Cyber-Physical Systems
Vol. 00, No. 00, October 2017, 1–16

Cooperative Identification of Advection-Diffusion Processes with Spatially Varying Coefficients based on a Multi-model Structure

Jie You[a] and Wencen Wu[a*]

[a]*Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA*

(*released September 2016*)

This work develops a practical real-time identification strategy for advection-diffusion processes with spatial varying coefficients. A novel multi-model structure that represents the processes in practical applications is proposed. The multi-model structure is parameterized as blended linear PDE models. Both the offline centralized and the online distributed identification algorithms using mobile sensor networks are proposed. Trajectories for mobile sensor networks in the advection-diffusion field with spatially varying parameters is found, and distributed control laws to guide the mobile sensor networks to follow the trajectories while keeping desired formations are designed. The effectiveness of this proposed method is demonstrated in simulations.

Keywords: Real-time identification ; Advection-diffusion processes; Multi-model structure

1. Introduction

In systems with thermal, fluid, or chemical reacting dynamics, parabolic partial differential equations (PDEs) are often used by researchers to describe the system dynamics. In such equations, physical parameters such as the diffusion coefficient are often unknown. Thus a need exists for developing practical and efficient approaches to identify uncertain systems so that predictions of the system states can be made. Many results have shown that the state prediction of PDEs have played key roles in services such as weather forecasting, earthquake early detection, and disaster recovery (Uciński 2004). For instance, a typical advection-diffusion PDE process is the dispersion of oils from a leakage source into an ambient environment, which results in a plume. The timely prediction of the propagation phenomenon of oils can be used in tracking the source and and implementing search/rescue missions (Uciński 2004).

Most existing approaches have only dealt with the identification problem using static sensor networks (Uciński 2004, Gay and Ray 1995, Krstic and Symshlyaev 2008, Omatu and Matumoto

[*]Corresponding author. Email: wuw8@rpi.edu

ISSN: 2333-5785 print/ISSN 2333-5777 online

DOI: 10.1080/0305215X.YYYY.CATSid
http://www.informaworld.com

1991). However, in missions of modeling a relatively large region, it is often impractical to use static sensor networks, which require high cost of installing enough static sensors to ensure the coverage of the entire field (Mourikis and Roumeliotis 2006, Burgard *et al.* 2005). Instead, a preferable choice is mobile sensor networks (MSNs). MSNs are composed of robotic agents, each of which has computational, communication, sensing, and locomotive capabilities (Uciński 2004). Over the past few years, there has been an increasing interest in the use of mobile sensor networks to detect, monitor, and model the systems that can be described by PDEs (Demetriou 2010, Tang and Ozguner 2005, Campbell and Whitacre 2007, Demetriou and Hussein 2009, You and Wu 2017, You *et al.* 2016). The literature Demetriou and Hussein (2009) has illustrated that a set of mobile sensors can achieve better state estimation performance than a set of immobile sensors. In You and Wu (2017), a distributed online passive identifier is proposed to estimate the constant diffusion coefficient of the advection-diffusion PDE using data collected by a mobile sensor network moving in the field. By combining cooperative filtering and RLS method, in You *et al.* (2016), parameter identification for diffusion processes is proposed using mobile sensor networks. There exist several literatures regarding the state estimation of PDE (see Demetriou (2010), Demetriou *et al.* (2014) and references therein).

One important aspect of PDE modeling is how to deal with systems with the spatially varying plant parameters. In studying the advection-diffusion PDEs with spatially varying coefficients, one direct approach is to parameterize the parameters using linear or nonlinear functions of the position variable (Uciński 2004, Uciński and Chen 2005). Because complex nonlinear functions often appear in the coefficients of the PDE model, intensive computations are required to solve the PDE in the entire spatial domain using, for example, finite element methods. On the other hand, indirect approaches exist that employ a linear combination of basis functions to approximate the advection-diffusion PDE by using the Garlerkin weighted-residual method (Li and Qi 2010). Most of these works suppose the parameters are known or can be estimated in an offline scheme. The main goal of these existing approaches is focused on how to derive an optimal sensor trajectory or location of a mobile sensor network to improve the performance of state estimation. However, in many realistic scenarios, timely state estimation and prediction of the processes are required while largely untreated in the existing literature. For example, in chemical plume tracking, mobile sensors have no prior knowledge of the diffusion coefficient of the diffusion process. In this case, it's preferable that the mobile sensors can estimate the unknown parameters recursively to provide real-time information while exploring the field.

To realize timely parameter identification and state estimation, this paper aims at developing a practical and effective model structure to represent the advection-diffusion process with spatially varying diffusion coefficient over a broad operating regime. Motivated by practical identification practice (Boukhris and Mourot 1999, Murray-Smith and Johansen 1997), we propose a novel multi-model structure to approximate the general advection-diffusion equation. Based on this multi-model structure, both the offline centralized and the online distributed parameter estimation algorithms are developed by using data collected by a group of coordinated mobile sensors moving in the field. In addition, the trajectory of the mobile sensor network is analyzed by maximizing the Fisher information matrix and distributed control laws to control the mobile sensor network to estimate and follow the trajectory while keeping a desired formation are designed.

The rest of the paper is organized as follows. Section II presents the problem formulation. Section III discusses the parameterization of spatially varying PDE models. Section IV shows the identification method. Section V designs the optimal trajectory and distributed control law for the mobile sensor network. Section VI introduces the simulation result, and Section VII draws conclusions and discusses future work.

