2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

# Multi-robot Cooperative Control for Monitoring and Tracking Dynamic Plumes

Shuai Li, Yi Guo and Brian Bingham

***Abstract*— We study robotic tracking of dynamic plume front modeled by the advection-diffusion equation in this paper. Different from existing work purely relying on gradient measurement, the transport model of pollution source is explicitly considered in tracking control design. We first study the problem using a single robot and solve the problem in an estimation and control framework. We then extend it to the multi-robot case in a nearest-neighbor communication structure, and have the robots take formation along the plume front. The distributed control is scalable to a large number of robots. Simulation results show satisfactory performances of the proposed method.**

## I. INTRODUCTION

The recent deep water horizon oil spill has posed great challenges to both robotics and ocean engineering communities. It took months to estimate the extent of the underwater plume, and the accuracy of these estimates will likely be debated for years to come. The challenges motivate us to consider utilizing advanced robotic techniques to monitor and track the propagation of oil plumes. In this paper, we propose a model-based method to track the dynamic plume front using multi-robot platforms.

Existing approaches related to plume tracking can be categorized into three classes: mapping based approach, behavior based approach, and control based approach. In the mapping based approach, a concentration map of the environment is first built, and then plume tracking is conducted based on the obtained map. Representative work includes [1], where a grid map was created to represent the concentration distribution based on data collected by a single robot. In [2] a hidden Markov method (HMM) was employed to map the environment. Similar strategies were applied in [3] to localize a chemical plume source. In [4] the likelihood grid mapping method was extended to map an environment with multiple chemical sources. As it is necessary to scan the environment for mapping, it introduces extra overhead of robot scanning trajectory design, which often makes the map based method time costly. On the contrary, the behavior based approach combines certain elementary behaviors to track an emergent plume. Li *et al*. in [5] developed bio-inspired chemical plume tracing on an autonomous underwater vehicle using elementary behaviors such as finding plume, tracking-in and tracking-out plume in a subsumption architecture. Social potential fields were used in [6] to coordinate group behaviors, where experiments demonstrated a robotic swarm was driven to find a spill and reach a perimeter formation. In [7], the Braitenberg model was employed to map the sensor reading of concentration to robot motor reaction in environments with odor plumes, and the effectiveness was validated by experimental results. Generally speaking, the behavior based approach is often intuitive, and it usually takes extensive simulations or experiments to validate the method ( [5]–[7]). Due to the lack of rigorousness such as convergency analysis, performances are difficult to guarantee thus a working example may be difficult to extend to other scenarios with different parameters.

The control based approach explicitly designs robot control laws with provable convergence. In [8], the plume tracking problem was modeled by a finite automaton and the hybrid control theory was employed in theoretical analysis. In [9], the robotic manifold tracking of coherent structures in flows was investigated. In [10], the authors used stochastic methods to model plume spikes and developed bio-inspired control laws for multi-robot plume tracking in turbulent flows. In [11], a level curve tracking problem was solved in two-dimensional space using multi-sensor platforms. A Kalman filter was developed to incorporate historical sensor reading into control laws that guide the robot movement. This approach was later extended to solve the same problem with control uncertainties in [12] and in three-dimensional space in [13]. However, due to the lack of a fixed concentration reference for the level curve tracking, this type of method is subject to concentration drifting in the presence of noises in sensor readings.

In this paper, we study dynamic plume tracking control using multiple robots. The approach falls in the category of control based method, and the plume tracking problem is solved in an estimation and control framework. We first discuss the single robot case, where the plume front dynamics is derived using the advection-diffusion equation governing the plume propagation. Then an observer is designed to estimate the dynamic movement of the plume front, and a feedback control law is constructed to track the plume front. We then extend the single robot case to a multi-robot scenario, where an additional behavior of formation along the plume front is added with robots' control laws explicitly given for a multi-robot team in a nearest-neighbor communication topology. The algorithms designed in both the single and multiple robot cases are tested in simulation, which show satisfactory performances.

The contribution of this paper are twofold. First, the proposed method utilizes the propagation model (*i.e.*, the

S. Li and Y. Guo are with Department of Electrical & Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA. lshuai@stevens.edu, yguo1@stevens.edu

B. Bingham is with Department of Mechanical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822, USA. bsb@hawaii.edu

978-1-4799-3685-4/14/$31.00 ©2014 IEEE

advection-diffusion model) of point-source pollution in marine environments, and incorporate the model into the controller design. Second, the designed control is analytically constructed with provable convergence, and numerical computation of partial differential equations is avoided for rapid response and real time control.

Comparing to existing work [11], [13], we consider *dynamic* plume where the plume front propagates along time, but the work in [11], [13] only considers *static* level curve tracking. Also, we exploit the transport of pollution source in water, and incorporate the advection-diffusion model into our control design, while the work in [11], [13] mainly replies on the gradient information (either implicitly or explicitly obtained) for plume tracking. The dynamic nature of our solution avoids the drifting problem that may occur during the static level curve tracking due to uncertainties or sensor noises. Also, we provide an analytic solution that does not need numerically solving the complex plume dynamics as done in [14] which is computational expensive.

## II. Problem Statement

In this section, we present the plume model, assumptions made on the robot and environments, and state the control problems studied in the paper.

