An Optimal Control Approach to Mapping GPS-Denied Environments Using a Stochastic Robotic Swarm

Ragesh K. Ramachandran, Karthik Elamvazhuthi and Spring Berman

1 Introduction

In recent years, there has been an increasing focus on the development of robot platforms that can be deployed in *swarms* to perform tasks autonomously over large spatial and temporal scales. In addition, swarms of nanoscale structures and devices such as nanoparticles, molecular machines, and magnetic nanocarriers are being developed for biomedical applications such as imaging and targeted drug delivery [21]. Many potential applications for robotic swarms, including exploration, environmental monitoring, disaster response, search-and-rescue, mining, and intelligence-surveillance-reconnaissance, will require the robots to operate in dynamic, uncertain environments. Moreover, the robots' highly restricted onboard power may preclude the use of GPS and communication devices, or the robots may be located in GPS-denied environments where communication is impractical or unreliable. Despite these limitations, it may still be necessary for the swarm to characterize its surroundings, for instance to map obstacles, target payloads, or hazardous areas to avoid. Nanoscale swarms, which will have extremely limited capabilities, may be used to map cellular structures inside the human body.

To address these challenges, we present a method for *mapping a feature of interest in an unknown environment* using a swarm of robots with local sensing capabilities, no localization, and no inter-robot communication. We consider scenarios where the robots exhibit significant randomness in their motion due to sensor and actuator noise

R.K. Ramachandran (✉) · K. Elamvazhuthi · S. Berman
School for Engineering of Matter, Transport and Energy, Arizona State University,
Tempe, AZ, USA
e-mail: rageshkr@asu.edu

K. Elamvazhuthi
e-mail: karthikevaz@asu.edu

S. Berman
e-mail: Spring.Berman@asu.edu

A. Bicchi and W. Burgard (eds.), *Robotics Research*, Springer Proceedings
in Advanced Robotics 2, DOI 10.1007/978-3-319-51532-8_29

or, at the nanoscale, the effects of Brownian motion and chemical interactions. Our mapping approach is scalable with the number of robots, so that arbitrary swarm populations can be used.

Our method relies on developing a continuous abstraction of the swarm population dynamics in the form of an advection-diffusion-reaction PDE model, which we call the *macroscopic model*. This model describes the spatial and temporal evolution of the population densities of robots in different states throughout the domain. To represent individual robots, we define a *microscopic model* that describes how each robot moves and responds upon encountering a feature of interest. The state transition of a robot is modeled as a irreversible chemical reaction with a high reaction rate. The macroscopic model becomes a more accurate model of the microscopic model as the number of robots increases.

We pose our mapping problem as the computation of a spatially varying function that represents the map of the feature of interest. To estimate this function, we use temporal data that is recorded by the robots during their exploration of the environment. This data yields the time evolution of the number of robots that are still exploring the domain; i.e., robots that have not encountered the feature. In practice, this data could be collected from the robots after their deployment by retrieving their recorded times of encounter with the feature. In biomedical imaging applications with nanoscale swarms, this data could be obtained from a measurable signal that corresponds to the density of the population that is still in the exploring state.

Once this data is obtained, we use techniques from optimal control to compute the function that represents the feature map. In general, optimal control entails the minimization or maximization of an objective functional that is defined in a finite-dimensional space and is subject to a set of ordinary or partial differential *constraint equations*, which govern the system of interest. From a computational perspective, optimal control methods are more effective than black box techniques, such as genetic algorithms and particle swarm optimization, in terms of the number of objective functional evaluations per cycle. This computational advantage mainly arises from their use of the problem structure to calculate the gradient of the control-to-state maps using the adjoint equation. The feature map is defined as the solution of an optimization problem that minimizes an objective functional which is based on the robot data. This optimization problem is solved numerically offline using standard techniques such as gradient descent algorithms. We validate our approach in simulation for features of varying shape, size, orientation, and location.

1.1 Related Work

In the literature, there have been exhaustive studies on mapping and exploring an environment using robots. SLAM (simultaneous localization and mapping) [16, 18], probabilistic mapping [3, 19], and topological and metric map building [15, 20] are some of the techniques that have been developed for environmental mapping by robots. These techniques have been used for path planning and mapping in small multi-robot groups. However, the problem of scaling these approaches to larger groups becomes intractable for swarms of hundreds or thousands of robots, due to their limitations on communication bandwidth and their spatially distributed nature. In addition, these techniques require the robots to have sophisticated sensing and processing capabilities, which are not feasible in swarm robotic platforms.

Mapping an environment using a robotic swarm is a relatively new area of research in the robotics community. An approach to this problem is given in [6, 7], in which a robotic swarm is used to identify the topological features of an environment from information about the times at which robots encounter other robots and environmental features. This work borrows tools from algebraic geometry and topological data analysis to compute a metric that can be used to classify the topological structure of the environment. The approach requires some minimal inter-robot communication, unlike our strategy which is communication-free.

Our mapping approach uses methods from [9], a stochastic task allocation approach that achieves target spatial distributions of robot activity without using communication or localization. Also, our approach is inspired by [13], a method for reconstructing environmental features from minimal robot data using compressed sensing techniques. In contrast to the scenarios that we consider, the robots in [9, 13] can move over the features to be mapped, which allows the mapping problem to be formulated as the inversion of a linear operator. Approaches with a similar mathematical framework for parameter estimation have been used extensively in the area of biomedical imaging, especially with MRI and CT scan images. In these approaches, the system is excited with a stimulus such as a magnetic field, X-rays, or ultrasound, and the system response is used to identify and estimate a spatially-dependent parameter that corresponds to the image [1, 17, 23].

2 Problem Statement

We consider a scenario in which N robots are deployed into an unknown, bounded environment to map a single feature of interest. We exclude cases in which the feature is located very close to the domain boundary, since robot collisions with this boundary and the high diffusion of swarms that start far from the feature will degrade the estimation. If a robot encounters the feature, it stops moving and records the time at which it stopped. Using data on the number of robots that are still moving at each instant, we aim to estimate the position and geometry of the encountered feature. We can improve the accuracy of this estimate by deploying the swarm in different directions from various locations, which will ensure greater coverage of the domain and result in robot collisions with a larger portion of the feature boundary. This approach may be used to map multiple sparsely distributed features by reconstructing each individual feature from its corresponding data set and computing the entire map as a linear combination of single-feature maps.

