Distributed Reduced Order Source Identification

Reza Khodayi-mehr, Wilkins Aquino, and Michael M. Zavlanos

***Abstract*—In this paper we propose a distributed approach for model-based Source Identification (SI) that minimizes communication cost and allows for on demand balancing of computational resources, based on the specifications of the sensors. Specifically, we consider the steady-state Advection-Diffusion equation which we discretize using the Finite Element (FE) method, and then apply Proper Orthogonal Decomposition to reduce the order of the model. The concentration measurements that are needed to solve the SI problem are collected by a team of mobile sensors that move in pre-assigned subdomains in the environment. We formulate an ℓ_2-regularized least squares optimization problem that the sensors solve in a distributed way using the Accelerated Distributed Augmented Lagrangian method. Our formulation results in an algorithm whose communication cost is independent of the FE mesh size and its time complexity grows only linearly with it. We present simulation results that show that the proposed method can handle large-scale SI problems and compare our formulation to an alternative derivation using Alternating Direction Method of Multipliers.**

I. Introduction

The problem of Source Identification (SI) refers to the estimation of a source using a set of measurements of a quantity, such as concentration, that is generated under the action of that source. The SI problem has various applications including locating atmospheric, underground, or underwater pollutants, finding the source of a hazardous chemical leakage, demining, and fire detection.

Because of its importance, SI problem has been studied in the literature for a long time. Particularly, odor localization has been addressed using a single robot, a robotic swarm, or multi-agent networks. Most of these approaches rely on heuristics that attempt to detect the plume and track it in the upwind direction until the robots reach the source [1]–[3]. These heuristic approaches, although very practical, do not generalize. They often localize a single or at most multiple point sources in convex domains and do not provide any information about the intensity of the sources. These issues can be resolved by using the mathematical model of the transport phenomenon which leads to model-based SI methods. For odor localization, this model is an Advection-Diffusion (AD) Partial Differential Equation (PDE).

Model-based SI is a difficult problem and often simplifying assumptions, e.g., point sources, are made to reduce the computational cost of evaluating the transport model. For example, the localization of a single point source in steady-state and in a semi-infinite domain is considered in [4]. The case of multiple point sources is often considered using optimization-based methods. For instance, the work in [5] addresses the detection and localization of multiple point sources using a wireless sensor network.

More general problems that involve sources of arbitrary shapes in arbitrary domains are typically solved numerically using, e.g., the Finite Element (FE) method. The authors in [6] use the FE method along with total variation regularization to solve the SI problem. Similarly, in our previous work [7], we proposed the Reweighted Debiased ℓ_1 algorithm, which is an iterative sparse recovery approach to the SI problem. Nevertheless, numerical methods such as FE become computationally demanding as the size of the domain grows.

Model order reduction techniques allow for efficient approximate solutions of PDEs by significantly reducing the dimension of the state space of the problem. As such, they can be used for online computation and control; any expensive computations required in these methods, e.g., application of the FE method, are typically performed offline [8]. In this work, we use Proper Orthogonal Decomposition (POD) to construct a closed-form reduced order model that given a source vector, predicts the corresponding concentration field [9]. Then using this reduced model, we decompose the unknown source vector among the agents such that none of them has access to the complete unknown source vector or the AD model. We define the SI problem as an ℓ_2-regularized least squares problem and formulate it such that the Accelerated Distributed Augmented Lagrangian (ADAL) algorithm [10] can be used to solve it in a decentralized way. This allows us to enforce consensus over concentration measurements instead of unknown source vector making the communication cost independent of the FE mesh size. Decomposition of the unknown source vector and the task of collecting measurements enables perfect load balancing according to the agents' computational capacity. Our method can handle large-scale problems with multiple sources of arbitrary shape in non-convex environments and is guaranteed to converge to the unique optimal solution of the SI problem. We analyze the time complexity of the proposed method and show that it grows only linearly as the FE mesh size increases. We also implement our algorithm in an online way, where the agents collect their measurements as they solve the SI problem, allowing them to maintain a real-time estimation of the source. Finally, we compare our algorithm to an alternative solution that we present based on Alternating Direction Method of Multipliers.

The remainder of this paper is organized as follows. In Section II, we introduce the AD model and formulate the SI problem. Section III is devoted to the description of the distributed SI algorithm. In Section IV, simulation results are presented demonstrating the performance of the algorithm and Section V concludes the paper.

This work is supported in part by the NSF award CNS #1261828.

Reza Khodayi-mehr and Michael M. Zavlanos are with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA, {reza.khodayi.mehr, michael.zavlanos}@duke.edu. Wilkins Aquino is with the Department of Civil and Environmental Engineering, Duke University, Durham, NC 27708, USA, wilkins.aquino@duke.edu.

II. Source Identification Problem

A. Advection-Diffusion Model

Let $\Omega \subset \mathbb{R}^d$ be the given domain of interest ($1 \leq d \leq 3$), and assume the presence of sources is modeled by a function $s : \Omega \to \mathbb{R}$. Let $c : \Omega \to \mathbb{R}$ be the measurable quantity, such as concentration, generated by this source function. Moreover, let the velocity at which this quantity is transported via advection be $\mathbf{q} \in \mathbb{R}^d$ and $D \in \mathbb{R}_+$ denote the diffusivity of the medium. Under a steady-state assumption on the Advection-Diffusion (AD) PDE and applying a zero-valued Dirichlet condition to the boundaries $\partial\Omega$ of domain, we arrive at the following Boundary Value Problem (BVP)

$$\nabla \cdot (-D\nabla c + c\,\mathbf{q}) - s = 0 \quad \text{in} \quad \Omega, \tag{1a}$$

$$c = 0 \quad \text{on} \quad \partial\Omega, \tag{1b}$$

which predicts the concentration field c for a given source function s; see [11, ch. 2]. Note that we can easily accommodate more sophisticated boundary conditions in this formulation and for smooth enough source function s, we can show that the BVP (1) has a unique stable solution. For theoretical details, see, e.g., [12, ch. 8,9].