2. Problem formulation

2.1. *Advection-diffusion PDE*

Consider the following two-dimensional (2D) advection-diffusion process with an unknown spatially varying diffusion coefficient defined on a domain $\Omega = [0, L_x] \times [0, L_y] \in \mathbb{R}^2$:

$$\frac{\partial z(r,t)}{\partial t} + v^T \nabla z(r,t) = \theta(r) \Delta z(r,t), \ r \subseteq \Omega, \tag{1}$$

where $r = [x, y]$ is the position, $z(r,t)$ is the concentration function, $\theta(r)$ is a smooth function of the position r, and v is a constant vector representing the flow velocity, which is supposed to be known through measurements or computation, e.g., solutions of the Saint-Venant equations (Chang *et al.* 2014). $\nabla z(r,t)$ is the spatial gradient of $z(r,t)$, and Δ represents the Laplacian operator. We assume Dirichlet boundary conditions without restriction on the boundary $\partial\Omega$,

$$z(r,t) = z_{bc}(r,t), \ r \subseteq \partial\Omega. \tag{2}$$

2.2. *Sensor dynamics*

Consider a formation of N coordinated sensing agents moving in the field, each of which carries a sensor that takes point measurements of the field $z(r,t)$. Same as in You and Wu (2017) You *et al.* (2016), we assume that the dynamic of the agents is described by a single integrator

$$\dot{r}_i(t) = u_i(t), i = 1, 2, ..., N. \tag{3}$$

where $r_i(t) \subseteq \mathbb{R}^2$ is the position, and $u_i(t) \subseteq \mathbb{R}^2$ is the velocity of the ith agent, respectively. As the agent moves in a field, the position $r_i(t)$ is a function of the time t. For simplicity, we drop the variable t in $r_i(t)$ hereafter. We make the following assumption for the sensing agents.

Assumption 2.1 The agent's sensors can obtain its position r_i, the measurement of concentration value $z(r_i,t)$, and the flow velocity $v(r_i,t)$.

2.3. *Communication*

Based on the agents' information exchange process, we define two different communication topologies: centralized and distributed communication, which are described in the following, respectively.

2.3.1. *Centralized communication*

In a centralized mobile sensor network, we assume that there is a centralized fusion center to exchange information with each agent. So the communication graph has an all-to-all connection, which allows communication and information exchange between any pair of agents.

2.3.2. *Distributed communication*

In a distributed mobile sensor network, we assume that every sensing agent can only communicate and exchange information with its neighboring agents within a limited communication range. Hence, it does not have the global knowledge of the network topology. If we consider the agents as nodes and the communication links as edges, then, the interconnection topology of the mobile sensor network can be specified by an undirected graph $G = (V, E)$ with the adjacency matrix $A = [a_{ij}]$. Let $V = \{1, ..., N\}$ be the node set and $E \subseteq V \times V$ be the edge set. Also let $\mathbb{N}_i = \{i \subseteq V : a_{ij} \neq 0\}$ denote the neighbor set of the ith robot. We have the following assumption for the graph G.

Assumption 2.2 The graph G is connected for all $t > 0$.

The problem is formulated as:

(1) Model the advection-diffusion field using a practical and effective model under Assumption 2.1.
(2) Based on the model, develop an parameter identification algorithm that estimates the unknown parameter.
(3) Utilizing the identified model, provide real-time state estimation of the advection-diffusion field.

Rather than parameterizing the spatially varying diffusion coefficient $\theta(r)$ using the linear or nonlinear functions of the position variable (Uciński 2004, Uciński and Chen 2005), which may lead to intensive computations and differentiability issues in the deviation of online trajectory for mobile sensor networks, we develop a practical and effective approximation model by interpolating several linear PDE models with constant diffusion coefficients. Based on this multi-model structure, corresponding parameter identification and state estimation algorithm will be developed using data collected by a group of coordinated mobile sensors moving in the field.

3. Spatially Varying PDE parameterizations

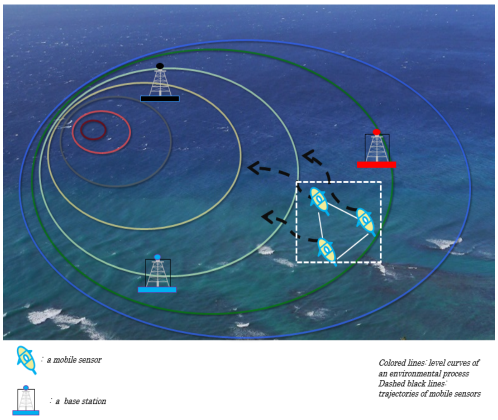

: a mobile sensor
: a base station
Colored lines: level curves of an environmental process
Dashed black lines: trajectories of mobile sensors

Figure 1. The illustration of monitoring a practical advection-diffusion process with spatially varying diffusion coefficient using a mobile sensor network.

To monitor some environmental processes, in many real fields, some static base stations are already installed at selected locations. For example, in oceanography, some buoys are installed to test chemical or biological contaminants transporting through subsurface aquifer, which are illustrated in Fig. 1. When we have no prior information, we also can use K-means clustering method to compute the positions of the stations. We first divide the whole area Ω into $n \times n$ grid sizes. Then the K-means clustering method is applied to partition the grid map into G disjoint subset S_j by minimizing the following cost function,

$$J = \sum_{j=1}^{G} \sum_{r_i \in S_j} ||r_i - c_j||^2. \tag{4}$$

At each static base station, there exist several static sensors collecting data. By using these prior collected data, we can first identify and obtain a PDE model with a constant diffusion coefficient, which will be referred to as the local linear PDE model. The local linear PDE model is a good representation of the environmental process in the vicinity of the corresponding static base station (Li and Qi 2010).

Motivated by the above practical setting and the multi-model structure (Boukhris and Mourot 1999, Murray-Smith and Johansen 1997), in this paper, we propose a multi-model structure to approximate the advection-diffusion PDE using blended linear PDE models. The basic principle of this approach is to first identify several local linear PDE models at the fixed setting positions (the positions of base stations). Then we navigate a group of mobile agents moving in the field to provide sampling coverage over a large area. Using data collected by the mobile sensor network, the global PDE model is obtained by interpolation using certain weighting functions. Compared with the local linear PDE models, the global PDE model represents the process in the whole region.

