

# Multi-Robot Dynamical Source Seeking in Unknown Environments

Bin Du<sup>†</sup>, Kun Qian<sup>‡</sup>, Hassan Iqbal<sup>‡</sup>, Christian Claudel<sup>‡</sup>, and Dengfeng Sun<sup>†</sup>

**Abstract**— This paper presents an algorithmic framework for the distributed on-line source seeking, termed as **DoSS**, with a multi-robot system in an unknown dynamical environment. Our algorithm, building on a novel concept called *dummy confidence upper bound* (D-UCB), integrates both estimation of the unknown environment and task planning for the multiple robots simultaneously, and as a result, drives the team of robots to a steady state in which multiple sources of interest are located. Unlike the standard UCB algorithm in the context of multi-armed bandits, the introduction of D-UCB significantly reduces the computational complexity in solving subproblems of the multi-robot task planning. This also enables our **DoSS** algorithm to be implementable in a distributed on-line manner. The performance of the algorithm is theoretically guaranteed by showing a sub-linear upper bound of the cumulative regret. Numerical results on a real-world methane emission seeking problem are also provided to demonstrate the effectiveness of the proposed algorithm.

## I. INTRODUCTION

Over the past decades, source seeking has been a fundamentally crucial problem and attracted increasing attentions, due to its various applications including surveillance [1], [2], environment and health monitoring [3]–[6], to name a few. Source seeking involves locating one or several positions, associated with measurements maxima, in a possibly unknown and noisy environment. In this paper, we are particularly interested in solving the problem of source seeking with a multi-robot system, in which a team of robots are deployed and expected to cooperatively locate as many local maxima sources as possible, by leveraging the communications among different robots. In addition, we consider that the environment is not only unknown but also changing dynamically as the robots acquire knowledge from it. Under this circumstance, the team of robots needs to track the moving sources in real-time. We remark that these two settings, i.e., the multi-robot system and dynamical environment, make our problem significantly challenging to solve.

A large number of algorithms have been developed in the literature [6]–[10] for solving the source seeking problem within different setups. In particular, a very recent approach devises an on-line scheme that suits the need of source seeking with mobile robots. This approach is called **AdaSearch**, and proposed in [10], by leveraging the notion of UCB in the

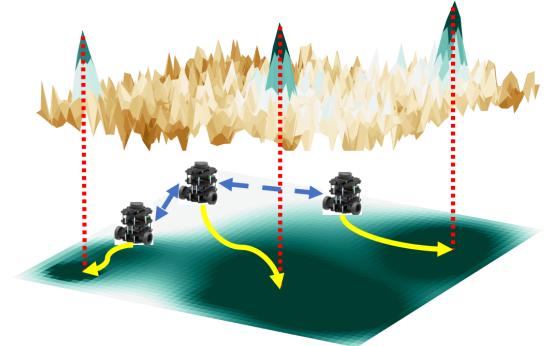


Fig. 1: Visualization of the **DoSS** setup: the lower layer corresponds to the unknown environment that needs to be explored; the upper layer depicts the D-UCB which determines the multi-robot's task planning. Robots exchange information with their immediate neighbors and cooperatively estimate the unknown environment.

study of multi-armed bandits problems. This **AdaSearch** algorithm maintains a set of candidate points which are likely to be the sources of interest, and let the robot repeat a predetermined trajectory so that it can adaptively collect information from the unknown environment and iteratively update the candidate set. Consequently, the robot will be able to eventually identify the desired sources after sufficient information is acquired. However, we should remark that there are two potential drawbacks of the **AdaSearch** scheme: i) it requires the robot to follow the pre-determined trajectory, which might be inefficient at the later stage of the algorithm; and ii) it is not applicable in our source seeking problem setup when considering the multi-robot system and dynamical environment.

In the present paper, we propose the **DoSS** algorithm in which the above two drawbacks are addressed. Inspired by [10], we also develop an on-line adaptive framework by integrating the estimation of unknown environment and task planning for multi-robot simultaneously. Nevertheless, in contrast to the **AdaSearch** algorithm, we here let the robots cooperatively determine their paths by themselves, and introduce the novel concept of D-UCB which greatly helps reduce the computational complexity in solving multi-robot task planning problems. These two points also make our **DoSS** algorithm implementable in both distributed and real-time manner. In addition, other differences between this paper and [10] are also noteworthy: 1) while the measurement noise is assumed to follow a Poisson process in [10], we consider the Gaussian distributed noise; see Sec. II-B; and 2) the **AdaSearch** scheme utilizes both lower and upper confidence bounds to guide the robot decision, in contrast, we only need to compute the upper bound with **DoSS**. The mechanism of the **DoSS** algorithm is illustrated in Fig. 1.

<sup>†</sup>Bin Du is Ph.D. student, and Dengfeng Sun is Associate Professor at Purdue University, West Lafayette, IN 47907, [{du185, dsun}@purdue.edu](mailto:{du185, dsun}@purdue.edu)

<sup>‡</sup>Kun Qian and Hassan Iqbal are Ph.D. students, and Christian Claudel is Assistant Professor, at The University of Texas at Austin, Austin, TX 78712, [{kunqian, hassan.iqbal, christian.claudel}@utexas.edu](mailto:{kunqian, hassan.iqbal, christian.claudel}@utexas.edu)

The authors would like to acknowledge NSF (National Science Foundation) CPS No. 1739964, NSF CIS No. 1636154 and TxDOT project 0-7007 for research funding.

Bin Du and Kun Qian contributed equally to this manuscript.

