

# High Resolution Atmospheric Sensing Using UAVs

Bobby Hodgkinson, Doug Lipinski, Liqian Peng, and Kamran Mohseni\*

**Abstract.** A technique to obtain high resolution atmospheric data using small mobile sensors is presented. A fluid based control scheme using smoothed particle hydrodynamics (SPH) is implemented to perform field measurements in a leader-follower arrangement for a team of unmanned aerial vehicles (UAVs) equipped with environmental sensors. A virtual leader is created by using a reduced density SPH particle to guide the unmanned aerial vehicles along a desired path. Simulations using the control scheme demonstrate excellent measurement ability, swarm coherence, and leader following capability for large swarms. A  $K$ -means algorithm is used to reduce the measurement error and provide accurate interpolation of the field measurement data. Experimental results are presented which demonstrate the guidance and collision avoidance properties of the control scheme using real UAVs. Readings from the UAV's temperature and humidity sensor suite are used with the  $K$ -means algorithm to produce a smooth estimation of the respective distribution fields.

## 1 Introduction

Several methods of sensing the atmosphere exist and can be classified into two primary groups: *remote sensing* or *in situ*. Well-known systems in the remote sensing group include RADAR, LIDAR, weather satellites, etc. [3, 5, 13, 29]. These systems are typically large (on the order of cubic meters), are generally focused on

---

Bobby Hodgkinson · Doug Lipinski · Liqian Peng

Department of Mechanical and Aerospace Engineering, Institute for Networked Autonomous Systems, University of Florida, Gainesville, FL

e-mail: {hodgkinson, dmlipinski, liqianpeng}@ufl.edu

Kamran Mohseni

Department of Mechanical and Aerospace Engineering, Department of Electrical and Computer Engineering, Institute for Networked Autonomous Systems, University of Florida, Gainesville, FL

e-mail: mohseni@ufl.edu

\* Corresponding author.

gathering data over very large areas (several kilometers) over a long period time, and are expensive to construct and maintain. The in situ group also contains several well-known systems such as dropsondes, weather balloons, barometers, etc. These systems are typically much smaller than remote sensing systems, and are focused on gathering high accuracy data at specific points in time and space. The data locations can be stationary or time varying but the in situ sensors are categorized by the ability to only provide data at a single location at a given time. Generally speaking, in situ sensors provide more accurate readings at a specific point than remote sensing systems but remote sensing systems are much better suited to give a *snapshot* of a desired measurement in a region of space. The atmospheric sensing community desires a sensor with the accuracy of an in situ device and the ability to construct a large area *pseudo-snapshot* of an environment. By definition, it is not possible for a system of in situ sensors to give a true snapshot, but the ability to gather a large amount of data in a time frame that is shorter than the time scale of the changing environment can be considered a pseudo-snapshot. This is where autonomous robotic aerial systems can find a tremendously beneficial niche. Aircraft with on board sensors are capable of providing high resolution readings at specific locations similar to dropsondes while also adding horizontal and vertical mobility to allow for a wider range pseudo-snapshot over a relatively short period of time.

A large amount of research has been conducted using mobile in situ devices such as dropsondes and radiosondes but the primary drawback of these sensors is that they are typically only able to measure data along a vertical line. Obviously a large number of appropriately spaced sensors are required to obtain a full three dimensional pseudo-snapshot of the environment thus increasing cost of deployment. Furthermore, dropsondes are not able to revisit specific locations and the only way to obtain information on how the environment changes as a function of time is to deploy additional sensors at different times which also increases the cost of deployment. Unmanned aerial vehicles (UAVs) are able to gather information over large areas and have the ability to revisit specific locations at later instances of time thus allowing for information about the time evolution of an environment, making the use of UAVs a viable option in atmospheric sensing. Possibly the most widely known mobile in situ atmospheric sensing system is the Aerosonde UAV [10]. The general concept of the Aerosonde UAV is to equip a small UAV with in situ atmospheric sensors and fly the aircraft in an environment. This concept yields a notable advantage over dropsondes in that the aircraft can maneuver in the horizontal and vertical directions. A sample of the work that has been conducted using similar concepts of the Aerosonde UAV with smaller, lower-cost UAVs can be found in: [6, 7, 31, 32].