We assume that there are p static base stations at locations:

$$(x_1, y_1), (x_2, y_2), ..., (x_p, y_p) \subseteq \Omega. \tag{5}$$

In the neighborhood of each base station, we describe the process using a local linear PDE model with a constant diffusion coefficient,

$$\begin{aligned} \frac{\partial \hat{z}^h(r,t)}{\partial t} + v^T \nabla \hat{z}^h(r,t) &= \theta^h \Delta \hat{z}^h(r,t), r \in \Gamma(t), \\ \hat{z}^h(r, t-T) &= z(r, t-T), r \in \Gamma(t). \end{aligned} \tag{6}$$

where $h = 1, 2, ...p$, T is the sampling interval, $\hat{z}^h(r,t)$ denotes the one step ahead prediction using the local linear PDE model with the constant diffusion coefficient θ^h, $\Gamma(t) \subseteq \Omega$ denotes the region covered by the mobile sensor network with N agents. It should be noted that $\Gamma(t)$ is relatively small compared to Ω.

Then the global PDE model can be obtained by interpolating the local linear PDE models in (6).

$$\hat{z}(r,t) = \sum_{h=1}^{p} \eta^h(r) \hat{z}^h(r,t), r \in \Gamma(t), \tag{7}$$

where $\eta^h(r), (h = 1, 2, ...p)$, which are static functions of the position r, are weights of the local linear PDE models.

Remark 1 The global model in (7) will be referred to as the multi-model structure in the following context. It also falls into the multi-model structure (Boukhris and Mourot 1999), where the weightings are assumed to be functions of so-called characteristic variables or scheduling variables, which need to be selected by the users. Assume that the model parameters such as the diffusion coefficient varies as monotone functions of r between each two neighboring static base stations. There are several advantages using the weighing functions and local linear PDE models: (1) the multi-model structure (7) considerably simplifies the task of model identification; (2) they can model nonlinearities of spatially varying diffusion coefficients; (3) they can reduce computational cost in the derivation of online trajectory for MSN, which will be illustrated in Section 5. The capability of this model will be demonstrated in the case study.

To obtain the multi-model structure (7), proper weighting functions should be selected for the interpolation, which posts an impart on the accuracy of the global model. Several common weighting functions are available in the literature, for example, linear weight function, polynomial function, cubic spline function, and Gaussian weight function (Huang *et al.* 2012). A preferable choice of determining the model weights is Gaussian function, which can achieve a good fitting performance. The representation of Gaussian weighting functions can be written as

$$\eta^h(r) = \frac{\alpha^h(r)}{\sum_{m=1}^{p} \alpha^m(r)}, \tag{8}$$

where

$$\alpha^h(r) = \alpha^h(x,y) = exp\left[-\frac{1}{2}\left((\frac{x-x_h}{\sigma_h^1})^2 + (\frac{y-y_h}{\sigma_h^2})^2\right)\right]. \tag{9}$$

Note that the Gaussian weights are smooth and differentiable at each position. The model weights in (8) are normalized and in the range of zero to one to avoid negative values.

Obviously, only two parameters (σ_h^1, σ_h^2) need to be estimated for each weighting function. The parameter vector to be identified for all weighting functions can be defined as

$$\Theta = [\sigma_1^1, \sigma_1^2, ..., \sigma_h^1, \sigma_h^2, ..., \sigma_p^1, \sigma_p^2]. \tag{10}$$

4. Identification of the multi-model structure

4.1. *Local linear PDE model identification at each setting position*

In order to identify the multi-model structure, local linear PDE models at each fixed setting position should be determined first. If there exist some base stations, one can choose these positions as the setting positions. Using the data collected in the vicinity of each setting position, all the parameters θ^h can be obtained by using the PDE identification approaches based on static sensor networks, for example, Lyapunov method and certainty equivalence approaches with passive and swapping identifiers (Symshlyaev and Krstic 2007).

To substantiate the feasibility of the proposed approach, we further consider the situation that there are no base stations. In this case, additional identification tests are required, which means we should navigate a group of mobile agents around setting positions to collect data. Then one can identify the local linear PDE models based on the methods using mobile sensors networks. Several choices are available in the literature, e.g., online passive identifier (You and Wu 2017), and the cooperative filtering method (You *et al.* 2016). In this paper, we adopt the distributed online passive identifier that we developed in You and Wu (2017) to identify the local linear PDE model using the mobile sensor networks. We briefly summarise the method using notations in this paper.

Within a distributed mobile sensor network, consider the ith agent and its neighbors as a group. Denote $\Gamma_i(t) \in \Gamma(t)$ as the region bounded by the ith agent and its neighbors. The boundary of the region $\Gamma_i(t)$ is $\partial\Gamma_i(t)$. As the mobile sensor network moves along a certain trajectory, $\Gamma(t)$ as well as $\Gamma_i(t)$ move correspondingly. For notational convenience, we drop the variable t in $\Gamma(t)$ and $\Gamma_i(t)$ hereafter.

Let us define the following notations for the 2D passive identifier in this section:

$$\begin{aligned} &\|\nabla z_{\Gamma_i}\|^2 = \int\int_{\Gamma_i}\left((\frac{\partial z(r,t)}{\partial x})^2 + (\frac{\partial z(r,t)}{\partial y})^2\right) d\Gamma_i \\ &\triangleq \int\int_{\Gamma_i} \nabla z(r,t)\cdot\nabla z(r,t) d\Gamma_i, \end{aligned} \tag{11}$$

