

Adaptive Observation of Stationary Fields with Mobile Robotic Systems

David D Zhang* and Thomas R Bewley

Abstract—Adaptive Observation (AO) strategies address the effective deployment of mobile sensors for the estimation and forecasting of physical systems. Of the many approaches to the AO problem, few incorporate fully the dynamics of moving sensors into the trajectory planning algorithm. We propose a new AO algorithm, dubbed Dynamic Adaptive Observation (DAO), which optimizes trajectories for the minimization of forecast uncertainty while rigorously respecting the dynamic constraints on the vehicle motion. This new algorithm is tested here on a variety of stationary problems, with effective results.

I. INTRODUCTION

Adaptive Observation (AO) is a curious problem midway between control theory and estimation theory. The idea of AO, ultimately, is to determine control inputs to route mobile sensors in the near future in a manner that minimizes the uncertainty of a future state estimate or forecast. AO algorithms can be either distributed or centralized.

In distributed AO, each mobile sensor has little or no knowledge of the sensed system, and deployment is planned locally. The idea is that collective simple behaviors may lead to global actions that improve the uniformity of the sensor distribution, while perhaps clustering sensors in regions of particular interest. Its inherent simplicity enables distributed AO to be deployed easily. Existing distributed AO algorithms typically reduce the AO problem to an optimal coverage problem (see [1], [2]), or to an extremum or level-set seeking problem (see [3], [4], [5]). While these approaches work adequately for certain applications, their performance is degraded in convection-driven (that is, wind-dominated rather than diffusion dominated) problems with complicated level sets, such as those encountered in atmospheric and oceanographic applications.

In contrast, centralized AO strategies leverage the sensed system model to optimize the sensor motions. The system models used in centralized AO are often computationally intensive; hence, the bulk of the computations performed in AO implementations are typically off-loaded to a centralized supercomputer cluster, and the optimized trajectories (or waypoints selected along them) broadcast to the vehicles. Note that [6] reviews some centralized AO algorithms used in the weather forecast community, and [7] reviews some AO implementations in the ocean forecasting community.

As we are interested in applying AO to large-scale systems, we focus the present work on centralized AO strategies. These strategies may be further divided into “sensitivity-based” and “uncertainty-based” approaches. In sensitivity-based AO (see [8], [9]), a system adjoint is used to reveal “sensitive regions” of the domain that contribute significantly

to the forecast uncertainty. However, most approaches that follow such an approach do not address how such “sensitive regions” should be optimally probed by the mobile sensors (note that this issue is partially addressed in [10] and [6]). The strength of sensitivity-based AO algorithms is their relative speed and computational efficiency, as the cost of the sensitivity computation is on the same order as the computational cost of the forward system propagation itself.

Uncertainty-based AO algorithms (see [11], [12]) take an altogether different approach. Rather than computing the sensitivity of the forecast, they seek a measurement location sequence that minimizes the forecast uncertainty. This is usually achieved by considering a large set of possible vehicle waypoint sequences, and computing the anticipated forecast uncertainty associated with each. For example, when implementing an Ensemble Transform Kalman Filter (ETKF) on a practical system, [13] simply considered 40 pre-approved feasible flight paths, and selected among them. In complex systems with a large number of sensors, this set might be extremely large, and it is computationally infeasible to search them all. As recognized by [11], as each forecast is computed independently, this algorithm is at least “embarrassingly” parallel (that is, the algorithm completion time is inversely proportional to the computational resources available); however, it is generally not very efficient. In [12], the authors search for a waypoint sequence using a Mixed Integer Linear Programming (MILP) method, thereby substantially reducing the set of trajectories that must be considered. In this work, vehicle dynamics are approximated with linear constraints in the MILP. Because linear constraints cannot model complex vehicle dynamic constraints, the authors had to be overly conservative with the imposition of the constraints to ensure a dynamically feasible solution in the end, resulting in unnecessarily sluggish optimized vehicle trajectories.

The aforementioned centralized AO algorithms consider systems with time scales much slower than the time scales of the moving sensors themselves. Thus, vehicle dynamics have not been fully incorporated into the formulation of any of them. The present algorithm fills this void. We appreciate the steps towards computational efficiency that have been achieved by both sensitivity-based AO strategies and uncertainty-based AO strategies leveraging MILP methods, but seek a method that more precisely takes the actual vehicle dynamic constraints into account.

Towards this end, we propose a new centralized AO algorithm, dubbed Dynamic Adaptive Observation (DAO), that combines various features from existing AO approaches while incorporating the full vehicle dynamics. DAO uses the Kalman Filter to predict the future estimation error covariance and to compute the best control, subject to the vehicle dynamic constraints, to minimize the forecast uncertainty. This is achieved by minimizing a relevant cost function

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

*Corresponding author: dazhang@ucsd.edu

balancing a metric of the forecast quality with another metric measuring the cost of the control applied to the vehicles. Because explicit formulation of the optimal control with respect to the cost function is difficult to derive analytically, we use adjoint analysis to calculate the local gradient, and compute the optimal control iteratively.

The rest of the paper is as follows: in §II, we formulate the AO problem, where our objective is to minimize a cost balancing a measure of the forecast quality with vehicle-related penalties. Adjoint analysis is performed on the cost to reveal local gradient information in §III; this local gradient is used to iteratively optimize the control. Various generalizations to DAO are discussed in §IV, and some example and our conclusions are presented in §V and §VI.

II. AO PROBLEM FORMULATION

Consider N autonomous vehicles are available in a domain. Within the domain there is a discretized PDE $\mathbf{f}$ with state variables defined over n grid points. The field evolves with the following underlying discrete-time linear dynamics

$$\mathbf{f}_{k+1} = A\mathbf{f}_k + B\mathbf{w}_k, \quad \mathbf{w}_k \sim N(0, W), \quad (1)$$

where the normally distributed zero-mean $\mathbf{w}_k$ with covariance W models random forcing and uncertainty in the model.

