2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Sensor Driven Feedback for Puff Estimation using Unmanned Aerial Vehicles

Liqian Peng and Kamran Mohseni, *Senior Member, IEEE*

***Abstract*— The puff (or the pollutant puff) represents an instantaneous pollution cloud released in the ambient atmosphere. This paper describes, and validates a complete dynamic data driven application system (DDDAS) for measuring and simulating a puff in a dynamic, urban environment. Unmanned aerial vehicles (UAVs) are used as mobile sensors to collect data from the concentration field which is then assimilated into a running advection diffusion simulation to predict the puff motion. In turn, the running simulation is used to determine desirable locations to place the sensors based on the previously collected data. We directly compare the error achieved in a real-time, low resolution simulation by using both static and mobile sensors in a the DDDAS. The scenario investigated here is analogous to a chemical puff that is released in an urban environment and travels downstream according to the advection diffusion equation. We find that a single mobile sensor in the DDDAS outperforms an array of several static sensors in this scenario. Additionally, groups of mobile sensors are able to further decrease the error levels in the simulation.**

I. Introduction

The possibility of using low cost miniature aerial robots provides a very promising option for safe and cost-effective data collection. The primary benefits of such a system are the relatively low cost of individual vehicles and the safety of using unmanned vehicles for missions involving harsh atmospheric conditions or toxic environments that are too dangerous to send humans. Unmanned aerial vehicles (UAVs) are able to gather information over large areas and have the ability to backtrack, allowing for more complete data collection in a dynamic environment. Possibly the most widely known mobile, in atmospheric sensing system is the Aerosonde unmanned aerial vehicle [1]. The Aerosonde is a UAV that can be equipped with atmospheric sensors and flown through the environment to be measured. This has the notable advantage over dropsondes of allowing maneuverability in the horizontal and vertical directions and, in principle, the ability to react to the data being collected. Despite the capabilities of the Aerosonde UAV, it is relatively large (approximate wingspan of 3 meters) and its rather high cost is not desirable for applications involving multiple vehicles. A sample of the work that has been conducted in the last few years using concepts similar to the Aerosonde UAV but with smaller and lower cost vehicles can be found in [2], [3]. In many sensing applications, it is more desirable to have many sensors spaced throughout a domain than only a single sensor.

In any mobile sensor application, a key consideration that must be addressed is how to position the sensors in order to gain substantially useful information from their locations. The framework of dynamic data-driven application systems (DDDAS) is driven by the goal of dynamically incorporating data into a running application (e.g. an environmental simulation) while simultaneously using the application to steer measurement processes [4]. The DDDAS framework is widely used in wildfire simulation [5], [6], identification of airborne contaminants [7]–[9], and weather forecasting [10]. More applications of UAVs for the Puff mapping problems can be found in [11], [12]. Rather than solving the inverse problem to estimate the initial contaminant state, we use a Kalman filter to update the current state. This leads to an efficient method that is suitable for real-time dynamic control of UAVs and miniature/micro-aerial vehicles (MAVs) while implementing a full DDDAS simulation with vehicle dynamics.

This paper builds upon the authors' previous work demonstrating the environmental volumetric monitoring capabilities of a small team of unmanned aerial vehicles [13], [14]. The primary contribution of this article is the implementation of a full DDDAS coupled feedback loop, giving autonomous, dynamic data-driven feedback to guide mobile sensors and maximize the utility of data collected by the sensors. Mobile sensors are guided by a running simulation of a concentration plume using several different measurement strategies and the measurements are fed back into the live simulation. This leads to a rapidly adapting measurement strategy that guides sensors to impactful areas of a dynamic environment and quickly improves the running application to maximize accuracy of current knowledge and future predictions.

The concentration puff measured in this paper is a simulated two-dimensional puff initialized as a Gaussian concentration field and propagating downwind past two building-like obstacles. A high-resolution simulation was carried out in advance and is treated as the true concentration field. A mobile sensor moves through the environment, collecting data on the puff concentration. Using a Kalman filter data assimilation technique, this data is fed back into a low resolution puff simulation running in real-time. The running simulation is then used to steer the mobile sensors and the measurement process.

The error in the running simulation is compared to the error achieved without data assimilation and from using static sensors rather than mobile sensors for data assimilation. We

L. Peng is with the Department of Mechanical and Aerospace Engineering and the Institute for Networked System, University of Florida, Gainesville, FL USA, e-mail: (liqianpeng@ufl.edu).

K. Mohseni is with the Department of Mechanical and Aerospace Engineering, the Department of Electrical and Computer Engineering, and the Institute for Networked System, University of Florida, Gainesville, FL USA, phone: 352-273-1834, e-mail: (mohseni@ufl.edu).

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

find that a single mobile sensor in the DDDAS outperforms an array of nine static sensors in this scenario. The mobile sensors decrease the initial error in the simulation more quickly than static sensors and achieve a lower error level overall.