The focus of this article is to present a system comprised of multiple small UAVs equipped with a simple atmospheric sensor suite to gather a three dimensional pseudo-snapshot of an environment. A  $K$ -means algorithm provides a smooth estimation of the environment using the discrete points obtained by the mobile sensors while also reducing the noise associated with the measurements. Additionally, the objective requires a flexible and easily implementable cooperative control scheme to guide the vehicles and ensure collision avoidance when the vehicles are in close proximity. While many options exist, we have chosen to use a fluid based control implemented

with the Smoothed Particle Hydrodynamics (SPH) fluid dynamics scheme due to several desirable characteristics. This scheme treats each vehicle as an individual fluid particle and the control forces are determined by the SPH approximation of the Navier-Stokes equations of fluid motion. This technique ensures collision avoidance and also creates a flexible controller that is computationally reasonable for implementation on resource constrained platforms. While UAVs are used in this demonstration, the SPH control scheme can be applied to ground and underwater robots as well as heterogeneous swarms of robots containing any combination of ground, aerial, and underwater platforms [11].

In the following sections we present an overview of the SPH control scheme, discuss the error reduction technique, and present test results using multiple UAVs demonstrating several highly desirable properties of the SPH control scheme. We also present error reduced results obtained from data gathered by two UAVs flown using the SPH control.

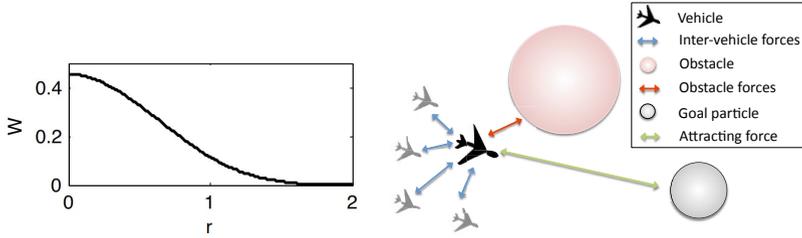
## 2 SPH Control Scheme

While there are many possible control schemes for use with small UAVs, fluid based control is especially appealing since fluid flows have many properties that a group of vehicles may wish to mimic [11, 15, 17, 18, 26]. Chiefly, fluids exhibit smooth motions and do not penetrate obstacles. In terms of vehicle control, these properties correspond to efficient motion and collision/obstacle avoidance. Additionally, UAVs operate in a fluid environment meaning a fluid based control scheme may allow for easier integration of strong background flows into the vehicle path planning process.

In particular, the smoothed particle hydrodynamics (SPH) discretization has proven to be an effective method of applying the Navier-Stokes equations in a control setting [11]. This Lagrangian technique treats each vehicle as a fluid particle, giving fluid-like motion for vehicle swarms with inherent collision and obstacle avoidance. A more complete discussion of the method is available in a review article by Monaghan [23] or the book by Liu and Liu [19]. Here we present only the aspects of SPH that are used in our cooperative control scheme.

The SPH algorithm is computationally efficient since each vehicle is represented by a single fluid particle. By choosing a compactly supported smoothing kernel for the particles, it is also possible to limit vehicle interactions to a short range. This results in vehicles interacting with only their nearest neighbors (typically no more than six vehicles in 2D). The localized interactions also make long range communication unnecessary. These advantages result in a control algorithm that is simple enough to run in real time using the limited processing capabilities of the robot [30]. Additionally, this is a distributed control scheme that requires no central controller since only local vehicle interactions are used. All the benefits of a distributed control or peer-to-peer control scheme are thereby included as well [12, 25].

The SPH scheme is dependent on choosing a Gaussian-like smoothing kernel which is used to apply fluid properties. We use the cubic spline kernel shown in Figure 1. This kernel is nonzero for  $\|r\| < 2h$  and defines the interaction range



**Fig. 1** (Left) The cubic spline smoothing kernel (with  $h = 1$ ) as a function of distance,  $r$ , used in this article. (Right) A schematic of the SPH control scheme and the forces involved.

between particles. Particles interact through pressure and viscous forces that are applied through the smoothing kernel, determining the particle motion. The SPH acceleration is given by

$$\frac{d\mathbf{v}_i}{dt} = -\sum_j m_j \left[ \left( \frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W(\mathbf{r}_{ij}, h) + \frac{\Pi_{ij}}{\|\mathbf{r}_{ij}\|} \frac{\partial W(r_{ij}, h)}{\partial \|\mathbf{r}_{ij}\|} \right] \quad (1)$$

where  $P_i$  is the pressure at particle  $i$ ,  $\rho_i$  is the density,  $\Pi_{ij}$  is a viscous force between particles  $i$  and  $j$ , and  $\mathbf{r}_{ij}$  is the relative position vector  $\mathbf{r}_j - \mathbf{r}_i$ . In this study, we neglect the viscous force for simplicity. The density is computed by summing over nearby particles

$$\rho_i = \sum_j m_j W(\mathbf{r}_{ij}, h) \quad (2)$$

and the pressure is computed using an equation of state

$$P_i = K \left( \frac{\rho_i}{\rho_0} - 1 \right) \quad (3)$$

where  $K$  is a positive coefficient.

