2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Cooperative Parameter Identification of Advection-diffusion Processes Using a Mobile Sensor Network

Jie You, Yufei Zhang, Mingchen Li, Kun Su, Fumin Zhang, and Wencen Wu

***Abstract*— Online parameter identification of advection-diffusion processes is performed using a mobile sensor network. A constrained cooperative Kalman filter is developed to provide estimates of the field values and gradients along the trajectories of the mobile sensor network so that the temporal variations of the field values can be estimated. Utilizing the state estimates from the constrained cooperative Kalman filter, a recursive least square (RLS) algorithm is designed to estimate the unknown parameters of the advection-diffusion process. We provide bias analysis of the RLS in the paper. In addition to validating the proposed algorithm in simulated advection-diffusion fields, we build a controllable CO_2 advection-diffusion field in a lab and design a sensor grid that collects the field concentration over time to allow the validation of the proposed algorithm in the CO_2 field. Experimental results demonstrate robustness of the algorithm under realistic uncertainties and disturbances.**

I. Introduction

Many complicated spatio-temporal processes have been observed in diverse fields including physical, chemical, and biological systems [1]. These spatio-temporal processes are often viewed as distributed parameter systems (DPSs), which are mathematically described by partial differential equations (PDEs) in model-based schemes. In many practical problems, the parameters of PDEs such as the diffusion coefficient may be unknown or inaccurate. Therefore, to better understand the processes, there is a need to use parameter identification methods to refine, update, or estimate these unknown parameters [2], [3]. On the other hand, the procedure of parameter identification will also provide important insights into the analysis, design, and control of DPSs under study. Many identified models have been used in applications [4].

Various aspects of parameter identification of DPSs have been investigated in [5]–[7] and references therein. The identification of PDEs from discrete samples can be done at least in two ways: indirect method [7] and direct method [2], [8]. A typical indirect method is based on a weak formulation using a Galerkin-like finite element procedure [7]. The most attractive feature of the indirect method is its flexibility to deal with PDEs with arbitrary initial conditions and complex geometric boundaries. In a direct method, the spatio-temporal variables are usually discretized with respect to both time and space. Derivatives of the functions at each discretization node have to be approximated using some standard finite difference approximations such as finite difference and finite volume method [2], [8]. The direct method can be readily used in all kinds of PDEs and maintain a straightforward link to the physical properties of the original DPS system.

Many of these studies require large numbers of static sensors to collect data in the whole domain. Due to the limited number of actuators and sensors in practical sensing, in a very large and complex field, it is preferable to employ mobile sensor networks (MSNs), which consist of groups of robotic agents with computational, communication, sensing, and locomotive capabilities [9]–[11], to perform parameter identification. Although there exist some contributions on the issue of parameter identification of PDEs using mobile sensor networks [3], [12], [13], most of these studies are based on an offline scheme and require high computational loads with few exceptions that investigate the online parameter identification [2], [14], [15]. There are a number of difficulties inherent in the online parameter identification of PDEs. First, it is a challenging inverse problem, which requires the identification of system parameters from collected finite-dimensional measurements. Second, online parameter identification using a mobile sensor network requires a combination of cooperative control and cooperative sensing.

In our previous work [2], we designed a cooperative filtering scheme for online parameter estimation of diffusion processes using four sensing agents arranged in a symmetric formation. The scheme consists of two parts: a cooperative Kalman filter and a recursive least square (RLS) estimator. We proved the convergence of the cooperative Kalman filter and validated the algorithm in simulations. In this paper, we investigate online parameter identification for 2D advection-diffusion processes. By using the finite volume method, we extend the cooperative filtering scheme [2] to the case with $N \geq 4$ agents in an arbitrary formation to allow flexibility in practical scenarios. Utilizing the state estimates from the cooperative filtering scheme, a RLS algorithm is designed to estimate the unknown model parameters of the advection-diffusion process. We provide necessary bias analysis of the proposed method. Additionally, we build a controllable CO_2 advection-diffusion field in a lab and design a sensor grid that collects the field concentration over time to allow the validation of the proposed algorithm in the CO_2 field. Experimental results show satisfactory performance.

The problem is formulated in Section II. Section III presents the finite volume approximation model and Section IV shows the cooperative Kalman filtering. Section V illustrates the RLS and bias analysis of the proposed method. Experiments results are presented in Section VI and

The research work is supported by NSF grant CNS-1446461. Jie You, Yufei Zhang, Mingchen Li, Kun Su, and Wencen Wu are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA. Fumin Zhang is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30318, USA youj@rpi.edu, zhangy47@rpi.edu, lim14@rpi.edu, suk2@rpi.edu, fumin@gatech.edu, wuw8@rpi.edu

conclusions follow in Section VII.

II. Problem formulation

In this section, we formulate the problem of online parameter estimation of advection-diffusion processes using mobile sensor networks.

A. The model

We assume that system dynamics is described by the following two-dimensional (2D) advection-diffusion process defined on a domain $\Omega = [0, L_x] \times [0, L_y] \in \mathbb{R}^2$:

$$\frac{\partial z(r,t)}{\partial t} = \theta \nabla^2 z(r,t) + v^T \nabla z(r,t), \; r \in \Omega, \tag{1}$$

where $z(r,t)$ is the concentration function, $\theta > 0$ is a constant diffusion coefficient, ∇ represents the gradient operator, ∇^2 represents the Laplacian operator, and v is a constant vector representing the flow velocity, which is supposed to be known through measurements. The initial and boundary conditions for Equation (1) are assumed as $z(r,0) = z_0(r)$, and $z(r,t) = z_b(r,t), \; r \in \partial\Omega$, where $z_0(r)$ and $z_b(r,t)$ are the arbitrary initial condition and Dirichlet boundary condition, respectively. Many natural processes can be described by the advection-diffusion equation (1). In many scenarios, θ is unknown or inaccurate, which requires identification.

