Cooperative Level Curve Tracking in Advection-Diffusion Fields

Sachin Guruswamy and Wencen Wu
Computer Engineering Department
San Jose State University
San Jose, California, USA 95192
sachin.guruswamy@sjsu.edu, wencen.wu@sjsu.edu

***Abstract*—Level curve tracking in a spatial-temporal varying field is an important problem in many applications such as tracking and monitoring the propagation of a wildfire. In this paper, we investigate the level curve tracking problem using a collaborating group of sensing agents in an advection-diffusion field with obstacles. We implement a cooperative Kalman filter that takes into consideration the spatial-temporal varying property of the field to estimate the field value and gradient at two consecutive time steps, which are used in the motion control law that applies to the center of the agent formation so that the center of the formation is able to detect and track a desired level curve value in the advection-diffusion field. Obstacle avoidance is considered in the motion control law design. Simulations show satisfactory results of the proposed strategy.**

I. Introduction

A level curve in a two dimensional field corresponds to the set of all points in the field that have the same given field value. Level curve tracking in a scalar field is an important problem in many environmental monitoring applications such as keeping track of the progression of the wildfire front, the spread of volcanic plumes, and the propagation of chemical container spreading in a water body [1]. Most of the scenarios are mostly unprecedented and people cannot deploy static sensors networks, e.g., we cannot place sensors near all water bodies or forests and static sensor networks [2], [3] are often impractical for such tasks due to the large area they need to cover and high cost of installing in such areas [4], [5]. To overcome these hurdles, using mobile sensor agents is a great option with more options like patrolling the area, climbing gradient of the scalar fields, monitoring boundaries etc. However, using a single mobile sensor acquires lot of noise in real-world so instead many existing algorithms use multiple sensing agents to carry out the level curve tracking task or boundary tracking task [6]–[8].

In practice, most of the environmental processes are not static. Instead, they are spatial-temporal varying, meaning the state changes w.r.t. both time and space [9]. Advection-diffusion phenomena is one typical spatial-temporal varying processes, which can be found in many scenarios [1]. In face of such processes, level curve tracking becomes challenging since level curves are not static in the field any more. While most of the existing work deal with level curve tracking in a static field [6]–[8], few of them consider the situation where the field is spatial-temporal varying.

In this paper, we investigate the problem of cooperative level curve tracking in an advection-diffusion field with obstacles in the field. Following the similar approach used in [6], which carries out level curve tracking in a static scalar field, we develop a modified cooperative Kalman filter that takes into consideration the spatial-temporal varying property of the field and is able to provide additional estimated states of the field, i.e., the estimated states in two consecutive time steps. This enables us to apply the same steering motion control law as in [6] to our problem and achieve the cooperative level curve tracking. In addition, we consider the case that there are obstacles in the field, so the control law design must include the obstacle avoidance capability. To tackle this problem, we use the potential field approach for obstacle avoidance and add an additional term to the motion control law. Simulation study is conducted for the proposed strategy and the simulation results are satisfactory.

The rest of the paper is organized as follows. Section II formulates the problem of cooperative level curve tracking in an advection-diffusion field using a group of collaborating sensing agents. Section III introduces the formation control of the multi-sensor group and discusses the level curve tracking control law. Section IV presents the cooperative Kalman filter that provides the estimates of the field values and gradients. Simulation results are discussed in Section V including the step-by-step algorithm, and conclusions and future work are introduced in Section VI.

II. Problem Formulation

A. Advection Diffusion Processes

Most of the real world scenarios that consists of transport of substance by bulk motion in the nature like river flow, spreading of smoke, or wildfire can be modeled as spatial-temporal varying processes. These different types of processes can usually be modelled using partial differential equations (PDEs), which govern the motion of the particles/substances in the field. In this paper, we consider the two-dimensional (2D) advection-diffusion process, which is expressed using the

The research work is supported by NSF grant CMMI-1917300.

