

Parameter Identification of Spatial–Temporal Varying Processes by a Multi-Robot System in Realistic Diffusion Fields

Wencen Wu^{†*}, Jie You[‡], Yufei Zhang[‡], Mingchen Li[‡]
and Kun Su[‡]

[†]*San Jose State University, San Jose, CA 95192, USA*

[‡]*Rensselaer Polytechnic Institute, Troy, NY 12180, USA*

E-mails: jyouy@sjtu.edu.cn, yufeiizhang@foxmail.com, dennisli1234@gmail.com, sukun1045@gmail.com

(Accepted August 5, 2020)

SUMMARY

In this article, we investigate the problem of parameter identification of spatial–temporal varying processes described by a general nonlinear partial differential equation and validate the feasibility and robustness of the proposed algorithm using a group of coordinated mobile robots equipped with sensors in a realistic diffusion field. Based on the online parameter identification method developed in our previous work using multiple mobile robots, in this article, we first develop a parameterized model that represents the nonlinear spatially distributed field, then develop a parameter identification scheme consisting of a cooperative Kalman filter and recursive least square method. In the experiments, we focus on the diffusion field and consider the realistic scenarios that the diffusion field contains obstacles and hazard zones that the robots should avoid. The identified parameters together with the located source could potentially assist in the reconstruction and monitoring of the field. To validate the proposed methods, we generate a controllable carbon dioxide (CO₂) field in our laboratory and build a static CO₂ sensor network to measure and calibrate the field. With the reconstructed realistic diffusion field measured by the sensor network, a multi-robot system is developed to perform the parameter identification in the field. The results of simulations and experiments show satisfactory performance and robustness of the proposed algorithms.

KEYWORDS: Parameter identification; Distributed parameter systems; Spatial–temporal varying processes; Multi-robot system; Experimental validation.

1. Introduction

1.1. Motivation and related work

In many civilian and military applications, people need to measure, monitor, and analyze surrounding physical processes.¹ For example, to detect toxic and flammable gas, a monitoring and warning system is built both inside some buildings² and in outdoor environment.³ In planetary exploration missions, a special robot system is designed to detect specific biogenic gases in the atmosphere of the Martian surface.⁶ To better understand the physical processes and ambient environment, mathematical models are often used to provide further insights. For example, the diffusion phenomena are being studied in detail using advection–diffusion partial differential equations (PDEs).⁷ Those

* Corresponding author. E-mail: wencen.wu@sjsu.edu

spatial–temporal varying processes are often considered as distributed parameter systems (DPSs). Researchers present several estimation, identification, and control techniques for those DPSs.⁸ Specifically, parameter identification aims to estimate unknown parameters from observed data so that the predicted response of the system is close to the process observations. Various parameter identification methods have been developed to estimate key parameters in DPS, which could help in the reconstruction and monitoring of the processes.^{7,9,10} Compared to lumped systems, parameter identification for spatial–temporal varying systems is challenging because of the spatial parameter variability and the presence of the spatial derivatives in PDEs.¹¹

In order to collect information of the spatial–temporal varying processes, static sensor networks are often built to provide direct measurements of ambient environment for parameter identification in many applications.¹² However, there are often cases that the spatial domain of interest is so large that not enough sensor nodes are available to provide high-resolution data. In addition, when unpredicted environmental disaster happens, no pre-deployed sensor nodes are available in some areas of interest, and the areas are dangerous for humans to get into. In contrast, a mobile sensor network (MSN) consisting of a distributed collection of “intelligent agents” such as autonomous mobile robots with sensing capabilities is more practical and robust in those scenarios. These platforms can be equipped with different kinds of sensors, such as chemical sensor, ultrasonic sensor, and light detection and ranging (LiDAR). Due to the mobility, the mobile robots are able to move along pre-determined or online optimized trajectories to take measurements of the field. Therefore, increasing attentions have been paid to using mobile sensor networks to perform parameter identification in harsh environments. Among the existing algorithms, offline scheme is more commonly used for identification problems¹⁰ because of the high computational load caused by PDE-related algorithms. However, an online parameter estimation scheme is more desirable in scenarios when real-time information of the field is required. In our previous work,^{14,24,28} online parameter identification strategies have been proposed to identify the constant diffusion coefficient of a diffusion process using a group of mobile sensing agents.

Experiments are necessary for validating the proposed approaches. Due to the difficulties that come with generating and controlling a spatial–temporal varying field and constructing a large-scale static sensor network, most existing approaches have been only tested in computer simulations or in pre-collected and smoothed fields.^{15,21,23} To test the algorithms in realistic processes that may have unpredictable uncertainties and disturbances, several experiments have been designed using different robot systems for odor sensing problems including tracking a pollution plume or mapping an unknown area.^{1,13,16–19} For example, in ref. [18], researchers propose an experiment using five robots to map a multi-source environment with forced ventilation. In ref. [16], a huge robot called Gasbot is designed to help workers detect methane leak with a special tunable laser absorption spectroscopy sensor in different environments. Recently, unmanned aerial vehicles (UAVs) are also designed for the chemical mapping purpose. In ref. [19], researchers use a prototype hexacopter carrying an enose to perform gas leakage localization tasks. Apart from these works, some other interesting works are also mentioned in refs. [1,13,17]. However, there are still lack of experiments designed for parameter identification problems using multi-robot systems.

1.2. Contributions

In this work, we investigate the problem of parameter identification of a spatial–temporal varying process described by a general nonlinear PDE and validate the feasibility and robustness of the proposed algorithm using a group of coordinated mobile robots equipped with sensors in a realistic diffusion field. In our previous work,¹⁴ we developed a strategy that combines a cooperative Kalman filter with the recursive least square (RLS) estimation to achieve online parameter identification in diffusion fields. In this article, we first generalize the problem to nonlinear PDE and propose a parametrized model that represents the spatially distributed field, then extend the strategy to the nonlinear case. In the experiments, we focus on the diffusion field and consider the realistic scenarios, that is, in many cases, the field may contain obstacles that the robots should avoid colliding with and in other cases, some areas of the field have high concentration of the chemicals that could damage the robots and the onboard sensors. Therefore, we extend the strategy in ref. [14] to deal with those scenarios that the diffusion field may contain obstacles and hazard zones that the robots should avoid.

To validate the proposed strategy, we develop a multi-robot testbed with a controllable diffusion field under realistic uncertainties and disturbances. For this purpose, we generate a carbon dioxide

(CO₂) diffusion field in a lab environment, in which a static CO₂ sensor network is constructed. The static sensor network is able to measure and calibrate the diffusion field. The measurements collected by the static sensor network are then visualized in Matlab and processed offline to provide the identified parameters of the diffusion field, which are considered as nominal values for further comparison with the results using the multi-robot system. The pre-collected diffusion field is first used to test the proposed strategies in simulations. We then deploy a multi-robot system equipped with CO₂ sensors in the field to conduct the proposed parameter identification strategy. Experimental results show satisfactory performance and demonstrate the robustness of the strategy using the multi-robot system.

1.3. Paper organization

The rest of this article is organized as follows. Section 2 introduces the mathematical background, which includes the field dynamics, the sensing models, the identification algorithm for the spatial-temporal varying field, and the motion planning algorithms for the mobile multi-robot system. In Section 3, we introduce the experimental settings including the generation of the CO₂ field and the construction of the static sensor network. In Section 4, we present the simulation results using several mobile robots in the pre-collected realistic diffusion field measured by the static sensor network. In Section 5, we illustrate and analyze the proposed online parameter identification strategy using a mobile robot system equipped with CO₂ sensors in the diffusion field. Conclusions and future work follow in Section 6.

2. Mathematical Background

In this section, we first introduce the online parameter identification strategy developed in our previous work in refs. [14, 20], then generalize the algorithm to cases when there are obstacles and hazard zones in the field.

2.1. Problem formulation

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

$$\frac{\partial z(r, t)}{\partial t} = f(z(r, t), \nabla z(r, t), \nabla^2 z(r, t)), \quad r \in \Omega, \quad (1)$$

with the initial condition:

$$z(r, 0) = z_0(r), \quad r \in \Omega, \quad (2)$$

and the boundary condition:

$$z(r_b, t) = z_b(r_b, t), \quad r_b \in \partial\Omega, \quad (3)$$

where $z_0(r)$ and $z_b(r_b, t)$ are arbitrary initial condition and Dirichlet boundary condition, respectively. In Eq. (4), $z(r, t)$ is the field concentration, which is a continuous function in domain Ω , and ∇^2 is the Laplacian operator. The field concentration $z(r, t)$ depends on both spatial variable r and time variable t .

$f(\cdot)$ is an unknown nonlinear function. This type of nonlinear PDEs in Eq. (1) is widely used to describe physical and engineering phenomena such as heat process, population dynamics, chemical reactors, and fluid dynamics, ref. [20]. Specifically, the net movement of molecules or atoms is a diffusion process, which can be formulated as a parabolic PDE

$$\frac{\partial z(r, t)}{\partial t} = \theta \nabla^2 z(r, t) + v^T \nabla z(r, t), \quad r \in \Omega, \quad (4)$$

where θ is the diffusion coefficient and v is the flow velocity,

We deploy N mobile robots equipped with sensors in the field, and these N robots form a multi-robot system. Note that in most cases, the number of robots in a multi-robot system is usually much less than the number of sensors in a static sensor network. Since the measurements are often taken at discrete time instant, we model the measurement equation for i th robot as

$$p(r_i^k, k) = z(r_i^k, k) + \epsilon_i(k), \quad (5)$$

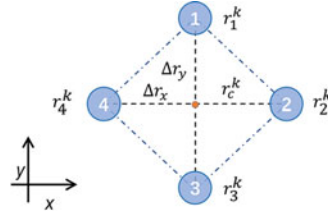


Fig. 1. A symmetric formation of four robots.

for $i = 1, \dots, N$, where i is the index of the robot. In Eq. (5), k denotes the discrete time instant, r_i^k is the location of the i th robot at time instant k , $z(r_i^k, k)$ is the field concentration at position r_i^k at time instant k , and $\epsilon_i(k)$ is assumed to be the measurement noise.