The i th vehicle's discrete-time dynamical equation with states and controls variables $\mathbf{q}^i$ and $\mathbf{u}^i$ is:

$$\mathbf{q}_{k+1}^i = F\mathbf{q}_k^i + G\mathbf{u}_k^i. \quad (2)$$

As the vehicles move and measure various aspect of $\mathbf{f}$, vehicle states such as position, heading, and velocity affect the measurement, and the measurement noise statistics may also be affected. Hence vehicle i 's measurement matrix H^i and measurement noise covariance R^i are dependent on the vehicle states $\mathbf{q}^i$. For convenience, the notation to emphasize the state dependencies are dropped with the understanding that it is implied when H_k^i and R_k^i are used. The measurement vector $\mathbf{y}_k^i$ from the i th vehicle is modeled

$$\mathbf{y}_k^i = H_k^i \mathbf{f}_k + \mathbf{v}_k^i, \quad \mathbf{v}_k^i \sim N(0, R_k^i), \quad (3)$$

with the collective R_k , H_k , and $\mathbf{y}_k$

$$R_k = \begin{bmatrix} R_k^1 & & 0 \\ & \ddots & \\ 0 & & R_k^N \end{bmatrix}, H_k = \begin{bmatrix} H_k^1 \\ \vdots \\ H_k^N \end{bmatrix}, \mathbf{y}_k = \begin{bmatrix} \mathbf{y}_k^1 \\ \vdots \\ \mathbf{y}_k^N \end{bmatrix}. \quad (4)$$

The vehicle measurements are assimilated to improve the field estimate, $\hat{\mathbf{f}}$, which is modeled

$$\hat{\mathbf{f}}_{k+1}^- = A\hat{\mathbf{f}}_k^+, \quad (5a)$$

$$\hat{\mathbf{f}}_k^+ = \hat{\mathbf{f}}_k^- - L_k(\mathbf{y}_k - \hat{\mathbf{y}}_k), \quad (5b)$$

with the predicted measurement vector $\hat{\mathbf{y}}_k$ defined as

$$\hat{\mathbf{y}}_k = H_k \hat{\mathbf{f}}_k^-, \quad (6)$$

where $()^+$ and $()^-$ denote the *background* (before the measurement update) and the *analysis* (after the measurement update), respectively. L_k is the standard discrete-time Kalman Filter gain

$$L_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \quad (7)$$

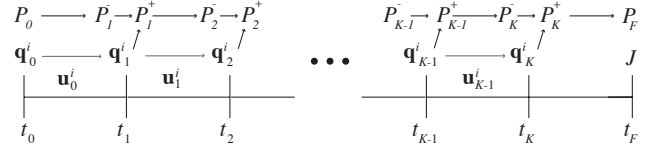


Fig. 1. Cartoon illustrating the problem formulation. The controls $\mathbf{u}_k^i$ affect the evolution of the sensor vehicle trajectories $\mathbf{q}_k^i$. The sensor vehicle positions at the measurement times, $\mathbf{q}_k^i$, in turn, affect the updates to the covariance of the estimation error P_k . The cost J depends on P_F and the $\mathbf{u}_k^i$; a set of controls $\mathbf{u}_k^i$ is sought to minimize this cost.

where P_k is the state estimate error covariance $P_k = \mathcal{E}\{\tilde{\mathbf{f}}_k \tilde{\mathbf{f}}_k^T\}$ with expected value operator $\mathcal{E}(\cdot)$ and estimation error $\tilde{\mathbf{f}}_k = \mathbf{f}_k - \hat{\mathbf{f}}_k$. The evolution of P_k is derived from (1) and (5)

$$P_{k+1}^- = AP_k^+ A^T + BWB^T \quad (8a)$$

$$P_k^+ = P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^-. \quad (8b)$$

Our AO problem is framed as followed. At initial time $k = 0$ the vehicle states and estimation error covariance, $\mathbf{q}_0^i$ and P_0 , are known. Design a minimal effort control input sequence $\mathbf{u}_k^i$ for each vehicle within the time window $k \in [0, K]$, such that a metric of the forecast quality at some final time $k = F$, $F \geq K$, conditioned on the measurements gathered by the vehicles, is minimized. We choose a quadratic penalty to quantify control effort and the sum of the variance as the forecast quality metric. This problem can be solved by finding a minimizing solution to the following (scalar) cost function J :

$$\min_{\mathbf{u}_k^i} J = \text{trace}(TP_F) + \frac{1}{2} \sum_{i=1}^N \sum_{k=0}^{K-1} (\mathbf{u}_k^i)^T Q_u \mathbf{u}_k^i, \quad (9)$$

where T is a diagonal matrix that *targets* specific regions of the domain (e.g. $T = I$ if the region is the entire domain), and Q_u is a symmetric positive-definite matrix. Fig. 1 illustrates the relationships between J , P_k , $\mathbf{q}_k^i$ and $\mathbf{u}_k^i$.

Due to the non-linear effect $\mathbf{u}_k^i$ has on P_K , J is a non-linear, possibly non-convex, function in $\mathbf{u}_k^i$; hence J may contain multiple local minimum.

III. COMPUTING OPTIMAL $\mathbf{u}_k^i$

One way to find the optimal solution for (9) is by analytically compute the derivative of J in terms of $\mathbf{u}_k^i$ and set the derivative equal to zero and solve for the solution in *one step*. However expressing J explicitly in $\mathbf{u}_k^i$ is complicated, as seen from Fig. 1. Another way is computing the solution *iteratively* using gradient-based optimization methods, which utilize the local gradient of J w.r.t. $\mathbf{u}_k^i$. We employ *adjoint analysis* to find this gradient $\nabla_{\mathbf{u}_k^i} J$, and this is illustrated in the following.

