Source Seeking by Dynamic Source Location Estimation

Tianpeng Zhang, Victor Qin, Yujie Tang, Na Li

***Abstract*— This paper focuses on the problem of multi-robot source-seeking, where a group of mobile sensors localizes and moves close to a single source using only local measurements. Drawing inspiration from the optimal sensor placement research, we develop an algorithm that estimates the source location while approaches the source following gradient descent steps on a loss function defined on the Fisher information. We show that exploiting Fisher information gives a higher chance of obtaining an accurate source location estimate and naturally leads the sensors to the source. Our numerical experiments demonstrate the advantages of our algorithm, including faster convergence to the source than other algorithms, flexibility in the choice of the loss function, and robustness to measurement modeling errors. Moreover, the performance improves as the number of sensors increases, showing the advantage of using multi-robots in our source-seeking algorithm. We also implement physical experiments to test the algorithm on small ground vehicles with light sensors, demonstrating success in seeking a moving light source.**

I. Introduction

Multi-agent source seeking is to use autonomous vehicles with measurement capabilities to locate a source of interest whose position is unknown. The source of interest can be a light source, a radio signal transmitter, or a chemical leakage point. The source-seeking vehicles, or mobile sensors, can measure the source's influence on the environment and use this information to locate the source.

A large body of source seeking research investigates field climbing methods [1]–[4]. Assuming the source signal gets stronger as the sensor-source distance shortens, mobile sensors can "climb" the source signal field to physically approach the source. These methods do not require explicit knowledge of the measurement model, making them easy to implement and generalizable to different applications. However, field climbing methods only exploit local information of the source field, as the sensors must maintain a tight formation to make a reasonable ascent direction estimate, as is the case in [3], [4]. Furthermore, to achieve a stable increase in measurement value, the sensors cannot move too fast as a group. Therefore field climbing methods are not necessarily the most effective source-seeking methods.

An alternative approach is to perform source identification/localization using various estimation methods, such as (Extended) Kalman Filter (EKF) [5], [6], Particle Filter (PF) [7], [8], and so on, to keep a constantly updated estimate of the source location over time. Despite requiring an explicit measurement model, these methods enable the fusion of measurements from multiple sensors to quickly identify a global view of the measurement field. To improve the estimation, studies on optimal sensor placement propose to optimize specific information metrics, in particular, variants of Fisher information measures [6], [7], [9]–[14], which relate closely to the famous Cramér-Rao lower bound [15], [16].

The authors are with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA. Emails: tzhang@g.harvard.edu, victor_l_qin@college.harvard.edu,{yujietang, nali}@seas.harvard.edu.

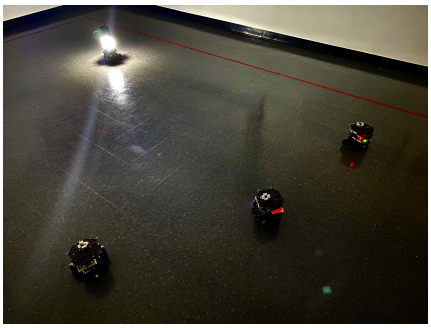


Fig. 1: A snapshot of seeking a moving light source in a dark room with three mobile light sensors.

However, these studies typically focus on deriving closed-form solutions of optimal sensor placement for particular types of measurement models [7], [11], [13]. Therefore, the result of one paper often does not generalize, not to mention that closed-form solutions might not exist for many general measurement models. Besides, a large number of the studies focus on sensors constrained to a restricted area [6], [10], [11], [14] rather than letting sensors move closer to the source, and the estimation methods are often not robust to the modeling error.

Contribution. This paper draws advantages from the two types of methods mentioned above (field climbing and source localization) to develop multi-robot source-seeking algorithms. In particular, we propose an algorithm using range-based measurements. Each iteration consists of three steps:

1) Collecting range-based measurements.

2) Performing source location estimation using the measurements.

3) Moving sensors along the gradient direction of the trace of the inverse of Fisher information.

The advantages and contributions of our methods can be summarized as,

- Our method improves the source location estimate while also moves sensors closer to the source. The source could be either stationary or moving. Compared with field climbing algorithms, numerical studies quantitatively demonstrate that our algorithm converges much faster to the source and performs more consistently over repeated trials. See IV-B.
- Our algorithm, especially its gradient-guided movement, provides flexibility in handling various measurement models and picking different types of information metrics as loss

functions. Section IV-C confirms at least three applicable metrics.

- The algorithm takes advantage of multi-sensors in the sense that the performance improves as the number of sensors increases, see IV-B.
- The algorithm is more robust to modeling error than the source localization with stationary sensors, see IV-D.
- We implement our algorithm on small ground vehicles carrying light sensors to seek a moving light source in a dark room, as shown in Figure 1. The hardware implementation further demonstrates the effectiveness of our algorithm. See IV-E.

A. Related Works

Field climbing methods. The field climbing problem stems from scientific research that studies animal behavior in exploring nutrition or chemical concentration fields [17], [18]. Inspired by these studies, some methods propose using field value measurements to estimate the field gradient and apply formation control to climb along the gradient [2], [3]. There are also methods not reliant on gradient estimates: [4] keeps the sensors in a circular formation and uses measured field values as directional weights to guide the overall movement; [19], [20] are based on Particle Swarm Optimization (PSO).

The main differences between field climbing and our algorithm are that i) field climbing maximizes the source field value, while ours exploits the Fisher information, an indicator of both estimation accuracy and source-sensor distance, ii) field climbing does not require knowing the measurement function, but ours does, iii) field climbing algorithms often demand a tight sensor formation for a stable field ascent, only exploiting local information, while our algorithm uses the sensors to collect global information for source localization. We will show in Section IV that the sensors under our algorithm tend to spread out to estimate the source better.

Source localization and optimal sensor placement. If a measurement model is available, various estimation methods can be applied to estimate source location, including EKF [5], [6], PF [7], [8], or even a reconstruction of the entire source field [9], [21]. To improve the estimation, people have studied optimal sensor placement by optimizing various information metrics including covariance [22], [23], mutual information [8], [24], and Fisher information measures [6], [7], [9]–[14] such as the determinant of Fisher information, the largest eigenvalue of its inverse, and the trace of its inverse (D-, E-, and A-optimality criterion respectively).

Our method also employs Fisher information but is different from the previous ones in that:

1) Most of the works focus on deriving closed-form solutions of optimal sensor placement for a particular type of measurement model, for example, RSS [11], pollutant diffusion [7], and gamma camera [13]. In contrast, our focus is not on providing closed-form solutions but instead providing a gradient-based method applicable to a large class of range-based measurement models. This method also provides flexibility in choosing different information metrics as loss functions, see IV-C.

