Mission space with notation definitions.

The MRS is deployed to guard or monitor the mission space in the long term. Therefore, the MRS is able to build a precise map of the environment in advance utilizing SLAM and other coordination techniques. As a result, these robots can localize themselves even without GNSS. However, the ground user $\boldsymbol{p} \in \boldsymbol{\Omega}$ may be a temporary robot that is traversing across or intruding into the mission space. The MRS is tasked to localize the user using range-based measurements. If the user is a cooperative unit, the MRS becomes a set of pseudolites and provides positioning service in a similar way as GNSS. For the non-cooperative user, passive range sensors (e.g. stereo vision or radar) can be utilized and the user is localized at the "MRS-end". As the mapping and self-localization steps are not our focus, we assume here that each robot is aware of its own position $\boldsymbol{S}_i$ exactly. We leave the problem of an inexact self-localization with uncertainties as future research.

Constrained by the practical capacities of sensors and flying platforms, we investigate a coverage model as ground circle $\boldsymbol{\Omega}_i \subset \boldsymbol{\Omega}$ whose center is projected from the i-th robot with a non-fixed and limited radius r_i. Applying the prevailing notation in geometry, we use the prefix ∂ to denote the boundary of a set, so that $\partial\boldsymbol{\Omega}_i$ is the boundary of $\boldsymbol{\Omega}_i, \forall \boldsymbol{S}_i \in \boldsymbol{S}$. In practice, the coverage area for a sensor is usually and directly relevant to its height s_{iz}. Therefore, we uniformly denote the coverage radius as a function of the robot's altitude i.e.

$$r_i = r(s_{iz}). \tag{1}$$

For example, Fig. 1 shows a reciprocal model with respect to the flight altitude s_{iz},

$$r_i = \sqrt{L^2 - s_{iz}^2}, \tag{2}$$

where L is the maximum communication distance and thus also the maximum relative-distance measurement. This reciprocal model in (2) shows that our coverage model is not ideal but limited by practical constraints. Note that any other differentiable models can also be used to derive the distributed control strategy in the following sections.

To evaluate the service quality for the ground user, we borrow the concept of PDOP from the scope of GNSS research [21]. Considering the user $\boldsymbol{p}$ in Fig. 1, let's denote

$\boldsymbol{N}_p^k := \{\boldsymbol{S}_i | \boldsymbol{p} \in \boldsymbol{\Omega}_i\}$ as the observed set of flying robots from point $\boldsymbol{p}$ at time step k and $n_{\boldsymbol{p}}^k$ the size of $\boldsymbol{N}_p^k$. After augmenting the user position $\boldsymbol{p}$ into a 3D point $\bar{\boldsymbol{p}} = [\boldsymbol{p}^T, 0]^T$, the unit direction vector $\boldsymbol{h}_i$ from user $\boldsymbol{p}$ to the i-th observed pseudolite is

$$\boldsymbol{h}_i = \boldsymbol{h}_{\boldsymbol{p}\to\boldsymbol{S}_i} = \frac{\bar{\boldsymbol{p}} - \boldsymbol{S}_i}{||\bar{\boldsymbol{p}} - \boldsymbol{S}_i||_2}. \tag{3}$$

Stacking all the unit vectors with $\forall \boldsymbol{S}_i \in \boldsymbol{N}_p^k$ derives,

$$\boldsymbol{H} = \left[\boldsymbol{h}_1^T \cdots \boldsymbol{h}_{n_{\boldsymbol{p}}^k}^T\right]^T. \tag{4}$$

Then the quality of positioning information provided by the MRS is indicated by the PDOP, which is computed directly from $\boldsymbol{H}$

$$PDOP(\boldsymbol{p}, \boldsymbol{S}) = tr\left(\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\right), \tag{5}$$

where $tr(\cdot)$ computes the trace of an input matrix. Notice that PDOP is a coefficient enlarging the noise of the relative-distance measurements and feeding it into the uncertainty of the user's location. Therefore, seeking a smaller PDOP becomes the main motivation to optimally arrange the MRS.

Instead of serving a single point on the ground plane, this paper focuses on providing a positioning service over a specific area, called hotspot coverage. The objective of this paper is to design a transition policy for all robots, so that whenever a hotspot is given within the mission space the MRS can be autonomously deployed to provide an optimal PDOP field over the hotspot.

III. Coverage control for PDOP field

In order to specify the hotspot areas, a density function $R(\boldsymbol{p}) : \Omega \to \mathbb{R}$ captures the importance of providing a better positioning service to any point on the ground plane i.e. lower PDOP. We assume that the density function is positive and finite for all points on the x-y plane, i.e. $R(\boldsymbol{p}) > 0$ and $\int_{\boldsymbol{\Omega}} R(\boldsymbol{p}) d\boldsymbol{p} < \infty, \forall \boldsymbol{p} \in \boldsymbol{\Omega}$. Typically, the Gaussian density function is used,

$$R(\boldsymbol{p}) = exp(-\alpha||\boldsymbol{p} - \boldsymbol{p}_d||_2), \tag{6}$$

where $\boldsymbol{p}_d$ is the central point and α is the parameter determining the decay rate of the density distribution.

We reformulate the active positioning service problem into maximizing the following locational optimization function:

$$\mathscr{H}(\boldsymbol{S}) = \int_{\boldsymbol{\Omega}} R(\boldsymbol{p}) f\left(PDOP(\boldsymbol{p}, \boldsymbol{S})\right) d\boldsymbol{p}, \tag{7}$$

where $f(\cdot)$ is a continuous and monotonically decreasing function that converts a lower PDOP to a larger value. The output from $f(\cdot)$ is therefore called a general PDOP value.