In addition to the pre-specified $\mathbf{q}_0^i$ and P_0 , suppose we assume an initial $\mathbf{u}_k^i$; with this, $\mathbf{q}_k^i$ and P_k are propagated and J evaluated. A perturbation is applied to the initial $\mathbf{u}_k^i$, which sets off a chain reaction that also causes perturbations

in $\mathbf{q}_k^i$, R_k^i , H_k^i , P_k^i and J . The first-order perturbations to these variables are:

$$\mathbf{q}_{k+1}^i = F\mathbf{q}_k^i + G\mathbf{u}_k^i, \quad \mathbf{q}_0^i = 0, \quad (10a)$$

$$P_{k+1}^+ = AP_k^+ A^T, \quad P_0^+ = 0, \quad W^+ = 0 \quad (10b)$$

$$P_k^+ = P_k^- - (P_k^- H_k^T + P_k^- (H_k')^T) L_k^T - L_k (H_k P_k^- + H_k' P_k^-) + L_k (H_k' P_k^- H_k^T + H_k P_k^- H_k'^T + H_k P_k^- (H_k')^T + R_k') L_k^T, \quad (10c)$$

$$J' = \text{trace}(TP_F') + \sum_{i=1}^N \sum_{k=0}^{K-1} (\mathbf{u}_k^i)^T Q_u \mathbf{u}_k^i, \quad (10d)$$

$$R_k' = \begin{bmatrix} R_k^{1'} & & 0 \\ & \ddots & \\ 0 & & R_k^{N'} \end{bmatrix}, \quad R_k^i = \left(\frac{dR_k^i}{d\mathbf{q}_k^i} \right)^T \mathbf{q}_k^i, \quad (10e)$$

$$H_k' = \begin{bmatrix} H_k^{1'} \\ \vdots \\ H_k^{N'} \end{bmatrix}, \quad H_k^i = \left(\frac{dH_k^i}{d\mathbf{q}_k^i} \right)^T \mathbf{q}_k^i. \quad (10f)$$

The reason why $\mathbf{q}_k^i$, P_0^+ , and W^+ are zero is because we don't have control over these variables. 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 contracts to rank-2 matrices R_k^i and H_k^i by the inner product with $\mathbf{q}_k^i$. The matrix inverse identity $(\Phi^{-1})' = -\Phi^{-1}\Phi'\Phi^{-1}$ [14] is used in (10c), with $\Phi = H_k P_k H_k^T + R_k$. The Taylor Expansion of J about the initial $\mathbf{u}_k^i$ is

$$J' = \sum_{i=1}^N \sum_{k=0}^{K-1} \left(\nabla_{\mathbf{u}_k^i} J(\mathbf{u}_k^i) \right)^T \mathbf{u}_k^i, \quad (11)$$

which resembles the second term in (10d). The remaining formulation illustrates how to convert $\text{trace}(TP_F')$ to a similar form.

We begin by first writing the forecast from P_K^+ to P_F using (8a) and compute the corresponding perturbation equation:

$$P_F = A(A(\cdots (AP_K^+ A^T + BWB^T) \cdots) A^T + BWB^T) A^T + BWB^T \quad (12a)$$

$$P_F' = A_{K,F} P_K^{+'} A_{K,F}^T, \quad A_{K,F} \triangleq \prod_{k=K}^{F-1} A. \quad (12b)$$

Substituting (12b) into (10d) and rearranging the terms using trace identity $\text{trace}(AB) = \text{trace}(BA) = \text{trace}(A^T B^T)$:

$$J' = \text{trace}(A_{K,F}^T T A_{K,F} P_K^{+'}) + \sum_{i=1}^N \sum_{k=0}^{K-1} (\mathbf{u}_k^i)^T Q_u \mathbf{u}_k^i. \quad (13)$$

Next (10b) is substituted into (10c) to form the evolution equation for P_k^+ , which allows us to only consider P_k^+ at any time k from now on; hence from hereon we shall use P_k in place of P_k^+ . The terms containing P_k' , R_k' , and H_k' are grouped together in the merged equation, and two operators

$\mathcal{L}(P')_k$ and $\mathcal{B}(R', H')_k$ are defined:

$$\mathcal{L}(P')_k \triangleq P_{k+1}' - AP_k' A^T - L_k H_k AP_k' A^T H_k^T L_k^T + AP_k' A^T H_k^T L_k^T + L_k H_k AP_k' A^T, \quad (14a)$$

$$\mathcal{B}(R', H')_k \triangleq L_k R_k' L_k^T - AP_k' A^T (H_k')^T L_k^T - L_k H_k' AP_k' A^T + L_k (H_k' AP_k' A^T H_k^T + H_k AP_k' A^T (H_k')^T) L_k^T, \quad (14b)$$

$$\mathcal{L}(P')_k = \mathcal{B}(R', H')_k. \quad (14c)$$

Note that (14c) is just a rearrangement of the merged equation replaced with the new operators. The $\mathcal{L}(P')_k$ operator defines the unforced dynamics of P_k' , while $\mathcal{B}(R', H')_k$ defines the forcing applied.

A matrix adjoint variable S_k is defined and an adjoint identity is framed based on a relevant inner product:

$$\langle S, \mathcal{L}(P') \rangle = \langle \mathcal{L}^*(S), P' \rangle + b, \quad \langle X, Y \rangle \triangleq \sum_{k=0}^{K-1} \text{trace}(X_k^T Y_k). \quad (15)$$

Note the RHS of the adjoint identity, namely $\mathcal{L}^*(S)_k$ and b , have yet to be defined. Doing the necessary rearrangement of sums and applying the trace identity when necessary, the LHS of (15) is recast into the form on the RHS, and the expression for $\mathcal{L}^*(S)_k$ and b are readily identified:

$$\mathcal{L}^*(S)_k = S_{k-1} - A^T (I - H_k^T L_k^T) S_k (I - L_k H_k) A, \quad (16a)$$

$$b = \text{trace}(S_{K-1}^T P_K'). \quad (16b)$$

Similar to the $\mathcal{L}(P')_k$ operator, the $\mathcal{L}^*(S)_k$ operator defines the unforced dynamics of S_k ; however unlike $\mathcal{L}(P')_k$, $\mathcal{L}^*(S)_k$ defines a propagation *backwards* in time. Furthermore, we have complete freedom in choosing the forcing applied to S_k and from b the starting condition of S . The goal here is to choose the forcing and starting condition of S carefully such that we could leverage (14c) to replace all P_k' terms into R_k' and H_k' .