### A. Plume Model

Chemicals introduced in the marine environment through point-source pollution generally propagate through two transport mechanisms: advection and diffusion. Advection is the transport of the chemical due to the motion of the flow, and diffusion is the motion from areas of higher concentration to areas of lower concentration. The propagation of chemicals is modeled by the following advection-diffusion equation:

$$\frac{\partial c(x,t)}{\partial t} + v^T(x,t)\nabla c(x,t) = k\nabla^2 c(x,t) \tag{1}$$

where $c \geq 0$, $c \in \mathbb{R}$ is the chemical concentration, $t$ denotes time, $v$ is the advection velocity, $k > 0$, $k \in \mathbb{R}$ is the diffusion coefficient in a homogenous space, $\nabla c(x,t) = \frac{\partial c(x,t)}{\partial x}$ is the spatial gradient of $c(x,t)$, $\nabla^2 c(x,t) = \frac{\partial^2 c(x,t)}{\partial x_1^2} + \frac{\partial^2 c(x,t)}{\partial x_2^2}$ is the divergence of $c(x,t)$ in two dimensional space, and $\frac{\partial c(x,t)}{\partial t}$ is the partial derivative of $c(x,t)$ along the time axis.

### B. Robot Model

In this paper, we consider plume front tracking using surface vehicles with the following kinematic model [15] in two-dimensional space,

$$\dot{x}_r = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \tau \tag{2}$$

where $x_r = [x_{r1}, x_{r2}]^T$ represents the Cartesian coordinates of the robot, $\tau = [\tau_1, \tau_2]^T$ are the surge and sway velocities respectively defined in the body-fixed frame, $\theta_r$ is the heading angle of the vehicle. Note that the determinant of $\begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix}$ is always 1 whatever the value of $\theta$, meaning that its inverse always exists. The inverse matrix can be analytically solved as $\begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix}$. Equation (2) can be converted into a single integrator model by defining a new control input metric $u = [u_1, u_2]^T$ satisfying,

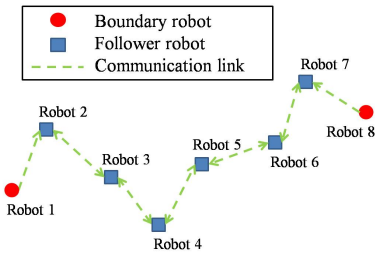


Fig. 1. The assignment of boundary robots and follower robots in an eight robot group, where the red disk and the blue square represent the boundary robots and the follower robots respectively, the green dash-line represents the directional communication links between neighboring robots.

$$\tau = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix} u \tag{3}$$

with which, the robot dynamics is reduced to a single integrator model,

$$\dot{x}_r = u. \tag{4}$$

### C. Available Information

For the single robot plume tracking problem, we assume that it has sensors to obtain local information, including its position and heading, and the chemical concentration information at its current position. For the multi-robot plume tracking case, we assume in addition that each robot is able to access information from its one-hop neighbors by communication. We made the following assumptions.

**Assumption 1:** The robot's onboard sensors obtain its position $x_r$, heading $\theta_r$, and the following information at its position: the chemical concentration $c_r$, the gradient of the concentration $\nabla c_r$, the divergence of the concentration gradient $\nabla^2 c_r$, and the flow velocity $v_r$.

*Remark 1:* Note that there may not exist sensors to directly measure the gradient and the divergence of the concentration. However, the values can be estimated either by counting the historical data of a single sensor [16], or by considering the readings from multiple spatially distributed sensors on a single robot [17], [18].

**Assumption 2:** For multiple robot plume tracking, two robots are assigned as boundary robots and all the other robots are assigned as follower robots. As shown in Fig. 1 as an example, the two boundary robots do not access information from other robots, and the follower robot, i.e., the $i$th robot ($1 < i < n$ with n denoting the total number of robots), only communicates with its one-hop neighbors on the communication graph, i.e., the $(i-1)$th robot and the $(i+1)$th robot.

*Remark 2:* Assumption 2 significantly reduces the communication complexity by restricting the robot communication along a line topology with a communication complexity $O(n)$, in contrast to an all-to-all communication with a communication complexity $O(n^2)$ [19], [20].

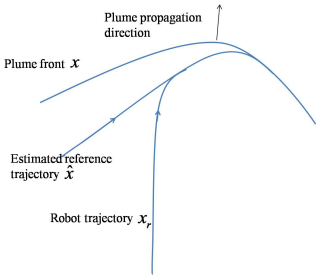


Fig. 2. The schematic of the proposed algorithm for single robot plume front tracking. There are two steps in the algorithm: 1) the robot estimates the plume front trajectory, 2) the robot tracks the estimated trajectory. The robot achieves front tracking if both the estimated and tracking trajectories converge.

### D. Control Objective

At time $t$, we consider those points $x(t)$ with the concentration $c(x(t),t) = c_0$ (where $c_0$ is the threshold concentration) to be the plume front. We denote the plume front using the position set $\{x(t) \in \mathbb{R}^2, c(x(t),t) = c_0\}$. The basic control objective is to drive the robot to the plume front. To monitor the whole perimeter of the plume front, we impose the second objective for the single robot to patrol along the plume front with a desired speed, while in the multi-robot case, we want the robots distributed evenly along the plume front. The control objectives are formally stated as follows.