In order to compute the concentration function c, we need to solve the BVP (1) for a given source term s. To this end, we utilize the Finite Element (FE) method which provides an approximation of the concentration function. Specifically, consider a FE-mesh with n grid points and let $\mathbf{c}, \mathbf{s} \in \mathbb{R}^n$ denote the vectors of nodal values of the concentration function c and source function s at these points. Then, following a standard procedure, we construct the system matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ and the mass matrix $\mathbf{R} \in \mathbb{R}^{n\times n}$ such that

$$\mathbf{K}\mathbf{c} = \mathbf{R}\mathbf{s}. \tag{2}$$

Having the nodal values $\mathbf{c}$ for a given source function s, we can approximate the concentration function at an arbitrary point $\mathbf{p} \in \Omega$ by piecewise polynomial interpolation; see, e.g., [13].

The computational cost of solving the linear system (2) becomes prohibitive as n grows large. This could happen if the domain Ω is very large or if we require highly accurate approximations of the continuous functions. Model order reduction methods can address this limitation by performing the computationally expensive operations offline and providing a model whose online evaluation is considerably faster [14]. In this paper, we use Proper Orthogonal Decomposition (POD) that relies on evaluating the FE model (2) for a candidate set of source terms offline to build a set of so called snapshots of the concentration field. Then, these snapshots are used to build a set of N optimal basis functions $\phi_i(\mathbf{p})$ for $i \in \{1, \ldots, N\}$ where $N \ll n$. Note that the functions $\phi_i(\mathbf{p})$ are represented in the FE bases by corresponding nodal values $\bar{\boldsymbol{\phi}}_i \in \mathbb{R}^n$. See [9] for details of implementation.

Let $\boldsymbol{\Phi} \in \mathbb{R}^{n\times N}$ be defined as $\boldsymbol{\Phi} = [\bar{\boldsymbol{\phi}}_1, \ldots, \bar{\boldsymbol{\phi}}_N]$. Then we reduce the order of the FE model (2) by projecting the matrices $\mathbf{K}, \mathbf{R}$ onto the space spanned by the POD bases as $(\boldsymbol{\Phi}^T\mathbf{K}\boldsymbol{\Phi})\,\bar{\mathbf{c}} = (\boldsymbol{\Phi}^T\mathbf{R})\,\mathbf{s}$, where $\bar{\mathbf{c}} \in \mathbb{R}^N$ denotes the concentration coefficients corresponding to the POD basis functions. Given a discretized source vector $\mathbf{s}$ corresponding to the source function s, we obtain an approximation to the concentration function as $\bar{c}(\mathbf{p}) = \boldsymbol{\phi}(\mathbf{p})\,\bar{\mathbf{c}} = \boldsymbol{\phi}(\mathbf{p})\,\mathbf{M}\,\mathbf{s}$, where $\boldsymbol{\phi}(\mathbf{p}) = [\phi_1(\mathbf{p}), \ldots, \phi_N(\mathbf{p})]$ is the $1 \times N$ row vector of POD basis functions evaluated at $\mathbf{p} \in \Omega$ and $\mathbf{M} \in \mathbb{R}^{N\times n}$ is the reduced order matrix defined as

$$\mathbf{M} = (\boldsymbol{\Phi}^T\mathbf{K}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{R}. \tag{3}$$

Note that in order to compute the multiplication of $\mathbf{M}$ with a source vector $\mathbf{s}$, we need to solve an $N \times N$ system of equations instead of (2). This is very favorable as N is typically orders magnitude smaller than n. In fact we can directly perform the inversion in (3) and compute $\mathbf{M}$ in closed-form. We compactly represent the concentration function as

$$\bar{c}(\mathbf{p}) = \mathbf{m}(\mathbf{p})\,\mathbf{s}, \tag{4}$$

where $\mathbf{m}(\mathbf{p}) = \boldsymbol{\phi}(\mathbf{p})(\boldsymbol{\Phi}^T\mathbf{K}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{R}$ belongs to $\mathbb{R}^{1\times n}$.

B. Measurement Model

Consider m sensors deployed in the domain Ω that make measurements of the concentration field c and let $\mathbf{y} \in \mathbb{R}^m$ denote the random vector of these measurements given as

$$y_i = c(\mathbf{p}_i) + \epsilon_1(\mathbf{p}_i) + \epsilon_2, \tag{5}$$

where $\mathbf{p}_i \in \Omega$ denotes the i-th measurement location for $i \in \{1, \ldots, m\}$. In this expression, ϵ_1 models the noise component that grows as the signal magnitude increases and ϵ_2 models the background white noise. Specifically, $\epsilon_1(\mathbf{p}_i) \sim N(0, (c(\mathbf{p}_i)\sigma_1)^2)$ and $\epsilon_2 \sim N(0, \sigma_2^2)$. Note that the sensor noise components are identically and independently distributed across the domain as the measurements at different locations do not affect each other. Thus, by adding the two independent Normal random variables, we have $\mathbf{y} \sim N(\boldsymbol{\mu}, \sigma_1^2\,\boldsymbol{\Sigma}_{\mathbf{yy}})$, where $\boldsymbol{\mu} = (c(\mathbf{p}_1), \ldots, c(\mathbf{p}_m))$ is the vector of concentration values (4) at measurement locations and $\boldsymbol{\Sigma}_{\mathbf{yy}} = \text{diag}(\boldsymbol{\mu})^2 + (\sigma_2/\sigma_1)^2\,\mathbf{I}_m$. In this expression, $\text{diag}(\boldsymbol{\mu})$ denotes the matrix whose diagonal entries are $\boldsymbol{\mu}$ and $\mathbf{I}_m$ denotes the $m \times m$ identity matrix.

