A Distributed Scalar Field Mapping Strategy for Mobile Robots

Tony X. Lin[1], Said Al-Abri[1], Samuel Coogan[2], Fumin Zhang[1]

***Abstract*— This paper proposes a distributed field mapping algorithm that drives a team of robots to explore and learn an unknown scalar field. The algorithm is based on a bio-inspired approach known as Speeding-Up and Slowing-Down (SUSD) for distributed source seeking problems. Our algorithm leverages a Gaussian Process model to predict field values as robots explore. By comparing Gaussian Process predictions with measurements of the field, agents search along the gradient of the model error while simultaneously improving the Gaussian Process model. We provide a proof of convergence to the gradient direction and demonstrate our approach in simulation and experiments using 2D wheeled robots and 2D flying autonomous miniature blimps.**

***Index Terms*— Scalar Fields, Mapping and Exploration, Gaussian Process Regression, Multi-agent Control.**

I. Introduction

Scalar field maps are used to describe the spatial behavior of an environmental characteristic of interest. For example, scalar maps of oil spill concentrations, chemical leaks, or occupancy provide useful information for environmental monitoring and path-planning applications [1, 2, 3] and may be used to provide intuition about the field's behavior in unknown regions. As a result, designing exploration strategies that can efficiently produce such maps is an important problem. In particular, finding the critical points of an unknown field is of great interest as samples near critical points help to define the field's spatial structure and may coincide with points of interest, like the source of a chemical leak.

In the literature, coverage strategies to sample a scalar field with an appropriate density have been explored. In [4], robot formation controllers are proposed to achieve a desired distribution sampling, and in [5], exhaustive search methods are reviewed that can guarantee visiting every point within some distance over a bounded domain. However, the first approach may fail to capture important details as the desired distribution does not evolve according to the field samples and exhaustive search methods may be prohibitively time-consuming depending on the size of the domain. Alternatively, the work [6, 7] demonstrates a search strategy that considers online field samples in order to balance exploration of the field with tight sampling near critical points. The strategy requires some prior knowledge of the environment though and is therefore not applicable in many situations where the scalar field is completely unknown.

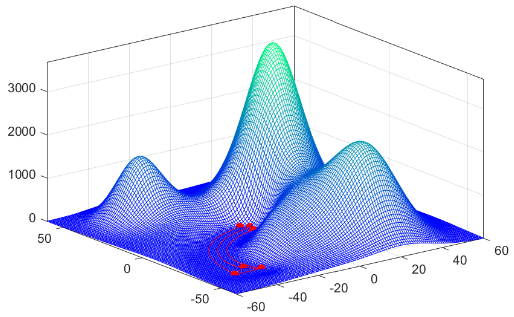

Fig. 1. 3D mesh plot of the Gaussian Process prediction error and three agents searching the field for large error. Previously explored regions have low modeling error due to collected samples. Agents start at around the position $(-40, -40)$.

Other search strategies based on Gaussian Process (GP) learning methods (also known as Kriging [8]) have also been described in the literature. The paper [9] describes a frontier-based approach based on the variance of the GP model and the paper [10] describes an RRT path-planning approach that incorporates the occupancy uncertainty of the GP. Both strategies require computing an information metric over either the frontiers or RRT trajectories which may be prohibitively expensive. To handle these expensive computations, [11] proposes a distributed estimation strategy of the joint entropy which can then be used to drive exploration objectives.

Source-seeking strategies have also been proposed to find the extrema of scalar fields. The paper [12] describes a distributed gradient tracking strategy that uses circular swarm formations to estimate the field gradient. However, explicit gradient estimation methods may require many local samples and might be ill-conditioned. As such, our previous works in [13, 14, 15, 16] demonstrate a derivative-free distributed source seeking algorithm that allows agents to follow the negative gradient. While agents following source-seeking strategies may efficiently find local extrema, agents may never leave and may fail to explore the rest of the field.

We, therefore, propose a novel field exploration algorithm based on our previous bio-inspired algorithm Speeding-Up and Slowing-Down (SUSD). Our approach does not require computing a joint information metric or gradient estimate. Instead, our approach uses a GP model that is refined online to predict the field at each iteration. By computing the modeling error between the predicted and actual values of the field, the SUSD algorithm drives the robots to search along the modeling error gradient without

[1] Tony X. Lin, Said Al-Abri, and Fumin Zhang are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA `tlin339, saidalabri@gatech.edu, fumin@gatech.edu`

[2] Samuel Coogan is with the School of Electrical and Computer Engineering and the School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA `sam.coogan@gatech.edu`

explicitly estimating it. Due to the unknown spatial structure of the scalar field, we treat the higher-order components of the field as a disturbance and provide an input-to-state stability proof of the gradient tracking controller. This sets the SUSD algorithm as an efficient exploration strategy that prioritizes minimizing the extrema in the field modeling error which may improve the GP learning process. The proposed method only requires agents to be able to share samples of the scalar field, observe the relative positions of other agents, and compute the posterior of the GP using the previous field samples. An example of the proposed approach with three simulated agents is shown in Fig. 1.

Our main contributions are threefold: **i)** a scalar function transformation that enables our previous SUSD exploitation algorithm to integrate with a GP prediction model to form an exploration strategy for mapping a smooth scalar field, **ii)** derivation of the SUSD dynamics in the existence of the GP prediction model and an Input-to-State Stability proof that shows agents track the gradient of the modeling error when the gradient is sufficiently large, and **iii)** simulations and experiments that contains the Georgia Tech Miniature Autonomous Blimps [17], as well as the Georgia Tech Robotarium [18].