Upon examining (13), S_{K-1} is chosen such that $S_{K-1} = A_{K,F}^T T A_{K,F}$ and $\mathcal{L}^*(S)_k = 0$ (that is, the evolution is S is unforced). Thus by (15) and (14c), (13) becomes:

$$J' = \langle S, \mathcal{B}(R', H') \rangle + \sum_{i=1}^N \sum_{k=0}^{K-1} (\mathbf{u}_k^i)^T Q_u \mathbf{u}_k^i. \quad (17)$$

Note by the symmetric construction of (16a) and the starting condition for S , for all time S_k is symmetric — just like the covariance matrix P_k during forward propagation. Inserting (14b) into (17) while leveraging the trace identity to shift R_k' and H_k' to the right,

$$J' = \sum_{i=1}^N \sum_{k=0}^{K-1} (\mathbf{u}_k^i)^T Q_u \mathbf{u}_k^i + \sum_{k=0}^{K-1} \text{trace}(L_k^T S_k L_k R_k') + \sum_{k=0}^{K-1} \text{trace}\left(2AP_k A^T (H_k^T L_k^T - I) S_k L_k H_k'\right),$$

and when the structure of H'_k and R'_k in (10e) and (10f) is also leveraged,

$$\begin{aligned} J' = & \sum_{i=1}^N \sum_{k=0}^{K-1} \text{trace}((L_k^T S_k^T L_k)_{ii} \left(\frac{dR_k^i}{d\mathbf{q}_k^i} \right)^T \mathbf{q}_k^{i'}) \\ & + \sum_{i=1}^N \sum_{k=0}^{K-1} \text{trace}((2AP_k A^T (H_k^T L_k^T - I) S_k L_k)_i \left(\frac{dH_k^i}{d\mathbf{q}_k^i} \right)^T \mathbf{q}_k^{i'}) \\ & + \sum_{i=1}^N \sum_{k=0}^{K-1} (\mathbf{u}_k^i)^T Q_u \mathbf{u}_k^{i'}, \end{aligned} \quad (18)$$

where $(L_k^T S_k^T L_k)_{ii}$ denotes the (i, i) block of the $N \times N$ block matrix $L_k^T S_k^T L_k$ and $(2AP_k A^T (H_k^T L_k^T - I) S_k L_k)_i$ denotes the i th column block of the $1 \times N$ block matrix $2AP_k A^T (H_k^T L_k^T - I) S_k L_k$.

So far (18) is not yet in the form required by (11). To further rewrite $\mathbf{q}_k^{i'}$ into $\mathbf{u}_k^{i'}$, the same adjoint analysis for the vehicle states is performed. From (10a) $\mathcal{M}(\mathbf{q}')_k^i$ is defined, along with N vector adjoint variables $\mathbf{r}_k^i$ and an adjoint identity based on a relevant inner product. After performing the necessary rearrangement of sums, $\mathcal{M}^*(\mathbf{r})_k^i$ and b^i are identified:

$$\underbrace{\mathbf{q}_{k+1}^{i'} - F \mathbf{q}_k^{i'}}_{\mathcal{M}(\mathbf{q}')_k^i} = G \mathbf{u}_k^{i'}, \quad (19a)$$

$$\langle \langle \mathbf{r}^i, \mathcal{M}(\mathbf{q}')^i \rangle \rangle = \langle \langle \mathcal{M}^*(\mathbf{r})^i, \mathbf{q}^{i'} \rangle \rangle + b^i, \quad \langle \langle \mathbf{x}, \mathbf{y} \rangle \rangle \triangleq \sum_{k=0}^{K-1} \mathbf{x}_k^T \mathbf{y}_k, \quad (19b)$$

$$\mathcal{M}^*(\mathbf{r})_k^i = \mathbf{r}_{k-1}^i - F^T \mathbf{r}_k^i, \quad (19c)$$

$$b^i = (\mathbf{r}_{K-1}^i)^T \mathbf{q}_K^{i'}. \quad (19d)$$

Leveraging these expressions, it follows that if the following adjoint conditions are also enforced:

$$\begin{aligned} \mathcal{M}^*(\mathbf{r})_k^i = & \text{trace}((2AP_k A^T (H_k^T L_k^T - I) S_k L_k)_i \frac{dH_k^i}{d\mathbf{q}_k^i}) \\ & + \text{trace}((L_k^T S_k^T L_k)_{ii} \frac{dR_k^i}{d\mathbf{q}_k^i}), \end{aligned} \quad (20a)$$

$$\mathbf{r}_{K-1}^i = 0, \quad (20b)$$

then the evolution equation and starting condition for $\mathbf{r}^i$ are both defined. Finally substituting (19a) and (20a) into (19b) reveals the relationship between $\mathbf{q}_k^{i'}$ and $\mathbf{u}_k^{i'}$, and (18) is readily expressed into the proper form required in (11) for extracting the local cost function gradient:

$$\begin{aligned} J' = & \sum_{i=1}^N \sum_{k=0}^{K-1} (\mathbf{r}_k^i)^T \mathcal{B}(\mathbf{u}')_k^i + \sum_{i=1}^N \sum_{k=0}^{K-1} (\mathbf{u}_k^i)^T Q_u \mathbf{u}_k^{i'} \\ = & \sum_{i=1}^N \sum_{k=0}^{K-1} \underbrace{(G^T \mathbf{r}_k^i + Q_u \mathbf{u}_k^i)^T}_{\nabla_{\mathbf{u}_k^i} J} \mathbf{u}_k^{i'}. \end{aligned} \quad (21)$$

This local gradient can then be used to update the current initial control sequence via a suitable minimization algorithm such as steepest descent, conjugate gradient, limited-memory BFGS, and etc. At the end of the minimization the updated

control sequence is used to start the next iteration of the optimization.