978-1-7281-9888-0/20/$31.00 ©2020 IEEE

equation below in (1) defined within the domain $\Omega \in \mathbb{R}^2$ with certain initial and boundary conditions.

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

where $z(r,t)$ is the concentration of the field at position r and time t, $\nabla z(r,t)$ represents the spatial gradient of $z(r,t)$, θ is the diffusion coefficient, and v is the flow velocity.

A level curve is defined as a set of all points in the field, at which the field takes on a given value. In a static scalar field, a level curve is described as $L_c(r) = \{r | z(r) = C\}$, where C is a given value. In the advection-diffusion field, we take the similar definition with the additional dimension of time, thus, the level curve can be written as $L_c(r,t) = \{r | z(r,t) = C\}$.

B. Sensor Dynamics

In most of the real world situations, there are uncertainties in the measurements of the field concentration due to noise and the sensing devices being imperfect. In this paper, we employ multiple sensing agents in a mobile sensor network to obtain the necessary estimates cooperatively and reduce the noise. Each of the agent is equipped of a sensor which is able to capture point measurements of the field at its current position $z(r,t)$. In most applications, we cannot obtain the continuous sensor measurements, so we obtain the measurements discretely over time. Let t_k be the time step at k^{th} interval and r_i^k be the agent i's position at k^{th} time step. The field concentration value at r_i^k will be denoted as $z(r_i^k, k)$. Let $r_c^k = [r_{c,x}^k, r_{c,y}^k]^T$ be the center of the formation of N agents at time t_k, i.e, $r_c^k = \frac{1}{N}\sum_{i=1}^N r_i^k$. The measurement captured by each agent can be modeled as

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

where w_i is assumed to be i.i.d Gaussian noise. We also make the following assumption for the sensing agents.

Assumption I: Each agent is capable of knowing it's location r_i^k and the field concentration measurement value $p(r_i^k, k)$ and is also capable of sharing these information with the other agents that are part of this formation.

In this paper, we aim to develop and verify a level curve tracking strategy for the group of sensing agents so that they can detect and track a level curve with a given level value in an advection-diffusion field. To-do so, a cooperative Kalman filter is needed to provide estimates of the field states, and formation and motion control laws need to be designed.

III. Formation and Motion Control

In this section, we introduce the control laws that enable the sensor network to detect and track a level curve in the advection-diffusion field while remaining in a desired formation.

A. Formation Shape and Orientation Control

We first explain how the formation is kept intact as we move the center of the formation. To achieve this, we use a transformation called Jacobi transformation Ψ [6]. Using

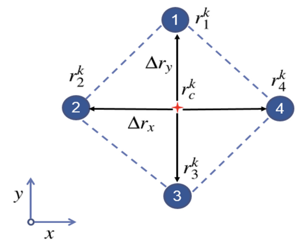


Fig. 1: The formation of four agents.

this transformation, we can decouple the formation shape and orientation dynamics from the center of the formation without taking the motion of the formation center into the account. Hence, the formation shape and orientation can be modified without worrying about the motion of the center of the formation. We can control the formation to have fixed orientation or we can have it rotate if we are using just two sensing agents. In this paper, we are considering that we have sensor agents $N > 3$ and our experiments are carried out using four sensor agents with a fixed orientation as shown in Figure 1. Details of the formation control can be found in [6], [8].

B. Formation Motion Control

Motion control is applied only to the center of the formation of the N agents since the formation can be considered as a rigid body after the formation control converges. Hence, we consider the center of the formation as a Newtonian particle and derive the equations governing this particle moving in the advection-diffusion field. This control algorithm is an extension of the one used in [6] for the static scalar field. This algorithm consists of how to find the curvature using Hessian matrix, how to calculate the motion control law derived from Lyapunov approach, and how to update the position of the center of the formation. Due to the page limit, we skip the complete derivation, instead, we use the control law derived in [6] and extend it to include an extra parameter to avoid obstacles along the path. Define the tangent vector to the level curve that passes the formation center as $\mathbf{x}_1$ and the perpendicular vector satisfying the right-hand convention as $\mathbf{y}_1$. The control law for level curve tracking applied to the formation center r_c is