Using equation (4) we define the design matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ for the linear model (5) as

$$\mathbf{X} = \begin{pmatrix} \mathbf{m}(\mathbf{p}_1) \\ \vdots \\ \mathbf{m}(\mathbf{p}_m) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}(\mathbf{p}_1) \\ \vdots \\ \boldsymbol{\phi}(\mathbf{p}_m) \end{pmatrix} \mathbf{M}, \tag{6}$$

where $\boldsymbol{\phi}(\mathbf{p}_i) = [\phi_1(\mathbf{p}_i), \ldots, \phi_N(\mathbf{p}_i)]$ and the reduced matrix $\mathbf{M}$ is defined by (3). Then, we can represent the distribution of the measurements as $\mathbf{y} \sim N(\mathbf{X}\,\mathbf{s}, \sigma_1^2\,\boldsymbol{\Sigma}_{\mathbf{yy}})$. Given this model, the Source Identification (SI) problem that we consider in this paper can be defined as follows.

Problem II.1 (Source Identification Problem)**.** *Given the reduced model* (4) *and the vector* $\mathbf{y}$ *of* m *noisy measurements, find an estimate* $\mathbf{s}$ *of the true source vector* $\bar{\mathbf{s}}$ *that minimizes the discrepancy between the predicted and measured concentrations in the least-squares sense.*

To solve Problem (II.1), we utilize the classical deterministic formulation of SI problems; see, e.g., [15]. Particularly,

given the global vector of measurements $\mathbf{y}$, we solve an ℓ_2-regularized least-squares problem of the form

$$\min_{\mathbf{s}\in\mathbb{R}^n} \frac{1}{2}\left\|\mathbf{y}-\mathbf{X}\mathbf{s}\right\|^2 + \frac{\tau}{2}\left\|\mathbf{W}\mathbf{s}\right\|^2, \tag{7}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\tau > 0$ is the regularization parameter, and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a diagonal matrix defined by $\mathbf{W} = \text{diag}(\mathbf{w})$, where $\mathbf{w} \in \mathbb{R}^n$ is a vector of weights that is used to enhance sparsity.

III. Distributed Source Identification

A distributed solution of the SI Problem II.1 is neceassry to tackle large-scale SI problems that are encountered in real life. Additionally, it is often desirable that SI problems are solved collaboratively by teams of mobile sensors that are also tasked with collecting the measurements. To solve the SI problem (7) distributedly, we decompose the unknown source vector and the task of collecting measurements among the agents. Then, utilizing the reduced model (4), we formulate the distributed SI problem such that it can be solved using the Accelerated Distributed Augmented Lagrangian (ADAL) algorithm [10].

A. Domain Decomposition over Multi-Agent Network

Assume a network of $\bar{n}$ mobile sensors tasked with solving the SI Problem II.1. Consider the graph representation $\mathcal{G}(\mathcal{V},\mathcal{E})$ of the network where $\mathcal{V} = \{1,\dots,\bar{n}\}$ denotes the set of vertices and $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ is the set of edges. Define also $\mathcal{N}_j = \{i \in \mathcal{V} \,|\, (i,j) \in \mathcal{E}\}$ as the set of neighbors of agent j. Without loss of generality, we assume that the topology of the network is fixed and decided based on the communication capabilities of the agents.

To solve the SI problem (7) distributedly, we decompose the unknown source vector. Specifically, we consider a non-overlapping decomposition of Ω where Ω_j for $j \in \mathcal{V}$ denotes the subdomain assigned to agent j such that $\cup_{j=1}^{\bar{n}}\Omega_j = \Omega$ and $\Omega_j \cap \Omega_i = \emptyset$ for $i \in \mathcal{V}\setminus\{j\}$. Assuming that the boundaries of Ω_j do not pass through any finite elements, this decomposition partitions the FE mesh into smaller meshes corresponding to the subdomains with n_j nodes such that $\sum_{j=1}^{\bar{n}} n_j = n$ where n is total number of FE grid points. We assign to each agent j, the sub-vector $\mathbf{s}_j \in \mathbb{R}^{n_j}$ corresponding to the decomposed FE mesh of subdomain Ω_j. This decomposition of the source vector requires the decomposition of the columns of matrix $\mathbf{M}$ accordingly, i.e., we have $\mathbf{M} = [\mathbf{M}_1,\dots,\mathbf{M}_{\bar{n}}]$ in (3) where $\mathbf{M}_j \in \mathbb{R}^{N\times n_j}$. This decomposition is possible since we have this matrix in closed-form as a result of using the POD method; cf. Section II-A. Note that we can arbitrarily select the entries of the source vector $\mathbf{s}$ that we assign to each agent, i.e., the subdomains Ω_j are not necessarily connected regions. This flexibility allows us to select these subdomains according to the computational resources of agents and achieve perfect load balancing. Furthermore, as a result of this decomposition no one agent has access to the global model or the complete source vector which can be important for security reasons.