Consider the following passive identifier computed by the ith agent with the associated Γ_i, $i = 1,2...,N$:

$$\frac{\partial \tilde{z}_i^h(r,t)}{\partial t} = \hat{\theta}_i^h \Delta \tilde{z}_i^h(r,t) + \gamma_0^2(z(r,t) - \tilde{z}_i^h(r,t))\|\nabla z_{\Gamma_i}\|^2, r \in \Gamma_i, \tag{12}$$

$$\tilde{z}_i^h(r,t) = z(r,t), r \in \partial\Gamma_i, \tag{13}$$

with the update law

$$\begin{aligned} \dot{\hat{\theta}}_i^h &= \gamma_1 \cdot \beta_i \cdot Proj\{\int\int_{\Gamma_i} \nabla z(r,t)\cdot\nabla(z(r,t) - \tilde{z}_i^h(r,t))d\Gamma_i\} \\ &+ \sum_{j\subseteq\mathbb{N}_i} W_{i,j}(\hat{\theta}_j^h - \hat{\theta}_i^h), \end{aligned} \tag{14}$$

where $h = 1,2,...p$, $\gamma_0{>}0$ and $\gamma_1{>}0$ are constants, $\beta_i \geq 1$ is the ratio between the area of Γ over the area of Γ_i, $\tilde{z}_i^h(r,t)$ represents the estimate of $z(r,t)$ by the ith passive identifer, and $\hat{\theta}_i^h$ is the estimated constant diffusion coefficient computed by the ith agent in the vicinity of the point $x = x_h$, $y = y_l$. $W_{i,j}$ are the weights in the Metropolis-Hastings model (Xiao *et al.* 2006) for the distributed estimation problem. Based on the received information from the neighbors, the weights can be calculated as

$$W_{i,j} = \begin{cases} \frac{1}{1+max\{d(i),d(j)\}} & \text{if } \{i,j\} \in E \text{ and } i \neq j \\ 1 - \sum\limits_{(i,h)\in E} W_{i,h} & \text{if } i = j \\ 0 & otherwise \end{cases}, \tag{15}$$

where $d(i)$ denotes the number of the ith robot's neighbors. Metropolis-Hastings model has the symmetric and doubly-stochastic properties, which is one of the widely used weighted adjacency matrices in distributed sensor fusion (Xiao *et al.* 2006).

In (14), the projection operator (Proj) is defined as:

$$\text{Proj}\{\rho\} = \begin{cases} 0 & \text{if } \hat{\theta}_i{=}\bar{\theta} \text{ and } \rho < 0 \\ \rho & \text{else} \end{cases}. \tag{16}$$

The projection operator is used to maintain the parabolic character of the PDE (1) by ensuring $\theta{>}\bar{\theta}{>}0$, where $\bar{\theta}$ is the lower bound of $\hat{\theta}_i$.

With the first term in (14), we aim to reduce the local prediction error. $\gamma_1 \cdot \beta_i > 0$ controls the local information fusion rate. The second term ensures global parameter consensus by summing over the values of the neighbors.

Remark 2 The passive identifier in (12) is implemented only based on information collected by a mobile sensor networks moving in the field. The inputs to the identifier (12) are $z(r,t)$, $\nabla z(r,t)$, $r \in \Gamma_i$, and the boundary condition $z(r,t)$, $r \in \partial\Gamma_i$. The inputs can be obtained from sensor measurements and the outputs of a distributed cooperative Kalman filter. The detail of the distributed cooperative Kalman filter can be found in our previous paper (You and Wu 2017, You *et al.* 2017).

Note that in a centralized network, there is a fusion center that collects and processes data from all the agents, therefore, the online passive identifier can be directly computed by the fusion center in a centralized way with Γ_i replaced by Γ.

We have proved the convergence of parameter estimation in (You and Wu 2017). After the identification procedure of the local linear PDE model using the data from the corresponding setting positions, $\hat{\theta}_i^h$ can reach consensus on the true parameter θ^h.

4.2. *Identification of multi-model structure with Gaussian weights*

In the previous section, we have shown how to obtain the local linear PDE models. After getting the local models, in this section, both the centralized and the distributed algorithms are presented to identify the parameter Θ in (10) of the multi-model structure.

4.2.1. *Offline Centralized Identification of multi-model structure*

To estimate Θ, nonlinear optimization algorithms, which can minimize the following output error loss function, are desired:

$$f(\Theta) = \sum_{t=1}^{t_f} \frac{1}{2}[\int_{\Gamma} z(r,t) - \sum_{h=1}^{p} \eta^h(r)\hat{z}^h(r,t)d\Gamma(t)]^2, r \subseteq \Gamma(t), \tag{17}$$

where t_f is the terminal time.

To solve the offline centralized optimization problem in (17), several nonlinear numerical optimization algorithms are available for this purpose, such as Gauss-Newton algorithm, gradient descent method, expectation-maximization algorithms, and Levenberg-Marquardt method (Dennis 1983, Ljung 1998, Hosseini *et al.* 2013). In this work, the proximal gradient descent algorithm is used (Hosseini *et al.* 2013). That is because it can be readily extended to the online distributed case, which will be illustrated in the next subsection. The iterative optimization flow is as follows.

Step 1: Initialize $\hat{\Theta}^0$ and iteration count $l = 0$.

Step 2: Calculate the simulated prediction output error as

$$\varepsilon(r,t,\hat{\Theta}^l) = z(r,t) - \sum_{h=1}^{p} \eta^h(r,\hat{\Theta}^l)\hat{z}^h(r,t), r \subseteq \Gamma. \tag{18}$$

Step 3: The output error loss function in (17) has the gradient:

$$f'(\hat{\Theta}^l) = \sum_{t=1}^{t_f} \int_{\Gamma} \Psi(r,t,\hat{\Theta}^l)\varepsilon(r,t,\hat{\Theta}^l)d\Gamma, \tag{19}$$

where $\Psi(r,t,\hat{\Theta}^l)$ is the gradient matrix of $\varepsilon(r,t,\hat{\Theta}^l)$ with respect to Θ, which can be expressed as:

$$\begin{aligned}\Psi(r,t,\hat{\Theta}^l) &= \frac{\partial}{\partial\hat{\Theta}}\varepsilon(r,t,\hat{\Theta}^l)\\ &= \begin{bmatrix}\frac{\partial}{\partial\sigma_1^1}\varepsilon(r,t,\hat{\Theta}^l)\\ \frac{\partial}{\partial\sigma_1^2}\varepsilon(r,t,\hat{\Theta}^l)\\ \vdots\\ \frac{\partial}{\partial\sigma_p^1}\varepsilon(r,t,\hat{\Theta}^l)\\ \frac{\partial}{\partial\sigma_p^2}\varepsilon(r,t,\hat{\Theta}^l)\end{bmatrix}, r\subseteq\Gamma.\end{aligned} \tag{20}$$

