Field Deployment of a Plume Monitoring UAV Flock

Matthew Silic and Kamran Mohseni, *Senior Member, IEEE*

***Abstract*—This letter describes and validates the robotic platform of an atmospheric plume monitoring system. The platform consists of a networked flock of unmanned aerial vehicles (UAVs) equipped with environmental sensors. The sensor flock forms an integral component of a dynamic data-driven application system (DDDAS) for plume monitoring. The goal of DDDAS is to dynamically incorporate data into a running simulation while simultaneously using the simulation to steer the measurement process. This letter takes a model-based approach to plume monitoring. From concentration measurements provided by the UAVs, a nonlinear online parameter estimator determines the model parameters. Based on the current knowledge of the model parameters, a hotspot identification routine directs the UAVs to information rich locations, or hotspots. The feasibility of deploying the testbed in uncontrolled outdoor environments is demonstrated with a flight test where three autonomous vehicles trace a simulated plume using simulated sensors.**

***Index Terms*—Multi-robot systems, aerial systems: applications, environment monitoring and management.**

I. Introduction

UNMANNED aerial vehicles (UAVs) continue to find traction in a diverse range of applications. One application with ties to homeland security and public health is that of concentration plume tracking [1]. Regardless of the cause of the plume—man-made or natural, accidental or malicious—the ability to monitor the plume in realtime is much needed by first responders and decision makers. UAVs are particularly well suited for this application: they can be deployed quickly, operate in hazardous environments, and obtain samples throughout a three-dimensional domain [2]. Furthermore, multiple UAVs can be deployed simultaneously, thereby harnessing the vast field of cooperative control.

A fundamental challenge is that of putting the sensor in the right place at the right time. This challenge has spurred the proliferation of innovative research by the community. For a recent, comprehensive review of the literature related to atmospheric dispersion and mobile sensing, the reader is referred to [3]. Only a selection of relevant work is presented here. In [4], a hexagonal lattice of mobile sensors is used to trace a plume to its source. The algorithm relies on mass flux measurements to ascertain the emitter location. In [5], a group of UAVs is tasked with tracking the boundary of an evolving plume, where the boundary of the plume is modeled using a splinegon. Other examples of boundary tracking algorithms can be found in [6], [7], in which an image segmentation algorithm is adapted to mobile agents. In [8], a moving plume source is tracked by a formation of UAVs. Multiple UAVs are needed in order to compute the concentration gradient, which is used by the estimator. In [9], a UAV assists a ground robot in estimating the emission rates of multiple, overlapping plumes. The UAV is used to resolve the ambiguity associated with the inverse problem. Information-theoretic approaches to the plume tracking problem can be found in [10], [11]. A fully-decentralized and scalable approach to environmental modeling and mobility control is presented in [12]. The algorithms in the previously cited works have been validated at the simulation level only. Hardware implementations are less common. Examples of hardware implementation can be found in [13]–[17]. Ground robots are used in [13]–[15] and a UAV is used in [16], [17].

Our method of plume monitoring is to use a dynamic data driven application system (DDDAS) [18]. The goal of a DDDAS is to dynamically incorporate data into a running simulation while simultaneously using the simulation to steer the measurement process. A defining characteristic of DDDAS is the interaction between the virtual domain (where the simulation exists) and the physical domain (where the measurements take place). The DDDAS approach is quite general and can be applied to many phenomena for which good simulation models are available. Our primary application is environmental monitoring, namely concentration puff/plume tracking using aerial vehicles equipped with environmental sensors [19]–[22].

An identifying feature of our DDDAS is the use of multiple, networked UAVs for sensor mobility. Additionally, the UAVs must work in parallel with a simulation. The objective of this work is to validate the hardware component of our plume monitoring system through a field deployment. A meaningful validation requires multiple UAVs to be airborne simultaneously, flying unscripted trajectories, and sending and receiving data that is both useful and necessary. The value of conducting field tests is that real-world effects must be confronted and overcome. Simulations, while necessary, may underestimate or overlook various hindrances: non-ideal vehicle dynamics, computational overhead, communication limitations, atmospheric

Manuscript received September 10, 2018; accepted December 24, 2018. Date of publication January 16, 2019; date of current version February 4, 2019. This letter was recommended for publication by Associate Editor M. A. Hsieh and Editor N. Y. Chong upon evaluation of the reviewers' comments. This work was supported in part by the AFOSR, in part by the ONR, and in part by the NSF. *(Corresponding author: Kamran Mohseni.)*

M. Silic is with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: msilic@ufl.edu).

K. Mohseni is with the Department of Mechanical and Aerospace Engineering, Department of Electrical and Computer Engineering, and Institute for Networked Autonomous Systems, University of Florida, Gainesville, FL 32611 USA (e-mail: mohseni@ufl.edu).

This letter has supplemental downloadable multimedia material available at http://ieeexplore.ieee.org, provided by the authors. The Supplementary Materials contain a video from telemetry downlinked by the UAVs. This material is 1.4 MB in size.