To distribute the task of measurement collection among the agents, we consider another decomposition of the domain Ω into subdomains $\bar{\Omega}_i$ for $i \in \mathcal{V}$. Note that unlike the previous case, these subdomains can overlap and must be connected since the agents traverse them to collect measurements. We assume that each agent i collects m_i measurements from subdomain $\bar{\Omega}_i$ where $m = \sum_{i=1}^{\bar{n}} m_i$ is the total number of measurements. Then, stacking together the m_i measurements of agent i in the design matrix (6) and using the above decomposition of the source term, we partition the design matrix (6) into $\bar{n}\times\bar{n}$ blocks as

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_{11} & \dots & \mathbf{X}_{1\bar{n}} \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{\bar{n}1} & \dots & \mathbf{X}_{\bar{n}\bar{n}} \end{pmatrix}, \tag{8}$$

where $\mathbf{X}_{ij} \in \mathbb{R}^{m_i\times n_j},\ \forall\, i,j \in \mathcal{V}$. The partition $\mathbf{X}_{ij}$ in column j and row i of matrix (8) belongs to agent j and corresponds to the measurements collected by agent i. In practice, we assign to each agent j the POD basis functions $\boldsymbol{\phi}(\mathbf{x})$ and the matrix $\mathbf{M}_j$ where $\mathbf{M} = [\mathbf{M}_1,\dots,\mathbf{M}_{\bar{n}}]$ is defined in (3). Then, given the measurement locations of agent i, agent j calculates the partition $\mathbf{X}_{ij}$ of the design matrix from (6).

B. Distributed Source Identification Algorithm

Using the domain decomposition discussed in Section III-A, we can rewrite the SI objective (7) as $\sum_{i=1}^{\bar{n}}(1/2\left\|\mathbf{y}_i - \sum_{j=1}^{\bar{n}}\mathbf{X}_{ij}\mathbf{s}_j\right\|^2 + \tau/2\left\|\mathbf{W}_i\mathbf{s}_i\right\|^2)$, where $\mathbf{s}_j \in \mathbb{R}^{n_j}$ is the local source vector of agent j, $\mathbf{W}_j = \text{diag}(\mathbf{w}_j)$ with $\mathbf{w}_j \in \mathbb{R}^{n_j}$ denoting the weights corresponding to subdomain Ω_j, and $\mathbf{y}_i \in \mathbb{R}^{m_i}$ is the vector of measurements of agent i.

Let $\mathbf{b}_i \in \mathbb{R}^{m_i}$, given by $\mathbf{b}_i = \sum_{j=1}^{\bar{n}}\mathbf{X}_{ij}\mathbf{s}_j$, denote the estimated concentration at the measurement locations of agent i given the local source estimates $\mathbf{s}_j \in \mathbb{R}^{n_j},\ \forall\, j \in \mathcal{V}$. Then, we can rewrite the SI problem (7) as

$$\begin{aligned} &\min_{\mathbf{s}_i,\mathbf{b}_i} \sum_{i=1}^{\bar{n}} \left\{ \frac{1}{2}\left\|\mathbf{y}_i - \mathbf{b}_i\right\|^2 + \frac{\tau}{2}\left\|\mathbf{W}_i\mathbf{s}_i\right\|^2 \right\}, \\ &\text{s.t. } \mathbf{b}_i = \sum_{j=1}^{\bar{n}} \mathbf{X}_{ij}\mathbf{s}_j,\ \forall i \in \mathcal{V}. \end{aligned} \tag{9}$$

Define local matrices $\mathbf{A}_j \in \mathbb{R}^{m\times(n_j+m_j)}$ as

$$\mathbf{A}_j = \begin{pmatrix} \mathbf{X}_{1j} & \mathbf{0}_{m_1} \\ \vdots & \vdots \\ \mathbf{X}_{jj} & -\mathbf{I}_{m_j} \\ \vdots & \vdots \\ \mathbf{X}_{\bar{n}j} & \mathbf{0}_{m_{\bar{n}}} \end{pmatrix}, \tag{10}$$

where $\mathbf{0}_{m_j}, \mathbf{I}_{m_j}$ denote the all-zero and identity matrices of size $m_j \times m_j$, respectively. Define, further, local variables $\mathbf{v}_j = (\mathbf{s}_j^T, \mathbf{b}_j^T)^T$. Then, we can replace the constraint in (9) with $\sum_{j=1}^{\bar{n}} \mathbf{A}_j\mathbf{v}_j = \mathbf{0}_m$.

The SI problem (9) can now be solved using the ADAL algorithm [10]. Specifically, define the local augmented Lagrangians for agent j as

$$\begin{aligned} \mathcal{L}_c^j(\mathbf{v}_j,\mathbf{v}^k,\boldsymbol{\lambda}) = &\frac{1}{2}\left\|\mathbf{y}_j - \mathbf{b}_j\right\|^2 + \frac{\tau}{2}\left\|\mathbf{W}_j\mathbf{s}_j\right\|^2 + \boldsymbol{\lambda}^T(\mathbf{A}_j\mathbf{v}_j) \\ &+ \frac{c}{2}\left\|\mathbf{A}_j\mathbf{v}_j + \boldsymbol{\alpha}_j^k\right\|^2, \end{aligned} \tag{11}$$