Robot capabilities: The robots are assumed have sufficient power to complete the mapping operation. The power requirement for the robots is low, since they are not equipped with communication devices or GPS. The robots have local sensing capabilities and can identify the feature at distances within their sensing range. We may also assume that the robots can detect other robots within their sensing range and perform collision avoidance maneuvers, although we do not simulate collision avoidance in this work. Each robot is equipped with a compass and thus can move in a specified heading. Additionally, the robots have sufficient memory to store the time of their encounter with the feature.

Robot controller: The robots begin at a specified location in the domain. During a swarm deployment, the robots move with a predetermined time-dependent velocity, $\mathbf{v}(t) \in \mathbb{R}^2$. This velocity is designed to guide the center of mass of the swarm along a desired trajectory through the environment. The velocity field may be initially transmitted to the robots by a computer at their starting location, or the robots may be directed according to the field using external stimuli such as magnetic fields or radiation. The robots' motion is affected appreciably by sensor and actuator noise, due to lack of feedback. If a robot detects a feature within its sensing range, it stops moving and records the time. At a predefined time t_f, the stationary robots around the feature boundary return to the starting point of the deployment and upload their encounter times to a computer. The computer then applies the optimal control method described in Sect. 4 to estimate the map of the feature using this robot data.

3 Models of the Mapping Scenario

3.1 Microscopic Model

This model is used to simulate a robot's motion and its response to an encounter with a feature in its path. The change in a robot's state that is triggered by an encounter is modeled as an irreversible chemical reaction,

$$A \xrightarrow{k} P, \tag{1}$$

where the species A represents an *active* (moving) robot, P represents a *passive* (stationary) robot, and k is the reaction rate constant, which in this case is a fixed probability per unit time. This constant is assigned a high value to enforce a high probability of transitioning from active to passive.

We model the robots as point masses with negligible size compared to the area of the domain. A particular robot i has position $\mathbf{X}_i(t) = [x_i(t)\ y_i(t)]^T$ at time t. The deterministic motion of the robot is directed by the time-dependent velocity field $\mathbf{v}(t) = [v_x(t)\ v_y(t)]^T$. The noise in the robot movement is modeled as a Brownian motion that drives diffusion with an associated diffusion coefficient D. We assume that the robots' navigation error can be modeled as diffusive noise and that the value of D can be estimated. The displacement of robot i over a time step Δt is given by the standard-form Langevin equation [11]:

$$\mathbf{X}_i(t + \Delta t) = \mathbf{X}_i(t) + (\sqrt{2D\Delta t})\mathbf{Z}(t) + \mathbf{v}(t)\Delta t, \tag{2}$$

where $\mathbf{Z}(t) \in \mathbb{R}^2$ is a vector of independent standard normal random variables that are generated at time t. The robots avoid collisions with the domain boundary by performing a specular reflection when they encounter this boundary.

3.2 Macroscopic Model

The macroscopic model governs the time evolution of the expected spatial distribution of the robotic swarm. For a swarm whose members move according to Eq. (2), the macroscopic model is given by an advection-diffusion PDE, as described in [5]. Since our microscopic model includes robot state changes that can be represented as chemical reactions, our macroscopic model takes the form of an advection-diffusion-reaction (ADR) PDE. The model is defined over a domain $\Omega \subset \mathbb{R}^2$ with Lipschitz continuous boundary $\partial\Omega$ and over a time interval T. We define $L = \Omega \times [0, T]$ and $\Gamma = \partial\Omega \times [0, T]$. The state of the macroscopic model is the population density field $u(\mathbf{x}, t)$ of active robots in the domain at points $\mathbf{x} \in \Omega$ and times $t \in T$. We specify a spatially varying *indicator function*, $K(\mathbf{x}) : \Omega \rightarrow \{0, 1\}$, that equals 0 at points $\mathbf{x}$ where the feature of interest is absent and equals 1 at points where it is present. The reaction term of the macroscopic model is determined by the rate constant k in Eq. (1), which is switched on or off by the indicator function $K(\mathbf{x})$ depending on whether the feature of interest occupies point $\mathbf{x}$. This term models the switching of individual robots from the active state to the passive state when they are in the vicinity of the feature. The advection term of the macroscopic model is governed by the velocity field $\mathbf{v}(t)$ that is defined in the microscopic model.

From the above definition, the macroscopic model is given by:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D\nabla u - \mathbf{v}(t)u) - kK(\mathbf{x})u \quad in \;\; L \tag{3}$$

with the no-flux boundary condition

$$\mathbf{n} \cdot (D\nabla u - \mathbf{v}(t)u) = 0 \quad on \;\; \Gamma, \tag{4}$$

where $\mathbf{n} \in \mathbb{R}^2$ is the outward normal of the boundary $\partial\Omega$. We specify that all robots start in the active state and set the initial condition,

$$u(\mathbf{x}, 0) = u_0, \tag{5}$$

to a Gaussian density centered at a point $\mathbf{x}_0$, which we assume is far from the feature. The macroscopic model is numerically solved using the explicit finite-volume method that is described in [9].

Our approach relies on the close correspondence of the macroscopic model solution to the average swarm density over an ensemble of microscopic model simulations. Therefore, the approach is robust to robot malfunctions and external disturbances as long as these factors do not significantly affect the model correspondence. This implies that the number of failed robots should be small compared to the total swarm size, and that the robots' trajectory drift due to wind, currents, and other environmental influences should be small relative to their modeled motion. In scenarios that violate these conditions, it would be necessary to improve the accuracy of the macroscopic model by estimating the components of $\mathbf{v}$, D, and k that are affected by unmodeled dynamics and disturbances. This is a topic of future work.