Digital Object Identifier 10.1109/LRA.2019.2893420

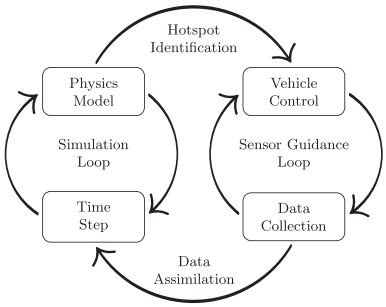

Fig. 1. Schematic of the online plume monitoring framework. The two main components of the system are a sensor guidance loop and a simulation loop. The two loops interact through data assimilation and hotspot identification algorithms.

Fig. 2. (a) The complete unmanned aerial system includes the vehicle, base-station laptop, base-station radio, and RC transmitter. (b) A close-up of the custom-built autopilot.

disturbances, etc. This letter contributes to the literature by showing how real, fixed-wing UAVs respond in a coordinated, plume monitoring scenario.

This letter also details the virtual domain that complements the hardware component. The virtual domain takes a model based approach. A nonlinear online parameter estimator determines the model parameters. Based on the current knowledge of the model parameters, a hotspot identification routine directs the UAVs to information rich locations. It is noted that the virtual domain designed herein may be replaced by a more sophisticated algorithm, without significant modifications to the physical domain.

The remainder of this manuscript is organized as follows. Section II details the elements of the plume monitoring framework. Section III presents the results of a flight test where three autonomous vehicles trace a simulated plume. Section IV states our conclusions and provides a direction for future work.

II. Methods

The framework for our plume monitoring system is shown in Fig. 1. The framework is characterized by two loops running in parallel. The simulation loop marches forward in time the advection-diffusion model of the concentration. The sensor guidance loop directs the mobile platforms to measurement locations. The two loops interact through hotspot identification and data assimilation. Both loops benefit from this interaction. The realism of the simulation is improved by injecting experimental data. The selection of future measurement locations is improved by consulting the simulation. In the remainder of this section, the primary components of the system are described. Specifically, the UAV flock, the physics model, the data assimilation algorithm and the hotspot identification algorithm are detailed.

A. UAV Flock

The flock comprises multiple, networked UAVs equipped with environmental sensors. Fig. 2a shows a single UAV along with its support equipment—a base station laptop, a base station radio, and a RC transmitter. The aerial vehicle is a tailless delta wing that is driven by a pusher propeller and actuated by two elevons. As a fixed-wing vehicle, the delta wing generates lift by maintaining forward velocity. A drawback of using fixed-wing vehicles is that they cannot hover like rotary-wing vehicles. The delta wing has been used for a variety of missions, including wireless communication characterization [23], cooperative control [19] and atmospheric sensing [24].

The autopilot, shown in Fig. 2b, was made in-house by our research group [25]. The autopilot is a collection of electronic devices, namely a GPS receiver, a radio transceiver, a barometer, an atmospheric sensor and a 9-axis IMU. The atmospheric sensor measures the ambient temperature and humidity. Processing is done using a 16 bit, 140 MHz microcontroller. Additionally, an Xbee radio can plug into headers on the underside of the board. Along the perimeter of the board is a header that breaks out all the unassigned pins of the microcontroller. Additional sensors can be wired into this header for mission-specific customization of the autopilot.

The autopilot firmware handles guidance, estimation, and control. The control system comprises a longitudinal and lateral control system. State estimation is effected through a nonlinear complementary filter [26]. Guidance is effected through vector fields designed from Lyapunov functions [27]. The specific vector field used in this study brings the vehicle into a circular loiter pattern. Complementing the autopilot firmware is a Ground Control Station (GCS) application built in MATLAB. The GCS application runs on the base station laptop. It interfaces with the base station radio and provides the flight supervisor with a continuous readout of the aircraft's telemetry. From the base station, the flight supervisor can adjust the loiter circle parameters.

B. Physics Model

Atmospheric dispersion is described by the advection-diffusion equation, which is a second-order, parabolic, partial differential equation [28]. In general, the advection-diffusion equation must be solved numerically. However, analytic solutions may be derived under simplifying assumptions. The simplest exact solution is called the *Gaussian plume*. The Gaussian plume assumes the contaminant is released at a constant rate from a single point into a unidirectional wind field. Additionally, the solution assumes the diffusivity is constant and the domain is infinite. Let u be the wind speed [m/s], Q be the emission rate [kg/s], and K be the diffusion constant [m^2/s]. The Gaussian plume solution in two-dimensions is given by

$$\psi = \frac{Q}{4\pi K x} \exp\left(\frac{-uy^2}{4Kx}\right), \tag{1}$$

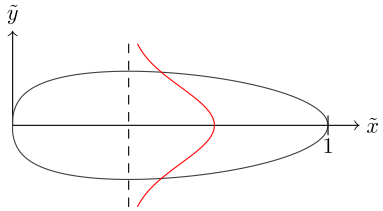

Fig. 3. The boundary of the plume is defined by a fixed concentration value. In 2D, the boundary defines a closed path. The boundary shown here corresponds to a Péclet number of 50. A concentration profile for a lateral cross section is also shown.

where x denotes the downwind direction and y denotes the crosswind direction. For a constant x, the concentration traces out a Gaussian distribution. The Gaussian plume model has been applied to a variety of applications, including industrial emissions [29], volcanic eruptions [30], pollen dispersal [31], and odor propagation [32].