The DAO algorithm may be summarized as follows:

- ① Propagate $\mathbf{q}_0^i$ to $\mathbf{q}_K^i$ with $\mathbf{u}_k^i$ using (2), then propagate P_0 to P_K^+ using (8)
- ② $S_{K-1} \leftarrow A_{K,F}^T T A_{K,F}$ (Note this is equivalent to defining $S_F \leftarrow T$ at time $k = F$ and propagating S_F to S_{K-1} using dynamics $S_{k-1} = A_k^T S_k A_k$), and $\mathbf{r}_{K-1}^i \leftarrow 0$.
- ③ Propagate S_{K-1} to S_0 using (16a), then propagate $\mathbf{r}_K^i$ to $\mathbf{r}_0^i$ using (20a)
- ④ $\nabla_{\mathbf{u}_k^i} J \leftarrow G^T \mathbf{r}_k^i + Q_u \mathbf{u}_k^i$. Use gradient-based optimization method to find the decent direction $\mathbf{p}_k^i$ and step size α
- ⑤ $\mathbf{u}_k^i \leftarrow \mathbf{u}_k^i + \alpha \mathbf{p}_k^i$, check convergence, and return to ① if not yet converged.

IV. GENERALIZATION

For clarity sake, we restricted the DAO derivation in section III to a specific cost function, linear dynamics, and identical dynamics and sensors in all vehicles. The generalizations of these restrictions are discussed here.

A. Generalize Cost Function

The vehicle penalty portion in (9) is not restricted to be quadratic and penalizes only $\mathbf{u}_k^i$, other types of penalties can be incorporated. In general,

$$J = \text{trace}(TP_F) + \sum_{i=1}^N \left(\sum_{k=0}^{K-1} g^i(\mathbf{q}_k^i, \mathbf{u}_k^i) + h^i(\mathbf{q}_K^i) \right). \quad (22)$$

Note if $g^i(\cdot, \cdot)$ and $h^i(\cdot)$ are quadratic, one would have the standard Linear Quadratic Regulator familiar in the controls community, where $g^i(\cdot, \cdot)$ is the state and control trajectory penalties and $h^i(\cdot)$ the terminal state penalty. Also this formulation allows the possibility to penalize each vehicle differently. Without re-deriving, the modifications to DAO are described in the following.

Suppose the perturbation of $g^i(\mathbf{q}_k^i, \mathbf{u}_k^i)$ and $h^i(\mathbf{q}_K^i)$ can be written as

$$\begin{aligned} g^i(\mathbf{q}_k^i, \mathbf{u}_k^i)' &= \left(\frac{\partial g^i(\mathbf{q}_k^i, \mathbf{u}_k^i)}{\partial \mathbf{q}_k^i} \right)^T \mathbf{q}_k^{i'} + \left(\frac{\partial g^i(\mathbf{q}_k^i, \mathbf{u}_k^i)}{\partial \mathbf{u}_k^i} \right)^T \mathbf{u}_k^{i'}, \\ h^i(\mathbf{q}_K^i)' &= \left(\frac{dh^i(\mathbf{q}_K^i)}{d\mathbf{q}_K^i} \right)^T \mathbf{q}_K^{i'}. \end{aligned}$$

The local cost function gradient is now expressed as

$$\nabla_{\mathbf{u}_k^i} J = G^T \mathbf{r}_k^i + \frac{\partial g^i(\mathbf{q}_k^i, \mathbf{u}_k^i)}{\partial \mathbf{u}_k^i}. \quad (23)$$

The terminal state penalty simply changes the starting condition of $\mathbf{r}^i$, and the state trajectory penalty introduces an additional forcing to the $\mathbf{r}^i$ evolution equation. Equation (20)

now becomes

$$\begin{aligned} \mathcal{L}^*(\mathbf{r})_k^i = & \text{trace}((2AP_k A^T(H_k^T L_k^T - I)S_k L_k)_i \frac{dH_k^i}{d\mathbf{q}_k^i}) \\ & + \text{trace}((L_k^T S_k^T L_k)_{ii} \frac{dR_k^i}{d\mathbf{q}_k^i}) + \frac{\partial g^i(\mathbf{q}_k^i, \mathbf{u}_k^i)}{\partial \mathbf{q}_k^i}, \end{aligned} \quad (24a)$$

$$\mathbf{r}_{K-1}^i = \frac{dh^i(\mathbf{q}_K^i)}{d\mathbf{q}_K^i}. \quad (24b)$$

B. Nonlinearities

In section II we assumed linear models and used the Kalman Filter for the covariance update. For nonlinear models DAO can be modified to work with the Extended Kalman Filter.

Dealing with nonlinear vehicle model is simple, one needs to propagate and store the trajectory of each $\mathbf{q}_k^i$. During the vehicle adjoint propagation and cost function gradient evaluation, the linearized $F(\mathbf{q}_k^i, \mathbf{u}_k^i)$ and $G(\mathbf{q}_k^i, \mathbf{u}_k^i)$ are evaluated along the trajectory of $\mathbf{q}_k^i$ and $\mathbf{u}_k^i$.

However having a nonlinear field model introduces additional complexities. Namely, the Extended Kalman Filter requires linearized $A(\hat{\mathbf{f}}_k^+)$ and $B(\hat{\mathbf{f}}_k^+)$ evaluated using the updated estimate $\hat{\mathbf{f}}_k^+$ and the linearized measurement operator $H(\hat{\mathbf{f}}_k^-)$ evaluated using the predicted estimate $\hat{\mathbf{f}}_k^-$ — both estimates are unknown during the forecast because the knowledge of the future measurements is required.

Since the best estimate at current time is $\hat{\mathbf{f}}_0^+$, the best guess on the future measurements $\mathbf{y}$ at time $k = 1$ is obtained by forecasting $\hat{\mathbf{f}}_0^+$ to $\hat{\mathbf{f}}_1^-$ and apply the output operator H_1 to predict $\mathbf{y}_1$. Using this predicted measurement in (5b) implies $\hat{\mathbf{f}}_1^+ = \hat{\mathbf{f}}_1^-$. In general if $\hat{\mathbf{f}}_k^+$ is given, one could follow the same steps to show that $\hat{\mathbf{f}}_{k+1}^+ = \hat{\mathbf{f}}_{k+1}^-$. Thus by induction it is clear to see that an open-loop forecast from $\hat{\mathbf{f}}_0^+$ gives $\hat{\mathbf{f}}_k^+$ and $\hat{\mathbf{f}}_k^+$ for linearizing A , B , and H . Therefore with nonlinear models, one also needs to know $\hat{\mathbf{f}}_0$ in order to propagate and store P_k , $\mathbf{q}_k^i$, and $\hat{\mathbf{f}}_k$.