2) Many previous works focus on finding the optimal angular placement of the sensors at a fixed distance to the source or on a restricted area [6], [10], [11], [14]. In contrast, we allow the sensors to move freely and eventually approach the source.

3) Some studies relax the restrictions on the sensor movement [12], [13]. However, these methods produce a spiraling sensor movement which is inefficient for source-seeking purposes, while our method does not produce such movement. Moreover, the way-point planning in these methods is based on the closed-form optimal placement solutions, but our method follows the gradient steps.

Bayesian inference and optimization. The recent advances in Bayesian learning inspire many studies to adopt the Bayesian methods for source seeking [25]–[27]. These studies view the environment as a field characterized by an (unknown) density function related to measurement and use Gaussian Process or other likelihood models as a surrogate to guide the sensor movements for new measurement collections. The computation (for running the posterior update and Bayesian optimization) and memory (for storing historical measurements as in non-parametric Bayesian methods) demands of these methods are often higher than our method. Overall, Bayesian-based methods and our method are designed from different principles. A detailed comparison is left for future work.

Finally, we would like to highlight that our work is significantly inspired by [6], which studies the optimal sensor placement problem on a surveillance boundary. We leverage their ideas to introduce the Fisher information in our objective, but we change the loss function from the determinant of Fisher information to the trace of its inverse. We also generalize the measurement model, so it fits our experiments and other real-world measurement settings. To handle such more general measurement models, the sensors only compute the gradient of the loss function rather than solving for its optimum at each time step.

II. Problem Setup

We consider the task of using a team of mobile sensors, with centralized coordination, to find a source whose location is unknown. For concreteness, in our experiment setting, the source is a lamp in a dark room, and the mobile sensors are ground vehicles carrying light sensors measuring the local light intensity. The sensor positions are assumed to be available. The goal is to let sensors move to the proximity of the source based on field measurements.

Specifically, we use $q \in \mathbb{R}^k$ to denote the source position, and use $p_1, p_2, \cdots, p_m \in \mathbb{R}^k$ to denote the sensor positions, where m is the number of sensors and k is the spatial dimension. We use the vector (in bold font) $\mathbf{p} = \begin{bmatrix} p_1^\top & p_2^\top & \cdots & p_m^\top \end{bmatrix}^\top$ to denote the joint location of all mobile sensors.

In this paper, we consider nonholomonic ground vehicles with $p_i = [p_i^1,\ p_i^2]^\top \in \mathbb{R}^2$. The states of mobile sensor i are $x_i := [p_i^1,\ p_i^2,\ \theta_i]^\top$ with θ being the robot orientation, and the control inputs u_i are the linear and angular velocities of the robot $u_i := [v_i,\ \omega_i]^\top$. The robot motion dynamics is given as

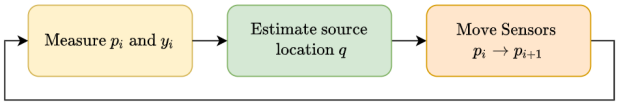


Fig. 2: The three consecutive steps in the proposed source seeking algorithm.

$$\dot{x} = \begin{bmatrix} \dot{p}^1 \\ \dot{p}^2 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} \tag{1}$$

In addition, we assume the measurement model is known, which is a reasonable assumption in our context since the sensor characteristics are usually known. Formally, the measurement made by the ith sensor, denoted by y_i, is generated by a known continuously differentiable function H_i subject to measurement noise ν_i:

$$y_i = H_i(p_i, q) + \nu_i. \tag{2}$$

The measurement function H_i may differ across different sensors, and we assume its value is independent of the locations of other sensors. We let $\mathbf{y} = [y_1, y_2, \ldots, y_m]^\top$ denote the vector of all measurement values, let $\boldsymbol{\nu} = [\nu_1, \nu_2, \ldots, \nu_m]^\top$ be the noise vector, and let $H : \mathbb{R}^{mk} \times \mathbb{R}^k \to \mathbb{R}^m$ be the mapping that describes the joint measurement made by the sensors, i.e.,

$$H(\mathbf{p}, q) = \begin{bmatrix} H_1(p_1, q) \\ \vdots \\ H_m(p_m, q) \end{bmatrix}. \tag{3}$$

Our measurement model can be rewritten as

$$\mathbf{y} = H(\mathbf{p}, q) + \boldsymbol{\nu}. \tag{4}$$

Given the information of sensor locations $\mathbf{p}$ and measurements $\mathbf{y}$, our objective is to have at least one of the sensors get within ϵ_0 distance to the source. In other words, we want to achieve

$$\|p_i - q\| \le \epsilon_0, \ \exists i \in \{1, 2, ..., m\}, \tag{5}$$

where ϵ_0 is some small positive number.

III. Algorithm

In this section, we will first present our algorithm, and then explain the rationale behind the algorithm design with theoretical justifications.

A. The Proposed Algorithm

Our source seeking algorithm consists of three consecutive steps in one iteration: measurement, source location estimation, and sensor movement, as illustrated in Figure 2 and detailed in Algorithm 1.

1) Measurement : In Lines 2–3 of Algorithm 1, the mobile sensors report their locations $\{p_i\}_{i=1}^m$ and latest measurements $\{y_i\}_{i=1}^m$, which then form the joint location vector $\mathbf{p}$ and the measurement vector $\mathbf{y}$.

2) Source Location Estimation: In Line 4 of Algorithm 1, using the information available (i.e., $\mathbf{p}$ and $\mathbf{y}$), the location estimation algorithm E produces the estimated source location $\hat{q}$. In our implementations, the Extended Kalman Filter (EKF) is employed as the estimation algorithm, which assumes that the source moves according to the following dynamics

$$s := \begin{bmatrix} q \\ \dot{q} \end{bmatrix}, \qquad \dot{s} = \begin{bmatrix} \dot{q} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} s. \tag{6}$$

This dynamics is sufficient to model the movement of a source that is moving slowly. At each time step, the EKF will take in the sensor measurements and sensor locations and return an estimate $\hat{q}$ of the source location. Please see [28, Definition 3.1] for details about the EKF update.