However, before developing our main control law, we need to further investigate the PDOP value computed from (5) as it tends to be singular for some special configurations of the MRS. Observing from (3)-(4), the size of the collected measurement matrix $\boldsymbol{H}$ is $n_{\boldsymbol{p}}^k \times 3$, whereby the size of $\boldsymbol{H}^T\boldsymbol{H}$ is exactly 3×3. However, if the number of observed robots from the ground point $\boldsymbol{p}$ is less than 3, i.e. $n_{\boldsymbol{p}}^k < 3$, or those unit vectors in $\boldsymbol{H}$ are linearly dependent so that $det(\boldsymbol{H}^T\boldsymbol{H}) < 3$, then the inverse operation in (5) is impossible and tends to infinity. Therefore, in order to deal with this problem, we separate the underlying ill-constructed configurations from the normal cases and once these singular configurations are detected, the general PDOP value computed from $f(\cdot)$ is assigned to a constant, i.e. let

$$\bar{f}(\cdot) = \begin{cases} a, & \text{if } n_{\boldsymbol{p}}^k = 1 \\ b, & \text{if } n_{\boldsymbol{p}}^k = 2 \\ c, & \text{if } n_{\boldsymbol{p}}^k \geq 3 \text{ but } det(\boldsymbol{H}^T\boldsymbol{H}) < 3 \\ f(\cdot), & \text{otherwise,} \end{cases} \tag{8}$$

where a, b, c are three positive constants and we may want the relationship as $a < b < c$ because more observed robots is always better for lowering its PDOP in future.

Note that the function $\bar{f}$ is non-continuous at some special MRS configurations due to the limited coverage range for each robot. More descriptions on designing $f(\cdot)$ will be provided in Section V. To simplify the following derivation, we just ignore the derivative operation for those special cases and set them to zeros. Only the last term in (8) is considered i.e.

$$\partial\bar{f}(\cdot) = \begin{cases} 0, & \text{if } n_{\boldsymbol{p}}^k = 1, 2 \\ 0, & \text{if } n_{\boldsymbol{p}}^k \geq 3 \text{ but } det(\boldsymbol{H}^T\boldsymbol{H}) < 3 \\ \partial f(\cdot), & \text{otherwise.} \end{cases} \tag{9}$$

Therefore, the modified utility of coverage control problem can be reformulated into

$$\mathscr{H}(\boldsymbol{S}) = \int_{\boldsymbol{\Omega}} R(\boldsymbol{p}) \bar{f}\left(PDOP\left(\boldsymbol{p}, \boldsymbol{S}\right)\right) d\boldsymbol{p}. \tag{10}$$

As a result, to solve the original active positioning service problem, we try to find a distributed control protocol for every mobile robot $\boldsymbol{S}_i$, such that an optimal configuration of the MRS can be reached for the simultaneous coverage utility (10) when the hotspot density distribution is given to all robots. This can be achieved by solving the following optimization:

$$\begin{aligned} \max_{\boldsymbol{S}} & \int_{\boldsymbol{\Omega}} R(\boldsymbol{p}) \bar{f}\left(PDOP\left(\boldsymbol{p}, \boldsymbol{S}\right)\right) d\boldsymbol{p} \\ \text{s.t.} \quad & \boldsymbol{S}_i \in \boldsymbol{F}, && i = 1, \cdots, N \\ & r_i = r(s_{iz}), && i = 1, \cdots, N \end{aligned} \tag{11}$$

IV. Distributed control method

Moving on, let's denote $\boldsymbol{N}_{\boldsymbol{S}_i}^k := \{\boldsymbol{S}_j | j \neq i, \boldsymbol{\Omega}_j \cap \boldsymbol{\Omega}_i \neq \varnothing\}$ the set of neighboring robots whose coverage areas intersect with the coverage of the i-th robot at time step k. We present an important assumption as prerequisite in the following,

Assumption 1: Every robot $\boldsymbol{S}_i$ can always obtain its set of neighboring robots $\boldsymbol{N}_{\boldsymbol{S}_i}^k$.

Note that this assumption is general in WSN research [25] and can be distributively established if each robot additionally equips a communication device that is responsible exclusively for exchanging positions among all the pseudolites. Then if the communication range of this device is larger than the sum of maximum coverage ranges of the two adjacent robots, the neighboring set $\boldsymbol{N}_{\boldsymbol{S}_i}^k$ can be formed by simply collecting all the received messages from the communication device on robot i.

Inspired by existing coverage control problems [25], [26], a gradient-based controller is implemented in this work. According to the gradient descent updating rule, the next position of robot i can be determined by

$$\boldsymbol{S}_i^{k+1} = \boldsymbol{S}_i^k + \lambda_k \frac{\partial\mathscr{H}(\boldsymbol{S})}{\partial\boldsymbol{S}_i^k}, \tag{12}$$

where k is the time step and λ_k is the step size [7].

A. Gradient derivation

Given a ground point $\boldsymbol{p}$, the positions of robots are coupled with each other when computing the PDOP value at $\boldsymbol{p}$. However, the partial derivative of the objective function over a single robot only exists within the coverage area of this robot, as the PDOP value at an outside-coverage ground point is irrelevant to the robot's position. Consequently, although the objective function in (11) cannot be partitioned according to the "dominance region" of each robot as [23], such a partition is applicable to its partial derivatives as

$$\frac{\partial \mathscr{H}(\boldsymbol{S})}{\partial \boldsymbol{S}_i} = \frac{\partial}{\partial \boldsymbol{S}_i}\left[\int_{\boldsymbol{\Omega}_i} R(\boldsymbol{p})\bar{f}\left(PDOP(\boldsymbol{p},\boldsymbol{S})\right)d\boldsymbol{p}\right]. \tag{13}$$

However, this derivative is still non-trivial as the integrand and dominant of the integration are functions of $\boldsymbol{S}_i$. In order to evaluate it, we apply the well-known Leibnitz rule for differentiating an integral and a "physical" extension in [27]. From an overview perspective, we have

$$\begin{aligned}\frac{\partial \mathscr{H}(\boldsymbol{S})}{\partial \boldsymbol{S}_i} &= \int_{\boldsymbol{\Omega}_i} R(\boldsymbol{p})\frac{\partial \bar{f}\left(PDOP(\boldsymbol{p},\boldsymbol{S})\right)}{\partial \boldsymbol{S}_i}d\boldsymbol{p} \\ &+ \int_{\partial\boldsymbol{\Omega}_i} R(\boldsymbol{p})\bar{f}\left(PDOP(\boldsymbol{p},\boldsymbol{S})\right)\boldsymbol{v}\cdot\boldsymbol{n}dw,\end{aligned} \tag{14}$$

where the newly introduced variables in the second term on the right hand side are illustrated in Fig. 2. Specifically, dw is the arc length element around a boundary point $\boldsymbol{p}$ and variables $\boldsymbol{n},\boldsymbol{v}$ are the unit outward normal and the velocity of $\boldsymbol{p}$ disturbed by the movement of $\boldsymbol{S}_i$, respectively.