The manuscript is organized as follows. We begin with an overview of the DDDAS, including the mobile sensor control scheme, the numerical simulations used, the data assimilation procedure, and the sensor placement algorithm. We then discuss and present the numerical simulations used to investigate the DDDAS. We demonstrate greatly improved simulation accuracy when the full DDDAS is used with mobile sensors rather than a more standard data assimilation method using static sensors. Finally, we conclude with some general comments and observations about the DDDAS presented here as well as DDDAS's in general.

II. DDDAS OVERVIEW

The methods used in this paper combine to form a complete dynamic data-driven application system (DDDAS) [4]. The DDDAS is characterized by two coupled feedback loops running in real-time to improve the accuracy of a running simulation while simultaneously improving the effectiveness of data collection strategies. A schematic of this system is shown in figure 1. We refer to the first loop in the DDDAS as the *sensor guidance and control loop*. The system components in this loop are responsible for sensor control and guidance including movement, collision avoidance, and data collection. The second loop in the system is referred to as the *simulation loop*. In this loop, a real time simulation attempts to model the physical system of interest; in our case, a concentration puff is modeled using the advection-diffusion equation.

The key to the DDDAS is an effective, two-way coupling between the simulation loop and the sensor guidance and control loop. This is accomplished by using a Kalman filter based data assimilation routine to assimilate data collected in the guidance and control loop into the running simulation loop, resulting in improved accuracy in the simulation. Meanwhile, the simulation loop is used to steer the mobile sensors to impactful measurement locations, improving the data collection process. We first discuss the nondimensionalization used in this paper, then move on to introduce the mobile sensor control scheme that is used to ensure sensor collision and obstacle avoidance while guiding sensors to desired locations. We also describe the simulation loop used to model the advection diffusion processes. We then describe the data assimilation and sensor methods that are used to couple these two loops.

A. Nondimensionalization

All variables, including time, length, and velocity, are given in nondimensional units unless otherwise specified. The nondimensionalization is based on a dimensional velocity V_0 and a length scale L. In the scenario presented below, V_0 is the inflow velocity into the domain and L is the size of the buildings used. These two parameters lead to dimensionless quantities as follows:

$$t = \frac{t' \cdot V_0}{L}, \quad \mathbf{x} = \frac{\mathbf{x}'}{L}, \quad \mathbf{v} = \frac{\mathbf{v}'}{V_0} \tag{1}$$

where t, $\mathbf{x}$, and $\mathbf{v}$ are the nondimensional time, length, and velocity and the primed variables are their dimensional counterparts.

B. Sensor mobility control

The sensor guidance and control element of the DDDAS governs the motion and behavior of the mobile sensors. While many options exist for the mobile sensor controller, a scheme based on smoothed particle hydrodynamics (SPH) is chosen as the basis to guide vehicles using a virtual attractor particle [15] while also maintaining a safe separation between vehicles for collision avoidance. The SPH control algorithm distributes computations among vehicles and limits vehicle-vehicle interactions to a finite radius, allowing for future scalability to larger vehicle groups and more complex control problems. Here, we describe only the most important aspects of the SPH control scheme. For additional details, the reader is referred to the references of this section.

The SPH method uses a particle-based (Lagrangian) discretization of the Navier-Stokes equations of fluid motion [16], [17]. This method has been recently used as a vehicle control scheme with built in fluid-like motion and collision avoidance [18]–[22]. The SPH discretization uses *Lagrangian* particles to represent a fluid. The particles take on all properties of the fluid, which may then be applied through the use of a "smoothing kernel". The relevant SPH equations can be found in many resources such as [16] or [17] and are summarized here. The smoothing kernel used in this paper is given by

$$W_{ij} = \frac{2}{3h_{ij}^2} \begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3 & \text{if} \quad 0 \le s \le 1 \\ \frac{1}{4}(2-s)^3 & \text{if} \quad 1 < s \le 2 \\ 0 & \text{if} \quad s > 2 \end{cases}, \tag{2}$$

where h_{ij} is the smoothing width of the kernel and $s = r_{ij}/h_{ij}$ where r_{ij} is the distance between particles i and j. The Navier-Stokes equations for conservation of mass and momentum are then discretized as

$$\rho_i = \sum_j m_j W_{ij}, \tag{3}$$

$$\frac{D\mathbf{v}_i}{Dt} = -\sum_j \left[m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} \right] + \sum_j \left(\frac{\mathbf{\Pi}_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}} \right), \tag{4}$$

where subscripts denote particle identity, ∇_i denotes the gradient with respect to particle i, and $\mathbf{\Pi}$ is the viscous stress. The pressure terms are computed using the equation of state

$$P_i = K_i \rho_i \left(\frac{\rho_i}{\rho_0} - 1 \right). \tag{5}$$

All spatial derivatives that are normally present in the Navier-Stokes equations now manifest themselves as the ∇W terms