B. 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 $z(r,t)$. We consider the sensing agents with single-integrator dynamics given by $\dot{r}_i(t) = u_i(t), i = 1, 2, ..., N$, where $r_i(t)$ and $u_i(t) \subseteq \mathbb{R}^2$ are the position and the velocity of the ith agent, respectively. In most applications, the sensor measurements are taken discretely over time. Let the moment when new measurements are available be t_k, where k is an integer index. Denote the position of the ith agent at the moment t_k be r_i^k and the field value at r_i^k be $z(r_i^k, k)$. The measurement of the ith agent can be modeled as

$$p(r_i^k, k) = z(r_i^k, k) + n_i, \tag{2}$$

where n_i is assumed to be i.i.d. Gaussian noise. We have the following assumption for the sensing agents.

Assumption II.1 *Each agent can measure its position r_i^k and concentration value $z(r_i^k, k)$, and share these information with other agents.*

The problem is formulated as:

1) Under Assumption II.1, develop a cooperative filtering scheme that estimate the states $z(r,t)$, $\nabla z(r,t)$, $\nabla^2 z(r,t)$, and $\frac{\partial z(r,t)}{\partial t}$ based on the collected measurements in Equation (2) using a mobile sensor network moving in the advection-diffusion field.
2) Utilizing the estimated state, develop an online parameter identification algorithm that estimates the unknown constant diffusion coefficient θ of the advection-diffusion equation (1).

III. The finite volume approximation

Under Assumption II.1, the proposed parameter identification algorithm is based on the discrete measurements taken by mobile agents over time. In the following, we will first build a finite volume approximation model of Equation (1). Suppose the current time step is k. Let $r_c^k = [r_{c,x}^k, r_{c,y}^k]^T$ be the center of the formation at the moment t_k, i.e., $r_c^k = \frac{1}{N}\sum_{i=1}^N r_i^k$. We discretize the advection-diffusion PDE (1) at the formation center r_c^k as,

$$\frac{z(r_c^k, k+1) - z(r_c^k, k)}{t_s} - v^T \nabla z(r_c^k, k) = \theta \nabla^2 z(r_c^k, k), \tag{3}$$

where t_s is the sampling period.

If we can get the estimates of $z(r_c^k, k+1)$, $z(r_c^k, k)$, $\nabla z(r_c^k, k)$, and $\nabla^2 z(r_c^k, k)$, then θ can be estimated using RLS based on the semi-discrete model (3). To obtain these state estimates, in Section IV, we will develop a constrained cooperative Kalman filter to estimate $z(r_c^k, k+1)$, $z(r_c^k, k)$, and $\nabla z(r_c^k, k)$ along the moving trajectory of a mobile sensor network. On the other hand, the Laplacian term $\nabla^2 z(r_c^k, k)$ also requires to be estimated simultaneously. One simple and straightforward way is to use finite difference method to approximate $\nabla^2 z(r_c^k, k) = \frac{\sum_{i=1}^4 z(r_i^k, k) - 4z(r_c^k, k)}{\Delta r}$, where Δr is the spatial interval [2]. Unfortunately, this method only works for the case when four agents are arranged in a symmetric formation, which limits its capability for usage in actual applications. In the following section, we will employ a finite volume method to allow the estimation of $\nabla^2 z(r_c^k, k)$ with $N \geq 4$ agents in an arbitrary formation.

We summarize the basic procedures following the finite volume method [8]. We first denote the cells of agents as $C_1^k, C_2^k, ... C_N^k$, and the corresponding cell-centers as $r_1^k, r_2^k, ...\, r_N^k$. We further denote the cells of the formation center r_c^k as C_c^k. The volume of the formation center cell C_c^k is denoted as Ω_c, which is a finite volume. Let the surface area of Ω_c be $\mathbf{S} = S\hat{n}$, where $\hat{n}$ is the outward unit vector. To illustrate the idea and for notation convenience, we consider the case when $N = 4$ in the following derivations. But our scheme can be straightforwardly extended to the case when $N \geq 4$, which will be specified in Remark III.1. As illustrated in Fig. 1, we arrange four agents in an arbitrary formation. In this case, the surface area $\mathbf{S}$ is the quadrilateral $ABCD$ and Ω_c is the volume of $ABCD$. The corresponding outward unit vector $\hat{n}$ for the edge AB is $\hat{n}_{AB} = r_1^k r_c^k$, which means $AB \perp r_1^k r_c^k$. In a similar way, we have $BC \perp r_2^k r_c^k$, $CD \perp r_3^k r_c^k$, and $DA \perp r_4^k r_c^k$.

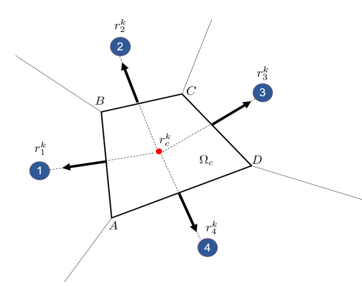


Fig. 1. Finite-volume construction for a mobile sensor network in 2D.

By integrating Equation (1) over a finite volume Ω_c, we can have the following expression:

$$\int\int_{\Omega_c}\frac{\partial}{\partial t}z(r,t)-v^T\nabla z(r,t)\ d\Omega_c+\oint_S \mathbf{F}\cdot\hat{n}dS=0, \quad (4)$$

where $\mathbf{F}=-\theta\nabla z(r,t)$ is obtained by applying the Green's theorem [16].

The integration (4) over a cell area $ABCD$ shown in Fig. 1 results in the semi-discrete equation as follows:

$$\frac{\partial z(r_c,t)}{\partial t}-v^T\nabla z(r_c,t)=-\frac{1}{\Omega_c}\sum_{faces}(\mathbf{F}_{AB}\cdot\mathbf{S}_{AB}+\mathbf{F}_{BC}\cdot\mathbf{S}_{BC}+\mathbf{F}_{CD}\cdot\mathbf{S}_{CD}+\mathbf{F}_{DA}\cdot\mathbf{S}_{DA}), \quad (5)$$

where $\mathbf{F}_{AB}\cdot\mathbf{S}_{AB}$ is the continuous flux on the edge AB, which is expressed as diffusive terms as follows,

$$\mathbf{F}_{AB}\cdot\mathbf{S}_{AB}=-\int_{AB}\theta\nabla z(r,t)\cdot\hat{n}_{AB}\ dl, \quad (6)$$

where $\hat{n}_{AB}$ is the unit outer normal on the edge AB. The flux terms with respect to the other edges $\mathbf{F}_{BC}\cdot\mathbf{S}_{BC}$, $\mathbf{F}_{CD}\cdot\mathbf{S}_{CD}$, and $\mathbf{F}_{DA}\cdot\mathbf{S}_{DA}$ have the similar definitions.

