

# Dynamic Data Driven Application System for Plume Estimation Using UAVs

Liqian Peng · Doug Lipinski · Kamran Mohseni

Received: 31 August 2013 / Accepted: 16 September 2013 / Published online: 6 October 2013  
© Springer Science+Business Media Dordrecht 2013

**Abstract** In this article, a full dynamic data-driven application system (DDDAS) is proposed for dynamically estimating a concentration plume and planning optimal paths for unmanned aerial vehicles (UAVs) equipped with environmental sensors. The proposed DDDAS dynamically incorporates measured data from UAVs into an environmental simulation while simultaneously steering measurement processes. In order to assimilate incomplete and noisy state observations into this system in real-time, the proper orthogonal decomposition (POD) is used to estimate the plume concentration by matching partial observations with pre-computed dominant modes in a least-square sense. In order to maximize the information gain, UAVs are dynamically driven to hot spots chosen based on the POD modes. Smoothed particle hydrodynamics (SPH)

techniques are used for UAV guidance with collision and obstacle avoidance. We demonstrate the efficacy of the data assimilation and control strategies in numerical simulations. Especially, a single UAV outperforms the ten static sensors in this scenario in terms of the mean square error over the full time interval. Additionally, the multi-vehicle data collection scenarios outperform the single vehicle scenarios for both static sensors at optimal positions and UAVs controlled by SPH.

**Keywords** DDDAS · UAV · Environmental measurement · POD · Path planning · SPH

## 1 Introduction

The use of cooperative, small unmanned aerial vehicles (UAVs) equipped with simple environmental sensors provides a promising option for safe and cost-effective data collection. Two benefits of such a system are the relatively low-cost of individual vehicles and the safety of using unmanned vehicles for missions involving harsh atmospheric conditions or toxic environments. The mobility of UAVs allows for information gathering over large areas and the ability to backtrack, providing for more complete data collection in a dynamic environment when compared to static sensors. A sample of the work that has been conducted in the last few years using smaller and lower-cost

---

The authors acknowledge the support from the US Air Force Office of Scientific Research (AFOSR).

L. Peng · D. Lipinski · K. Mohseni  
Department of Mechanical and Aerospace  
Engineering, Institute for Networked Autonomous  
Systems, University of Florida,  
Gainesville, FL 32611, USA

K. Mohseni (✉)  
Department of Electrical and Computer Engineering,  
University of Florida, Gainesville, FL 32611, USA  
e-mail: mohseni@ufl.edu

vehicles equipped with atmospheric sensors can be found in [1–6]. The extensive amount of data obtained from small, cooperative UAVs equipped with simple, low accuracy sensors allows for the potential of more efficient data collection compared to a single vehicle with more expensive, more accurate sensors.

Regardless of the number of UAVs employed in an application, the key consideration in the use of UAVs is how to position them in order to gain useful information from their locations and/or path. Dynamic data driven application systems (DDDAS) provide a means to position the UAVs in an efficient manner using the real time data obtained from the sensors. The framework of DDDAS is driven by the goal of dynamically incorporating data into a running application (e.g. an environmental simulation) while simultaneously using the application to steer measurement processes [7]. The DDDAS framework is widely used in wildfire simulation [8, 9], identification of airborne contaminants [10–12], and weather forecasting [13].

In this article, a two-dimensional plume evolution problem is considered. The concentration plume is released upwind of two square obstacles (representing buildings) and travels downstream according to the advection-diffusion equation. For real-time plume simulation and estimation, the unknown parameters in our problem include not only the initial condition of the plume concentration, but also the plume source distribution. This leads to an ill-posed inverse problem with high-dimensional control spaces in the form of unknown parameters, whose solution is not unique. This motivates the construction of reduced-order surrogate models to uniquely determine these parameters in DDDAS. It is noticed that the proper orthogonal decomposition (POD) [14] seeks to replace a high-dimensional system with a system of substantially lower dimension while preserving the dominant features of the original system. An offline-online splitting strategy is applied to achieve fast online computation. In an offline stage, we sample the possible plume source locations and solve the advection-diffusion equation, which results in possible trajectories of the plume concentration. Then, the POD method is used to identify the dominant modes of plume concentra-

tion. In the online stage, point observations of the plume concentration are recorded by UAVs and an estimation of the global plume concentration is computed by fitting the dominant POD modes to the observations in a least squares sense. Once the plume concentration is known, it is possible to solve for the plume source term in the governing equation.

The proposed DDDAS framework also creates solutions for efficient data collection and real-time vehicle control. UAVs are used to collect concentration data throughout the domain which is then assimilated into a running advection-diffusion simulation to predict the plume motion. In order to maximize the information gain of UAVs, the DDDAS dynamically drives the vehicles to *hot spots* that contribute the most information to the dominant modes. A smoothed particle hydrodynamics based scheme is used for vehicle control to provide obstacle and collision avoidance. Once the vehicles are driven to the hot spots and collect the data, the data assimilation process will estimate the instantaneous plume concentration as well as the plume source. We demonstrate the efficacy of the assimilation strategy as well as the control strategy of this dynamic data-driven application system via numerical simulations.

The main contributions of this paper are twofold: (1) the use of the POD method to reduce the dimension of a solution space for the data assimilation process, such that the estimation for plume concentration can be solved by a least-squares optimization process and (2) a control strategy to drive vehicles to the hot spots such that the main POD modes are captured. The remainder of this article is organized as follows. We first formulate the problem of plume evolution within an urban environment. We then discuss and present the reduced-order data assimilation procedure. Next, we discuss the path planning strategy for mobile vehicles. Simulation results are also provided. Finally, we give a short conclusion.

## 2 Problem Formulation

The DDDAS approach is quite general and can be applied to many phenomena where good simulation models are available. Here, we focus on

measuring and tracking a concentration plume from sparse observations in an advection-diffusion system described by

$$\frac{\partial \psi}{\partial t} + \mathbf{v} \cdot \nabla \psi = \frac{1}{Pe} \nabla^2 \psi + f; \quad \psi(0, x) = \psi_0(x), \tag{1}$$

where  $\mathbf{v} = \mathbf{v}(t, x)$  is the background wind velocity,  $Pe$  is the Péclet number which quantifies the relative strength of the advection and diffusion terms,  $\psi(t, x)$  is the concentration field,  $f(t, x)$  is the plume source, time  $t \in (0, T)$ , and  $x \in \Omega$ . We examine the case where a Gaussian plume source releases a constant plume, which is advected under the action of the velocity field  $\mathbf{v}$  while also dispersing due to the diffusion. After discretization in space and time through finite differences, Eq. 1 yields

$$\psi_k = A_{k-1} \psi_{k-1} + f_{k-1}; \tag{2}$$

where  $\psi_k \in \mathbb{R}^n$  is the concentration at time step  $k$ ,  $A \in \mathbb{R}^{n \times n}$  is the discretized linear operator accounting for the advection and diffusion effects and  $f \in \mathbb{R}^n$  is the plume source. In this model, we ignore the noise introduced by the uncertainty in the background velocity field, the diffusion constant, and the concentration source, for simplicity. The wind velocity  $\mathbf{v}$  could be obtained by a CFD model or weather forecasting. In this paper,  $\mathbf{v}$  is found by numerically solving the incompressible NS equations using a finite difference method at  $n$  grid points. The wind velocity  $\mathbf{v}$  varies with time. Therefore, both Eqs. 1 and 2 are linear non-autonomous systems.

The advection-diffusion equation is solved to simulate the “real” plume concentration by using a semi-Lagrangian advection step and an implicit central difference scheme for the diffusion term as described in [15]. To limit the numerical diffusion and improve accuracy, a second order midpoint method is used for the advection step and cubic splines are used to interpolate from the grid to the advection points.