Assume that the robots are able to communicate with other robots in the multi-robot system as well as a central controller, for example, through bluetooth or wireless communication networks, to exchange their locations and measurements of the field. The problem is formulated as: design an online identification strategy to estimate the diffusion coefficient θ of the diffusion process (4) based on the measurements of the robots while they are moving in the field. More specifically, two subproblems need to be solved: (i) develop an online parameter identification algorithm to identify the unknown parameters in $f(\cdot)$ using the limited measurements taken by the robots along their trajectories; (ii) design formation control and motion planning algorithms for the multi-robot system taking into considerations of obstacles and hazard zones in the field.

2.2. Identification using multi-robot system

This section introduces the online identification algorithm using the multi-robot system. The algorithm consists of two major steps: a cooperative Kalman filter that provides state estimates of the field, and an RLS algorithm that utilizes the results from the cooperative Kalman filter to produce the online estimate of the unknown parameters.

Denote the position of the center of the multi-robot network as r_c^k , which is determined by $r_c^k = \frac{1}{N} \sum_{i=1}^N r_i^k$. Figure 1 illustrates a symmetric formation of four robots. The online identification algorithm will be based on the discretization of Eq. (1). Suppose the current time step is t_k . The temporal variations of the concentration can be approximated with finite difference as,

$$\left. \frac{\partial z(r_c, t)}{\partial t} \right|_{t=t_k} \approx \frac{z(r_c^k, k+1) - z(r_c^k, k)}{t_s}, \quad (6)$$

where t_s is the sampling interval. Apply the above finite difference to Eq. (1) gives,

$$z(r_c^k, k+1) \approx z(r_c^k, k) + t_s f(z(r_c^k, k), \nabla z(r_c^k, k), \nabla^2 z(r_c^k, k)). \quad (7)$$

Exact solutions for the nonlinear PDE (7) are difficult to obtain due to diverse nonlinearity, different structures, and complex boundary conditions.⁴ Therefore, we parameterize the nonlinear function $f(\cdot)$ in Eq. (7) by assuming that the unknown nonlinear function $f(\cdot)$ takes a form of polynomial. Several works have illustrated that the polynomial expression of $f(\cdot)$ can be a good approximation of the original model in Eq. (4).⁴ The polynomial form of Eq. (7) is given by the model,

$$\begin{aligned} z(r_c^k, k+1) = & z(r_c^k, k) + \sum_{i=1}^M \theta_i (t_s \phi_i(z(r_c^k, k), \nabla z(r_c^k, k), \nabla^2 z(r_c^k, k))) \\ & + e(r_c^k, k), \end{aligned} \quad (8)$$

where M denotes the order of the polynomial, θ_i is the coefficient of the i th polynomial term, and $\phi_i(z(r_c^k, k), \nabla z(r_c^k, k), \nabla^2 z(r_c^k, k))$ is the corresponding monomial, which is the product of different spatial derivatives $z(r, k)$, $\nabla z(r_c^k, k)$, and $\nabla^2 z(r_c^k, k)$. $e(r_c^k, k)$ is the approximation modeling error. $e(r_c^k, k)$ is a higher order term of the space sampling interval, which allows us to assume it as an independent noise sequence with zero mean and finite variance.^{20,28}

We observe that Eq. (8) is just a semi-discrete representation of the original continuous PDE (4). That is because a direct differentiation process of higher order spatially derivative terms such

as $\nabla z(r_c^k, k)$ and $\nabla^2 z(r_c^k, k)$ tends to amplify the effects of the noise.^{4,27} Therefore, different from existing lumped models that discretize each time and spatial derivative term,^{4,27} we only consider the time derivative discretization in our work. We will employ a cooperative Kalman filter to directly estimate the spatial derivative terms $\nabla z(r_c^k, k)$ and $\nabla^2 z(r_c^k, k)$ along the trajectory of the MSN. This part of work will be introduced in Section 2.4. In the following, we will use the hat notations to represent the estimates of the corresponding variables. Then, we can rewrite the above equation in a vector form,

$$\hat{z}(r_c^k, k+1) = \hat{\Phi}(r_c^k, k)\Theta + h(r_c^k, k), \quad (9)$$

where $\Theta^T = [1, \theta_1, \dots, \theta_M]$ is the parameter vector that we will identify, $h(r_c^k, k)$ is the error term consisting of discretization and estimation errors, and

$$\begin{aligned} \hat{\Phi}(r_c^k, k) = & [\hat{z}(r_c^k, k), t_s \phi_1(\hat{z}(r_c^k, k), \nabla \hat{z}(r_c^k, k), \nabla^2 \hat{z}(r_c^k, k)), \\ & \dots, t_s \phi_M(\hat{z}(r_c^k, k), \nabla \hat{z}(r_c^k, k), \nabla^2 \hat{z}(r_c^k, k))]. \end{aligned} \quad (10)$$

In Eq. (9), $\hat{z}(r_c^k, k+1)$ and $\hat{\Phi}(r_c^k, k)$ will be determined by the cooperative Kalman filtering in Section 2.4. In the following section, a cooperative Kalman filter will be designed to provide state estimates of the field based on the model (9).

For a diffusion process, Eq. (8) is simplified to

$$\hat{z}(r_c^k, k+1) = \hat{z}(r_c^k, k) + t_s(\theta \nabla^2 \hat{z}(r_c^k, k) + v^T \nabla \hat{z}(r_c^k, k)) + e(r_c^k, k), \quad (11)$$

And thus $\hat{\Phi}(r_c^k, k)$ becomes $[\hat{z}(r_c^k, k), t_s \theta \nabla^2 \hat{z}(r_c^k, k), t_s v^T \nabla \hat{z}(r_c^k, k)]$ and $\Theta^T = [1, \theta]$.

In order to identify Θ , a cooperative Kalman filter will be constructed, followed by the RLS algorithm. For the convergence of the discretization method, the sampling time t_s must satisfy the inequality $t_s \leq \frac{\Delta r_x^2 + \Delta r_y^2}{4\theta}$.²⁷

2.3. RLS estimation

Based on the parametrized model (9), we show how to estimate the unknown parameters using the RLS method. Let $\hat{\Theta}(k)$ be the estimate of Θ at time instant k . Given an initial estimate of $\hat{\Theta}(0)$, a direct application of the RLS method can iteratively update the estimate of Θ . The proposed RLS algorithm can be stated as follows:

$$\begin{aligned} \hat{\Theta}(k) = & \hat{\Theta}(k-1) + R(k) \hat{\Phi}^T(r_c^k, k) \cdot \\ & (\hat{z}(r_c^k, k) - \hat{\Phi}(r_c^k, k-1) \hat{\Theta}(k-1)), \\ R^{-1}(k) = & R^{-1}(k-1) + \hat{\Phi}^T(r_c^k, k) \hat{\Phi}(r_c^k, k), \end{aligned} \quad (12)$$

where $R(k)$ is the error covariance and

$$\begin{aligned} \hat{\Phi}(r_c^k, k-1) = & [\hat{z}(r_c^k, k-1), t_s \phi_1(\hat{z}(r_c^k, k-1), \nabla \hat{z}(r_c^k, k-1), \nabla^2 \hat{z}(r_c^k, k-1)), \\ & \dots, t_s \phi_M(\hat{z}(r_c^k, k-1), \nabla \hat{z}(r_c^k, k-1), \nabla^2 \hat{z}(r_c^k, k-1))]. \end{aligned} \quad (13)$$

We should point out that to run the recursive update law in Eq. (12), we require $\hat{\Phi}(r_c^k, k-1)$ and $\hat{\Phi}(r_c^k, k)$, which can be obtained from the cooperative Kalman filter introduced in the next section.

2.4. Cooperative Kalman filter

In this section, we develop a cooperative Kalman filter that provides necessary information needed to enable the RLS algorithm. From Eq. (12), we know that we require $\hat{\Phi}(r_c^k, k)$ and $\hat{\Phi}(r_c^k, k-1)$, which consist of the states $z(r_c^k, k)$, $\nabla z(r_c^k, k)$, $\nabla^2 z(r_c^k, k)$, $z(r_c^k, k-1)$, $\nabla z(r_c^k, k-1)$, and $\nabla^2 z(r_c^k, k-1)$. While the MSN is moving in the field, the field value along the trajectory of the formation center r_c evolves according to

$$\dot{z}(r_c, t) = \nabla z(r_c, t) \cdot \dot{r}_c + \frac{\partial z(r_c, t)}{\partial t}, \quad (14)$$

Substituting the PDE (4) into (14), we obtain

$$\dot{z}(r_c, t) = \nabla z(r_c, t) \cdot \dot{r}_c + f(z(r, t), \nabla z(r, t), \nabla^2 z(r, t)). \quad (15)$$

We also derive the total time derivative of $\nabla z(r_c, t)$ as

$$\dot{\nabla} z(r_c, t) = H(r_c, t) \cdot \dot{r}_c + \frac{\partial \nabla z(r_c, t)}{\partial t}, \quad (16)$$

where $H(r_c, t)$ is the Hessian matrix. To construct the cooperative Kalman filter, define the information state as $X(k) = [z(r_c^k, k), \nabla z(r_c^k, k), z(r_c^k, k-1), \nabla z(r_c^k, k-1)]^T$. In practice, sensors take measurements discretely with sampling interval t_s . By discretizing Eqs. (15) and (16), we obtain the state equation as the information state evolves according to the following equation:

$$X(k+1) = A(k)X(k) + U(k) + e(k), \quad (17)$$

where $e(k) = [e(r_c^k, k), 0, e(r_c^k, k-1), 0]^T$ represents the model error terms in Eq. (7). We denote the covariance matrix of $e(k)$ as $E[e(k)e(k)^T] = W$. The matrices $A(k)$ and $U(k)$ are defined by

$$A(k) = \begin{bmatrix} 1 & (r_c^{k+1} - r_c^k)^T & 0 & 0 \\ 0 & I_{2 \times 2} & 0 & 0 \\ 0 & 0 & 1 & (r_c^{k+1} - r_c^k)^T \\ 0 & 0 & 0 & I_{2 \times 2} \end{bmatrix},$$

$$U(k) = \begin{bmatrix} \sum_{i=1}^M \hat{\theta}_i (t_s \phi_i(z(r_c^k, k), \nabla z(r_c^k, k), \nabla^2 z(r_c^k, k))) \\ H(r_c^k, k)(r_c^{k+1} - r_c^k) \\ \sum_{i=1}^M \hat{\theta}_i (t_s \phi_i(z(r_c^k, k-1), \nabla z(r_c^k, k-1), \nabla^2 z(r_c^k, k-1))) \\ H(r_c^k, k-1)(r_c^{k+1} - r_c^k) \end{bmatrix},$$

where $H(r_c^{k-1}, k)$ is the Hessian matrix and $\hat{\theta}_i$ can be obtained by the RLS algorithm in Eq. (12).

