2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Estimation and Adaptive Observation of Environmental Plumes

David Zhang*, Christopher Colburn, and Thomas Bewley

Abstract— **The estimation and forecasting environmental plume movement based on information from mobile sensors recently received renewed attention due to the Gulf coast oil and Icelandic ash problems, and remains of sustained interest today in homeland security settings (plant explosions, dirty bombs, etc.). The present work refines and tests the scientific algorithms at the heart of this problem. In particular, we combine the Ensemble Kalman Filter (EnKF), which provides a computationally feasible low-rank approximation of the uncertainty of the estimate, with our recently developed Dynamic Adaptive Observation (DAO) algorithm for optimizing feasible sensor vehicle trajectories that minimize forecast uncertainty. A numerical experiment is performed which applies this combined EnKF/DAO algorithm to determine waypoints along optimized feasible sensor vehicle trajectories that improve the forecast of an environmental plume represented by a passive scalar convectively driven in a 2D fluid flow.**

I. INTRODUCTION

Adaptive Observation (AO) is the problem of planning optimized trajectories of sensor-equipped vehicles for the purpose of minimizing forecast uncertainty. This class of problems is thus something that is a fusion of more pure "control" and "estimation" problems, as it contains important elements of both. Proposed methods so solve problems of this class are either distributed or centralized in nature.

In distributed AO strategies, such as those proposed in [1], [2], [3], and [4], each mobile sensor has little knowledge of the overall sensed system, and sensor deployment is planned locally. The hope is that simple local rules might lead to vehicle motions that distribute the sensors effectively. This is achieved by distributing sensors essentially uniformly, perhaps clustering sensors in areas of particular interest. Their inherent simplicity make such strategies easy to implement.

As the size of the domain under consideration increases with respect to the field of view of the sensors, the effectiveness of distributed AO algorithms diminishes due to the sparseness of the sensor coverage attained. Further, distributed AO algorithms essentially neglect the dynamics of the evolution of the system under consideration, which further reduces forecast accuracy. In such problems, it is beneficial to plan the sensor movement more deliberately with a centralized AO algorithm, as suggested by [5], [6], and [7]. Such algorithms account for the current uncertainty distribution and the underlying system model in order to optimized sensor positions or trajectories that maximally reduce the forecast uncertainty. Due to the model complexity, centralized AO algorithms are computationally intensive, and thus cannot be solved locally on the individual vehicles. Rather, the necessary computations are performed centrally (e.g., on a supercomputer cluster), and optimized sensor waypoints are periodically broadcast back to the sensor vehicles. We restrict our focus in this paper to centralized AO strategies applied to fluid-mechanical systems.

The various models to which centralized AO algorithms have been applied in the past have been assumed to evolve slowly as compared with the time scales of motion of the sensor vehicles themselves; thus, previous centralized AO algorithms have not focused on respecting the practical constraints on the vehicle dynamics themselves. In the formulation considered here, however, this assumption is relaxed; that is, the time scales of the fluid flow and the time scales of the vehicle dynamics are taken to be comparable. Towards this end, we developed a new AO algorithm, dubbed the Dynamic Adaptive Observation (DAO), which incorporates vehicle dynamics while minimizing the forecast uncertainty.

In this paper, we formulate the DAO algorithm and demonstrate its capabilities in conjunction with the Ensemble Kalman Filter (EnKF) on a relevant 2D convection-driven environmental plume problem. Section II presents the DAO formulation, and §III reviews the EnKF. In §IV both DAO and EnKF are combined in a numerical experiment which estimates an environmental plume.

II. DAO: DYNAMIC ADAPTIVE OBSERVATION

The state vector $\mathbf{x}(t) \in \mathbb{R}^n$ is assumed to be governed by a nonlinear ODE obtained via spatial discretization of a PDE over a physical domain Ω of interest:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{d}(t), \mathbf{w}(t)), \tag{1}$$

where $\mathbf{w}(t) \sim N(0, W)$ models the (unknown) external disturbances and $\mathbf{d}(t)$ models is the (known) external forcing. Assume there are M sensor vehicles moving within the domain Ω; the continuous-time dynamics of the ith vehicle, with state $\mathbf{q}^i(t) \in \mathbb{R}^m$ and control $\mathbf{u}^i(t)$, is taken to be

$$\frac{d\mathbf{q}^i(t)}{dt} = g(\mathbf{q}^i(t), \mathbf{u}^i(t)). \tag{2}$$

Each vehicle moves about the domain Ω *continuously*, while taking measurements $\mathbf{y}_k$ at *discrete* times t_k:

$$\mathbf{y}_k^i = h_k^i(\mathbf{x}_k, \mathbf{q}_k^i) + \mathbf{v}_k^i, \tag{3}$$

where $\mathbf{v}_k^i \sim N(0, R_k^i(\mathbf{q}^i))$; note that vehicle states such as position, heading, and velocity affect both the measurement operator h as well as the statistics of the measurement noise. (For convenience, dependence on the vehicle state $\mathbf{q}$ is notationally suppressed in the analysis that follows.) The collection of measurements from all vehicles is given by

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_k^1 \\ \vdots \\ \mathbf{y}_k^M \end{bmatrix}, h_k = \begin{bmatrix} h_k^1(\mathbf{x}_k) \\ \vdots \\ h_k^M(\mathbf{x}_k) \end{bmatrix}, R_k = \begin{bmatrix} R_k^1 & & 0 \\ & \ddots & \\ 0 & & R_k^M \end{bmatrix}. \tag{4}$$

All authors are with the Flow Control Lab, Dept of MAE, UC San Diego, La Jolla, CA 92093. The authors gratefully acknowledge the support of the National Security Education Center at Los Alamos National Laboratory.