For the case of a discrete puff of concentration being released, the forcing term  $f$  in Eq. 1 is zero for  $t > 0$ , and the governing equation is completely known. For this reason, puff estimation problems may be solved by using a Kalman

filter technique or via optimization for the initial condition [10–12]. However, for plume estimation,  $f$  is an unknown, infinite dimensional parameter giving the plume source. Additionally, estimation of  $f$  is an inverse problem with a generally non-unique solution. To address this, we must add some additional assumptions for well-posedness, such as  $f$  is smooth and compactly supported. Let  $\mathcal{V} = \text{supp}(f)$ , so  $\mathcal{V}$  is bounded. In addition, we seek an approximating solution  $\hat{f}$  that resides on a linear subspace, which is spanned by a set of basis functions  $\{\phi_i(x)\}_{i=1}^v$  with  $\text{supp}(\phi_i) \subset \mathcal{V}$ , and

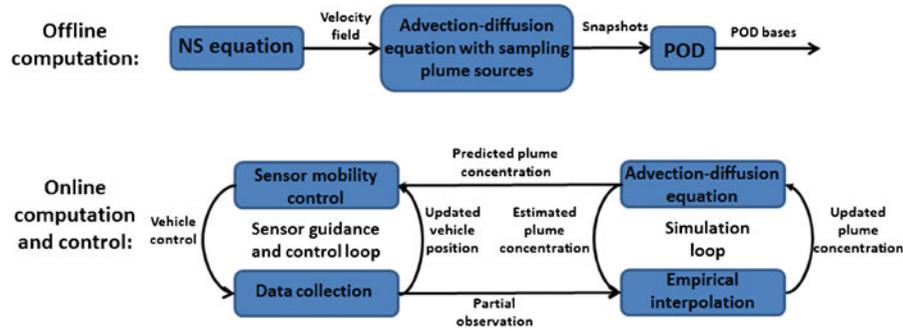
$$\hat{f}(t, x) = \sum_{i=1}^v a^i(t) \phi_i(x). \tag{3}$$

For the discrete case,

$$\hat{f}_k = \sum_{i=1}^v a_k^i \phi_i, \tag{4}$$

where  $a_k^i$  denotes the coefficient corresponding to basis vector  $\phi_i$  at time step  $k$ ,  $\hat{f}_k \in \mathbb{R}^n$ , and  $\phi_i \in \mathbb{R}^n$ . In the offline stage, we can precompute the trajectories of the plume concentration for the basis functions  $\{\phi_i\}_{i=1}^v$ , which will span another linear subspace  $\mathcal{U}$ . If the plume source  $\hat{f}_k$  is a constant for  $k > 0$ , and the initial plume concentration is zero, then the trajectory of  $\hat{f}_k$  resides on  $\mathcal{U}$ .

The methods used in this paper combine to form a full dynamic data-driven application system (DDDAS) [7] to effectively and efficiently control UAVs and estimate plume concentration. This system can be thought of as an intelligent autonomous control system [16] designed to perform well under significant uncertainties in an unknown environment for extended periods of time. A schematic of this system is shown in Fig. 1. An offline-online splitting methodology is applied. In the offline stage, we sample the possible plume source locations and solve the advection-diffusion equation, which results in possible trajectories of the plume concentration. This process could be very expensive if a large ensemble of plume sources are computed. However, in this DDDAS application, where the goal is to achieve a very low marginal cost to compute real-time plume concentration, we can accept an increased offline cost



**Fig. 1** (Color online) Schematic of the DDDAS. In the offline stage, the wind velocity is obtained by solving the NS equations. Then some sample plume concentration solutions are computed using the advection-diffusion equation with a sample of basis plume sources. The POD

method is applied to obtain the dominant modes. In the online stage, the two main components of the system are a sensor guidance and control loop and a simulation loop. The two loops interact through data assimilation and sensor placement algorithms

in exchange for greatly decreased online cost for data assimilation and vehicle control procedures. In the final offline step, the dominant modes of plume concentration are identified by the POD method. In the online stage, two coupled feedback loops are run in real-time to improve the accuracy of a running simulation while simultaneously improving the effectiveness of data collection strategies. We refer to the first loop in the DDDAS as the sensor guidance and control loop. The system components in this loop are responsible for movement, collision avoidance, and data collection. The second loop in the system is referred to as the simulation loop. In this loop, a real-time simulation attempts to model the physical system of interest; in our case, a concentration plume is modelled using the advection-diffusion equation mentioned above. The key to the DDDAS is an effective, two-way coupling between the simulation loop and the sensor guidance and control loop. This is accomplished by using a POD based data assimilation routine to assimilate data collected in the guidance and control loop into the running simulation loop, resulting in improved accuracy in the simulation. Meanwhile, in order to improve the data collection process, the simulation loop could be used to adaptively construct local POD modes, which are then applied in the proposed sensor placement algorithm to steer the UAVs to impactful measurement locations.

In the next two sections, we first discuss the data assimilation process used in this paper, then move on to introduce the UAV control scheme

that is used to ensure sensor collision and obstacle avoidance while guiding sensors to optimal locations.

### 3 Data Assimilation

Since observations are incomplete and noisy, it is not sufficient to estimate the instant plume concentration in the high dimensional space  $\mathbb{R}^n$  with the measured data alone. However, if we restrict this inverse problem to a subspace  $\mathcal{S}$  with a lower number of unknown parameters, the measured data is sufficient to find an approximate solution for plume concentration on  $\mathcal{S}$  with high accuracy. The error between the real plume concentration  $\psi$  and the estimated concentration  $\hat{\psi}$  can be defined as  $e := \hat{\psi} - \psi$ . Let  $e_o$  denote the error component orthogonal to  $\mathcal{S}$ , and  $e_i$ , denote the component of error parallel to  $\mathcal{S}$ . The reduced-order data assimilation technique proposed in this section seeks to decrease  $e_o$  and  $e_i$  simultaneously.

It is noted that  $e_o$  directly comes from the projection onto the subspace  $\mathcal{S}$ . The proper orthogonal decomposition method (POD, also known as Karhunen-Loève decomposition, singular systems analysis, singular value decomposition and principle component analysis, PCA) [17] is used to capture  $\mathcal{S}$ . In the fields of dynamical systems and computational fluid dynamics, POD is used to formulate a simplified surrogate of the original high-dimensional problem to improve computational efficiency. In the fields of machine learning, signal processing, and image processing, POD is

used to find dominant patterns in signals or input parameters. In the context of DDDAS, POD was used in [11, 18] for the former purpose. In this paper we use POD for the later purpose.

On the other hand, since the observation is noisy, we cannot completely remove  $e_i$ . However, within the subspace  $\mathcal{S}$ , we can optimize the coefficients for each basis vector by the regularized least mean square method. Statistically, this method gives an optimal estimation for all the collected data to reduce  $e_i$  while ensuring that the reformulating inverse problem is well-posed. The following two subsections discuss the techniques that reduce  $\|e_o\|$  and  $\|e_i\|$ .

### 3.1 Proper Orthogonal Decomposition

In many applications, conventional methods of direct numerical simulation (DNS) for resolving all scales in a system are computationally intensive, and therefore reduced order models (ROMs) are often desirable. The core of model order reduction is to provide an efficient computational prototyping tool to replace a high order system of differential equations with a system of substantially lower dimension, whereby only the most dominant properties of the original system are preserved. As a model reduction technique, POD is widely used in CFD-based modelling and control [19], electrical power grids [20, 21], structural dynamics [22], and chemical reaction systems [23, 24] to name a few. Although POD is a linear model reduction technique which uses a linear subspace to represent generally nonlinear submanifolds, it is computationally tractable and typically requires only a few modes to capture the most important aspects of a system. Roughly speaking, a typical POD procedure involves the offline-online strategy: given an ensemble of data, collected from a set of numerical simulations of the original dynamical system, find a linear subspace representation of the data via truncated singular value decomposition (SVD) in the offline stage. Then the reduced order model is obtained via Galerkin projection during computations in the online stage.