4 Optimal Control Approach to Mapping Features

The feature reconstruction problem is framed as an optimal control problem. A gradient descent algorithm is used to compute the optimal control for the problem. An adjoint state equation approach is used to compute the gradient required for the algorithm [4]. The key advantage of this approach is that it derives an explicit formula for the gradient of the objective functional with respect to the control, subject to the constraints. The Hamiltonian and Pontryagin maximum principle can be to used to derive the adjoint equation for finite-dimensional systems. However, in the case of infinite-dimensional systems, the existence of the Hamiltonian has been proven only for a limited class of systems [10]. This motivated us to derive the directional directive of the control-to-state mapping and use the generalized chain rule of differentiation of composite mappings in Banach spaces, as is found in the literature [2, 22]. In order to make the derivatives of certain maps well-defined, an appropriate choice of spaces is made for the parameters and the solutions satisfying the system of differential equations. We present a Lagrangian-based analysis of these derivatives in the Appendices. The proof for the existence of optimal control for the problem is the same as the one shown in [8].

The optimization procedure uses data on the ratio of the number of active robots at each instant of time to the initial number of active robots at the start of the swarm deployment. To ensure sufficient coverage of the domain, the swarm can be deployed from multiple starting positions and directed along different trajectories. Once this data is obtained, the optimization procedure is performed to find the feature map that would produce data that is similar to the data obtained from the deployments. The computational cost increases greatly with the number of data sets (one from each deployment) that are used for optimization, since the number of PDEs to be solved per iteration varies linearly with the data sets. However, we can obtain a better estimate of the feature map with more data. Hence, there is a tradeoff between the computational cost of the optimization and the accuracy of the estimate. In order to resolve this issue, we discard data sets from deployments in which few robots undergo a state transition compared to the other deployments. A paucity of state transitions indicates that the swarm trajectories infrequently intersect the feature. In addition, our procedure can be easily parallelized since the most computationally intensive part is the solution of the PDEs.

The optimal control problem is formulated as follows. Each of the i swarm deployments yields a sequence of times at which active robots encounter the feature and switch to the passive state. From this data, we can determine the fraction $g_i(t) \in L^2([0,T])$ of active robots in the swarm at each time t during deployment i. The solution $u_i(\mathbf{x},t)$ of the corresponding macroscopic model Eqs. (3)–(5) can be used to compute the integral $\int_\Omega u_i(\mathbf{x},t)d\mathbf{x}$, the expected fraction of active robots in the domain at time t. We assume that the swarm size is sufficiently large for $g_i(t)$ to closely match this integral if the feature map, represented by the function $K(\mathbf{x})$ in Eq. (3), is known. Therefore, we can frame our optimization objective as the computation of the input $K(\mathbf{x})$ that minimizes the function

$$J_i(u_i) = \frac{1}{2}\left\| \int_\Omega u_i(\mathbf{x},t)d\mathbf{x} - g_i(t) \right\|^2_{L^2([0,T])}. \tag{6}$$

Suppose that the data from N deployments are selected to compute the optimal controls. The swarm velocity and initial distribution for deployment i are given by $\mathbf{v}_i(t)$ and u_0^i, respectively. The macroscopic model with these parameters is considered to be the i^{th} set of constraints, which we denote by $\Psi_i(u_i, K)$ as in [22]. The solution to this model is given by u_i, and the set of solutions for all N deployments is $\mathbf{u} := \{u_1, u_2, \ldots, u_i, \ldots, u_N\}$. We define the space of macroscopic model solutions as $U = C([0,T]; L^2(\Omega))$ and the space of admissible input functions as $\Theta_{ad} = \{K(\mathbf{x}) \in L^2(\Omega);\ K_{min} \le K(\mathbf{x}) \le K_{max}\}$. Furthermore, W_i is a weight that quantifies the significance of the data from deployment i relative to data from the other deployments, and λ is the Tikhonov regularization parameter [14]. Using these definitions, we can frame the optimal control problem as:

$$\min_{(\mathbf{u},K(\mathbf{x}))\in U^N\times\Theta_{ad}} \mathbf{J}(\mathbf{u},K) = \sum_{i=1}^{N} W_i J_i(u_i) + \frac{\lambda}{2}\|K(\mathbf{x})\|^2_{L^2(\Omega)}, \tag{7}$$

subject to the constraints $\Psi_i(u_i, K)$, $i = 1, \ldots, N$.

We must compute the gradient of the objective functional $\mathbf{J}(\mathbf{u}, K)$ with respect to the control inputs in order to perform the gradient descent algorithm for minimizing this functional. We introduce the Lagrangian functional $\mathscr{L}$ and Lagrangian multipliers p_i, with $\mathbf{p} := \{p_1, p_2, \ldots, p_i, \ldots, p_N\}$:

$$\mathscr{L}(\mathbf{u},\mathbf{p},K) = \mathbf{J}(\mathbf{u},K) + \sum_{i=1}^{N}\langle p_i, \Psi_i(u_i,K)\rangle. \tag{8}$$

The functions p_i, also known as the adjoint variables, express the sensitivity of the objective functional to variations in the input control variable $K(\mathbf{x})$. The necessary condition for optimality is $\nabla \mathscr{L} = 0$, which implies the following three conditions: (1) $\nabla_{\mathbf{u}} \mathscr{L} = 0$, the adjoint equation; (2) $\nabla_{\mathbf{p}} \mathscr{L} = 0$, the state equation in weak form; and (3) $\nabla_K \mathscr{L} = 0$, the optimal control constraint. These three equations are used to compute the gradient of $\mathbf{J}(\mathbf{u}, K)$. The derivation of the adjoint and gradient equations is described in the Appendices.