**Problem 1 (Single Robot Plume Front Tracking):** For the plume dynamics modeled by the advection-diffusion equation (1), design a control law under Assumption 1 to drive a single robot, which is subject to the dynamic constraint (4), to track the plume front $\{x(t) \in \mathbb{R}^2, c(x(t),t) = c_0\}$, and patrol along the plume front with a desired speed $v_d(t) \in \mathbb{R}$.

**Problem 2 (Distributed Multi-robot Plume Front Tracking):** For the plume modeled by the advection-diffusion equation (1), design a control law under Assumptions 1 and 2 to drive a group of robots, which are subject to the dynamic constraint (4), to track the plume front $\{x(t) \in \mathbb{R}^2, c(x(t),t) = c_0\}$, and simultaneously reach an even distribution around the plume front.

## III. Single Robot Plume Front Tracking

In this section, we solve Problem 1 for the single robot plume tracking. The problem is modeled in an estimation and control perspective. A state equation and a measurement equation are constructed to design a nonlinear observer for the position estimation of a reference point on the plume front. Then, a control law is presented to steer the robot to the estimated position.

### A. Algorithm Overview

We partition the plume front tracking task into two parts: the first part is the plume front estimation part, which estimates the position of a reference point on the front; the second part is the tracking control part, which drives the robot to the estimated position. The estimation part and the tracking control part run simultaneously to reach the design objective. As illustrated in Fig. 2, the estimation part maintains a variable $\hat{x}(t)$, which estimates the position of the reference point $x(t)$ on the plume front. The estimation part is designed to enable the convergence of $\hat{x}(t)$ to $x(t)$, i.e., the convergence of the estimation error $e(t) = x(t) - \hat{x}(t)$ to zero as time elapses. The tracking control part steers the robot positioned at $x_r(t)$ to the estimated place $\hat{x}(t)$, to reduce the control error $e'(t) = \hat{x}(t) - x_r(t)$ along time. When both the estimation error $e(t)$ and the control error $e'(t)$ converge to zero, $x(t) - x_r(t) = e(t) + e'(t)$ will converge to zero and the design objective of driving the robot to the reference point is reached.

### B. State Equation of the Plume Front Dynamics

In this subsection, we first derive the plume front dynamics based on the partial differential equation (1), and then use the plume front dynamics to construct the state equation of the reference point, which describes the expected behavior of the robot in Assumption 1.

*1) Plume Front Dynamics:* In this part, we derive the plume front dynamics. Because the reference point describes the expected behavior of the robot, it locates on the plume front according to Assumption 1. Therefore, at time $t$, the reference point $x(t)$ satisfies,

$$c(x(t),t) = c_0 \tag{5}$$

Without introducing confusion, we drop the variable $t$ in $x(t)$ hereafter. Computing time derivative on both sides of Eq. (5) yields,

$$\dot{c}(x,t) = \frac{\partial c}{\partial t} + \nabla^T c \dot{x} = 0 \tag{6}$$

where $\frac{\partial c}{\partial t} = \frac{\partial c(x,t)}{\partial t}$ is the time derivative of the concentration, $\nabla c = \nabla c(x,t) = \frac{\partial c}{\partial x}$ is the spatial derivative, i.e., the gradient of the concentration. With Eq. (1) and Eq. (6), we get

$$\dot{x}^T \nabla c = -v_x^T \nabla c - k \nabla^2 c \tag{7}$$

where $v_x = v(x,t)$ and $\nabla^2 c = \nabla^2 c(x,t)$.

*2) Derivation of the State Equation based on Plume Front Dynamics:* Because the robot is also expected to patrol with speed $v_d = v_d(t)$ along the plume front, the reference point, which describes the ultimate behavior of the robot, is then subject to the following constraint,

$$\frac{(A\nabla c)^T}{\|\nabla c\|} \dot{x} = v_d \tag{8}$$

where $A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is the orthogonal matrix and $A\nabla c$ represents a vector along the tangent direction of the plume front. Regarding Eq. (7) and Eq. (8) as linear equations relative to $\dot{x}$, $\dot{x}$ can be uniquely solved as follows,

$$\dot{x} = -\frac{(v_x^T \nabla c + k \nabla^2 c)\nabla c}{\|\nabla c\|^2} + \frac{v_d A \nabla c}{\|\nabla c\|} \tag{9}$$

### C. Measurement Equation

For the robot positioned at $x_r$, the concentration field can be approximated by the first-order Taylor expansion as,

$$c(x,t) = \nabla^T c_r (x - x_r) + c_r \tag{10}$$

where $x_r$ denotes the position of the robot at time $t$, $c_r = c(x_r, t)$ is the concentration at position $x_r$, and $\nabla c_r = \frac{\partial c(x,t)}{\partial x}|_{x=x_r}$ is the gradient at $x_r$. Define $y(x) = c(x,t)$ as the output of the measurement at time $t$. Then, the above approximation yields,

$$y(x) = \nabla^T c_r (x - x_r) + c_r \tag{11}$$

Note that the true value of $y(x)$ is $y(x) = c_0$ since the reference point $x$ locates on the front. In addition, the measurement equals $y(\hat{x}) = c(\hat{x}) = \nabla^T c_r(\hat{x} - x_r)$ at the point $\hat{x}$.