The outline of the paper is as follows. Section II formulates the field-learning problem for multi-agent systems. Section III provides some preliminary detail on Gaussian Process Regression methods which are used in this paper to achieve field mapping. Section IV proposes the exploration strategy that extends our previous SUSD method and Section V provides an analysis of convergence to the error gradient. Section VI describes validation trials using a swarm of simulated agents, the Georgia Tech Miniature Autonomous Blimps, and the Robotarium. We conclude with Section VII and describe our future works.

II. Problem Formulation

Consider a swarm of N agents in 2-D space where $\boldsymbol{r}_i \in \mathbb{R}^2$, $i = 1, \dots, N$ denotes the position of the i^{th} agent. In this paper, we assume the following:

Assumption 2.1: Each agent i is able to observe the relative position $(\boldsymbol{r}_j - \boldsymbol{r}_i)$, $j = 1, \dots, N$.

Assumption 2.2: Each agent i is able to identify its own position $\boldsymbol{r}_i$ with respect to a fixed frame.

In practice, *Assumption 2.1* and *Assumption 2.2* may be realized with sensors that can measure relative positions (such as a camera or LiDAR system) and can achieve localization (such as by GPS).

Assumption 2.3: The field $z : \mathbb{R}^2 \to \mathbb{R}$ is bounded over a search domain $\mathscr{D} \subset \mathbb{R}^2$, i.e. $0 \leq z_{min} \leq z(\boldsymbol{r}) \leq z_{max}, \forall \boldsymbol{r} \in \mathscr{D}$. We also assume agents are able to measure the field at their current positions, i.e. $z(\boldsymbol{r})$. While we assume the field is bounded, the bounds of z do not need to be known. We further assume agents have a local communication system that allows them to share measurements with all other agents.

Assumption 2.4: Each agent retains a history of location and measurement pairs of all agents in the swarm, i.e. $\bar{\boldsymbol{r}}^{(m)} = \cup_{i=1}^{N} \{(\boldsymbol{r}_i(j\Delta t),\, z(\boldsymbol{r}_i(j\Delta t)))\}_{j=1}^{m}$ where m is the current sampling iteration such that $t - \Delta t \leq m\Delta t < t$.

With a fixed sampling interval Δt, we assume agents improve the estimate of the field model by aggregating the field measurements of all agents every Δt units of time. Let $\hat{z}^{(m)} : \mathbb{R}^2 \to \mathbb{R}$ be the GP estimate of the field associated with $\bar{\boldsymbol{r}}^{(m)}$ and let the velocity of each agent be described by

$$\dot{\boldsymbol{r}}_i(t) = \boldsymbol{u}_i(t) \tag{1}$$

Then, the problem we are interested in solving is designing a velocity controller $\boldsymbol{u}_i(t)$ so that the model error $(z(\boldsymbol{r}) - \hat{z}^{(m)}(\boldsymbol{r})) \to 0$ as $t \to \infty$ for all $\boldsymbol{r} \in \mathscr{D}$.

Remark 1: We note that the problem of designing a controller such that $(z(\boldsymbol{r}) - \hat{z}^{(m)}(\boldsymbol{r})) \to 0$ as $t \to \infty$ for all $\boldsymbol{r} \in \mathscr{D}$ could be solved using other exhaustive search methods that can guarantee visiting every point in the space $\mathscr{D}$ within a distance ξ [5]. However, for extremely large spaces, these approaches may take a prohibitively long period of time and may not construct useful approximations of the field until the search is nearly complete. As such, we are interested in designing a controller that adaptively samples the field in order to construct a model that captures the major characteristics efficiently before refining the estimate.

III. Gaussian Process Regression

In this section, we introduce the Gaussian Process (GP) learning method which we use to model the unknown field z. GPs are a non-parametric supervised learning methods that can be used for the regression of nonlinear maps $g : \mathbb{R}^d \to \mathbb{R}$. These methods use labeled sample points to define a posterior over a space of functions given a prior function $\mu : \mathbb{R}^d \to \mathbb{R}$ and a covariance function, also known as a kernel, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that defines the similarity between sample points [8]. Given a dataset $\mathscr{D} = \{(\boldsymbol{x}_i, \hat{g}(\boldsymbol{x}_i))\}_{i=1}^{M}$ where $\hat{g}(\boldsymbol{x}) = g(\boldsymbol{x}) + \omega$ and $\omega \sim \mathscr{N}(0, \sigma^2)$, the Gaussian posterior of a set of query points $\{\boldsymbol{x}_j^*\}_{j=1}^{L}$ is

$$\begin{gathered} g(\boldsymbol{x}^*) = \mu(\boldsymbol{x}) + K(\boldsymbol{x}^*, \boldsymbol{x})^\intercal (K(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2 I)^{-1} \hat{g}(\boldsymbol{x}) \\ \mathbb{V}[g(\boldsymbol{x}^*)] = K(\boldsymbol{x}^*, \boldsymbol{x}^*) - K(\boldsymbol{x}^*, \boldsymbol{x})^\intercal (K(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2 I)^{-1} K(\boldsymbol{x}^*, \boldsymbol{x}) \end{gathered} \tag{2}$$