The key idea of POD is to deliver a set of empirical eigenfunctions (or POD modes) so that the original data is optimally captured by these

modes in the least squares sense [14]. Suppose a data ensemble for different trajectories is given by  $\{\psi(t)\}_\alpha$  with  $t \in [0, T]$ , where  $\alpha$  is the index for a trajectory calculated from the original system. POD seeks a subspace  $\mathcal{S}$  of fixed dimension  $d$  that minimizes the total error of all state variables in the data ensemble and their projection on  $\mathcal{S}$  [14]. In the numerical simulation, it is more feasible to rewrite the entire data ensemble in a discretized form,

$$X := [\psi_1, \dots, \psi_m], \tag{5}$$

where the subscript denotes the different snapshots in different trajectories. The data ensemble  $X$  can be decomposed by singular value decomposition (SVD),

$$X = U \Lambda V^T, \tag{6}$$

where  $U \in \mathbb{R}^{n \times r}$  and  $V \in \mathbb{R}^{m \times r}$  are orthogonal, and  $\Lambda \in \mathbb{R}^{r \times r}$  is a diagonal matrix which consists of  $r = \min(n, m)$  nonnegative diagonal elements arranged in decreasing order, i.e.  $\lambda_1 \geq \dots \geq \lambda_r \geq 0$ . If the first  $d$  singular values are significantly larger than the rest, one can make a good approximation of  $X$  by only calculating  $d$  column vectors of  $U$  and  $V$  corresponding to the  $d$  largest singular values. This can be much quicker and more efficient than dealing with the full system if  $d \ll r$ . The POD basis  $\{\varphi_i\}_{i=1}^d$  is given by the first  $d$  columns of  $U$ , which span a subspace  $\mathcal{S}$ .  $\mathcal{S}$  can be represented by the following matrix in the Stiefel manifold

$$\Phi := [\varphi_1, \dots, \varphi_d] \in \mathbb{R}^{n \times d}. \tag{7}$$

In the rest of this paper, a SVD process refers to truncated SVD unless otherwise specified.

Let  $E_r$  denote the energy of the full system,  $E_r = \sum_{i=1}^r \lambda_i^2$  and  $E_d$  denote the energy in the optimal  $d$  dimensional subspace,  $E_d = \sum_{i=1}^d \lambda_i^2$ . The fractional error in energy  $\eta$  can be presented by

$$\eta = 1 - \frac{E_d}{E_r}. \tag{8}$$

The POD basis is optimal for model reduction in the sense that it provides a better energy approximation than any other linear basis in the same dimension with respect to the data ensemble  $X$ . If the state variables in  $X$  span a linear space

that contains  $\psi$ , then the orthogonal component of error  $\|e_o\|$  is completely determined by truncation, and can be estimated by  $\|e_o\| \approx \eta$ . Otherwise,  $\|e_o\|$  can be estimated by  $\|e_o\| \approx \xi\eta$ , and  $\xi > 1$  is a constant that denotes the deviation from  $\psi$  from the linear space determined by  $X$ .

In the online stage, Galerkin projection can provide a lower dimensional approximation by projecting the full system onto a linear (or an affine) subspace. With respect to Eq. 2, the reduced-order equation is given by left multiplication of  $\Phi$  on both sides of Eq. 2,

$$\tilde{\psi}_k = \tilde{A}_{k-1}\tilde{\psi}_{k-1} + \tilde{f}_{k-1}; \quad (9)$$

where  $\tilde{\psi}_k = \Phi^T \psi_k \in \mathbb{R}^d$ ,  $\tilde{f}_k = \Phi^T f_k \in \mathbb{R}^d$ , and  $\tilde{A}_k = \Phi^T A_k \Phi \in \mathbb{R}^{d \times d}$ . Since  $\tilde{\psi}_k$  is updated in a much lower dimensional space, calculation of Eq. 9 is much more efficient than Eq. 2. However, since the velocity field is time-dependent, it requires a lot of memory to save  $\tilde{A}_k$  for all the time steps. Therefore, although we use POD to find the dominant modes of plume concentration, we do not use Eq. 9 for online computation.

### 3.2 Empirical Interpolation Method

The original empirical interpolation method [25] and its variant, discrete empirical interpolation method (DEIM) [26], are used for model order reduction that reduces the complexity of evaluating the nonlinear term of the reduced model to a cost proportional to the number of reduced variables obtained by the POD method. Rather than calculating the nonlinear vector field of the original dynamical system in the original space, DEIM seeks a small number of critical points, and then interpolates the vector field by matching the values at these points. We are motivated by this idea for the DDDAS for plume detection in two ways. First, we send our UAVs to approach these critical points, or hot spots, to collect the most useful information about the system state. This control strategy will be discussed in the next section. Secondly, after measuring plume concentration at only a small number of the total grid points, we use the interpolation method to estimate the plume concentration for the entire spatial domain. In a slight abuse of terminology we refer to this method of fitting POD modes to

observations as *empirical interpolation*. The estimation process is based on the regularized least mean square method.

Let  $\psi_k \in \mathbb{R}^n$  denote the “real” plume concentration field at time step  $k$ .  $\Phi \in \mathbb{R}^{n \times d}$ , as defined in Eq. 7, formed by the first  $d$  empirical eigenfunctions. The approximation given by projecting  $\psi_k$  onto the subspace spanned by column vectors of  $\Phi$  is of the form

$$\psi_k \approx \Phi c_k, \quad (10)$$

where  $c_k \in \mathbb{R}^d$  is the coefficient vector with respect to  $\Phi$ . Suppose at time step  $k$ , we have  $l$  observations. Let  $\{\beta_i\}_{i=1}^l$  denote the indices of grid points corresponding to these observations. Let a matrix  $P = [e_{\beta_1}, \dots, e_{\beta_l}] \in \mathbb{R}^{n \times l}$ , where  $e_{\beta_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$  is the  $\beta_i$ th column of the identity matrix  $I_n \in \mathbb{R}^{n \times n}$  for  $i = 1, \dots, l$ . Let  $\xi_k \in \mathbb{R}^l$  denote the noise vector in the observation and  $\omega_k \in \mathbb{R}^l$  denote the measured data at time step  $k$ . Then, the relationship between the original plume field  $\psi_k$  and the observation  $\omega_k$  can be expressed as

$$\omega_k = P_k^T \psi_k + \xi_k, \quad (11)$$

where we add subscript  $k$  for the projection matrix  $P$  to emphasize that the observation could be taken at different positions and different times, which is the case for the UAV applications in this article.

The aim is to find an estimation of plume concentration  $\hat{\psi}_k$  to minimize the  $\ell^2$  error between the estimated plume concentration  $P_k^T \hat{\psi}_k$  at measured grid points and the real observation  $\omega_k$ . In order to obtain a unique, well-behaved solution, we seek  $\hat{\psi}_k$  on the subspace spanned by the column vectors of  $\Phi$ . Substituting  $\hat{\psi}_k$  by the RHS of Eq. 10 yields

$$\hat{\psi}_k = \Phi c_k. \quad (12)$$

By defining the quadratic function

$$\mathcal{E}(c_k) = \frac{1}{2} \|P_k^T \Phi c_k - \omega_k\|^2 + \frac{\alpha}{2} \|c_k\|^2 \quad (13)$$

with a regularization coefficient  $\alpha$ , the plume estimation problem is reduced to minimizing  $\mathcal{E}(c_k)$  with respect to  $c_k$ , which leads to

$$c_k = (\Phi^T P_k P_k^T \Phi + \alpha I)^{-1} \Phi^T P_k \omega_k, \quad (14)$$

where  $I$  denotes the  $d \times d$  identity matrix. After solving  $c_k$ , the approximation  $\hat{\psi}_k$  is given by Eq. 12 which is the regularized least squares estimation of plume concentration.

After solving for  $\hat{\psi}_k$  for two consecutive time steps, based on the governing equation of plume evolution (Eq. 2), the instantaneous plume concentration could be estimated as

$$g_{k-1} \approx \hat{\psi}_k - A_{k-1} \hat{\psi}_{k-1}. \tag{15}$$

Usually, there is no guarantee that  $\text{supp}(g_{k-1})$  is bounded due to the error introduced by  $\psi_{k-1}$  and  $\psi_k$ . If the basis functions for the plume source  $\{\phi_i\}_{i=1}^v$  in Section 3.1 are orthonormal, one may filter the noise and limit the estimation of plume source within the domain  $\mathcal{V}$  by

$$\hat{f}_{k-1} = \sum_{i=1}^v \phi_i \phi_i^T g_{k-1}. \tag{16}$$

Otherwise, one may use the Gram-Schmidt process to reconstruct an orthonormal basis for the same subspace and use the above equation to filter the error from  $g_{k-1}$ . Based on this empirical interpolation method, a control strategy for UAVs is presented in the next section to obtain maximal measured information and to minimize the estimation error.