$$u_c = \kappa_1 \cos\alpha + \kappa_2 \sin\alpha - 2\tilde{f}(z_c)||\nabla z_c|| \cos^2\left(\frac{\alpha}{2}\right) + K_4 \sin\left(\frac{\alpha}{2}\right), \tag{3}$$

where α is the angle between $\mathbf{x}_1$ and the moving direction of the formation, $\tilde{f}(z_c)$ is a function satisfying $\tilde{f}(C) = 0$, and $\tilde{f}(C) \neq 0$ if $z_c \neq C$, $\kappa_1 = -\frac{\mathbf{x}_1^T \nabla^2 z_c \mathbf{x}_1}{||\nabla z_c||}$ and $\kappa_2 = \frac{\mathbf{x}_1^T \nabla^2 z_c \mathbf{y}_1}{||\nabla z_c||}$ are two parameters related to the curvature of the level curve that can be estimated through computing the Hessian of the field $\nabla^2 z_c$, and K_4 is a constant. z_c represents the field value at the formation center r_c. This control law does not take into account the presence of any obstacles in the path of the sensor network.

To achieve obstacle avoidance, we use the potential field method [10] to add obstacle avoidance repulsive force to the steering control law. Consider M obstacles. Then the repulsive force from the i^{th} obstacle is given by:

$$F_i^{rep}(r_c, r_i^{obst}, l) = \nabla U_i^{obst}(r_c, r_i^{obst}, l) \\ = c_{obst} \cdot \log\left(\frac{||r_c - r_i^{obst}||}{l}\right), \quad (4)$$

where U_i^{obst} defines the potential field, l refers to the sensor range, $r_i^{obst} = (x_i^{obst}, y_i^{obst})$ provides us with the location of the i^{th} obstacle, and c_{obst} is a constant, which helps in defining how much farther the formation center should move away from the obstacle. Using the log will help us easily set the boundary at $\log(1) = 0$ and if it goes above zero it would mean that there would be no repulsive force. Hence, we sum up the repulsive forces from all the M obstacles as $F^{obst}(r_c) = c_{obst} \sum_{i=1}^{M} F_i^{rep}(r_c, r_i^{obst}, l)$, where $F^{obst}(r_c)$ is the total amount of repulsive force applied to the center of the formation, and add $F^{obst}(r_c)$ to the steering control law in Equation (3), which produces the final motion control law:

$$u_c^* = \kappa_1 \cos\alpha + \kappa_2 \sin\alpha \\ - 2f(\tilde{z_c})||\nabla z_c|| \cos^2\left(\frac{\alpha}{2}\right) + K_4 \sin\left(\frac{\alpha}{2}\right) + F^{obst}(r_c). \quad (5)$$

For a static field, two important inputs to the control law (5) are the field value z_c and gradient ∇z_c at the formation center r_c at each time step k, denoted as $z(r_c^k, k)$ and $\nabla z_c(r_c^k, k)$, which remain constant for the given position r_c^k over time. However, in a spatial-temporal varying field, z_c and ∇z_c change over time at a given position, so we need to obtain the delayed values while implementing the control law, i.e., z_c and ∇z_c at the same position r_c^k at time step $k+1$, denoted as $z(r_c^k, k+1)$ and $\nabla z_c(r_c^k, k+1)$, so that when the control law is applied to the center of the formation, the correct values are used.

IV. Cooperative Filtering

In this section, we discuss how to obtain the delayed estimates of the field value z_c and gradient ∇z_c at a given position through a constrained cooperative Kalman filter to enable the motion control law (5).