The expansion in (14) contains two main parts, which are the gradients within the coverage area and on the boundary. They are discussed separately in the following.

Gradient within coverage area: According to the chain rule, the derivative of $\bar{f}$ over robot i's position is

$$\frac{\partial \bar{f}\left(PDOP(\boldsymbol{p},\boldsymbol{S})\right)}{\partial \boldsymbol{S}_i} = \frac{\partial \bar{f}}{\partial PDOP}\frac{\partial PDOP(\boldsymbol{p},\boldsymbol{S})}{\partial \boldsymbol{S}_i}. \tag{15}$$

After incorporating the modifications in (9), the term $\frac{\partial \bar{f}}{\partial PDOP}$ is general if a continuous and differentiable function f in (10) is selected. Moving on, the following facts [28] about matrix calculations are presented to help further derivations:

$$\frac{\partial}{\partial x}f(\boldsymbol{A}(x)) = tr\left[\frac{\partial f}{\partial \boldsymbol{A}}\frac{\partial \boldsymbol{A}}{\partial x}\right] \tag{16}$$

$$\frac{\partial}{\partial \boldsymbol{A}}tr(\boldsymbol{A}) = \boldsymbol{I} \tag{17}$$

$$\frac{\partial}{\partial x}\boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1}\left(\frac{\partial \boldsymbol{A}}{\partial x}\right)\boldsymbol{A}^{-1} \tag{18}$$

Therefore, the chain rule can again be applied to the calculation of $\frac{\partial PDOP(\boldsymbol{p},\boldsymbol{S})}{\partial \boldsymbol{S}_i}$ after recalling the computation of PDOP in (5):

$$\begin{aligned}\frac{\partial PDOP(\boldsymbol{p},\boldsymbol{S})}{\partial \boldsymbol{S}_i} &= \frac{\partial tr\left[\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\right]}{\partial \boldsymbol{S}_i} \\ &= tr\left[\frac{\partial tr\left[\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\right]}{\partial\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}}\frac{\partial\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}}{\partial \boldsymbol{S}_i}\right] \\ &= tr\left[-\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\frac{\partial \boldsymbol{H}^T\boldsymbol{H}}{\partial \boldsymbol{S}_i}\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\right].\end{aligned} \tag{19}$$

Now let's consider a ground point $\boldsymbol{p}\in\boldsymbol{\Omega}_i$. To derive the value of the above equation, the observed set $\boldsymbol{N}_{\boldsymbol{p}}^k$ is needed to form the joint measurement matrix $\boldsymbol{H}$ by (3)-(4). Taking

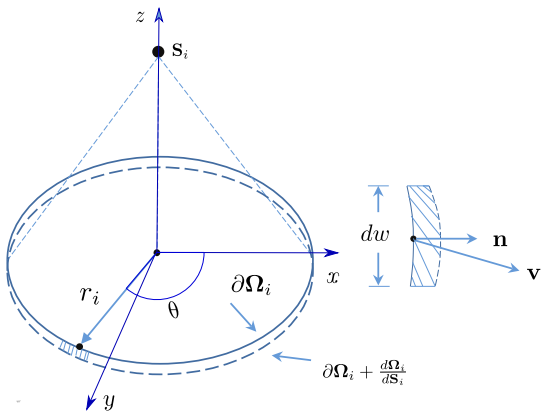


Fig. 2. Illustration of the gradient on the boundary.

Assumption 1 and recalling the definition of the neighboring set $\boldsymbol{N}_{\boldsymbol{S}_i}^k$, it is easy to get the following relationship,

$$\boldsymbol{N}_{\boldsymbol{p}}^k \subset \boldsymbol{N}_{\boldsymbol{S}_i}^k \bigcup \boldsymbol{S_i}. \tag{20}$$

Therefore, we know that the calculation of $\boldsymbol{H}$ can be completed based on local information available to robot i given Assumption 1 and hence the derivative in (19) can also be computed $\forall \boldsymbol{p}\in\boldsymbol{\Omega}_i$ in a decentralized manner.

Next, let's discuss the term $\frac{\partial \boldsymbol{H}^T\boldsymbol{H}}{\partial \boldsymbol{S}_i}$ in (19). We have

$$\frac{\partial \boldsymbol{H}^T\boldsymbol{H}}{\partial \boldsymbol{S}_i} = \frac{\partial \boldsymbol{h}_1^T\boldsymbol{h}_1}{\partial \boldsymbol{S}_i} + \frac{\partial \boldsymbol{h}_2^T\boldsymbol{h}_2}{\partial \boldsymbol{S}_i} + \cdots + \frac{\partial \boldsymbol{h}_{n_{\boldsymbol{p}}^k}^T\boldsymbol{h}_{n_{\boldsymbol{p}}^k}}{\partial \boldsymbol{S}_i} = \frac{\partial \boldsymbol{h}_i^T\boldsymbol{h}_i}{\partial \boldsymbol{S}_i}. \tag{21}$$