*Corresponding author: dazhang@ucsd.edu

978-1-4577-0079-8/11/$26.00 ©2011 AACC

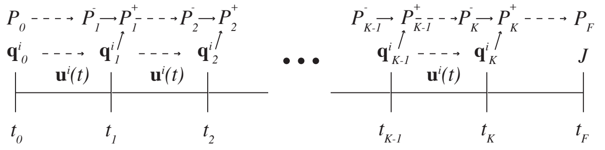

Fig. 1. Cartoon illustrating the problem formulation. $\mathbf{u}^i(t)$ affects the *continuous-time* evolution of sensor vehicle trajectories $\mathbf{q}^i(t)$. In turn, the sensor vehicle positions at the measurement times, $\mathbf{q}^i_k$, affect the *discrete-time* update of the estimation error covariance P_k; this covariance otherwise evolves continuously between the measurements, and between t_K and t_F. The cost J depends on P_F and $\mathbf{u}^i(t)$ within time window $[t_0, t_K]$; a set of controls $\mathbf{u}^i(t)$ is sought to minimize this cost. Dashed arrows denote continuous-time propagations; solid arrows denote discrete-time updates.

For state estimation, we use a mixed Kalman Filter (KF) formulation with a continuous-time state evolution between measurements, together with discrete-time measurement updates. Denote the state estimate $\hat{\mathbf{x}}$ and estimation error covariance $P \triangleq \mathcal{E}\{\delta\mathbf{x}\,\delta\mathbf{x}^T\}$, where $\delta\mathbf{x} \triangleq \mathbf{x} - \hat{\mathbf{x}}$. The continuous-time propogation of P between measurements is given by

$$\frac{dP(t)}{dt} = AP(t) + P(t)A^T + BWB^T, \tag{5}$$

where A and B are given by (1) linearized about $\hat{\mathbf{x}}(t)$. The discrete-time update of P at each measurement is given by

$$P_k^+ = (I - L_k H_k)P_k^-, \quad L_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \tag{6}$$

where $()^-$ and $()^+$ denote before the update and after the update, respectively, H_k is given by (4) linearized about $\hat{\mathbf{x}}_k$, and L_k is the optimal KF feedback gain. Fig. 1 illustrates this mixed continuous/discrete time formulation, and the relationships between the several quantities involved. Note that, as $P(t)$ is updated during each measurement, the trajectory $P(t)$ is *piecewise smooth.*

The problem DAO is formulated to solve is framed as follows: *At time t_0, the initial vehicle states $\mathbf{q}^i_0$ and estimation error covariance P_0 are known. Design a set of control trajectories $\mathbf{u}^i(t)$ for each vehicle over the time window $[t_0, t_K]$ to minimize a cost function balancing control effort and forecast accuracy at the final time t_F, where $t_F \geq t_K$, conditioned on the measurements taken by the vehicles at times $\{t_1, t_2, \cdots, t_K\}$.*

For simplicity, we choose here a quadratic measure for control effort, and a weighted sum of the variances to quantify forecast accuracy:

$$J = \mathsf{trace}(TP_F) + \frac{1}{2}\sum_{i=1}^{M}\int_0^{t_K} \mathbf{u}^i(t)^T Q_{\mathbf{u}} \mathbf{u}^i(t)\,dt, \tag{7}$$

where T is a diagonal weighting matrix that biases the measure of P_F more heavily in certain "regions of interest" in the domain Ω, and $Q_{\mathbf{u}}$ is a positive definite weighting matrix. Not that, since $\mathbf{u}^i(t)$ affects the cost function nonlinearly, J is in general non-convex; global optimization of this cost function can thus not be guaranteed with a computationally tractable algorithm. Thus, a locally minimizing solution is sought via an iterative approach by initially assuming a nominal control trajectory for each vehicle, then computing a local gradient $\nabla_{\mathbf{u}^i(t)} J$. The following shows how such gradient can be tractably computed using *adjoint analysis*.

Applying perturbations to an assumed set of control trajectories causes a chain reaction that perturbs other variables; the first-order perturbations of these variables are:

$$\frac{d\mathbf{q}^i(t)'}{dt} = F\mathbf{q}^i(t)' + G\mathbf{u}^i(t)', \qquad \mathbf{q}^i_0{}' = 0, \tag{8a}$$

$$\frac{dP(t)'}{dt} = AP(t)' + P(t)'A^T, \qquad P_0' = 0, \tag{8b}$$

$$\begin{aligned} P_k^{+\prime} = {} & P_k^{-\prime} - (P_k^{-\prime} H_k^T + P_k^- (H_k')^T) L_k^T - L_k (H_k P_k^{-\prime} + H_k' P_k^-) \\ & + L_k (H_k' P_k^- H_k^T + H_k P_k^{-\prime} H_k^T + H_k P_k^- (H_k')^T + R_k') L_k^T, \end{aligned} \tag{8c}$$

$$J' = \mathsf{trace}(TP_F') + \sum_{i=1}^{M}\int_0^{t_K} \mathbf{u}^i(t)^T Q_u \mathbf{u}^i(t)'\,dt, \tag{8d}$$