It is worth mentioning that the idea of UCB has also been commonly adopted in solving the problems of environment monitoring [11]–[13] and sensor coverage [14]–[16]. In these problems, the environment is often modeled as a Gaussian process [17]. However, as suggested in [10], the Gaussian process may not be able to reflect some specific scenarios of the source seeking problem. On this basis, we apply a state-space model for the dynamical environment; see details in Sec. III-C. This also makes our work significantly different with other literature relying on Gaussian processes.

We summarize the contributions of this paper as follows:

- 1) A novel DoSS algorithm is proposed for solving the source seeking problem with the multi-robot system, and its performance is theoretically guaranteed by evaluating the asymptotic no-regret.
- 2) A new notion of D-UCB is introduced which enables the DoSS algorithm to be implementable in both distributed and real-time manner.
- 3) The DoSS algorithm is further adapted with the scenario of a dynamical environment. To the best of our knowledge, this is the first work that considers the source seeking problem in the setting of both multi-robot system and dynamical environment.

## II. DISTRIBUTED SOURCE SEEKING

In this section, we formalize the problem of distributed source seeking with the multi-robot system. For the sake of presentation, let us first concentrate on a basic version of the problem, i.e., assuming that the state of the environment is static and readily known. Later on, this scenario will be extended to a situation in which the state is unknown and thus has to be estimated by the robots noisy measurements.

### A. Problem Statement: A Basic Version

Let us consider a bounded and obstacle-free environment, in which sources of interest are present. In particular, we specify the considered environment by a set of points  $\mathcal{S}$  with each element  $\mathbf{s} \in \mathcal{S}$  representing the position of the point. Since the environment has been assumed to be bounded, it is easy to see that the set  $\mathcal{S}$  is finite. We denote  $N$  the number of points in the set, i.e.,  $N = |\mathcal{S}|$ . For each point  $\mathbf{s}$  in  $\mathcal{S}$ , there exists a real-valued function  $\phi_0(\cdot) : \mathcal{S} \rightarrow \mathbb{R}_+$  that maps the point's positional information  $\mathbf{s}$  to a positive quantity  $\phi_0(\mathbf{s})$  indicating the level of emission of the source. Naturally, in order to locate the sources, our objective is to deploy the multiple robots to the points with the highest quantities  $\phi_0(\mathbf{s})$ . More specifically, let us employ a team of  $I$  robots which are capable of moving among  $\mathcal{S}$  and communicating with other connected robots, and expect them to locate as many sources as possible. Furthermore, we denote  $\mathbf{p}[i] \in \mathcal{S}$  the position of the robot  $i \in \mathcal{I} := \{1, 2, \dots, I\}$ , and assume that each robot has perfect knowledge of the state  $\phi_0(\mathbf{s})$  for each point  $\mathbf{s}$ . On this account, it is clear to see that the sources can be easily located by deploying the robots to the points which have the highest  $\phi_0(\mathbf{p}[i])$ .

Nevertheless, redundancies might be present when assigning the multi-robot system to the multiple sources. To

elaborate on this, let us suppose that there is no cooperation between any pair of two robots, then each robot  $i$  in the team will tend to locate the same target  $\mathbf{s}^*$  which has the maximum  $\phi_0(\mathbf{s}^*)$ , i.e.,  $\mathbf{p}[i] = \mathbf{s}^*, \forall i \in \mathcal{I}$ . This is obviously not appealing when the multi-robot system is employed. In contrast, we shall expect that the team of  $I$  robots can locate as many sources as possible, e.g., the (locally) highest  $I$  quantities  $\phi_0(\mathbf{s})$ , by leveraging the communications between the connected robots. In order to achieve such a goal, we formalize the problem of distributed source seeking with the multi-robot system as the following optimization,

$$\underset{\mathbf{p}[i] \in \mathcal{S}, i \in \mathcal{I}}{\text{maximize}} \quad F(\mathbf{p}[1], \mathbf{p}[2], \dots, \mathbf{p}[I]) = \sum_{\mathbf{s} \in \cup_{i=1}^I \mathbf{p}[i]} \phi_0(\mathbf{s}). \quad (1)$$

The objective function  $F(\cdot) : \mathcal{S}^I \rightarrow \mathbb{R}_+$  maps the positions of  $I$  robots to a positive scalar that sums all *distinct* measured quantities. Throughout this paper, we assume that the maximizer of problem (1) is unique and express it as a compact form  $\mathbf{p}^* = [\mathbf{p}^*[1], \mathbf{p}^*[2], \dots, \mathbf{p}^*[I]] \in \mathcal{S}^I$ .

It should be noted that, since the set  $\mathcal{S}$  is finite, the above maximization problem can be naively solved by assigning the  $i$ -th robot to the point  $\mathbf{p}[i]$  which has the  $i$ -th largest quantity  $\phi_0(\mathbf{p}[i])$ . However, such a naive scheme inherently assumes each robot to be aware of its exclusive global ID which is a restrictive requirement in a fully distributed architecture [18]. As an alternative way to solve the optimization problem (1), we shall remark that the problem can be viewed as a special case of the monotone submodular maximization, and thus can be solved by the distributed algorithm proposed in our previous work [19]. The key idea of this algorithm is to find the equilibrium solution, and interestingly, it can be verified that the problem (1) has a unique equilibrium which is coincident with the optimal solution. We refer the interested reader to our work [19] for details on the distributed algorithm.

### B. Source Seeking via Estimation on the Environment

Notice that the problem considered in the previous subsection is somewhat trivial, since we assumed that each robot perfectly knows the state  $\phi_0(\mathbf{s})$  of the entire environment. This is unrealistic for the real-world applications. We next let the team of robots cooperatively estimate the environment based on the local noisy measurements, and in the following, we first introduce the measurement model.