where $c > 0$ is the penalty parameter, $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the Lagrange multiplier corresponding to the constraint, and $\boldsymbol{\alpha}_j^k = \sum_{i=1,i\neq j}^{\bar{n}} \mathbf{A}_i \mathbf{v}_i^k$ is a constant in which $\mathbf{v}_i^k$ corresponds to the latest estimation of the primal variable from agent i. Using $\mathcal{L}_c^j(\cdot)$, the iteration $k+1$ of the ADAL algorithm is given by

$$\hat{\mathbf{v}}_j^k = \underset{\mathbf{v}_j}{\operatorname{argmin}}\, \mathcal{L}_c^j(\mathbf{v}_j, \mathbf{v}^k, \boldsymbol{\lambda}^k), \tag{12a}$$

$$\mathbf{v}_j^{k+1} = \mathbf{v}_j^k + \rho\,(\hat{\mathbf{v}}_j^k - \mathbf{v}_j^k), \tag{12b}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + c\,\rho \sum_{j=1}^{\bar{n}} \mathbf{A}_j \mathbf{v}_j^{k+1}, \tag{12c}$$

where $\rho > 0$ is the primal step-size.

In (12), it is possible to derive the primal update (12a) in closed-form using (11). Specifically, setting $\nabla \mathcal{L}_c^j = 0$ we get

$$\hat{\mathbf{s}}_j^k = (\tau\, \mathbf{W}_j^2 + c\, \mathbf{X}_j^T \mathbf{X}_j - \frac{c^2}{1+c} \mathbf{X}_{jj}^T \mathbf{X}_{jj})^{-1} \\ [-\mathbf{X}_j^T (\boldsymbol{\lambda}^k + c\,\boldsymbol{\alpha}_j) + \frac{c}{1+c} \mathbf{X}_{jj}^T \boldsymbol{\beta}_j^k], \tag{13a}$$

$$\hat{\mathbf{b}}_j^k = \frac{1}{1+c}(\boldsymbol{\beta}_j^k + \mathbf{X}_{jj}\, \hat{\mathbf{s}}_j^k), \tag{13b}$$

where $\mathbf{X}_j$ is the column j and $\mathbf{X}_{jj}$ is the block in row and column j of the design matrix (8). The vector $\boldsymbol{\beta}_j^k \in \mathbb{R}^{m_j}$ is defined as $\boldsymbol{\beta}_j^k = \mathbf{y}_j - \boldsymbol{\lambda}_j^k - c\,\boldsymbol{\alpha}_{jj}^k$, where $\boldsymbol{\lambda}_j^k$ and $\boldsymbol{\alpha}_{jj}^k$ are the components of $\boldsymbol{\lambda}^k$ and $\boldsymbol{\alpha}_j^k$ corresponding to the measurements of agent j. Define $\overline{\mathbf{X}}_j \in \mathbb{R}^{m \times n_j}$ so that $\overline{\mathbf{X}}_j^T \overline{\mathbf{X}}_j = c\, \mathbf{X}_j^T \mathbf{X}_j - c^2/(1+c) \mathbf{X}_{jj}^T \mathbf{X}_{jj}$. Then, using the matrix inversion formula

$$(\tau\, \mathbf{W}_j^2 + \overline{\mathbf{X}}_j^T \overline{\mathbf{X}}_j)^{-1} = \\ \frac{1}{\tau} \mathbf{W}_j^{-2} \left[\mathbf{I}_{n_j} - \overline{\mathbf{X}}_j^T (\tau\, \mathbf{I}_m + \overline{\mathbf{X}}_j \mathbf{W}_j^{-2} \overline{\mathbf{X}}_j^T)^{-1} \overline{\mathbf{X}}_j \mathbf{W}_j^{-2} \right]. \tag{14}$$

Substituting (14) in (13a) completes the derivation of the closed-form solution for the primal optimization problem (12a). The proposed Distributed Source Identification (DSI) algorithm for agent j is summarized in Algorithm 1. Since the SI objective is strictly convex, selecting the primal step-size such that $0 < \rho < 1/\bar{n}$, guarantees the convergence of Algorithm 1 to the unique optimal source vector $\mathbf{s}^*$; see [10]. More precisely, $\mathbf{s}_j^k$ converges to the sub-vector $\mathbf{s}_j^*$ corresponding to agent j.

C. Complexity Analysis

In this section we discuss the communication cost and time complexity of Algorithm 1. Specifically, there is one communication round with every iteration of the algorithm, as described in line 8. Note that agent j only needs the sum $\boldsymbol{\alpha}_j^{k+1} = \sum_{i=1,i\neq j}^{\bar{n}} \mathbf{A}_i \mathbf{v}_i^{k+1}$ and not the individual terms. Thus, given a routing plan that determines the communication sequence in the network, the agents can communicate one vector of length m instead of individual $\mathbf{A}_i \mathbf{v}_i^{k+1}$ vectors. In worst case, agent j receives partial sums from all of its neighbors, adds its own value $\mathbf{A}_j \mathbf{v}_j^{k+1}$, and communicates back to all of them with cost of $2\,|\mathcal{N}_j|\, m$, where $|\mathcal{N}_j| \leq \bar{n} - 1$ is the number of its neighbors. Thus, the communication cost is bounded by $O(\bar{n}m)$ at each iteration for agent j and does not depend on the FE mesh size n. This is in contrast to FE-based Domain Decomposition methods that require communication of boundary nodes between neighbors [16] and other optimization-based methods that require consensus over the global source vector [17].