By substituting the (8) into (18) and differentiating (18) with respect to σ_h^1, we can have the following gradient term:

$$\begin{aligned}\frac{\partial}{\partial\sigma_h^1}\varepsilon(r,t,\hat{\Theta}^l) = &-\frac{(x-x_h)^2}{(\sigma_h^1)^3}\cdot[\frac{\alpha^h(r)}{\sum_{m=1}^p\alpha^m(r)}\hat{z}^h(r,t)\\ &-\frac{\alpha^h(r)}{(\sum_{m=1}^p\alpha^m(r))^2}\sum_{m=1}^p\alpha^m(r)\hat{z}^m(r,t)].\end{aligned} \tag{21}$$

Similarly,

$$\begin{aligned}\frac{\partial}{\partial\sigma_h^2}\varepsilon(r,t,\hat{\Theta}^l) = &-\frac{(y-y_h)^2}{(\sigma_h^2)^3}\cdot[\frac{\alpha^h(r)}{\sum_{m=1}^p\alpha^m(r)}\hat{z}^h(r,t)\\ &-\frac{\alpha^h(r)}{(\sum_{m=1}^p\alpha^m(r))^2}\sum_{m=1}^p\alpha^m(r)\hat{z}^m(r,t)].\end{aligned} \tag{22}$$

Step 4: The centralized proximal gradient descent iteration is

$$v^{l+1} = v^l + g^l, \tag{23}$$

where $g^l = f'(\hat{\Theta}^l)$; then

$$\hat{\Theta}^{l+1} = arg\min_{\hat{\Theta}^l}\{\langle v^{l+1},\hat{\Theta}^l\rangle\}, \tag{24}$$

$l = l+1$. Go to Step 2.

4.2.2. *Online Distributed Identification of the multi-model structure*

In this subsection, we extend our proposed method to an online and distributed setting using a parallel computing process. This online and distributed algorithm can be applied to a more realistic distributed comnunication in Section 2.3.2, which features limited communication range and time varying network topologies. With a distributed multi-agent system, individual agent cooperatively optimizes a global objective function. The global objective is to minimize

$$
\begin{aligned}
\sum_{t=1}^{t_f} f_t(\Theta) &= \sum_{t=1}^{t_f}\sum_{i=1}^{N} f_{t,i}(\Theta) \\
&= \sum_{t=1}^{t_f}\sum_{i=1}^{N}\frac{1}{2}[\int_{\Gamma_i} z(r,t) - \sum_{h=1}^{p}\eta_i^h(r)\hat{z}_i^h(r,t)d\Gamma_i]^2, r \subseteq \Gamma_i,
\end{aligned}
\tag{25}
$$

where $f_{t,i}(\Theta)$ is a convex cost function associated with agent i and evolves over time steps in an unpredictable manner. In other words, at time step t, each agent estimates Θ based on the local information available to it and to its neighbors. In this distributed scheme, $\hat{z}_i^h(r,t)$ denotes the one step ahead prediction of ith agent using the local linear PDE model, which can be written as:

$$
\begin{aligned}
\frac{\partial \hat{z}_i^h(r,t)}{\partial t} + v^T\nabla\hat{z}_i^h r,t) &= \theta_i^h\Delta\hat{z}_i^h(r,t), r \in \Gamma_i, \\
\hat{z}_i^h(r,t-T) &= z(r,t-T), r \in \Gamma_i.
\end{aligned}
\tag{26}
$$

To solve the distributed online optimization proposed above, we adapt the Online Distributed Dual Averaging (ODD) algorithm (Hosseini *et al.* 2013). The procedures of implementing the ODD algorithm is described in Algorithm 1.

Algorithm 1 Online Distributed Dual Averaging (ODD) algorithm for Multi-Agents Optimization

```
Initialize $\hat{\Theta}_i(0)$, $v_i(0) = \hat{\Theta}_i(0)$
for t=1 to $t_f$ do
    At time step $t$, the local objective function
    $f_t(\hat{\Theta}(t)) = \{f_{t,i}(\hat{\Theta}_i(t));\ for\ i = 1,...,N\}$
    Compute subgradient $g_i(t) = f'_{t,i}(\hat{\Theta}_i(t))$
    for Each agent i do
        Compute the neighborhood average
        $v_i(t+1) = \sum_{j\in\mathbb{N}_i} P_{i,j}v_j(t+1) + g_i(t)$
        where $P$ is doubly stochastic, so that $\sum_{j=1}^{N} P_{i,j} = 1$ and $\sum_{i=1}^{N} P_{i,j} = 1$
        Compute the next iterate $\bar{\Theta}_i(t+1)$:
        $\bar{\Theta}_i(t+1) = arg\min_{\bar{\Theta}_i}\{\langle v_i(t+1), \bar{\Theta}_i\rangle\}$
        Run local average
        $\hat{\Theta}_i(t+1) = \frac{t}{t+1}\hat{\Theta}_i(t) + \frac{1}{t+1}\bar{\Theta}_i(t+1)$
    end for
end for
```

5. Distributed Online Trajectory Design for the Multi-model Structure

We deploy a mobile sensor network in the field to collect information along its trajectory to identify the parameter of the multi-model structure. Many literatures have illustrated that the locations or trajectories of sensors have a great influence on the estimation results (Uciński 2004). Hence, it is desirable to optimize certain criteria to determine optimal locations or trajectories for identifying the multi-model structure (7). In order to design the optimal trajectory, we use the optimization criterion based on the Fisher information matrix (FIM) (Uciński 2004), which describes the amount of information that the measurements carry about the unknown parameters. The criterion has been widely used in optimum experimental design theory for DPSs (Uciński 2004). It should be noted that we only present a distributed online trajectory design in this work. In the centralized network case, the global information is available for each agent. Therefore, the design can be readily and straightforwardly extended to the centralized scheme.