### D. Observer Design

Until now, we have obtained the state equation (9) and the measurement equation (11), which re-write as follows,

$$\dot{x} = -\frac{(v_x^T \nabla c + k\nabla^2 c)\nabla c}{\|\nabla c\|^2} + \frac{v_d A \nabla c}{\|\nabla c\|} \tag{12}$$

$$y(x) = \nabla^T c_r (x - x_r) + c_r \tag{13}$$

For the system described by (12) and (13), we use the following extended Luenberger observer [21] for the state estimation,

$$\dot{\hat{x}} = -\frac{(v_{\hat{x}}^T \nabla c_{\hat{x}} + k\nabla^2 c_{\hat{x}})\nabla c_{\hat{x}}}{\|\nabla c_{\hat{x}}\|^2} + \frac{v_d A \nabla c_{\hat{x}}}{\|\nabla c_{\hat{x}}\|} - k_3 \nabla c_r \big(\nabla^T c_r(\hat{x} - x_r) + c_r - c_0\big) \tag{14}$$

where $k_3 > 0$ is a coefficient, $v_{\hat{x}} = v_{\hat{x}}(\hat{x}, t)$, $\nabla c_{\hat{x}} = \nabla c(\hat{x})$, $\nabla^2 c_{\hat{x}} = \nabla^2 c(\hat{x})$.

Remarkably $v_{\hat{x}}$, $\nabla c_{\hat{x}}$ and $\nabla^2 c_{\hat{x}}$ are all quantities at position $\hat{x}$. For the robot at position $x_r$, they are neither measurable nor computable since the above quantities, as functions of position $\hat{x}$ and time $t$, is unknown. For remedy, we simply replace them with the associated quantities at $x_r$, which are accessible by the robot according to Assumption 1. In this way, we have the following observer expression,

$$\dot{\hat{x}} = -\frac{(v_r^T \nabla c_r + k\nabla^2 c_r)\nabla c_r}{\|\nabla c_r\|^2} + \frac{v_d A \nabla c_r}{\|\nabla c_r\|} - k_3 \nabla c_r \big(\nabla^T c_r(\hat{x} - x_r) + c_r - c_0\big) \tag{15}$$

where $v_r = v(x_r, t)$, $\nabla c_r = \nabla c(x_r, t)$ and $\nabla^2 c_r = \nabla^2 c(x_r, t)$.

*Remark 3:* The absolute value of $\big(\nabla^T c_r(\hat{x} - x_r) + c_r - c_0\big)\nabla c_r$ in Eq. (15) goes small very fast when $\nabla^T c_r$ is small. To gain a fast enough response towards the desired reference point, it is necessary to tune $k_3$ large in this case. In contrast, when $\nabla^T c_r$ is large, the absolute value of this term becomes large very fast and thus needs a small $k_3$ as the gain. To adaptively adjust the gain $k_3$, we can set $k_3 = k_2 / \|\big(\nabla^T c_r(\hat{x} - x_r) + c_r - c_0\big)\nabla c_r\|$ in implementation with $k_2 > 0$ as a constant to improve the performance.

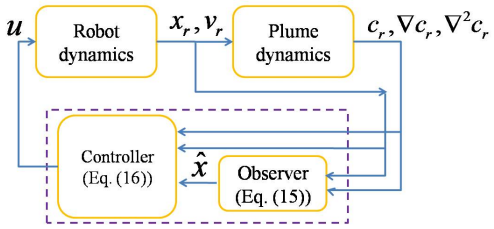


Fig. 3. Control block diagram for the single robot plume front tracking.

### E. Estimation-based Tracking Control

For the robot model (4), we present the following control law to reduce the control error $e' = \hat{x} - x_r$,

$$u = -\frac{(v_r^T \nabla c_r + k\nabla^2 c_r)\nabla c_r}{\|\nabla c_r\|^2} + \frac{v_d A \nabla c_r}{\|\nabla c_r\|} - k_3 \nabla c_r \cdot \big(\nabla^T c_r(\hat{x} - x_r) + c_r - c_0\big) - k_6(x_r - \hat{x}) \tag{16}$$

where $k_6 > 0$ is a constant. As illustrated in Fig. 3, the proposed control law consists of two parts (as shown in the dashed rectangle in the figure): the observer part and the controller part. The observer collects inputs from the robot (i.e., $x_r$ and $v_r$), and inputs from the environment (i.e., $c_r$, $\nabla c_r$, and $\nabla^2 c_r$) and outputs $\hat{x}$, which is the estimation on the plume front trajectory. Together with the sensory information $x_r$, $v_r$, $c_r$, $\nabla c_r$, and $\nabla^2 c_r$, the observer output $\hat{x}$ is feeded into the controller, which yields the control input $u$ to the robot. The diagram illustrates the interplay between the robot, the environment and the control law.

*Remark 4:* In the control law (16), the term $-k_6(x_r - \hat{x})$ forms a negative feedback on the control error. The rest terms is identical to $\dot{\hat{x}}$ according to the observer (14) and it is a velocity compensation term to regulate the tracking error.

It can be shown that the proposed control strategy is locally stable. The robot dynamic (4) becomes $\dot{x}_r = -k_6(x_r - \hat{x}) + \dot{\hat{x}}$ by substituting the control law (16) and the observer (14) inside). The control error $e'(t) = \hat{x}(t) - x_r(t)$ then turns to $\dot{e}' = -k_6 e'$, which converges to zero, implying that $\hat{x} \to x_r$ when $t \to \infty$. When time goes to infinity, the observer (15) reduces to equation (14). The convergence of the observer guarantees the convergence of the whole estimation based control system.