Let $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i=1}^{M}$ and $\boldsymbol{y}' = \{\boldsymbol{y}'_i\}_{i=1}^{N}$ be two sample data sets. The matrix $K(\boldsymbol{y}, \boldsymbol{y}')$ is defined as

$$K(\boldsymbol{y}, \boldsymbol{y}') = \begin{bmatrix} k(\boldsymbol{y}_1, \boldsymbol{y}'_1) & \dots & k(\boldsymbol{y}_M, \boldsymbol{y}'_1) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{y}_1, \boldsymbol{y}'_N) & \dots & k(\boldsymbol{y}_M, \boldsymbol{y}'_N) \end{bmatrix} \tag{3}$$

For this work, we use the squared exponential kernel defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \alpha^2 \exp\Big(\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|}{2\beta^2}\Big) \tag{4}$$

and use a prior $\mu(\boldsymbol{x}) = 0$. The parameters α and β are hyperparameters of the GP and are often optimized according to the given data. In our approach, we assume $\sigma = 0$ and the hyperparameters are fixed and may be chosen if a small sampling of the field is known before runtime.

IV. The SUSD Field Exploration Strategy

In this section, we introduce the SUSD-based field exploration algorithm. The proposed approach allows the SUSD source seeking algorithm to search for model errors over a bounded domain.

A. Field Transformation

We propose here a field transformation that incorporates an online learning model $\hat{z}^{(m)}$, which models the underlying field of interest z, realized with a GP with a smooth posterior, which occurs with an appropriate choice of kernel (for example, the squared exponential). For a scalar field that meets *Assumption 2.3*, we apply the field transformation $f : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f^{(m)}(\boldsymbol{r}) = \frac{1}{(z(\boldsymbol{r}) - \hat{z}^{(m)}(\boldsymbol{r}))^2 + \varepsilon} + \eta \tag{5}$$

where $\varepsilon > 0$ and $\eta \geq 0$ are constant positive terms to design the bounds of $f^{(m)}(\boldsymbol{r})$. The transformation (5) inverts the squared modeling error of $\hat{z}^{(m)}$ such that the field is now a smooth field with local maxima at points of smallest modeling error. This allows us to transform the field exploration problem into a local minima source seeking problem where local minima are eliminated as we improve the GP with sampling. We now describe the design of a distributed source-seeking controller that leverages (5).

B. Velocity Design

For the swarm of agents as described in Section II and the transformed field (5), the SUSD-based search strategy is

$$\boldsymbol{u}_i(t) = k_1 \boldsymbol{n}(t) f^{(m)}(\boldsymbol{r}_i(t)) \tag{6}$$

where $k_1 > 0$ is a tuning gain and $\boldsymbol{n}(t)$ is the eigenvector corresponding to the smallest eigenvalue $\lambda_n(t)$ from the position covariance matrix of all agent positions. The position covariance matrix for all agents at time t (as all agents have the same covariance matrix) is computed as

$$\boldsymbol{C}(t) = \sum_{j=1}^{N} (\boldsymbol{r}_j(t) - \boldsymbol{r}_c(t))(\boldsymbol{r}_j(t) - \boldsymbol{r}_c(t))^\intercal \tag{7}$$

where $\boldsymbol{r}_c(t) = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{r}_j(t)$. Computing the eigenvectors $(\boldsymbol{q}(t), \boldsymbol{n}(t))$, and the corresponding eigenvalues $(\lambda_q(t), \lambda_n(t))$ is done through the Principal Component Analysis (PCA) algorithm [19]. Employing (6), agents move along the $\boldsymbol{n}(t)$ direction, which we call the SUSD direction, and naturally turn towards the negative gradient of the transformed field. If we remove the GP sampling, i.e. $\hat{z}^{(m)} = \hat{z}^{(0)}$, then our approach employs the normal distributed source-seeking strategy as in [15]. Additional details regarding the SUSD source-seeking strategy are in our previous work [14, 15, 16]. The full field-learning algorithm is provided in Algorithm 1.

Remark 2: Observe that in (5), the transformed field has multiple local maxima at locations where $(z(\boldsymbol{r}) - \hat{z}(\boldsymbol{r})) = 0$. We will show in Section V that the controller (6) drives the swarm to move against the negative gradient direction of the transformed field (5). Hence, the SUSD-based controller in this paper drives the swarm away from local maxima to explore landscapes of large modeling errors. This is different from our previous works in [14, 15, 16] where the swarm is designed to converge to one local maximum (or minimum).

Algorithm 1: One Iteration of Field-Learning SUSD

```
Input  :  $t$, $m$, $z$, $\{(\boldsymbol{r}_j, z_j)\}_{j=1}^{N}$, $\bar{\boldsymbol{r}}$, $\bar{z}$
Output:  $\boldsymbol{u}_i$ ($i^{th}$ agent input)
$K \leftarrow K(\bar{\boldsymbol{r}}, \bar{\boldsymbol{r}})$                 ▷ As in (3)
$K^* \leftarrow K(\boldsymbol{r}_i, \bar{\boldsymbol{r}})$                ▷ As in (3)
$\hat{z}_i \leftarrow (K^*)^\intercal K^{-1} \bar{z}$          ▷ Predict the field value
$z_i \leftarrow z(\boldsymbol{r}_i)$           ▷ Sample the true field value

if $t - m\Delta t > \Delta t$ then
    $\bar{z} \leftarrow extend(\bar{z}, \{z_j\}_{j=1}^{N})$  ▷ Extend the GP dataset
    $\bar{\boldsymbol{r}} \leftarrow extend(\bar{\boldsymbol{r}}, \{\boldsymbol{r}_j\}_{j=1}^{N})$   ▷ Extend the GP dataset
end

$\boldsymbol{n} \leftarrow \mathrm{PCA}(\{\boldsymbol{r}_i - \boldsymbol{r}_j\})$ ▷ Compute the SUSD direction
$u_i \leftarrow k_1 \boldsymbol{n}(\frac{1}{(z_i - \hat{z}_i)^2 + \varepsilon} + \eta)$         ▷ Compute Input
```