Cooperative Kalman filter was first developed in [6] to estimate the field value and gradient, i.e., $z(r_c^k, k)$ and $\nabla z(r_c^k, k)$, at the formation center at time step k in a static field. In order to provide the real-time estimate of the diffusion coefficient θ in an advection-diffusion field, reference [11] improved the cooperative Kalman filter with modified state and measurement equations that take into consideration the field dynamics, which results in a constrained cooperative Kalman filter that provides two additional estimates $z(r_c^k, k+1)$ and $\nabla z(r_c^k, k+1)$ at each time step. The difference is that $z(r_c^k, k+1)$ and $\nabla z(r_c^k, k+1)$ are the field value and gradient at position r_c^k at time step $k+1$, which are different from the values at time step k at the same position r_c^k due to the time-varying nature of the advection-diffusion field. The constrained cooperative Kalman filter requires the estimated diffusion coefficient θ. To track a level curve in an advection-diffusion field, $z(r_c^k, k+1)$ and $\nabla z(r_c^k, k+1)$ are needed in the steering control law (5) due to the latency in applying the control law in practice while the estimate of the diffusion coefficient θ is not necessary. Therefore, we adopt the constrained cooperative Kalman filter in [11] in this paper with modifications so that we are able to estimate $z(r_c^k, k+1)$ and $z(r_c^k, k+1)$ of the field with unknown diffusion coefficient.

To construct this cooperative Kalman filter, we need to construct a state equation and a measurement equation. To locally approximate the field values, we make use of Taylor expansion as shown in the following two equations.

$$z(r_i^k, k) \approx z(r_c^k, k) + (r_i^k - r_c^{k-1})\nabla z(r_c^k, k) \\ + \frac{1}{2}(r_i^k - r_c^{k-1})^T H(r_c^{k-1}, k)(r_i^k - r_c^{k-1}). \quad (6)$$

$$z(r_i^k, k+1) \approx z(r_c^k, k+1) + (r_i^k - r_c^{k-1})\nabla z(r_c^k, k) \\ + \frac{1}{2}(r_i^k - r_c^{k-1})^T H(r_c^{k-1}, k+1)(r_i^k - r_c^{k-1}). \quad (7)$$

In the above equations, $H(\cdot)$ represents the Hessian of the field. We define the state estimated using Kalman filter as $X(k+1) = [z(r_c^k, k), \nabla z(r_c^k, k), z(r_c^k, k+1), \nabla z(r_c^k, k+1)]$. By using first order Taylor expansion and finite difference method and expressing them in terms of matrix multiplication, we can write the prediction model of cooperative Kalman as

$$X(k+1)^- = A_x(k)X(k) + h(k) + w(k), \quad (8)$$

where $w(k)$ is the the i.i.d noise with covariance matrix Q. $A_x(k)$ and $h(k)$ can be defined as $A_x(k) = \begin{bmatrix} 1 & \left(r_c^k - r_c^{k-1}\right)^T & 0 & 0 \\ 0 & I_{2\times2} & 0 & 0 \\ 0 & 0 & 1 & \left(r_c^k - r_c^{k-1}\right)^T \\ 0 & 0 & 0 & I_{2\times2} \end{bmatrix}$ and $h(k) = \begin{bmatrix} 0 \\ H\left(r_c^{k-1}, k-1\right)\left(r_c^k - r_c^{k-1}\right) \\ 0 \\ H\left(r_c^{k-1}, k\right)\left(r_c^k - r_c^{k-1}\right). \end{bmatrix}$. We also have the field value measurements from the N sensor agents. So we denote these measurements as $P(k) = [p(r_1^{k-1}, k-1), \cdots, p(r_N^{k-1}, k-1), p(r_1^k, k), \cdots, p(r_N^k, k)]^T$. The measurement equation can be modeled as

$$P(k) = C(k) \cdot X(k)^- + D(k)\vec{H}(k) + v(k), \quad (9)$$

where $\vec{H}(k) = [\vec{H}(r_c^{k-1}, k-1), \vec{H}(r_c^{k-1}, k)]^T$ represents the estimate of the Hessian at the center of the formation r_c^k and $v(k)$ represents the i.i.d noise. $C(k)$ and $D(k)$ can be defined as $C(k) = \begin{bmatrix} 1 & \left(r_1^{k-1} - r_c^{k-1}\right)^T & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \left(r_N^{k-1} - r_c^{k-1}\right)^T & 0 & 0 \\ 0 & 0 & 1 & \left(r_1^k - r_c^{k-1}\right)^T \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & \left(r_N^k - r_c^{k-1}\right)^T \end{bmatrix}$ and $D(k) =$