3) Sensor Movement: In Lines 5–10, the sensors move along the gradient descent directions of the following loss function

$$L(\mathbf{p}, q) = \mathrm{Tr}\left[\left(\nabla_q H(\mathbf{p}, q) \cdot \nabla_q H(\mathbf{p}, q)^\top\right)^{-1}\right]. \tag{7}$$

We will explain the motivation of this loss function later in Section III-B, by relating it to the Fisher information matrix and the Cramér-Rao lower bound [15], [16]. This step includes two parts: waypoint planning and waypoint tracking.

a) Waypoint planning: In Lines 5–8, with the source location estimate $\hat{q}$ obtained in Line 4, we generate a set of waypoints $\hat{\mathbf{p}}(0), \ldots, \hat{\mathbf{p}}(T)$ by applying gradient descent on L with respect to $\mathbf{p}$, with $q = \hat{q}$ fixed. Here in Line 7, $\alpha_t > 0$ is the step size and $M_t \succ 0$ is a directional regularization matrix.

b) Waypoint tracking: In Lines 9–10, after the waypoints $\hat{\mathbf{p}}(0), \ldots, \hat{\mathbf{p}}(T)$ have been calculated by the gradient descent on L, we extract the waypoints $\hat{p}_i(0), \ldots, \hat{p}_i(T)$ for each mobile sensor i, and use the motion planner MP to calculate a sequence of control inputs $u_i(1), \ldots, u_i(T)$ and apply the control at the first instance $u_i(1)$ to the corresponding mobile sensor.[1] The motion planner MP can

[1]This is inspired by the framework of receding horizon control/model predictive control [29] which works well when the tracking trajectory is time-varying.

Algorithm 1 Source Seeking by Exploiting Fisher information

Input: Small constant $\epsilon_0 > 0$, the location estimator $E : (\mathbf{y}, \mathbf{p}) \mapsto \hat{q}$, motion planner $MP : (\hat{p}_i(0), \hat{p}_i(1), \ldots, \hat{p}_i(T)) \mapsto (u_i(1), \ldots, u_i(T))$.

1: **repeat**
2: Get sensor locations p_i from all mobile sensors i, forming $\mathbf{p} = \left[p_1^\top, p_2^\top, \ldots, p_m^\top\right]^\top$.
3: Get measurement y_i from all mobile sensors i, forming $\mathbf{y} = [y_1, y_2, \ldots, y_m]^\top$.
4: Estimate the location of the source by $\hat{q} \leftarrow E(\mathbf{y}, \mathbf{p})$.
5: Set the initial waypoints $\hat{\mathbf{p}}(0) \leftarrow \mathbf{p}$.
6: **for** $t = 1$ **to** T **do**
7: $\hat{\mathbf{p}}(t+1) \leftarrow \hat{\mathbf{p}}(t) - \alpha_t M_t \nabla_{\mathbf{p}} L|_{\hat{\mathbf{p}}(t), \hat{q}}$.
8: **end for**
9: Extract waypoints $(\hat{p}_i(t))_{t=0}^T$ from $(\hat{\mathbf{p}}(t))_{t=0}^T$ for mobile sensor i, and generate the control inputs $(u_i(1), \ldots, u_i(T)) \leftarrow MP(\hat{p}_i(0), \ldots, \hat{p}_i(T))$.
10: Each mobile sensor i execute control input $u_i(1)$.
11: **until** $\min_{i=1,2...m}\{||p_i - \hat{q}||\} \le \epsilon_0$

be viewed as a device that transforms the planned waypoints into low-level actuation of a mobile sensor: by applying $(u_i(1), \ldots, u_i(T)) = MP(\hat{p}_i(0), \ldots, \hat{p}_i(T))$ to sensor i, the sensor will follow the trajectory of the waypoints $\hat{p}_i(0), \ldots, \hat{p}_i(T)$. The motion planner MP typically requires the knowledge of the sensors' dynamics to compute the control inputs, and any method that can fulfill this task can be a motion planner in Algorithm 1. In our implementation, the motion planner is a combination of spline-based motion generation described in [30] and the Linear Quadratic Regulator (LQR). Regarding the choice of the planning horizon T, in the Gazebo simulations and hardware implementation, we set $T = 20$ to ensure stability in sensor movement and robustness to disturbances. Meanwhile, in numerical studies where the robot dynamics is not simulated, it suffices to set $T = 1$.

Finally, the loop resets, and the sensors make a new set of measurements. The loop repeats until one sensor is within ϵ_0 distance to $\hat{q}$.

Remark 1. Note the algorithm aims to minimize $L(\mathbf{p}, \hat{q})$ rather than $L(\mathbf{p}, q)$, which raises the question of whether improving the former leads to a decrease in the value of the latter. Intuitively, it is expected that if $\hat{q}$ and q are sufficiently close to each other, i.e., the estimation error is small, then improving $L(\mathbf{p}, \hat{q})$ will decrease $L(\mathbf{p}, q)$. These two components of the algorithm rely on each other to function as a whole.

B. The Design Rationale of the Loss Function

In location-estimation-based source seeking, one uses the estimated location as a reference to move the sensors. Naturally, we want the location estimation to be as accurate as possible. However, the *Cramér–Rao Lower Bound* (CRLB) shows an intrinsic limitation that prevents the location estimation from being arbitrarily accurate. Formally, we have the following definition and theorem:

Definition 1 (Fisher Information)**.** Given the measurement function $H(\mathbf{p}, q)$, and assuming the measurement noise is Gaussian white, the Fisher information matrix relative to the source location q is

$$FIM = \nabla_q H(\mathbf{p}, q) \cdot \nabla_q H(\mathbf{p}, q)^\top.$$

Here $\nabla_q H(\mathbf{p}, q)$ denotes the $k \times m$ matrix whose ith column is equal to the (partial) gradient $\nabla_q H_i(p_i, q)$.

Theorem 1 (Cramér–Rao Lower Bound (CRLB) [15], [16])**.** *For any unbiased estimator $\hat{q}$ of q, the following matrix inequality holds*[2]

$$\mathbb{E}[(\hat{q} - q)(\hat{q} - q)^\top] \succeq FIM^{-1}. \tag{8}$$

Martinez and Bullo [6] built on this result to maximize $\det(FIM)$ in the hope of making FIM^{-1} small so that the estimation error is more likely to achieve low values. However, a large value of $\det(FIM)$ does not necessarily mean that the eigenvalues of FIM^{-1} will be uniformly small, and the lower bound provided by Theorem 1 might still be large. In Section IV-C, we show by simulation results that it is indeed not ideal for our application. Therefore, in this paper, we modify their idea and consider minimizing $\mathrm{Tr}(FIM^{-1})$ instead, which is also known as the *A-optimality criterion* [31] in literature.