V. Dynamics and Convergence Analysis

In this section, we derive the SUSD exploration dynamics for the team of agents using (6). We show the derivations for the SUSD direction and show that the SUSD direction is attracted to the gradient of the GP prediction error.

A. SUSD Exploration Dynamics

Let f be the scalar field at some learning iteration m, f_i be a shorthand for $f_i = f(\boldsymbol{r}_i)$, and f_c be the shorthand for the transformation at the center position $f_c = f(\boldsymbol{r}_c)$. Assuming f is real analytic and using Taylor expansion, we may write:

$$f_i - f_c = \langle \boldsymbol{r}_i - \boldsymbol{r}_c, \nabla f \rangle + \nu_i, \tag{8}$$

where $\nu_i = \mathscr{O}(||\boldsymbol{r}_i - \boldsymbol{r}_c||)$ is the remaining higher-order components of the function. Then, using (6), we can show that the dynamics of direction $\boldsymbol{n}$ are described by [15]

$$\dot{\boldsymbol{n}} = -\frac{\lambda_q}{\lambda_q - \lambda_n}(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^\intercal)\nabla f + \boldsymbol{\nu}, \tag{9}$$

$$\dot{\boldsymbol{N}} = \frac{f_a}{||\nabla f||}(\boldsymbol{I} - \boldsymbol{N}\boldsymbol{N}^\intercal)\boldsymbol{H}\boldsymbol{n}, \tag{10}$$

where λ_q and λ_n are the larger and smaller eigenvalues of $\boldsymbol{C}$, $\boldsymbol{\nu} = -(1/(\lambda_q - \lambda_n))\sum_i \nu_i \langle \boldsymbol{r}_i - \boldsymbol{r}_c, \boldsymbol{q}\rangle \boldsymbol{q}$ is due to the higher-order terms of the function, $f_a = (1/M)\sum_i f(\boldsymbol{r}_i)$ is the average measurement, $\boldsymbol{H} = \nabla^2 f$ is the hessian matrix, and $\boldsymbol{N} = \frac{\nabla f}{||\nabla f||}$ is the normalized gradient direction. From now and on, for any variable l, we write $l_c = l(\boldsymbol{r}_c)$ to denote the value of l at position $\boldsymbol{r}_c$. We now present the following result for the field transformation (5).

Lemma 5.1: For the function map defined defined in (5), the SUSD dynamics is described by

$$\dot{\boldsymbol{n}} = \frac{-2k_1\lambda_q}{\lambda_q - \lambda_n}|z_c - \hat{z}_c|(f_c - \eta)^2 ||\nabla z_c - \nabla \hat{z}_c||(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^\intercal)\boldsymbol{N} + \boldsymbol{\nu}, \tag{11}$$

$$\dot{\boldsymbol{N}} = \frac{\mathrm{sign}(\hat{z}_c - z_c) f_a}{||\nabla z_c - \nabla \hat{z}_c||}(\boldsymbol{I} - \boldsymbol{N}\boldsymbol{N}^\intercal)(\boldsymbol{H} - \hat{\boldsymbol{H}})\boldsymbol{n}, \tag{12}$$

where $\boldsymbol{\nu}$ and f_a are as defined in (9), and $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ are the hessian matrices $\nabla^2 z_c$ and $\nabla^2 \hat{z}_c$, respectively.

Proof: To prove (11), using (5), we first derive

$$\nabla f_c = -2(f_c-\eta)^2(z_c-\hat{z}_c)(\nabla z_c - \nabla \hat{z}_c), \tag{13}$$

$$\|\nabla f_c\| = 2(f_c-\eta)^2|z_c-\hat{z}_c|\|\nabla z_c - \nabla \hat{z}_c\|, \tag{14}$$

$$\boldsymbol{N} = \frac{\nabla f_c}{\|\nabla f_c\|} = \operatorname{sign}(\hat{z}_c - z_c)\frac{\nabla z_c - \nabla \hat{z}_c}{\|\nabla z_c - \nabla \hat{z}_c\|}. \tag{15}$$

Substituting (13) into (9), and using (14) and (15), yields the desired result (11). To prove (12), we first obtain

$$\dot{\boldsymbol{N}} = (\frac{\boldsymbol{I}}{\|\nabla f_c\|} - \frac{\nabla f_c \nabla f_c^{\mathsf{T}}}{\|\nabla f_c\|^3})\frac{d}{dt}\nabla f_c = f_a(\boldsymbol{I} - \boldsymbol{N}\boldsymbol{N}^{\mathsf{T}})\frac{\nabla^2 f_c}{\|\nabla f_c\|}\boldsymbol{n} \tag{16}$$