In the distributed scheme, the output of the multi-model structure for the ith agent is

$$\hat{z}_i(r,t) = \sum_{h=1}^{p} \eta_i^h(r)\hat{z}_i^h(r,t), r \in \Gamma_i. \tag{27}$$

Taking the derivative of $\hat{z}_i(r,t)$ with respect to the parameter $\hat{\Theta}$, we obtain

$$g_i(r,t) = \left(\frac{\partial \hat{z}_i(r,t,\hat{\Theta}^l)}{\partial \hat{\Theta}^l}\right)^T, r \subseteq \Gamma_i, \tag{28}$$

where the definition of $\frac{\partial \hat{z}_i(r,t,\hat{\Theta}^l)}{\partial \hat{\Theta}^l}$ is similar to $\frac{\partial \hat{\varepsilon}_i(r,t,\hat{\Theta}^l)}{\partial \hat{\Theta}^l}$ in (20) and thus omited here.

Then the FIM of the ith agent at position r_i can be written down as:

$$M(r_i) = \frac{1}{t_p}\int_t^{t+t_p} det[g_i(r,t)g_i^T(r,t)]dt, i = 1,...,N, \tag{29}$$

where t_p is the predictive time interval.

To design the distributed online trajectory, we obtain the velocity of the formation center by:

$$\dot{r}_c = \tau \cdot \frac{\sum_{i=1}^N M(r_i)\cdot r_{iF}^d}{\sum_{i=1}^N M(r_i)} = \tau \cdot \frac{\frac{1}{N}\sum_{i=1}^N M(r_i)\cdot r_{iF}^d}{\frac{1}{N}\sum_{i=1}^N M(r_i)}, \tag{30}$$

where τ is the step size, r_{iF}^d is a normalized vector with the direction pointing from the formation center to the ith agent. The illustration of the setting of the r_{iF}^d is shown in Fig. 2.

Remark 3 We can see from (30) that $\dot{r}_c$ follows the direction of the FIM average value, which equals the average of $M(r_i)\cdot r_{iF}^d$ divided by the average of $M(r_i)$. By running an average consensus algorithm, each agent can obtain the moving direction of the formation center distributively and simultaneously.

After obtaining the moving direction of the virtual formation center, distributed formation control can be applied for the agent level control. Motivated by Zhang and Leonard (2010), Wu and Zhang (2012), Ren and Beard (2008), we apply the following consensus tracking algorithm for each agent,

$$u_i = \dot{r}_i = \dot{r}_i^d - \lambda_i(r_i - r_i^d) - \sum_{j=1}^{N} a_{ij}[(r_i - r_i^d) - (r_j - r_j^d)], \tag{31}$$

where u_i is the control input for the ith robot, λ_i is a positive scalar, a_{ij} is the (i,j) entry of the $N \times N$ adjacency matrix associated with the interaction topology, r_i represents the ith agent's

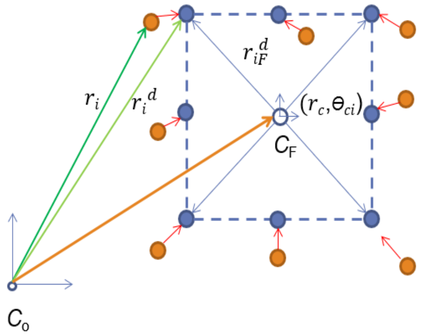


Figure 2. A formation composed of eight agents with a virtual formation center.

actual position and r_i^d represents the desired position of the ith agent, which is denoted as,

$$r_i^d = r_c + \mathbf{R}_i \cdot r_{iF}^d, \tag{32}$$

where r_{iF}^d represents the desired deviation of the ith agents relative to the virtual center r_c and $\mathbf{R}_i$ is the transformation matrix from body frame to inertia frame. In a 2D setting, $\mathbf{R}_i$ can be defined as $\begin{bmatrix} cos(\theta_{ci}) & -sin(\theta_{ci}) \\ sin(\theta_{ci}) & cos(\theta_{ci}) \end{bmatrix}$, where θ_{ci} is the orientation between the body frame and inertia frame. Fig. 2 shows an illustrative example of the virtual center approach with a formation composed of eight agents, where C_0 represents the inertial frame and C_F represents a body frame located at the virtual center r_c with an orientation θ_{ci} relative to C_0.

6. Simulation

For the demonstration of the proposed distributed algorithm, we consider a spatially distributed process in (1) with impulse initial condition at point (50,50). The spatially varying diffusion coefficient $\theta(r)$ is illustrated in Fig. 3. The whole domain is on the rectangle $0 \leq x \leq 100$, $0 \leq y \leq 100$. We implemented an implicit ADI finite-difference scheme in MATLAB, with 100-by-100 spatial grid.