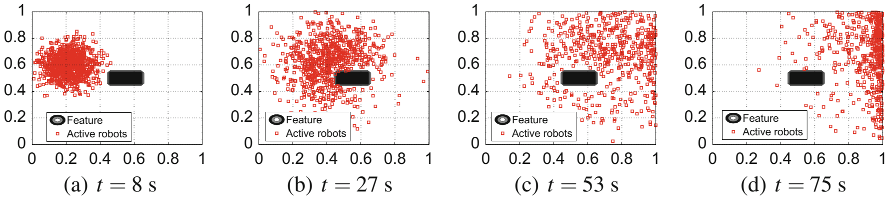
The solution to an optimization problem that is obtained by a gradient descent algorithm is sensitive to the choice of the initial guess and may be a local minimum of the objective functional rather than the global minimum. To increase the likelihood of obtaining the global minimum, we choose an initial guess for the feature map, represented by $K(\mathbf{x})$, that is guaranteed to include the actual map. This initial guess is that the feature covers the entire area traversed by the swarm during each of its i deployments (in actuality, the feature will occupy a subset of this area). Formally, we define $\gamma_i := [0, 1] \rightarrow \mathbb{R}^2$ as the trajectory of the swarm center during the i^{th} deployment and $B_2(\gamma_i(\tau), \delta)$ as a ball with radius δ centered at the point $\gamma_i(\tau)$, and we initially set $K(\mathbf{x}) = 1$ for all $\mathbf{x} \in \left(\cup_{i=1}^{N} B_2(\gamma_i(\tau), \delta)\right) \cap \Omega$, $\tau \in [0, 1]$. We choose δ to be 3 times the standard deviation of the initial Gaussian swarm distribution.

5 Simulated Mapping Scenarios

We developed microscopic and macroscopic models of a robotic swarm for six mapping scenarios, each with a single feature in the domain. The six features varied in position, size, shape, and orientation. We applied the method described in Sect. 4 to reconstruct each feature from the simulated robot data on feature encounter times. For each simulation, we used a swarm of 1000 robots in a normalized domain of size 1 m × 1 m. The value of k was chosen to be $1/dt$, where dt is the time step of the microscopic model, in order to ensure that robots always switched to the passive state when they encountered the feature boundary. For simplicity, the designated velocity fields $\mathbf{v}_i(t)$ of the robots were each assigned a constant heading. The robots moved at a speed of 0.012 m/s with a diffusion coefficient of $D = 5 \times 10^{-4}$ m^2/s, and each simulation ran for 80 s. The microscopic model was simulated in a 26 × 26 grid, while the macroscopic model was solved in a finer grid of 51 × 51 grid cells to account for numerical diffusion. In the optimization procedure, $K(\mathbf{x})$ was bounded between $K_{min} = 0$ and $K_{max} = 1$.

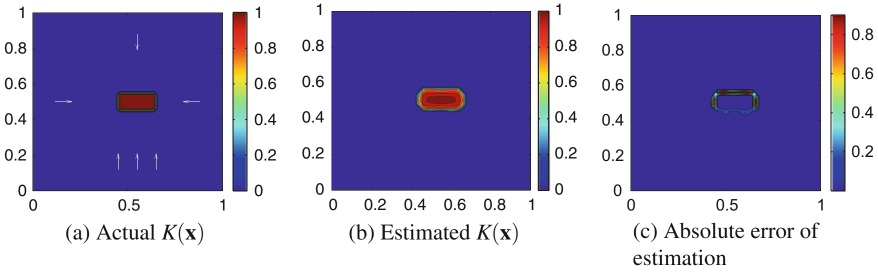
Figure 1 shows snapshots of the active robots in a swarm at various times t during a sample deployment. The robots behave according to the microscopic model and move through a domain that contains a rectangular feature. Robots that have switched to the passive state are not shown. The population of active robots decreases as the robots move eastward and encounter the feature in their path.

Figures 2, 3, 4, 5, 6, 7, 8 and 9 illustrate the results of our mapping procedure for the six scenarios that we investigated. Each figure shows the actual feature, the map of the feature given by the estimated $K(\mathbf{x})$, and the error between these two plots.



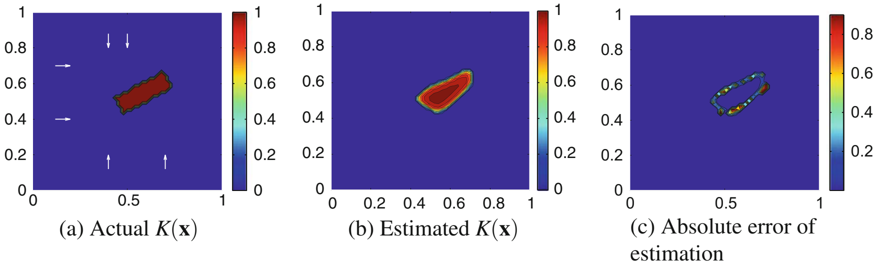
(a) $t = 8$ s (b) $t = 27$ s (c) $t = 53$ s (d) $t = 75$ s

Fig. 1 Snapshots of the simulated swarm moving through a domain with a rectangular feature



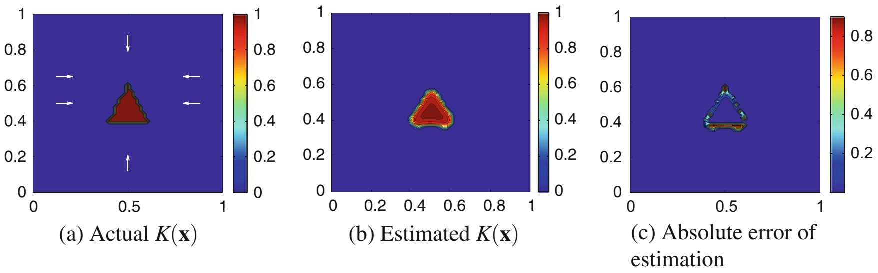
(a) Actual $K(\mathbf{x})$ (b) Estimated $K(\mathbf{x})$ (c) Absolute error of estimation

Fig. 2 $K(\mathbf{x})$ estimated from 6 data sets for a domain that contains a *rectangle*. The *white arrows* show the starting locations and directions of the swarm deployments



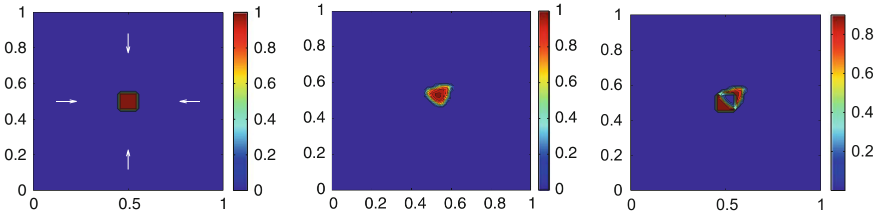
(a) Actual $K(\mathbf{x})$ (b) Estimated $K(\mathbf{x})$ (c) Absolute error of estimation