$$R_k' = \begin{bmatrix} R_k^1{}' & & 0 \\ & \ddots & \\ 0 & & R_k^M{}' \end{bmatrix}, \quad H_k' = \begin{bmatrix} H_k^1{}' \\ \vdots \\ H_k^M{}' \end{bmatrix}, \tag{8e}$$

$$R_k^i{}' = \left(\frac{dR_k^i}{d\mathbf{q}_k^i}\right)^T \mathbf{q}_k^i{}', \quad H_k^i{}' = \left(\frac{dH_k^i}{d\mathbf{q}_k^i}\right)^T \mathbf{q}_k^i{}', \tag{8f}$$

where F and G are (2) linearized about $\mathbf{q}^i(t)$ and $\mathbf{u}^i(t)$ respectively, and $\mathbf{q}^i_0{}'$, P_0', and W' are zero because they are not affected by perturbations in $\mathbf{u}^i(t)$. Note that $\frac{dR_k^i}{d\mathbf{q}_k^i}$ and $\frac{dH_k^i}{d\mathbf{q}_k^i}$ are rank-3 tensors that contract by the inner product with $\mathbf{q}_k^i{}'$ to yield matrices $R_k^i{}'$ and $H_k^i{}'$. The purpose of the analysis below is to reëxpress (8d) in the form

$$J' = \sum_{i=1}^{M}\int_0^{t_K^-} \left(\nabla_{\mathbf{u}^i(t)} J\right)^T \mathbf{u}^i(t)'\,dt. \tag{9}$$

Equation (9) is similar to (8d) except for the $\mathsf{trace}(TP_F')$ term; the remainder of the analysis focuses on rewriting $\mathsf{trace}(TP_F')$ in this form.

We simplify the (8a) and (8b) description by introducing linear operators $\mathcal{L}(P')$, $\mathcal{M}(\mathbf{q}')^i$, and $\mathcal{B}(\mathbf{u}')^i$:

$$\mathcal{L}(P') \triangleq \frac{dP(t)'}{dt} - AP(t)' - P(t)'A^T, \tag{10a}$$

$$\mathcal{M}(\mathbf{q}')^i \triangleq \frac{d\mathbf{q}^i(t)'}{dt} - F\mathbf{q}^i(t)', \tag{10b}$$

$$\mathcal{B}(\mathbf{u}')^i \triangleq G\mathbf{u}^i{}', \tag{10c}$$

so that $\mathcal{L}(P') = 0$ by (8b) and $\mathcal{M}(\mathbf{q}')^i = \mathcal{B}(\mathbf{u}')^i$ by (8a). An adjoint variable $S(t) \in \mathbb{R}^{n\times n}$ is defined over the time window $[t_K^+, t_F]$ and an adjoint identity based on a relevant inner product is defined:

$$\langle S, \mathcal{L}(P')\rangle_{t_K^+, t_F} = \langle \mathcal{L}^*(S), P'\rangle_{t_K^+, t_F} + a, \tag{11a}$$

$$\langle X, Y\rangle_{t_K^+, t_F} \triangleq \int_{t_K^+}^{t_F} \mathsf{trace}(X(t)^T Y(t))\,dt. \tag{11b}$$

Using integration by parts, it can be shown that

$$\mathcal{L}^*(S) = -\frac{dS(t)}{dt} - A^T S(t) - S(t)A, \tag{12a}$$

$$a = \mathsf{trace}(S_F^T P_F') - \mathsf{trace}((S_K^+)^T P_K^+{}'). \tag{12b}$$

Taking $\mathcal{L}^*(S) = 0$ and $S_F = T$, (8d) is rewritten using relationships established in (11) and (12) into

$$J' = \mathsf{trace}((S_K^+)^T P_K^+{}') + \sum_{i=1}^{M} \int_0^{t_K^-} \mathbf{u}^i(t)^T Q_u \mathbf{u}^i(t)' \, dt. \tag{13}$$

Note by setting $\mathcal{L}^*(S) = 0$ and $S_F^T = T$, this is equivalent to defining a backward-in-time evolution equation for $S(t)$ such that $\frac{dS(t)}{dt} = -A^T S(t) - S(t)A$ with starting condition $S_F^T = T$. By the special structure of the S evolution equation and S_F, it is clear that S_K^+ is also symmetric in (13).

Substituting (8c), (8e), and (8f) into $P_K^+{}'$ in (13) and leveraging the trace identity $\mathsf{trace}(AB) = \mathsf{trace}(BA) = \mathsf{trace}(A^T B^T)$, the $P_K^-{}'$ and $\mathbf{q}_K^i{}'$ terms are gathered to the right. Leveraging the block matrix structure of H_K' and R_K', (13) becomes