The minimization of the loss function (7) will result in two consequences: First, by taking the trace on both sides of (8), we get $\mathbb{E}\big[\|\hat{q} - q\|^2\big] \geq \mathrm{Tr}\left(FIM^{-1}\right) = L(\mathbf{p}, q)$, i.e., the loss function is a lower bound on the mean squared error, and so by minimizing (7), we increase the chance of getting accurate source location estimates. Second, as we will show in the following subsection, a small value of the loss function implies the existence of at least one sensor that gets sufficiently close to the source under certain assumptions on the measurement model.

Remark 2. While we choose $\mathrm{Tr}(FIM^{-1})$ as the loss function in this paper, in principle one can also pick other information metrics as the loss function L in Algorithm 1. In Section IV-C, we will numerically test the performance using different options of the loss function.

C. Reaching the Source

We now provide analytical results showing that minimizing $\mathrm{Tr}(FIM^{-1})$ will indeed enforce some sensor to approach the source. For the sake of theoretical analysis, we need some extra assumptions on the measurement model. In particular, we assume that H_i is a monotonic function of the distance between sensors and the source.

Assumption 1. *We make the following assumptions on the measurement functions H_i:*

1) **Isotropic measurement***: The measurement values depend only on source-sensor distance, i.e.,*

$$H_i(p_i, q) = h_i(\|p_i - q\|) = h_i(r_i) \tag{9}$$

for some function $h_i : (0, +\infty) \to \mathbb{R}$, where $r_i \coloneqq \|p_i - q\|$.

2) **Monotonicity***: Each function $h_i(r)$ is nonnegative and is monotonically decreasing in r. In addition, as $r \to 0$, we have $h_i(r) \to +\infty$ as well as $|h_i'(r)| \to +\infty$.*

Here the term *isotropic* means that the measurement does not depend on source-sensor bearings, a.k.a. relative angles. The isotropic and monotonic properties hold for a wide range of functions that model the decay of measurement signals over distance, including, e.g., the function $h(r) = -\log r$ for RSS sensors, and the functions $h(r) = 1/r^b,\ b > 0$ for light sensors. We can then prove the following proposition, showing that minimizing $\mathrm{Tr}(FIM^{-1})$ indeed results in at least one sensor getting sufficiently close to the source.

Proposition 1 (Reaching the Source)**.** *Let $\hat{r}_i$ denote the unit direction vector from the source to the ith mobile sensor, i.e., $\hat{r}_i = (p_i - q)/\|p_i - q\|$. Suppose the smallest eigenvalue of $\sum_{i=1}^m \hat{r}_i \hat{r}_i^\top$ is not zero at all times. Then under Assumption 1, we have $\min_i r_i \to 0$ whenever $L(\mathbf{p}, q) \to 0$.*

We remark that for the case $k = 2$, i.e., the source and the mobile sensors move on a 2D plane, the smallest eigenvalue of $\sum_{i=1}^m \hat{r}_i \hat{r}_i^\top$ is not zero as long as there is more than one mobile sensor, and all mobile sensors together with

[2]Strictly speaking, an additional constant factor should be multiplied to the FIM as in Definition 1 for CRLB to hold. The constant is related to the covariance of measurement noise but is independent of $(q, \mathbf{p})$ and is inconsequential for our algorithm and results.

the source are not on the same straight line. The proof of Proposition 1 can be found in Appendix I.

IV. Experimental Results

This section illustrates the performance of our proposed algorithm through experiments. The algorithm performance under actual robot dynamics is studied in simulations in IV-A. In the other three numerical studies, we remove the robot dynamics in simulations to efficiently conduct repetitive trials and assume the sensors follow the gradient steps exactly. We study the influence of the number of sensors in IV-B, the difference of various information metrics in IV-C, and the robustness to modeling error in IV-D. Finally, we describe our real robot implementations in IV-E with a supplementary video recording the experimental results available at [32].

A. Gazebo Numerical Experiments

The following numerical experiments are carried out using the Gazebo simulation toolbox [33], with virtual mobile sensors simulating the same dynamics as the actual robots. We generate simulated measurement values of the sensors by $y_i = 1/r_i^2 + \nu_i$, with ν_i drawn independently from $\mathcal{N}(0, 0.01)$. The measurement function $h_i(r) = 1/r_i^2$ is given to the EKF for estimation.

1) Stationary Source: In the first set of simulations, we use three mobile sensors to seek a stationary source. The source is fixed at position $(6.0, 6.0)$, while the mobile sensors are initially placed at $(1.0, 2.0), (2.0, 2.0), (3.0, 2.0)$. The initial guess of source location given to the EKF is $(3.0, 4.0)$. The terminal condition threshold $\epsilon_0 = 0.5$. We compare the convergence to a stationary source among three algorithms: (a) our algorithm; (b) the field climbing algorithm introduced by [4] that only maximizes measured signal strength; (c) following straight lines to the estimated location. (c) is included to show the importance of exploiting Fisher information in obtaining accurate estimation. The results are displayed in Figure 3.

First, notice that (c) fails to converge to the source, as shown in Figure 3c. We suspect the reason is sensors cluster together quickly as they move to the (same) estimated location and cannot provide sufficiently rich, diverse measurements for a reasonable estimation. Consequently, the estimate gradually deviates from the source location and so follows the sensors. On the other hand, if taking a trajectory that improves the Fisher information, the sensors cover the space more thoroughly, resulting in a stable decrease in the estimation error and the final success of reaching the source, as shown in Figure 3a.

Comparing Figure 3a and 3b, note that sensors using our algorithm first spread out to estimate the source location better, then converge to the source, whereas sensors doing field climbing maintain a tight formation while steadily approach the source. Since we use constant rather than diminishing step sizes for 3b, the virtual robots do not stop entirely near the source and perform a looping behavior. Although our algorithm and the field climbing algorithm [4] are both successful with a stationary source, our algorithm converges faster consistently over repetitive trials, as is shown later in IV-B. Our algorithm also outperforms field climbing in seeking a moving source, as is shown next.

2) Moving Source: All parameters are kept the same as the stationary case in this experiment set, except the source moves in a circular motion with constant speed. See Figures 4a, 4b, 4c. Note (c) again leads to a sensor formation that causes the estimation to deviate from the actual source location quickly. Both (a) and (b) successfully get close to the source. However, the field climbing method exhibits unnecessary irregular motion when the sensors are near the source. We suspect that as the sensors get close to the source, the field climbing direction becomes very sensitive to the source movement. In comparison, sensors following our algorithm trace much more stable paths. We later discuss more advantages of (a).