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

$$\begin{aligned} z(r_i^k, k) &\approx z(r_c^k, k) + (r_i^k - r_c^k)^T \nabla z(r_c^k, k) \\ &\quad + \frac{1}{2} (r_i^k - r_c^k)^T H(r_c^k, k) (r_i^k - r_c^k), \\ z(r_i^{k-1}, k-1) &\approx z(r_c^k, k-1) + (r_i^{k-1} - r_c^k)^T \nabla z(r_c^k, k-1) \\ &\quad + \frac{1}{2} (r_i^{k-1} - r_c^k)^T H(r_c^k, k-1) (r_i^{k-1} - r_c^k). \end{aligned} \quad (18)$$

Let $P(k)$ be the vector that contains all measurements from all the agents at time k and $k-1$. Then, the measurement equation can be modeled as,

$$P(k) = C(k) \cdot X(k) + D(k) \hat{H}(k) + D(k) \varepsilon(k) + n(k), \quad (19)$$

where $\hat{H}(k)$ represents the estimate of the Hessian $H(k) = [H(r_c^k, k), H(r_c^k, k-1)]^T$ at the center r_c^k in a vector form and $\varepsilon(k)$ represents the error in the estimation of the Hessian matrices. Denote $E[n(k)n(k)^T] = R$ and $E[\varepsilon(k)\varepsilon(k)^T] = Q$. The matrices $C(k)$ and $D(k)$ are defined by

$$C(k) = \begin{bmatrix} 1 & (r_1^k - r_c^k)^T & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (r_N^k - r_c^k)^T & 0 & 0 \\ 0 & 0 & 1 & (r_1^{k-1} - r_c^k)^T \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & (r_N^{k-1} - r_c^k)^T \end{bmatrix},$$

$$D(k) = \begin{bmatrix} \frac{1}{2}((r_1^k - r_c^k) \otimes (r_1^k - r_c^k))^T & 0 \\ \vdots & \vdots \\ \frac{1}{2}((r_N^k - r_c^k) \otimes (r_N^k - r_c^k))^T & 0 \\ 0 & \frac{1}{2}((r_1^{k-1} - r_c^k) \otimes (r_1^{k-1} - r_c^k))^T \\ \vdots & \vdots \\ 0 & \frac{1}{2}((r_N^{k-1} - r_c^k) \otimes (r_N^{k-1} - r_c^k))^T \end{bmatrix},$$

where $\otimes$ is the Kronecker product. Note that the Hessian $\hat{H}(k)$ in Eq. (19) are needed to enable the cooperative Kalman filter and provide the states $\nabla^2 z(r_c^k, k)$ and $\nabla^2 z(r_c^k, k-1)$. By designing a one-step filter,^{20,22} $\hat{H}(k)$ can be estimated. We omit this part here due to page limitation. Given (17) and (19), the equations for the cooperative Kalman filter can be readily constructed.²² For more details about the cooperative Kalman filter, please refer to our previous works.^{20,28}

Remark 1. The configuration of the robots depends on the spatial variations of the field. If the field values change slowly over a spatial domain, the distance between individual robots needs to be larger to capture the spatial variation that enables the gradient estimates produced by the cooperative Kalman filter. On the contrary, if the field values change significantly over a spatial domain, the robots need to stay closer to increase the accuracy of the state estimation. The optimal formation can be determined in real-time by solving an optimization problem.²² Since the main focus of this work is to validate the proposed parameter identification algorithm in realistic diffusion fields, we choose the fixed symmetric formation using four robots to simplify the construction of the cooperative Kalman filter.

2.5. Motion planning

In this section, we introduce motion planning algorithms for the multi-robot system by taking into consideration of the obstacles in the field and hazard avoidance zones. We first follow the approach in refs. [22, 29], which views the multi-robot system as a deformable body and uses Jacobin transform to decouple the motion control and the formation control of the center of the multi-robot system. Interested readers can refer to ref. [22] for the details about the formation control part. Since the formation control law converges exponentially as demonstrated in refs. [22, 28], in the following discussion, we assume that the formation control is achieved and focus on the motion control for the center of the multi-robot system. Three aspects are considered in the motion control: moving along information-rich trajectories, avoiding obstacles, and avoiding hazard zones. The resulting trajectories can prevent the multi-robot system from colliding with obstacles and entering areas with high concentration of corrosion and radioactive chemicals.

Many results have shown that the performance of parameter estimation depends on the locations or trajectories of the sensors.⁹ In ref. [20] we developed a Lyapunov redesign-based online trajectory planning algorithm for the multi-robot system so that the robots can use local real-time information to guide them to move along information-rich paths that can improve the performance of the parameter identification and state estimation in a spatial-temporal varying field described by (1). For a diffusion field (4), it is proved in ref. [9] that the D-optimization criterion, which is the determinant of the Fisher information matrix, reaches its maximum value when $r = r_0 - vt$, where $r_0 = (x_0, y_0)$ is the source position and $r_0 - vt$ corresponds to the time-varying location with maximum field value. Since the gradient direction is associated with the maximum convergence rate to the maximum location, we control the multi-robot system to move along the gradient direction in this article. We realize that in fields with multiple sources, gradient following will lead the robots to local minima; thus, we developed a reinforcement learning-based path planning strategy to deal with the problem of multiple sources. Interested readers can refer to ref. [29] for details.

Potential field method^{5,25,26} is used to achieve obstacle/hazard zone avoidance and gradient following, where the repulsive potentials are determined by the obstacles and the boundaries of the hazard zones in the field and the attractive potentials are determined by the field gradient

$$F^{attr}(r_c) = -\nabla z(r_c), \quad (20)$$

where $F^{attr}(r_c)$ represents the potential force applied to the center of the multi-robot system r_c . The gradient of the field $\nabla z(r_c)$ at the center r_c is directly obtained from the output of the cooperative Kalman filter designed in Section 2.4.

2.5.1. Obstacle avoidance. A repulsive potential is designed for the purpose of obstacle avoidance. Suppose there are M obstacles in the field, the repulsive force F_{rep}^i from the i th obstacle is given by:

$$\begin{aligned} F_i^{rep}(r_c, r_i^{obst}, l) &= \nabla U_i^{obst}(r_c, r_i^{obst}, l) \\ &= c_{obst} \cdot \log\left(\frac{|r_c - r_i^{obst}|}{l}\right), \end{aligned} \quad (21)$$

where U_i^{obst} defines the potential field, l is the sensing range of the robots, $r_i^{obst} = (x_i^{obst}, y_i^{obst})$ is the position of the i th obstacle, and c_{obst} is a constant. Since we can easily define a boundary at $\log(1) = 0$, we use a logarithmic function instead of a commonly used term $\frac{1}{r_c}$ in the potential function. Beyond this boundary, we set the repulsive force to 0, to simulate the limited sensing range of the robots. Given the attractive force (20) and the repulsive force (21), the total force applied to the center of the formation can be obtained as

$$F^{obst}(r_c) = c_{attr} \cdot F^{attr}(r_c) + c_{obst} \cdot \sum_{i=1}^M F_i^{rep}(r_c, r_i^{obst}, l), \quad (22)$$

where c_{attr} is a weighting constant.

2.5.2. Hazard zone avoidance. We define a hazard boundary $\Gamma(r) = \{r | z(r) = z^H\}$, which can be seen as a contour line with a given field concentration z^H . However, the sensors can neither directly measure the contour graph nor the distance between the formation center r_c and the contour. Consider r_H as the closest position on the contour line to the formation center r_c , that is, the vector $r_H - r_c$ is perpendicular to the tangent vector of the contour line passing through r_H . In order to obtain the repulsive force that utilizes the distance between the formation center and the contour, we use Taylor expansion to the first order to approximate $z(r_H)$ with respect to the formation center r_c as

$$z(r_H) = z(r_c) + \nabla z(r_c) \cdot (r_H - r_c) + O((r_H - r_c)^2), \quad (23)$$

where $O(\cdot)$ represents higher order terms, which are treated as noise in simulations and experiments. The field value and gradient at the formation center are provided by the cooperative Kalman filter. Given Eq. (23), the shortest distance between formation center and the contour line $|r_c - r_H|$ can be calculated. Thus, a repulsive force F^{rep} is generated using the same model in Eq. (21) as

$$F^{rep}(r_c, r_H, h) = c_{haz} \cdot \log\left(\frac{|r_c - r_H|}{h}\right), \quad (24)$$

where c_{haz} is a weighting constant and h is a scaling quantity similar to the sensing range in Eq. (21). Given the attractive force (20) and the repulsive force (24), the total force applied to the center of the formation in the hazard avoidance case can be obtained as

$$F^{haz}(r_c) = c_{attr} \cdot F^{attr}(r_c) + c_{haz} \cdot \log\left(\frac{|r_c - r_H|}{h}\right). \quad (25)$$

Based on the above design, for the purpose of simultaneous obstacle avoidance, hazard zone avoidance, and gradient following, we obtain the control law applied to the formation center as

$$F(r_c) = c_{attr} \cdot F^{attr}(r_c) + c_{obst} \cdot \sum_{i=1}^M F_i^{obst}(r_c, r_i^{obst}, l) + c_{haz} \cdot \log\left(\frac{|r_c - r_H|}{h}\right). \quad (26)$$



Fig. 2. The regulator and flowmeter mounted on a gas cylinder.