$$\begin{aligned} J' &= \mathsf{trace}((I - H_K^T L_K^T) S_K^+ (I - L_K H_K) P_K^-{}') \\ &\quad + \mathsf{trace}(2P_K^- (H_K^T L_K^T - I) S_K^+ L_K H_K') \\ &\quad + \mathsf{trace}(L_K^T S_K^+ L_K R_K') + \sum_{i=1}^{M} \int_0^{t_K^-} \mathbf{u}^i(t)^T Q_u \mathbf{u}^i(t)' \, dt, \\ &= \mathsf{trace}((I - H_K^T L_K^T) S_K^+ (I - L_K H_K) P_K^-{}') \\ &\quad + \sum_{i=1}^{M} \mathsf{trace}((2P_K^- (H_K^T L_K^T - I) S_K^+ L_K)_i \left(\frac{dH_K^i}{d\mathbf{q}_K^i}\right))^T \mathbf{q}_K^i{}' \\ &\quad + \sum_{i=1}^{M} \mathsf{trace}((L_K^T S_K^+ L_K)_{ii} \left(\frac{dR_K^i}{d\mathbf{q}_K^i}\right))^T \mathbf{q}_K^i{}' \\ &\quad + \sum_{i=1}^{M} \int_0^{t_K^-} \mathbf{u}^i(t)^T Q_u \mathbf{u}^i(t)' \, dt, \end{aligned} \tag{14}$$

where $(L_k^T S_k^+ L_k)_{ii}$ denotes the (i,i) block of $M \times M$ block matrix $L_k^T S_k^+ L_k$ and $(2AP_k^- A^T (H_k^T L_k^T - I) S_k^+ L_k)_i$ denotes the i'th column block of $1 \times M$ block matrix $2AP_k^- A^T (H_k^T L_k^T - I) S_k^+ L_k$. Note the $()^-$ superscript on $\mathbf{q}_K^i{}'$ is dropped because the $\mathbf{q}^i(t)$ trajectory is smooth.

If the same inner product and adjoint identity as in (11) are defined, but now over the time window $[t_{K-1}^+, t_K^-]$, and if in addition M adjoint vectors $\mathbf{r}^i(t) \in \mathbb{R}^n$ are defined over the same time window with the appropriate adjoint identity

$$\langle\!\langle \mathbf{r}^i, \mathcal{M}(\mathbf{q}')^i \rangle\!\rangle_{t_{K-1}^+, t_K^-} = \langle\!\langle \mathcal{M}^*(\mathbf{r})^i, \mathbf{q}^i{}' \rangle\!\rangle_{t_{K-1}^+, t_K^-} + b^i, \tag{15a}$$

$$\langle\!\langle \mathbf{x}, \mathbf{y} \rangle\!\rangle_{t_{K-1}^+, t_K^-} \triangleq \int_{t_{K-1}^+}^{t_K^-} \mathbf{x}(t)^T \mathbf{y}(t) \, dt, \tag{15b}$$

$$\mathcal{M}^*(\mathbf{r})^i = -\frac{d\mathbf{r}^i(t)}{dt} - F^T \mathbf{r}^i(t), \tag{15c}$$

$$b^i = (\mathbf{r}_K^{i-})^T \mathbf{q}_K^i{}' - (\mathbf{r}_{K-1}^{i+})^T \mathbf{q}_{K-1}^i{}', \tag{15d}$$

then by letting

$$\mathcal{L}^*(S) = 0, \qquad \mathcal{M}^*(\mathbf{r})^i = 0, \tag{16a}$$

$$S_K^- = (I - H_K^T L_K^T) S_K^+ (I - L_K H_K), \tag{16b}$$

$$\begin{aligned} \mathbf{r}_K^{i-} &= \mathsf{trace}\left[(2P_K^- (H_K^T L_K^T - I) S_K^+ L_K)_i \left(\frac{dH_K^i}{d\mathbf{q}_K^i}\right)\right] \\ &\quad + \mathsf{trace}\left[(L_K^T S_K^+ L_K)_{ii} \left(\frac{dR_K^i}{d\mathbf{q}_K^i}\right)\right], \end{aligned} \tag{16c}$$

and substitute into (14), leveraging the adjoint identities (14) is transformed to:

$$\begin{aligned} J' &= \mathsf{trace}((S_{K-1}^+)^T P_{K-1}^+{}') + \sum_{i=1}^{M} \int_0^{t_K^-} \mathbf{u}^i(t)^T Q_u \mathbf{u}^i(t)' \, dt \\ &\quad + \sum_{i=1}^{M} \Big(\int_{t_{K-1}^+}^{t_K^-} \mathbf{r}^i(t)^T \mathcal{B}(\mathbf{u}')^i \, dt + (\mathbf{r}_{K-1}^{i+})^T \mathbf{q}_{K-1}^i{}' \Big). \end{aligned} \tag{17}$$

Equation (17) bears a strong resemblance to (13), except for the shifted time index on the first term and the additional third and fourth terms. In general for a given measurement interval $[t_{k-1}^+, t_k^+]$ where the initial J' equation is

$$\begin{aligned} J' &= \mathsf{trace}((S_k^+)^T P_k^{+\prime}) + \sum_{i=1}^{M} \int_0^{t_K^-} \mathbf{u}^i(t)^T Q_u \mathbf{u}^i(t)' \, dt \\ &\quad + \sum_{i=1}^{M} \Big(\int_{t_k^+}^{t_K^-} \mathbf{r}^i(t)^T \mathcal{B}(\mathbf{u}')^i \, dt + (\mathbf{r}_k^{i+})^T \mathbf{q}_k^{i\prime} \Big), \end{aligned} \tag{18}$$

if the following is enforced:

$$S_k^- = (I - H_k^T L_k^T) S_k^+ (I - L_k H_k), \tag{19a}$$

$$\begin{aligned} \mathbf{r}_k^{i-} &= \mathbf{r}_k^{i+} + \mathsf{trace}\left[\left(2P_k^- (H_k^T L_k^T - I) S_k^+ L_k\right)_i \left(\frac{dH_k^i}{d\mathbf{q}_k^i}\right)\right] \\ &\quad + \mathsf{trace}\left[(L_k^T S_k^+ L_k)_{ii} \left(\frac{dR_k^i}{d\mathbf{q}_k^i}\right)\right], \quad \mathbf{r}_K^{i+} = 0, \end{aligned} \tag{19b}$$