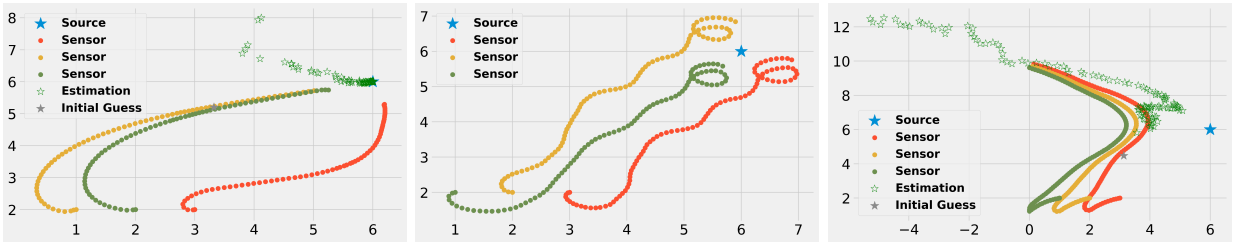
B. Performance with Sensor Swarms

We now investigate the influence of the number of mobile sensors on the algorithm performance. We perform repetitive simulations to study the convergence rate of source-sensor distances and the evolution of estimation error. We use a stationary source fixed at position $(6.0, 6.0)$ and randomize the initial sensor locations following uniform distribution in a 3.0-by-3.0 rectangle, as shown in Figure 6a.

The first set of experiments studies the convergence rate of *the sensors to the source*. We look at mobile sensor team performance with sizes at 3, 10, 20, 50, and also compare our algorithm with two other field climbing methods [3] [4]. The results in Figure 5 show that our algorithm converges much faster than the others. In addition, we see that increasing the number of sensors reduces the variance of all algorithms substantially while not affecting the convergence time. Because the speed of sensors has an upper bound regardless of the number of sensors, the overall convergence rate of source-sensor distance is limited. But having more sensors provides more measurements, which reduces the variance and results in more consistent, stable trials. Our algorithm benefits the most from having more sensors since more measurements contribute to better estimation, thus faster convergence with smaller variance.

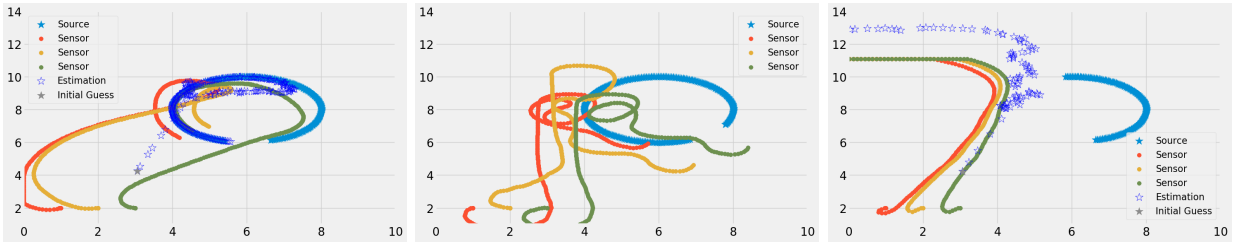
To further test the above conjecture that sensors moving *freely* and *in the direction of improving Fisher information* brings richer measurement and therefore enhances the estimation, we conduct the following three experiments: 1) The sensors move freely guided by our proposed algorithm; 2) The sensors are restricted to stay outside a radius of 3.0 from the source, performing projected gradient descent of our proposed loss function at the boundary; 3) The sensors do not move but only perform location estimation. The results are shown in Figure 6. We see that a larger team corresponds to a faster decline in the expected estimation error with a smaller variance. Moreover, sensors moving freely as in Figure 6b results in a quicker decrease in estimation error with smaller variance.

Scalability. Note the current algorithm requires the sensors to communicate with a central coordinator to submit measurements and receive waypoints. System scalability then becomes one potential issue, as the increasing number of sensors puts a burden on communication/computation. One solution is to extend our algorithm to a distributed setting where sensors only communicate with local neighbors, perform distributed estimation such as distributed EKF [34], and



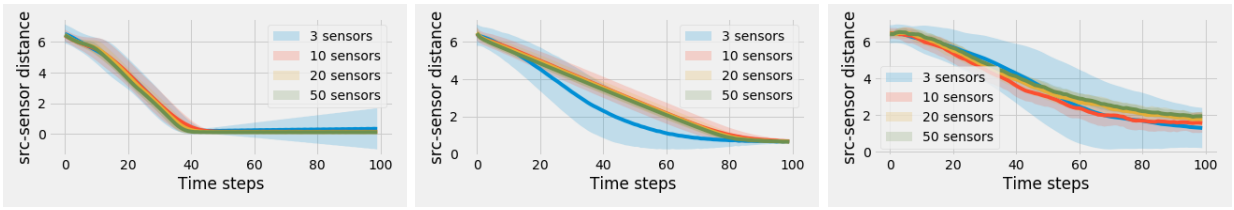
(a) Our Algorithm (b) Field climbing [4]. (c) Following straight lines to the estimated location.

Fig. 3: Comparing the general behavior of seeking a stationary source. The sensors in (b) do not make any estimations, since field climbing only uses measured signal strength to guide sensor movements.



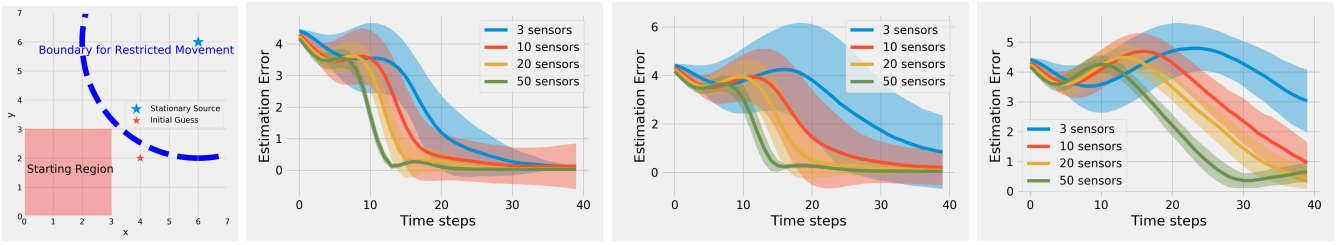
(a) Our Algorithm (b) Field climbing [4]. (c) Following straight lines to the estimated location.

Fig. 4: Comparing the general behavior of seeking a moving source.



(a) Our Algorithm (b) Circular formation field climbing [4]. (c) Climbing estimated gradient [3].