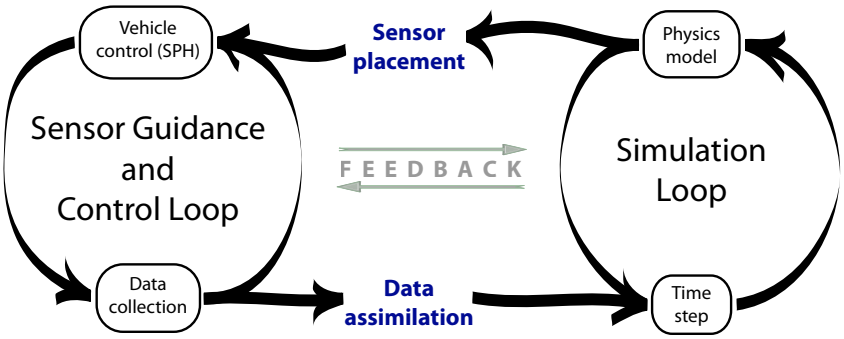

Fig. 1. Schematic of the DDDAS. The two main components of the system are a sensor guidance and control loop and a simulation loop. In the first loop, the UAVs are guided through the SPH algorithm to collect useful information and avoid collisions. In the second loop, a simulation is carried out to estimate the evolution of puff evolution. Through data assimilation, dynamic data is injected into the puff simulation to improve the estimation accuracy. Meanwhile, the simulation-based sensor placement algorithm can be used to determine the hot spots of puff evolution and improve the measurement efficiency.

and the SPH equations form a system of ordinary differential equations that may be integrated in time.

In the context of cooperative control, each vehicle is associated with a particle in the SPH representation and the vehicle positions are given by

$$\ddot{\mathbf{x}}_i = \frac{D\mathbf{v}_i}{Dt} + \frac{\mathbf{F}_i}{m_i}. \tag{6}$$

Often, an external force $\mathbf{F}_i$ is used to provide vehicle guidance while the SPH forces provide for smooth motion with obstacle and collision avoidance. To avoid the computational cost of generating an appropriate external force, we use reduced density virtual particles for vehicle guidance [15]. These reduced density particles create low pressure regions in the domain and are used to attract the vehicles to their desired locations. In this paper, a reduced density virtual particle will be used as a target point to guide vehicles through the domain to the desired measurement locations.

C. Simulation

The DDDAS approach is quite general and can be applied to many phenomena for which good simulation models are available. Here, we focus on measuring and tracking a concentration puff from sparse observations in an advection-diffusion system described by

$$\frac{\partial \psi}{\partial t} + \mathbf{v} \cdot \nabla \psi = \frac{1}{Pe} \nabla^2 \psi; \qquad \psi(0, x) = \psi_0(x), \tag{7}$$

where $\mathbf{v} = \mathbf{v}(t, x)$ is the background wind velocity, Pe is the Péclet number which quantifies the relative strength of the advection and diffusion terms, $\psi(t, x)$ is the concentration field, time $t \in (0, T)$, and $x \in \Omega$. Since there is some uncertainty or error in the values of ψ caused by background noise as well as noise in $\mathbf{v}$ and ν, this may be treated as a stochastic equation. We examine the case where an initial Gaussian puff of concentration is advected under the action of the velocity field $\mathbf{v}$ while also dispersing due to the diffusion. After discretization in space and time, (7) yields

$$\psi_k = A\psi_{k-1} + w_{k-1}; \qquad \psi_0 = f_0 \tag{8}$$

where $\psi_k \in \mathbb{R}^n$ is the concentration at time step k, $A \in \mathbb{R}^{n \times n}$ is the discretized linear operator accounting for the advection and diffusion effects and $w_k \in \mathbb{R}^n$ is the noise introduced by the uncertainty in the background velocity field, the diffusion constant, and the concentration source. For simplicity, the noise w_k is assumed to follow a multivariate normal distribution whose covariance matrix is $c_1 I$, where I is an identity matrix and c_1 is a constant. In an offline computation, which simulates the real puff concentration, the advection-diffusion equation is solved using a semi-Lagrangian advection step and an implicit central difference scheme for the diffusion term as described by Stam [23]. To limit the numerical diffusion and improve accuracy, a second order midpoint method is used for particle advection and cubic splines are used to interpolate from the grid to the advection points. Since speed is crucial for the online computation and control, a much lower resolution discretization is used to maintain speed. Data assimilation will also be used to improve accuracy of the online simulation.

D. Data assimilation

To assimilate measured data into the running simulation (and couple the simulation loop to the control loop) we use the standard Kalman Filter technique [24]. In contrast to batch estimation techniques, the Kalman filter is a recursive estimator and does not require a history of observations and estimates. The state of the filter is represented by a state estimate at time k and the error covariance matrix (a measure of the estimated accuracy of the state estimate). The Kalman filter is most often conceptualized as a prediction stage and a correlation stage. The prediction stage uses a noiseless, low resolution version of equation (8) to produce an updated estimate of the state at the current time step. This predicted state estimate does not include observation information from the current time step. In the update phase, the current a priori prediction is combined with current observation at selected positions to refine the state estimate. This improved estimate is termed the a posteriori state estimate. The initial state is assumed to have an uncertainty whose covariance matrix is set as $\psi_0 \psi_0^T$. In the data collection stage, each sensor reads

the plume concentration at its location. Some measurement noise is also added and is assumed to follow a multivariate normal distribution whose covariance matrix is $c_2 I$, where c_2 is another constant.