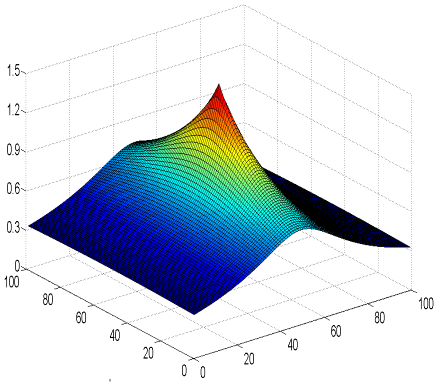
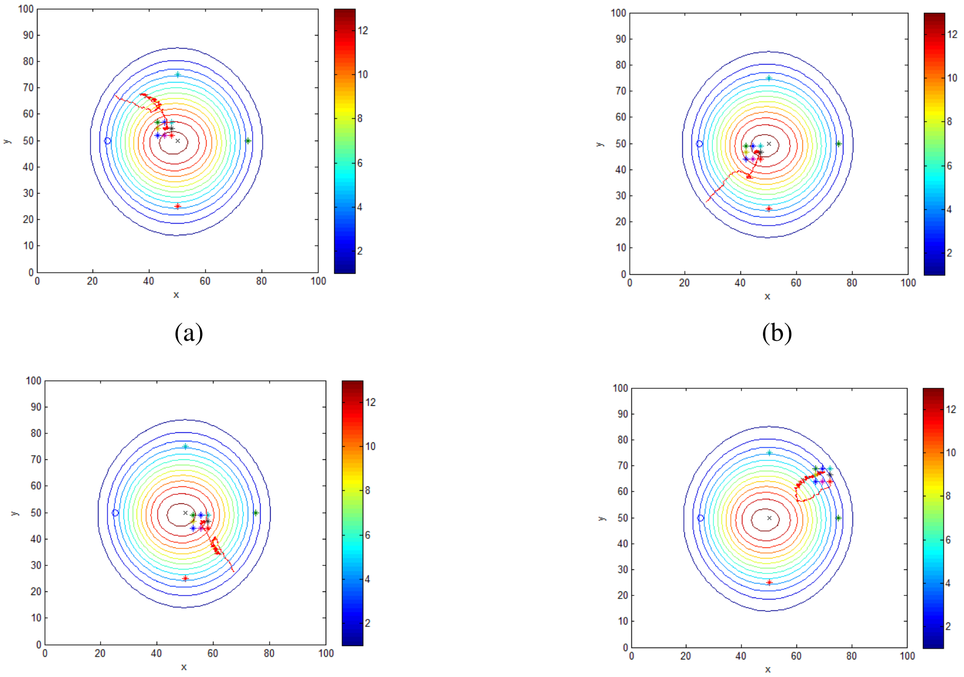


Figure 3. The spatially varying diffusion coefficient $\theta(r)$.

The selection of the setting positions is based on trial-and-errors. In this case study, five setting positions are used. The five setting positions are at $(50,25)$, $(50,75)$, $(50,50)$, $(25,50)$, and $(75,50)$. Eight mobile agents move around the corresponding setting positions and run the distributed online passive identifier (12)∼(14). After identification, the estimates of the diffusion coefficients at the five setting locations are $\hat{\theta}(50,25) = 0.5420$, $\hat{\theta}(50,75) = 0.5420$, $\hat{\theta}(50,50) = 1.5$, $\hat{\theta}(25,50) = 0.9621$, and $\hat{\theta}(75,50) = 0.9621$.

After obtaining the local linear PDE models, we select four different initial locations of the agents and adopt the control law (31) for each agent to find the optimal trajectory. The optimal trajectories of the mobile sensor network with four different initial positions are shown in the Fig. 4.



(a)

(b)

(c)

(d)

Figure 4. Optimal trajectories of the mobile sensor network with different initial positions.

Using the measurements along the optimal trajectories, the global multi-model PDE is identified by interpolating the five local linear PDE models. The initial value of the parameter vector is [9 9 9 9 9 9 9 9 9 9]. Following the procedures described in Section V, all parameters can be estimated. For system parameter identification, the mean square errors (MSE) of the predictions of the identified model against the real process data are calculated based on the following equation:

$$MSE_{\Theta} = \frac{\sum_{t=1}^{t_f} \sum_{i=1}^{N} \int_{\Gamma_i} [z(r,t) - \hat{z}_i(r,t)]^2 d\Gamma_i}{t_f}, r \subseteq \Gamma_i. \tag{33}$$

The mean square error of the online distributed optimization procedure is illustrated in Fig. 5. We can observe that the mean square error is gradually reduced as mobile agents collect more measurements and learn about the process. Table 1 compares the estimated parameters for each weighting function and the MSE of offline centralized optimization and online distributed optimization method. As we can observe from the table that, the online distributed method can achieve very similar performance as the offline centralized one.

Moreover, to achieve a fair validation of the identified model, additional data are generated

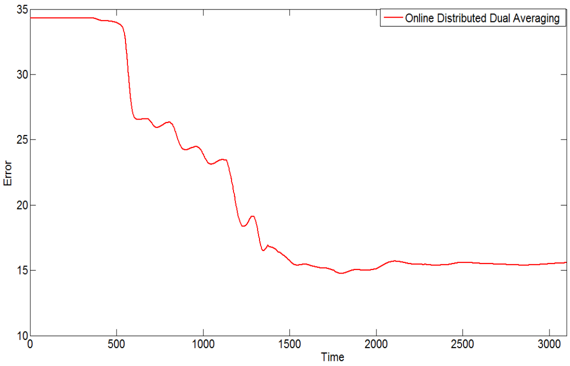


Figure 5. Mean square error of the online distributed optimization procedure

Table 1. The comparisons between Offline centralized and Online distributed optimization.

Θ	Offline centralized	Online distributed
$\sigma_{1,1}$	10.2315	11.0224
$\sigma_{1,2}$	12.2108	11.8405
$\sigma_{2,1}$	9.2905	9.3588
$\sigma_{2,2}$	12.0693	11.4153
$\sigma_{3,1}$	7.7377	6.9548
$\sigma_{3,2}$	5.8747	4.6665
$\sigma_{4,1}$	10.0448	9.3490
$\sigma_{4,2}$	9.2352	9.0354
$\sigma_{5,1}$	9.9517	9.6827
$\sigma_{5,2}$	8.7102	7.8066

Table 2. Mean square error comparison using the validation trajectory.

Models	MSE
Linear PDE (constant θ=0.5420)	9.284×10^{-3}
Linear PDE (constant θ=0.9621)	8.559×10^{-3}
Linear PDE (constant θ=1.5)	30.479×10^{-3}
Multi-model PDE without optimization (Initial value)	5.556×10^{-3}
Multi-model PDE with offline centralized optimization	3.064×10^{-3}
Multi-model PDE with online distributed optimization	3.751×10^{-3}

throughout a new validation trajectory that is different from the one in the previous simulation. In other words, the feasibility of the identified multi-model structure is verified by data attained from a different trajectory, which is illustrated in Fig. 6. Mean square error comparison along the valid trajectory is shown in Table 2. One can see that the identified multi-model structure with Gaussian weighting functions can effectively approximate the actual spatial varying advection-diffusion process. The proposed multi-model structure can achieve much better performance than the linear PDE model.