Suppose that the vector  $\phi_0 \in \mathbb{R}_+^N$  stacks each individual state  $\phi_0(\mathbf{s})$  for all points  $\mathbf{s}$  in the environment  $\mathcal{S}$ . We consider the following stochastic measurement model for each robot  $i$ ,

$$\mathbf{z}^i = H^i(\mathbf{p}[i])\phi_0 + \mathbf{n}^i, \quad (2)$$

where the vector  $\mathbf{z}^i \in \mathbb{R}^{m_i}$  represents the obtained measurement of dimension  $m_i$ ;  $H^i(\mathbf{p}[i]) \in \mathbb{R}^{m_i \times N}$  denotes the measurement matrix depending on the robot's position  $\mathbf{p}[i]$ ; and  $\mathbf{n}^i \in \mathbb{R}^{m_i}$  is corresponding to the measurement noise. In addition, we assume that the noise  $\mathbf{n}^i$  follows the independent and identically distributed Gaussian distribution with zero-mean and covariance  $V^i = v^i \cdot \mathbf{I} \in \mathbb{R}^{m_i \times m_i}$  where  $\mathbf{I}$

denotes the identity matrix with appropriate dimensions. Particularly, we denote  $\bar{v} = \max_{i \in \mathcal{I}} v^i$  and  $\underline{v} = \min_{i \in \mathcal{I}} v^i$ .

*Remark 1:* We note that the measurement matrix  $H^i(\mathbf{p}[i])$  is not specified in the above model (2). In fact, it can be defined by various means based on the robot's position. One of the simplest way is to let  $H^i(\mathbf{p}[i]) = \mathbf{e}_l^\top$  where  $\mathbf{e}_l \in \mathbb{R}^N$  is an unit vector, i.e., the  $l$ -th column of the identity matrix, and  $l \in \{1, 2, \dots, N\}$  denotes the index of the position  $\mathbf{p}[i]$  in the environment  $\mathcal{S}$ . This means that the robot only measures the quantity at the point where it currently is. Such a choice of  $H^i(\mathbf{p}[i])$  is actually adopted in [10] as the so-called point-wise sensing model. Besides, some other specifications of the measurement matrix are also used in the existing works. For instance, a circular sensing area with radius  $r_i$  is applied in [20], which implies that,

$$H^i(\mathbf{p}[i]) = [\mathbf{e}_l]_{l \in \mathcal{C}^i}^\top, \quad (3)$$

where the set  $\mathcal{C}^i := \{l \mid \|\mathbf{s}_l - \mathbf{p}[i]\| \leq r_i\}$  includes the indices of all points  $\mathbf{s}_l$  that fall into the disk which is centered at  $\mathbf{p}[i]$  and has radius  $r_i$ .

Based on the measurement model (2), one should notice that the true value of  $\phi_0$  can be estimated by many techniques, such as least-squares, classical Kalman filter, to name a few, when some mild conditions on the measurement matrices are satisfied. Therefore, the problem of distributed source seeking with an unknown environment can be addressed by a simple approach which contains the following two phases separately: i) let the team of robots move around the environment and obtain an accurate enough estimation of the state; and ii) specify the robots' target positions by solving the maximization problem (1) based on the estimated states. However, this is essentially an off-line approach, since the robots do not have specific targets when estimating the environment in the phase i) and the phase ii) cannot be started until an accurate enough estimate is obtained. Motivated by this, in the next section, we aim to integrate the above two phases together and propose an adaptive on-line framework – the **DoSS** algorithm. That is, the robots recursively update their target positions; meanwhile, measure and estimate the unknown environment, until the steady state is reached in which the team of  $I$  robots manages to identify the top  $I$  sources.

### III. AN ADAPTIVE ON-LINE FRAMEWORK

Before proceeding to the development of the **DoSS** algorithm, let us first introduce some additional notations. Considering that our on-line approach is developed in a recursive manner, we denote  $\mathbf{p}_k[i]$  as the  $i$ -th robot current position at iteration  $k \in \mathbb{N}_+$ . Accordingly, the measurement model (2) is adapted into

$$\mathbf{z}_k^i = H^i(\mathbf{p}_k[i])\phi_0 + \mathbf{n}_k^i. \quad (4)$$

Note that here  $\mathbf{z}_k^i \in \mathbb{R}^{m_i}$  and  $\mathbf{n}_k^i \in \mathbb{R}^{m_i}$  follow the same definitions as  $\mathbf{z}^i$  and  $\mathbf{n}^i$  in model (2). In addition, we further assume that the measurement noise  $\mathbf{n}_k^i$  is independent and identically (Gaussian) distributed for each individual robot  $i$ , with zero mean and covariance matrix  $V^i$ . With

the help of the above adapted measurement model (4), we can now specify the technique for estimating the unknown environment.

#### A. Kalman Consensus Filter

Let us begin by rewriting the measurement model (4) into the following compact form:

$$\mathbf{z}_k = H_k \phi_0 + \mathbf{n}_k. \quad (5)$$

Here,  $\mathbf{z}_k = [(\mathbf{z}_k^1)^\top, (\mathbf{z}_k^2)^\top, \dots, (\mathbf{z}_k^I)^\top]^\top \in \mathbb{R}^M$  denotes the measurements obtained by all robots with  $M = \sum_{i=1}^I m_i$ ;  $H_k = [H^1(\mathbf{p}_k^1)^\top, H^2(\mathbf{p}_k^2)^\top, \dots, H^I(\mathbf{p}_k^I)^\top]^\top \in \mathbb{R}^{M \times N}$  stacks all local measurement matrices as a collective global one<sup>1</sup>; and  $\mathbf{n}_k = [(\mathbf{n}_k^1)^\top, (\mathbf{n}_k^2)^\top, \dots, (\mathbf{n}_k^I)^\top]^\top \in \mathbb{R}^M$  is the zero-mean Gaussian noise whose covariance is expressed as

$$V := \text{Diag}\{V^1, V^2, \dots, V^I\} \in \mathbb{R}^{M \times M}. \quad (6)$$