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

where $\otimes$ is the Kronecker product. Using Equations (8) and (9), we are able to construct a constrained cooperative Kalman filter following the equations in [11]. The constraint comes from the dynamic equation (1) and is a function of the estimated diffusion coefficient θ. There are two ways to deal with the constraint. First, we can follow the strategy in [11] to estimate θ in real time. Or second, we can substitute an estimated value into the constraint. We omit the equations here due to the page limit. With the constructed cooperative Kalman filter, the estimated state $X(k+1)$ can be obtained and $z(r_c^k, k+1)$ and $\nabla z(r_c^k, k+1)$ will be used in the steering control law. For the Hessian estimation, we use the same approach as in [6] so we omit the derivations.

V. Simulation Results

In this section, we verify the proposed strategy using simulations. We first summarize the step-by-step algorithm, then implement the algorithm in a simulated advection-diffusion field with obstacles using four sensing agents.

A. The cooperative level curve tracking algorithm

To verify the cooperative level curve tracking strategy, we first construct an advection-diffusion simulated field using finite difference method [12]. Using this method, we can define parameters such as the diffusion coefficient and the flow velocity to make the field concentration diffuse outwards overtime and also change its source location at each time step.

Algorithm 1 provides a brief overview of how the center of the formation r_c gets updated over time while tracking the desired field value in the simulated field. We start by setting the discretization time step denoted as sim_dt at which the diffusion of the field happens. The notation $agent\_dt$ determines the rate at which sensors capture the field values in the simulation. In addition, we need to set the initial positions of the agents $r_1 \cdots, r_4$ with a constraint that they need be within the bounds of the simulated field. Along with this, we will also set Δr_x and Δr_y, which are the max distance between agents along x and y directions, respectively. In the case of using 4 agents, we will align themselves in rhombus formation over time.

Since this is a simulated environment, we define the total number of times the simulated map will be updated by setting t_{max}. At each time step k, we use the cooperative Kalman filter defined in section IV to determine the field concentration and it's gradient at r_c at both k^{th} and $(k+1)^{th}$ time step. Using the estimated Hessian of the field, we apply the motion control algorithm as explained in section III-B to obtain where the center of the formation should move to at $k+1$ timestep. With the updated r_c^* we will perform formation control as explained

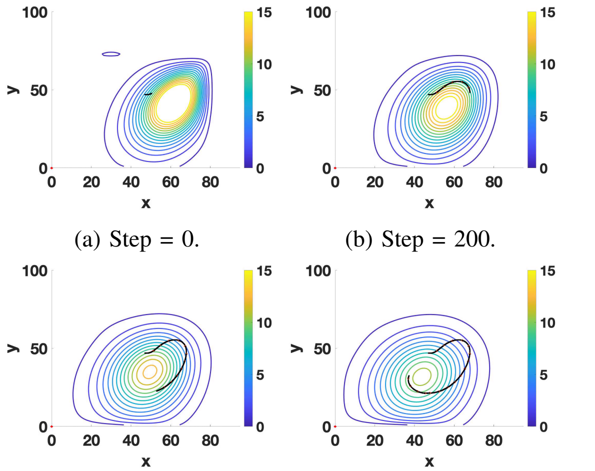
Algorithm 1: Level Curve Tracking in an Advection-Diffusion Field