Based on the definition of unit vector $\boldsymbol{h}_i$ in (3) and denoting $\boldsymbol{p} = [p_x, p_y]^T$, we can obtain,

$$\frac{\partial \boldsymbol{h}_i^T\boldsymbol{h}_i}{\partial s_{ix}} = \frac{1}{||\tilde{\boldsymbol{p}}-\boldsymbol{S}_i||_2^2}\begin{bmatrix}2(s_{ix}-p_x) & s_{iy}-p_y & s_{iz}\\ s_{iy}-p_y & 0 & 0\\ s_{iz} & 0 & 0\end{bmatrix}, \tag{22}$$

$$\frac{\partial \boldsymbol{h}_i^T\boldsymbol{h}_i}{\partial s_{iy}} = \frac{1}{||\tilde{\boldsymbol{p}}-\boldsymbol{S}_i||_2^2}\begin{bmatrix}0 & s_{ix}-p_x & 0\\ s_{ix}-p_x & 2(s_{iy}-p_y) & s_{iz}\\ 0 & s_{iz} & 0\end{bmatrix}, \tag{23}$$

$$\frac{\partial \boldsymbol{h}_i^T\boldsymbol{h}_i}{\partial s_{iz}} = \frac{1}{||\tilde{\boldsymbol{p}}-\boldsymbol{S}_i||_2^2}\begin{bmatrix}0 & 0 & s_{ix}-p_x\\ 0 & 0 & s_{iy}-p_y\\ s_{ix}-p_x & s_{iy}-p_y & 2s_{iz}\end{bmatrix}, \tag{24}$$

where $\tilde{\boldsymbol{p}}$ is the augmented 3D position of $\boldsymbol{p}$ through $\tilde{\boldsymbol{p}} = [\boldsymbol{p}^T 0]^T$. As a result, the first term on the right hand side of (14) can be computed by:

$$\begin{aligned}&\int_{\boldsymbol{\Omega}_i} R(\boldsymbol{p})\frac{\partial \bar{f}\left(PDOP(\boldsymbol{p},\boldsymbol{S})\right)}{\partial \boldsymbol{S}_i}d\boldsymbol{p} = \\ &\begin{bmatrix}\int_{\boldsymbol{\Omega}_i} -R(\boldsymbol{p})\frac{\partial \bar{f}}{\partial PDOP}tr\left[\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\frac{\partial \boldsymbol{h}_i^T\boldsymbol{h}_i}{\partial s_{ix}}\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\right]d\boldsymbol{p}\\ \int_{\boldsymbol{\Omega}_i} -R(\boldsymbol{p})\frac{\partial \bar{f}}{\partial PDOP}tr\left[\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\frac{\partial \boldsymbol{h}_i^T\boldsymbol{h}_i}{\partial s_{iy}}\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\right]d\boldsymbol{p}\\ \int_{\boldsymbol{\Omega}_i} -R(\boldsymbol{p})\frac{\partial \bar{f}}{\partial PDOP}tr\left[\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\frac{\partial \boldsymbol{h}_i^T\boldsymbol{h}_i}{\partial s_{iz}}\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\right]d\boldsymbol{p}\end{bmatrix}.\end{aligned} \tag{25}$$

Gradient on the boundary: To compute the second term in (14), the two newly introduced variables $\boldsymbol{n},\boldsymbol{v}$ are the key factors to be explained. Given a boundary point $\boldsymbol{p}\in\partial\boldsymbol{\Omega}_i$, the relationship between this 2D boundary position and the robot's 3D position $\boldsymbol{S}_i$ can be rewritten in polar coordinates:

$$p_x = r_i\cos(\theta) + s_{ix}, \tag{26}$$

$$p_y = r_i\sin(\theta) + s_{iy}. \tag{27}$$

Vector $\boldsymbol{n}$ is the unit outward normal. Therefore, its expression can be derived by rotating a tangent vector at $\boldsymbol{p}$ backward. From the relationship in (26)-(27), the tangent vector can be calculated by $\frac{1}{dw}[dp_x, dp_y]^T = \frac{1}{dw}[-r_i\sin(\theta)d\theta, r_i\cos(\theta)d\theta]^T$. Then the normal vector $\boldsymbol{n}$ can be computed as:

$$\boldsymbol{n} = \frac{1}{dw}\begin{bmatrix} dp_y \\ -dp_x \end{bmatrix} = \frac{d\theta}{dw}\begin{bmatrix} r_i\cos(\theta) \\ r_i\sin(\theta) \end{bmatrix}. \tag{28}$$

Regarding the variable $\boldsymbol{v}$, its physical meaning is the velocity of a boundary point $\boldsymbol{p}$ caused by the movement of $\boldsymbol{S}_i$. Therefore, this velocity can be deduced by taking the derivative of boundary points over the robot's position, which yields

$$\boldsymbol{v} = \begin{bmatrix} \frac{\partial p_x}{\partial s_{ix}} & \frac{\partial p_y}{\partial s_{ix}} \\ \frac{\partial p_x}{\partial s_{iy}} & \frac{\partial p_y}{\partial s_{iy}} \\ \frac{\partial p_x}{\partial s_{iz}} & \frac{\partial p_y}{\partial s_{iz}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \frac{\partial r}{\partial s_{iz}}\cos(\theta) & \frac{\partial r}{\partial s_{iz}}\sin(\theta) \end{bmatrix}. \tag{29}$$

Substituting both (28) and (29) into (14) and replacing the integration domain from the coverage boundary $\boldsymbol{\Omega}_i$ with the relative angle θ through the relationship in (26)-(27), the second term on the right hand side can be rewritten as

$$\int_{\partial\boldsymbol{\Omega}_i} R(\boldsymbol{p})\bar{f}\left(PDOP(\boldsymbol{p},\boldsymbol{S})\right)\boldsymbol{v}\cdot\boldsymbol{n}dw = \int_0^{2\pi} R(\boldsymbol{p}(\theta))\bar{f}\left(PDOP(\boldsymbol{p}(\theta),\boldsymbol{S})\right)\begin{bmatrix} r_i\cos(\theta) \\ r_i\sin(\theta) \\ \frac{\partial r_i}{\partial s_{iz}} \end{bmatrix} d\theta, \tag{30}$$

where the notation $\boldsymbol{p}(\theta)$ represents the ground points on the boundary line denoted by the relative angle θ in (26)-(27).