Remark 2. For the motion planning in the spatial–temporal varying field described by the nonlinear PDE (1), the attractive force can be replaced by the terms developed in ref. [20]. The repulsive forces can remain the same for the purposes of obstacle avoidance and hazard zone avoidance.

3. CO₂ Diffusion Field Construction and Calibration

In order to validate the proposed algorithms in realistic diffusion environment, we first generate a controllable carbon dioxide (CO₂) diffusion field and build a static sensor network to measure and calibrate the field. We then perform simulations using several collaborating robots in the pre-collected and smoothed realistic field to validate the parameter identification, obstacle avoidance, and hazard zone avoidance algorithms. Furthermore, we deploy four mobile robots equipped with CO₂ sensors in the diffusion field and conducted experiments in it. In this section, we first introduce the construction of the diffusion field and the static sensor network.

3.1. The controllable CO₂ diffusion field

There are lots of physical phenomena following a diffusion equation as described in Section 2.1. We choose the CO₂ gas for reasons that (1) CO₂ gas is non-flammable in common conditions, and non-toxic when inhaled; (2) a significant diffusion process lasts for about 1.5 min in our experimental field (16 × 16 square feet), which is compatible with the speed of the Khepera robots that we use in the experiments taking into consideration of the response time of the CO₂ sensors; and (3) the density of CO₂ gas is heavier than the standard air density at 101.325 kPa (1 atm). As a result of this, CO₂ may deposit near the ground, and the flow in the experiment can be diffusion dominated instead of convection dominated.

A cylinder of food grade high-pressure CO₂ gas is used to generate the diffusion field, and a single-stage regulator and a flowmeter are mounted on the cylinder to control the flow velocity as shown in Fig. 2. During the experiments, CO₂ gas is emitted for 1 min with a flow rate at 0.566 m³/h (20 cfh). At this rate, 0.009 m³ of CO₂ gas can flow into the field and create a high-concentration field with a radius of about 1 m. When the gas release is stopped, the CO₂ gas diffuses freely until its concentration drops back to normal level.

3.2. Static sensor network construction

We build a static sensor network in the diffusion field for the following reasons: (1) The pre-collected data by the static sensor network can be smoothened and visualized in Matlab, which help us better understand the dynamics of the realistic diffusion field and allow us to verify the proposed algorithms in simulations prior to real experiments. (2) Based on the data, offline parameter identification algorithms can be applied to generate estimates of the diffusion coefficient, which will be used as the nominal value to be compared with the parameter identification results produced by the proposed online algorithms using mobile robots.

The structure of the static sensor network is illustrated in Fig. 3. An “H” shaped steel frame is built to support the sensor grid, which is 15 cm above the ground. Twenty-four CO₂ sensors are installed on the eight arms of the steel frame arranged in an asterisk shape. On each arm, three CO₂ sensors

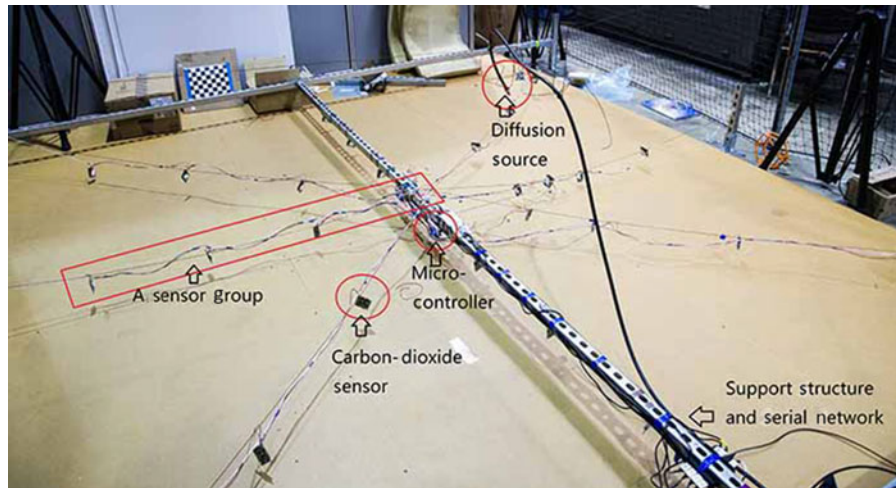


Fig. 3. The static sensor network in the CO₂ diffusion field.



Fig. 4. A K-30 CO₂ sensor.

are evenly distributed and controlled by an LPC-1768 micro-controller. The distance between two sensors on each arm is about 65 cm. In the sensor network, the micro-controllers are used to collect and store data from the sensors, establish a real-time connection with a central computer, send the data to the central computer, and handle some potential sensor errors. We choose the type K-30 CO₂ sensors (Fig. 4) to measure the gas concentration. The range of the sensor measurement is [0,10,000] ppm and the measuring frequency is set to 0.5 Hz, which can guarantee the successful tracking of the dynamics of CO₂ gas.

3.3. CO₂ field calibration

The procedure of generating and visualizing a diffusion field is as follows. During the diffusion process of CO₂ gas, sensors measure the gas concentration at the fixed locations. The micro-controllers then obtain the data from sensors and send the data to Matlab, which is running on the central computer. Matlab then reproduces the diffusion process by interpolating the field values at every discrete time instant using a biharmonic equation. The computation of the fitting model is fast enough for real-time virtualization. Fig. 5 illustrates the diffusion field measured by the static sensor network and visualized in Matlab. The CO₂ gas is first emitted for about 1 min, then the diffusion process starts at time $t = 0$ and ends at around $t = 83.65$ s. The slight rise of the concentration measurement in the second subfigure at $T = 20.54$ s is due to the response time of the sensors. In addition, we observe that at the boundary of the surface, the concentration drops below 0 ppm, which is impossible in physical world. This is because of the errors introduced by the fitting algorithms. In simulations and experiments, we will control the robots to avoid getting too close to the boundaries.

Based on the measurements collected by the static sensor network, an RLS algorithm is implemented to minimize the cost function

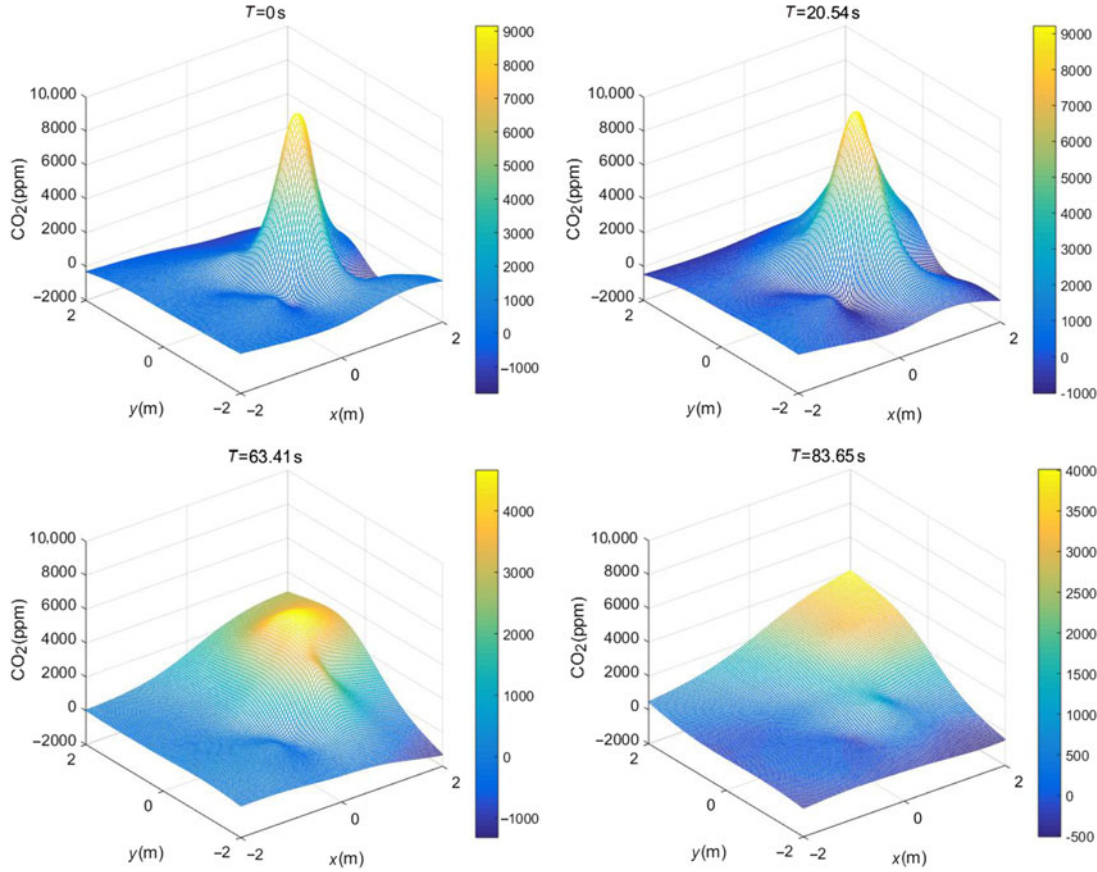


Fig. 5. The diffusion process of the CO₂ field measured by the static sensor network.

$$J(\theta) = \frac{1}{2} \sum_{k=0}^T \|\mathbf{p}(k) - \hat{\mathbf{z}}(r, k, \hat{\theta})\|^2, \quad (27)$$

where $\mathbf{z}(r, k, \theta)$ is the vector of the true field values at the fixed sensor locations at time step k , the estimate of which $\hat{\mathbf{z}}(r, k, \hat{\theta})$ is obtained by solving the diffusion Eq. (4) with the estimated diffusivity $\hat{\theta}$. $\mathbf{p}(k)$ is an N ($N = 24$ in our case) dimensional observation vector that consists of the measurements from all the N sensors in the field at step k as

$$\mathbf{p}(k) = \mathbf{z}(r, k, \theta) + \epsilon(k), \quad (28)$$

where $\epsilon(k)$ is the measurement noise vector.