E. Sensor placement

To maximize the measurement efficiency and improve sensor placement we also couple the control loop back to the running simulation loop by using the simulated data to steer the mobile sensors. The sensor locations are chosen using several vehicle guidance strategies to investigate their effectiveness at decreasing the total simulation error. Fuel usage and the ability of mobile sensors to reach their goal points are of primary concern in the guidance strategy. To efficiently choose measurement locations, we use a series of 2D weighting functions to determine the desirability of measuring points in the simulation domain. The desirability is quantified by the function

$$W(\mathbf{x}) = W_1 \cdot W_2 \cdots W_{n-1} \cdot W_n, \tag{9}$$

where the W_i's represent various effects that may increase or decrease the desirability of traveling to and measuring points in the domain. For this study, we consider three different effects:

1) Simulated concentration field:

$$W_1 = \psi(\mathbf{x}).$$

We wish to focus the sensor measurements near the concentration puff so the suitability of a measurement location is chosen to be proportional to the simulated concentration field.

2) Distance from the vehicle:

$$W_2 = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}^*||_2^2}{2R_1^2}\right).$$

Since it is less costly to measure points near the current vehicle locations, we use a Gaussian function to weight nearby points more strongly than those far away from the sensors. In this representation $\mathbf{x}^*$ is the sensor location and R_1 is a range parameter.

3) Vehicle orientation:

$$W_3 = \cos^2\left(\frac{\theta-\theta^*}{2}\right) + \left(1-\exp\left(-\frac{||\mathbf{x}-\mathbf{x}^*||_2^2}{2R_2^2}\right)\right)\sin^2\left(\frac{\theta-\theta^*}{2}\right).$$

In the case of vehicles with non-negligible momentum (particularly aerial vehicles), it is much more costly to measure points immediately behind the vehicle than those in front of it. In fact, aerial vehicles must perform a complete circular turn to reach a point immediately behind their starting location. W_3 considers the sensors' headings and decreases the weight of points directly behind the sensors. In this representation, θ^* is the vehicle heading and θ is the direction to the point of interest.

In section III, we consider two possible sensor placement strategies. First, we let $W = W_1$, placing sensors based solely on the simulated concentration field. This results in all sensors being guided toward the same location and there is a unique point of maximum concentration in the domain. Secondly, we consider $W = W_1 \cdot W_2 \cdot W_3$. This sensor placement strategy considers the location and heading of the mobile sensors and allows different sensors to be steered toward different points in the domain. The combined effect of W_1 and W_2 is shown in figure 2. We expect that this will improve the measurement effectiveness. Once sensor placement locations have been chosen based on the value of W, the sensor mobility scheme described in section II-B is used to guide the mobile sensors toward these desired locations by placing attracting particles at the desired measurement locations.

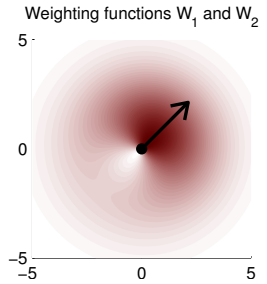

Fig. 2. The combined effect of W_1 and W_2 for $R_1 = 2$ and $R_2 = 3$. The shading level shows the product $W = W_1 W_2$ which corresponds to the "desirability" of measuring points in the domain. The mobile sensor heading is shown with a black arrow. W_2 decays with distance and W_3 creates a blind spot behind the sensor.

III. Numerical simulation results

In this section, the DDDAS described above is shown to successfully monitor and characterize a simulated concentration puff that is analogous to the release of a puff of toxic chemicals in an urban environment. The puff motion is governed by the advection diffusion equation given in II-C. However, in this test, neither the precise location nor the extent of the puff is initially known. As time progresses the DDDAS is able to identify, follow, and update the initial model of the puff by guiding the mobile sensors to measurement hot spots while assimilating the data measured along the sensor paths.

A. Numerical puff simulation

As a source of data, we first complete an offline, high-resolution puff simulation. For the remainder of our tests, the data resulting from this high resolution simulation is treated as being exactly correct. A coupled, non-dimensional Navier-Stokes and advection diffusion system is solved simultaneously on the spatial domain $\Omega = [0, 30] \times [0, 10]$. A Reynolds number of $10,000$ and a Peclet number of 5,000 were used to provide a reasonable velocity field with a mildly diffusive puff. The velocity boundary conditions are set as $u = 1,\ v = 0$ at the left and right boundaries with free-slip wall conditions at the top and bottom of the domain. Additionally, two square obstacles are placed within the domain to simulate buildings in an urban environment.