### 4 Sensor Mobility Control

Although the above mentioned data assimilation method can effectively reduce the stochastic noise in the measurements, the final estimation of plume concentration is highly dependent on where the data is collected. In the worst case, if all the UAVs are located in a region where there is no plume concentration, the least squares problem is ill-conditioned and the results will be *determined* by the stochastic measurement noise rather than helping to reduce this noise. Sensor mobility control refers to a strategy for gathering sensor measurements to support a sensing objective, such as environmental measurement. When the sensors are installed on robotic platforms an important part of the problem is planning the sensor path to achieve low working time or low energy consumption [27], obstacle avoidance in

unstructured dynamic environments [28–30], or efficiently gather target information [31]. Using the POD-based data assimilation method, a hierarchical control strategy is proposed here to maximize the useful information collected by mobile sensors. Based on the speed limitation of UAVs, the entire time domain is artificially partitioned into  $N$  smaller subintervals such that during each subinterval, any UAV has the ability to reach (or at least approach) any important measurement location. Inspired by the greedy algorithm in [26], we develop an algorithm to find the important positions, or hot spots, for a fixed time interval where sensors may obtain more information than arbitrarily chosen positions. At a lower level, smoothed particle hydrodynamics (SPH) is used to guide UAVs to the chosen hot spots at each time interval while simultaneously avoiding collisions with obstacles or other sensors.

#### 4.1 Sensor Placement

The sensor placement procedure is employed to choose sensor locations at hot spots in the domain to improve the estimation of plume concentration. There are several common strategies to address this fundamental task to maximize the entropy or mutual information [31, 32]. For our DDDAS application, a small number of UAVs have the potential to gain much more information than fixed sensors, but also provide more challenges. Since physical plumes typically have a large Péclet number, their evolution is mainly governed by the advection term in Eq. 1, and the concentration varies in time due to the time varying velocity field. Therefore, we cannot expect to obtain maximal information by setting hot spots at the same positions for the whole time domain. Instead, we need to dynamically locate hot spots for each time interval. By adaptively choosing snapshots in the database, we can calculate POD modes corresponding to the local time interval. For each POD mode, the greedy algorithm proposed in this section is a polynomial-time approximation to detect the most informative positions with respect to minimizing the  $\ell^2$  error for the estimated plume concentration. Compared with A-, D- or E- optimal design for entropy maximization [32], it is much more efficient.

After spatial discretization,  $m$  hot spots can be determined by the location indices  $\{\gamma_i\}_{i=1}^m$ . Due to the presence of obstacles (buildings) in the domain and the uncertainty of vehicle trajectories, some portions of the domain inaccessible or unsafe for UAV placement. We still use  $\{\varphi_i\}_{i=1}^d$  to denote the first  $d$  dominant modes for a given time interval, which change from one interval to another. We call these the local POD modes. The proposed algorithm is to choose  $\{\gamma_i\}_{i=1}^m$  such that at each time step the coefficients corresponding to these modes (Eq. 14) may be calculated with the highest accuracy in a statistical manner. Let  $\varphi_i(j) \in \mathbb{R}$  denote the  $j$ th element of mode  $i$ , then we can define a projection  $\tilde{\varphi}_i \in \mathbb{R}^n$  of  $\varphi_i$ , such that  $\tilde{\varphi}_i(j) = \varphi_i(j)$  when  $j$  is an accessible position for UAVs and  $\tilde{\varphi}_i(j) = 0$  otherwise.

The proposed greedy algorithm seeks to find the optimal positions for hot spots within each period, whereby the measured data at these positions enables accurate computation of the POD mode coefficients. Intuitively, the positions corresponding to higher values of  $\tilde{\varphi}_i$  are desirable because observations at these positions are certain to contain information about the contribution of mode  $\varphi_i$ . So we set  $\gamma_1$  to be the location of  $\tilde{\varphi}_1$  with the largest magnitude. The remaining interpolation indices,  $\gamma_i$  for  $i = 2, \dots, m$ , correspond to the largest magnitude of the residual  $r_i = \tilde{\varphi}_i - U d_i$ , where  $r_i$  is the residual or the error between the input basis  $\tilde{\varphi}_i$  and its projection on a subspace spanned by interpolation basis  $U = [\varphi_1, \dots, \varphi_{i-1}]$ . Algorithm 1 shows the complete algorithm details, whereby the notation “max” is such that  $[[\rho], \gamma_1] = \max\{|\tilde{\varphi}_1|\}$  implies position  $\gamma_1$  has the largest magnitude of  $\tilde{\varphi}_1$ , with a value of  $|\rho|$ .

#### 4.2 Path Planning and Vehicle Control

The sensor guidance and control aspect of the DDDAS is managed using a smoothed particle hydrodynamics (SPH) based control scheme to drive UAVs toward the hot spots. SPH has been previously used in many cooperative control applications [33–38] and results in a computationally efficient control scheme that gives fluid-like vehicle motions while providing obstacle and collision avoidance. The SPH equations are derived from the NS equations of fluid motion [39, 40] and

---

#### Algorithm 1 Construct the indices for hot spots

---

**Require:** Dominant modes of the plume concentration  $\{\varphi_i\}_{i=1}^d \subset \mathbb{R}^n$  and its projection to the accessible region  $\{\tilde{\varphi}_i\}_{i=1}^d \subset \mathbb{R}^n$ .

**Ensure:** Indices  $\{\gamma_i\}_{i=1}^d$  for the first  $d$  hot spots.

Select the first hot spot index:  $[[\rho], \gamma_1] = \max\{|\tilde{\varphi}_1|\}$ . Initialize  $U = [\varphi_1]$ .

**for**  $i = 2$  to  $d$  **do**

- 1: Solve the coefficient vector  $d_i$  for  $\tilde{\varphi}_i$ ,  $(E^T U)d_i = E^T \tilde{\varphi}_i$ .
- 2: Calculate the residual  $r_i = \tilde{\varphi}_i - U d_i$ .
- 3: Select the interpolation index corresponding to the largest magnitude of the residual  $[[\rho], \gamma_i] = \max\{|r_i|\}$ .
- 4: Add a new mode,  $U = [U \varphi_i]$ .

**end for**

---

provide an intuitive way of changing the sensor motion qualities by adjusting relevant fluid properties such as Reynolds number and Mach number. Since this technique is not new, only the most essential parts of the method are presented here.

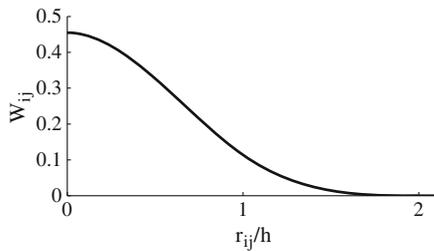
In the SPH representation, *Lagrangian* particles are used to represent the fluid; all fluid properties are applied through the use of a “smoothing kernel” centered at each particle location. The smoothing kernel used in this application is given by

$$W_{ij} = \frac{10}{7\pi h^2} \begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3 & \text{if } 0 \leq s \leq 1 \\ \frac{1}{4}(2-s)^3 & \text{if } 1 < s \leq 2 \\ 0 & \text{if } s > 2 \end{cases}, \quad (17)$$

where  $h$  is the smoothing width of the kernel,  $s = r_{ij}/h$ , and  $r_{ij}$  is the distance between particles  $i$  and  $j$ . The kernel is shown in Fig. 2. The NS equations for conservation of mass and momentum are then discretized as

$$\rho_i = \sum_j m_j W_{ij}, \quad (18)$$

$$\frac{D\mathbf{u}_i}{Dt} = - \sum_j \left[ m_j \left( \frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} \right] + \sum_j \left( \frac{\Pi_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}} \right) + \mathbf{F}_i, \quad (19)$$



**Fig. 2** The smoothing kernel used for the SPH computations

where subscripts denote particle identity,  $\nabla_i$  denotes the gradient with respect to particle  $i$ ,  $\rho$  is density,  $\mathbf{u}$  is velocity,  $m$  is mass,  $\mathbf{\Pi}$  is the viscous stress, and  $\mathbf{F}$  is an external forcing term. We slightly abuse the notion and denote by  $P$  the pressure, which is computed using the equation of state

$$P_i = K_i \rho_i \left( \frac{\rho_i}{\rho_0} - 1 \right). \tag{20}$$

All spatial derivatives that are normally present in the NS equations now manifest themselves as the  $\nabla W$  terms and the SPH equations form a system of ordinary differential equations that may be integrated in time.