B. Distributed control law

Given the derivations in the previous subsection, one can immediately complete the main control law for updating each robot's position locally. The structure of the final motion strategy is (12) with the gradient computed by (14) whose sub-items are derived through (25) and (30). This is the standard process of gradient descent update. However, practical applications reveal that it tends to obtain an extremely small gradient when the robots are far away from the density center. As a result, robots that follow the update rule in (12) cannot quickly move toward the desired service center when their initial positions are sparsely distributed. In order to overcome this problem, we instead apply the direction of gradient and a fixed step size λ to update the robot positions. Let's denote $\Delta\boldsymbol{S}_i^k = \frac{\partial\mathcal{H}(\boldsymbol{S})}{\partial\boldsymbol{S}_i^k}$, then the gradient-based update rule in (12) is slightly modified into

$$\boldsymbol{S}_i^{k+1} = \boldsymbol{S}_i^k + \lambda\frac{\Delta\boldsymbol{S}_i^k}{||\Delta\boldsymbol{S}_i^k||_2}. \tag{31}$$

We use the same approach as [26] for the computation of integrals in (25) and (30), where a numerical integration method is used to approximate the integrals over the ground plane and boundary by discretizing the integral domains into meshes and line segments and then summing up all the integrand values that are multiplied by the mesh size or segment length. It is clear that the motion strategy calculation for robot i only depends on the neighboring set $\boldsymbol{N}_i^k$. As under Assumption 1, this set can be derived in a distributed manner.

Algorithm 1: Controller for robot i at timestep k

Result: $\boldsymbol{S}_i^{k+1}$
Data: $\boldsymbol{S}_i^k, \boldsymbol{N}_i^k, \lambda, W, U$;
1 Discretize the ground plane into W^2 meshes and the boundary into U segments and represent each mesh and segment by its central point $\boldsymbol{p}_m$ and $\boldsymbol{p}_s$;
2 Collect meshes within radius (1) of robot i into the set $\boldsymbol{\Omega}_i$. Determine the set $\partial\boldsymbol{\Omega}_i$ from (26)-(27).
3 For each point $\boldsymbol{p}_m$ in $\boldsymbol{\Omega}_i$, compute the integrand (25) using $\boldsymbol{S}_i^k$ and $\boldsymbol{N}_i^k$. After multiplying by the mesh size, the gradient within the covered area is computed by summing up all the integrand values.
4 For each point $\boldsymbol{p}_s$ in $\partial\boldsymbol{\Omega}_i$, compute the integrand (30) using $\boldsymbol{S}_i^k$ and $\boldsymbol{N}_i^k$. After multiplying by the segment length, the gradient on the boundary is computed by summing up all the integrand values.
5 Combing steps 3 and 4 together, $\Delta\boldsymbol{S}_i^k$ is derived;
6 Given a step size λ, update $\boldsymbol{S}_i^{k+1}$ using (31) ;
7 **Return** $\boldsymbol{S}_i^{k+1}$

Thus, the proposed control law in (31) is also distributed. Given W, U, the tunable resolution parameters for ground plane and boundary discretization, respectively, the controller for the i-th robot at time step k is summarized in Algorithm 1.

The analysis of convergence time for the proposed control algorithm can be generally divided into two phases. The first is the time that each robot arrives at the density center, which we call the *arriving time*. The other is the time it takes the pre-arrived robots to react to the introduction of new robots, i.e. the *reaction time*. Although it is tricky to figure out the reaction phase, the arriving time is, however, simple and can be approximated and estimated by treating the fixed update step size λ as the traversing velocity of the robot.

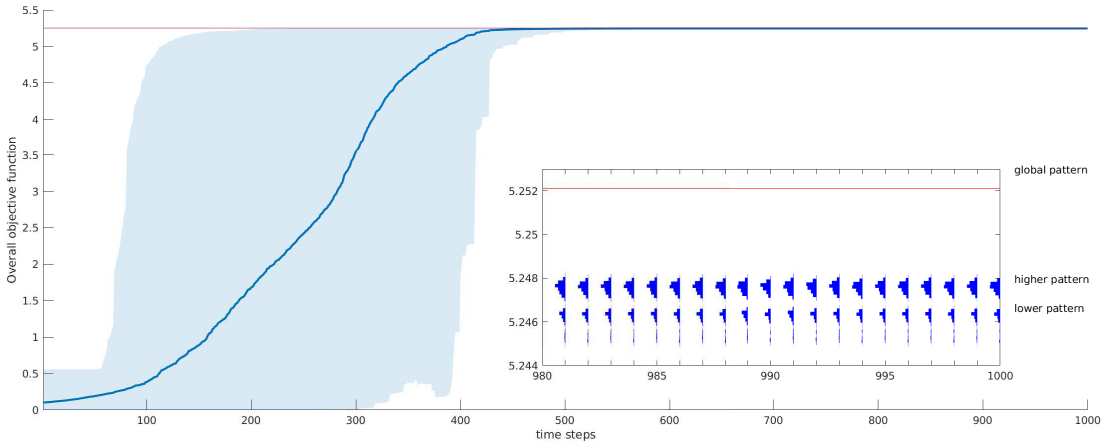
Similar to the discussion in [26], the worst-case computation of the proposed control law is of order $O\left(3N^2(W^2+U)\right)$, where $O(3N^2)$ is responsible for the most expensive operation of matrix multiplication in (5). The worst case assumes the jointly covered area intersected by all robots is nonempty and therefore the matrix $\boldsymbol{H}$ in (4) is of N rows. However, as the size of $\boldsymbol{H}^T\boldsymbol{H}$ is merely 3×3, the complexity of its inversion is omitted.

Finally, a complete 2D application, e.g. robots operating on the ground plane, can be easily achieved from our proposed method by eliminating the elements corresponding to the z axis in (25) and (30).