The parameter identification result is shown in Fig. 6. As it can be observed from the figure, the estimated diffusivity converges to $\hat{\theta} = 0.2392$. Because this value is obtained using the global information from the 24 sensors, we consider this value as the nominal value for θ in the experimental field, and we will use this nominal value to verify the performance of the proposed online parameter identification algorithms.

Remark 3. In this work, we generate the diffusion field by first emitting the CO₂ gas for 1 min, then letting the gas diffuse freely afterwards. Note that if the gas emission continues, depending on different boundary conditions, the field would behave differently. For example, If we have an infinite field, the diffusion phenomenon would continue and the algorithm would still work in this situation since the field concentration is mainly governed by the diffusion process. If the field is constrained, for example, the gas emits in a small room, then as the gas emission goes on, the concentration of the gas in the room would keep increasing until there is no diffusion in the field. In this case, it is hard

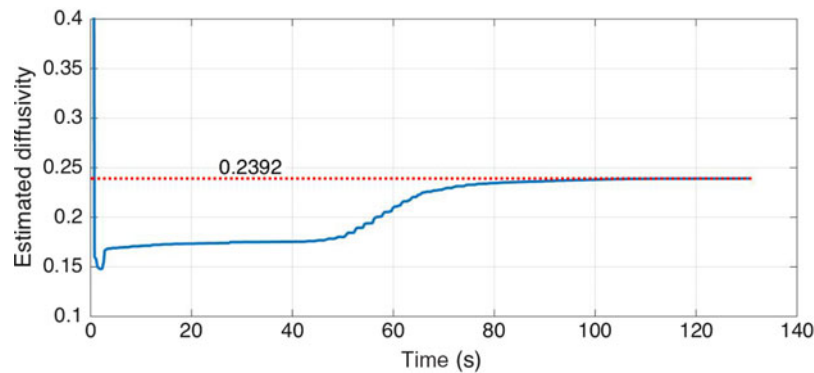


Fig. 6. Estimated diffusivity using the static sensor network. The red dashed line is the nominal value of the diffusivity and the blue solid line is the estimated diffusivity.

to evaluate the algorithms since the assumption that the field is spatial–temporal varying is no longer satisfied.

4. Simulation Results

In this section, we validate the proposed algorithms in simulations in the pre-collected diffusion field. Compared with simulations in a virtual field generated by a group of pre-selected parameters, simulations in realistic diffusion fields allow us to validate algorithms under realistic unknown disturbances, boundary and initial conditions, as well as other uncertainties that could not be provided by the virtual fields.

We first simulate the online parameter identification algorithm introduced in Section 2.2 using four collaborating mobile robots in the field without obstacles and hazard zones. As seen in Fig. 7, the four blue circles represent the four robots, the bold orange line is the trajectory of the formation center of the four robots, and the background contour graph represents the current field concentration. The robot formation starts from a randomly selected point on the left part of the field (subfigure 1), tracks the gradient of the field based on the gradient estimates provided by the cooperative Kalman filter, and finally reaches the location with the highest concentration (subfigure 4). The diffusion coefficient is estimated while the robots are moving in the field based on the proposed algorithm, which is shown in Fig. 8. We can observe that the identified diffusion parameter converges to the value of $\theta = 0.25$ with the steady-state error 0.0108 compared to the nominal value $\theta = 0.2392$.

We then validate the online parameter identification algorithm when obstacles are present in the field assuming that the obstacles do not interact with the surrounding chemical field, that is, the presence of the obstacles would not change the dynamics of the field. As shown in Fig. 9, we place several stationary round-shaped obstacles in the original diffusion field and apply the control law in Eq. (22) to the center of the formation. The robots start from the same place as in the previous simulation and reach the accessible highest concentration area at the end of the simulation. This initial condition makes the trajectories in two simulations shown in Figs. 7 and 9 comparable with each other. The identified diffusion coefficient is shown in Fig. 10, which also converges to a value around 0.25 with the steady-state error 0.0108.

In situations when the robots are vulnerable in high concentration areas, for example, the chemical field may be radioactive or corrosive, hazard avoidance algorithms are necessary. We simulate this scenario and show the results in Figs. 11 and 12. In Fig. 11, the high concentration area is surrounded by the bold blue contour lines, and the same static obstacles are sketched in grey circles as in the previous simulation. The control law in Eq. (26) is applied to the formation center. Figure 11 shows that the robots not only avoid the obstacles but also stay away from the high concentration area, due to the existence of our hazard avoidance algorithm. Figure 12 illustrates that the identified diffusion coefficient converges to 0.245 with the steady-state error 0.0058.

5. Experiment Result

In this section, we introduce the mobile robot testbed that is used to validate the proposed online parameter identification algorithm and present the experimental results.

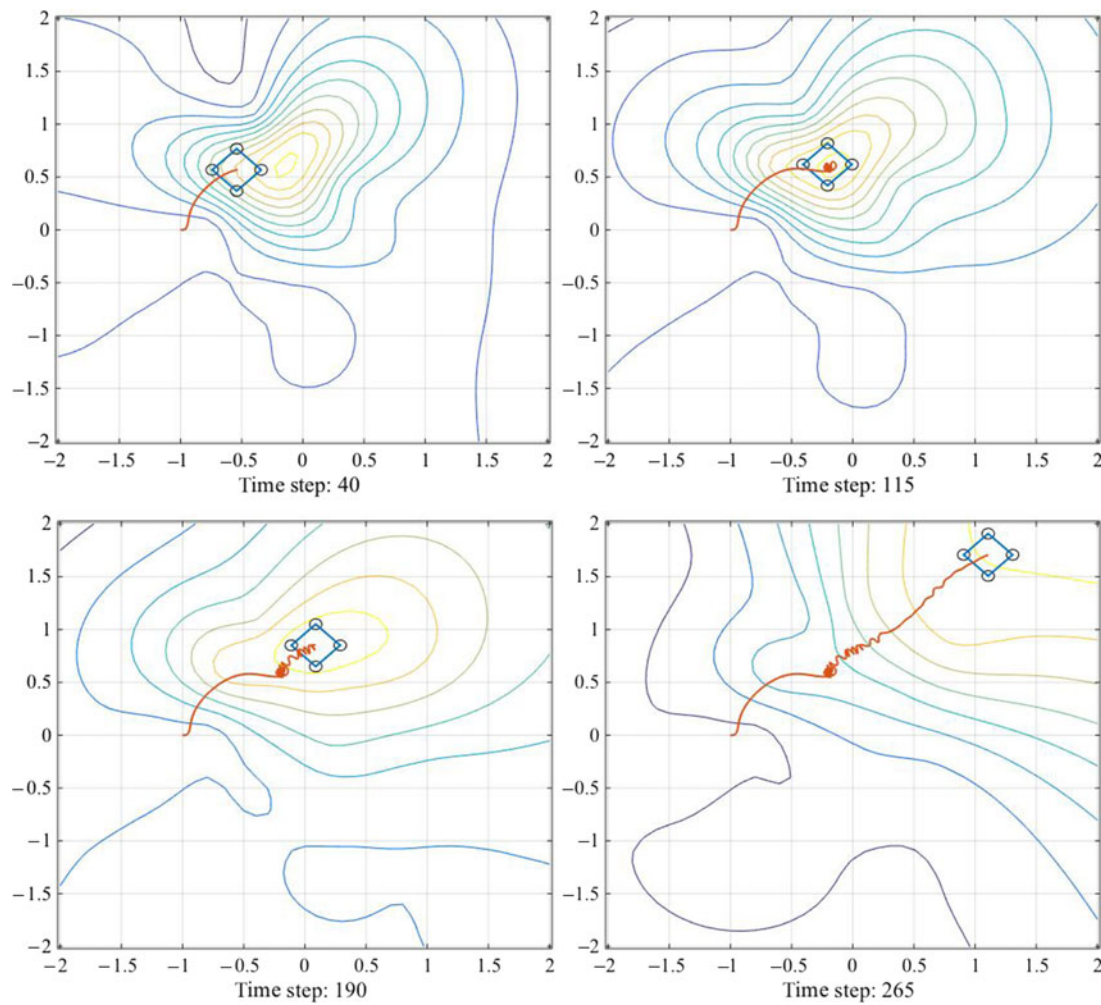


Fig. 7. Four robots perform online parameter identification while moving in the diffusion field in simulation. The four circles represent four robots. The red solid line is the trajectory of the center of the multi-robot system.

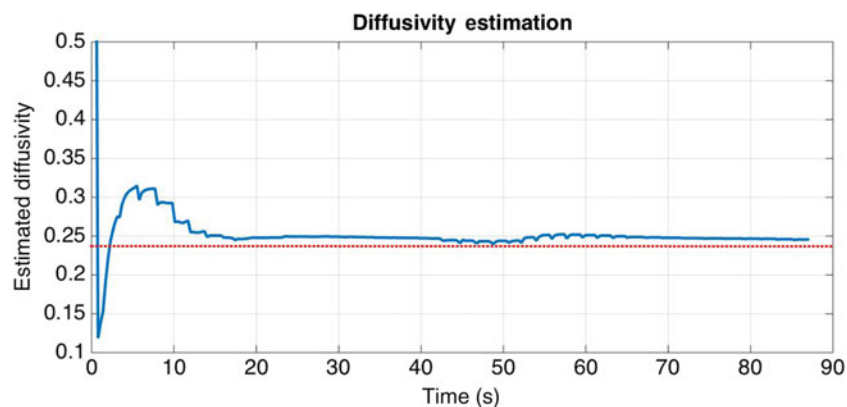


Fig. 8. The identified diffusion coefficient in the field without obstacles and hazard zones. The red dashed line is the nominal value of the diffusivity and the blue solid line is the estimated diffusivity.