To apply the SPH equations of motion to a cooperative control system, each vehicle in the system is treated as a single fluid particle. Since  $W$  is compactly supported, vehicle-vehicle interactions have a limited range and the SPH equations are very efficient to compute on resource constrained UAVs. The SPH equations provide a repulsive force between particles that come too close, providing for collision avoidance, and obstacles are treated using one or more virtual particles to ensure obstacle avoidance.

Just as fluid flows may be driven by external forces such as gravity, vehicle guidance is handled through external forcing terms. A loiter circle is often necessary for aerial vehicles that must maintain a minimum speed and in this application an external force is used to guide vehicles into a loiter circle at a desired location. A potential force  $\mathbf{F}_r = \nabla P$  is used to guide the vehicles to the loiter circle based on the potential

$$P = (r - R) \cdot \tanh \left( 4 \frac{r}{R} - 4 \right), \tag{21}$$

where  $R$  is the loiter circle radius and  $r$  is the distance from the loiter circle center. A circulation force is also added to guide the vehicles around the loiter circle. This circulation force has the form

$$\mathbf{F}_\theta = \left( \frac{r}{R} \right)^2 \exp \left( 2 - 2 \left( \frac{r}{R} \right)^2 \right) \mathbf{e}_\theta, \tag{22}$$

where  $\mathbf{e}_\theta$  is a unit vector in the circumferential direction. The total external force is then given by

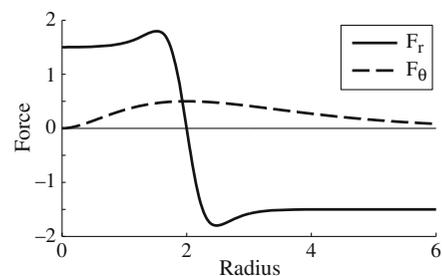
$$\mathbf{F} = C(\mathbf{F}_r + \mathbf{F}_\theta) \tag{23}$$

where  $C$  is a scaling coefficient to make the external force the same order as the SPH forces. This external force is shown in Fig. 3.

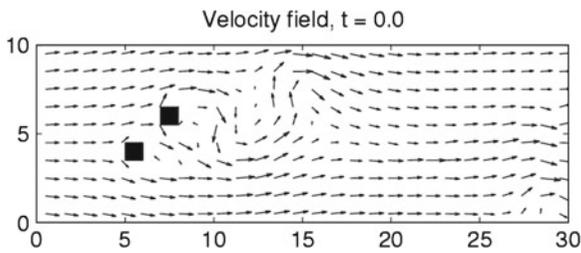
In simulation, the vehicle velocities are computed by applying velocity and acceleration constraints to the SPH equations and then adding on the effect of the background wind speed. For simplicity, no vehicle dynamics or inertial effects are considered in the contribution of the background wind speed.

### 5 Simulation

In this section, the proposed data assimilation and control strategy described above are shown to successfully monitor and characterize a simulated concentration plume that is analogous to the release of a plume of toxic chemicals in an urban environment. The plume motion is governed by the advection-diffusion equation given in Eq. 1. For simplicity, the initial plume concentration is zero. However, in this test, neither the precise location nor the extent of the plume source is



**Fig. 3** The external force that is used to drive vehicles to a loiter circle, in this case the loiter circle has radius  $R = 2$ .  $\mathbf{F}_r$  denotes the force in the radial direction and  $\mathbf{F}_\theta$  denotes the circumferential force



**Fig. 4** The background wind velocity field at  $t = 0$

known. As time progresses the DDDAS is able to identify and update the plume concentration by guiding the UAVs to measurement plume concentration near hot spots while assimilating the data measured along the sensor paths. Finally, the plume source is calculated.

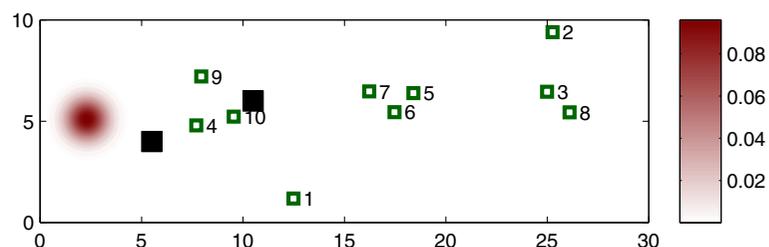
### 5.1 Simulation Setup

In the simulation, the wind velocity is computed by solving the non-dimensional NS equation on the spatial domain  $\Omega = [0, 30] \times [0, 10]$  (see Fig. 4), which is treated as the “real wind”. This is then assumed to be a known quantity. The plume concentration is governed by the advection-diffusion Eq. 1, which is solved on the same domain. A Reynolds number of 10,000 and a Péclet number of 5,000 were used to provide a reasonable velocity field with a mildly diffusive plume. The velocity boundary conditions are set as  $u = 1$ ,  $v = 0$  at the left and right boundaries with free-slip wall conditions at the top and bottom of the domain. Additionally, two square obstacles are placed within the domain to simulate buildings in an urban environment. The squares are of size 1 and are centered at (5.5, 4) and (7.5, 6). These obstacles were applied using no-slip boundary conditions. All concentration boundary conditions were set

to  $\psi = 0$  for simplicity. As the concentration field evolves, the plume interacts with the buildings in the domain, passing between and around them. The simulation is run from  $t = -40$  to  $t = 0$  with no concentration field in order to allow the velocity field to spin up. For time  $t > 0$ , a constant plume source is set as  $f(t, x) = 1.2 \exp(-\|x - x^*\|_2^2)$  where  $x^* = (2.3, 4.1)$  (see Fig. 5). This corresponds to a plume source centered at (2.3, 4.1). This will be used as the true plume solution and as a reference for all measures of accuracy. For our purposes, the physical accuracy of the simulation is not as important as having a reasonable plume data set to use. The total time domain of interest is  $[0, 30]$  from the release of plume source to the time for the front of the plume to reach the right edge of the spatial domain. This simulation uses a grid spacing of  $\Delta x = 0.2$  and a time step of  $\Delta t = 0.2$ .

The DDDAS described in previous sections is then applied to this data set. In the offline stage, the plume source is assumed to be unknown so it is initially estimated to be within the square region  $[1.5, 3] \times [3.5, 5]$ . A linear combination of kernel basis functions are used in Eq. 4 to approximate the real plume source. The basis functions used here are set as  $\phi_i(x) = 0.1 \exp(-\|x - x_i\|_2^2/0.8^2)$ . The centers  $x_i$  of these Gaussian kernel functions are located on the grid points at 2, 2.5 in horizontal direction and 4, 4.5 in vertical direction. Using these basis functions for plume source, we generate corresponding trajectories for the evolution of plume concentration and build a discrete data ensemble. The POD method is then used to capture the dominant modes of all the possible states for a given time domain. The contribution to energy is characterized by the corresponding singular values. If fixed sensors are used to measure the plume concentration, we take all the snapshots for the

**Fig. 5** (Color online) The plume source and randomly chosen positions for fixed sensors

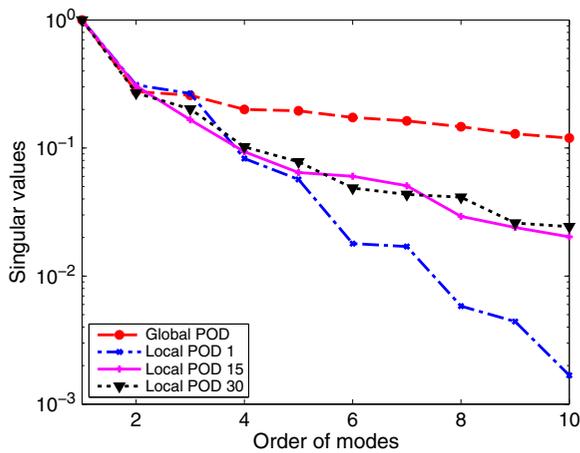


global time domain. However, if UAVs are used, we can artificially partition the global time domain into 30 periods, and focus on the local POD modes for each time interval. Thus, each interval lasts for 1 dimensionless time. It is noticed that if the estimated or predicted plume source is local in space, then only the overlapped plume sources in the precomputed database could contribute useful snapshots for the local POD modes. For simplicity, we simply assume all the plume sources in the database overlap the “real” plume source, and use all the plume sources in the database to form local POD modes. Figure 6 shows the singular values for the first 10 global POD modes as well as the singular values for the first 10 local POD modes for subinterval 1, 15, and 30. All the singular values are normalized such that for a same data ensemble, the maximal singular value is 1. Singular values for local POD modes drops much faster than global modes. From the 4th local POD modes, the values are less than 0.1. Based on Eq. 8, the truncation error from the three modes are less than 0.1 in terms of the fractional energy. Since the real plume concentration does not reside precisely on the subspace spanned by this data ensemble, only the first three local POD

modes are used to approximate the real plume concentration. The number of modes used may be altered to achieve a different balance between accuracy and computational speed. However, in the rest of this section, we only use three local POD modes in the data assimilation process for all the measurement schemes. Since there are only three unknown coefficients for the corresponding modes, solving the inverse problem in Eq. 14 is very efficient.