Then we derive

$$\nabla^2 f_c = a(\nabla z_c - \nabla \hat{z}_c)(\nabla z_c - \nabla \hat{z}_c)^{\mathsf{T}} + b(\nabla^2 z_c - \nabla^2 \hat{z}_c), \tag{17}$$

where $a = 2(f_c-\eta)^2[4(f_c-\eta)(z_c-\hat{z}_c)^2 - 1]$ and $b = 2(f_c-\eta)^2(z_c-\hat{z}_c)$. Substituting (17) in (16), and using $\|\nabla f_c\| = 2(f_c-\eta)^2|z_c-\hat{z}_c|\|\nabla z_c - \nabla \hat{z}_c\|$ along with the fact that $(\boldsymbol{I} - \boldsymbol{N}\boldsymbol{N}^{\mathsf{T}})(\nabla z_c - \nabla \hat{z}_c)(\nabla z_c - \nabla \hat{z}_c)^{\mathsf{T}}\boldsymbol{n} = \|\nabla z_c - \nabla \hat{z}_c\|^2(\boldsymbol{I} - \boldsymbol{N}\boldsymbol{N}^{\mathsf{T}})\boldsymbol{N}\boldsymbol{N}^{\mathsf{T}}\boldsymbol{n} = 0$, complete the proof. ■

Observe that the first term of (11) represents a consensus-on-a sphere control law [20] between $\boldsymbol{n}$ and $-\boldsymbol{N}$. This suggests that the SUSD direction tends to align with the negative gradient direction of the function (5). Since (5) has a maximum when the error $(z-\hat{z}) = 0$, moving along the opposite gradient direction of f implies that SUSD is moving the swarm towards areas of high modeling error in the GP.

B. Convergence Analysis

In this section, we will prove that the SUSD direction $\boldsymbol{n}$ converges to $-\boldsymbol{N}$, i.e. the negative model error gradient $\nabla z_c - \nabla \hat{z}_c$. For this, we define $\theta = 1 + \langle \boldsymbol{N}, \boldsymbol{n} \rangle$ so that $\theta = 0$ if and only if $\langle \boldsymbol{N}, \boldsymbol{n} \rangle = -1$, i.e. the desired equilibrium. Then, using (11), we obtain

$$\begin{aligned}\dot{\theta} &= h(t,\theta,\delta) \\ &= -2k_1\frac{\lambda_q}{\lambda_q-\lambda_n}|z_c-\hat{z}_c|(f_c-\eta)^2\|\nabla z_c - \nabla \hat{z}_c\|\theta(2-\theta) + \delta,\end{aligned} \tag{18}$$

where $\delta = \langle \boldsymbol{N}, \boldsymbol{\nu} \rangle + \langle \dot{\boldsymbol{N}}, \boldsymbol{n} \rangle$ is viewed as an input disturbance due to the nonlinear terms of the function. Since at each time, δ depends on where the swarm is in the search space and we cannot control it, we use an input-to-state stability analysis approach to obtain a convergence result. Note that $\dot{\theta} = 0$ when $\theta \in \{0,2\}$. Furthermore, when $\theta \in \{0,2\}$, then $\boldsymbol{n} = \pm\boldsymbol{N}$, and hence from (12) and definition of $\boldsymbol{\nu}$, the disturbance δ vanishes at the two equilibria.

Theorem 5.1: Consider the system (18). Consider the set $\mathscr{B} = \{\boldsymbol{r}_c | \|\nabla z_c - \nabla \hat{z}_c\| > \mu\}$ where $\mu > 0$ is a constant that we derive later in *Remark 3*. Then the equilibrium $\theta = 0$ of the unforced system $\dot{\theta} = h(t,\theta,0)$ is asymptotically stable. Furthermore, for an input disturbance satisfying $|\delta| < 2k_1\varepsilon_1|z_c-\hat{z}_c|(f_c-\eta)^2\mu$ for some $\varepsilon_1 \in (0,1)$, the forced system $h(t,\theta,\delta)$ is locally input-to-state stable.

Proof: Define $V = \theta/(2-\theta)$ to be Lyapunov candidate function. Note that $V = 0$ if and only if $\theta = 0$, and $V \to \infty$ as $\theta \to 2$. Consider the domain $D = \{\theta | \theta \in [0,2)\}$. Then, when $\delta = 0$, we find $\dot{V} \leq -s\|\nabla z_c - \nabla \hat{z}_c\|V$, where $s = 4k_1\frac{\lambda_q}{\lambda_q-\lambda_n}|z_c-\hat{z}_c|(f_c-\eta)^2$ is a positive definite function. Since $\dot{V} = 0$ if and only if $\theta = 0$, then the origin of the unforced system $h(t,\theta,0)$ is asymptotically stable. Additionally, since $\dot{V} \to -\infty$ as $\theta \to 2$, and the fact that $V \to \infty$ whenever $\theta \to 2$ and $||\nabla z_c - \nabla \hat{z}_c|| > \mu > 0$, implies that $\mathbf{D}$ is a forward invariant set, and thus $\theta \in [0,2)$ for all t. For the forced system $h(t,\theta,\delta)$, we obtain

$$\dot{V}_1 \leq -s(1-\varepsilon_1)||\nabla z_c - \nabla \hat{z}_c||V, \quad \forall|\theta| \geq \rho(|\delta|), \tag{19}$$