C. Multiple Vehicle and Sensor Types

In some instances, vehicles with different dynamical properties carrying different instruments are deployed. For example, in weather forecasting one aircraft may carry a Doppler Radar to give a global view of the weather system while several UAVs are carrying barometers, temperature sensors, and humidity sensors to provide pin-point measurements. By replacing F and G with F^i and G^i , allowing H^i and R^i to be different for each vehicle, and adjust the size of block matrices of $(L_k^T S_k^T L_k)_{ii}$ and $(2AP_k A^T(H_k^T L_k^T - I)S_k L_k)_i$ appropriately based on the size of R^i and H^i , DAO is capable in handling multiple vehicles with different vehicle dynamics and sensor types.

D. Routine Measurements

In some situations supplemental routine measurements are also available in the future. These measurements typically come from existing stationary sensor networks that makes routine measurements (e.g. sensor buoys), while some

other times these measurements come from non-controllable sources (i.e. wind data from boats). Routine measurements should be incorporated into the AO formulation to avoid redundant measurements. This is done by augmenting H_k and R_k such that

$$H_k \triangleq \begin{bmatrix} H_k^r & 0 \\ 0 & H_k^{AO} \end{bmatrix}, \quad R_k \triangleq \begin{bmatrix} R_k^r & 0 \\ 0 & R_k^{AO} \end{bmatrix}. \quad (25)$$

Note when performing perturbation analysis, the perturbation of H_k^r and R_k^r to $\mathbf{q}_k^i$ are zero since we cannot control the routine measurement placements.

V. EXAMPLE

Considered a square domain of size 100×100 . The target region is the entire domain ($T = I$) with $K = F = 300$ as the optimization and forecast time window (i.e. no forecast from P_K^+ to P_F). Two ($N = 2$) vehicles are available for AO purposes with each vehicle having the point-mass dynamics

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \underbrace{\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}}_{\mathbf{q}^i} + \underbrace{\begin{bmatrix} 0 \\ u_x \\ 0 \\ u_y \end{bmatrix}}_{\mathbf{u}^i}.$$

At present time $k = 0$ both vehicles start at $(0, 0)$ with initial control trajectories $\mathbf{u}_k^1$ and $\mathbf{u}_k^2$ that produce initial state trajectory $\mathbf{q}_k^1$ and $\mathbf{q}_k^2$. The initial vehicle waypoints sequence are shown in Fig. 2. For simplicity, $Q_u = I$ for the cost function, vehicle state trajectory and terminal state penalties are ignored, $\mathbf{f}$ is evolving linearly without disturbance ($W = 0$), each vehicle has the same linear vehicle dynamic and sensor, and the initial P_0 is diagonal. Three different P_0 are considered: the first is an uniform unit background variance, the second consists 3 Gaussian bumps of different amplitudes and attenuations in the domain, and the third is the sum of the first two. Fig. 3 illustrates the three different P_0 . A Kalman Filter is used for the covariance updates.

Suppose we have a specialized imaging system which takes a 360-degree-view image of the domain. The camera

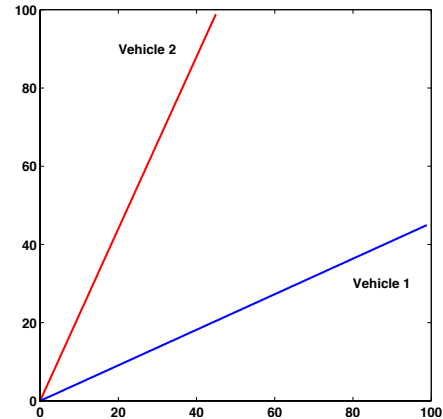


Fig. 2. Initial vehicle waypoints sequences created by initial control sequences $\mathbf{u}_k^1$ and $\mathbf{u}_k^2$. The waypoint sequences are symmetric about the domain diagonal.

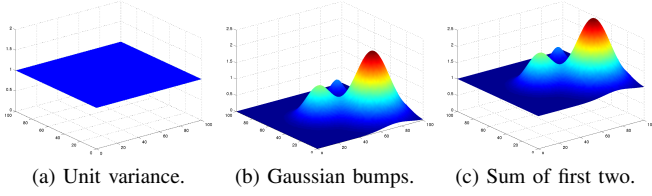


Fig. 3. The three different initial P_0 used in the example. (a) is an uniform unit variance. (b) is composed of 3 Gaussian bumps of different amplitude and attenuations. (c) is the sum of (a) and (b).

can “see” all n grid points in the domain; so for each vehicle $H_k^i = I$. Like real-world imaging systems, the image quality gets progressively worse as the subject is further away from the camera (poor pixel resolution). This is modeled with R_k^i , where the measurement noise variance at each grid point is proportional to the distance to the vehicle’s location squared

$$R_k^i = \frac{\beta}{2} \text{diag}((q_{1k}^i \mathbf{1} - \mathbf{z}_1) \cdot (q_{1k}^i \mathbf{1} - \mathbf{z}_1) + (q_{3k}^i \mathbf{1} - \mathbf{z}_2) \cdot (q_{3k}^i \mathbf{1} - \mathbf{z}_2) + \epsilon \mathbf{1}), \quad (26)$$

where β is a proportional constant, q_{1k}^i and q_{3k}^i are the first and third elements of $\mathbf{q}_k^i$, $\mathbf{z}_1$ and $\mathbf{z}_2$ are the discretized grid point locations, $\mathbf{1}$ is a vector of 1s of the appropriate size, and $(\cdot)$ denotes element-wise multiplication. The ϵ term serves two purposes: first it models the camera intrinsic digital background noise such as quantization error, and second it ensures the positive-definiteness of R_k^i required by the Kalman Filter. Simply put, R_k^i is a quadratic bowl in space centered at the vehicle i ’s position (x^i, y^i) , while β scales the bowl; thus a grid point measurement further away would have a greater measurement noise covariance. Note if a vehicle’s position coincides with a grid point, then for a small ϵ (this is true in general since quantization error is much smaller than the signal) we essentially have the state measurement at that grid point, and the estimation uncertainty at that grid point is driven to nearly zero *regardless of the initial uncertainty*. Two different proportionality constants, $\beta = 1$ and $\beta = 0.1$, are used to investigate the effect of β to the vehicle behaviors. One may interpret $\beta = 1$ as a narrow-view camera, and $\beta = 0.1$ as a wide-view camera. In both cases $\epsilon = 0.001$.