Fig. 3 $K(\mathbf{x})$ estimated from 6 data sets for a domain that contains an inclined *rectangle*



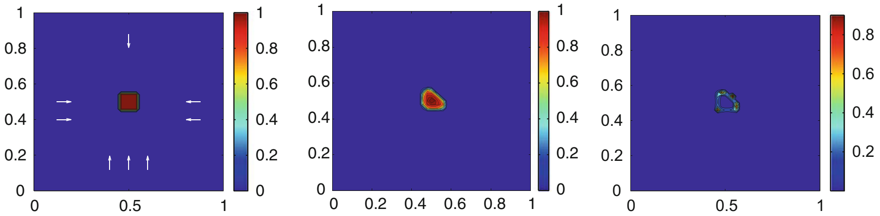
(a) Actual $K(\mathbf{x})$ (b) Estimated $K(\mathbf{x})$ (c) Absolute error of estimation

Fig. 4 $K(\mathbf{x})$ estimated from 6 data sets for a domain that contains a *triangle*



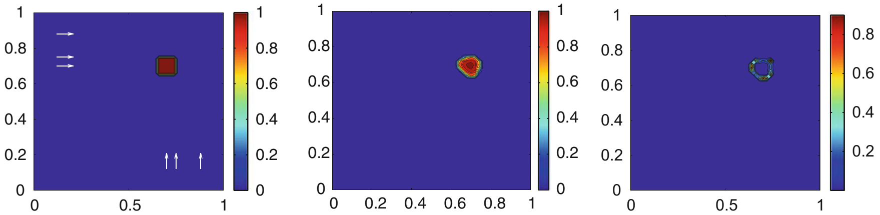
(a) Actual $K(\mathbf{x})$ (b) Estimated $K(\mathbf{x})$ (c) Absolute error of estimation

Fig. 5 $K(\mathbf{x})$ estimated from 4 data sets for a domain that contains a *square* at the center



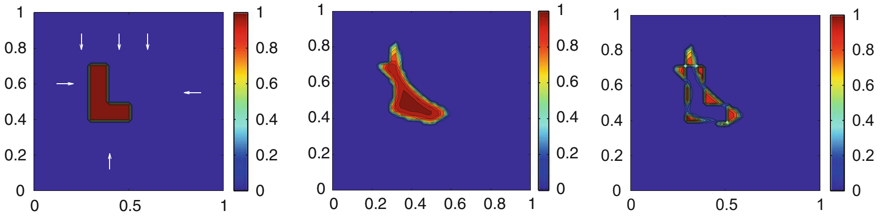
(a) Actual $K(\mathbf{x})$ (b) Estimated $K(\mathbf{x})$ (c) Absolute error of estimation

Fig. 6 $K(\mathbf{x})$ estimated from 8 data sets for a domain that contains a *square* at the center



(a) Actual $K(\mathbf{x})$ (b) Estimated $K(\mathbf{x})$ (c) Absolute error of estimation

Fig. 7 $K(\mathbf{x})$ estimated from 8 data sets for a domain that contains a *square* in the corner



(a) Actual $K(\mathbf{x})$ (b) Estimated $K(\mathbf{x})$ (c) Absolute error of estimation

Fig. 8 $K(\mathbf{x})$ estimated from 6 data sets for a domain that contains a non-convex L-shaped object

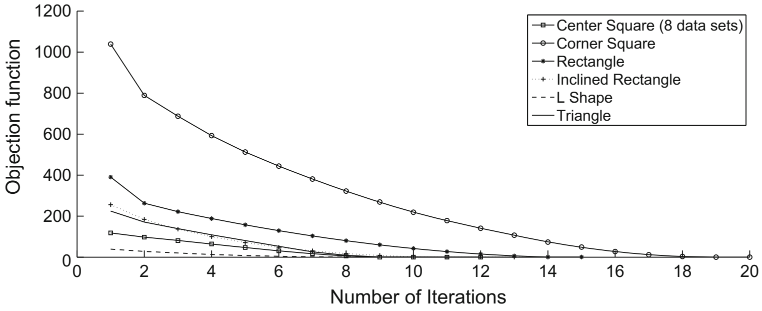


Fig. 9 Objective function value versus number of iterations for the different scenarios examined

In the plots of the actual features, the white arrows indicate the starting points and directions of the swarm center of mass during deployments, each of which yields one data set. Figures 2, 3, and 4 show that we can obtain a fairly accurate map of a rectangle at two different orientations and a triangle using 6 data sets for each scenario. We consider smaller features in the next three figures. From Figs. 5 and 6, we see that the map of a feature increases in accuracy when more non-redundant data sets are used in the optimization procedure. Figure 7 represents a worst-case scenario, in which the map is estimated using data from swarms that start at locations far from the feature, which is in one corner of the domain. The swarms are highly diffused by the time they reach the vicinity of the square; however, 8 data sets yield a relatively accurate map. Lastly, Fig. 8 shows that 6 data sets yield a fairly poor estimate of a non-convex L-shaped feature; we will work further on extending our technique to mapping non-convex shapes. Figure 9 shows that for each scenario considered, the optimal control approach effectively minimizes the objective function by driving it close to zero from its initial value.

6 Conclusion

We have presented a method for mapping an environmental feature using a robotic swarm that exhibits diffusive motion and lacks localization and inter-robot communication. Our approach employs optimal control techniques to reconstruct a spatially varying function that represents the feature of interest. This function is estimated using temporal data on the proportion of active robots, which have not encountered the feature, at each instant of time. Our simulation results indicate that this methodology can accurately reconstruct the feature when the data is obtained from multiple swarm deployments that originate at different locations throughout the domain.