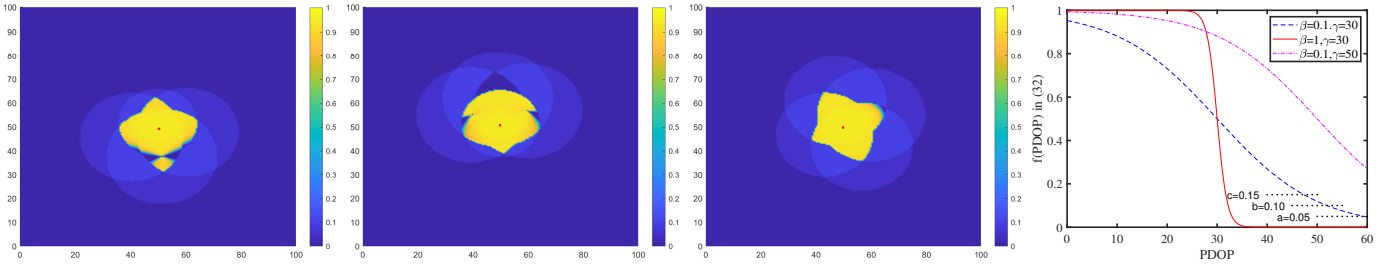
V. Simulation and Discussion

To evaluate the performance of our proposed control law, we present three groups of simulations[1]. The first tests a given density function and random initial positions of robots to validate correctness and efficiency of the distributed control scheme. Then the size of the MRS is intermittently changed to test its robustness to node failures and the introduction of new robots. Lastly, a long-term simulation with changing

[1]More details about these simulations can be found in the supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors, including three video clips corresponding to the three simulations.



(a) PDOP performance computed by (10) over 1000 time steps for 100 trials. The red line denotes the global optimal. The blue line is the mean while the shaded area contains all objectives from 100 trials at the corresponding time step.



(b) The higher pattern (c) The lower pattern (d) The pattern with global optimality (e) The parameters in $\bar{f}(PDOP)$

Fig. 3. Statistical results for four robots serving a fixed density. (b,c,d) are the distributions of $\bar{f}(\cdot)$ for each pattern. Red dots are the maximum point

density distributions is provided to test the system's ability to dynamically track and serve different target areas.

Consider the mission space $\boldsymbol{F} := \{[x, y, z]^T | 0 \le x, y \le 100, 0 \le z \le 20\}$ and the ground plane $\boldsymbol{\Omega}$ is defined as the x-y plane in $\boldsymbol{F}$. The coverage model used in this section is the reciprocal one in (2) whose maximum communication and relative distance measurement range with the user is chosen as $L = 20m$. We design a function $f(\cdot)$ in (10) from the sigmoid function,

$$f(x) = \frac{1}{1 + e^{\beta(x-\gamma)}}, \quad \frac{\partial f}{\partial x} = \beta f(x)\big(f(x) - 1\big). \tag{32}$$

Therefore, there are five parameters to define $\bar{f}(\cdot)$, which are respectively β, γ in (32) and a, b, c in (8). Their effects on $\bar{f}(\cdot)$ are shown in Fig. 3(e). In principle, γ can be treated as a threshold that separates the normal configurations of MRS from singular cases, while the rate β should be relatively small to avoid sharply decreasing as illustrated by the red solid line in Fig. 3(e). Since $f(x)|_{x=\gamma} = \frac{1}{2}$, a, b, c for singular cases should be far smaller than this value while keeping the relationship of $a < b < c$. In the following simulations, these parameters are set to $\beta = 0.1$, $\gamma = 30$, $a = 0.05$, $b = 0.1$, $c = 0.15$.

We choose the Gaussian density in (6) to represent the hotspot area to be served by the MRS. The decay rate depends on the specific task and is fixed to $\alpha = 1$ in the following simulations. The center of density can be changed if different hotspots should be covered for better positioning service. The resolutions U, W affect the computational complexity and the accuracy of numerical integration. Theoretically, larger values of U, W always result in better integration accuracy but worse run time. In the following simulations, they are set to $W = 100, U = 200$. Concerning the fixed step size λ, one should guarantee that the robots can reach the desired density center with the velocity given by λ. In the following, we set $\lambda = 0.1$ for the simulations in Sections V-A and V-B. Then it is changed to $\lambda = 0.2$ in Section V-C to speed up the robots for the last simulation.

A. Fixed-size MRS serving a static hotspot density

This set of simulation experiments tested four robots with random initial positions tasked with serving a fixed density distribution. We conducted 100 trials in total and each trial lasted for 1000 time steps. During each simulation, the initial positions of the robots were randomly generated within the mission space, and then each robot applied Algorithm 1 to compute its motion strategy. The hotspot density center in the Gaussian function (6) was $\boldsymbol{p}_d = [50, 50]^T$.

The statistical results are shown in Fig. 3. To better demonstrate the efficiency of Algorithm 1, we also derived a global optimal solution by repeatedly calling a coarse-to-fine search algorithm and finally returning a solution with the maximum objective score as the "global optimality". Its coverage performance is presented as the red upper line. As we can see from the distribution, all 100 trials finally converge to local solutions whose performances are close to the global bound. Notice that there are various equilibria and we can observe from the results in Fig. 3(a) that most trials converge to two clusters, which we call the "higher pattern" and the "lower pattern", respectively. Two representative cases are