*Remark 5:* In practice, the mismatch between the real and the nominal models, the measurement error, etc., may result in additive noises. Although the above results are derived without considering noises, it is expected that the plume front tracking error is bounded for bounded additive noises, i.e., the robot tracks the plume front approximately and the bounded tracking error depends on the amplitude of the noises. We will provide formal analysis in our future work.

## IV. Multi-robot Plume Front Estimation and Tracking

In this section, we propose a solution for the distributed multi-robot plume tracking (i.e., Problem 2) based on the results on the single robot plume tracking. A group of robots

is deployed and distributed evenly on the plume front. Like before, we use the two-step procedure to complete the task: the estimation step and the tracking control step. In the estimation step, we define $n$ (the number of robots in the group) moving reference points on the plume front with the desired spatial distribution, and define an observer to estimate the position of reference points recursively. In the tracking control step, we drive the robots to the estimated reference point.

For the task considered in this section, as sketched in Fig. 1, we assign two robots as boundary robots and all the other robots as follower robots. We present the control laws for both of them as given next.

*1) Boundary Robots:* For the boundary robots, i.e., robot $i = 1$ and robot $i = n$, we take the same control law as (16) based on a similar observer as (15), which writes,

$$\begin{aligned} u_i &= -\frac{(v_{ri}^T\nabla c_{ri} + k\nabla^2 c_{ri})\nabla c_{ri}}{\|\nabla c_{ri}\|^2} + \frac{k_{4i}A\nabla c_{ri}}{\|\nabla c_{ri}\|} - k_3\nabla c_{ri} \\ &\quad \cdot\big(\nabla^T c_{ri}(\hat{x}_i - x_{ri}) + c_{ri} - c_0\big) - k_6(x_{ri} - \hat{x}_i) \quad (17a) \\ \dot{\hat{x}}_i &= -\frac{(v_{ri}^T\nabla c_{ri} + k\nabla^2 c_{ri})\nabla c_{ri}}{\|\nabla c_{ri}\|^2} + \frac{k_{4i}A\nabla c_{ri}}{\|\nabla c_{ri}\|} - k_3\nabla c_{ri} \\ &\quad \cdot\big(\nabla^T c_{ri}(\hat{x}_i - x_{ri}) + c_{ri} - c_0\big) \quad (17b) \\ &\quad \text{for } i = 1, n \end{aligned}$$

where $\hat{x}_i$, $x_{ri}$ are the estimated position of the $i$th reference point and the position of the $i$th robot, respectively, $v_{ri} = v(x_{ri})$, $c_{ri} = c(x_{ri})$, $\nabla c_{ri} = \nabla c(x_{ri})$ and $\nabla^2 c_{ri} = \nabla^2 c(x_{ri})$ are the flow velocity at the $i$th robot position $x_{ri}$, the concentration at $x_{ri}$, the gradient of the concentration at $x_{ri}$, and the divergence of the concentration gradient at $x_{ri}$, respectively, $k_{4i} = k_{4i}(t)$ represents the desired speed along the tangent direction of the plume front.

The boundary robot control law (17) has a design parameter $k_{4i}$ for $i = 1$, and $i = n$, which assigns a desired patrolling speed to the robot. By choosing $k_{41} > 0$ and $k_{4n} < 0$, the boundary robots 1 and $n$ patrol in the counter-clockwise and clockwise directions, respectively. This generates a patrolling behavior for each of the boundary robots but with an opposite direction. Combining with the follower robots whose controllers are presented below, the group of robots spread out to cover the whole plume front. As the two boundary robots moves in an opposite direction along the plume front, they will meet on sometime and then the parameter $k_{4i}$ is switched to the following one allowing them to stop patrolling and stay together:

$$k_{4i} = k_7\|\hat{x}_i - \hat{x}_{n+1-i}\|\text{sign}\big((\hat{x}_i - \hat{x}_{n+1-i})^T A\nabla c(\hat{x}_i)\big) \quad (18)$$

where $k_7 > 0$ is a positive constant, $i = 1, n$.

*2) Follower Robots:* We derive the follower control law based on the results for single robot plume front tracking. The parameter $k_{4i} = k_{4i}(t)$ in (15), which represents the desired speed along the tangent direction, gives us an additional design freedom. This freedom is exploited in this part to design a distributed formation behavior. To this end, the gain $v_d$ in (15) is replaced with an adaptive one relying on the distance difference between the two neighbors to form a cooperative observer for the reference points. Thus, we propose the following observer based control law for the $i$th ($i = 2, 3, ..., n-1$) robot,

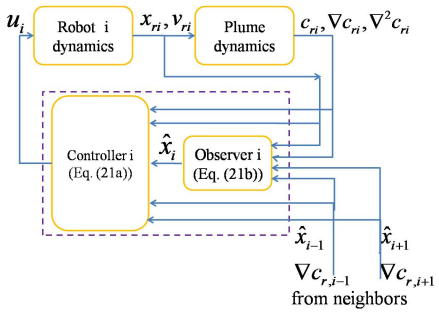


Fig. 4. Control block diagram for a follower robot in the multi-robot plume front tracking.