Fig. 5: The time evolution of the source-sensor distance. The color bands show the standard deviation of distances across 100 repetitive experiments.



(a) Experiment Setting (b) Sensors move freely. (c) Sensors stay outside the boundary. (d) Stationary sesnors.

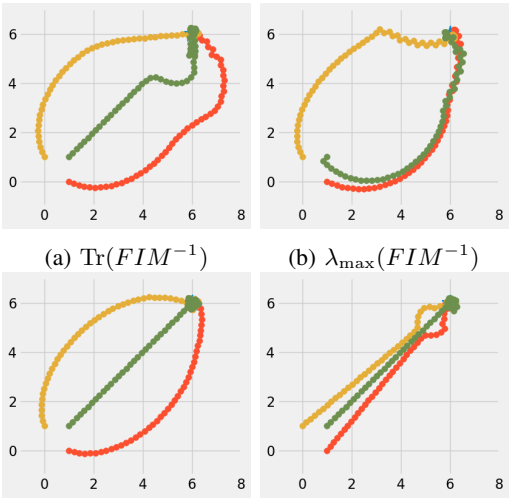
Fig. 6: The time evolution of source location estimation error. The color bands show the standard deviation of errors across 100 repetitive experiments.

move according to local Fisher information. This is beyond the scope of this paper and is left as future work.

C. Comparison of Different Information Metrics

Although we specify the loss function to be the A-optimality criterion in our algorithm, one can, in principle, replace it with other alternatives. In the following experiments, we test our algorithm's performance with four different loss functions: i) $\mathrm{Tr}(FIM^{-1})$; ii) $\lambda_{\max}(FIM^{-1})$; iii) $-\log\det(FIM)$, and iv) $\mathrm{Tr}\,P$ where P is the posterior covariance of the EKF defined by [22, Equation (15)], $P = (\nabla_q H^\top R^{-1} \nabla_q H + P_0^{-1})^{-1}$, with P_0 being the current estimation covariance and R being the measurement noise parameter of the EKF.

All loss functions are tested with a stationary source and three freely-moving sensors for 100 repetitive trials. Each trial is initialized randomly in the same way as in the previous subsection. Figure 7 shows the gradient descent trajectories for the tested metrics. Figure 8 shows that all the tested metrics except $-\log\det(FIM)$ yield relatively good



(a) $\mathrm{Tr}(FIM^{-1})$ (b) $\lambda_{\max}(FIM^{-1})$

(c) $-\log\det(FIM)$ (d) Covariance

Fig. 7: Assuming the source location is known, the above are the gradient descent trajectories of the tested information metrics.

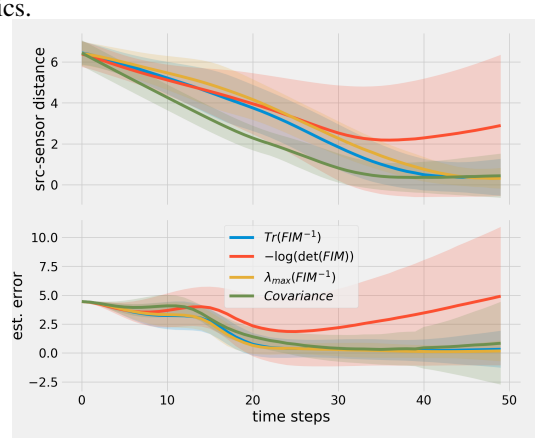


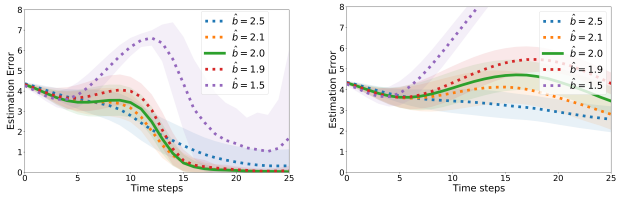
Fig. 8: Comparison of the algorithm performance using different information metrics.

performance when using our algorithm, with $\mathrm{Tr}(FIM^{-1})$ and $\lambda_{\max}(FIM^{-1})$ achieving a better balance in convergence and estimation than the covariance metric $\mathrm{Tr}\,P$. This confirms the generality of our algorithms.

D. Robustness to Measurement Modeling Error

We next investigate whether our algorithm can function despite the error in the measurement model. We simulate source seeking with 10 mobile sensors and a stationary source, in which the measurement is generated by $y = 1/r^2 + \nu$, $\nu \overset{i.i.d}{\sim} \mathcal{N}(0, 0.01)$. We provide imperfect measurement models to the EKF in the form of $h(r) = 1/r^{\hat{b}} = 1/r^{2+\Delta b}$, with Δb taking values in $0, \pm 0.1, \pm 0.5$. We study the estimation error in two settings: 2) The sensors move freely using our algorithm; 2) The sensors are stationary. The results are shown in Figure 9.

The robustness of our algorithm is two-fold: In Figure 9a, when compared to using the perfect measurement model ($\Delta b = 0$), our algorithm shows no significant degrade in estimation when Δb is small. Therefore our algorithm maintains



(a) Our Algorithm (b) Stationary sensors

Fig. 9: Robustness. The solid lines correspond to the perfect measurement model; the dotted lines are measurement models with errors. The color bands show the standard deviation of errors across 100 repetitive trials.

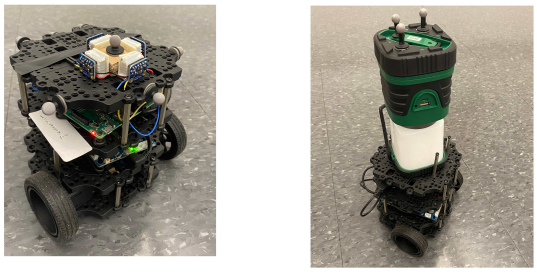


Fig. 10: TurtleBot3 Burger: Left is a mobile sensor with 4 light strength sensors installed on top; Right is the moving light source.

reasonable estimation quality even if the measurement model used is imperfect. Besides, comparing Figure 9a and Figure 9b, our algorithm shows more robustness than stationary sensors, whose estimations tend to divergence when Δb is at the value of 0.5.

E. Real Robot Implementation

We implement our algorithm on ROBOTIS TurtleBot3 Burger robots, as shown in Figure 10, to seek a light source in a dark room. Figure 1 is a snapshot of the experiment. We use one robot carrying a lamp as the light source and install light sensors on other robots to act as mobile sensors. A motion capture system records the mobile sensor locations in real-time. We use the Robotic Operating Systems (ROS) [35] to communicate measurement signals, sensor locations, and control inputs between the mobile sensors and a central computer through a WiFi router.