$$S_k^- \rightarrow S_{k-1}^+ \quad \text{via } \frac{dS(t)}{dt} = -A^T S(t) - S(t)A, \tag{19c}$$

$$\mathbf{r}_k^{i-} \rightarrow \mathbf{r}_{k-1}^{i+} \quad \text{via } \frac{d\mathbf{r}^i(t)}{dt} = -F^T \mathbf{r}^i(t), \tag{19d}$$

where $\rightarrow$ denotes propagation, then (18) is rewritten into

$$\begin{aligned} J' &= \mathsf{trace}((S_{k-1}^+)^T P_{k-1}^+{}') + \sum_{i=1}^{M} \int_0^{t_K^-} \mathbf{u}^i(t)^T Q_u \mathbf{u}^i(t)' \, dt \\ &\quad + \sum_{i=1}^{M} \Big(\int_{t_{k-1}^+}^{t_K^-} \mathbf{r}^i(t)^T \mathcal{B}(\mathbf{u}')^i \, dt + (\mathbf{r}_{k-1}^{i+})^T \mathbf{q}_{k-1}^i{}' \Big). \end{aligned} \tag{20}$$

Through the structure of (19a) and (19c), it is clear that in fact the entire $S(t)$ trajectory is symmetric. Also we see that by defining the variables according to (19), the time index in the J' equation is iteratively shifted toward t_0. Eventually, the J' transformation reaches time t_0, where from earlier result $P_0' = 0$ and $\mathbf{q}_0^i{}' = 0$. Hence the final J' equation is

$$J' = \sum_{i=1}^{M} \int_0^{t_K^-} \mathbf{r}^i(t)^T \underbrace{\mathcal{B}(\mathbf{u}')^i}_{G\mathbf{u}^i(t)'} + \mathbf{u}^i(t)^T Q_u \mathbf{u}^i(t)'\, dt$$
$$= \sum_{i=1}^{M} \int_0^{t_K^-} \underbrace{(G^T\mathbf{r}^i(t) + Q_u\mathbf{u}^i(t))^T}_{\nabla_{\mathbf{u}^i(t)} J} \mathbf{u}^i(t)'\, dt, \qquad (21)$$

which is in the necessary form to obtain the local gradient information. This gradient information is ready to be used by an iterative optimization method.

To recap, we started with (8d), which is not in the correct form as in (9) to obtain the local gradient information. Through defining the proper adjoint identities (11), (12), and (15), leveraging (8c), (8e), (8f), and the trace identity, and correctly setting $\mathcal{L}^*(S)$, S_k^-, $\mathcal{M}^*(\mathbf{r})^i$, and $\mathbf{r}_k^{i-}$ in (19), (8d) is transformed iteratively until the final form in (21). The local gradient can be easily extract at this point to be used by an iterative optimization methods.

III. The Ensemble Kalman Filter

In environmental flow systems, spatial discretization of the relevant PDEs routinely requires $O(10^5)$ or larger state dimension for adequate-fidelity representation of the physics of interest. In such large-scale systems, classical estimation methods like the Kalman Filter and the Extended Kalman Filter (EKF) are numerically intractable, due both to their poor computational scaling with increasing state dimension as well as their inability to represent nongaussian statistics. The Ensemble Kalman Filter (EnKF, see [8] and [9]) addresses these challenges in an efficient stochastic manner by propagating an ensemble of N perturbed candidate system trajectories, inferring the principle directions of estimation uncertainty from the ensemble distribution.

The KF propagates the estimation error covariance, P, and uses it to perform measurement updates. In contrast, the EnKF measurement update is based on a sample covariance, denoted Σ, obtained via an outer product of the ensemble perturbations:

$$\Sigma = \frac{(\delta\hat{\mathbf{X}})(\delta\hat{\mathbf{X}})^T}{N-1}, \qquad \delta\hat{\mathbf{X}} = \begin{bmatrix}\delta\hat{\mathbf{x}}^1 & \cdots & \delta\hat{\mathbf{x}}^N\end{bmatrix}, \qquad (22a)$$

$$\delta\hat{\mathbf{x}}^j = \hat{\mathbf{x}}^j - \bar{\mathbf{x}}, \qquad \bar{\mathbf{x}} = \frac{1}{N}\sum_{j=1}^{N}\hat{\mathbf{x}}^j, \qquad (22b)$$

where $\hat{\mathbf{x}}^j$ is the j'th ensemble member state estimate. The sample covariance Σ is then used instead of the full covariance P to perform measurement updates on each ensemble member independently:

$$\hat{\mathbf{x}}_k^{j+} = \hat{\mathbf{x}}_k^{j-} + \Sigma_k^- H_k^T (H_k \Sigma_k^- H_k^T + R_k)^{-1} \times$$
$$(\mathbf{y}_k - H_k \hat{\mathbf{x}}_k^{j-} + \mathbf{v}_k^j), \qquad (23)$$

where $\mathbf{v}_k^j \sim N(0, R_k)$. The reason for additional $\mathbf{v}_k^j$ perturbation is to maintain statistical consistency between EnKF and KF [10]. Because EnKF is a Monte-Carlo method, the ensemble estimate and sample covariance converge to the KF solution as $N \to \infty$ (see [11]).