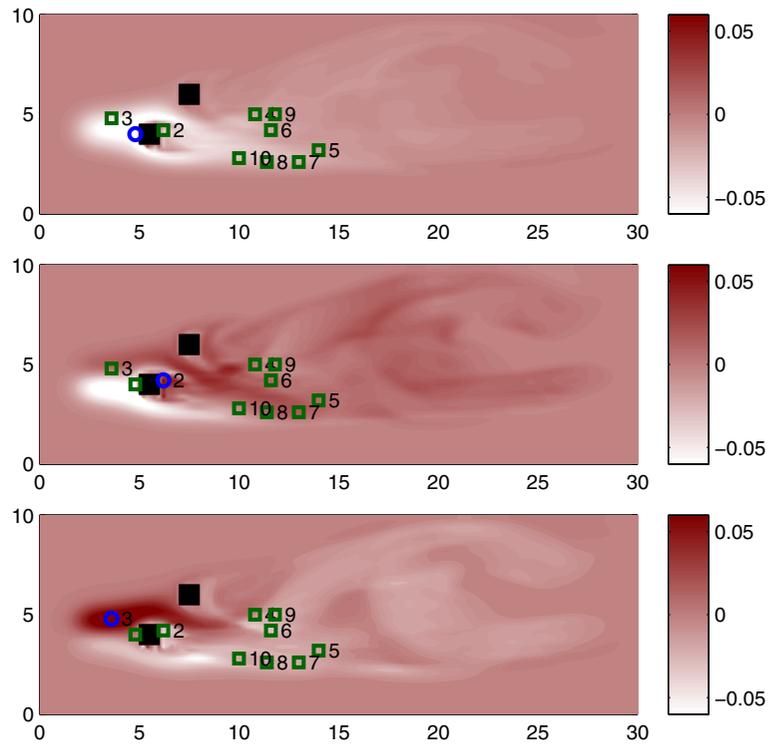
### 5.2 Data Assimilation and Error Analysis

In order to test the proposed data assimilation technique and the sensor place strategy, we focus on the effect of varying two parameters in the DDDAS simulation: the number of UAVs and the type of control scheme. There are four types of control scheme. The first scheme uses randomly placed fixed sensors. The sensor locations are chosen from a uniform distribution on the simulation domain (excluding the building locations). Figure 5 shows one example sensor distribution generated by the pseudo-random numbers. This scheme is certainly not optimal and is used for comparison with other schemes. The second scheme is an optimal scheme for fixed sensors, where sensors are exactly located at hot spots chosen based on the global POD modes (see Fig. 7). The hot spots are calculated based on the greedy algorithm discuss in Section 4.1. The third scheme is an ideal (but unrealistic) case. It is assumed that UAVs are able to be placed precisely at the different hot spots at different time intervals. Due to the limitations of the UAVs, primarily the limited vehicle speed and uncertainty in the trajectories, the sensor placement routine considers only six time intervals with time period  $\Delta t = 5$  to limit the number of hot spots that the sensors must attempt to reach. Figure 8 shows the first 10 hot spots for the interval [25, 30] as well as the first 3 corresponding local POD modes for the same interval. Different from fixed sensors, UAVs cannot be located very close to buildings for safety reasons, therefore, all the hot spots are excluded from circular regions around buildings. Figure 9 shows the trajectories of the hot spots for the first 3 modes. In the last scheme, the UAVs are guided toward the hot spots (in scheme 3) using

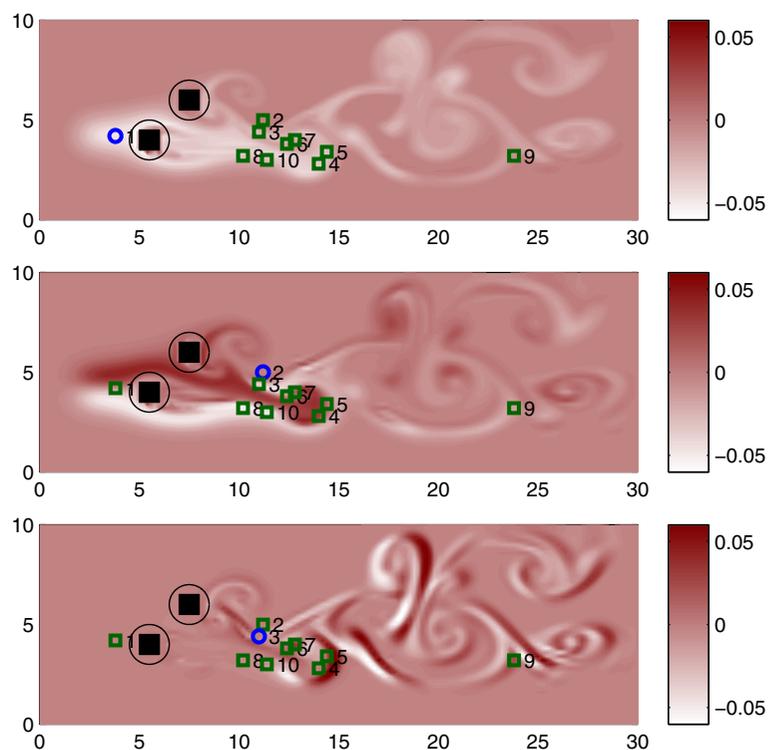


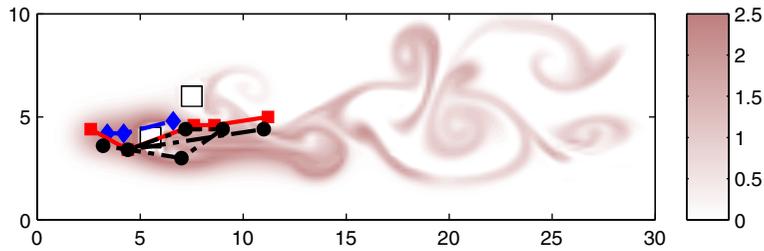
**Fig. 6** (Color online) The normalized singular values. Global POD modes are computed from SVD for the all the snapshots in the time domain [0, 30]. “Local POD 1” refers to the local POD modes for the first time interval [0, 1], “Local POD 15” corresponds to the 15th time interval [14, 15], and “Local POD 30” corresponds to the last time interval [29, 30]

**Fig. 7** (Color online) The hot spots corresponding to the first 10 global POD modes (for time interval  $[0, 30]$ ) with the profile of the first global POD mode at *top*, with the second one in the *middle*, and with the third one at *bottom*



**Fig. 8** (Color online) For the last time interval  $[25, 30]$ , the hot spots corresponding to the first 10 local POD modes with the profile of the first local POD mode at *top*, with the second one in the *middle*, and with the third one at *bottom*. Hot spots are chosen outside the circular regions around buildings