The Gaussian plume lacks a definitive boundary; instead, the concentration asymptotically approaches zero as one travels away from the plume source. However, the boundary may be defined by a fixed concentration value, Ψ. For example, Ψ could be the measurement threshold of the sensor or an exposure limit recommended by health officials. An equation for the plume boundary can be obtained by solving $\psi(x, y) = \Psi$. The solution is given by

$$y^2 = -\frac{4Kx}{u} \ln\left(\frac{4\pi K\Psi}{Q} x\right). \tag{2}$$

A characteristic length that appears in the previous equation is $L \equiv Q/(4\pi K\Psi)$. Defining the nondimensional coordinates $\tilde{x} \equiv x/L$ and $\tilde{y} \equiv y/L$, the boundary equation given by (2) is expressed as

$$\tilde{y}^2 = -\frac{4}{P_e} \tilde{x} \ln \tilde{x}, \tag{3}$$

where $P_e \equiv uL/K$ is the nondimensional Péclet number, which quantifies the relative strength of the advection and diffusion terms. The plume boundary is shown in Fig. 3, along with a nondimensional concentration profile. The nondimensional concentration is defined as $\tilde{\psi} \equiv \psi/\Psi$ and is given by

$$\tilde{\psi} = \frac{1}{\tilde{x}} \exp\left(-\frac{P_e}{4} \frac{\tilde{y}^2}{\tilde{x}}\right). \tag{4}$$

The nondimensional concentration is used in the remainder of this work. The nondimensional coordinates $(\tilde{x}, \tilde{y})$ are defined with respect to the wind frame. Let χ represent the angle between the wind vector and the x-axis of the ground frame. Let (x_s, y_s) represent the source location in the ground frame. To construct the wind frame, we translate the origin of the ground frame to (x_s, y_s) and then rotate the transplanted frame χ radians about its $+z$-axis. Let (x, y) represent the sensor location in the ground frame. The nondimensional coordinates $(\tilde{x}, \tilde{y})$ are given by

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} \cos\chi & \sin\chi \\ -\sin\chi & \cos\chi \end{bmatrix} \begin{bmatrix} x - x_s \\ y - y_s \end{bmatrix}. \tag{5}$$

Because of atmospheric turbulence, a snapshot of a real plume will never resemble a Gaussian plume. As such, our approach is unable to reconstruct the concentration field. Fortunately, our objective is to determine a limited set of time-invariant parameters (source-location and plume intensity) from a set of noisy observations. Our assumption is that the mean field of the concentration can be modeled by the Gaussian plume. In reality, the concentration will fluctuate around the mean field. However, if we incorporate measurements over a long enough time scale, we can reject the variability associated with the fluctuations. Furthermore, the observations need not be fixed to a single location, because the parameters to be estimated are associated with the entire field, not just a single point.

C. Data Assimilation

The subsequent discussion refers to (4) and (5) as "the model". In this case, the model generates a single output, $\tilde{\psi}$, given a sensor location, (x, y), and values for the various parameters, $\{x_s, y_s, L, \chi, P_e\}$. We assume χ and P_e are known and collect the remaining unknown parameters in a parameter vector $\theta = [x_s, y_s, L]^T$. The Jacobian of the model with respect to θ is denoted by X and is given by

$$X^T = \frac{\tilde{\psi}}{4\tilde{x}L} \begin{bmatrix} \cos\chi & -\sin\chi \\ \sin\chi & \cos\chi \\ \tilde{x} & \tilde{y} \end{bmatrix} \begin{bmatrix} 4 - P_e\tilde{y}^2/\tilde{x} \\ 2P_e\tilde{y} \end{bmatrix}.$$

The row vector X is also called the *measurable information vector* or the *sensitivity matrix*. For a model that is linear in the parameters, $\tilde{\psi} = X\theta$. However, our model is nonlinear in the parameters.

The data assimilation algorithm uses a recursive least squares (RLS) algorithm with exponential forgetting. The RLS algorithm is given by [33]

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \frac{P_{k-1}X_k^T}{\lambda + X_k P_{k-1} X_k^T} \left(y_k - X_k\hat{\theta}_{k-1}\right) \tag{6a}$$

$$P_k = \frac{1}{\lambda}\left(P_{k-1} - \frac{P_{k-1}X_k^T X_k P_{k-1}}{\lambda + X_k P_{k-1} X_k^T}\right) \tag{6b}$$

where $\hat{\theta}$ is the parameter vector estimate, P is the covariance matrix of the parameter vector, y is the measurement, and $\lambda \in [0, 1]$ is an exponential forgetting factor. Setting $\lambda = 1$ in (6) gives the standard RLS algorithm. The exponential forgetting factor prevents the covariance matrix, P, from going to zero. Should P go to zero, the estimator becomes insensitive, which may result in a steady-state error in $\hat{\theta}$.