In future work, we would like to extend this approach to more accurately reconstruct non-convex shapes, as well as multiple features in a domain. Our mathematical framework can in principle be used to reconstruct an arbitrary feature geometry, provided that we can design swarm trajectories that yield robot interactions with all facets of the feature. In general, however, it is impossible to identify trajectories a priori that can produce sufficient data for accurate reconstruction. This limitation makes it difficult to map complex feature geometries, as illustrated by Fig. 8. Another factor that contributes to mapping inaccuracies is the decrease in number of active robots during a swarm deployment, which can reduce the correspondence between the density fields of active robots from the macroscopic and microscopic models. This issue could be resolved if the robots perform an obstacle avoidance maneuver upon encountering a feature, staying in the active state rather than entering the passive state. The corresponding macroscopic PDE would need to model this avoidance behavior, which would increase the complexity of computing the gradient of the objective functional. In addition, we plan to implement our mapping approach as the initial step in other swarm strategies, such as collective transport tasks [24] that first require estimating the location and geometry of the payload.

Acknowledgements This work was supported by NSF Awards CMMI-1363499 and CMMI-1436960.

Appendix 1: Mathematical Preliminaries

We study the solution to PDEs in the weak sense, which can be found in the Sobolev space $H^1(\Omega) = \left\{ y \in L^2(\Omega) : \frac{\partial y}{\partial x_1} \in L^2(\Omega), \quad \frac{\partial y}{\partial x_2} \in L^2(\Omega) \right\}$. Here, the spatial derivative is to be understood as a weak derivative defined in the distributional sense. The space is equipped with the common Sobolev space norm, $\|y\|_{H^1(\Omega)} = \sqrt{\left(\|y\|^2_{L^2(\Omega)} + \sum_{i=1}^{2} \left\| \frac{\partial y}{\partial x_i} \right\|^2_{L^2(\Omega)} \right)}$. We also define $V = H^1(\Omega)$, which has the dual space $V^* = H^1(\Omega)^*$.

We consider the general system for Eqs. (3)–(5):

$$\begin{aligned} \frac{\partial u}{\partial t} &= Au + \sum_{i=1}^{2} v_i B_i u - K(\mathbf{x})u + f \quad in \;\; L, \\ \mathbf{n} \cdot (D\nabla u - \mathbf{v}u) &= g \quad on \;\; \Gamma, \\ u(\mathbf{x}, 0) &= u_0, \end{aligned} \tag{9}$$

where A is a formal operator and B_i is an operator defined as $B_i : L^2(0, T; V) \to L^2(0, T; L^2(\Omega))$, $K(\mathbf{x}) \in L^2(\Omega)$, $f \in F = L^2(0, T; L^2(\Omega))$ is the forcing function in the system, $g \in G = L^2(0, T; L^2(\partial\Omega))$, and $u_0 \in L^2(\Omega)$. The variational form of the operator A, called A_g, is defined as $A_g : L^2(0, T; V) \to L^2(0, T; V^*)$. The solution of the system in the weak sense is given by $u \in U = L^2(0, T; V)$ with $u_t \in U^* = L^2(0, T; V^*)$ if it satisfies the equation:

$$\left\langle \frac{\partial u}{\partial t}, \phi \right\rangle_{U^*,U} = \left\langle A_g, \phi \right\rangle_{U^*,U} + \sum_{i=1}^{2} \langle v_i B_i u, \phi \rangle_F - \langle K(\mathbf{x})u, \phi \rangle_F + \langle f, \phi \rangle_F \quad (10)$$

for all $\phi \in L^2(0, T; V)$. The boundary conditions are equipped with A_g in the variational formulation using Green's theorem. This is essentially the variational form of the Laplacian,

$$\left\langle A_g u, \phi \right\rangle_{U^*,U} = -\langle D\nabla u, \nabla\phi \rangle_{L^2(\Omega)} + \int_{\partial\Omega} (g + \mathbf{n} \cdot \mathbf{v}u)\, \phi dx. \quad (11)$$

In the macroscopic model Eqs. (3)–(5), we define $A = \nabla^2$, $B_i = \frac{\partial}{\partial x_i}$, $f = 0$, and $g = 0$.

Appendix 2: Adjoint Equations

The adjoint equation $\nabla_{\mathbf{u}}\mathscr{L} = 0$ implies that $[\nabla_{u_1}\mathscr{L}, \ldots, \nabla_{u_i}\mathscr{L}, \ldots, \nabla_{u_N}\mathscr{L}] = 0$. From Eq. (8),

$$\begin{aligned} \nabla_{u_i}\mathscr{L} &= \nabla_{u_i}\mathbf{J}(\mathbf{u}, K) + \nabla_{u_i} \sum_{j=1}^{N} \langle p_j, \Psi_j(u_j, K)\rangle \\ &= \nabla_{u_i}\mathbf{J}(u_i, K) + \nabla_{u_i}\langle p_i, \Psi_i(u_i, K)\rangle, \end{aligned} \quad (12)$$

since a term in the sum is a function of u_i only when $i = j$. By Eq. (7),

$$\nabla_{u_i}\mathbf{J}(u_i, K) = \nabla_{u_i} \sum_{j=1}^{N} W_j J_j(u_j) = W_i \nabla_{u_i} J_i(u_i). \quad (13)$$

From Eq. (6),

$$\nabla_{u_i} J_i(u_i) = \nabla_{u_i} \left(\frac{1}{2} \|(Du_i)(t) - g_i(t)\|^2_{L^2([0,T])} \right), \quad (14)$$

where $D := U \to L^2([0, T])$ and $(Du_i)(t) = \int_\Omega u_i(\mathbf{x}, t)d\mathbf{x}$. Then, by the chain rule of differentiation [4, 12], the directional derivative of $J_i(u_i)$, $\nabla_{u_i} J_i(u_i)$, is given by

$$\langle \nabla_{u_i} J_i(u_i), s\rangle_U = \langle (Du_i)(t) - g_i(t), Ds\rangle_{L^2([0,T])} = \langle D^*((Du_i)(t) - g_i(t)), s\rangle_U. \quad (15)$$