Require: Simulated field, the positions of the agents $\mathbf{r}_k = \{r_1^k, \cdots, r_4^k\}$, and the measured fields at each time step $\mathbf{p}_k = \{p_1^k, \cdots, p_4^k\}$.
load u; //load the simulated field;
// set the discretization time steps for the field and agent motion ;
$sim\_dt \leftarrow 0.01$;
$agent\_dt \leftarrow 0.01$;
$z\_desired \leftarrow C$;
set Ψ, $\Delta r_x, \Delta r_y$;
// set current time step k and max time steps t_{mac};
set $k \leftarrow 0$;
set t_{max};
initialization;
while $k \leq t_{max}$ **do**
  load r;
  calculate $\nabla z^2(r_c^k, k)$;
  $S = [z(r_c^k, k), \nabla z(r_c^k, k), z(r_c^k, k+1), \nabla z(r_c^k, k+1)]$;
  $S^* =$ CoopKalmanFilter$(\mathbf{p}_k, \nabla^2 z(r_c, k), \mathbf{r}_k, r_c^k, S)$;
  // Apply motion control explained in section III-B;
  $r_c^{k+1} =$ MotionControl$(z_{desired}, r_c^k, S^*, \nabla^2 z(r_c^k, k))$;
  // Apply formation control in section III-A to calculate the next positions where the $N = 4$ agents should move to.
  $\mathbf{r}_{k+1} =$ FormationControl$(\mathbf{r}_k, \Psi, r_c^{k+1})$;
  // update the simulated field;
  Update(**u**, sim_dt);
end

in section III-A to determine where the agents should move in the next time step.

B. Simulation Results

We conduct simulations to verify the proposed cooperative level curve tracking strategy based on Algorithm 1. We first generate an advection-diffusion field without obstacles and test the level curve tracking behavior of the sensing agents. Figures 2 (a) - (d) illustrate the level curve tracking process using four sensing agents in the simulated advection-diffusion field. As shown in the figures, the field is gradually diffusing while also advecting towards the lower left corner. The black lines in the figures represent the trajectory of the center of the formation. The formation center starts at an arbitrary position with field concentration less than 5, as shown in Figure 2 (a). It gradually approaches the level curve with field concentration of 10, which is the desired field value $z\_desired$, as shown in Figures 2 (b) - (d). Figure 3 shows the level values at the center of the formation along its trajectory. Observing the figures, the level value at the center of the formation gradually converges to the desired level value.



(a) Step = 0. (b) Step = 200.

(c) Step = 500. (d) Step = 1000.

Fig. 2: The level curve tracking process using four sensing agents in a simulated advection-diffusion field. The black lines represent the trajectory of the formation center. The formation center starts at an arbitrary position with field value less than 5, as shown in (a). It gradually approaches and tracks the level curve with $z\_desired = 10$, as shown in (b) - (d).

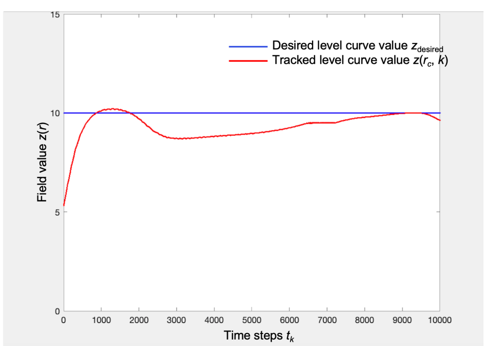
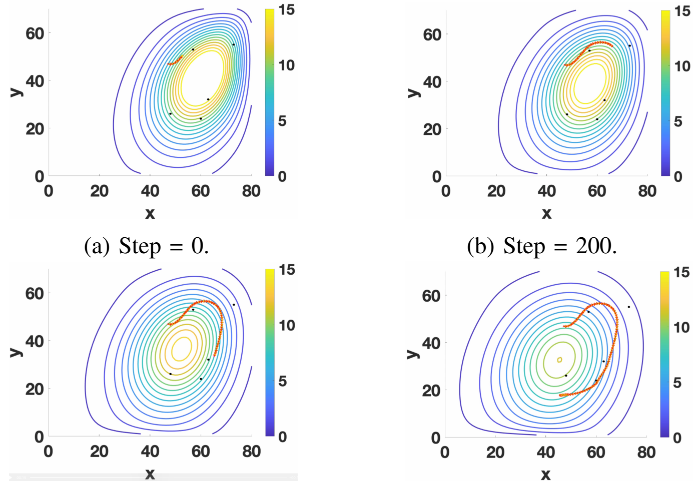


Fig. 3: The field values at the center of the formation v.s. the desired field value.