Although the RLS algorithm is designed for linear models, it can be extended to nonlinear models with the appropriate modification. The modification is motivated by noting that (6a) has the form $\hat{\theta}_k = h(\hat{\theta}_{k-1})$. For linear models, this expression is exact, because X_k is independent of $\hat{\theta}_k$. For nonlinear models, the expression may be approximately solved using fixed-point iteration. The iteration continues until there is a negligible change in the estimated parameters. Although the modified algorithm can handle nonlinear models, convergence is never guaranteed. For convergence to occur, the starting value of the parameter

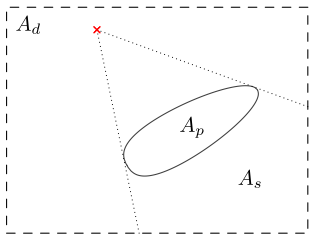

Fig. 4. The probability of encountering the plume using the random search is $(A_p + A_s)/A_d$, where A_d is the area of the domain, A_p is the area of the plume, and A_s is the region behind the plume as viewed from the current location of the agent.

estimate must be reasonably close to the minima of the sum of squares function and the minima must be reasonably well-defined.

D. Hotspot Identification

A caveat to the nonlinear parameter estimator is that the initial estimates have to be reasonably close to the true values in order for the parameters to converge. In practice, this means that the estimated plume and the true plume must overlap. Otherwise, the parameter estimates will converge to the trivial solution (i.e. no plume). Intuitively, this makes sense: the trivial solution corresponds to a nil sensor reading, which occurs when the sensor is outside the plume. To ensure the estimated plume and the true plume overlap *before* the estimator is engaged, the hotspot identification algorithm is split into two phases: an *exploration* phase and an *estimation* phase. In the exploration phase, the hotspots are drawn from a uniform distribution bounded by the experimental domain. The exploration phase continues until a non-zero concentration measurement is made. The initial estimate of the source location is set to the location at which the non-zero measurement was made. This ensures that the initial guess of the source location resides in the plume, thus preventing the estimator from converging to the trivial solution.

The following argument shows that, during the exploration phase, the probability of encountering the plume linearly converges to one. Let A_d be the area of the domain and let A_p be the area enclosing the sensible region of the plume (Fig. 4). The probability that a randomly selected hotspot, $w \in A_d$, lies outside the plume is given by γ where $\gamma \equiv 1 - A_p/A_d$. Since $A_p < A_d$, we know that $\gamma \in (0, 1)$. The probability of missing the plume after n casts is γ^n. A simple analysis shows the sequence $\{\gamma^n\}$ converges linearly to 0 with a convergence rate of γ. Linear convergence, though far from desirable, can be improved by making γ as small as possible. The convergence rate may be decreased by employing more agents in the search. If N agents are engaged in the search, the convergence rate becomes

$$\gamma = (1 - A_p/A_d)^N. \tag{7}$$

In actuality, we expect the convergence to be faster because the agents collect measurements en route to the hotspot. For a visual proof, suppose the position of the agent is given by the red "x" in Fig. 4. Assuming an agent beelines to the hotspot from its current position, the probability of encountering the plume becomes $(A_p + A_s)/A_d$, where A_s is the "shadow" region behind the plume as viewed from the current position of the agent. Because the probability of encountering the plume is greater than A_p/A_d, (7) gives an upper bound for the convergence rate.

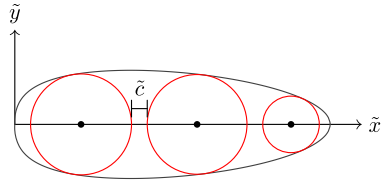

Fig. 5. Loiter circle distribution for three hotspot locations. This distribution corresponds to $P_e = 50$ and $\tilde{c} = 0.05$.

After the plume has been detected and the parameter estimates have been initialized, the algorithm switches to the estimation phase. During the estimation phase, the hotspot identification algorithm places the loiter circles within the plume and scales the loiter circles such that they touch the boundary of the plume, as shown in Fig. 5. To prevent vehicle collision, the loiter circles are separated by a distance $\tilde{c}$. However, even if the loiter circles intersect, it is unlikely for the vehicles to collide, since the vehicles occupy a single point at any given time.

Here, we briefly describe how to determine the radii and centers for the distribution shown in Fig. 5. Let $\tilde{r}_j$ be the radius of the jth loiter circle and let $\tilde{x}_j$ be location of the corresponding center. The radius and center must satisfy the following set of equations:

$$\tilde{r}_j^2 = -\frac{4}{P_e}\tilde{x}_j^* \ln \tilde{x}_j^* + \left(\tilde{x}_j - \tilde{x}_j^*\right)^2, \tag{8a}$$

$$\tilde{x}_j = \tilde{x}_{j-1} + \tilde{r}_j + \tilde{r}_{j-1} + \tilde{c}, \tag{8b}$$

$$P_e\left(\tilde{x}_j^* - \tilde{x}_j\right) = 2\left(\ln \tilde{x}_j^* + 1\right). \tag{8c}$$

These equations are valid for $j = 1, \ldots, n$ provided that $\tilde{x}_0 = \tilde{r}_0 = 0$. In these equations, $\tilde{x}^*$ is the $\tilde{x}$ coordinate of the point of contact between the loiter circle and the plume boundary. Equation (8c) requires the radius to be perpendicular to the boundary at the point of contact. The previous equations can be solved numerically with a root finding algorithm.