where $\rho(|\delta|) = 1 - \sqrt{1 - |\delta|/(2k_1\varepsilon_1|z_c-\hat{z}_c|(f_c-\eta)^2\mu)}$ is a class $\mathscr{K}$ function. Since it is assumed that $|\delta| < 2k_1\varepsilon_1|z_c-\hat{z}_c|(f_c-\eta)^2\mu$, then the set $\theta \in [0,\rho(|\delta|))$ is not empty. Let $\alpha_1(|\theta|) = \alpha_2(|\theta|) = \frac{|\theta|}{2-|\theta|}$ which are class $\mathscr{K}$ functions that satisfy $\alpha_1(|\theta|) \leq V(\theta) \leq \alpha_2(|\theta|)$. Therefore, using a local definition of input-to-state stability [21, 22], and according to *Theorem 4.19* in [23], the origin of the forced system $h(t,\theta,\delta)$ is locally input-to-state stable. ■

Theorem 5.1 reveals that the SUSD direction $\boldsymbol{n}$ aligns with the negative gradient direction $-\boldsymbol{N}$ of (5) whenever $\|\nabla z_c - \nabla \hat{z}_c\|$ is large enough compared to the higher-order terms of the model error $(z_c - \hat{z}_c)$. In other words, SUSD drives the swarm to the areas of high model error the fastest possible (as moving against the gradient of (5)) until the error becomes small. In this time, $\boldsymbol{n}$ might not align with $-\boldsymbol{N}$ until the swarm enters another neighborhood where $\|\nabla z_c - \nabla \hat{z}_c\| > \mu$.

Remark 3: To derive the lower bound μ, we need to have $||\nabla z_c - \nabla \hat{z}_c|| > \mu \geq \frac{|\delta|}{2k_1\varepsilon_1|z_c-\hat{z}_c|(f_c-\eta)^2}$. But, using (12), $|\delta| \leq \frac{k_1}{\lambda_q-\lambda_n}|\upsilon| + \frac{k_1 f_a}{||\nabla z_c - \nabla \hat{z}_c||}||H - \hat{H}||$, where $\upsilon = \sum_k \nu_k \langle \boldsymbol{r}_k - \boldsymbol{r}_c, \boldsymbol{q} \rangle$. Therefore we need $||\nabla z_c - \nabla \hat{z}_c|| > \frac{k_1}{\lambda_q-\lambda_n}|\upsilon| + \frac{k_1 f_a}{||\nabla z_c - \nabla \hat{z}_c||}||\boldsymbol{H} - \hat{\boldsymbol{H}}||$. Solving this inequality yields that $\mu = \frac{|\upsilon| + \sqrt{|\upsilon|^2 + 8\varepsilon_1 f_a|z_c-\hat{z}_c|(f_c-\eta)^2\lambda_q(\lambda_q-\lambda_n)||\boldsymbol{H}-\hat{\boldsymbol{H}}||}}{4\varepsilon_1|z_c-\hat{z}_c|(f_c-\eta)^2\lambda_q}$. Clearly this bound depends on the higher-order terms υ and $\boldsymbol{H} - \hat{\boldsymbol{H}}$, and the inter-agent distances captured by $\lambda_q - \lambda_n$. In general, we can say μ is small when the gradient is large, or when the GP error is large. As such, the swarm ignores well-modeled regions and searches towards areas with large error.

VI. Simulations and Experiments

In this section, we demonstrate simulations and experiments to validate our proposed strategy. We show simulations to demonstrate the approach over large spaces, and experiments with robot blimps and the Robotarium to demonstrate the approach's applicability with a variety of real robots. Videos of the experiments are also provided. For each experiment, the scalar field is generated according to a sum of Gaussian functions $z(\boldsymbol{r}) = \sum_{l=1}\gamma_l \exp(-\|\boldsymbol{r} - \boldsymbol{r}_{s,l}\|/\delta_l)$. In addition, in order to ensure agents stay close together and within the bounded domain, we modify (6) with a formation controller $\boldsymbol{u}_{i,f} = \sum_{j\neq i}(d_0 - d_{ij})(\boldsymbol{r}_i - \boldsymbol{r}_j))$ where d_0 is a desired separation distance and $d_{ij} = \|\boldsymbol{r}_i - \boldsymbol{r}_j\|$, and a boundary breaching function $z_{\mathscr{D}} : \mathbb{R}^2 \to \mathbb{R}$ which is positive and increasing when outside of the domain. By ensuring the speed of agents increases when outside of the domain, SUSD naturally re-directs agents back into the domain. The modified algorithm for our simulations and experiments is

$$\boldsymbol{u}_i(t) = \boldsymbol{n}(t)\left[k_1 f^{(m)}(\boldsymbol{r}_i(t)) + z_{\mathscr{D}}(\boldsymbol{r}_i(t))\right] + \boldsymbol{u}_{i,f}(t). \tag{20}$$