Next, we will derive $\nabla z(r,t)$, $r\in AB$ in Equation (4) at time step k. With r_1^k being close to r_c^k, $z(r_1^k,k)$ can be locally approximated by using the Taylor expansion as,

$$z(r_1^k,k)-z(r_c^k,k)\approx(r_1^k-r_c^k)^T\nabla z(r,k)+\int_0^1(H_{r_1^k}-H_{r_c^k})\xi d\xi, \quad r\in AB, \quad (7)$$

where $H_{r_1^k}=(r_1^k-r)^T H\left(\xi r+(1-\xi)r_1^k,k\right)(r_1^k-r)$ with $H\left(\xi r+(1-\xi)r_1^k,k\right)$ being the Hessian matrix at the point $\xi r+(1-\xi)r_1^k$, $r\in AB$. The other notations $H_{r_2^k}$, $H_{r_3^k}$, $H_{r_4^k}$, and $H_{r_c^k}$ have the similar definition. Hence, by reorganizing Equation (7), we can obtain the gradient term $\nabla z(r,k)$ for the edge $r\in AB$.

Substituting the expression of $\nabla z(r,k)$ into $\int_{AB}\theta\nabla z(r,k)\cdot\hat{n}_{AB}\ dl$ gives,

$$\int_{AB}\theta\nabla z(r,k)\cdot\hat{n}_{AB}\ dl=\theta\frac{|A-B|}{|r_1^k-r_c^k|}\left(z(r_1^k,k)-z(r_c^k,k)\right)-\frac{\theta}{|r_1^k-r_c^k|}\int_{AB}\int_0^1(H_{r_1^k}-H_{r_c^k})\xi d\xi dl. \quad (8)$$

Denote $E_{AB}=-\frac{\theta}{|r_1^k-r_c^k|}\int_{AB}\int_0^1(H_{r_1^k}-H_{r_c^k})\xi d\xi dl$. Since E_{AB} is the integration of the differences of two Hessian matrices, it is obvious that $E_{AB}=O(h^2)$, which is the higher order term of the grid size $h=\sup_{i=N} diam(C_i^k)^{1/2}$. Here, $diam(C_i^k)$ is the diameter of cell C_i^k.

By substituting (8) into (6), we can obtain

$$\mathbf{F}_{AB}\cdot\mathbf{S}_{AB}=-\theta\frac{|A-B|}{|r_1^k-r_c^k|}\left(z(r_1^k,k)-z(r_c^k,k)\right)+E_{AB}. \quad (9)$$

In a similar way, we can obtain the normal flux on the other sides, $\mathbf{F}_{BC}\cdot\mathbf{S}_{BC}=-\theta\frac{|B-C|}{|r_2^k-r_c^k|}\left(z(r_2^k,k)-z(r_c^k,k)\right)+E_{BC}$, $\mathbf{F}_{CD}\cdot\mathbf{S}_{CD}=-\theta\frac{|C-D|}{|r_3^k-r_c^k|}\left(z(r_3^k,k)-z(r_c^k,k)\right)+E_{CD}$, and $\mathbf{F}_{DA}\cdot\mathbf{S}_{DA}=-\theta\frac{|D-A|}{|r_4^k-r_c^k|}\left(z(r_4^k,k)-z(r_c^k,k)\right)+E_{DA}$. Then we can rewrite Equation (5) as follows,

$$\frac{z(r_c^k,k+1)-z(r_c^k,k)}{t_s}-v^T\nabla z(r_c^k,k)=\theta\frac{-1}{\Omega_c}[\alpha_{AB}\cdot z(r_1^k,k)+\alpha_{BC}\cdot z(r_2^k,k)+\alpha_{CD}\cdot z(r_3^k,k)+\alpha_{DA}\cdot z(r_4^k,k)+\alpha_{center}\cdot z(r_c^k,k)]+e(r_c^k,k), \quad (10)$$

where the α coefficients are as follows:

$$\alpha_{AB}=\frac{|A-B|}{|r_1^k-r_c^k|},\alpha_{BC}=\frac{|B-C|}{|r_2^k-r_c^k|},$$
$$\alpha_{CD}=\frac{|C-D|}{|r_3^k-r_c^k|},\alpha_{DA}=\frac{|D-A|}{|r_4^k-r_c^k|}, \quad (11)$$
$$\alpha_{center}=-\frac{|A-B|}{|r_1^k-r_c^k|}-\frac{|B-C|}{|r_2^k-r_c^k|}-\frac{|C-D|}{|r_3^k-r_c^k|}-\frac{|D-A|}{|r_4^k-r_c^k|},$$

and $e(r_c^k,k)=-\frac{1}{\Omega_c}(E_{AB}+E_{BC}+E_{CD}+E_{DA})$ is the approximation error from omitting higher order terms using finite-volume method. This then allows us to assume that the modeling error $e(r_c^k,k)$ is an independent noise sequence with zero mean and finite variance.

For notation simplification, let us denote $\sum_{i=1}^4\alpha_{\sigma_i}\cdot z(r_i^k,k)=\alpha_{AB}\cdot z(r_1^k,k)+\alpha_{BC}\cdot z(r_2^k,k)+\alpha_{CD}\cdot z(r_3^k,k)+\alpha_{DA}\cdot z(r_4^k,k)$. Then a finite volume approximation of the advection-diffusion PDE (1) can be written as follows:

$$\frac{z(r_c^k,k+1)-z(r_c^k,k)}{t_s}-v^T\nabla z(r_c^k,k)=\theta\frac{-1}{\Omega_c}[\sum_{i=1}^4\alpha_{\sigma_i}\cdot z(r_i^k,k)+\alpha_{center}\cdot z(r_c^k,k)]+e(r_c^k,k). \quad (12)$$

We can observe that Equation (12) is also a discretized version of Equation (1) with a replacement of $\nabla^2 z(r_c^k,k)$ with $\frac{-1}{\Omega_c}[\sum_{i=1}^4\alpha_{\sigma_i}\cdot z(r_i^k,k)+\alpha_{center}\cdot z(r_c^k,k)]$ in Equation (3). It should be noted that t_s must obey the inequalities $t_s\le\frac{4\theta}{|v|^2}$ and $t_s\le\frac{h^2}{4\theta}$ for the discretization method to converge [8].