The algorithm adapts to uncertainty in the source location by adding a random component to the deterministic hotspot locations. The random component is normally distributed, zero mean, with a variance proportional to the variance of the source location estimate. Thus, when the variance in the source location estimate is large (which occurs when the source location is poorly known) the hotspot locations will essentially be random. However, as the variance approaches zero, the randomized hotspot locations converge to the deterministic locations provided by the algorithm above.

III. Results

Three delta wing UAVs were used for this test. The same software was running on each autopilot. Table I lists the values assigned to the various parameters. The virtual plume is given by the Gaussian plume model. To simulate the sensor readings, the

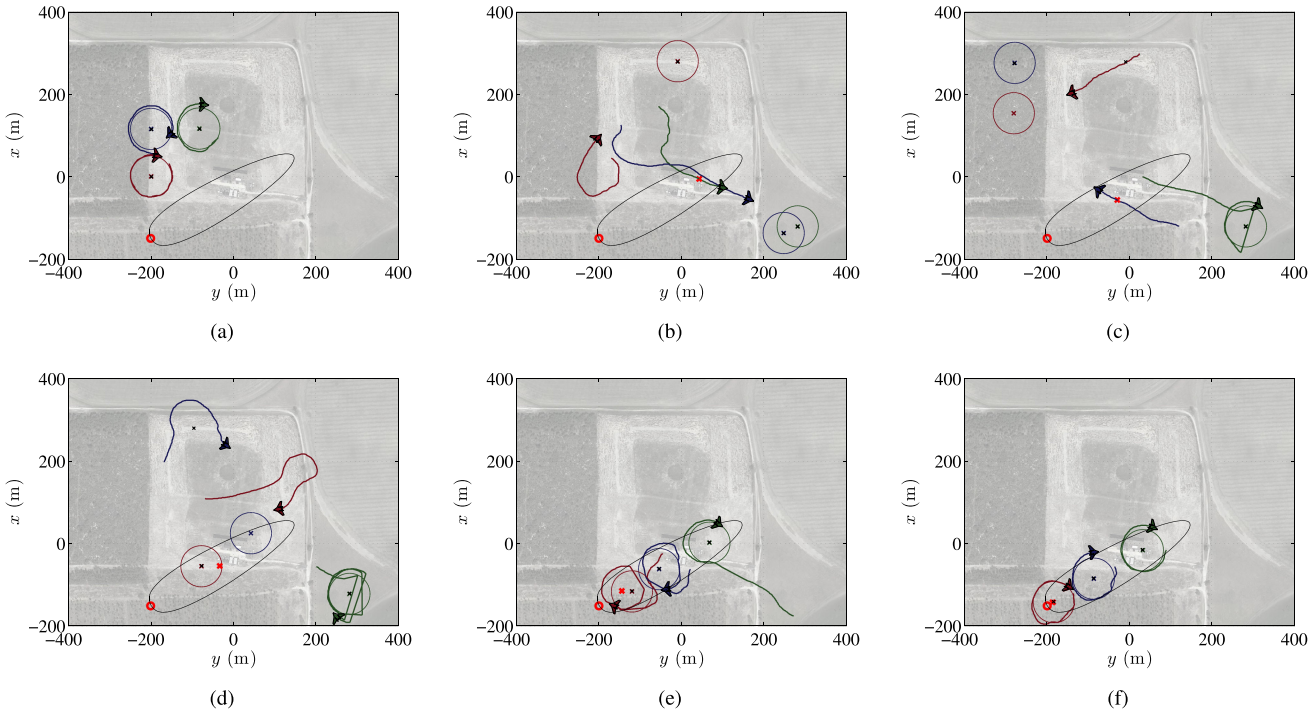

Fig. 6. Snapshots from a three vehicle plume monitoring experiment. Initially, the vehicles are loitering outside the plume (a). The plume monitoring algorithm is initiated and the vehicles begin exploring the domain in a random fashion (b). The estimated location of the plume source (red "x") is set to the location of the highest concentration reading (c). With the source location estimate initialized, the estimator is engaged (d). As the uncertainty in the estimated parameters decreases, the hotspots converge to uniform distribution along the plume centerline (e). The experiment is terminated when the plume source is localized and the loiter circles stop shifting (f).

TABLE I
MODEL AND ESTIMATION PARAMETERS

Parameter	Value	Units
P_e	100	-
χ	$\pi/3$	rad
L	400	m
λ	0.999	-
(x_s, y_s)	$(-150, -200)$	m
A_d	480×10^3	m^2
A_p	92.6×10^3	m^2
$\hat{\theta}_0$	$[999, 999, 600]^T$	m
P_0	$\text{diag}([10^3, 10^3, 10^2])$	m^2

vehicles downlink their GPS coordinates to the ground station. On the ground station, these global positions are transformed into local coordinates, which are feed into the plume model with the parameters set to their true values—values that are unknown to the parameter estimator. To simulate sensor noise and modeling errors, white noise with a standard deviation of 0.2Ψ is added to the model output.