7. Conclusions

In this paper, we propose a novel multi-model structure to approximate the advection-diffusion process with spatially varying coefficients. It is demonstrated that under the proposed framework, both online distributed state estimation and parameter identification problem, and online

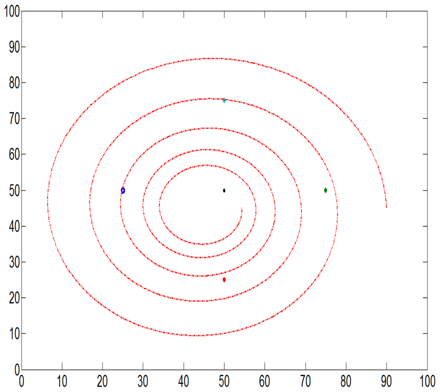


Figure 6. The validation trajectory for the mobile sensor network.

optimal trajectory design problem can be effectively solved. Simulation results show satisfactory performance. Future work includes the extension to other boundary conditions of PDE models and experimental validation.

References

Boukhris, A. and Mourot, J., 1999. Ragot non-linear dynamic system identification: a multi-model approach. *International Journal of Control*, 72, 591–604.

Burgard, W., *et al.*, 2005. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21 (3), 376–386.

Campbell, M.E. and Whitacre, W.W., 2007. Cooperative tracking using vision measurements on season UAVs. *IEEE Transactions on Control Systems Technology*, 15 (4), 613–626.

Chang, D., *et al.*, 2014. *Real-time Modeling of Ocean Current for Navigating Underwater Glider Sensing Networks*. Vol. 507. Springer.

Demetriou, M.A., 2010. Guidance of mobile actuator-plus-sensor networks for improved control and estaimation of distributed parameter system. *IEEE Transations on Automatic Control*, 55 (7), 1570–1584.

Demetriou, M.A., Gatsonis, N.A., and Court, J.R., 2014. Coupled control-computational fluids approach for the estimation of the concentration form a moving gaseous source in a 2-D domain with a Lyapunov- guided sensing aerial vehicle. *IEEE Transactions on Control Systems Technology*, 22 (3), 853–867.

Demetriou, M.A. and Hussein, I.I., 2009. Estimation of spatially distributed process using mobile spatially distributed sensor network. *SIAM Journal on Control and Optimization*, 48 (1), 266–291.

Dennis, J.E., 1983. *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice Pren, Englewood Cliffs, NJ.

Gay, D.H. and Ray, W.H., 1995. Identification and control of distributed parameter systems by means of the singular value decomposition. *Chemical Engineering Science*, 50, 1519–1539.

Hosseini, S., Chapman, A., and Mesbahi, M.C., 2013. Online distributed optimization via dual averaging. *In*: *Proc. 52th IEEE Conference on Decision and Control*, 1484–1489.

Huang, J., *et al.*, 2012. Identification of multi-model LPV models with two scheduling variables. *Journal of Process Control*, 22 (7), 1198–1208.

Krstic, M. and Symshlyaev, A., 2008. Adaptive control for PDEs. *Annual Reviews in Control*, 32 (2), 149–160.

Li, H. and Qi, C., 2010. Modeling of distributed parameter systems for application-A synthesized review from time-space separation. *Journal of Process Control*, 20 (891-901).

Ljung, L., 1998. *System Identfication, 2nd ed.*. Prentice-Hall, Englewood Cliffs.

Mourikis, A.I. and Roumeliotis, S.I., 2006. Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics*, 22 (4), 666–681.

Murray-Smith, R. and Johansen, T., 1997. *Multiple Model Approaches to Modelling and Control*. Taylor and Francis.

Omatu, S. and Matumoto, K., 1991. Distributed parameter identification by regularization and its application to predication of air pollution. *Internation Journal of Systems Science*, 22 (10), 2001–2012.

Ren, W. and Beard, R.W., 2008. *Distributed consensus in multi-vehicle cooperative control, communications and control engineering series*. London: Springer-Verlag.

Symshlyaev, A. and Krstic, M., 2007. Adaptive boundary control for unstable parabolic PDEs - Part II.: Estimation-based designs. *Automatica*, 43 (9), 1543–1556.

Tang, Z. and Ozguner, U., 2005. Motion planning for multitarget surveillance with mobile sensor agents. *IEEE Transactions on Robotics*, 21 (5), 898–908.

Uciński, D., 2004. *Optimal measurment methods for distributed parameter system identification*. CRC Press.

Uciński, D. and Chen, Y., 2005. Timecoptimal path planning of moving sensors for paramter estimation of distributed systems. *In*: *Proc. of 44th IEEE Conf. on Decision and Control and European Control Conference*.

Wu, W. and Zhang, F., 2012. Robust cooperative exploration with a switching strategy. *IEEE Transactions on Robotics*, 28 (4), 828–839.

Xiao, L., Boyd, S., and Lall, S., 2006. A space-time diffusion scheme for peer-to-peer least-squares estimation. *In*: *Proc. of the 5th international conference on Information processing in sensor networks*, April, 19–26.

You, J. and Wu, W., 2017. Online passive identifier for spatially distributed parameter systems using mobile sensor networks. *IEEE Transactions on Control System Technology* Accepted.

You, J., Zhang, F., and Wu, W., 2016. Cooperative filtering for parameter identification of diffusion processes. *In*: *Proc. of IEEE Conference on Decision and Control*, 4327-4333 Accepted.

You, J., Zhang, F., and Wu, W., 2017. Cooperative Parameter Identification of Advection-diffusion Process Using a Mobile Sensor Network. *In*: *American Control Conference* Accepted.

Zhang, F. and Leonard, N.E., 2010. Cooperative control and filtering for cooperative exploration. *IEEE Transactions on Automatic Control*, 55 (3), 650–663.