Remark III.1 *Even though we only consider four agents in the above derivation, the finite volume approximation of the advection-diffusion model* (1) *can be readily extended to the case when $N\ge 4$ following the implementation of the standard finite-volume method outlined in [8].*

IV. COOPERATIVE FILTERING FOR PARAMETER IDENTIFICATION

In this section, we show how to design a cooperative filtering scheme to sequentially estimate the states $z(r_c^k,k+1)$, $z(r_c^k,k)$, $\nabla z(r_c^k,k)$ and $\sum_{i=1}^4 z(r_i^k,k)$ over time.

A. Information dynamics for the cooperative Kalman filter

We first introduce the motivation of designing a cooperative filter by pointing out the difference between $z(r_c^k,k+1)$ and $z(r_c^k,k)$ in Equation (12). By designing a cooperative filter similar to the one developed in [17], $z(r_c^k,k)$ may be directly estimated by combining the measurements taken by the sensing agents at time step k. However, at time step $k+1$, the formation center of the group is at position r_c^{k+1}. Therefore, the cooperative filter design in [17] can only provide the estimate of $z(r_c^{k+1},k+1)$, not $z(r_c^k,k+1)$. In order to estimate the temporal variations of the field value along the trajectory, we first need to derive a cooperative filter to estimate both $z(r_c^k,k)$ and $z(r_c^k,k+1)$.

To construct a cooperative Kalman filter to obtain the estimates of $z(r_c^k,k)$ and $z(r_c^k,k+1)$, we first analyze the dynamics of the advection-diffusion field value along the trajectory of the formation center r_c according to

$$\dot{z}(r_c,t) = \frac{\partial z(r_c,t)}{\partial r_c}\frac{dr_c}{dt} + \frac{\partial z(r_c,t)}{\partial t} = \nabla z(r_c,t)\cdot \dot{r}_c + \frac{\partial z(r_c,t)}{\partial t}, \tag{13}$$

where $\nabla z(r_c,t)$ is the gradient of $z(r_c,t)$. To discretize Equation (13), the finite differences of each term of (13) at time $t=t_{k-1}$ and at position $r_c = r_c^{k-1}$ give:

$$\dot{z}(r_c,t)|_{t=t_{k-1},r_c=r_c^{k-1}} \approx \frac{z(r_c^k,k)-z(r_c^{k-1},k-1)}{t_s}, \tag{14}$$
$$\nabla z(r_c,t)\cdot \dot{r}_c|_{t=t_{k-1},r_c=r_c^{k-1}} \approx \frac{(r_c^k-r_c^{k-1})^T\nabla z(r_c^{k-1},k-1)}{t_s}.$$

Substituting Equation (14) and the finite volume equation (12) into Equation (13) gives the information dynamics of $z(r_c^k,k)$ as

$$z(r_c^k,k) = \left(1+\frac{\alpha_{center}\hat{\theta}_k t_s}{\Omega_c}\right) z(r_c^{k-1},k-1) - \frac{\hat{\theta}_k t_s}{\Omega_c}\sum_{i=1}^{4}\alpha_{\sigma_i} \tag{15}$$
$$\cdot z(r_i^{k-1},k-1) + (r_c^k - r_c^{k-1}+vt_s)^T\nabla z(r_c^{k-1},k-1) + w(r_c^k,k),$$

where $\hat{\theta}_k$ is the estimate of θ, which can be obtained from the RLS method that will be introduced in Section V. $w(r_c^k,k)$ is the error term, which accounts for positioning errors, estimation errors for the Hessian matrix, and errors caused by higher-order terms omitted from the finite volume scheme.

Similarly, we also obtain the dynamics of $z(r_c^k,k+1)$ by discretizing Equation (13) at $t=t_k$ and $r_c = r_c^{k-1}$.

$$z(r_c^k,k+1) = \left(1+\frac{\alpha_{center}\hat{\theta}_k t_s}{\Omega_c}\right) z(r_c^{k-1},k) - \frac{\hat{\theta}_k t_s}{\Omega_c}\sum_{i=1}^{4}\alpha_{\sigma_i}$$
$$\cdot z(r_i^k,k) + (r_c^k - r_c^{k-1}+vt_s)^T\nabla z(r_c^{k-1},k) + w(r_c^k,k). \tag{16}$$

Furthermore, we are also interested in estimating $\nabla z(r_c,t)$ since the gradient estimate is not only necessary for the RLS method, but also used in the motion control that will be introduced in Section IV-D. We derive the total time derivative of $\nabla z(r_c,t)$ as

$$\dot{\nabla} z(r_c,t) = H(r_c,t)\cdot \dot{r}_c + \frac{\partial \nabla z(r_c,t)}{\partial t}, \tag{17}$$

where $H(r_c,t)$ is the Hessian matrix, and $\frac{\partial \nabla z(r_c,t)}{\partial t}$ is the higher order term, which can be considered as noise. By discetizing Equation (17) at $t=t_{k-1}, r_c = r_c^{k-1}$ and $t=t_k, r_c=r_c^{k-1}$, respectively, we can get that $\nabla z(r_c^k,k)$ and $\nabla z(r_c^k,k+1)$ evolve according to the following equations:

$$\nabla z(r_c^k,k) = \nabla z(r_c^{k-1},k-1) + H(r_c^{k-1},k-1)(r_c^k - r_c^{k-1}),$$
$$\nabla z(r_c^k,k+1) = \nabla z(r_c^{k-1},k) + H(r_c^{k-1},k)(r_c^k - r_c^{k-1}). \tag{18}$$