Subsequently, the centralized Kalman filter for estimating the mean  $\hat{\phi}_k \in \mathbb{R}^N$  and covariance  $\Sigma_k \in \mathbb{R}^{N \times N}$  performs the following recursions,

$$\Sigma_{k+1} = \left( \Sigma_k^{-1} + Y_k \right)^{-1}; \quad (7a)$$

$$\hat{\phi}_{k+1} = \hat{\phi}_k + \Sigma_{k+1}(\mathbf{y}_k - Y_k \hat{\phi}_k), \quad (7b)$$

where the two variables  $Y_k := H_k^\top V^{-1} H_k \in \mathbb{R}^{N \times N}$  and  $\mathbf{y}_k := H_k^\top V^{-1} \mathbf{z}_k \in \mathbb{R}^N$ , often referred to as the new information, incorporate the measurements into the updates. One should notice that the above recursion (7) is slightly different with the standard Kalman filter since it only serves as the correction step and the prediction step is absent. This is due to the fact that the environment is assumed to be static. Later on, the predict step of the Kalman filter will show up, when we deal with the dynamical environment in Sec. III-C.

It is also worth mentioning that the Kalman filter (7) readily estimates the unknown environment in an on-line manner, i.e., the team of  $I$  robots moves to new positions, obtains the new measurements, and updates their estimations. However, we should note that two issues may arise: i) the statistical property of the classical Kalman filter may no longer hold due to the sequential decision process; ii) such an on-line procedure is performed in a centralized way, since the new information  $Y_k$  and  $\mathbf{y}_k$  are involved with the data obtained/maintained by all robots. In order to devise a distributed scheme to run the Kalman filter (7), many existing works, e.g., [21]–[23], leverage the special structure of the noise covariance  $V$ . Considering the diagonal structure of the matrix  $V$ , as shown in (6), the new information can be further expressed as

$$Y_k = \sum_{i=1}^I H^i(\mathbf{p}_k^i)(V^i)^{-1} H^i(\mathbf{p}_k^i)^\top; \quad (8a)$$

$$\mathbf{y}_k = \sum_{i=1}^I H^i(\mathbf{p}_k^i)(V^i)^{-1} \mathbf{z}_k^i, \quad (8b)$$

<sup>1</sup>When writing  $H_k$ , with slight abuse of notation, we have absorbed the dependency on the robots' positions  $\mathbf{p}_k^i$ 's into the index  $k$ .

which means that  $Y_k$  and  $\mathbf{y}_k$  can be computed by simply summing all the local information together. This motivates the development of Kalman consensus filter, in which each robot first carries out an average/sum consensus procedure to fuse local information and then performs (7).

### B. The DoSS Algorithm

In the previous subsection, we focused on the estimation of the unknown environment. Our question now becomes: how to integrate the estimation together with the robots decision-making process. A naive idea here would be using the estimated state  $\hat{\phi}_k$  at each iteration  $k$ , and then solving

$$\mathbf{p}_k \in \arg \max_{\mathbf{p}[i] \in \mathcal{S}, i \in \mathcal{I}} \sum_{\mathbf{s} \in \cup_{i=1}^I \mathbf{p}[i]} \hat{\phi}_k(\mathbf{s}). \quad (9)$$

Here, we use  $\hat{\phi}_k(\mathbf{s}) \in \mathbb{R}$  to denote one component of the vector  $\hat{\phi}_k$  which corresponds to the point  $\mathbf{s}$  in the environment. It should be emphasized that such a scheme cannot guarantee the team of robots to locate the sources with the highest true  $\phi_0(\mathbf{s})$ 's. An undesired but possible scenario is that the robots significantly underestimate the maximum value  $\phi_0(\mathbf{s}^*)$  at the initial stage, i.e.,  $\hat{\phi}(\mathbf{s}^*) \ll \phi_0(\mathbf{s}^*)$ , and as a result, the robots will never have another chance to visit the key point  $\mathbf{s}^*$ . On this account, it can be seen that merely utilizing the estimated mean is insufficient to drive the team of robots to the desired positions. To address this, we next take advantage of both the estimated mean  $\hat{\phi}_k$  and covariance  $\Sigma_k$  to develop our DoSS algorithm.

Based on  $\hat{\phi}_k$  and  $\Sigma_k$ , let us introduce an additional variable  $\mu_k \in \mathbb{R}^N$ , which we refer to as D-UCB,

$$\mu_k := \hat{\phi}_k + \beta_k(\delta) \cdot \text{diag}^{1/2}(\Sigma_k). \quad (10)$$

Note that the operator  $\text{diag}^{1/2}(\cdot) : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^N$  maps the square root of the matrix diagonal elements to a vector, and the parameter  $\beta_k(\delta) > 0$  depending on the critical confidence level  $\delta$  will be specified later on. In fact, the intuition behind this notion of D-UCB is straightforward: each  $\mu_k$  provides a probabilistic upper bound of the true value  $\phi_0$  by utilizing the current mean and covariance. Next, we formalize, with the following lemma, how the true value  $\phi_0$  is upper bounded by the D-UCB  $\mu_k$  with the probability related to  $\delta$ . Note that Lemma 1 is a special case of the subsequent Lemma 2 and we refer the proof to the complete version of this paper [24].