$$\begin{aligned} u_i &= -\frac{(v_{ri}^T\nabla c_{ri} + k\nabla^2 c_{ri})\nabla c_{ri}}{\|\nabla c_{ri}\|^2} + \frac{k_{4i}A\nabla c_{ri}}{\|A\nabla c_{ri}\|} - k_3\nabla c_{ri} \\ &\quad \cdot\big(\nabla^T c_{ri}(\hat{x}_i - x_{ri}) + c_{ri} - c_0\big) - k_6(x_{ri} - \hat{x}_i) \quad (19a) \\ \dot{\hat{x}}_i &= -\frac{(v_{ri}^T\nabla c_{ri} + k\nabla^2 c_{ri})\nabla c_{ri}}{\|\nabla c_{ri}\|^2} + \frac{k_{4i}A\nabla c_{ri}}{\|A\nabla c_{ri}\|} \\ &\quad -k_3\big(\nabla^T c_{ri}(\hat{x}_i - x_{ri}) + c_{ri} - c_0\big)\nabla c_{ri} \quad (19b) \end{aligned}$$

with,

$$\begin{aligned} k_{4i} &= \frac{k_5(\hat{x}_{i-1}^T A\nabla c_{r,i-1} + \hat{x}_{i+1}^T A\nabla c_{r,i+1} - 2\hat{x}_i^T A\nabla c_{ri})}{\|A\nabla c_{ri}\|} \\ \text{for} &\quad i = 2, 3, ..., n-1 \quad (19c) \end{aligned}$$

where $k_5 > 0$ is a constant.

The block diagram of the multi-robot cooperative plume front tracking for a follower robot is illustrated in Fig. 4. In addition to the information from the environment and the $i$th robot itself, this control scheme also requires information of $\hat{x}_{i-1}$ and $\hat{x}_{i+1}$, which are the observer outputs from the $(i-1)$th and the $i$th robot, respectively.

The convergence of the boundary robots to the plume front follows the convergence results of the designed observer in the single robot case. The proof idea includes three parts. First, we construct a set of reference points with formation interactions on the plume front and each of them is associated with a robot in the group. Second, we prove the convergence of the robots to the reference points. Third, we prove the reference points converge to even distribution along the plume front between the reference points associated with the two boundary robots. We omit the rigorous proof of convergence due to space limitation.

## V. Simulations

In this section, we present Matlab numerical simulations to validate the effectiveness of the proposed algorithms.

### A. Simulation Environment

The simulation of the environment includes two part: one is the simulator of a flow field and the other is the simulator of the concentration field. The flow field is generated numerically by solving incompressible Navier-Stokes

equations in a rectangular domain with prescribed velocities along the boundary using the program presented in [22]. The concentration field is generated using finite difference method to numerically solve the partial differential equation (1) with a fixed value of the concentration at sources as boundary conditions. The flow field is visualized in Fig. 5 (a), where the strength and direction of the flow at various positions are shown. Both of the two chemical sources locating at $(13, 3)$ and $(7, 7)$ has a fixed concentration 3. The propagation of the chemical in the flow field results in a time-varying concentration field.

### *B. Single Robot Plume Front Tracking*

Here we validate Algorithm 1 that uses a single robot to tack and patrol the plume front with concentration value $c_0 = 0.3$. The diffusion coefficient $k$ is set as $k = 0.5$. Due to the advection effect, the chemicals are blown along the flow direction and form a non-symmetric contour in the plume front. In the simulation, the patrolling speed is set as $k_4 = 6$, the gradient gain is set as $k_3 = 5$, and the estimated reference tracking gain is set as $k_6 = 5$. The robot is deployed to the field at time $t = 2s$ after the sources starts propagation at time $t = 0s$. As shown in Fig. 5 (b), due to the chemical propagation, the plume front contour expands with time. With the proposed method, the robot adaptively adjusts its orbits to track the plume front. We denote the difference between the plume front concentration $c_0$ and the measured concentration at the robot position $c_r$ as the plume front tracking error. As shown in Fig. 5 (c), the error attenuates to zero along time. Note that the lug in the error curve in Fig. 5 (c) around time $t = 8.5s$ happens at the time when the robot travels across the lower right tip of the plume front curve, where the concentration changes sharply since the area is mostly close to the source and locates in a place with the flow direction adverse to the diffusion direction. Any small variation in the robot trajectory in this area results in a large error in the concentration values.

### *C. Multi-robot Distributed Plume Front Tracking*

We validate Algorithm 2 with 30 robots to cover the dynamic plume front with concentration $c_0 = 0.1$. The diffusion coefficient $k$ in (1) is set as $k = 0.5$. For the boundary robots, the patrolling speed is set as $k_{4i} = 3$ for $i = 1$ and $k_{4i} = -3$ for $i = 30$. For follower robots, the gain $k_5$ is set to be $k_5 = 5$. For all robots, the gradient gain is set as $k_3 = 5$ and the estimated reference tracking gain is set as $k_6 = 5$. After the sources start propagation for 2 seconds, 30 robots are deployed as shown in Fig. 6, where the follower robots are marked by a hollow square in blue and the boundary robots are marked by a hollow circle in red. We want the robots to distribute on the plume front evenly, move with the plume propagation and expand the formation when the plume front expands. For this purpose, starting from time $t = 6s$, when the condition $\|\hat{x}_1 - \hat{x}_{30}\| \leq l_{dis}$ with $l_{dis} = 2$ is satisfied, $k_{4i}$ switches to the control law (17) with the parameter $k_{4i}$ set as (18) for the boundary robots $i = 1$ and $i = 30$, where the gain $k_7$ is set to be 20 in the simulation.