5.1. Mobile robot system

The testbed consists of a central computer, four Khepera IV robots equipped with CO₂ sensors, an OptiTrack motion-capture system that is used to provide real-time information about the locations

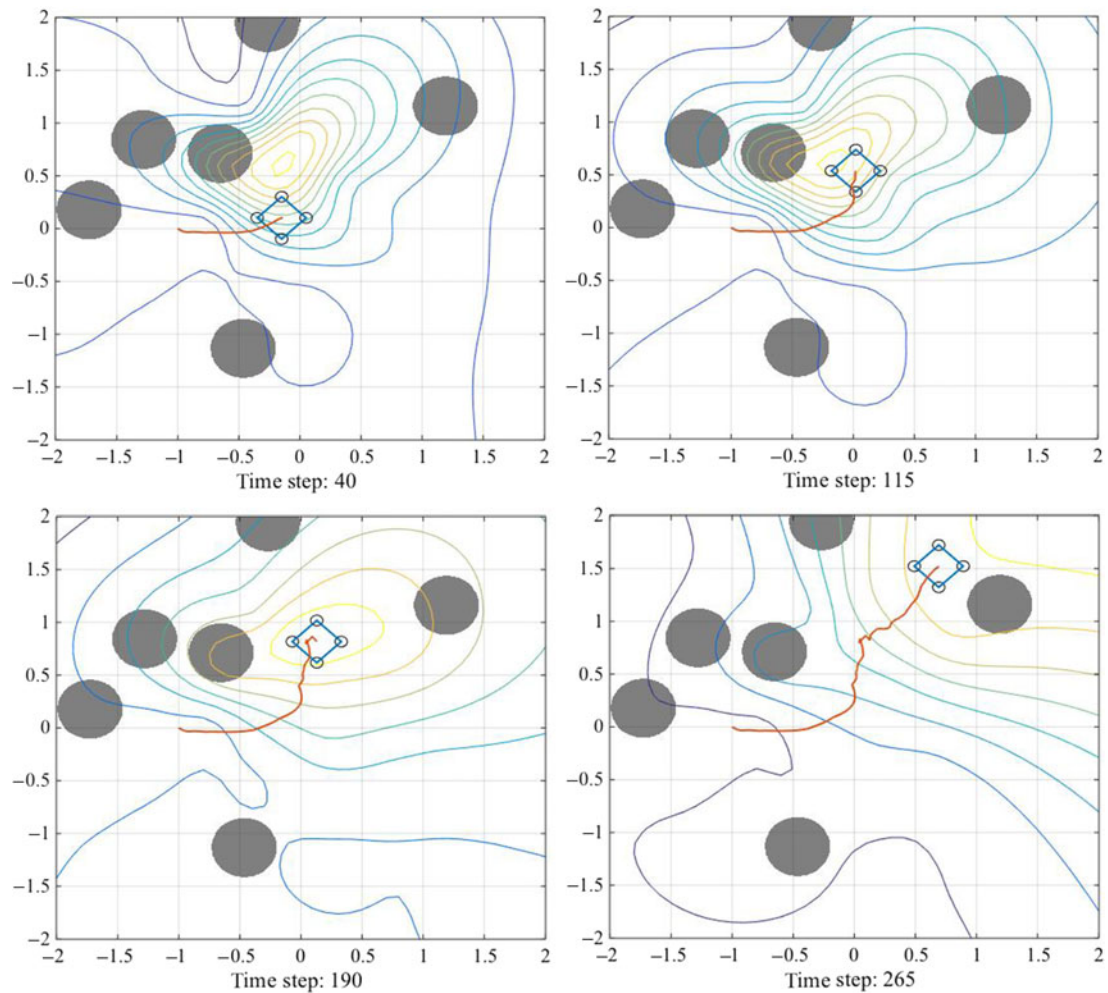


Fig. 9. Four robots perform online parameter identification while moving in a diffusion field with obstacle avoidance algorithm in simulation. The four circles represent four robots. The red solid line is the trajectory of the center of the multi-robot system.

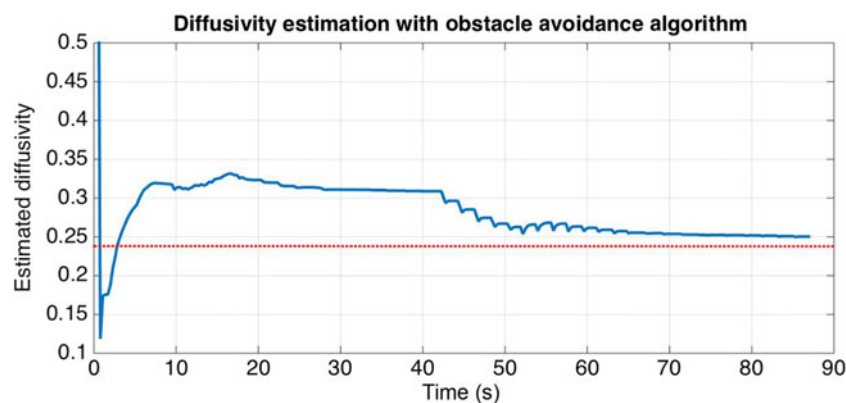


Fig. 10. The identified diffusion coefficient in the field with obstacles. The red dashed line is the nominal value of the diffusivity and the blue solid line is the estimated diffusivity.

and orientations of the robots, and the facilities used to generate the CO_2 field, as illustrated in Fig. 13.

Khepera IV robots are round mobile robots with two differential drive wheels and two sliding supports. On each robot, a group of controller, transmitter, and CO_2 sensor are installed as shown

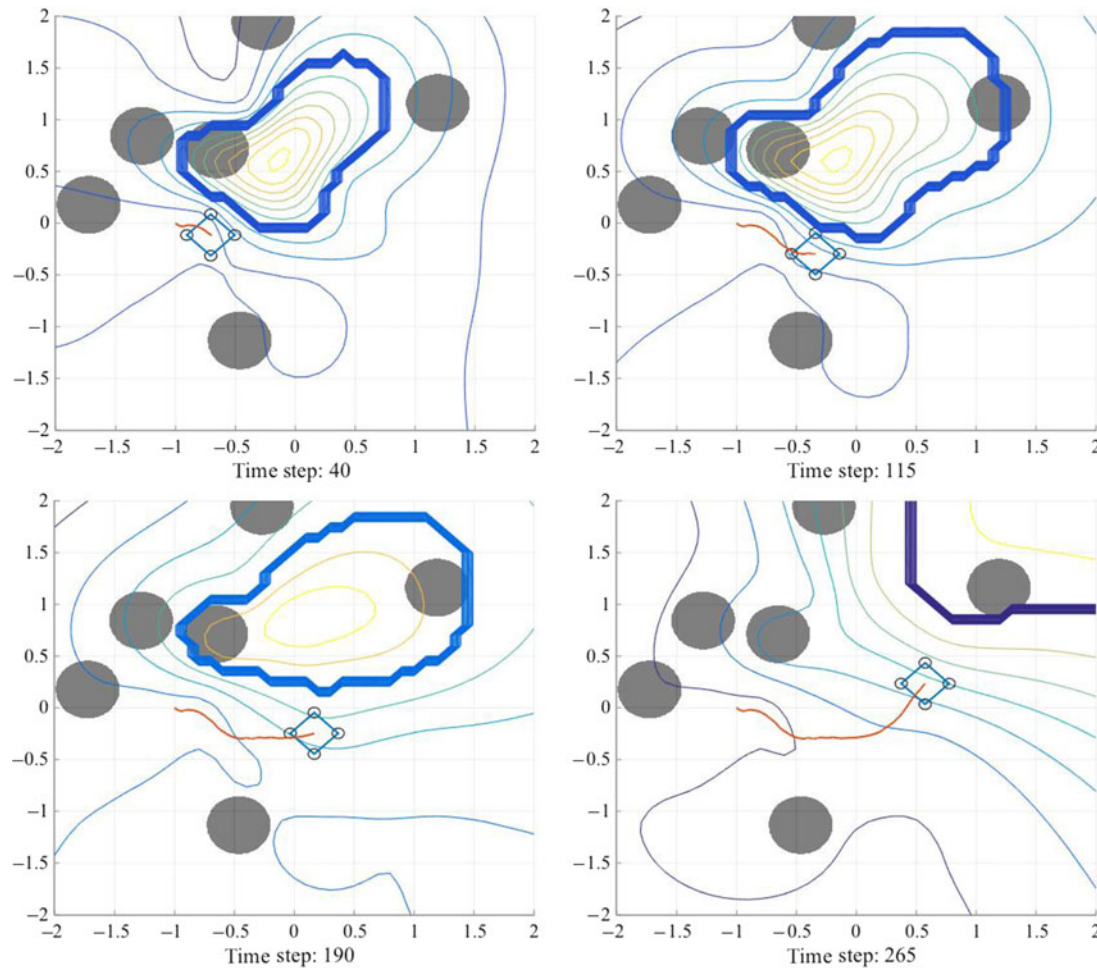


Fig. 11. Four robots perform online parameter identification while moving in a diffusion field with hazard zone avoidance algorithm in simulation. The four circles represent four robots. The red solid line is the trajectory of the center of the multi-robot system.