Define the information state as $X(k+1) = [z(r_c^k,k), \nabla z(r_c^k,k), z(r_c^k,k+1), \nabla z(r_c^k,k+1)]^T$. By combing (15), (16), and (18), the information state evolves according to the following equation:

$$X(k+1) = A_{\hat{\theta}}(k)X(k) + U(k) + w(k), \tag{19}$$

where $w(k) = [w(r_c^k,k-1), 0, w(r_c^k,k), 0]^T$ represents the model error terms in Equation (15) and (16). We denote the covariance matrix of $w(k)$ as $E[w(k)w(k)^T] = W$. The matrices $A_{\hat{\theta}}(k)$ and $U(k)$ are defined by

$$A_{\hat{\theta}}(k) = \begin{bmatrix} 1+\frac{\alpha_{center}\hat{\theta}_k t_s}{\Omega_c} & (r_c^k - r_c^{k-1}+vt_s)^T & 0 & 0 \\ 0 & I_{2\times 2} & 0 & 0 \\ 0 & 0 & 1+\frac{\alpha_{center}\hat{\theta}_k t_s}{\Omega_c} & (r_c^k - r_c^{k-1}+vt_s)^T \\ 0 & 0 & 0 & I_{2\times 2} \end{bmatrix}. \tag{20}$$

$$U(k) = \begin{bmatrix} -\frac{\hat{\theta}_k t_s}{\Omega_c}\sum_{i=1}^{4}\alpha_{\sigma_i}\cdot z(r_i^{k-1},k-1) \\ H(r_c^{k-1},k-1)(r_c^k - r_c^{k-1}) \\ -\frac{\hat{\theta}_k t_s}{\Omega_c}\sum_{i=1}^{4}\alpha_{\sigma_i}\cdot z(r_i^k,k) \\ H(r_c^{k-1},k)(r_c^k - r_c^{k-1}) \end{bmatrix}, \tag{21}$$

where $H(r_c^{k-1},k)$ is the Hessian matrix. We observe that $U(k)$ is determined by the values of $z(r_i^{k-1},k-1)$, $z(r_i^k,k)$, and the Hessian matrix, which will be specified in Section IV-C.

A measurement equation is also required for the cooperative Kalman filter. By applying formation control, r_i^k and r_i^{k-1} can be controlled to be close to r_c^{k-1}. Therefore, the concentration can be locally approximated by a Taylor series up to second order as

$$z(r_i^{k-1},k-1) \approx z(r_c^{k-1},k-1) + (r_i^{k-1}-r_c^{k-1})^T\nabla z(r_c^{k-1},k-1)$$
$$+\frac{1}{2}(r_i^{k-1}-r_c^{k-1})^T H(r_c^{k-1},k-1)(r_i^{k-1}-r_c^{k-1}),$$
$$z(r_i^k,k) \approx z(r_c^{k-1},k) + (r_i^k - r_c^{k-1})^T\nabla z(r_c^{k-1},k) \tag{22}$$
$$+\frac{1}{2}(r_i^k - r_c^{k-1})^T H(r_c^{k-1},k)(r_i^k - r_c^{k-1}).$$

Let $P(k) = [p(r_1^{k-1},k-1)\cdots p(r_N^{k-1},k-1)\ p(r_1^k,k)\cdots p(r_N^k,k)]^T$. By combing Equation (2) and Equation (22), the measurement equation can be modelled in a vector form as,

$$P(k) = C(k)\cdot X(k) + D(k)\hat{H}(k) + D(k)\varepsilon(k) + n(k), \tag{23}$$

where $\hat{H}(k) = [\hat{H}(r_c^{k-1},k-1)\ \hat{H}(r_c^{k-1},k)]^T$ is a column vector obtained by rearranging elements of the estimate of Hessian terms, $\varepsilon(k)$ represents the error in the estimation of the Hessian matrices, $n(k)$ is the Gaussian measurement n_i in a vector form, $E[\varepsilon(k)\varepsilon(k)^T] = Q$, $E[n(k)n(k)^T] = R$. $D(k)$ is a matrix with its first N rows defined by $[\frac{1}{2}((r_i^{k-1}-r_c^{k-1})\bigotimes(r_i^{k-1}-r_c^{k-1}))^T\ \ 0]$ and last N rows defined by $[0\ \ \frac{1}{2}((r_i^k - r_c^{k-1})\bigotimes(r_i^k - r_c^{k-1}))^T]$, where $i=1,2,\cdots,N$ and $\bigotimes$ is the Kronecker product. $C(k)$ is a matrix with its first N rows defined by $[1\ \ (r_i^{k-1}-r_c^{k-1})^T\ \ 0\ \ 0]$ and last N rows defined by $[0\ \ 0\ \ 1\ \ (r_i^k - r_c^{k-1})^T]$ for $i=1,2,\cdots,N$.

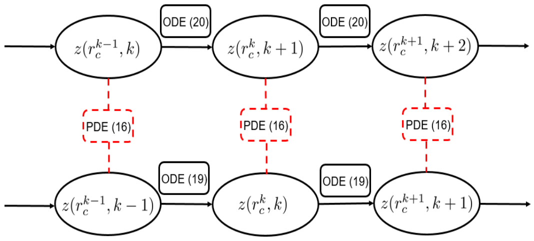


Fig. 2. Block diagram of the relationship between $z(r_c^k,k+1)$ and $z(r_c^k,k)$.

B. The PDE state constraint

We observe that the matrices $A_{\hat{\theta}}(k)$, $C(k)$, and $D(k)$ are block diagonal matrices. That means the information dynamics (19) was obtained here as a direct combination of the semidiscrete ODE (16) for the state $z(r_c^k,k+1)$ and ODE (15) for the state $z(r_c^k,k)$. As a matter of fact, the state $z(r_c^k,k+1)$ and $z(r_c^k,k)$ are the "future and present" state estimates at a given position $r=r_c^k$, which are discretized terms of $\frac{\partial z(r_c,t)}{\partial t}$ in Equation (12). Hence, there is a PDE constraint between the state $z(r_c^k,k+1)$ and $z(r_c^k,k)$ at each step, which is shown in Fig. 2. By rewriting Equation (12), we can obtain the PDE constraint:

$$z(r_c^k,k+1)+(\frac{\alpha_{center}\hat{\theta}_k t_s}{\Omega_c}-1)z(r_c^k,k)+v^T t_s \nabla z(r_c^k,k)$$
$$=\frac{-\hat{\theta}_k t_s}{\Omega_c}\sum_{i=1}^{4}\alpha_{\sigma_i}\cdot z(r_i^k,k) \tag{24}$$

The state equality constraint can be rewritten as follow:

$$G(k)\cdot X(k)=d(k), \tag{25}$$

where $G(k)=[(\frac{\alpha_{center}\hat{\theta}_k t_s}{\Omega_c}-1)\quad 0\quad 1\quad v^T t_s]$ and $d(k)=\frac{-\hat{\theta}_k t_s}{\Omega_c}\sum_{i=1}^{4}\alpha_{\sigma_i}\cdot z(r_i^{k-1},k-1)$.