Next, we conduct simulations in an advection-diffusion field with obstacles. The results are demonstrated in Figures 4 (a) - (d). All the settings are the same as the first simulation except that we add obstacles in the field. So the obstacle avoidance part is activated for the motion control of the center of the formation. In Figures 4 (a) - (d), the black dots represent the obstacles. We can observe from the figures that the center of the formation is able to detect and track the desired level curve value while avoiding obstacles.



(a) Step = 0. (b) Step = 200.

(c) Step = 500. (d) Step = 1000.

Fig. 4: The level curve tracking process in a simulated advection-diffusion field with obstacles. The black dots represent the obstacles and the red lines represent the trajectory of the formation center. The formation center gradually approaches and tracks the level curve with $z\_desired = 10$ while avoiding obstacles as shown in (a) - (d).

VI. Conclusions and Future Work

In this paper, we developed a level curve tracking strategy that allows a group of sensing agents to detect and track a level curve with the desired level value in an advection-diffusion field with obstacles. A cooperative Kalman filter is implemented to provide estimated state of the spatial-temporal varying field so that the motion control laws can be enabled to control the center of the formation. We provide the detailed algorithm for the implementation and performed simulation study. The future work includes the extension of the algorithm to other types of spatial-temporal varying fields and the verification of the proposed strategy in realistic fields.

References

[1] R. Ghez, *Diffusion phenomena: cases and studies*. Springer Science & Business Media, 2013.

[2] C. H. Hsieh, Z. Jin, D. Marthaler, B. Q. Nguyen, D. J. Tung, A. L. Bertozzi, and R. M. Murray, "Experimental validation of an algorithm for cooperative boundary tracking," in *Proceedings of the 2005 American Control Conference*, 2005, pp. 1078–1083.

[3] D. Zarzhitsky, D. F. Spears, and W. M. Spears, "Swarms for chemical plume tracing," in *Proceedings of the IEEE Swarm Intelligence Symposium*, 2005, pp. 249–256.

[4] J. Clark and R. Fierro, "Cooperative hybrid control of robotic sensors for perimeter detection and tracking," in *Proceedings of the 2005, American Control Conference*, 2005, pp. 3500–3505.

[5] A. H. Mong-ying and V. Kumar, "Pattern generation with multiple robots," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006, pp. 2442–2447.

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

[7] Z. Jin and A. L. Bertozzi, "Environmental boundary tracking and estimation using multiple autonomous vehicles," in *Proceedings of 46th IEEE Conference on Decision and Control*, 2007, pp. 4918–4923.

[8] W. Wu and F. Zhang, "Cooperative exploration of level surfaces of three dimensional scalar fields," *Automatica*, vol. 47, no. 9, pp. 2044–2051, 2011.

[9] M. Stein, "A simple model for spatial-temporal processes," *Water Resources Research*, vol. 22, no. 13, pp. 2107–2110, 1986.

[10] W. Wu, J. You, Y. Zhang, and L. Mingchen, "Parameter identificaion of spatial-temporal varying process by a multi-robot system in realistic diffusion fields," *accepted, Robotica*, 2020.

[11] J. You, F. Zhang, and W. Wu, "Cooperative filtering for parameter identification of diffusion processes," in *Proceedings of the IEEE 55th Conference on Decision and Control*, 2016, pp. 4327–4333.

[12] H. P. Langtangen, "Finite difference methods for diffusion processes," *University of Oslo*, 2013.