A. Simulations

For a team of 3 agents, we test a simulation where the parameters of the field, parameters of the GP kernel, breaching function, and domain are given by

$$\begin{aligned}
\gamma &= \begin{bmatrix} -30 & -30 & -30 & -30 & -30 \end{bmatrix}^{\mathsf{T}} \\
\delta &= \begin{bmatrix} 16 & 22 & 50 & 18 & 22 \end{bmatrix}^{\mathsf{T}} \\
\boldsymbol{r}_s &= \begin{bmatrix} -32 & 25 & -5 & 30 & 40 \\ 40 & -35 & -10 & 35 & 40 \end{bmatrix} \\
\alpha &= 7, \quad \beta = 5.5 \\
z_{\mathscr{D}}(\boldsymbol{r}) &= \begin{cases} 0.002\|\boldsymbol{r}_{\mathscr{D}} - \boldsymbol{r}\|, & \text{if } \boldsymbol{r} \notin \mathscr{D} \\ 0, & \text{otherwise} \end{cases} \\
\mathscr{D} &= \{\boldsymbol{r} : -50 \le r_1 \le 50, -50 \le r_2 \le 50\}
\end{aligned} \tag{21}$$

where $\boldsymbol{r}_{\mathscr{D}}$ is the closest point to $\boldsymbol{r}$ on the boundary of the set $\mathscr{D}$. Fig. 2 shows the trajectory of the team overlaid on the squared error of the GP model throughout the simulation. Note that over the course of the simulation, agents seek out regions of highest model error as evidenced by the reduction of the major hot-spots first in the error field. By employing the SUSD strategy, agents align their search with the gradient of the error field.

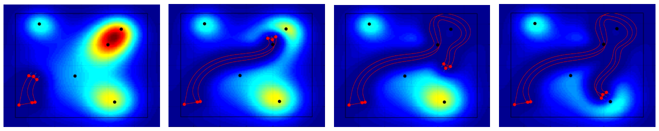

Fig. 2. Heat-map of $(z - \hat{z})^2$ with 3 agents over the course of the simulation. Position of agents are shown as red dots, black dots indicate $\boldsymbol{r}_s$ locations, and the black border shows the domain of interest. Agents start in the bottom left corner.

B. Experiments using the GT-MAB

We employ two Georgia Tech Miniature Autonomous Blimp (GT-MAB) [17, 24] to validate our approach. Due to the holonomic configuration of the platform (shown in Fig. 3), the proposed velocity controller may be used without any mapping of the single integrator dynamics. We employ our technique by constraining the autonomous blimps to fly in a 2D plane. While our approach is applicable in 3D settings provided at minimum 3 agents (needed for PCA), due to space limitations and we constrain our blimps to fly in a 2D plane. The parameters of the field, parameters of the GP kernel, the breaching function, and the domain are given by

$$\begin{aligned}
&\gamma = -55, \delta = \frac{1}{3}, \boldsymbol{r}_s = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathsf{T}}, \alpha = 20, \beta = 0.4 \\
&z_{\mathscr{D}}(\boldsymbol{r}) = \begin{cases} 0.01\|\boldsymbol{r}_{\mathscr{D}} - \boldsymbol{r}\|, & \text{if } \boldsymbol{r} \notin \mathscr{D} \\ 0, & \text{otherwise} \end{cases} \\
&\mathscr{D} = \{\boldsymbol{r} : -0.7 \le r_1 \le 0.7, -1.1 \le r_2 \le 1.1\}
\end{aligned} \tag{22}$$

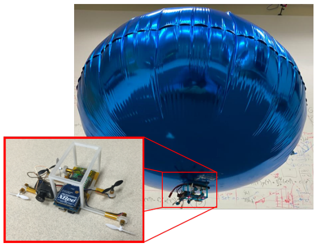

Fig. 3. The GT-MAB and GT-MAB gondola that houses the electrical payload of the autonomous blimp. Motors are arranged in a perpendicular fashion in order to achieve holonomic flight.

Fig. 4 shows the trajectory of the team overlaid on the squared error field of the GP model as the autonomous blimps explore the field. Note that the agents are able to successfully return to the center of the Gaussian function as the location of largest GP modeling error.

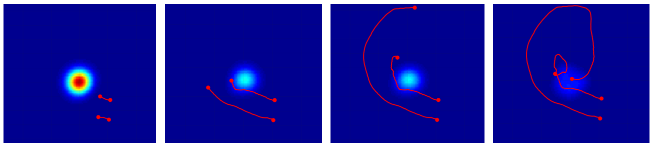

Fig. 4. Trajectory snapshots of the GT-MAB experiment overlaid on the squared-error field $(z - \hat{z})^2$. As agents learn the field, the squared-error field reduces to 0.

C. Experiments with the Robotarium

We also demonstrate our approach using the Georgia Tech Robotarium with more agents and a more complex field. The parameters of the field, parameters of the GP kernel, the breaching function, and the domain are given by

$$\begin{aligned}
&\gamma = \begin{bmatrix} -25 & -24 & -30 \end{bmatrix}^{\mathsf{T}}, \delta = \begin{bmatrix} 0.45 & 0.35 & 0.4 \end{bmatrix}^{\mathsf{T}} \\
&\boldsymbol{r}_s = \begin{bmatrix} -0.1 & 0.8 & -0.8 \\ 0.3 & -0.4 & -0.4 \end{bmatrix}, \alpha = 7.5, \beta = 0.1 \\
&z_{\mathscr{D}}(\boldsymbol{r}) = \frac{1}{\|\boldsymbol{r}_{\mathscr{D}} - \boldsymbol{r}\|^2} \\
&\mathscr{D} = \{\boldsymbol{r} : -1.6 \le r_1 \le 1.6, -1 \le r_2 \le 1\}
\end{aligned} \tag{23}$$

Fig. 5 shows the final learned GP model compared to the true field. Observe that the four agents successfully find all three sources and generate a good approximation of the field.