The domain is discretized with $n = 101 \times 101 = 10201$ grid points; hence $\mathbf{f}$ has 10201 state variables. Typically this poses a significant simulation challenge since the storage requirement for the covariance matrix P_k , which has size 10201×10201 , is large. This problem would not resolve even we leverage the symmetric property of P_k and only store the upper diagonal part of P . Furthermore propagating and updating P_k involves matrix multiplications, which requires significant computation time. However if we restrict the A matrix in (1) and P_0 to be diagonal, then the entire computation can be carried out by tracking only the diagonals of the relevant matrices. This is proved in the Appendix. We choose $A = \alpha I$ where $\alpha = \sqrt{1.01}$; thus if no measurement updates are performed, at the end of the time window P_0 would have grown by $1.01^{300} \approx 20$ times.

1) *Simulation Results:* The converged optimal vehicle waypoint sequences for all three P_0 in Fig. 3 with $\beta = 0.1$ are shown in Fig. 4.

As seen in Fig. 4a, when the uncertainty is uniform the resulting waypoint sequences take on an uniform-coverage approach. The symmetry in the converged waypoint sequences is due to the symmetric vehicle initial conditions. We have also ran the same simulation with non-symmetric vehicle initial conditions (not shown). The symmetry of the resulting waypoint sequences is destroyed while the uniform-coverage attribute remains. This shows the symmetric solution is not an unique solution.

When the initial uncertainty is just the three Gaussian bumps, the solution is completely different. Unless the vehicles are reasonably close to the Gaussian bumps, there will be little to no uncertainty reduction. Therefore we see the vehicles move straight into the Gaussian bumps in Fig. 4b. Also, comparing the inter-waypoint spacing shows the vehicles head toward the Gaussian bumps at a higher speed than when they are inside. As vehicle 2 leaves the main Gaussian bump and head toward the isolated small Gaussian bump, we see vehicle 2 speeds up once again in order to maximize time spent inside Gaussian bump.

Since the P_0 in Fig. 3c is the sum of Fig. 3a and 3b, the converged vehicle waypoint sequences embody characteristics from both. While vehicle 1 concentrates on reducing the lower-triangular region of the uniform uncertainty and the main bump, vehicle 2 focuses on the upper-triangular region and the isolated small bump. Reducing the second largest bump is divided among both vehicles.

The same experiments are repeated with $\beta = 1$, a narrow-view camera. For brevity only the solution for the P_0 in Fig. 3b is shown. The converged waypoint sequences are shown in Fig. 5. Comparing to Fig. 4b, we once again see both vehicles speed towards the uncertainties from the origin, and vehicle 2 speeds toward the isolated uncertainty after leaving the main uncertainty. However the waypoint sequences in Fig. 5 are more aggressive and closer to each other. This is the direct result of a bigger β . Intuitively, since a narrow-

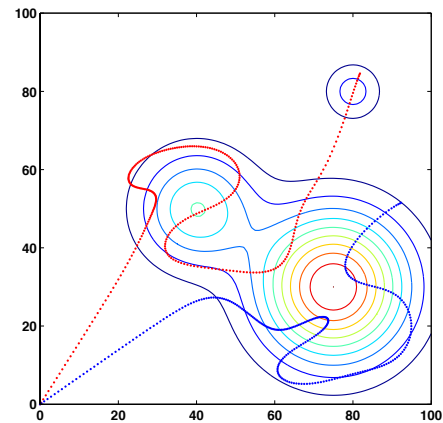


Fig. 5. Converged vehicle waypoint sequences for Gaussian bumps initial covariance with $\beta = 1$.

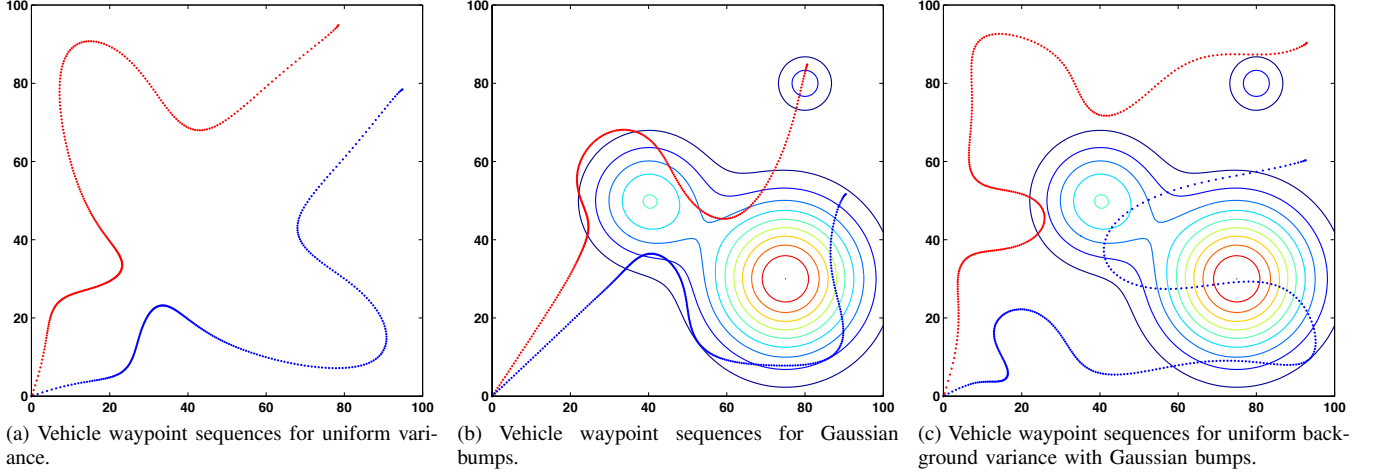


Fig. 4. Converged camera vehicle waypoint sequences for all three initial covariances, $\beta = 0.1$.

view camera “sees” less, more pronounced maneuvers are required to cover the domain; furthermore a narrower view means each vehicle needs to be placed closer together to achieve any sort of cooperation. Similar behavior also occurs in the other two P_0 experiments.