Algorithm 1 Distributed Source Identification Algorithm

Require: The subdomain Ω_j and the set of neighbors $\mathcal{N}_j$;
Require: The matrix $\mathbf{M}_j$ given in (3), the POD basis functions $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_N]$, and the weight matrix $\mathbf{W}_j$;
Require: The penalty parameter c, the regularization parameter τ, and the threshold η of stopping criterion (15);
1: Collect m_j measurements in $\bar{\Omega}_j$ and broadcast the measurement locations to agents in $\mathcal{N}_j$;
2: Receive measurement locations of all agents and build the matrix $\mathbf{A}_j$ as in (10);
3: Initialize $\mathbf{v}_j^0 \in \mathbb{R}^{n_j + m_j}$ and $\boldsymbol{\lambda}^0 = \mathbf{0}_m$;
4: **while** the algorithm has not converged **do**
5: Compute $\hat{\mathbf{s}}_j^k$ and $\hat{\mathbf{b}}_j^k$ from (13);
6: Update the primal variable $\mathbf{v}_j^{k+1}$ using (12b);
7: Check the stopping criterion (15);
8: Broadcast $\mathbf{A}_j \mathbf{v}_j^{k+1}$ to agents in $\mathcal{N}_j$ and receive $\boldsymbol{\alpha}_j^{k+1} = \sum_{i=1,i\neq j}^{\bar{n}} \mathbf{A}_i \mathbf{v}_i^{k+1}$;
9: Update the dual variable $\boldsymbol{\lambda}^{k+1}$ using (12c);
10: $k \leftarrow k+1$;
11: **end while**

Next, we consider the time complexity of Algorithm 1. Updates of $\hat{\mathbf{s}}_j^k$ in line 5 require computation of the matrix $\overline{\mathbf{X}}_j \mathbf{W}_j^{-2} \overline{\mathbf{X}}_j^T$, given in the right-hand-side of (14). This cost depends on the value of weight matrix. For the unweighted case where $\mathbf{W}_j = \mathbf{I}_{n_j}$, it is given by $O(m^2 n_j)$ and for diagonal weight matrix it is given by $O(mn_j + m^2 n_j)$. Note that this matrix is constant across iterations and can be stored. The cost of computing the vector in the right-hand-side of (13a), is bounded by $O(2n_j(m+m_j))$ and the cost of solving the resultant linear system of equations is bounded by $O(m^3)$. The cost of updates of $\hat{\mathbf{b}}_j^k$ in line 5, given by (13b), is bounded by $O(2m_j n_j)$. Finally, the cost of computing the vector $\mathbf{A}_j \mathbf{v}_j^{k+1}$ in line 8 is bounded by $O(2mn_j)$. Thus, the time complexity of each iteration is bounded by $O(8mn_j + m^3)$ and increases linearly with FE mesh size n_j. This allows our algorithm to handle large scale problems efficiently.

D. Online Source Identification

So far we have assumed that the measurements are collected prior to the implementation of the DSI Algorithm 1. Nevertheless, it is often desirable to estimate the source vector in an online fashion, so that the agents solve the SI problem as they move in their corresponding subdomains and collect more measurements. Assume that agent j has collected m_j^{init} measurements prior to commencing iterations of Algorithm 1 and while the algorithm is running, it collects an additional measurement every l_j iterations, until all remaining $m_j - m_j^{\text{init}}$ measurements have been collected. Every time an agent i takes a new measurement, it broadcasts the measurement location and each agent j appends a new row to the block $\mathbf{X}_{ij},\ \forall\, j \in \mathcal{V} \setminus \{i\}$. This modified version of the algorithm

converges to the unique optimal solution $\mathbf{s}^*$ of the SI problem (7), as long as the number of measurements m_j of all agents are finite. Note that modifying design matrices $\mathbf{X}_j$ of agents requires re-computation of the matrix in right-hand-side of (14). In the worst case, this matrix needs to be computed at every iteration increasing the time complexity to $O(m^2 n_j)$.

E. Reweighting and Stopping Criterion

Given that the source function s in the AD-PDE (1) is compactly supported, the discretized source vector $\mathbf{s}$ in (2) is sparse. To exploit this *a priori* knowledge, after solving the unweighted SI problem with $\mathbf{W}_j = \mathbf{I}_{n_j}$, we can adjust the weights and resolve the problem one more time using the solution of the unweighted problem as an initial condition. Specifically, after solving the unweighted problem, agent j updates its weights as $[\mathbf{w}_j]_i = 1/(\xi + |[\mathbf{s}_j]_i|)$, where $[\cdot]_i$ denotes the i-th entry of the vector, $0 < \xi \ll 1$ is a small positive number, and $|\cdot|$ is the absolute value. This process forces small entries of the solution toward zero and introduces an outer loop in Algorithm 1 that, in practice only needs to be run for a handful of iterations [18].

To stop the iterations of Algorithm 1, agent j monitors the evolution of its source estimates. Specifically, given the threshold η, it checks the following condition

$$\left\|\mathbf{s}_j^k - \mathbf{s}_j^{k-1}\right\| \leq \eta. \tag{15}$$

IV. Simulations

In this section we provide numerical simulations to demonstrate the performance of DSI Algorithm 1. The algorithm is implemented in MATLAB on a laptop with 2.2 GHz INTEL core i7 processor and 8 GB of memory. We define the identification error between the optimal source vector $\mathbf{s}^*$, obtained from the solution of (7), and the true source vector $\bar{\mathbf{s}}$ as err $= \|\mathbf{s}^* - \bar{\mathbf{s}}\| \,/\, \|\bar{\mathbf{s}}\|$. We also define the signal to noise ratio in dBs as SNR $= 20 \log_{10} (\|\mathbf{y}\| \,/\, \|\boldsymbol{\epsilon}\|)$, where $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is the noise vector corresponding to network-wide measurements.