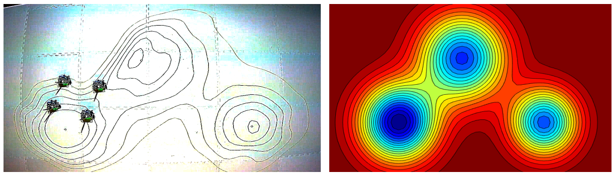

Fig. 5. Left: contour plot of final field estimate $\hat{z}^{(m)}$ from the Robotarium experiment. Right: contour plot of the true field.

VII. Conclusion

In this paper, we introduced a novel exploration method for scalar field mapping that leverages our previous work in distributed source seeking. Our approach does not require computing a joint information gain and instead leverages the SUSD algorithm to perform a derivative-free search. We prove that by using the SUSD algorithm our approach is attracted to the gradient of the modeling error which is then eliminated through the GP learning. We also provide experiments to demonstrate our approach's applicability with real robots. In future work, we will investigate how the learning behavior of the field estimator affects the search dynamics and a communication-free variant of our proposed algorithm.

ACKNOWLEDGEMENTS

T. Lin, S. Al-Abri, and F. Zhang were supported by ONR grants N00014-19-1-2556 and N00014-19-1-2266; NSF grants OCE-1559475, CNS-1828678, and S&AS-1849228; NRL grants N00173-17-1-G001 and N00173-19-P-1412 ; and NOAA grant NA16NOS0120028. T. Lin and S. Coogan were partially supported by the NSF under grant ECCS-1836932.

REFERENCES

[1] Hung Manh La and Weihua Sheng. "Distributed sensor fusion for scalar field mapping using mobile sensor networks". In: *IEEE Transactions on cybernetics* 43.2 (2013), pp. 766–778.

[2] Shuai Li, Yi Guo, and Brian Bingham. "Multi-robot cooperative control for monitoring and tracking dynamic plumes". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 67–73.

[3] Hung M La, Weihua Sheng, and Jiming Chen. "Cooperative and active sensing in mobile sensor networks for scalar field mapping". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45.1 (2014), pp. 1–12.

[4] Mohammad Rahimi et al. "Adaptive sampling for environmental robotics". In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*. Vol. 4. IEEE. 2004, pp. 3537–3544.

[5] Alan R Washburn. *Search and detection*. Institute for Operations Research and the Management Sciences, 2002.

[6] Kian Hsiang Low et al. "Adaptive sampling for multi-robot wide-area exploration". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE. 2007, pp. 755–760.

[7] Kian Hsiang Low, John M Dolan, and Pradeep Khosla. "Adaptive multi-robot wide-area exploration and mapping". In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. 2008, pp. 23–30.

[8] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[9] Maani Ghaffari Jadidi et al. "Exploration on continuous Gaussian process frontier maps". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 6077–6082.

[10] Kwangjin Yang, Seng Keat Gan, and Salah Sukkarieh. "A Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV". In: *Advanced Robotics* 27.6 (2013), pp. 431–443.

[11] Ruofei Ouyang et al. "Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena". In: (2014).

[12] Lara Brinón-Arranz and Luca Schenato. "Consensus-based source-seeking with a circular formation of agents". In: *2013 European Control Conference (ECC)*. IEEE. 2013, pp. 2831–2836.

[13] Wencen Wu, Iain D Couzin, and Fumin Zhang. "Bio-inspired source seeking with no explicit gradient estimation". In: *IFAC Proceedings Volumes* 45.26 (2012), pp. 240–245.

[14] W. Wu and F. Zhang. "A Speeding-up and Slowing-down Strategy for Distributed Source Seeking with Robustness Analysis". In: *IEEE Transactions on Control of Network Systems* 3.3 (2016), pp. 231–240.

[15] Said Al-Abri, Wencen Wu, and Fumin Zhang. "A Gradient-Free Three-Dimensional Source Seeking Strategy With Robustness Analysis". In: *IEEE Transactions on Automatic Control* 64.8 (2018), pp. 3439–3446.

[16] S. Al-Abri, S. Maxon, and F. Zhang. "Integrating a PCA learning algorithm with the SUSD strategy for a collective source seeking behavior". In: *the Proceedings of the American Control Conference*. 2018, pp. 2479–2484.

[17] Ningshi Yao et al. "Monocular vision-based human following on miniature robotic blimp". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3244–3249.

[18] Daniel Pickem et al. "The robotarium: A remotely accessible swarm robotics research testbed". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1699–1706.

[19] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal component analysis". In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[20] J. Markdahl, J. Thunberg, and J. Gonçalves. "Almost Global Consensus on the n-Sphere". In: *IEEE Transactions on Automatic Control* 63.6 (June 2018), pp. 1664–1675.

[21] E. D. Sontag and Yuan Wang. "New characterizations of input-to-state stability". In: *IEEE Transactions on Automatic Control* 41.9 (Sept. 1996), pp. 1283–1294.

[22] Andrii Mironchenko. "Local input-to-state stability: Characterizations and counterexamples". In: *Systems & Control Letters* 87 (2016), pp. 23–28.

[23] H. K. Khalil. *Nonlinear Systems, Third Edition*. Prentice Hall, 2002.

[24] Sungjin Cho et al. "Autopilot design for a class of miniature autonomous blimps". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2017, pp. 841–846.