In practice, due to computational constraints, the number of ensemble members [typically, $O(100)$] is rarely even comparable to the state dimension itself [typically, $O(10^5)$ or larger]. Due to this discrepancy, Σ is a low-rank, under-resolved approximation of the covariance P; generally, Σ contains spurious correlation between pairs of states separated over a large physical distances which, it may be argued on physical grounds, should be small. Localization (see [12]) is an ad-hoc method designed to suppress such non-physical long-distance correlations. It is typically applied as a weighting or "damping" function on the sample covariance

$$\Sigma = \frac{\rho \bullet (\delta\hat{\mathbf{X}})(\delta\hat{\mathbf{X}})^T}{N-1}, \qquad (24)$$

where ρ weights the ith and jth state correlection based on the physical distance between the two, and $\bullet$ denotes the element-wise product. Typically ρ is chosen to diminish to zero as this distance increases, and approaches unity as this distance decreases. Applying ρ directly on elements of Σ is not practical due to the size of Σ, thus approximations are made such that (23) is rewritten as

$$\hat{\mathbf{x}}_k^{j+} = \hat{\mathbf{x}}_k^{j-} + \rho_s \bullet (\Sigma_k^- H^T)(\rho_m \bullet (H_k \Sigma_k^- H_k^T) + R_k)^{-1} \times$$
$$(\mathbf{y}_k - H_k \hat{\mathbf{x}}_k^{j-} + \mathbf{v}_k^j), \qquad (25)$$

where $\rho_s^{i,j}$ applies weighting based on distance between the ith state and the jth measurement, and $\rho_m^{i,j}$ applies weighting based on the distance between the ith and jth measurement.

IV. EXPERIMENTAL DESIGN AND RESULTS

The PDE system considered is the 2D Navier-Stokes Equation (NSE) with additive low-frequency forcing coupled with a passive scalar field [that is, a scalar field (modeling, e.g., smoke density) advected by the velocity field, but which itself does not affect the evolution of the velocities field]. There is source of the scalar near the center of the physical domain. The governing equations are

$$\frac{\partial\mathbf{x}}{\partial t} = -\mathbf{x}\cdot\nabla\mathbf{x} + \nu\nabla^2\mathbf{x} + \frac{1}{\rho}\nabla p + \mathbf{f}_{\mathbf{d}} \qquad (26a)$$

$$\frac{\partial\phi}{\partial t} = -\mathbf{x}\cdot\nabla\phi + \kappa\nabla^2\phi + \mathbf{f}_\phi \qquad (26b)$$

$$h_k^i(\mathbf{q}_k^i, \mathbf{x}_k) = H_k^i(\mathbf{q}_k^i)\mathbf{x}_k \qquad (26c)$$

with density ρ, kinematic viscosity ν, pressure p, and diffusion constant κ. $\mathbf{x}$ is a velocity vector field containing the horizontal and vertical velocities, and ϕ is the passive scalar field simulating an environmental plume. The ith measurement operator measures the local quantity of $\mathbf{x}$ and ϕ at the ith sensor position contained in $\mathbf{q}_k^i$. Each sensor vehicle is modeled with classic point-mass dynamics with damping, where the point-mass horizontal and vertical accelerations are controlled independently.

Numerical simulation of (26) uses the pseudo-spectral code developed in [13] on a 64×64 uniform square grid. In all numerical experiments the "truth" simulation uses an

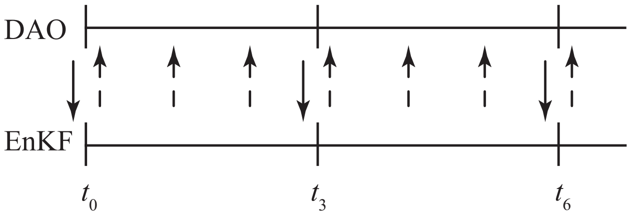

Fig. 2. Cartoon illustrating how DAO and EnKF interacts in time. During each measurement the EnKF sends new the Σ to DAO (dashed up arrow) to update the optimization problem. The optimization lasts 3 measurements and the future vehicle waypoint sequences are sent to the vehicles (solid down arrow).

identical model (with different initial conditions and random forcing) running in parallel with the EnKF. The simulations are done in a periodic domain nondimensionalized with width $L = 2$. To simulate estimating a non-periodic domain, the estimation is isolated to a subregion of width $R = 1.35$ centered in the domain. The targeting matrix T in (7) is chosen in DAO to focus on the entire estimation subregion. For flow computation stability, a relatively small marching time-step ($\Delta t = 0.005$ time unit) is used, and measurements are taken every $\tau_{meas} = 0.15$ time units. For simplicity, the forecast time and the time of the last measurement are taken to coincide (that is, $T_F = T_K$). The localization strategy for ρ_s and ρ_m are both taken as

$$\rho = e^{-4d^2}, \tag{27}$$

where d is the distance between the quantity of interest.