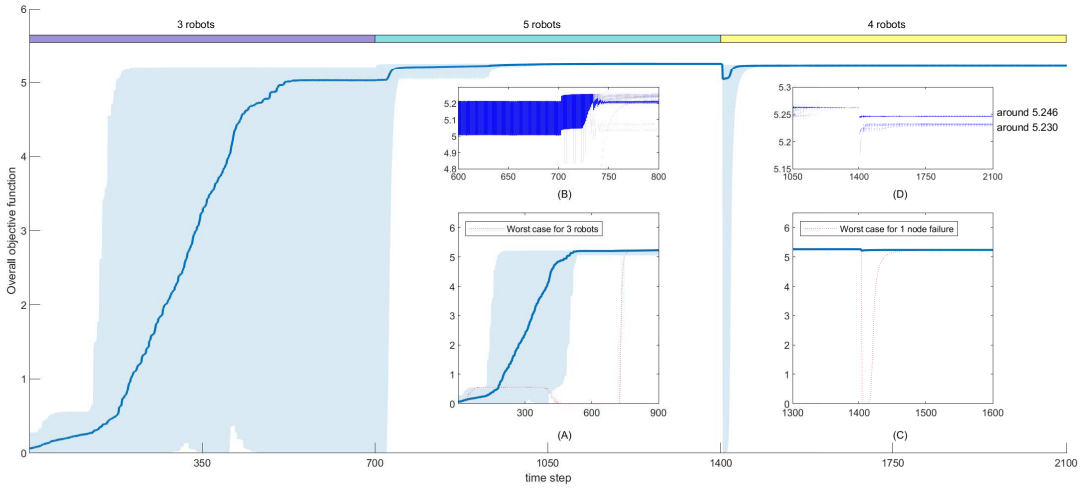
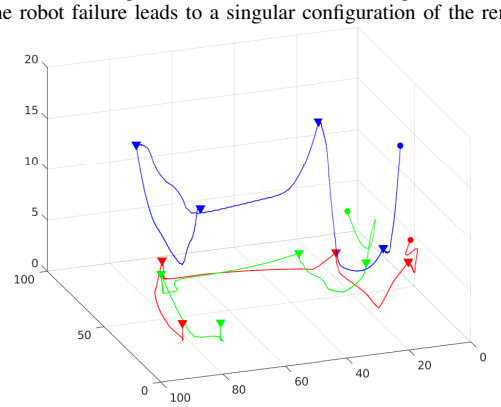
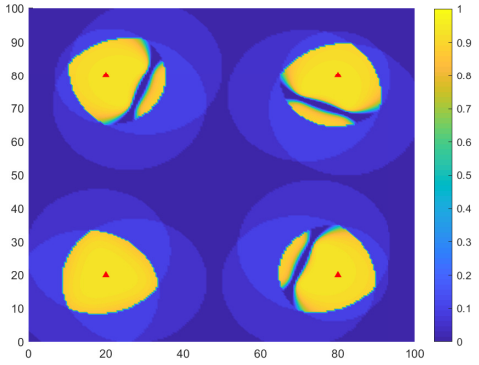


Fig. 4. Distribution of the overall PDOP performance computed by (10) for the node failure and recovery simulation over 30 trials. The insets unveil more details of the experiment. (A) shows the distribution of the performance using 3 robots to serve a fixed density after excluding the worst case. The system performance after introducing two more robots and removing one robot are presented in (B) and (C), respectively. (C) also applies the exclusion of the worst case, where the robot failure leads to a singular configuration of the remaining MRS. (D) shows the final equilibria over 30 trials.



(a) Trajectories for three robots. The spherical markers are their initial locations and the triangles represent the equilibria achieved for each hotspot density.



(b) $\bar{f}(PDOP)$ distributions when the robots reach their stable coverage positions for the four hotspot density functions. Red triangles indicate density centers.

Fig. 5. Simulation results for the time-varying hotspot densities over 2000 steps, whose centers are switched every 500 steps. The sequence of centers is $d_{c1} = [20,20]^T$, $d_{c2} = [80,20]^T$, $d_{c3} = [80,80]^T$, $d_{c4} = [20,80]^T$. These centers are also the points with maximum $\bar{f}(\cdot)$ values

selected and their corresponding coverage maps for $\bar{f}(PDOP)$ are shown in Figs 3(b) and (c). Compared against the global solution in Fig. 3(d), we see that the local equilibria cause coverage fractures on the map and hence perform worse than the global solution.

Another interesting observation here is that no trial converged to the global solution. However, as we will illustrate in the third simulation, we note that our proposed algorithm is still capable of achieving global optimality in the presence of appropriate initialization.

B. Resilience to robot failure and changing MRS size

This set of simulations tested the scalability of our control law with respect to MRS size. The simulation setup follows:

1) Initially three robots are randomly deployed in the mission space to serve a static density.
2) After reaching to the equilibrium within 700 steps, two new nodes are introduced into the mission space, whose starting positions are randomly generated.
3) Lastly after another 700 steps, one of the five nodes is randomly removed from the MRS to demonstrate stochastic robot failure.

This was repeated for 30 trials and the overall performance is shown in Fig. 4. Notice that the results within the first 700 steps are actually a preservation and extension of the previous results to 3 robots. Differing from the 4-robot case, it suffers from a worst trial where all robots finally converge to a singular equilibrium such that the overall objective is almost zero i.e. no ground point is effectively served by the MRS. Apart from this special case, however, Fig. 4(A) shows that most trials are efficiently stabilized.

From Fig. 4(B), the introduction of two new robots after the 700-th step helps enhance the performance of the positioning service. Even the singular case is quickly promoted to the main distribution. After that the average performance of the 5 robots increases with the distribution tightening until next event. Generally speaking, the removal of a robot lowers the service quality as seen from both the overall distribution and Fig. 4(C). However, it also shows that the failure of a robot does not lead to severe reactions from the remaining MRS, which may be explained by the local optimality of the derived equilibria. Fig. 4(D) shows us the reallocation of robots after node failure further degrades the performance of our control scheme as some trials assemble into a poorer position than the

"lower pattern" shown in Fig 3(a).

C. Dynamically tracking hotspots

In this final set of simulation experiments we evaluated three robots dynamically serving four hotspots in a time sequence. A single demonstration is given in Fig. 5. It shows that the designed control protocol is capable of following the hotspot transitions during service. However, it also reveals that our method can only achieve a local equilibrium due to the separation between two PDOP areas that is caused by the co-planar ground positions with three robots. Note that the equilibrium formed by 3 robots when serving the first density is actually equal to the global solution.