We observe that the proposed cooperative Kalman filter is based on the time-varying information dynamics (19) with the state equality constraint (25). This type of filter has been previously investigated in [18]. By following canonical procedures in [18], the equations for the cooperative Kalman filter with state equality constraints can be obtained. Details of the cooperative Kalman filter can be found in [2].

C. Cooperative estimation of the Hessian

Estimates of $z(r_i^k,k)$, $z(r_i^{k-1},k-1)$, and the Hessian $\hat{H}(k)$ in the matrix $U(k)$ (21) are needed to enable the cooperative Kalman filter.

1) Estimates of $z(r_i^k,k)$ and $z(r_i^{k-1},k-1)$: Since the sensor measurements $p(r_i^k,k)$ and $p(r_i^{k-1},k-1)$ are available in the measurement vector $P(k)$, one straightforward and simple way is to replace $z(r_i^k,k)$ and $z(r_i^{k-1},k-1)$ with the sensor measurements $p(r_i^k,k)$ and $p(r_i^{k-1},k-1)$, which is adopted in this paper.

2) Cooperative estimation of the Hessian: By time step $k-1$, we have obtained an estimate of $\hat{X}^+(k-1)$ from the cooperative Kalman filter. Using the computed estimates $\hat{X}^+(k-1)$ and $U(k-1)$, before the arrival of measurements at time step k, we can obtain a prediction for $X(k)$ as $\hat{X}^-(k)=A_{\hat{\theta}}(k-1)\hat{X}^+(k-1)+U(k-1)$. Here, we use subscript $(-)$ to indicate predictions and $(+)$ to indicate updated estimates. If we assume the number of sensor $N\geq 4$ and the formation is not co-linear, we have $P(k)=C(k)\cdot\hat{X}^-(k)+D(k)\hat{H}(k)$. The Hessian estimate can be solved by using the least mean square method, $\hat{H}(k)=\left(D(k)^T D(k)\right)^{-1}D(k)^T\left(P(k)-C(k)\hat{X}^-(k)\right)$.

D. Formation and motion control

Control laws for the velocities of the agents are required so that the mobile sensor network can move along a certain trajectory while maintaining a desired formation. Thus, the finite volume Ω_c, as well as the coefficients α_{AB}, α_{BC}, α_{CD}, α_{DA}, and α_{center} can be considered as constants. We view the entire formation as a deformable body. Thus, there are two parts of control: motion control and formation control. With the gradient estimates provided by the cooperative Kalman filter, the motion control for the agents can be easily realized by setting the velocities of the agents to be aligned with the estimated gradient direction. Thus, the mobile sensor network can achieve simultaneously parameter estimation and gradient climbing. Furthermore, there exists several results about the formation control for mobile agents [17] [19]. We omit the detailed design of control here due to space limitation. Interested readers can refer to [17] [19].

V. Recursive Least Square Estimation

A. The RLS method

In this section, we use the RLS method to iteratively update the estimate of θ based on the discretized model (12). We do this using the information state $\hat{X}(k+1)=[\hat{z}(r_c^k,k),\nabla\hat{z}(r_c^k,k),\hat{z}(r_c^k,k+1),\nabla\hat{z}(r_c^k,k+1)]^T$ obtained from the cooperative Kalman filter to calculate the temporal variations of the field value $\frac{\hat{z}(r_c^k,k+1)-\hat{z}(r_c^k,k)}{t_s}$. By combing the left terms of Equation (12), we denote the term $\hat{Y}(r_c^k,k)$ as $\hat{Y}(r_c^k,k)=\frac{\hat{z}(r_c^k,k+1)-\hat{z}(r_c^k,k)}{t_s}-v^T\nabla\hat{z}(r_c^k,k)$, where the hat notation indicates that $\hat{Y}(r_c^k,k)$ is the estimate from the cooperative Kalman filter. Note that the time index k is the same as the index in the cooperative Kalman filter.

Since the convergence of the cooperative Kalman filter has already been proved in [2], we can have

$$\hat{Y}(r_c^k,k)=Y(r_c^k,k)+\varsigma(r_c^k,k), \tag{26}$$

where $\hat{Y}(r_c^k,k)$ is the estimate of $Y(r_c^k,k)$, all elements of which come from the cooperative Kalman filter and $\varsigma(r_c^k,k)$ is a Gaussian noise with zero mean and bounded covariance. With the combination of Equation (12) and (26), the finite volume approximation model can be represented as

$$\hat{Y}(r_c^k,k)=\theta\cdot\frac{-1}{\Omega_c}[\sum_{i=1}^{4}\alpha_{\sigma_i}\cdot z(r_i^k,k)+\alpha_{center}\cdot z(r_c^k,k)]$$
$$+\varsigma(r_c^k,k)+e(r_c^k,k)=P_{\hat{z}}\theta+h(k), \tag{27}$$

where $h(k)=\varsigma(r_c^k,k)+e(r_c^k,k)+(P_z-P_{\hat{z}})\theta$, $P_z=\frac{-1}{\Omega_c}[\sum_{i=1}^{4}\alpha_{\sigma_i}\cdot z(r_i^k,k)+\alpha_{center}\cdot z(r_c^k,k)]$, and $P_{\hat{z}}=\frac{-1}{\Omega_c}[\sum_{i=1}^{4}\alpha_{\sigma_i}\cdot p(r_i^k,k)+\alpha_{center}\cdot\hat{z}(r_c^k,k)]$. The RLS parameter identification is based on minimizing the mean squared error criterion $J = E[h(k)^2]$, where $E[.]$ denotes the expectation value. Therefore, based on the cooperative filtering scheme, the diffusion coefficient can be directly estimated without the need of numerically solving the diffusion equation. Given an initial estimate for the diffusion coefficient, a simple application of the RLS method can iteratively update the estimate of θ. Following the canonical procedure of RLS estimation outlined in [20], we derive the following equations to update the estimate θ.

$$\begin{aligned} \hat{\theta}_k &= \hat{\theta}_{k-1} + g(k)\left(\hat{Y}(r_c^k, k) - P_{\hat{z}}\hat{\theta}_{k-1}\right); \\ g(k) &= \eta(k-1)P_{\hat{z}}^T\left[P_{\hat{z}}\eta(k-1)P_{\hat{z}}^T + R_e\right]^{-1}; \\ \eta(k) &= (I - g(k)\cdot P_{\hat{z}})\,\eta(k-1), \end{aligned} \tag{28}$$

where $g(k)$ is the estimator gain matrix, $\eta(k)$ is the estimation error covariance matrix, and R_e is the noise covariance.