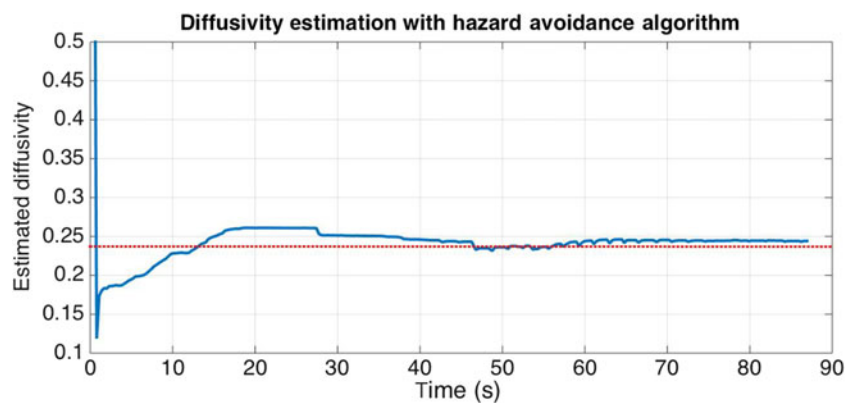


Fig. 12. The identified diffusion coefficient in the field with hazard zones. The red dashed line is the nominal value of the diffusivity and the blue solid line is the estimated diffusivity.

in Fig. 14. COZIR GC-0008 (Fig. 15) CO₂ gas sensor is selected as the onboard CO₂ sensor with a sampling rate of 20 Hz, which is significantly faster than the sampling rate of the K-30 sensor used in the static sensor network. The GC-0008 sensor also has a wide sensing range from 0 to 10,000 ppm with a 3% error rate as described in the data sheet. In addition, the GC-0008 sensor has a faster response time than the K-30 sensor. All the improved performance of the GC-0008 sensor

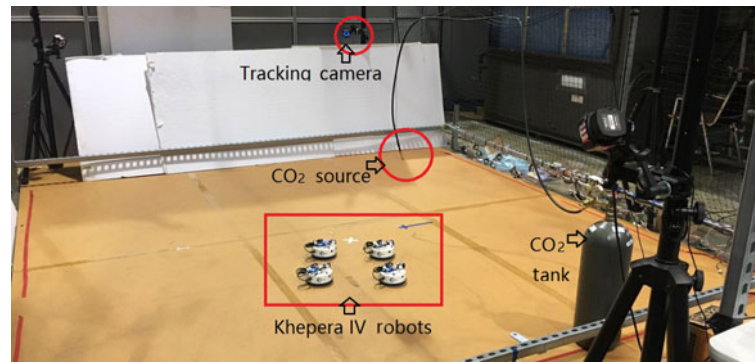


Fig. 13. The multi-robot testbed.

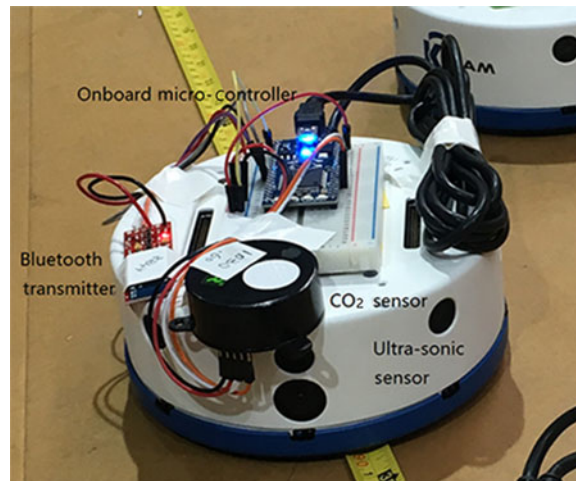


Fig. 14. A Khepera IV robot with a CO₂ sensor.



Fig. 15. A GC-0008 CO₂ sensor is in the same scale as an SD card.

is necessary for the multi-robot system since the online parameter identification algorithm requires real-time measurements while the robots are moving in the field.

A central controlling Matlab program is running on a central computer. In the Matlab program, the motion-capture services, data collecting services, the identification algorithm, the motion planning algorithm, and the control algorithms for four Khepera robots are running step by step sequentially. First, the motion-capture system sends the current locations and orientations of the robots to the computer. Meanwhile, the program obtains the corresponding gas concentration at those locations

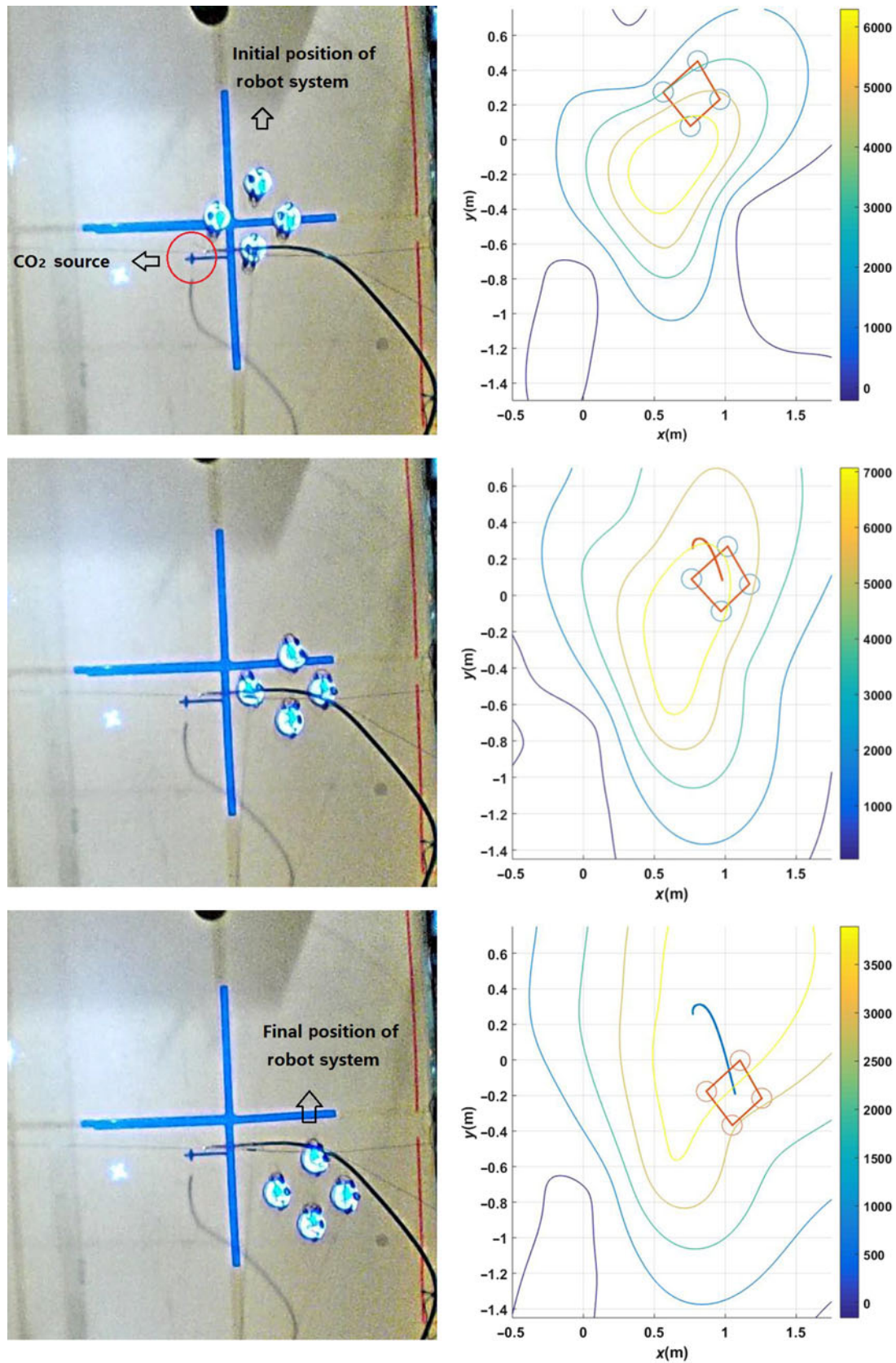


Fig. 16. Trajectories of the robots in one experiment (at $T = 0$ s, $T = 49.84$ s, and $T = 85.48$ s) in the realistic field (left) and pre-collected field (right).

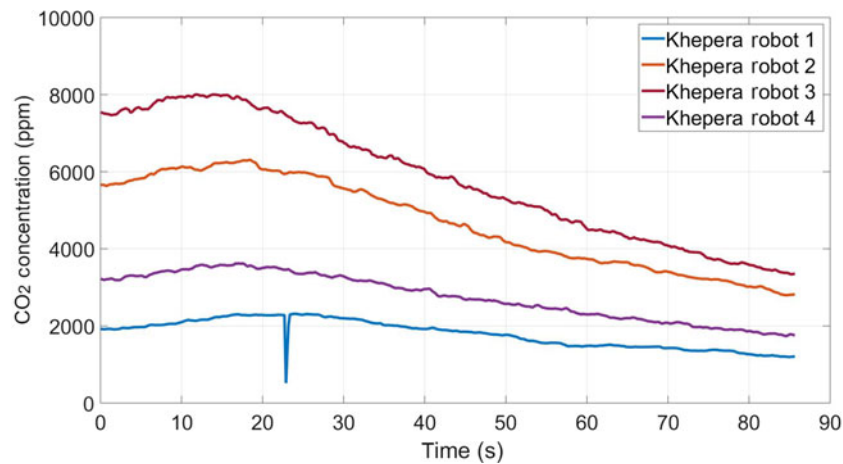


Fig. 17. CO₂ concentration measurements by the four Khepera robots.