**Lemma 1:** Suppose that the state estimates  $\hat{\phi}_k$  and  $\Sigma_k$  are generated by the Kalman (consensus) filter (7)–(8) with the initialization  $\hat{\phi}_0$  and  $\sigma \cdot \mathbf{I} \leq \Sigma_0 \leq \bar{\sigma} \cdot \mathbf{I}$ , then it holds that,

$$\mathbb{P}(|\hat{\phi}_k - \phi_0| \preceq \beta_k(\delta) \cdot \text{diag}^{1/2}(\Sigma_k)) \geq 1 - \delta, \quad (11)$$

where  $|\cdot|$  and  $\preceq$  are defined element-wise, the probability  $\mathbb{P}(\cdot)$  is taken on random noises  $(\mathbf{n}_0, \mathbf{n}_1, \dots, \mathbf{n}_k)$ , and the sequence  $\{\beta_k(\delta)\}_{k \in \mathbb{N}_+}$  is non-decreasing satisfying

$$\beta_k(\delta) \geq N^{3/2} C_1 + N^2 C_2 \cdot \sqrt{\log\left(\frac{\bar{\sigma}/\sigma + \bar{\sigma} \cdot k/\bar{v}^2}{\delta^{2/N}}\right)}, \quad (12)$$

with  $C_1 = \|\hat{\phi}_0 - \phi_0\|/\sqrt{\sigma}$  and  $C_2 = \bar{v}^2 \sqrt{\max\{2, 2/\bar{v}\}}$ .

The above Lemma 1 inherently constructs a polytope centered at  $\hat{\phi}_k$  such that the true value  $\phi_0$  falls into it with probability at least  $1 - \delta$ . Based on the polytope defined by the inequality in (11), it can be seen that the D-UCB  $\mu_k$  takes the upper bounds marginally and each element  $\mu_k(\mathbf{s})$  is guaranteed to have  $\mu_k(\mathbf{s}) \geq \phi_0(\mathbf{s})$  with probability at least  $1 - \delta$ . Next, we use the defined D-UCB  $\mu_k$  to update the robots target positions online, by solving the following maximization problem:

$$\mathbf{p}_k \in \arg \max_{\mathbf{p}[i] \in \mathcal{S}, i \in \mathcal{I}} \sum_{\mathbf{s} \in \cup_{i=1}^I \mathbf{p}[i]} \mu_k(\mathbf{s}). \quad (13)$$

We now summarize the DoSS algorithm in the following Algorithm 1 and establish its performance with the following theorem. Likewise, we refer the proof of Theorem 1, as a special case of Theorem 2, to our complete version [24].

---

#### Algorithm 1: Distributed on-line Source Seeking

---

**Initialization:** Each agent  $i$  initializes its own estimates  $\hat{\phi}_0$  and  $\Sigma_0$ , and computes the target position  $\mathbf{p}_1^i$ . Set the confidence level  $\delta$  and also  $\{\beta_k(\delta)\}_{k \in \mathbb{N}_+}$ . Let  $k = 1$ .

**while** the stopping criteria is NOT satisfied **do**

Each sensor  $i$  **simultaneously** performs

**Step 1 (Measuring):** Obtain the measurement  $\mathbf{z}_k^i$  based on the measurement matrix  $H^i(\mathbf{p}_k^i)$ ;

**Step 2 (Kalman Filtering):** Collect information from neighbors, obtain mean  $\hat{\phi}_k$  and covariance  $\Sigma_k$  by Kalman consensus filter (7);

**Step 3 (D-UCB Computing):** Compute via (10) the updated D-UCB  $\mu_k$  based on  $\hat{\phi}_k$  and  $\Sigma_k$ ;

**Step 4 (Target Positions Updating):** Assign the new target position  $\mathbf{p}_{k+1}^i$  by solving (13).

Let  $k \leftarrow k + 1$ , and continue.

**end**

---

**Theorem 1:** Suppose that  $\{\mathbf{p}^k\}_{k \in \mathbb{N}_+}$  is the sequence generated by Algorithm 1 under the conditions in Lemma 1, then it holds that, with probability  $1 - \delta$ , for  $\forall K \in \mathbb{N}_+$ ,

$$\sum_{k=1}^K \left( F(\mathbf{p}^*) - F(\mathbf{p}_k) \right) \leq \mathcal{O}\left(\sqrt{K} \log(K)\right). \quad (14)$$

**Remark 2:** A significant difference between the Linear UCB algorithm [25] and our DoSS algorithm is that we construct the D-UCB, rather than the standard UCB, to drive the update of  $\mathbf{p}_k$ 's. Due to this difference, one cannot immediately prove the above Theorem 1 by following exactly the steps in [25]. A remarkable idea of our proof is to define a specific vector norm which interplays with the form of D-UCB and then establish the regret analysis with respect to the specific norm. This makes our theoretical results non-trivial. In addition, we should also emphasize that the introduction of D-UCB helps reducing the computational complexity of

our D<sub>OSS</sub> algorithm significantly, when solving the problem in the multi-robot setting. Since the standard UCB is defined in a joint sense, when solving the multi-robot maximization problem (13) with the standard UCB, it is inherently a combinatorial optimization and can be extremely complicated to find the exact solution. In contrast, due to the fact that the D-UCB takes the upper bounds marginally as mentioned before, the maximization (13) can be essentially decomposed and becomes much easier to solve for exact solutions. We remark this as one of the most important contributions of the proposed D<sub>OSS</sub> algorithm.