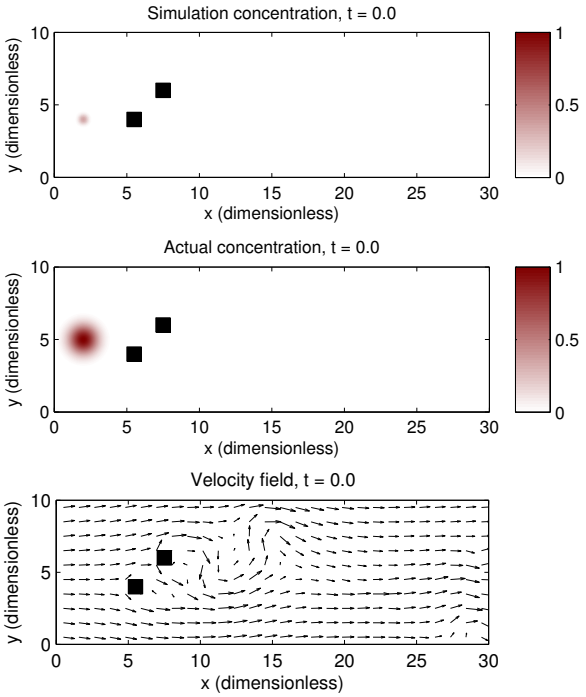

Fig. 3. The initial conditions for the results discussed here. The puff is detected and initialized at $(2, 4)$ (top), but the actual puff is much larger and centered at $(2, 5)$ (middle). The velocity field (at $t = 0$) is shown at bottom.

The squares are of size 1 and are centered at $(5.5, 4)$ and $(7.5, 6)$. These obstacles were applied using no-slip boundary conditions. All concentration boundary conditions were set to $\psi = 0$ for simplicity. The simulation is run from $t = -40$ to $t = 0$ with no concentration field in order to allow the velocity field to spin up. A time $t = 0$, the concentration field is initialized to $\psi(0) = \exp(-||\mathbf{x} - \mathbf{x}^*||_2^2)$ where $\mathbf{x}^* = (2, 5)$. This corresponds to a puff centered at $(2, 5)$. For our purposes, the physical accuracy of the simulation is not as important as having a reasonable puff data set to use.

The DDDAS described in previous sections is then applied to this data set. Beginning with the assumption that the puff has been detected by a sensor located at $\mathbf{x}_0 = (2, 4)$, a low resolution simulation is initialized with the concentration field $\psi(0) = \exp(-1)\exp(-10||\mathbf{x} - \mathbf{x}_0||_2^2)$. This initial configuration is shown in figure 3 and corresponds to a small puff with the correct concentration value at $(2, 4)$. This simulation uses a grid spacing of $\Delta x = 0.2$ and a time step of $\Delta t = 0.2$. This resolution is much too low to accurately capture the small scale details that develop in the concentration puff due to the high Reynolds number velocity field.

The mobile sensors are then guided toward the puff using the scheme described in section II-E. If the size of mobile sensors (~ 0.1m) is significantly smaller than radius of the puff ($\sim$ 100m), we may assume that the UAVs will not significantly alter the puff motion. The sensors begin near the origin $(0, 0)$ and quickly move toward the simulated puff to begin data collection. The sensors "measure" the data that

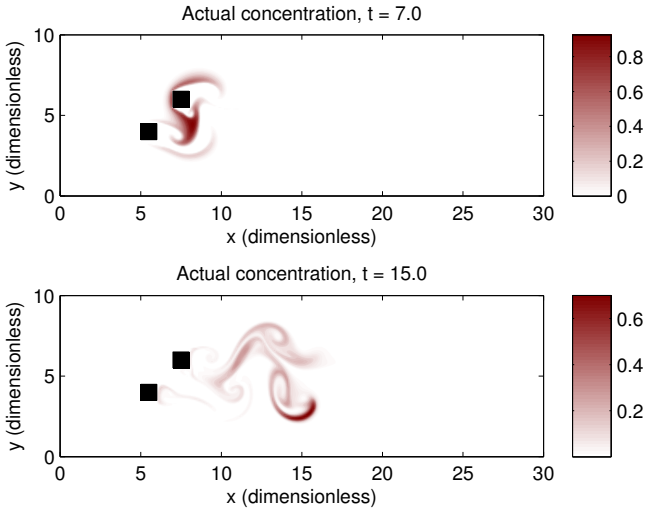

Fig. 4. The true concentration field at time $t = 7$ and $t = 15$. Most of the initial puff passes between the buildings, but some passes above the top building, splitting the original puff in two.

was pre-computed offline for assimilation into the running simulation. In turn, the running simulation is used to steer the sensors to new measurement locations. As the concentration field evolves, the puff interacts with the buildings in the domain, passing between and around them as shown in figure 4.