In this experiment DAO optimizes the vehicle waypoints subjected to vehicle dynamics over six measurement times, thus the event horizon is $T_F = \tau_{meas} \times 6 = 0.9$. At each measurement DAO receives the current estimate sample covariance Σ from the EnKF and updates the optimization problem. The optimization lasts 3 measurements, and at the end the optimal control trajectory iterate (whether the optimization has converged or not) is used to forecast future optimal vehicle waypoints. The future six-vehicle-waypoints sequences are sent to the vehicles, but only the first three are used; the last three serve as "backup" should the vehicles fail to receive the next set of waypoints (though this is not necessary for the present idealized numerical simulation, it could be useful in future physical realizations). The control trajectories for the last three waypoints are used as part of the initial control trajectories for the next optimization. Fig. 2 illustrates how DAO and EnKF interacts.

Because the DAO algorithm is formulated based on the Kalman Filter, it doesn't escape the fact that it requires the full covariance matrix P propagation, which as mentioned earlier, is computationally intractable. Rather than propagating $P(t)$ using the fully model dynamics with (5), a simple "growing" model

$$\frac{dP(t)}{dt} = 0.1P(t) + 0.1P(t)^T, \tag{28}$$

is used instead. With an additional assumption that P is diagonally dominant (effectively, that the spatial cross-correlations are negligible), it can be shown the DAO algorithm can be carried out entirely without approximations by propagating and storing only the diagonals of $P(t)$ and $S(t)$, which dramatically reduces the computational and storage requirements. Both approximations are justified by the small event horizon T_F considered in the present simulation, where numerical experiment data collected on (26) (not shown) suggests the equations are essentially linear with little cross-correlations between states within this event horizon.

To quantify the estimate quality, we consider the steady-state, infinite-time averaged absolute error, defined by [14] as the difference between the estimate and the truth squared, integrated over the estimation subregion:

$$\mathrm{Errn}(\hat{d}, d_{tru}) = \int_{\Omega} (\, \hat{d} - d_{tru}\,)^2 \, d\Omega, \tag{29}$$

where $()_{tru}$ corresponding the "truth" values. Long-time averages of this measure applied to both the velocity field and the scalar are used to approximate the expected value, $\mathcal{E}[\mathrm{Errn}(\hat{d}(t), d_{tru}(t))] \triangleq 1/T \int_0^T \mathrm{Errn}(\hat{d}(t), d(t))\, dt$, at statistical steady-state.

Fig. 3 and Fig. 4 compare the time-averaged error within a time interval using three different adaptive observation strategies: sensors following a random walk, sensors distributed uniformly in the estimation subregion, and the present DAO strategy. As a baseline to compare against, when no measurements are taken, the average estimation error as defined in (29) for the flow velocity is 34.5, and 4.06 for the scalar. Fig. 5 provides a typical example of the waypoints optimized by DAO.

These results in Fig. 3 and Fig. 4 demonstrate that significant improvements in the estimate can be accomplished via path planning. The DAO algorithm is able to achieve a 47% reduction for the flow and 42% reduction for the scalar estimation error compare to the random walk scheme, and 25% and 17% respectively compared against uniformly distributed sensors. Further, these results also suggest that deliberate sensor placement is more important than unorganized movement, as demonstrated by the performance of uniformly distributed sensors against the random walk.

V. CONCLUSIONS AND FUTURE WORK

This paper combines our new Dynamic Adaptive Observation (DAO) algorithm and the well-known Ensemble Kalman Filter (EnKF) to estimate an environmental flow represented with a passive scalar emanating from a source and driven convectively in a 2D randomly-forced flow. Unlike existing AO algorithms, the DAO algorithm rigorously incorporates vehicle dynamics, and computes optimal vehicle waypoints that (locally) minimizes the forecast uncertainty. Numerical experiment demonstrates a significant reduction in the forecast error over less deliberate sensor routing strategies.

Because DAO is formulated with the Kalman Filter, one disadvantage is that propagation of the full covariance matrix is required, which is not practical due to the problem size. We side-stepped this problem in the numerical experiment

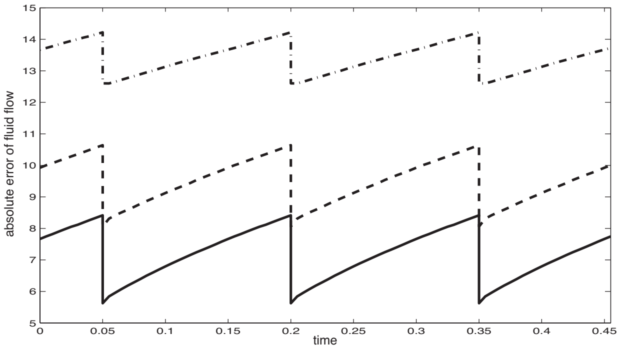

Fig. 3. Time-averaged absolute error of the flow velocities within a time interval, following three AO strategies: (1) sensors following a random walk, average 13.4 (dot-dashed) (2) stationary sensors uniformly distributed over the estimation subregion, average 9.5 (dashed), and (3) sensor trajectories provided by DAO, average 7.1 (solid). The error increases between measurements, and decreases at the EnKF measurement updates, thus creating the "saw-tooth" shape in the error plot.