Here, $D^* := L^2([0, T]) \to U$ and $(D^* f)(t) = f(t) \cdot \mathbf{1}_\Omega(\mathbf{x})$, where $f(t) \in L^2([0, T])$ and $\mathbf{1}_\Omega$ is the indicator function of $\Omega \subset \mathbb{R}^2$. We can show that $\langle Dy, f\rangle = \langle y, D^* f\rangle \; \forall y \in U, \; f \in L^2([0, T])$. Therefore,

$$\nabla_{u_i} J_i(u_i) = D^*((Du_i)(t) - g_i(t)). \tag{16}$$

By definition,

$$\langle p_i, \nabla_{u_i} \Psi_i(u_i, K)s\rangle = \langle \nabla_{u_i} \Psi_i(u_i, K)^* p_i, s\rangle \;\; \forall s \in U, \tag{17}$$

where $\nabla_{u_i} \Psi_i(u_i, K)^*$ is the adjoint operator of $\nabla_{u_i} \Psi_i(u_i, K)$ corresponding to the inner product of the Hilbert space. Now, by taking the directional derivative of $\Psi_i(u_i, K)$ at u_i in the direction of s, we obtain

$$\nabla_{u_i} \Psi_i(u_i, K)s = \frac{\partial s}{\partial t} - (\nabla \cdot (D\nabla s - \mathbf{v}_i(t)s) - kK(\mathbf{x})s). \tag{18}$$

Substituting Eq. (18) into Eq. (17) yields

$$\langle p_i, \nabla_{u_i} \Psi_i(u_i, K)s\rangle = \int_0^T \langle p_i, \frac{\partial s}{\partial t}\rangle_{L^2(\Omega)} - \langle p_i, D\nabla^2 s\rangle + \langle p_i, \nabla \cdot \mathbf{v}_i(t)s\rangle + \langle p_i, kK(\mathbf{x})s\rangle.$$

Using integration by parts on the integral term in the equation above, we get

$$\int_0^T \langle p_i, \frac{\partial s}{\partial t}\rangle_{L^2(\Omega)} = \langle p_i(T), s(T)\rangle - \langle p_i(0), s(0)\rangle - \int_0^T \langle s, \frac{\partial p_i}{\partial t}\rangle_{L^2(\Omega)}.$$

As this is true for all $s \in U$, we could choose the s with $s(0) = 0$ and construct $p_i(T)$ such that $\int_0^T \langle p_i, \frac{\partial s}{\partial t}\rangle_{L^2(\Omega)} = \int_0^T \langle -\frac{\partial p_i}{\partial t}, s\rangle_{L^2(\Omega)}$. Thus, we choose the final condition of the adjoint equation as $p_i(T) = 0$. We now make use of the following lemma:

Lemma 1 *Let L and L^* be operators defined by $L : L^2(0, T; V) \to L^2(0, T; V^*)$ and $L^* : L^2(0, T; V) \to L^2(0, T; V^*)$, respectively. The variational form of L is:*

$$\langle Lu, \phi\rangle_{V^*,V} = -\langle D\nabla u, \nabla\phi\rangle_{L^2(\Omega)} - \langle \mathbf{v} \cdot \nabla u, \phi\rangle_{L^2(\Omega)} + \int_{\partial\Omega} \mathbf{n} \cdot (\mathbf{v}u\phi)dx$$

$\forall \phi \in V$. *Also, by Lagrange's identity,* $\langle Lu, p\rangle_{V^*,V} = \langle u, L^* p\rangle_{V,V^*}$ $\forall u, p \in L^2(0, T; V)$. *We use the zero-flux boundary condition in Eq.* (4) *to compute the variational form of the operator L^* to be* $\langle L^* p, \phi\rangle_{V^*,V} = -\langle D\nabla p, \nabla\phi\rangle_{L^2(\Omega)} + \langle \mathbf{v} \cdot \nabla p, \phi\rangle_{L^2(\Omega)}$ $\forall p \in L^2(0, T; V)$ *and* $\forall \phi \in V$.

Using the variational form of the Laplacian as in Eq. (11) and applying Lemma 1 and integration by parts, we can show that $-\langle p_i, D\nabla^2 s\rangle + \langle p_i, \nabla \cdot \mathbf{v}_i(t)s\rangle$ can be transformed into $-\langle D\nabla^2 p_i, s\rangle - \langle \nabla \cdot \mathbf{v}_i(t)p_i, s\rangle$ with the boundary condition $\mathbf{n} \cdot \nabla p_i = 0$. Finally, we observe that $\langle p_i, K(\mathbf{x})s\rangle = \langle p_i K(\mathbf{x}), s\rangle$. By combining these results with Eqs. (12), (15), and (17), we obtain

$$\langle \nabla_{u_i} J_i(u_i), s\rangle + \langle -\frac{\partial p_i}{\partial t} - D\nabla^2 p_i - \nabla \cdot \mathbf{v}_i(t)p_i + p_i kK(\mathbf{x}), s\rangle = 0.$$

Thus, the set of adjoint equations for the system defined by the i^{th} set of constraints, $\Psi_i(u_i, K)$, with respect to the objective functional, $\mathbf{J}$, is given by

$$-\frac{\partial p_i}{\partial t} = \nabla \cdot (D\nabla p_i + \mathbf{v}_i(t)p_i) - p_i k K(\mathbf{x}) - \nabla_{u_i} J_i(u_i) \quad in \ \ L \tag{19}$$

with the Neumann boundary conditions

$$\mathbf{n} \cdot \nabla p_i = 0 \quad on \ \ \Gamma, \quad p_i(T) = 0, \quad i = 1, \ldots, N. \tag{20}$$

Here, Eq. (19) with Eq. (20) has a solution in the weak sense.