B. Results

In this section, we focus on the effect of varying two parameters in the DDDAS simulation: the number of mobile sensors and the type of control scheme. In all cases, the simulation component of the DDDAS system was initialized as shown in figure 3. The mobile sensors were released near the origin $(0, 0)$ and travel at a dimensionless speed of about 10 (compared to the incoming windspeed of 1). Each mobile sensor measures the true concentration value at its location and data assimilation was performed using these values at a frequency of $\Delta t = 0.2$. To increase realism, Gaussian noise with standard deviation of 0.01 was added to the data before assimilation into the running simulation. As a baseline case, the DDDAS results from the mobile sensors are compared to the results obtained using a single static sensor located at $(2, 4)$ (where the puff was initially detected) and the results obtained using 5 or 9 static sensors spaced around a circle of radius 3 centered at $(6.5, 5)$ (the point between the buildings). We also compared with the results of the low resolution simulation simulation starting with exactly correct initial conditions to demonstrate the error level caused solely by using a low resolution simulation. Our metric for comparison is the percent error in the simulated concentration puff as a function of time. This error is computed using the ℓ^2 norm.

Figure 5 shows the results we will discuss in this section. We first examine the importance of the number of mobile sensors used in the DDDAS. As expected, performing data assimilation with the static sensors decreases the error, but

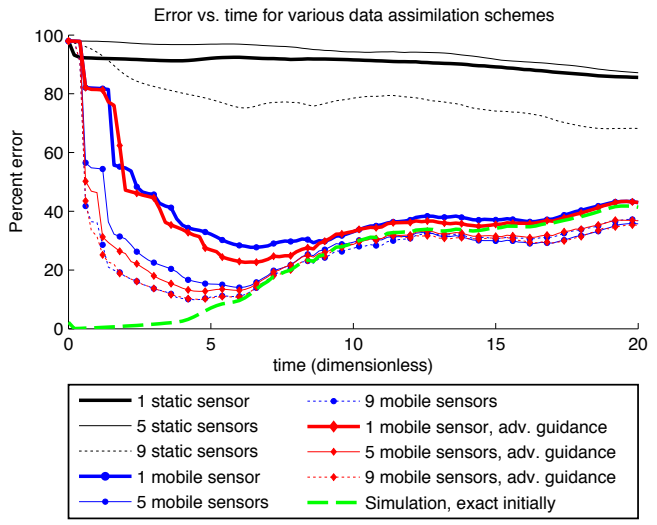

Fig. 5. Percent error v. time for the DDDAS. The black curves correspond to static sensors, the blue correspond to mobile sensors guided solely by the simulated concentration, and the red correspond to mobile sensors guided by the weighting functions described in section II-E. The green curve represents the low resolution simulation without data assimilation, but starting from exactly correct initial conditions. The results show that a low number of mobile sensors are able to decrease the error between the estimated and actual concentration fields faster and to lower levels than a higher number of static sensors. Specifically, we find that a single mobile sensor outperforms an array of nine static sensors for this scenario.

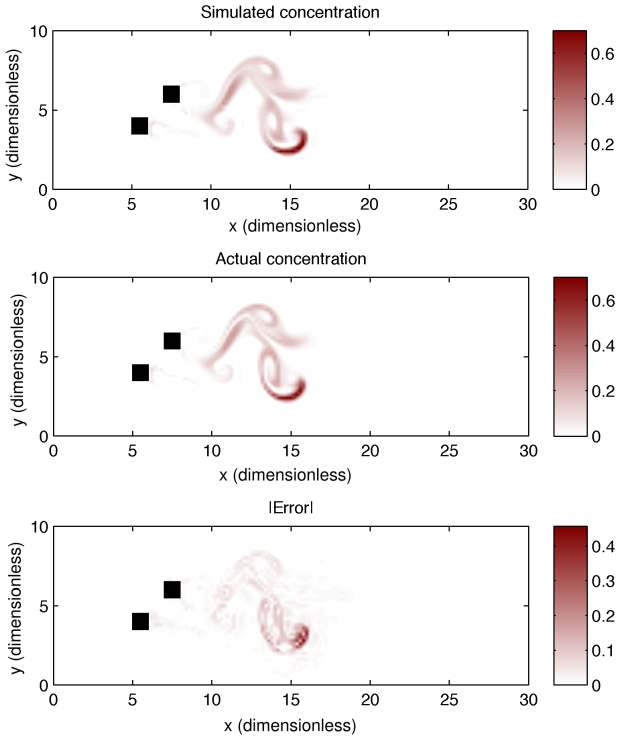

Fig. 6. Top: the simulated concentration obtained from the DDDAS using nine mobile sensors for data collection at time $t = 15$. Center: the actual concentration field at this same time. Bottom: the spatial distribution of the error in the simulation.

not nearly as much as when a DDDAS is used with mobile sensors. In fact, a single mobile sensor outperforms the nine static sensors in this scenario. The static sensors reach final error levels of 85.6, 87.2, and 68.2 percent error for one, five, and nine sensors while the simulation with exactly correct initial conditions has a final error level of 41.4 percent.