The three autopilots and the ground station all communicate on the same network using the Xbee radios. To avoid packet collision, the autopilots communicate in a cyclic fashion. During a cycle, each autopilot is assigned a timeslot in which to send data through the network. The hotspot identification algorithm is set to uplink an updated hotspot every 10 seconds. Because the hotspots are uplinked cyclically, each vehicle receives a new loiter circle every 30 seconds. This interim allows the vehicle to converge to a given loiter circle before switching to a new loiter circle. Furthermore, it prevents the network from being flooded with communication packets. Because the ground station coordinates all the agents, no information is exchanged between the agents.

The flight test presented here was conducted on July 23, 2018 at a RC flying field outside of Gainesville, FL. Snapshots from the flight test are shown in Fig. 6; these snapshots were reconstructed from telemetry downlinked by the UAVs. The vehicles were hand launched and brought to an altitude of 100 meters manually. The autopilots were engaged and the vehicles converged to loiter circles set by the flight supervisor (Fig. 6a). Following this, the plume monitoring algorithm was initiated. The algorithm started in the exploration phase in which the hotspots were generated randomly within the experimental domain (Figs. 6b and 6c). During the exploration phase, the estimated source location was set to the location of the highest concentration reading obtained thus far. Once a reasonable guess for the source location was obtained, the algorithm was switched to the estimation phase (Figs. 6d and 6e). In this phase, the hotspots converged to their ideal locations as the uncertainty in the estimated parameters decreased. Fig. 6f shows the final distribution of the vehicles. The algorithm successfully determined the parameters and the hotspots are distributed along the plume centerline.

In Figs. 6c and 6d, straight lines appear in the groundtrack of the green vehicle (the vehicle in the lower right corner). The straight lines are artifacts of dropped communication packets. It was later determined that a low-lying structure was obstructing

TABLE II
NETWORK STATISTICS

Vehicle	Packets Sent	Packets Dropped	Avg. RSSI (dBm)
1	3369	11.0%	-74
2	3368	10.2%	-73
3	3381	26.5%	-77

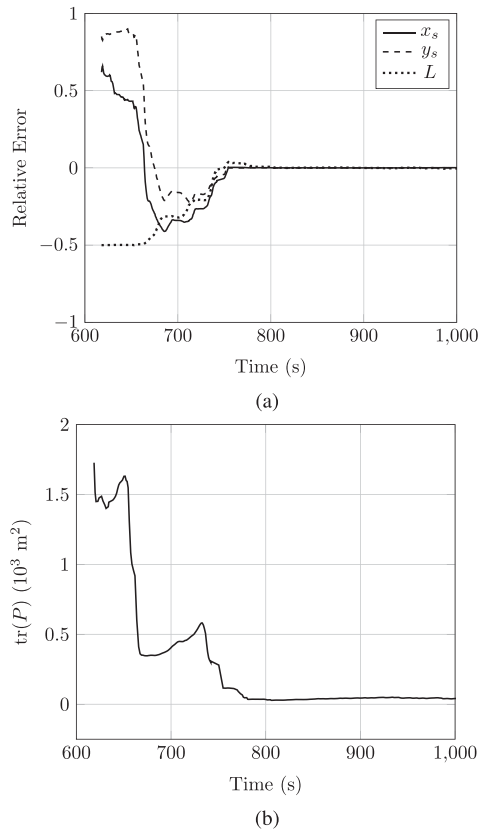

Fig. 7. (a) Relative error in the parameter estimates during the estimation phase of the experiment. (b) Trace of the covariance matrix during the estimation phase of the experiment.

the line-of-sight between the aircraft and the ground station, causing the dropped packets. Table II shows various statistics related to the wireless network. We see that vehicle 3—the green vehicle in Fig. 6—has the most dropped packets and the lowest average receiver signal strength indicator (RSSI).

The system performance is evaluated by examining the relative error of the parameter estimates (Fig. 7a). To compute the relative errors, the true parameter vector, θ, is needed. Because the plume is virtual, θ is known. Only in post-processing is θ directly used; otherwise, it is kept hidden from the plume monitoring system. If the plume was real, θ would be unknown and an independent measurement would be needed to verify the system. Such independent measurements could be provided by aerial photography [30] or dust fall jars [28].

The random search algorithm locates the plume after one cast. Given the areas of the plume and the domain, the probability of locating the plume after one cast is 19%. After 70 seconds of searching, the exploration phase brings the relative error in the source location estimate to under 100%. Armed with a reasonable initial guess for the source location, the RLS algorithm reduces the relative errors to under 0.1% in approximately 200 seconds (Fig. 7a). The trace of the covariance matrix is shown in Fig. 7b. Because of the forgetting factor, the covariance matrix does not decrease to zero, but maintains a small positive value. Should the covariance matrix go completely to zero, the estimator becomes insensitive to errors, which could introduce steady-state errors in the parameter estimates.