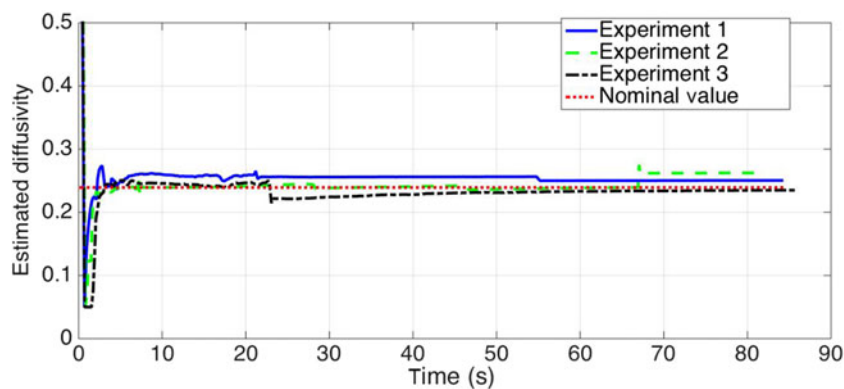


Fig. 18. The estimated CO₂ diffusivity in two experiments.

from the micro-controllers of the robots. Given these information, the central computer calculates new locations and orientations of robots for the next time step and sends the corresponding moving distances and turning angles back to the robots. After this, the robots execute the instructions and move to the new locations according to the received commands within a closed loop control strategy.

Four Khepera IV robots are deployed in the experimental field to implement the proposed online parameter identification algorithm. In the experiment, all the settings are kept the same as in the experiment using the static sensor network in Section 3.3. Figure 16 illustrates the trajectories of the four Khepera IV robots in the realistic CO₂ diffusion field (left column) in one experiment and the trajectories of four simulated mobile robots in the pre-collected field by the static sensor network (right column). These groups of figures are captured at three different time instants of the experiment: $T = 0$ s, $T = 49.84$ s, and $T = 85.48$ s (final time). Comparing those trajectories, especially shown in Fig. 16, our robot systems rarely finally approach the CO₂ source, though the robot system is moving toward the source location by the gradient descent algorithm. This indicates the diffusivity estimation process does not depend on the trajectories of our robots. If the robots follow the gradient direction, the estimation process may converge faster, not better. This feature is useful when the robot system is not able to follow the gradient direction. For example, we can still reach an accurate estimation when different navigation algorithms and obstacle avoidance algorithms are running together in the multi-robot system.

Figure 17 illustrates the sensor readings by the four robots in the experiment (Experiment 3) and Fig. 18 show the identified diffusion coefficient in three separate experiments with the robots starting from different initial positions. The sudden drop of the CO₂ concentration shown in Fig. 17 is caused by a failure of the sensor installed on Khepera robot 4, and this causes a sudden jump in the estimation result, as shown in Experiment 3 (the sudden jump at around 22 s) in Fig. 18. It can be observed from the three experiments that the estimated diffusivity converges fast using

the online identification algorithm with the steady-state errors 0.0108, 0.0208, and 0.0042, respectively. The algorithm is robust to the change of the initial positions of the robots, and also robust to sensor errors.

6. Conclusion

In this article, we introduce an online parameter identification algorithm using a multi-robot system in realistic scenarios when obstacles and hazard zones are present in the field. To verify the algorithms, we design and build a controllable CO₂ field in our laboratory and construct a static sensor network to measure and calibrate the field. Simulations are performed in the pre-collected field and experiments are done using a group of four collaborating Khepera IV robots equipped with sensors. Both simulation and experimental results show satisfactory results. Future work includes the extension of the algorithm to more complicated situations such as time-varying coefficients and the field change due to the interactions between the obstacles and the surrounding field.

Acknowledgments

This work was supported by the National Science Foundation (grant number CMMI-1663073, 1917300), (grant number CNS-1446461).

References

1. M. Dunbabin and L. Marques, "Robotics for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.* **19**(1), 24–39 (2012).
2. Z. Liu, "A Supervisory Approach for Hazardous Chemical Source Localization," *2013 IEEE International Conference on Mechatronics and Automation (ICMA)* (2013) pp. 4–7.
3. Y. Fukazawa and H. Ishida, "Estimating gas-source location in outdoor environment using mobile robot equipped with gas sensors and anemometer," *IEEE Sens.*, 1721–1724 (2009).
4. L. Z. Guo, S. A. Billings and H. L. Wei, "Estimation of spatial derivatives and identification of continuous spatio-temporal dynamical systems," *Internal J. Control* **79**(9), 1118–1135 (2006).
5. Y. Yashiro, K. Eguchi, S. Iwasaki, Y. Yamauchi and M. Nakata, "Development of obstacle avoidance control for robotic products using potential method," *Mitsubishi Heavy Ind. Tech. Rev.* **51**(1), 34–39 (2014).
6. G. Anderson, C. Sheesley, J. Tolson, E. Wilson and E. Tunstel, "A Mobile Robot System for Remote Measurements of Ammonia Vapor in the Atmosphere," *2006 IEEE International Conference on System, Man and Cybernetics (SMC)* (2006) pp. 241–246.
7. L. A. Rossi, B. Krishnamachari and C.-C. Jay Kuo, "Distributed Parameter Estimation for Monitoring Diffusion Phenomena Using Physical Models," *2004 IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)* (2004) pp. 460–469.
8. M. Krstic and A. Smyshlyaev, "Adaptive control of PDEs," *Ann. Rev. Control* **32**(2), 149–160 (2008).
9. D. Ucinski, *Optimal Measurement Methods for Distributed Parameter System Identification* (CRC Press, Boca Raton, London, New York, Washington, DC., 2004).
10. M. A. Demetriou and I. I. Hussein, "Estimation of spatially distributed processes using mobile spatially distributed sensor network," *SIAM J. Control Optim.* **48**(1), 266–291 (2009).
11. R. Abdoolee, B. Champagne and A. H. Sayed, "Estimation of space-time varying parameters using a diffusion LMS algorithm," *IEEE Trans. Sig. Process.* **62**(2), 403–418 (2014).
12. S. Li, Y. Guo and B. Bingham, "Multi-robot Cooperative Control for Monitoring and Tracking Dynamic Plumes," *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014) pp. 67–73.
13. V. M. H. Bennetts, A. J. Lilienthal, A. A. Khaliq, V. P. Sese and M. Trincavelli, "Towards Real-World Gas Distribution Mapping and Leak Localization Using a Mobile Robot with 3D and Remote Gas Sensing Capabilities," *IEEE International Conference on Robotics and Automation (ICRA)* (2013) pp. 2327–2332.
14. J. You, F. Zhang and W. Wu, "Cooperative Filtering for Parameter Identification of Diffusion Processes," *2016 IEEE Conference on Decision and Control (CDC)* (2016) pp. 4327–4333.
15. M. Li, K. Su, Y. Zhang, J. You and W. Wu, "Experimental Validation of Diffusion Coefficient Identification Using a Multi-Robot System," *2016 IEEE MIT Undergraduate Research Technology Conference (URTC)* (2016).
16. L. Marques, A. Martins and A. T. de Almeida, "Environmental Monitoring with Mobile Robots," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005) pp. 3624–3629.
17. H. Ishida, Y. Wada and H. Matsukura, "Chemical sensing in robotic applications: A review," *IEEE Sens. J.* **12**(11), 3153–3173 (2012).
18. M. Rossi and D. Brunelli, "Autonomous gas detection and mapping with unmanned aerial vehicles," *IEEE Trans. Inst. Meas.* **65**(4), 765–775 (2016).
19. A. Marjovi, J. Nunes, P. Sousa, R. Faria and L. Marques, "An Olfactory-Based Robot Swarm Navigation Method," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 65(4) (2010) pp. 4598–4963.
20. J. You and W. Wu, "Sensing-Motion Co-Planning for Reconstructing a Spatially Distributed Field Using a Mobile Sensor Network," *IEEE 56th Conference on Decision and Control (CDC)* (2017) pp. 3113–3118.

21. D. Ucinski and M. Patan, "Sensor network design for the estimation of spatially distributed processes," *Int. J. Appl. Math. Comput. Sci.* **20**(3), 459–481 (2010).
22. F. Zhang and N. E. Leonard, "Cooperative control and filtering for cooperative exploration," *IEEE Trans. Autom. Control* **55**(3), 128–136 (2010).
23. G. Cabrita, L. Marques and V. Gazi, "Virtual Cancellation Plume for Multiple Odor Source Localization," *2007 IEEE International Conference on Intelligent Robots and Systems (IROS)* (2013) pp. 5552–5558.
24. J. You, Y. Zhang, M. Li, K. Su, F. Zhang and W. Wu, "Cooperative Parameter Identification of Advection-diffusion Processes Using a Mobile Sensor Network," *2017 American Control Conference (ACC)* (2017) pp. 3230–3236.
25. E. Rimón and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.* **8**(5), 501–518 (1992).
26. R. Silva-Ortigoza, C. Marquez-Sanchez, F. Carrizosa-Corral, V. M. Hernandez-Guzman, J. R. Garcia-Sanchez, H. Taud, M. Marciano-Melchor and J. A. Alvarez-Cedillo, "Obstacle Avoidance Task for a Wheeled Mobile Robot: A Matlab-Simulink-Based Didactic Application," *In: MATLAB Applications for the Practical Engineer*, Chapter 2 (2014) pp.79–102.
27. M. A. Demetriou, N. A. Gatsonis and J. R. Court, "Coupled control-computational fluids approach for the estimation of the concentration from a moving gaseous source in a 2-D domain with a Lyapunov-guided sensing aerial vehicle," *IEEE Trans. Control Syst. Tech.* **22**(3), 853–867 (2014).
28. J. You and W. Wu, "Online passive identifier of spatially distributed systems using mobile sensor networks," *IEEE Trans. Control Syst. Tech.* **25**(6), 2151–2159 (2017).
29. J. You and W. Wu, "Geometric Reinforcement Learning Based Path Planning for Mobile Sensor Networks in Advection-Diffusion Field Reconstruction," *IEEE 57th Conference on Decision and Control (CDC)* (2018) pp. 1949–1954.