We first focus on the blue curves of figure 5 which correspond to mobile sensors using a simple guidance system that drives all sensors toward the location of highest simulated concentration. In this case, the sensors require a small amount of time to first reach the puff and start collecting useful data. After that, the error decreases very rapidly as the sensors quickly collect and assimilate data in and around the puff. The final error levels for one, five, and nine mobile sensors are 42.9, 35.7, and 36.8 percent, respectively.

The red curves in figure 5 show the error for mobile sensors using the guidance scheme described in section II-E. In this case, the final error levels for one, five, and nine sensors are 43.1, 36.8, and 35.3 percent respectively. At shorter times (i.e. $t < 10$), this more advanced guidance scheme typically achieves better results than the simple guidance scheme that considers only the concentration field. However, the final error levels for both schemes are very similar. On the other hand, the multi-vehicle data collection scenarios outperform the single vehicle scenarios in both cases. This indicates that the amount of data being collected may be more important than the specific guidance scheme. Finally, all the DDDAS scenarios quickly approach the error levels achieved by the low resolution simulation that was initialized with the exactly correct initial conditions. The multi-vehicle schemes outperform this simulation for later times since the data assimilation process is able to correct some of the error caused by the low resolution.

Finally, we address the spatial distribution of the error within the simulation domain. This is shown in figure 6. Note that even though the puff location and magnitude appears to be essentially correct, errors (on the order of the grid spacing) in the location or size of the puff show up as relatively large errors in concentration, especially in regions where large concentration gradients are present.

IV. Conclusions

We have demonstrated the feasibility and accuracy of a dynamic data-driven application system (DDDAS) for data collection and simulation of a puff of concentration in a two-dimensional velocity field. The method is computationally inexpensive enough to be implemented in real time on a standard laptop and onboard current generation UAVs. This DDDAS uses a simple advection-diffusion solver to simulate the evolution of a concentration field while simultaneously using mobile sensors to measure the actual concentration. These two processes are coupled together by assimilating the mobile sensor data into the running simulation using a Kalman filter and by using the simulated concentration field to steer the mobile sensors to desirable measurement locations. It should be noted that the DDDAS techniques used here also apply to any general situation where a model exists to simulate a process that can be dynamically

measured. More advanced data assimilation techniques or sensor steering may be required, but the general framework is applicable.

The results shown here clearly demonstrate the capability of a simple DDDAS for environmental monitoring and concentration puff detection and simulation. The accuracy of the simulation rapidly increases with the use of mobile sensors in the DDDAS. Even a single mobile sensor DDDAS outperforms data assimilation using nine static sensors for this sensor arrangement. As the number of mobile sensors increases, so does the measurement capability and the accuracy for the DDDAS. Using multiple mobile sensors, final error levels as low as 35 percent are achieved for this example of a concentration puff moving between buildings.

The numerical simulations presented above were run in near real-time on a several year old laptop computer. Additionally, field tests of this DDDAS using two small, resource constrained unmanned aerial vehicles further verify the feasibility of the system. Future work will also focus on the importance of the vehicle guidance procedure and possible application of more advanced path planning techniques. Additionally, more field tests are planned using a physical plume rather than a simulated concentration field.

The results suggest a hybrid approach for applications requiring a combination of cost effective, always-on environmental monitoring as well as high accuracy event simulation and forecasting. A collection of cheap, low power static sensors may be used for continuous monitoring. If these sensors detect a contaminant, a small group of mobile sensors can be released as part of a DDDAS to quickly and efficiently collect and assimilate detailed information about the environment.

The primary limitation in this setup is the low resolution of the running simulation. The limited grid size simply prevents accurately capturing small scale phenomena, even with the use of data assimilation techniques. Additionally, the field tests have revealed minor difficulties with GPS blackouts, limited vehicle speeds, and communication delays. However, the effective error reduction seen in this investigation indicates that these effects are relatively minor. Many additional limitations of the current setup, including the restriction to linear problems (due to the use of the Kalman filter), and possible path planning issues could be overcome by using more advanced components in the DDDAS.

Despite the inherent complexities of a DDDAS, we have shown that such systems can provide a significant advantage in data collection and assimilation in the right setting. In many situations, the initial investment required to develop such a system may be worth the dramatic increase in accuracy of the results. As mobile sensing capabilities continue to improve, we expect DDDAS's to play a growing role in real-time data collection and simulation.

REFERENCES

[1] G. J. Holland, P. J. Webster, J. Curry, G. Tyrell, D. Gauntlett, G. Brett, J. Becker, R. Hoag, and W. Vaglienti, "The aerosonde robotic aircraft: A new paradigm for environmental observations," *Bulletin of the American Meteorological Society*, vol. 82, no. 5, May 2001.

[2] A. van den Kroonenberg, S. Martin, F. Beyrich, and J. Bange, "Spatially-Averaged Temperature Structure Parameter Over a Heterogeneous Surface Measured by an Unmanned Aerial Vehicle," *Boundary-Layer Meteorology*, vol. 142, pp. 55–77, October 2011.