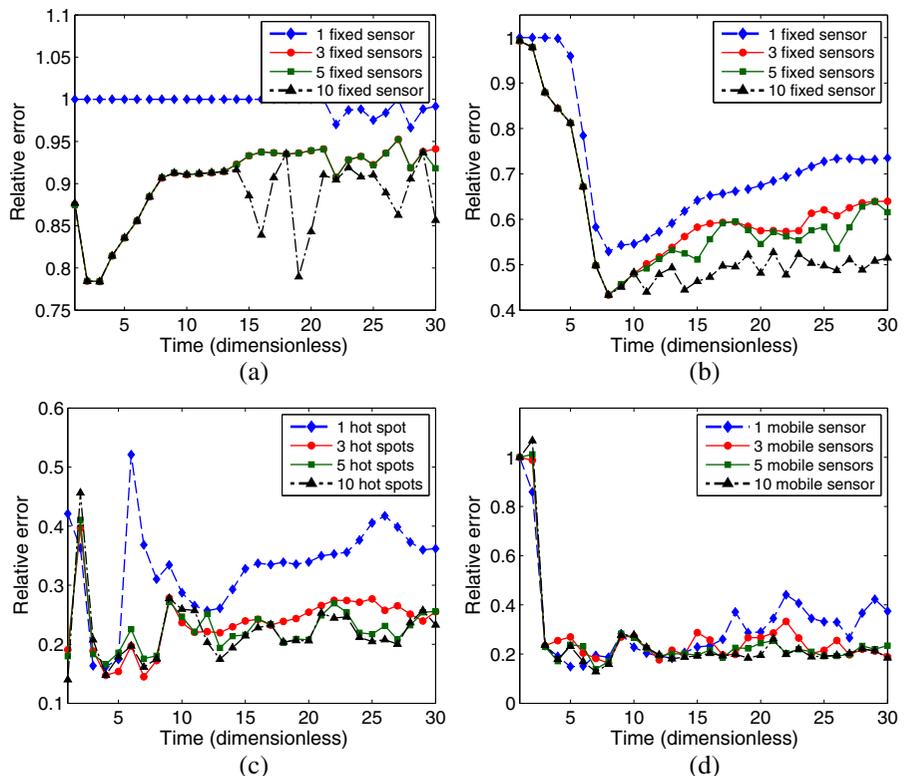
**Fig. 9** (Color online) The trajectories for the hot spots corresponding to the first three local POD modes. The first hot spot (blue) stays near the plume source, the second one (red) moves toward the right region as plume is advected

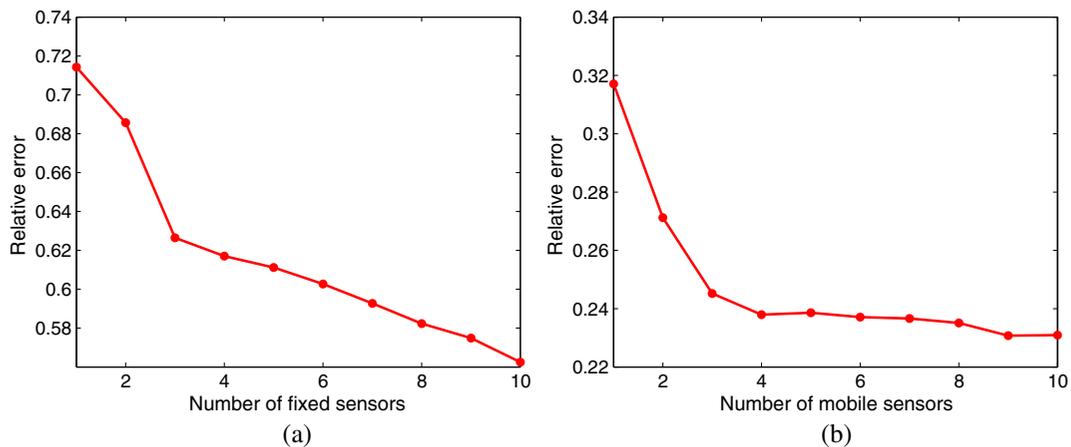
by wind, the third one (green) is around the building area to minimize the residual error. The plume profile shows the instantaneous plume concentration of the “real plume” for  $t = 30$

the SPH method described in Section 4.2. Each virtual particle is set at the corresponding hot spot to attract the real vehicle. In order to maintain minimal velocity and obtain a wide coverage for measurement, each vehicle moves in a loiter circle once it nears the desired hot spot. The sensors begin near the origin (0, 0) and travel at a dimensionless speed of about four to five (compared to the unit incoming wind speed) toward the hot spots for the first time interval to begin data collection.

For direct comparison, all these control schemes use the same procedure for data assimilation. Specifically, each sensor measures the true concentration value at its position and data assimilation is performed using these values at a frequency of 1 dimensionless time to estimate the coefficients for local POD modes corresponding to this time interval. To increase realism, Gaussian noise with standard deviation of 0.2 was added to the data before assimilation.

**Fig. 10** (Color online) The magnitude of relative error, defined as  $E(t) := \|\hat{\psi}(t) - \psi(t)\| / \|\psi(t)\|$ , of plume concentration with time evolution. **a** Sensors are located at fixed positions randomly chosen. **b** Sensors are located at fixed positions according to global POD modes. **c** Sensors are located at the hot spots corresponding to local POD modes, which update every  $\Delta T = 5$ . **d** UAVs are guided towards to the first few hot spots corresponding to local POD modes by SPH





**Fig. 11** (Color online) The average error of plume concentration, given by  $\bar{E} := \int_0^T E(t)/T$ , with the number of UAVs for **a** static sensors at the hot spots corresponding

to global POD modes (range from 0.56 to 0.72), **b** UAVs governed by the SPH model (range from 0.22 to 0.32). Note that the y-axis has a different range in each plot

Since the maximum concentration is about 2.5, the relative noise of the measurement is more than 8 %.

In Fig. 10, our metric for comparison is the relative error in the simulated concentration plume as a function of time for the above mentioned procedure using the four different control schemes. This error is computed using the  $\ell^2$  norm. For scheme 1, the relative errors are near 1 in Fig. 10a, which means randomly distributed static sensors rarely obtain much useful information. Adding more static sensors beyond three does not significantly improve the results. This can be explained by the fact that for the sensor distribution in Fig. 5, the second and the third sensors are located at more informative positions while the other sensors are not. The second scheme, as shown in Fig. 10b, still shows a large relative error in the range of about 0.4–0.8 for most of the time. In a global sense, all the static sensors are at the optimal position. But since they lack flexibility to move over time as the plume evolves, even 10 static sensors are still not enough to obtain a good estimation. For scheme 3, when sensors are located at the hot spots for each time interval, the error drops significantly, as seen in Fig. 10c. When three or more sensors are used, the error drops below 0.3 for most of the time domain. For the last scheme, as seen in Fig. 10d, the sensors require a small amount of time to first reach the plume and start collecting useful data. Once the UAVs get close to the hot spots,

the  $\ell^2$  error drops significantly. For one vehicle (blue line), the error tends to increase with time evolution. This is because the plume is advected by the wind while the first hot spot is very near to the plume source, and the vehicle does not measure the using additional UAVs which measure near the hotspots in the downwind direction as well.

Lastly, we examine the importance of the number of UAVs used in the DDDAS. In Fig. 11, as expected, the multi-vehicle data collection scenarios outperform the single vehicle scenarios for both (a) static sensors at optimal positions and (b) UAVs controlled by SPH. On the other hand, performing data assimilation with the static sensors decreases the error, but not nearly as much as when a DDDAS is used with UAVs. In fact, a single UAV outperforms the ten static sensors in this scenario in terms of the mean  $\ell^2$  error over the full time interval. In particular, Fig. 11b shows that three UAVs can decrease the average error below 0.25.

## 6 Conclusions

In this article, we propose a complete dynamic data-driven application system (DDDAS) for measuring and simulating a concentration plume and estimating the plume source in a dynamic environment. The dynamic evolution of plume

concentration is governed by the advection-diffusion equation with unknown plume source. In order to assimilate incomplete and noisy state observations into this system in real-time, an offline-online approach is used. In the offline stage, we build a database by sampling several possible plume source regions and solving the governing equation to obtain possible trajectories for the plume concentration. We then use the proper orthogonal decomposition (POD) to find a few dominant modes. In the online stage, estimations of plume concentration are restricted to the subspace of the POD modes to best match partial observations in a least-squares sense. The proposed data assimilation method also leads to a hierarchical vehicle control strategy. On one level, hot spots are chosen such that maximal information can be obtained by UAVs. On another level, smoothed particle hydrodynamics (SPH) is used for vehicle path planning and control. Both the assimilation method and the control strategies are very computationally efficient and can be carried out in real time. The simulation results verify the utility of the proposed DDDAS. Specifically, three UAVs are able to approach three hot spot locations and produce a greatly improved estimate of plume concentration when compared to the use of static sensors. The relative  $\ell^2$  error for three UAVs reaches about 0.25 in the simulation, which outperforms 10 randomly placed static sensors.