IV. CONCLUSION

In this letter, we have experimentally validated the sensor mobility component of a plume monitoring DDDAS. Sensor mobility is accomplished using multiple, networked UAVs that work in conjunction with a virtual domain. In general, the virtual domain is a simulation, the realism of which is improved by the assimilation of experimental data. In this experiment, a physical plume is replaced by the Gaussian plume model and the simulation is restricted to just the data assimilation algorithm. However, the sensor mobilizers are fully implemented. That is, three autonomous vehicles trace a simulated plume in an uncontrolled outdoor environment. Using a wireless network, the vehicles exchange data with a base station computer, which runs the data assimilation and hotspot identification algorithms in realtime. The plume source is localized, the emission rate is estimated, and the vehicles distribute themselves along the plume centerline, all in an unscripted manner. As demonstrated by the flight test, the parameter estimator is able to reduce the uncertainty in the estimated parameters.

From this experiment, three key takeaways stand out. The first takeaway is the necessity of a good initial guess for the parameter estimate when attempting nonlinear parameter estimation. The exploration phase was added so that reasonable initial guesses could be obtained autonomously, thus improving the usefulness of the system. The second takeaway is the necessity of a stable autopilot. During the exploration phase, it is not uncommon for the hotspot identification algorithm to send the vehicle out to the limits of the pilot's visual range. These extreme ranges force a greater reliance on the autopilot. The third and final takeaway is the necessity of reliable communication links. The links are important because, through them, the virtual and physical domains interact. The impact of communication dropouts was confirmed by the flight test. Because of shadowing, a vehicle missed two hotspot updates. Furthermore, when the vehicle was flying through the dead-zone, the sensor readings were lost, effectively disabling an otherwise functioning agent. Fortunately, because the system uses multiple agents, it is robust to the loss of a single agent.

The experience, and the improvements acquired throughout the implementation of this experiment, will undoubted be invaluable for future, more sophisticated DDDAS implementations, which are currently in the planning stages. The next test will feature five vehicles, obstacle avoidance and an online advection-diffusion solver. Additionally, we hope to use an exploration algorithm that gives super-linear convergence.

ACKNOWLEDGMENT

The authors would like to thank Y. Mitikiri, R. O'Donnell, and E. Juarez-Garcia for their indispensable help with the flight tests.

REFERENCES

[1] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Automat. Mag.*, vol. 19, no. 1, pp. 24–39, Mar. 2012.

[2] J. Elston, B. Argrow, M. Stachura, D. Weibel, D. Lawrence, and D. Pope, "Overview of small fixed-wing unmanned aircraft for meteorological sampling," *J. Atmospheric Ocean. Technol.*, vol. 32, no. 1, pp. 97–115, 2015.

[3] M. Hutchinson, H. Oh, and W. H. Chen, "A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors," *Inf. Fusion*, vol. 36, pp. 130–148, 2017.

[4] D. Zarzhitsky, D. F. Spears, and W. M. Spears, "Swarms for chemical plume tracing," in *Proc. IEEE Swarm Intell. Symp.*, Pasadena, CA, USA, Jun. 8–10, 2005, pp. 249–256.

[5] B. A. White, A. Tsourdos, I. Ashokaraj, S. Subchan, and R. Zbikowski, "Contaminant cloud boundary monitoring using network of UAV sensors," *IEEE Sensors J.*, vol. 8, no. 10, pp. 1681–1692, Oct. 2008.

[6] I. Triandaf and I. Schwartz, "A collective motion algorithm for tracking time-dependent boundaries," *Math. Comput. Simul.*, vol. 70, no. 4, pp. 187–202, 2005.

[7] D. Marthaler and A. Bertozzi, "Tracking environmental level sets with autonomous vehicles," in *Recent Developments in Cooperative Control and Optimization*, S. Butenko, R. Murphey, and P. Pardalos, Eds. Berlin, Germany: Springer, 2004, pp. 317–332.

[8] T. Egorova, M. A. Demetriou, and N. A. Gatsonis, "Plume estimation using static and dynamic formations of unmanned aerial vehicles," in *Proc. IEEE Conf. Decis. Control*, Las Vegas, NV, USA, Dec. 12–14, 2016, pp. 2270–2275.

[9] J. R. Gemerek, S. Ferrari, and J. D. Albertson, "Fugitive gas emission rate estimation using multiple heterogeneous mobile sensors," in *Proc. IEEE Int. Symp. Olfaction Electron. Nose*, Montreal, Canada, May 28–31, 2017, pp. 1–3.

[10] M. Vergassola, E. Villermaux, and B. Shraiman, "'Infotaxis' as a strategy for searching without gradients," *Nature*, vol. 445, pp. 406–409, 2007.

[11] H. Hajieghrary, M. A. Hsieh, and I. B. Schwartz, "Multi-agent search for source localization in a turbulent medium," *Phys. Lett. A*, vol. 380, no. 20, pp. 1698–1705, 2016.

[12] K. Lynch, I. Schwartz, P. Yang, and R. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 710–724, Jun. 2008.