### C. D<sub>OSS</sub> on a Dynamic Environment

Now, we extend our problem setup into a more general and applicable case, i.e., when the unknown environment follows some dynamics so that the multiple robots need to track the moving sources. More specifically, let us consider that the state of dynamical environment  $\phi_t \in \mathbb{R}_+^N$ , and is governed by the following linear time-varying model:

$$\phi_{t+1} = A_{t+1}\phi_t, \quad (15)$$

where the subscript  $t$  represents the discrete time-step and  $A_t \in \mathbb{R}^{N \times N}$  denotes the state transition matrix. Now, since the unknown environment state is changing with time, the objective of the multiple robots becomes to track the positions  $\mathbf{p}_t^* \in \mathcal{S}^I$  of the moving sources, which is defined by the following maximization,

$$\mathbf{p}_t^* = \arg \max_{\mathbf{p}[i] \in \mathcal{S}, i \in \mathcal{I}} F_t(\mathbf{p}) = \sum_{\mathbf{s} \in \cup_{i=1}^I \mathbf{p}[i]} \phi_t(\mathbf{s}). \quad (16)$$

In (16), we use  $\phi_t(\mathbf{s}) \in \mathbb{R}_+$  to represent the component of the vector  $\phi_t$  which corresponds to position  $\mathbf{s}$ , and thus the objective function  $F_t(\cdot) : \mathcal{S}^I \rightarrow \mathbb{R}_+$  should also depend on the time-step  $t$ . Furthermore, in order to ensure that the above maximization (16) is well-defined, we assume that  $\phi_t$  is always bounded and also will not vanish to zero as the time-step  $t$  increases. It also means that the sources, characterized by the maximum components of  $\phi_t$ , are always recognizable for the multiple robots.

In order to achieve the goal of tracking moving sources  $\mathbf{p}_t^*$ , we here adopt the same framework as our D<sub>OSS</sub> algorithm. However, considering that the environment is subject to the linear dynamics (15), we will need to incorporate the prediction step into the Kalman consensus filter, as mentioned in Sec. III-A. Consequently, the updated Kalman consensus filter reads<sup>2</sup>,

$$\Sigma_{k+1} = A_{k+1}(\Sigma_k^{-1} + Y_k)^{-1}A_{k+1}^\top; \quad (17a)$$

$$\hat{\phi}_{k+1} = A_{k+1} \left( \hat{\phi}_k + (\Sigma_k^{-1} + Y_k)^{-1}(\mathbf{y}_k - Y_k \hat{\phi}_k) \right), \quad (17b)$$

where the new information  $Y_k$  and  $\mathbf{y}_k$  follow the same definitions as in Sec. III-A, however, the robot's measurement model needs to adapt with the environment dynamics, i.e.,

$$\mathbf{z}_k^i = H^i(\mathbf{p}_k^i)\phi_k + \mathbf{n}_k^i. \quad (18)$$

<sup>2</sup>It is assumed that the discrete time-step  $t$  of the dynamics is consistent with the iteration index  $k$  of our algorithm, and henceforth we use the index  $k$  to replace  $t$  in the system model.

Now, the D<sub>OSS</sub> algorithm for the dynamical environment runs the same procedure as in Algorithm 1. The following Lemma 2 and Theorem 2 together establish the convergence result for the algorithm; see the proof in [24].

**Lemma 2:** Suppose that the state estimates  $\hat{\phi}_k$  and  $\Sigma_k$  are generated by the Kalman (consensus) filter (17) with the initialization  $\hat{\phi}_0$  and  $\sigma \cdot \mathbf{I} \leq \Sigma_0 \leq \bar{\sigma} \cdot \mathbf{I}$ , then it holds that,

$$\mathbb{P}\left(|\hat{\phi}_k - \phi_k| \leq \beta_k(\delta) \cdot \text{diag}^{1/2}(\Sigma_k)\right) \geq 1 - \delta, \quad (19)$$

where  $\{\beta_k(\delta)\}_{k \in \mathbb{N}_+}$  is non-decreasing satisfying

$$\beta_k(\delta) \geq N^{3/2}C_1 + N^2C_2 \cdot \sqrt{\log\left(\frac{\bar{\sigma}/\sigma + \bar{\sigma} \cdot k/\underline{\sigma}^2}{\delta^{2/N}}\right)}, \quad (20)$$

with  $C_1$  and  $C_2$  be defined as in Lemma 1.

**Theorem 2:** Suppose that  $\{\mathbf{p}^k\}_{k \in \mathbb{N}_+}$  is the sequence generated by Algorithm 1 with the Kalman filtering step replaced by (17), and let the conditions in Lemma 2 holds, then one can have that, with probability  $1 - \delta$ , for  $\forall K \in \mathbb{N}_+$ ,

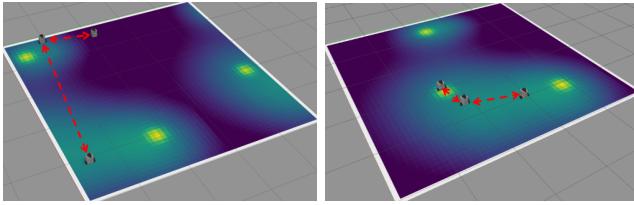
$$\sum_{k=1}^K \left( F_k(\mathbf{p}_k^*) - F_k(\mathbf{p}_k) \right) \leq \mathcal{O}\left(\sqrt{K} \log(K)\right). \quad (21)$$

**Remark 3:** It is also worth noting that Theorem 2 guarantees that the multi-robot system is capable of tracking the desired sources accurately, even if they are moving around the unknown environment. This can be achieved primarily due to the fact that the environment is assumed to be noise-free. The process noise in the model of environment dynamics will be considered in future work.