VI. CONCLUSION AND DISCUSSION

We have demonstrated DAO, our AO algorithm that combines features from sensitivity-based and uncertainty-based AO algorithm while incorporating vehicle dynamics. Generalizations such as nonlinear cost function, nonlinear model dynamics, and multiple vehicle and sensor types are shown to easily adopted into the algorithm. We demonstrated DAO using an example with cameras as sensors. Simulation results are sensible, suggesting the algorithm is able to balance between uncertainty reduction and vehicle control cost.

Storing and propagating the covariance matrix P is computationally intractable for large systems such as a discretized Navier-Stokes’ equation. For the camera example, we were able to bypass this issue by assuming a diagonal P_0 and a special A matrix in the evolution equation. However in general, such assumption is unrealistic. The weather and oceanic forecast community uses the Ensemble Kalman Filter [15], an approximation of the standard Kalman Filter, for data assimilation in large systems; to this end we are working on an approximation for DAO that uses the Ensemble Kalman Filter.

In this paper, the evolution of the sensed systems and vehicle are in discrete time with propagation time step coincides with measurement update time interval. This is not typical in real-world implementation, where measurements are taken far apart in time, but the propagation time steps of the sensed system and vehicles are small in order to capture small-scale dynamics. If the proposed algorithm is used out-of-the-box, one would need to discretize the systems with large time steps to match the measurement time intervals. Doing so would damp out small time scale dynamics and

possibly introduce large propagation errors or numerical instability. We are working on reformulating DAO in such a way such that the measurement interval is decoupled with the system and vehicle propagation.

APPENDIX

DIAGONAL P_k AND S_k FOR THE CAMERA EXAMPLE

For each camera sensor, $H_k^i = I$ and R_k^i is diagonal; thus the collective H_k and R_k are:

$$H_k = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix}, \quad R_k = \begin{bmatrix} R_k^1 & & 0 \\ & \ddots & \\ 0 & & R_k^N \end{bmatrix}.$$

With $A = \alpha I$ and $W = 0$ in the example, from (8a) the evolution of P_k^+ is simply

$$P_{k+1}^+ = \alpha^2 P_k^+ - \alpha^4 P_k^+ H_k^T (\alpha^2 H_k P_k^+ H_k^T + R_k)^{-1} H_k P_k^+.$$

Applying the matrix inversion lemma $(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1}$ to the inverse, where $A = R_k$, $B = H_k$, $C = H_k^T$, and $D = -(\alpha^2 P_k^+)^{-1}$, it becomes:

$$P_{k+1}^+ = \alpha^2 P_k^+ - \alpha^4 P_k^+ H_k^T R_k^{-1} H_k P_k^+ - \alpha^4 P_k^+ H_k^T R_k^{-1} H_k (-\alpha^2 P_k^+)^{-1} H_k^T R_k^{-1} H_k P_k^+.$$

By construction $H_k^T R_k^{-1} H_k = \sum_{i=1}^N (R_k^i)^{-1}$ is diagonal; thus $(-\alpha^2 P_k^+)^{-1} - H_k^T R_k^{-1} H_k$ is also diagonal if P_k^+ is diagonal. This guarantees P_{k+1}^+ to be diagonal. By induction P_k would remain diagonal as long as P_0 is diagonal. Similarly, it is easy to proof the adjoint variable S_k remains diagonal during the adjoint propagation if the starting condition S_K is diagonal.

REFERENCES

- [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] P. Bhatta, E. Fiorelli, N. E. Leonard, and I. Shulman, "Adaptive sampling using feedback control of an autonomous underwater glider fleet," in *Proceedings of 13th Int. Symp. on Unmanned Untethered Submersible Technology (UUST)*. Citeseer, August 2003.
- [6] C. Bishop, "Estimation theory and adaptive observing techniques," *Quarterly Journal of the Royal Meteorological Society*, 2000 (submitted).
- [7] P. F. Lermusiaux, "Adaptive modeling, adaptive data assimilation and adaptive sampling," *Physica D: Nonlinear Phenomena*, vol. 230, no. 1-2, pp. 172–196, June 2007.
- [8] R. Langland and G. Rohaly, "Adjoint-Based Targeting of Observations for FASTEX Cyclones," *Defense Technical Information Center*, 1996.
- [9] R. Buizza and A. Montani, "Targeting Observations Using Singular Vectors," *Journal of the Atmospheric Sciences*, vol. 56, no. 17, pp. 2965–2985, 1999.
- [10] N. Baker and R. Daley, "Observation and background adjoint sensitivity in the adaptive observation-targeting problem," *Quarterly Journal of the Royal Meteorological Society*, vol. 126, no. 565, pp. 1431–1454, 2000.
- [11] 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.
- [12] N. K. Yilmaz, C. Evangelinos, P. F. Lermusiaux, and N. M. Patrikalaks, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 522–537, Oct 2008.
- [13] S. J. Majumdar, C. H. Bishop, B. J. Etherton, and Z. Toth, "Adaptive Sampling with the Ensemble Transform Kalman Filter. Part II: Field Program Implementation," *Monthly Weather Review*, vol. 130, pp. 1356–1369, May 2002.
- [14] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Technical University of Denmark, oct 2008.
- [15] G. Evensen, "The Ensemble Kalman Filter: theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 2003.