[3] S. Leven, J.-C. Zufferey, and D. Floreano, "Dealing with Mid-Air Collisions in Dense Collective Aerial Systems," *Journal of Field Robotics*, vol. 28, no. 3, pp. 405–423, 2011.

[4] F. Darema, "Dynamic data driven applications systems: A new paradigm for application simulations and measurements," *International Conference on Computational Science*, pp. 662–669, 2004.

[5] J. Mandel, L. Bennethum, J. Beezley, J. Coen, C. Douglas, M. Kim, and A. Vodacek, "A wildland fire model with data assimilation," *Mathematics and Computers in Simulation*, vol. 79, pp. 584 – 606, 2008.

[6] R. Rodriguez, A. Cortés, and T. Margalef, "Injecting dynamic real-time data into a dddas for forest fire behavior prediction," in *ICCS 2009 Proceedings of the 9th International Conference on Computational Science*, vol. 5545, Baton Rouge, Louisiana, May 2009, pp. 489 – 499.

[7] V. Akcelik, G. Biros, A. Drăgănescu, O. Ghattas, J. Hill, and B. van Bloeman Waanders, "Dynamic data-driven inversion for terascale simulations: Real-time identification of airborne contaminants," in *Proceedings of SC2005*, Seattle, Washington, November 2005, pp. 43–58.

[8] V. Akcelik, G. Biros, A. Draganescu, O. Ghattas, and J. Hill, "Inversion of airborne contaminants in a regional model," in *Lecture Notes in Computer Science*, vol. 3993, 2006, pp. 481–488.

[9] L. Peng, D. Lipinski, and K. Mohseni, "Dynamic data driven application system for plume estimation using UAVs," *J. Intell. Robot. Syst.*, vol. 74, pp. 421–436, 2014.

[10] M. Fisher, J. Nocedal, Y. Trémolet, and S. J. Wright, "Data assimilation in weather forecasting: a case study in pde-constrained optimization," *Optimization and Engineering*, vol. 10, no. 3, pp. 409 – 426, 2009.

[11] D. V. Zarzhitsky, D. F. Spears, and D. R. Thayer, "Experimental studies of swarm robotic chemical plume tracing using computational fluid dynamics simulations," *Int. J. Intell. Comput. Cybern.*, vol. 3, no. 4, pp. 1–44, 2010.

[12] C. Tricaud and Y. Chen, *Optimal mobile sensing and actuation policies in cyber-physical systems*. New York, NY: Springer, 2012.

[13] J. Allred, A. Hasan, B. Pisano, S. Panichsakul, P. Gray, R. Han, D. Lawrence, and K. Mohseni, "SensorFlock: A mobile system of networked micro-air vehicles," in *The ACM SenSys 2007: The 5th ACM Conference on Embedded Networked Sensor Systems*, Sydney, Australia, 6-9 November 2007.

[14] B. Hodgkinson, D. Lipinski, L. Peng, and K. Mohseni, "High resolution atmospheric sensing using UAV swarms," in *International Symposium on Distributed Autonomous Robotic Systems (DARS 2012)*, Baltimore, MD, November 8-11 2012.

[15] D. Lipinski and K. Mohseni, "A master-slave fluid cooperative control algorithm for optimal trajectory planning," in *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9-13 2011, paper WeA212.2.

[16] J. Monaghan, "Smoothed Particle Hydrodynamics," *Annual Review of Astronomy and Astrophysics*, vol. 30, pp. 543–574, 1992.

[17] G. Liu and M. Liu, *Smoothed Particle Hydrodynamics: A meshfree particel method*. World Scientific Publishing Company, 2003.

[18] L. Pimenta, M. Mendes, R. Mesquita, and G. Pereira, "Fluids in electrostatic fields: An analogy for multirobot control," *IEEE Transactions on Magnetics*, vol. 43, no. 4, 2007.

[19] L. Pimenta, N. Michael, R. Mesquita, G. Pereira, and V. Kumar, "Control of swarms based on hydrodynamic models," in *IEEE International Conference on Robotics and Automation*, May 2008.

[20] S. Huhn and K. Mohseni, "Cooperative control of a team of AUVs using smoothed particle hydrodynamics with restricted communication," in *Proceedings of the ASME 28th International Conference on Ocean, Offshore and Arctic Engineering*, no. OMAE 2009-79869, Honalulu, HA, May 31-June 5 2009.

[21] 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 Journal, Special Issue on Cognitive Sensor Networks*, vol. 11, no. 3, pp. 722–736, March 2011.

[22] D. Lipinski and K. Mohseni, "Cooperative control of a team of unmanned vehicles using smoothed particle hydrodynamics," AIAA

Guidance, Navigation, and Control Conference, Toronto, Ontario, Canada, AIAA paper 2010-8316, August 2-5 2010.
[23] J. Stam, "Stable fluids," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 121–128.
[24] J. M. Lewis, *Dynammic data assimilation, a least square approach*. Cambridge University Press, 2006.