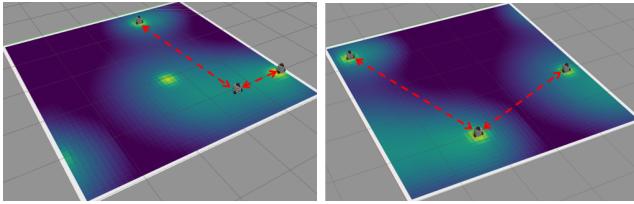
## IV. SIMULATION

In this section, we demonstrate the effectiveness of our D<sub>OSS</sub> algorithm, by considering a real-world methane leaking source seeking problem. In fact, such a problem has been broadly studied in the area of robotics; see e.g., [26], [27]. Compared to these existing works, a primary difference here is that we deploy multiple robots, rather than a single one, to the target methane field. As a result, we expect that the individual robots will be able to track the distinct and possibly moving leaking sources by leveraging the cooperations among the entire team of robots.

Let us suppose that the target methane field is described by a  $D \times D$  lattice, as shown in the background of Fig. 2. Each cell  $l \in \{1, 2, \dots, D^2\}$  in the lattice is represented by its position  $\mathbf{s}^l$  and also the quantity  $\phi_t(\mathbf{s}^l)$  which indicates the level of methane concentration at the time-step  $t$ . Overall, the  $N$ -dimensional vector  $\phi_t = [\phi_t(\mathbf{s}^1), \phi_t(\mathbf{s}^2), \dots, \phi_t(\mathbf{s}^N)]^\top$  with  $N = D^2$  characterizes the state of the entire methane field of interest. More specifically, in this simulation, we set the size of the methane field as  $D = 50$ . The initialized methane state  $\phi_0$  is generated through Gaussian kernels with leaking sources having largest concentrations among the field, and then we let leaking sources move within the field so that the time-varying  $\phi_t$  is generated. In order to explore the unknown target methane field and furthermore track the moving leaking sources, we employ a team of three robots, each of them equipped with a sensor that is



(a) Snapshot at iteration  $k = 100$  (b) Snapshot at iteration  $k = 250$



(c) Snapshot at iteration  $k = 450$  (d) Snapshot at iteration  $k = 600$

Fig. 2: Demonstration of three robots' tracking of the moving leaking sources in an unknown methane field.

capable of measuring a circular area with radius  $r = 3$ ; see the detailed measurement model (18) and the description of measurement matrix (3) in Remark 1. In particular, we assume that the sensing noise of each robot is independent and identically distributed Gaussians with zero-mean and covariance  $V^i = \mathbf{I}$ , where  $\mathbf{I}$  denotes the identity matrix with appropriate dimension. Note that, since the maximum value of the state  $\phi_t$  is set around 5, the noise covariance is reasonably large so that the overall problem is not trivial to solve. Besides, it is also assumed that the three robots can exchange information with their immediate neighbors, and the communication channels, shown as the red dot lines in Fig. 2, follow a simple undirected connected graph.

To demonstrate the result of tracking of the moving leaking sources, four snapshots are taken and shown in Fig. 2 at the iterations  $k = 100, 250, 450, 600$ , respectively. It can be observed that the team of robots is able to locate all three moving leaking sources at the 600-th iteration. In addition, in order to show the simulation result quantitatively, Fig. 3 plots both the regret  $r_k = F_k(\mathbf{p}^*(k)) - F_k(\mathbf{p}_k)$  at each iteration as a blue line, and the cumulative regret  $\sum_{t=1}^k r_t$  as a black line. Note that each line is obtained by the data averaged from 20 Monte-Carlo trials; the standard deviation is also reported in the figure. It can be concluded from Fig. 3 that the regret  $r_k$  decreases to zero as the number of iterations grows, which confirms that the team of robots will be able to track the moving leaking sources. Besides, the cumulative regret shows a sub-linear increase, which is also consistent with the theoretical result of Theorem 2.

In order to validate the effectiveness of the proposed algorithm, we further compare the performance of our DoSS algorithm with two benchmark schemes: 1) the AdaSearch algorithm; and 2) a naive approach, termed as NaiveSearch, in which the robots scan the whole unknown field repeatedly and determine the position of leaking sources by the current estimation of the field. Note that, to evaluate these three scheme fairly, we here only

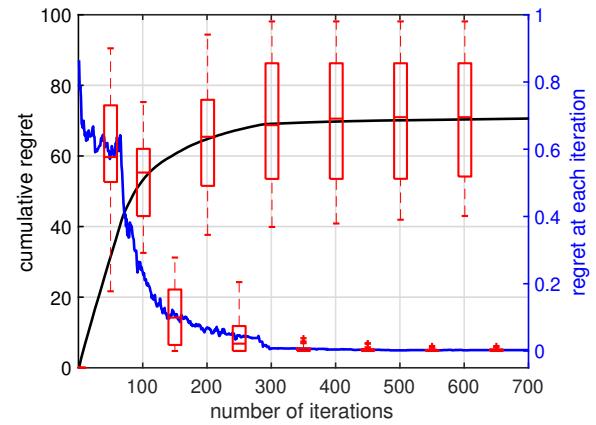


Fig. 3: Regret analysis

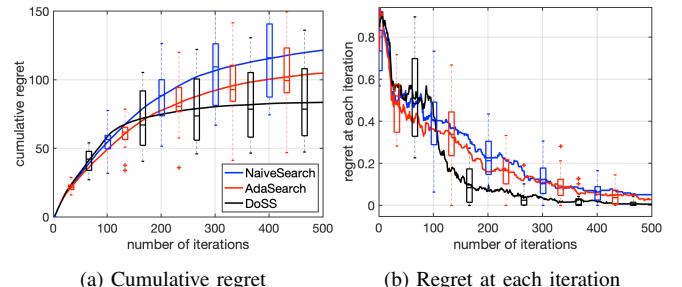


Fig. 4: Comparison of the regret with three different schemes

consider a static target methane field. As previously, we run each of the three schemes for 20 Monte-Carlo trials, and Fig. 4 shows the simulation results. It can be observed from this figure that our DoSS algorithm outperforms both AdaSearch and NaiveSearch algorithms in terms of the regret descent rate, which means that the our algorithm can locate the leaking sources more efficiently in an unknown methane field than the two others.