In the above framework, we can observe that the proposed recursive cooperative filtering scheme is based on two subsystems: cooperative Kalman filtering subsystem (Equations (19) and (23)) and RLS subsystem in (28). In the cooperative Kalman filtering subsystem, assume that the parameter $\hat{\theta}_k$ is known, we run the cooperative Kalman filter to estimate the states based on the collected measurements. In the RLS subsystem, assume that the estimated states can track the true values, we employ the RLS method to iteratively update the estimate of θ. It should be noted that the convergence of the proposed closed loop recursive scheme heavily depends on the property that the convergence of the Kalman filter is independent of the estimated parameter $\hat{\theta}_k$, which is used in the Kalman filter in Equation (19). In other words, the estimated states from the cooperative Kalman filtering can successfully track the true values even though the estimated parameter $\hat{\theta}_k$ is biased. This part of convergence proof has been published in our previous work [2].

B. The bias analysis of the RLS method

We further provide the bias analysis of the RLS method and have the following proposition.

Proposition V.1 *Consider the RLS updated laws in* (28)*. Under the assumption that the modelling error $e(r_c^k, k)$ is an independent noise with zero mean and finite variance, the RLS algorithm produces a biased estimation of θ in the presence of Gaussian noise $\varsigma(r_c^k, k)$.*

Proof: The RLS algorithm in (28) gives rise to the estimate of θ as,

$$\hat{\theta} = E[P_{\hat{z}}^T P_{\hat{z}}]^{-1} E[P_{\hat{z}}^T \hat{Y}(r_c^k, k+1)]. \tag{29}$$

Since $\varsigma(r_c^k, k)$ and $e(r_c^k, k)$ are i.i.d. Gaussian noises, then

$$\begin{aligned} E[P_{\hat{z}}^T h(r_c^k, k+1)] &= E[P_{\hat{z}}^T \varepsilon(r_c^k, k+1)] + E[P_{\hat{z}}^T e(r_c^k, k)] \\ &\quad + E[P_{\hat{z}}^T (P_z - P_{\hat{z}})]\theta = E[P_{\hat{z}}^T (P_z - P_{\hat{z}})]\theta, \end{aligned} \tag{30}$$

which is generally not zero and yields a biased estimation of $\hat{\theta}$, even if we assume the modelling error $e(r_c^k, k+1)$ is a Gaussian noise, as follows

$$\hat{\theta} = \theta + E[P_{\hat{z}}^T P_{\hat{z}}]^{-1} E[P_{\hat{z}}^T (P_z - P_{\hat{z}})]\theta. \tag{31}$$

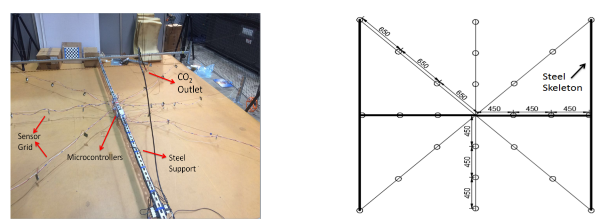
■

VI. Experiment

In this section, we introduce the design of a controllable CO_2 diffusion field in our lab. By validating the proposed algorithm in this real field, we demonstrate that the algorithm is robust under realistic uncertainties and disturbances.

A. Generating and visualizing a diffusion Field

A reference CO_2 diffusion field is produced in our lab in an area of $3.5 \times 3.5m^2$. When experiment begins, 15 CFH (ft^3/h) amount of CO_2 gas is released from an outlet 0.49 meters above the area for 8 minutes. Then the release stops and the gas diffuses freely for 10 minutes until the gas concentration in the room decreases back to normal values. Since CO_2 is a transparent and invisible gas, a sensor grid is assembled to measure the concentration of the gas over the area. We illustrate the structure of the sensor grid in Fig. 3, which consists of 24 CO_2 sensors, 8 ARM-mbed microcontrollers, and an 'H' shaped steel frame. In this experiment, the sensors are evenly distributed as an asterisk shape as indicated in Fig. 3. The microcontrollers are used to collect and store the data from the sensors and send them to a central computer. The 'H' shaped steel frame is built to support the sensor grid. We choose K-30 CO_2 sensors to capture the gas concentration. The range of the sensor measurement is $[0, 10000]$ ppm. The measuring frequency of the sensors is set to 0.5Hz, which can guarantee the successfully tracking of the dynamics of CO_2 gas. The diffusion process obtained from the real field is shown in Fig. 4. CO_2 begins diffusing at step k = 0 and ends at k = 625. The computational time step is 1 second.



(a) (b)

Fig. 3. The illustration of the sensor grid.

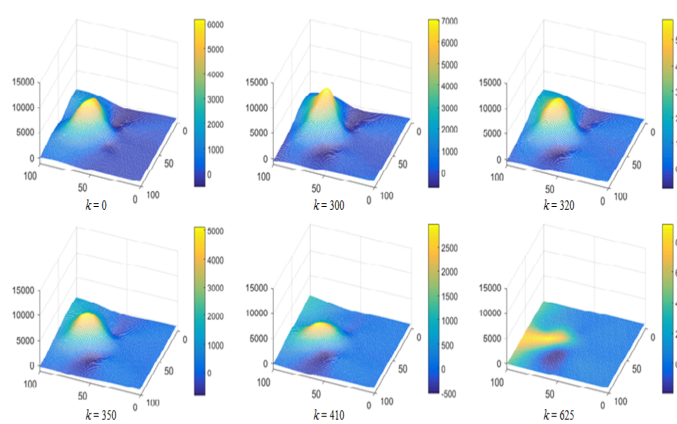


Fig. 4. The diffusion field collected and visualized by MATLAB.