[13] J. Soares, A. Aguiar, A. Pascoal, and A. Martinoli, "A distributed formation-based odor source localization algorithm-design, implementation, and wind tunnel evaluation," in *Proc. IEEE Int. Conf. Robot. Automat.*, Seattle, WA, USA, May 26–30, 2015, pp. 1830–1836.

[14] H. Hajieghrary, D. Mox, and M. A. Hsieh, "Information theoretic source seeking strategies for multiagent plume tracking in turbulent fields," *J. Mar. Sci. Eng.*, vol. 5, no. 1, pp. 1–20, 2017, Art no. 3.

[15] R. Khodayi-Mehr, W. Aquino, and M. Zavlanos, "Model-based active source identification in complex environments," 2017, arXiv:1706.01603.

[16] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Autonomous gas-sensitive microdrone: Wind vector estimation and gas distribution mapping," *IEEE Robot. Automat. Mag.*, vol. 19, no. 1, pp. 50–61, Mar. 2012.

[17] P. P. Neumann, H. Kohlhoff, D. Hüllmann, A. J. Lilienthal, and M. Kluge, "Bringing mobile robot olfaction to the next dimension—UAV-based remote sensing of gas clouds and source localization," in *Proc. IEEE Int. Conf. Robot. Automat.*, Singapore, 2017, pp. 3910–3916.

[18] F. Darema, "Dynamic data driven applications systems: A new paradigm for application simulations and measurements," in *Proc. Int. Conf. Comput. Sci.*, Kraków, Poland, Jun. 6-9, 2004, pp. 662–669. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24688-6

[19] B. Hodgkinson, D. Lipinski, L. Peng, and K. Mohseni, "Cooperative control using data-driven feedback for mobile sensors," in *Proc. IEEE Int. Conf. Robot. Automat.*, Karlsruhe, Germany, May 6–10, 2013, pp. 772–777. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2013.6630660

[20] L. Peng and K. Mohseni, "Sensor driven feedback for puff estimation using unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, Orlando, FL, USA, May 27–30, 2014, pp. 562–569. [Online]. Available: http://dx.doi.org/10.1109/ICUAS.2014.6842298

[21] L. Peng, D. Lipinski, and K. Mohseni, "Dynamic data driven application system for plume estimation using UAVs," *J. Intell. Robot. Syst.*, vol. 74, no. 1-2, pp. 421–436, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10846-013-9964-x

[22] L. Peng, M. Silic, and K. Mohseni, "A DDDAS plume monitoring system with reduced Kalman filter," in *Proc. Int. Conf. Comput. Sci.*, 2015, vol. 51, pp. 2533–2542.

[23] A. Shaw and K. Mohseni, "A fluid dynamic based coordination of a wireless sensor network of unmanned aerial vehicles: 3-D simulation and wireless communication characterization," *IEEE Sensors J.*, vol. 11, no. 3, pp. 722–736, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1109/JSEN.2010.2064294

[24] D. A. Lawrence and B. B. Balsley, "High-resolution atmospheric sensing of multiple atmospheric variables using the datahawk small airborne measurement system," *J. Atmospheric Ocean. Technol.*, vol. 30, no. 10, pp. 2352–2366, 2013.

[25] A. Bingler and K. Mohseni, "Dual radio autopilot system for lightweight, swarming micro/miniature aerial vehicles," *J. Aerosp. Inf. Syst.*, vol. 14, no. 5, pp. 293–305, 2017. [Online]. Available: https://doi.org/10.2514/1.I010445

[26] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, Jun. 2008.

[27] D. A. Lawrence, E. W. Frew, and W. J. Pisano, "Lyapunov vector fields for autonomous unmanned aircraft flight control," *J. Guid., Control, Dyn.*, vol. 31, no. 5, pp. 1220–1229, 2008.

[28] J. M. Stockie, "The mathematics of atmospheric dispersion modeling," *SIAM Rev.*, vol. 53, no. 2, pp. 349–372, 2011.

[29] S. Baroutian, A. Mohebbi, and A. S. Goharrizi, "Measuring and modeling particulate dispersion: A case study of Kerman cement plant," *J. Hazardous Mater.*, vol. 136, no. 3, pp. 468–474, 2006.

[30] R. Turner and T. Hurst, "Factors influencing volcanic ash dispersal from the 1995 and 1996 eruptions of Mount Ruapehu, New Zealand," *J. Appl. Meteorol.*, vol. 40, no. 1, pp. 56–69, 2001.

[31] C. Loos, R. Seppelt, S. Meier-Bethke, J. Schiemann, and O. Richter, "Spatially explicit modelling of transgenic maize pollen dispersal and cross-pollination," *J. Theor. Biol.*, vol. 225, no. 2, pp. 241–255, 2003.

[32] R. Smith, "Dispersion of odours from ground level agricultural sources," *J. Agricultural Eng. Res.*, vol. 54, no. 3, pp. 187–200, 1993.

[33] R. Johnstone, R. Johnson, R. Bitmead, and B. Anderson, "Exponential convergence of recursive least squares with exponential forgetting factor," in *Proc. IEEE Conf. Decis. Control*, Orlando, FL, USA, Dec. 8–10, 1982, pp. 994–997.