## V. CONCLUSION

In this paper, we proposed a novel algorithmic framework, termed as DoSS, for solving the multi-robot source seeking problem in a dynamical unknown environment. Building on the notion of D-UCB, our algorithm integrates the estimation of the unknown environment and task planning for multiple robots. Both theoretical analysis and numerical simulations show that the DoSS algorithm can drive the team of robots to a steady state in which multiple sources of interest are located. They also show that DoSS outperforms other benchmark algorithms. Future work will focus on a more general problem setup in which process noise is present in the environment dynamics.

## REFERENCES

- [1] Zhijun Tang and Umit Ozguner. Motion planning for multitarget surveillance with mobile sensor agents. *IEEE Transactions on Robotics*, 21(5):898–908, 2005.
- [2] Alireza Ghaffarkhah and Yasamin Mostofi. Path planning for networked robotic surveillance. *IEEE Transactions on Signal Processing*, 60(7):3560–3575, 2012.

- [3] Rongxing Lu, Xiaodong Lin, and Xuemin Shen. SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency. *IEEE transactions on parallel and distributed systems*, 24(3):614–624, 2012.
- [4] Qiang Lu, Qing-Long Han, Bota Zhang, Dongliang Liu, and Shirong Liu. Cooperative control of mobile sensor networks for environmental monitoring: an event-triggered finite-time control scheme. *IEEE transactions on cybernetics*, 47(12):4134–4147, 2016.
- [5] Kun Qian and Christian G. Claudel. Real-time mobile sensor management framework for city-scale environmental monitoring. *Journal of Computational Science*, 45, 2020.
- [6] Frank Mascarich, Taylor Wilson, Christos Papachristos, and Kostas Alexis. Radiation source localization in gps-denied environments using aerial robots. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6537–6544. IEEE, 2018.
- [7] Shuai Li and Yi Guo. Distributed source seeking by cooperative robots: All-to-all and limited communications. In *2012 IEEE International Conference on Robotics and Automation*, pages 1107–1112. IEEE, 2012.
- [8] Senqiang Zhu, Danwei Wang, and Chang Boon Low. Cooperative control of multiple uavs for moving source seeking. *Journal of Intelligent & Robotic Systems*, 74(1-2):333–346, 2014.
- [9] Nikolay A Atanasov, Jerome Le Ny, and George J Pappas. Distributed algorithms for stochastic source seeking with mobile robot networks. *Journal of Dynamic Systems, Measurement, and Control*, 137(3), 2015.
- [10] Esther Rolf, David Fridovich-Keil, Max Simchowitz, Benjamin Recht, and Claire Tomlin. A successive-elimination approach to adaptive robotic source seeking. *IEEE Transactions on Robotics*, 2020.
- [11] Roman Marchant and Fabio Ramos. Bayesian optimisation for intelligent environmental monitoring. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 2242–2249. IEEE, 2012.
- [12] Shi Bai, Jinkun Wang, Fanfei Chen, and Brendan Englot. Information-theoretic exploration with bayesian optimization. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1816–1822. IEEE, 2016.
- [13] Lauren M Miller, Yonatan Silverman, Malcolm A MacIver, and Todd D Murphey. Ergodic exploration of distributed information. *IEEE Transactions on Robotics*, 32(1):36–52, 2015.
- [14] Wenhao Luo and Katia Sycara. Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6359–6364. IEEE, 2018.
- [15] Wenhao Luo, Changjoo Nam, George Kantor, and Katia Sycara. Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1488–1496, 2019.
- [16] Alessia Benevento, María Santos, Giuseppe Notarstefano, Kamran Paynabar, Matthieu Bloch, and Magnus Egerstedt. Multi-robot coordination for estimation and coverage of unknown spatial fields. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7740–7746. IEEE, 2020.
- [17] Anthony O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society: Series B (Methodological)*, 40(1):1–24, 1978.
- [18] ElMoustapha Ould-Ahmed-Vall, Douglas M Blough, Bonnie Heck Ferri, and George F Riley. Distributed global id assignment for wireless sensor networks. *Ad Hoc Networks*, 7(6):1194–1216, 2009.
- [19] Bin Du, Kun Qian, Christian Claudel, and Dengfeng Sun. Jacobian-style iteration for distributed submodular maximization. *arXiv preprint arXiv:2010.14082*, 2020.
- [20] Jalal Habibi, Hamid Mahboubi, and Amir G Aghdam. A gradient-based coverage optimization strategy for mobile sensor networks. *IEEE Transactions on Control of Network Systems*, 4(3):477–488, 2016.
- [21] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005.
- [22] R. Olfati-Saber and J. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6698–6703. IEEE, 2005.
- [23] R. Olfati-Saber. Distributed kalman filtering for sensor networks. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, 2007.
- [24] Bin Du, Kun Qian, Christian Claudel, and Dengfeng Sun. Multi-robot dynamical source seeking in unknown environments. *arXiv preprint arXiv:2103.11016*, 2021.
- [25] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.
- [26] Nikolay Atanasov, Roberto Tron, Victor M Preciado, and George J Pappas. Joint estimation and localization in sensor networks. In *53rd IEEE Conference on Decision and Control*, pages 6875–6882. IEEE, 2014.
- [27] Victor Manuel Hernandez Bennetts, Achim J Lilienthal, Ali Abdul Khaliq, Victor Pomareda Sese, and Marco Trincavelli. Towards real-world gas distribution mapping and leak localization using a mobile robot with 3d and remote gas sensing capabilities. In *2013 IEEE International Conference on Robotics and Automation*, pages 2335–2340. IEEE, 2013.