B. Experimental results

We perform two different experiments for diffusion coefficient identification with four sensing agents deployed in the field. We choose two different starting points for the agents: northeast (NE) starting point and southeast (SE) starting point. The robots are controlled to move along the gradient direction of the field estimated from the cooperative Kalman filter while keeping a constant formation. In Fig. 5, the contours represent the level curves of the diffusion field, the colored stars represent the four sensing agents, the red line and blue line represent the trajectories of the center of the mobile sensor network staring from NE and SE, respectively. Since the field is pre-collected in MATLAB, it can be repeatably used for both experiments. As we can observe from the figure, the robots trace the gradient of the diffusion field in both experiments to find the diffusion source of the CO_2 gas, which is the point with the highest CO_2 concentration. Both of the two groups arrive at the source around step $k = 550$. While the mobile sensor network is searching for the source, it also achieves real-time identification of the diffusion coefficient by implementing the cooperative Kalman filter and the RLS. The estimation results of the diffusion coefficient are shown in Fig. 6. As we can observe from Fig. 6 that, the estimates of the parameter converge to stabilized values in both experiments. The two values differ by an amount of 0.1996. This difference may be caused by the influence of the velocity components of the CO_2 flow. Nevertheless, it is seen that the proposed algorithm is robust under realistic uncertainties and disturbances.

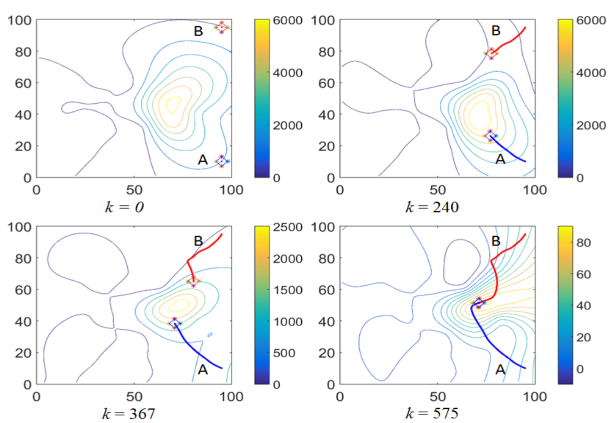


Fig. 5. The trajectories of the robots in the two experiments.

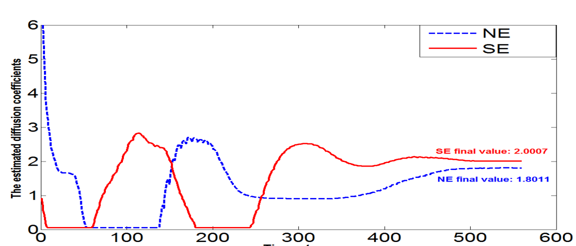


Fig. 6. The estimated diffusion coefficient.

VII. Conclusion

We propose a novel filtering scheme for performing online parameter estimation for advection-diffusion processes utilizing a mobile sensor network. By using the finite volume approximation, the proposed scheme can deal with the case when $N \geq 4$ agents are arranged in an arbitrary formation. Theoretical justifications are provided for the biased analysis of RLS. Experiment results based on a real CO_2 field show satisfactory performance. Future work includes extending the proposed algorithm to other types of PDEs.

References

[1] R. Ghez, *Diffusion Phenomena*. Kluwer Academic/ Plenum Publishers, 2nd edition, 2001.

[2] J. You, F. Zhang, and W. Wu, "Cooperative filtering for parameter identification of diffusion processes," in *Proc. of IEEE Conference on Decision and Control*, 2016, pp. 4327–4333.

[3] D. Uciński and M. Patan, "Sensor network design for the estimation of spatially distributed processes," *Int. J. Appl. Math. Comput. Sci.*, vol. 20, no. 3, pp. 459–481, 2010.

[4] L. Rossi, B. Krishnamachari, and C. C. J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Sensor and Ad Hoc Communications and Networks, IEEE SECON, First Annual IEEE Communications Society Conference*, 2004, pp. 460–469.

[5] D. Uciński, *Optimal measurment methods for distributed parameter system identification*. Boca Raton, FL: CRC Press, 2004.

[6] L. Z. Guo, S. A. Billings, and H. L. Wei, "Estimation of spatial derivatives and identificaiton of continuous spatio-temporal dynamical systems," *Internal Journal of Control*, vol. 79, no. 9, pp. 1118–1135, 2006.

[7] H. Li and C. Qi, "Modeling of distributed parameter systems for application - a synthesized review from time-space separation," *Journal of Process Control*, vol. 20, pp. 891–901, 2010.

[8] J. Droniou, "Finite volume schemes for diffusion equations: introduction to and review of modern methods," *Mathematical Models and Methods in Applied Sciences*, vol. 24, no. 8, pp. 1575–1619, 2014.

[9] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

[10] Z. Tang and U. Ozguner, "Motion planning for multitarget surveillance with mobile sensor agents," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 898–908, 2005.

[11] A. I. Mourikis and S. I. Roumeliotis, "Performance analysis of multirobot cooperative localization," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 666–681, 2006.

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

[13] M. A. Demetriou and I. I. Hussein, "Estimation of spatially distributed process using mobile spatially distributed sensor network," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 266–291, 2009.

[14] V. N. Christopoulos and S. Roumeliotis, "Adaptive sensing for instantaneous gas release parameter estimation," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 4450–4456.

[15] J. You and W. Wu, "Online passive identifier for spatially distributed systems using mobile sensor networks," *IEEE Transactions on Control Systems Technology*, 2017, accepted.

[16] M. A. Demetriou, N. A. Gatsonis, and J. R. Court, "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*, vol. 22, no. 3, pp. 853–867, 2014.

[17] F. Zhang and N. E. Leonard, "Cooperative control and filtering for cooperative exploration," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 650–663, 2010.

[18] D. Simon and T. L. Chia, "Kalman filtering with state equality constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 128–136, 2002.

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

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