Before running the source seeking algorithm, we need to obtain the measurement model for location estimation. We fix the source at a known location q and collect a set of $\{(\mathbf{p}, q, \mathbf{y})\}$ to fit the measurement functions h_i for each mobile sensor. The measurement model we use is

$$h_i(r_i) = k_i(r_i - C_{1,i})^{-b_i} + C_{0,i} \tag{10}$$

, where y_i, r_i are the measured data from the ith mobile sensor, and $k_i > 0$, $b_i > 0$, $C_{0,i}$, $C_{1,i}$ are the associated model parameters.

In the supplementary video clip [32], we demonstrate the mobile sensors implementing our algorithm locate and follow the moving light source in a dark room. More details of our implementation can be found in Appendix II.

V. Conclusion and Future Directions

This paper develops a multi-robot source-seeking algorithm that exploits the Fisher information associated with the source location estimation to guide the sensors' movement. We show in theoretical analysis that improving the trace of the inverse Fisher information benefits the source location estimation and naturally leads sensors to converge to the source. The algorithm is verified both numerically and physically on small ground vehicles.

There are several interesting future directions. For example, besides the scalability issue discussed in Section IV-B, another vital point to be addressed in future work is collision avoidance, which can be incorporated into our algorithm by, for example, adding a collision penalty to the loss function or employing a reactionary actuation control on top of waypoint planning. We expect that collision avoidance can improve the robustness of our algorithm without hindering either the convergence rate or estimation accuracy drastically. In addition, comparison to other types of source-seeking methods such as Bayesian optimization needs more investigation. Some hybrid versions between these methods and our method might draw advantages from both sides.

References

[1] B. A. Angélico, L. F. O. Chamon, S. Paternain, A. Ribeiro, and G. J. Pappas, "Source seeking in unknown environments with convex obstacles," *arXiv preprint arXiv:1909.07496*, 2019.

[2] R. Bachmayer and N. E. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 1, 2002, pp. 112–117.

[3] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.

[4] B. J. Moore and C. Canudas-de Wit, "Source seeking via collaborative measurements by a circular formation of agents," in *Proceedings of the 2010 American Control Conference*, 2010, pp. 6417–6422.

[5] F. Morbidi and G. L. Mariottini, "Active target tracking and cooperative localization for teams of aerial vehicles," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1694–1707, 2013.

[6] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[7] B. Bayat, N. Crasta, H. Li, and A. Ijspeert, "Optimal search strategies for pollutant source localization," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1801–1807.

[8] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2010.

[9] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Model-based active source identification in complex environments," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 633–652, 2019.

[10] D. Moreno-Salinas, A. Pascoal, and J. Aranda, "Optimal sensor placement for multiple target positioning with range-only measurements in two-dimensional scenarios," *Sensors*, vol. 13, no. 8, pp. 10 674–10 710, 2013.

[11] S. Xu, Y. Ou, and W. Zheng, "Optimal sensor-target geometries for 3-D static target localization using received-signal-strength measurements," *IEEE Signal Processing Letters*, vol. 26, no. 7, pp. 966–970, 2019.

[12] A. N. Bishop and P. N. Pathirana, "Optimal trajectories for homing navigation with bearing measurements," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 12 117–12 123, 2008.

[13] M. S. Lee, D. Shy, W. R. Whittaker, and N. Michael, "Active range and bearing-based radiation source localization," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1389–1394.

[14] S. Ponda, R. Kolacinski, and E. Frazzoli, "Trajectory optimization for target localization using small unmanned aerial vehicles," in *AIAA Guidance, Navigation, and Control Conference*, 2009, doi: 10.2514/6.2009-6015.

[15] H. Cramér, *Mathematical Methods of Statistics*. Princeton University Press, 1946.

[16] C. R. Rao, "Information and the accuracy attainable in the estimation of statistical parameters," *Bulletin of the Calcutta Mathematical Society*, vol. 37, no. 3, pp. 81–91, 1945.

[17] V. Gazi and K. Passino, "Stability analysis of social foraging swarms: combined effects of attractant/repellent profiles," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, 2002, pp. 2848–2853.

[18] A. Okubo, "Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds," *Advances in Biophysics*, vol. 22, pp. 1–94, 1986.

[19] R. Zou, V. Kalivarapu, E. Winer, J. Oliver, and S. Bhattacharya, "Particle swarm optimization-based source seeking," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 865–875, 2015.

[20] Y. Cai and S. X. Yang, "An improved PSO-based approach with dynamic parameter tuning for cooperative target searching of multi-robots," in *2014 World Automation Congress (WAC)*, 2014, pp. 616–621.

[21] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Distributed reduced order source identification," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 1084–1089.

[22] C. Yang, L. Kaplan, and E. Blasch, "Performance measures of covariance and information matrices in resource management for target state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2594–2613, 2012.

[23] K. Zhou and S. I. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 678–695, 2011.

[24] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-theoretic planning with trajectory optimization for dense 3D mapping." in *Proceedings of Robotics: Science and Systems*, 2015, doi: 10.15607/RSS.2015.XI.003.

[25] Y. A. Prabowo, R. Ranasinghe, G. Dissanayake, B. Riyanto, and B. Yuliarto, "A bayesian approach for gas source localization in large indoor environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4432–4437.

[26] C. Sánchez-Garrido, J. G. Monroy, and J. G. Jiménez, "Probabilistic estimation of the gas source location in indoor environments by combining gas and wind observations." in *APPIS*, 2018, pp. 110–121.

[27] A. Benevento, M. Santos, G. Notarstefano, K. Paynabar, M. Bloch, and M. Egerstedt, "Multi-robot coordination for estimation and coverage of unknown spatial fields," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7740–7746.

[28] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, "Stochastic stability of the discrete-time extended kalman filter," *IEEE Transactions on Automatic control*, vol. 44, no. 4, pp. 714–728, 1999.

[29] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[30] R. Walambe, N. Agarwal, S. Kale, and V. Joshi, "Optimal trajectory generation for car-type mobile robot using spline interpolation," *IFAC–PapersOnLine*, vol. 49, no. 1, pp. 601–606, 2016.

[31] D. Ucinski, *Optimal measurement methods for distributed parameter system identification*. CRC press, 2004.

[32] T. Zhang, "Following a moving source(with description)," Mar. 2021. [Online]. Available: https://www.youtube.com/watch?v=mwrKunptnZQ

[33] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2149–2154.