Appendix 3: Gradient Equation

Using a similar analysis to the one in Appendix 2, we find that $\nabla_K \mathscr{L}$ reduces to

$$\nabla_K \mathscr{L} = \nabla_K \mathbf{J}(\mathbf{u}, K) + \sum_{i=1}^{N} \nabla_K \langle p_i, \Psi_i(u_i, K)\rangle. \tag{21}$$

From Eq. (7), we can derive the following expressions:

$$\nabla_K \mathbf{J}(\mathbf{u}, K) = \nabla_K \frac{\lambda}{2} \|K(\mathbf{x})\|^2_{L^2(\Omega)}, \quad \langle \nabla_K \mathbf{J}(\mathbf{u}, K), s\rangle = \langle \lambda K(\mathbf{x}), s\rangle. \tag{22}$$

As in Appendix 2, we could express $\langle p_i, \nabla_K \Psi_i(u_i, K)s\rangle$ as $\langle \nabla_K \Psi_i(u_i, K)^* p_i, s\rangle \ \forall s \in L^2(\Omega)$, where $\nabla_K \Psi_i(u_i, K)^*$ is the adjoint operator of $\nabla_K \Psi_i(u_i, K)$ corresponding to the inner product of the Hilbert space. Now, by taking the directional derivative of $\Psi_i(u_i, K)$ at K in the direction of s, we find that $\nabla_K \Psi_i(u_i, K)s = k u_i s$. Therefore, with further simplification, we can show that

$$\langle \nabla_K \Psi_i(u_i, K)^* p_i, s\rangle = \langle (\Xi(k u_i p_i))(\mathbf{x}), s\rangle_{L^2(\Omega)}, \tag{23}$$

where $\Xi := L^2(0, T; \Omega) \to L^2(\Omega)$ and $(\Xi f)(\mathbf{x}) = \int_0^T f dt$ for all $f \in L^2([0, T]; \Omega)$ and $\mathbf{x} \in \Omega$. By combining Eqs. (21)–(23), we formulate the objective functional derivative as

$$\mathbf{J}' = \sum_{i=1}^{N} (\Xi(k u_i p_i))(\mathbf{x}) + \lambda K(\mathbf{x}). \tag{24}$$

Thus, the computation of $\mathbf{J}'$ requires u_i and p_i, which can be obtained by solving $\Psi_i(u_i, K)$ forward and solving Eqs. (19), (20) backward.

References

1. Ammari, H.: An Introduction to Mathematics of Emerging Biomedical Imaging, vol. 62. Springer, Berlin (2008)
2. Belmiloudi, A.: Stabilization, Optimal and Robust Control: Theory and Applications in Biological and Physical Sciences. Springer, London (2008)
3. Biswas, R., Limketkai, B., Sanner, S., Thrun, S.: Towards object mapping in dynamic environments with mobile robots. In: International Conference on Intelligent Robots and Systems (IROS) (2002)
4. Borzì, A., Schulz, V.: Computational optimization of systems governed by partial differential equations. In: Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2012)
5. Correll, N., Hamann, H.: Probabilistic modeling of swarming systems. In: J Kacprzyk WP (ed) Springer Handbook of Computational Intelligence, pp 1423–1431. Springer, Heidelberg (2015)
6. Dirafzoon, A., Lobaton, E.: Topological mapping of unknown environments using an unlocalized robotic swarm. In: International Conference on Intelligent Robots and Systems (IROS) (2013)
7. Dirafzoon, A., Betthauser, J., Schornick, J., Benavides, D., Lobaton, E.: Mapping of unknown environments using minimal sensing from a stochastic swarm. In: International Conference on Intelligent Robots and Systems (IROS) (2014)
8. Elamvazhuthi, K.: A variational approach to planning, allocation and mapping in robot swarms using infinite dimensional models. Master's thesis, Arizona State University (2014)
9. Elamvazhuthi, K., Berman, S.: Optimal control of stochastic coverage strategies for robotic swarms. In: International Conference on Robotics and Automation (ICRA) (2015)
10. Fattorini, H.O.: Infinite Dimensional Optimization and Control Theory, vol 54. Cambridge University Press, Cambridge (1999)
11. Gardiner, C.W.: Stochastic Methods: A Handbook for the Natural and Social Sciences, 4th edn. Springer, Evanston, IL, USA (2009)
12. Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: Optimization with PDE Constraints, vol. 23. Springer, Netherlands (2009)
13. Horning, M., Lin, M., Siddarth, S., Zou, S., Haberland, M., Yin, K., Bertozzi, A.L.: Compressed sensing environmental mapping by an autonomous robot. In: Proceedings of the Second International Workshop on Robotic Sensor Networks. Seattle, WA (2015)
14. Kirsch, A.: An Introduction to the Mathematical Theory of Inverse Problems, 2nd edn. vol 120. Springer, New York (2011)
15. Kuipers, B., Byun, Y.T.: A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Int. J. Robot. Auton. Syst. **8**(1–2), 47–63 (1991)
16. Liu, S., Mohta, K., Shen, S., Kumar, V.: Towards collaborative mapping and exploration using multiple micro aerial robots. In: International Symposium on Experimental Robotics (ISER) (2014)
17. Robb, R.A.: Biomedical Imaging, Visualization, and Analysis. Wiley Inc, New York, NY, USA (1999)
18. Robertson, P., Angermann, M., Krach, B.: Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors. In: Proceedings of the 11th International Conference on Ubiquitous Computing (Ubicomp) (2009)
19. Thrun, S.: A probabilistic online mapping algorithm for teams of mobile robots. Int. J. Robot. Res. **20**(5), 335–363 (2001)
20. Thrun, S., Bücken, A.: Integrating grid-based and topological maps for mobile robot navigation. In: Proceedings of the AAAI 13th National Conference on Artificial Intelligence (1996)
21. Tong, S., Fine, E.J., Lin, Y., Cradick, T.J., Bao, G.: Nanomedicine: tiny particles and machines give huge gains. Ann. Biomed. Eng. **42**(2), 243–259 (2014)
22. Tröltzsch, F.: Optimal Control of Partial Differential Equations: Theory, Methods, and Applications, vol. 112. American Mathematical Society, Providence, RI, USA (2010)

23. Tuchin, V.V.: Tissue Optics, Light Scattering Methods and Instruments for Medical Diagnosis, 3rd edn. vol. PM254. Spie Press Book (2015)
24. Wilson, S., Pavlic, T.P., Kumar, G.P., Buffin, A., Pratt, S.C., Berman, S.: Design of ant-inspired stochastic control policies for collective transport by robotic swarms. Swarm Intell. **8**(4), 303–327 (2014)