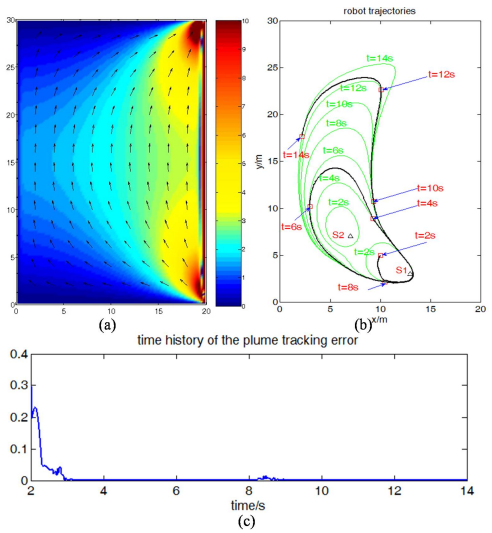


Fig. 5. (a). The velocity field considered in the simulation, where the pseudo-color and the yellow arrow indicate the strength and the direction of the flow, respectively. (b). The robot trajectory for single robot plume front tracking and patrolling, where $S_1$ and $S_2$ are the two sources, the green curves and the red squares represent the plume front contour and the robot positions, respectively; The black curve represents the robot trajectory. (c). The plume front tracking error.

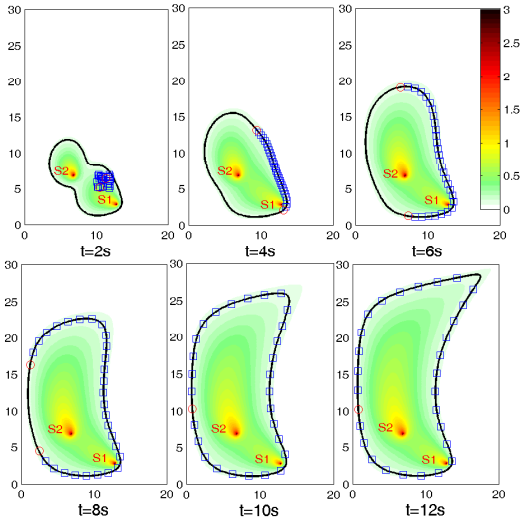


Fig. 6. The snapshots of the robot movements for a typical simulation run with 30 robots for cooperative plume front tracking, where the pseudo-color indicates the concentration distribution at each time step with the scale shown in the color-bar aside, the black curve is the contour of the plume front with a concentration value $c_0 = 0.01$, the hollow square in blue and the hollow circle in red represent the position of the follower robots and the true position of the boundary robots, respectively, the positions marked with S1 and S2 are the two sources starting propagating chemicals in the flow field shown in Fig. 5 at time $t = 0$.

Fig. 6 shows the snapshots of the simulation results along time. After the robots are deployed in the field, it rapidly forms the desired even distribution and reach the plume front as shown at time $t = 4s$. As time elapses, the robots move around the plume front shown as the black contour, and simultaneously follow the expansion movement of the plume front as shown in the snapshot $t = 6s$. After this time, the

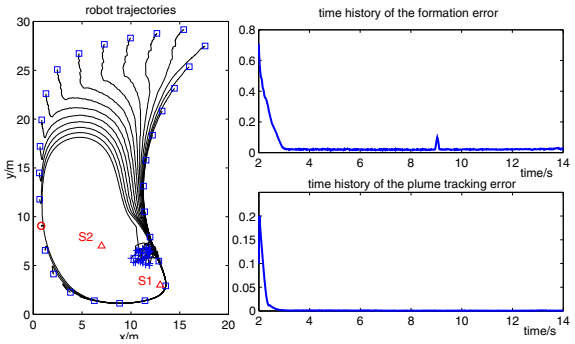


Fig. 7. The robot trajectories, the time profile of the formation error, and the plume front tracking error, where $S_1$ and $S_2$ represent the sources, '+' represents the initial position of robots, the hollow squares and the hollow circles represent the final positions of the follower robots and the boundary robots, respectively.

boundary robot keep moving in an opposite direction to cover the plume front until the distance between them is within the value $l_{dis}$, which is satisfied at time $t = 10s$. Finally, as shown at $t = 12s$, the two boundary robots reach consensus in their positions and the follower robots are successfully deployed evenly covering the plume front.