[34] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 5492–5498.

[35] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. Version ROS Melodic Morenia, 2018-05-23. [Online]. Available: https://www.ros.org

APPENDIX I
PROOF OF PROPOSITION 1

Recall that $\hat{r}_i$ is the directional unit vector pointing from q (the source) to p_i (sensor i's position), i.e.,

$$\hat{r}_i = \frac{p_i - q}{\|p_i - q\|}.$$

We first derive expressions for FIM and the loss function L.

Lemma 1. *Suppose individual measurements are isotropic as in Assumption 1. Then*

$$FIM = \sum_{i=1}^{m} |h_i'(r_i)|^2 \hat{r}_i \hat{r}_i^T. \tag{11}$$

Moreover,

$$L(\boldsymbol{p}, q) = \mathrm{Tr}(FIM^{-1}) = \sum_{i=1}^{m} \frac{1}{\lambda_i(FIM)}, \tag{12}$$

where $\lambda_i(FIM)$ is the ith eigenvalue of FIM.

Proof. By noticing that $r_i = \|p_i - q\|$ and using the chain rule of calculus, one can show that

$$\nabla_q h_i(p_i, q) = -h_i'(r_i)\, \hat{r}_i.$$

Denote $A = \nabla_q H(\mathbf{p}, q)$ and $A_i = \nabla_q h_i(p_i, q)$. Then from the definition of H in (3), we have

$$A = \left[A_1 | A_2 | \cdots | A_m\right],$$

and so

$$FIM = AA^T = \sum_{i=1}^{m} A_i A_i^T = \sum_{i=1}^{m} |h_i'(r_i)|^2 \hat{r}_i \hat{r}_i^T.$$

The equality (12) follows since the eigenvalues of FIM^{-1} are the reciprocals of those of FIM. □

With the formulas of L and FIM, by further imposing the monotonicity property of h_i in Assumption 1.2, we can show that minimizing L leads to reaching the source as stated in Proposition 1.

Proof of Proposition 1. First, we take the maximum over all the summation coefficients in FIM to get

$$FIM \preceq \max_{i=1,2\ldots m} |h_i'(r_i)|^2 \sum_i \hat{r}_i \hat{r}_i^T$$

Let $\lambda_{\min}(P)$ denote the smallest eigenvalue of a positive semi-definite matrix P. Then by (12), we can see that

$$\frac{1}{\lambda_{\min}(FIM)} \leq L(\mathbf{p}, q) \leq \frac{k}{\lambda_{\min}(FIM)},$$

where we recall that k is the dimension of q and also the size of FIM. Consequently,

$$\begin{aligned} \frac{1}{L(\mathbf{p}, q)} &\leq \lambda_{\min}(FIM) \\ &\leq \max_i |h_i'(r_i)|^2 \cdot \lambda_{\min}\left(\sum_{i=1}^{m} \hat{r}_i \hat{r}_i^T\right). \end{aligned}$$

Now suppose $\lambda_{\min}(\sum_{i=1}^{m} \hat{r}_i \hat{r}_i^T) > 0$ holds for all times. Then because $\lambda_{\min}(\sum_{i=1}^{m} \hat{r}_i \hat{r}_i^T) \leq m$, we have

$$\frac{1}{m \cdot L(\mathbf{p}, q)} \leq \max_i |h_i'(r_i)|^2, \tag{13}$$

which implies that $\max_i |h_i'(r_i)| \to \infty$ as $L(\mathbf{p}, q) \to 0$. By Assumption 1, $|h_i(r_i)|$ monotonically increases to infinity as $r_i \to 0$ (and vice versa), and so we have $\min_i r_i \to 0$ whenever $L(\mathbf{p}, q) \to 0$, which completes the proof. □

APPENDIX II
DETAILS OF THE REAL ROBOT IMPLEMENTATION

The movement of the sensors is dictated by the central computer, which runs a comprehensive control pipeline consisting of three main components: location estimation, waypoint planner, and motion planner.

The first component is an Extended Kalman Filter assuming the measurement model (10). The filter state $\hat{z}_t$ is defined as the estimated position and velocity of the source,

$$\hat{z}_t \equiv [\hat{q}_t, \hat{v}_t] \tag{14}$$

The filter uses the constant-velocity dynamics as the model for source movement

$$\hat{z}_{t+1} = [\hat{q}_t + \hat{v}_t, \hat{v}_t] \tag{15}$$

Though the true dynamics of the source can be much more complicated, the model (15) is shown to perform well in localizing sources moving at a low speed.

The central computer gathers measurement values from all the sensors to perform filter update at a 10Hz frequency, generating new estimations of the source location. A reasonable range of measurement and process noise parameter for the EKF is $0.01I \sim I$.

The second component is a waypoint planner that plans a $T-$step discrete trajectory for every sensor following Algorithm 1, based on the current source estimate and sensor locations. Here the length T is typically taken to be ~ 20, with step size ≈ 0.22, matching the maximum speed of the robots we use. To ensure a stable sensor movement, the waypoints are re-planned at a relatively low frequency, typically within $0.5 \sim 5$Hz.

The third component is an motion planner that generates control actions for the sensors to track the waypoints specified by the second component. The core of the third component are:

1)A trajectory smoothing algorithm that fits a smooth cubic spline that goes through the discrete waypoints. For details, please refer to [30].

2)The LQR algorithm that generates a sequence of angular/linear velocities that tries to realize the smoothened trajectory given by 1). Specifically, we use (1) as the motion model for the sensors, and linearize the model around the smoothened trajectory, then apply LQR on the linearization. The reasonable choice of Q and R matrices are in the range $0.1I \sim I$. The combination of trajectory smoothing and LQR is shown to produce much more stable and smoother sensor movements than other methods, for example, PID control for waypoint tracking.

The third component re-plans new actions at a much higher frequency than the second component, typically at

$10 \sim 20$ Hz. It usually plans for action sequences with length ~ 20, with time interval between two consecutive actions around $0.05 \sim 0.1$s, matching the planning frequency. In this way, the sensors only execute their first planned actions, then the entire control sequences are re-planned to correct the effect of disturbances or to accommodate updated waypoints. Note this is very different from our simulations, where the waypoint planner only plans one-step ahead and the sensors are assume to move exactly to the waypoints so motion planning is unnecessary. The main reason of planning with a larger look-ahead length, and with different frequencies between different components, is to ensure stability and robustness as real-world issues add complication to the system, such as robot dynamics, unforeseen disturbances, communication, and so on.