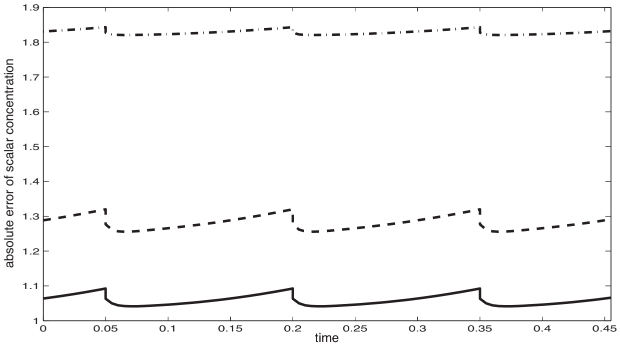

Fig. 4. Time-averaged absolute error of the scalar within a time interval, with the sensor motion as described in Fig. 3. Because the scalar evolution is primarily driven by the flow velocities, the scalar estimate absolute error dips slightly after each measurement update, due to the improved velocity estimate. The average errors are 1.82 for random walk, 1.28 for uniformly distributed stationary sensors, and 1.06 for DAO.

reported here by assuming a simplified model of the covariance evolution that reduces the necessary computations significantly. For better performance, the underlying model should instead be fully leveraged; hence we are working toward adopting the EnKF into the DAO formulation in place of the KF.

A new Hybrid variational / Ensemble Kalman Smoother (HEnS) algorithm for state estimation has also recently been proposed by our group [15]. Preliminary tests have show that this new algorithm outperforms the EnKF in the presence of substantial non-Gaussian uncertainties. Steps are being taken to combine HEns with DAO in the near future.

References

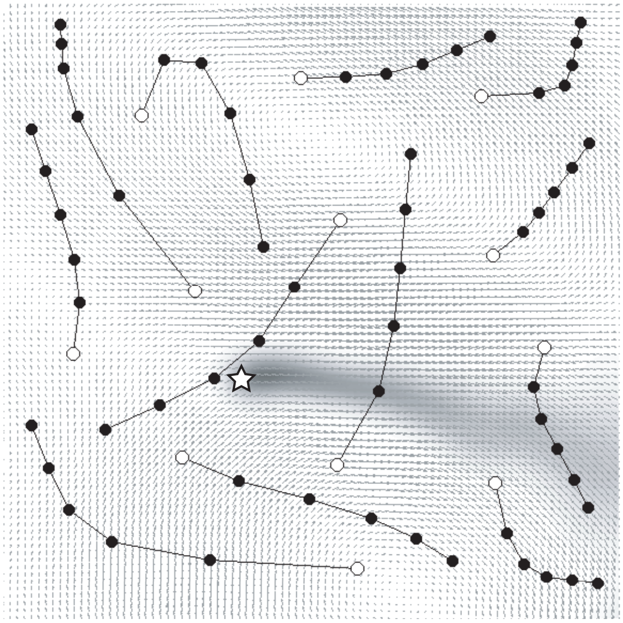

Fig. 5. A typical example of the sensor waypoints optimized by DAO, overlaying the velocity field and passive scalar estimated by the EnKF. The passive scalar source (star) is located near the center of the domain. Empty circles denote the initial vehicle positions with five subsequent optimized vehicle waypoints denote as black circles.

[1] S. Martínez, J. Cortés, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.

[2] K. Laventall and J. Cortés, "Coverage control by multi-robot networks with limited-range anisotropic sensory," *International Journal of Control*, vol. 82, pp. 1113–1121, 2009.

[3] M. S. Stanković and D. M. Stipanović, "Stochastic extremum seeking with applications to mobile sensor networks," in *ACC'09: Proceedings of the 2009 conference on American Control Conference*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 5622–5627.

[4] 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, March 2010.

[5] R. Langland and G. Rohaly, "Adjoint-Based Targeting of Observations for FASTEX Cyclones," *Defense Technical Information Center*, 1996.

[6] R. Buizza and A. Montani, "Targeting Observations Using Singular Vectors," *Journal of the Atmospheric Sciences*, vol. 56, no. 17, pp. 2965–2985, 1999.

[7] C. Bishop, B. Etherton, and S. Majumdar, "Adaptive Sampling with the Ensemble Transform Kalman Filter. Part I: Theoretical Aspects," *Monthly Weather Review*, vol. 129, no. 3, pp. 420–436, 2001.

[8] G. Evensen, "The Ensemble Kalman Filter: theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[9] ——, *Data assimilation: The ensemble Kalman filter*. Springer Verlag, 2009.

[10] G. Burgers, P. Jan van Leeuwen, and G. Evensen, "Analysis Scheme in the Ensemble Kalman Filter," *Monthly Weather Review*, vol. 126, no. 6, pp. 1719–1724, 1998.

[11] M. Butala, J. Yun, Y. Chen, R. Frazin, and F. Kamalabadi, "Asymptotic convergence of the ensemble kalman filter," in *ICIP 15th IEEE International Conference on Image Processing*, 2008, pp. 825 –828.

[12] P. Houtekamer and H. Mitchell, "A sequential ensemble Kalman filter for atmospheric data assimilation," *Monthly Weather Review*, vol. 129, no. 1, pp. 123–137, 2001.

[13] T. Bewley, P. Moin, and R. Temam, "Dns-based predictive control of turbulence: an optimal benchmark for feedback algorithms," *Journal of Fluid Mechanics*, vol. 447, pp. 179–225, 2001.

[14] T. Bewley and B. Protas, "Skin friction and pressure: the "footprints" of turbulence," *Physica D: Nonlinear Phenomena*, vol. 196, no. 1-2, pp. 28–44, 2004.

[15] J. Cessna and T. Bewley, "A hybrid (variational/kalman) ensemble smoother for the estimation of nonlinear high-dimensional discretizations of pde systems," availible at http://fccr.ucsd.edu.