VI. CONCLUSIONS

We have presented a decentralized solution to guide a MRS in 3D mission space for the purpose of providing a positioning service over a designated coverage area where performance is qualified by the PDOP field. Our proposed algorithm implements the rigorous derivations used to obtain the gradients of the PDOP field over each individual robot to apply a gradient descent update control law for each respective robot. Since our proposed method is distributed, from both theoretical analysis and simulation demonstrations, we show that it is (i) robust to the initial positions of the robots, (ii) scalable to system size and hence is robust to node failure and robot recovery, and (iii) adjustable to the convergence time by tuning the parameter λ in Algorithm 1.

However, this method still suffers from some drawbacks. Firstly, current analysis is completely based on perfect knowledge of each robot's individual location. How to integrally analyze the influence of a robot's self-localization uncertainties in the active positioning service is still very challenging. Further, the current control scheme assumes that all robots have access to the hotspot distribution, which is challenging to implement in reality, especially for a decentralized system. Finally, the designed control scheme is constrained by local optimality. From our statistical results in Sections V-A and V-B, the local equilibria for larger team sizes is close to the global solution. Our results have also shown that the difference in performance is negligible in general. However, a singular equilibrium is still able to stabilize the 3-robot MRS as shown in Fig 4(A).

Real communication devices also tend to suffer from limited energy and bandwidth resources across the network. Therefore, another important issue in practical applications is the impact of imperfect communication, such as network delay and packet loss. Thus, a robot may not immediately detects its neighbors and receive their latest positions at decision time (when computing the gradient). As a result, the positioning performance may deteriorate in terms of convergence and ability to instantly react to changes in the high-level task. More investigations are needed to analyze how packet loss and network delay affect the performance of convergence and to design more robust control strategies. In addition, the MRS is exposed to collision risks under a severe network delay or large rate of packet loss, especially during the reaction phase. Guaranteed collision-free strategies should be considered for real experiments.

REFERENCES

[1] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 131–149, 2013.

[2] R. Mautz, "The challenges of indoor environments and specification on some alternative positioning systems," in *2009 6th Workshop on Positioning, Navigation and Commun.* IEEE, 2009, pp. 29–36.

[3] L. Tetley and D. Calcutt, *Electron. Navigation Syst.* Routledge, 2007.

[4] J. Barnes, C. Rizos, M. Kanli, A. Pahwa, D. Small, G. Voigt, N. Gambale, and J. Lamance, "High accuracy positioning using Locata's next generation technology," in *Proc. of the 18th Int. Tech. Meeting of the Satellite Division of The Inst. of Navigation (ION GNSS 2005)*, 2005, pp. 2049–2056.

[5] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A study of vicon system positioning performance," *Sensors*, vol. 17, no. 7, p. 1591, 2017.

[6] N. B. Priyantha, "The cricket indoor location system," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[7] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Syst. Mag.*, vol. 27, no. 4, pp. 75–88, 2007.

[8] C. Kim, H. So, T. Lee, and C. Kee, "A pseudolite-based positioning system for legacy GNSS receivers," *Sensors*, vol. 14, no. 4, pp. 6104–6123, 2014.

[9] L. Dai, J. Wang, C. Rizos, and S. Han, "Pseudo-satellite applications in deformation monitoring," *GPS Solutions*, vol. 5, no. 3, pp. 80–87, 2002.

[10] K. Lee, H. Noh, and J. Lim, "Airborne relay-based regional positioning system," *Sensors*, vol. 15, no. 6, pp. 12 682–12 699, 2015.

[11] G. Novakovic, A. Dapo, and D. Kodzic, "Pseudolite applications in positioning and navigation," *Int. Multidisciplinary Scientific GeoConference: SGEM*, vol. 1, p. 829, 2010.

[12] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern., Part C (Appl. and Rev.)*, vol. 37, no. 6, pp. 1067–1080, 2007.

[13] S. Holm, "Hybrid ultrasound-RFID indoor positioning: Combining the best of both worlds," in *2009 IEEE Int. Conf. on RFID*. IEEE, 2009, pp. 155–162.

[14] J. M. Kahn and J. R. Barry, "Wireless infrared communications," *Proc. of the IEEE*, vol. 85, no. 2, pp. 265–298, 1997.

[15] T.-H. Do and M. Yoo, "An in-depth survey of visible light communication based positioning systems," *Sensors*, vol. 16, no. 5, p. 678, 2016.

[16] B. Li, A. G. Dempster, J. Wang *et al.*, "3D DOPs for positioning applications using range measurements," *Wireless Sensor Netw.*, vol. 3, no. 10, p. 334, 2011.

[17] B. W. Parkinson and K. T. Fitzgibbon, "Optimal locations of pseudolites for differential GPS," *Navigation*, vol. 33, no. 4, pp. 259–283, 1986.

[18] M. H. Saka, "Optimal positioning of pseudolites augmented with GPS," *Sea Technol.*, vol. 49, no. 8, pp. 39–42, 2008.

[19] R. Yarlagadda, I. Ali, N. Al-Dhahir, and J. Hershey, "GPS GDOP metric," *IEE Proc. - Radar, Sonar and Navigation*, vol. 147, no. 5, pp. 259–264, 2000.

[20] S. Frattasi and F. Della Rosa, *Mobile Positioning and Tracking: From Conventional to Cooperative Techniques*. John Wiley & Sons, 2017.

[21] R. B. Langley, "Dilution of precision," *GPS World*, vol. 10, no. 5, pp. 52–59, 1999.

[22] N. Megiddo and K. J. Supowit, "On the complexity of some common geometric location problems," *SIAM J. on Comput.*, vol. 13, no. 1, pp. 182–196, 1984.

[23] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, 2004.

[24] Ö. Arslan, "Statistical coverage control of mobile sensor networks," *IEEE Trans. Robot.*, vol. 35, pp. 889–908, 2019.

[25] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2445–2455, 2011.

[26] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European J. of Control*, vol. 11, no. 4-5, pp. 436–463, 2005.

[27] H. Flanders, "Differentiation under the integral sign," *The American Math. Monthly*, vol. 80, no. 6, pp. 615–627, 1973.

[28] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," http://www2.imm.dtu.dk/pubdb/edoc/imm3274.pdf, 2012.