We solve the SI Problem II.1 for a 2D model in a domain that resembles the ground floor of the Hudson Engineering hall at Duke University, cf. Figure 1. We assume that the flow enters the hall through two side doors and we use an in-house FE code with mesh size of $n = 15034$ nodes to solve for the fluid dynamics equations. The FE-mesh is generated in CUBIT. We set the diffusivity to be $D = 2 \times 10^{-3}$ which results in Peclet number $Pe = 2.5$. This number is a measure of the relative importance of advection to diffusion. For $Pe = 2.5$ both advection and diffusion are important means of transport. The FE model (2) is generated using an in-house FE code based on the DiffPack C++ library [19]. We consider two rectangular sources in subdomains (2) and (6) with intensities 1 and 1.2 respectively. The contours of concentration are given in Figure 1. Using the POD method with $N = 291$ basis functions, we obtain the reduced order model given in (3).

We utilize $\bar{n} = 6$ agents to solve the problem. The network topology, communication graph, and the subdomains $\bar{\Omega}_j$ that the agents take their measurements from are shown in Figure 1. We assume a heterogeneous network where the agents

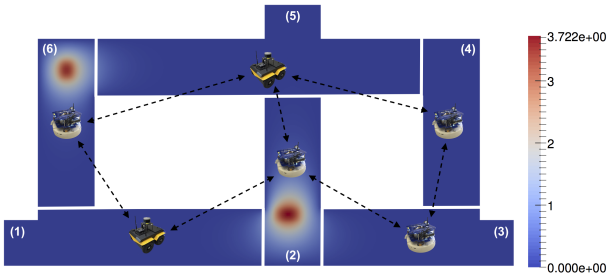

Fig. 1: Concentration contours and the network of mobile sensors. The communication graph is shown by black arrows while the white lines indicate the subdomains that the agents collect their measurements from them.

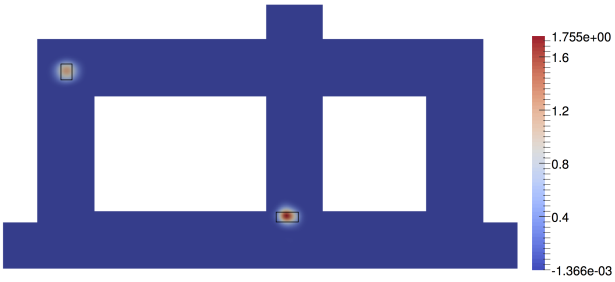

Fig. 2: Estimated source contours for noisy measurements with SNR $= 19.10$. The boundaries of the true sources are overlaid on the contours by black lines.

$j \in I_1 = \{1, 5\}$ have higher computational specifications than agents $j \in I_2 = \{2, 3, 4, 6\}$. We decompose the unknown source vector such that $n_j = 4001$ for $j \in I_1$ and $n_j = 1758$ for $j \in I_2$. The agents use a grid of equidistant measurements in each subdomain where we set $m_j = 6 \times 12 = 72$ for $j \in I_1$ and $m_j = 5 \times 10 = 50$ for $j \in I_2$ since subdomains $\bar{\Omega}_1$ and $\bar{\Omega}_5$ are larger.

A. Advection-Diffusion Source Identification

Using the DSI Algorithm 1 with penalty parameter $c = 10^{-5}$ and regularization parameter $\tau = 10^{-6}$, the SI problem is solved in 107 iterations and 6.49 sec. The final error is err $= 65.55\%$. Adding noise with $\sigma_1 = 0.05$ and $\sigma_2 = 0.05$ to the measurements results in SNR $= 19.10$ for which the algorithm converges in 81 iterations and 3.96 sec with err $= 74.21\%$. Using the unweighted solution to update the weight matrices $\mathbf{W}_j$ and solving the problem one more time, the error reduces to err $= 54.69\%$ while it takes 5.86 sec and 87 iterations for the algorithm to solve the reweighted problem. We also solve the problem using the online version of the algorithm, cf. Section III-D. Particularly, we set $m_j^{\text{init}} = 15$ and $l_j = 2$ for $j \in I_1$ and $m_j^{\text{init}} = 20$ and $l_j = 3$ for $j \in I_2$. The algorithm solves the last instance of the problem with err $= 54.71\%$ and it takes 18.36 sec to solve the unweighted problem followed by a reweighted problem.

The final estimated source for the noisy measurements is depicted in Figure 2. Note that the localization of the sources is accurate, however, the boundaries of the sources are not recovered. This is expected for ℓ_2 regularization used here. The evolution of the identification error is given in Figure 3, where the agents solve the unweighted SI problem until convergence and then solve one additional instance of the reweighted SI

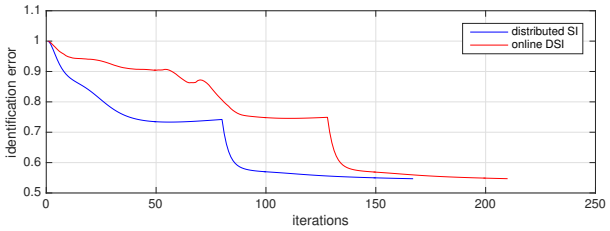

Fig. 3: Evolution of the identification error as a function of the iterations.

problem, cf. Section III-E. The sharp decreases in the error correspond to the update of weights. Moreover, the fluctuations at the early iterations of the online implementation happen as a result of adding new measurements while solving the problem.