## References

- Corrigan, C.E., Roberts, G.C., Ramana, M.V., Kim, D., Ramanathan, V.: Capturing vertical profiles of aerosols and black carbon over the Indian Ocean using autonomous unmanned aerial vehicles. *Atmos. Chem. Phys. Discuss.* **7**, 11429–11463 (2007)
- Spiess, T., Bange, J., Buschmann, M., Vörsmann, P.: First application of the meteorological mini UAV ‘M2AV’. *Meteorol. Z.* **16**(2), 159–169 (2007)
- van den Kroonenberg, A., Martin, S., Beyrich, F., Bange, J.: Spatially-averaged temperature structure parameter over a heterogeneous surface measured by an unmanned aerial vehicle. *Bound. Layer Meteorol.* **142**, 55–77 (2011)
- Hardin, P., Jensen, R.: Small-scale unmanned aerial vehicles in environmental remote sensing: challenges and opportunities. *GISci. Remote Sens.* **48**(1), 99–111 (2011)
- Allred, J., Hasan, A., Pisano, B., Panichsakul, S., Gray, P., Han, R., Lawrence, D., Mohseni, K.: SensorFlock: a mobile system of networked micro-air vehicles. In: *The ACM SenSys 2007: The 5th ACM Conference on Embedded Networked Sensor Systems*, Sydney, Australia (2007)
- Leven, S., Zufferey, J.-C., Floreano, D.: Dealing with mid-air collisions in dense collective aerial systems. *J. Field Robot.* **28**(3), 405–423 (2011)
- Darema, F.: Dynamic data driven applications systems: a new paradigm for application simulations and measurements. In: *Int. Conf. Comput. Sci. (ICCS)*, vol. 3038, pp. 662–669. Kraków, Poland (2004)
- Mandel, J., Bennethum, L., Beezley, J., Coen, J., Douglas, C., Kim, M., Vodacek, A.: A wildland fire model with data assimilation. *Math. Comput. Simul.* **79**, 584–606 (2008)
- Rodriguez, R., Cortés, A., Margalef, T.: Injecting dynamic real-time data into a DDDAS for forest fire behavior prediction. In: *Int. Conf. Comput. Sci. (ICCS)*, vol. 5454, pp. 489–499. Baton Rouge, LA (2009)
- Akcelik, V., Biros, G., Drăgănescu, A., Ghattas, O., Hill, J., van Bloeman Waanders, B.: Dynamic data-driven inversion for terascale simulations: real-time identification of airborne contaminants. In: *Proceedings of SC2005*, pp. 43–58. Seattle, WA (2005)
- Lieberman, C., Fidkowski, K.W., van Bloemen Waanders, B.: Hessian-based model reduction: large-scale inversion and prediction. *Int. J. Numer. Methods Fluids* **71**, 135–150 (2013)
- Akcelik, V., Biros, G., Drăgănescu, A., Ghattas, O., Hill, J.: Inversion of airborne contaminants in a regional model. In: *Int. Conf. Comput. Sci. (ICCS)*, vol. 3993, pp. 481–488. Reading, UK (2006)
- Fisher, M., Nocedal, J., Trémolet, Y., Wright, S.J.: Data assimilation in weather forecasting: a case study in PDE-constrained optimization. *Optim. Eng.* **10**(3), 409–426 (2009)
- Holmes, P., Lumley, J., Berkooz, G.: *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Univ. Press, Cambridge, UK (1998)
- Stam, J.: Stable fluids. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 121–128. Los Angeles, CA (1999)
- Antsaklis, P.J., Passino, K.M.: Towards intelligent autonomous control system: architecture and fundamental issues. *J. Intell. Robot. Syst.* **1**, 315–342 (1989)
- Loève, M.: *Probability Theory*. Van Nostrand, NY (1955)
- Cortial, J., Farhat, C., Guibas, L.J., Rajashekhar, M.: Compressed sensing and time-parallel reduced-order modeling for structural health monitoring using a DDDAS. In: *Int. Conf. Comput. Sci. (ICCS)*, vol. 4487, pp. 1171–1179. Beijing, China (2007)
- Rowley, C.W., Colonius, T., Murra, R.M.: Model reduction for compressible flows using POD and Galerkin projection. *Phys. D.* **189**, 115–129 (2004)
- Parrilo, P., Paganini, F., Verghese, G., Lesieutre, B., Marsden, J.E.: *Model Reduction for Analysis of*

- Cascading Failures in Power Systems, pp. 4208–4212. American Control Conference, San Diego, CA (1999)
21. Rathinam, M., Petzold, L.R.: Dynamic iteration using reduced order models: a method for simulation of large scale modular systems. *SIAM J. Numer. Anal.* **40**, 1446–1474 (2002)
  22. Amabili, M., Sarkar, A., Paidoussis, M.P.: Reduced-order models for nonlinear vibrations of cylindrical shells via the proper orthogonal decomposition method. *J. Fluids Struct.* **18**, 227–250 (2003)
  23. Graham, M., Kevrekidis, I.: Alternative approaches to the Karhunen-Loève decomposition for model reduction and data analysis. *Comput. Chem. Eng.* **20**, 495–506 (1996)
  24. Shvartsman, S., Kevrekidis, I.: Low-dimensional approximation and control of periodic solutions in spatially extended systems. *Phys. Rev. E* **58**, 361–368 (1998)
  25. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An ‘Empirical Interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Ser. I* **339**, 667–672 (2004)
  26. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **32**, 2737–2764 (2010)
  27. Hsu, P.-M., Lin, C.-L., Yang, M.-Y.: On the complete coverage path planning for mobile robots. *J. Intell. Robot. Syst.* 1–19 (2013). doi:10.1007/s10846-013-9856-0
  28. Lazanas, A., Latombe, J.C.: Motion planning with uncertainty: a landmark approach. *Artif. Intell.* **76**, 287–317 (1995)
  29. Rao, N.: Robot navigation in unknown generalized polygonal terrains using vision sensors. *IEEE Trans. Syst. Man Cybern.* **25**(6), 947–962 (1995)
  30. Peng, L., Zhao, Y., Tian, B., Zhang, J., Wang, B.-H., Zhang, H.-T., Zhou, T.: Consensus of self-driven agents with avoidance of collisions. *Phys. Rev. E* **79**, 026113 (2009)
  31. Zhang, G., Ferrari, S., Qian, M.: An information roadmap method for robotic sensor path planning. *J. Intell. Robot. Syst.* **56**(1–2), 69–98 (2009)
  32. Krause, A., Singh, A., Guestrin, C.: Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.* **9**, 235–284 (2008)
  33. Pimenta, L., Mendes, M., Mesquita, R., Pereira, G.: Fluids in electrostatic fields: an analogy for multirobot control. *IEEE Trans. Magn.* **43**(4), 1765–1768 (2007)
  34. Pimenta, L., Michael, N., Mesquita, R., Pereira, G., Kumar, V.: Control of swarms based on hydrodynamic models. In: *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 1948–1953. Pasadena, CA (2008)
  35. Huhn, S., Mohseni, K.: Cooperative control of a team of AUVs using smoothed particle hydrodynamics with restricted communication. In: *Proceedings of the ASME 28th International Conference on Ocean, Offshore and Arctic Engineering*, vol. OMAE 2009–79869, pp. 531–538. Honolulu, HA (2009)
  36. Lipinski, D., Mohseni, K.: Cooperative control of a team of unmanned vehicles using smoothed particle hydrodynamics. *AIAA Paper 2010–8316*, AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, Canada (2010)
  37. Shaw, A., Mohseni, K.: A fluid dynamic based coordination of a wireless sensor network of unmanned aerial vehicles: 3-D simulation and wireless communication characterization. *IEEE Sensors J., Special Issue on Cognitive Sensor Networks* **11**(3), 722–736 (2011)
  38. Lipinski, D., Mohseni, K.: A master-slave fluid cooperative control algorithm for optimal trajectory planning. In: *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 3347–3351. Shanghai, China (2011)
  39. Monaghan, J.: Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.* **30**, 543–574 (1992)
  40. Liu, G., Liu, M.: *Smoothed Particle Hydrodynamics: A Meshfree Particel Method*. World Scientific Publishing Company, Hackensack, NJ (2003)