To evaluate the performance, we define two indices, one is the formation error and the other is the plume tracking error. The formation error measures whether the formation is achieved for the follower robots, $i = 2, 3, ..., n-1$. As the distance between any two neighboring robots is $\big|\|x_{ri} - x_{r,i+1}\| - \|x_{ri} - x_{r,i-1}\|\big|$, we define the average distance of the group of $(n-2)$ robots as the group formation error: $\sqrt{\frac{\sum_{i=2}^{n-1}(|\|x_{ri}-x_{r,i+1}\|-\|x_{ri}-x_{r,i-1}\||)^2}{n-2}}$. We define the plume tracking error of the group to be $\sqrt{\frac{\sum_{i=1}^{n}(c(x_{ri})-c_0)^2}{n}}$, which is the average tracking error of each robot in the group. The time histories of the formation error and the plume tracking error are plotted in Fig. 7, where the trajectories of the robots are also plotted. It can be seen that both the formation error and the tracking error reduce rapidly to a small value. It also shows that a small peak appears at around $t = 10s$ in the formation error subfigure, which is caused by the switching of the boundary robots' controllers from spread-out to stay-close and stop-rotating. It can be observed that the performance is overall satisfactory.

## VI. Conclusion

In this paper, the dynamic plume tracking problem is solved using a group of cooperating robots. The advection-diffusion equation is used to model the pollution plume propagation, and a model-based observer is built to estimate the plume front trajectory. Solutions are then provided for single robot tracking and multi-robot tracking with simultaneous formation based on an estimation and control framework. The multi-robot plume tracking is distributed and scalable with a nearest-neighboring communication topology. Simulations show the effectiveness of the proposed strategy. We will conduct experimental validation and characterize noise effects in the future work.

## Acknowledgement

The work was partially supported by the National Science Foundation under Grants IIS-12181558 (Li and Guo) and IIS-1217659 (Bingham).

## References

[1] A. Lilienthal and T. Duckett. Building concentration gridmaps with a mobile robot. *Robotics and Autonomous Systems*, 48:3–16, 2004.
[2] J.A. Farrell, S. Pang, and W. Li. Plume mapping via hidden markov methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 33(6):850–863, 2003.
[3] S. Pang and J.A. Farrell. Chemical plume source localization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(5):1068–1080, 2006.
[4] G. Ferri, M.V. Jakuba, and A. Mondini. Mapping multiple gas/odor sources in an uncontrolled indoor environment using a Bayesian occupancy grid mapping based method. *Robotics and Autonomous Systems*, 59(11):988 – 1000, 2011.
[5] W. Li, J.A. Farrell, S. Pang, and R.M. Arrieta. Moth-inspired chemical plume tracing on an autonomous underwater vehicle. *IEEE Transactions on Robotics*, 22(2):292–307, 2006.
[6] M.D. Mckay D.J. Bruemmer, D.D. Dudenhoeffer and M.O. Anderson. A robotic swarm for spill finding and perimeter formation. In *IEEE Spectrum*, pages 23–30, 2002.
[7] A. Lilienthal and T. Duckett. Experimental analysis of smelling braitenberg vehicles. *Environment*, 5:375–380, 2003.
[8] J. Clark and R. Fierro. Cooperative hybrid control of robotic sensors for perimeter detection and tracking. In *Proceedings of the American Control Conference*, pages 3500–3505, 2005.
[9] M.A. Hsieh, E. Forgoston, and T.W. Mather. Robotic manifold tracking of coherent structures in flows. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4242–4247, 2012.
[10] D. Chang, W. Wu, D.R. Webster, M.J. Weissburg, and F. Zhang. A bio-inspired plume tracking algorithm for mobile sensing swarms in turbulent flow. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 921–926, 2013.
[11] F. Zhang, E. Fiorelli, and N.E. Leonard. Exploring scalar fields using multiple sensor platforms: tracking level curves. In *Proceedings of IEEE Conference on Decision and Control*, pages 3579–3584, 2007.
[12] M. Malisoff and F. Zhang. Adaptive control for planar curve tracking under controller uncertainty. *Automatica*, 49(5):1411 – 1418, 2013.
[13] W. Wu and F. Zhang. Cooperative exploration of level surfaces of three dimensional scalar fields. *Automatica*, 47(9):2044–2051, 2011.
[14] R.N. Smith, Y. Chao, P.P. Li, D.A. Caron, B.H. Jones, and G.S. Sukhatme. Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model. *International Journal of Robotic Research*, 29:1475–1497, 2010.
[15] C.R. Sonnenburg and C.A. Woolsey. Modeling, identification, and control of an unmanned surface vehicle. *Journal of Field Robotics*, 30(3):371–398, 2013.
[16] E. Biyik and M. Arcak. Gradient climbing in formation via extremum seeking and passivity-based coordination rules. In *Proceedings of 46th IEEE Conference on Decision and Control*, pages 3133–3138, 2007.
[17] T.C. Henderson and E. Grant. Gradient calculation in sensor networks. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1792–1795, 2004.
[18] S. Li and Y. Guo. Distributed source seeking by cooperative robots: all-to-all and limited communications. In *Proceedings of International Conference on Robotics and Automation*, pages 1107–1112, 2012.
[19] R.O. Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *In Proceedings of IEEE*, pages 215–233, 2007.
[20] E. Klavins. Communication complexity of multi-robot systems. In J.D. Boissonnat, J.W. Burdick, K. Goldberg, and S. Hutchinson, editors, *Algorithmic Foundations of Robotics*, Berlin Heidelberg, 2003.
[21] M. Zeitz. The extended luenberger observer for nonlinear systems. *Systems & Control Letters*, 9(2):149 – 156, 1987.
[22] B. Seibold. A compact and fast matlab code solving the incompressible navier-stokes equations on rectangular domains. *Massachusetts Institute of Technology*, 2008.