B. Comparison with ADMM Formulation

In Table I, we compare our method to an alternative formulation using Alternating Direction Method of Multipliers (ADMM) algorithm [17]. The iterations of ADMM algorithm can be obtained using a derivation similar to Section III and despite the similarities, practical differences are notable. Our formulation decomposes both the domain and task of collecting the measurements, cf. Section III-A, whereas ADMM formulation only decomposes the measurement collection. This means greater flexibility for DSI algorithm to balance the load through domain decomposition instead of changing the number of measurements that the agent collects. Our formulation enforces consensus over measurements with communication cost of $O(\bar{n}m)$ whereas the ADMM formulation enforces consensus over unknown source vector with cost of $O(\bar{n}n)$ where $m \ll n$; cf. Table I. Time complexity of our method is bounded by $O(8mn_j + m^3)$ whereas ADMM algorithm is bounded by $O(2nm_j + m_j^3)$. This means less computation per iteration for ADMM formulation at the expense of considerably larger communication cost. Comparing cases (i) and (ii) in Table I, we observe that the time complexity of DSI algorithm is independent of m_j and increases with m while ADMM algorithm is independent of m and increases with m_j. Comparing cases (ii) and (iii), we observe a linear increase in time of both algorithms as a function of mesh size.

TABLE I: Comparison with ADMM formulation

case			(i)		(ii)		(iii)	
n			15034		15034		59020	
n_j		I_1	4001		4001		9838	
		I_2	1758		1758		9836	
m			300		344		344	
m_j		I_1	50		72		72	
		I_2	50		50		50	
comm. (KB)			14.4	361	16.5	361	16.5	1420
time (ms)	nw	I_1	11	1.5	12	2.2	28	9.0
		I_2	4.5	1.5	6.1	1.5	26	6.2
	w	I_1	11	1.6	13	2.2	29	7.8
		I_2	4.4	1.6	6.5	1.6	27	5.6
	total	I_1	2561	307.0	2764	448.5	6139	2031
		I_2	1045	307.2	1439	320.3	5763	1439
iterations			234	192	227	197	215	246
error (%)			51.06	52.31	50.97	52.22	54.43	58.16
algorithm			DSI	ADMM	DSI	ADMM	DSI	ADMM

V. Conclusion

In this paper we proposed a distributed algorithm to solve the Source Identification (SI) problem for steady-state Advection-Diffusion (AD) transport model using a network of mobile sensors while achieving perfect load balancing among them. We showed that the communication cost of our method does not scale with FE mesh size while its time complexity scales linearly, enabling it to solve large-scale problems efficiently. We implemented our algorithm in an online way and compared it to an alternative formulation.

References

[1] L. Marques, U. Nunes, and A. T. de Almeida, "Olfaction-based mobile robot navigation," *Thin solid films*, vol. 418, no. 1, pp. 51–58, 2002.

[2] D. Zarzhitsky, D. F. Spears, and W. M. Spears, "Distributed robotics approach to chemical plume tracing," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 4034–4039, 2005.

[3] J. M. Soares, et al., "A distributed formation-based odor source localization algorithm-design, implementation, and wind tunnel evaluation," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1830–1836, IEEE, 2015.

[4] J. Matthes, L. Groll, and H. B. Keller, "Source localization by spatially distributed electronic noses for advection and diffusion," *IEEE Transactions on Signal Processing*, vol. 53, pp. 1711–1719, May 2005.

[5] J. Weimer, B. Sinopoli, and B. H. Krogh, "Multiple source detection and localization in advection-diffusion processes using wireless sensor networks," in *IEEE Real-Time Systems Symposium*, pp. 333–342, 2009.

[6] V. Akçelik, et al., "A variational finite element method for source inversion for convective-diffusive transport," *Finite Elements in Analysis and Design*, vol. 39, pp. 683–705, May 2003.

[7] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Model-based sparse source identification," in *Proceedings of American Control Conference*, pp. 1818–1823, July 2015.

[8] R. Khodayi-mehr, W. Aquino, and M. M. Zavlanos, "Nonlinear reduced order source identification," in *Proceedings of American Control Conference*, pp. 6302–6307, July 2016.

[9] J. A. Atwell and B. B. King, "Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations," *Mathematical and Computer Modeling*, vol. 33, no. 1, pp. 1–19, 2001.

[10] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos, "An augmented lagrangian method for distributed optimization," *Mathematical Programming*, vol. 152, no. 1-2, pp. 405–434, 2015.

[11] S. A. Socolofsky and G. H. Jirka, *Special topics in mixing and transport processes in the environment*. Coastal and Ocean Engineering Division, Texas A&M University, 5 ed., 2005.

[12] B. D. Reddy, *Introductory functional analysis: with applications to boundary value problems and finite elements*, vol. 27. Springer, 2013.

[13] D. L. Logan, *A First Course in the Finite Element Method*. Nelson, 2007.

[14] A. C. Antoulas, *Approximation of large-scale dynamical systems*, vol. 6. SIAM, 2005.

[15] P. C. Hansen, *Discrete Inverse Problems*. SIAM, 2010.

[16] A. Toselli and O. Widlund, *Domain decomposition methods: algorithms and theory*, vol. 34. Springer, 2005.

[17] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[18] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 6, pp. 877–905, 2008.

[19] H. Langtangen, *Computational Partial Differential Equations - Numerical Methods and Diffpack Programming*. Springer, 2 ed., 2003.