Previously, SPH control schemes have used external forces for swarm guidance while the SPH forces have mainly provide collision avoidance [11, 27, 30]. However, we take the approach of using *reduced density virtual particles* for guidance. Just as a reduced density region in a fluid creates pressure gradients that drive fluid to this region, the reduced density particles attract vehicle particles to them.

By defining a particle's mass to be large enough that its density is always at least as large as the reference density,  $\rho_0$ , the particle is ensured a non-negative pressure, but if the mass is chosen to be less than this threshold, negative pressures can result. In the SPH control scheme, all particles representing vehicles will be given sufficient mass to ensure positive pressure and a single attracting virtual particle will be used which uses a reduced mass to create a negative pressure region. This negative pressure creates attracting forces and therefore acts as a goal region. In the low mass limit the acceleration terms simplify, slightly reducing computational cost. A schematic showing the interaction of vehicles, obstacles and attracting particles is shown in Figure 1.

In the control scheme, the SPH accelerations for a particle due to all nearby (within  $2h$ ) particles are computed and passed to the vehicle controller, which attempts to enforce the desired motion through a combination of roll, pitch, and thrust commands.

### 3 UAV Data Sampling and $K$ -means Approximation

Suppose we want to measure the temperature or humidity of a 3-dimensional region. Our UAVs can quickly collect large amounts of data through the custom temperature and humidity sensor suite. However, in experimental flights these sensors may not be highly accurate due to the limitations of sensor response time which is related to the rate of change of temperature and humidity over time and space and vehicle flight speed. For this reason, we use a  $K$ -means based error reduction scheme to effectively reduce the noise.  $K$ -means algorithms can effectively cluster  $N$  data points into  $K$  clusters. To this effect, the field function has only  $K$  unknown coefficients corresponding to  $K$  basis functions. On the other hand, Kriging and Gaussian interpolation are spanned by  $N$  basis functions. Therefore, the  $K$ -means algorithm can be considered as a dimension reduction technique that approximates the field function without significant loss of information. Additionally, the computational cost is significantly lower for  $K$ -means than Kriging and Gaussian process regression if  $K \ll N$ . One drawback of the  $K$ -means algorithm is that it may filter the high frequency components of the original field. However, as long as the ensemble of the data set is large enough and the field is smooth, the  $K$ -means algorithm is able to capture the main modes of the field. This method also enables us to approximate data at unsampled locations, and potentially (in future work) suggest a path for the UAVs to follow and collect additional data to minimize the existing uncertainty. This section will discuss the effects of noise on the data and introduce the statistical method for noise mitigation and interpolation.

Let  $x$  be any point in the measurement domain and  $x_i$  denote a spatial location of a sensor measurement at time step  $i$ . Each sensor inevitably introduces some location error,  $\xi_i$ , and some measurement error,  $\varepsilon_i$ . We devise a scheme based on the  $K$ -means algorithm to extract information from limited measurements and reduce measurement noise.

Suppose the original field  $f(x)$  is a smooth function of position, sampling points  $x_i$  near  $x$  can be used to approximate the field of  $x$ ,

$$f(x) = y_i + \varepsilon_i + J(x) \cdot \xi_i + J(x) \cdot (x - x_i) + O(|x - x_i|^2) + O(|x - x_i| \cdot |\xi_i|) + O(|\xi_i|^2)$$

where  $J(x)$  is the Jacobian matrix at point  $x$ . A better estimator  $\hat{f}(x)$  can be obtained through a linear combination of some measurement results  $y_i$ . Our goal is to find an optimal weighting function,  $\varphi_n(x, x_i)$  so that the error in  $\hat{f}$  is minimized.

If many points are measured and the measurement error is potentially large the variance becomes the dominant error. To reduce the noise error, we will use a cluster of points instead of a single point to estimate the unknown field. We use the well know  $K$ -means algorithm [21] to cluster the data points into  $K$  groups. Each cluster

is assigned a value near the mean of its members and interpolation may be performed using a radial basis function network. Let  $C$  be the encoder,  $C(i) = j$  means the  $i$ th points belongs to the  $j$ th cluster. To optimize the clustering process, we use the following cost function [8]

$$J(C) = \sum_{j=1}^K \sum_{C(i)=j} \|x_i - u_j\|^2. \quad (4)$$

where  $u_j$  is the center of cluster  $j$ . To minimize the cost function  $J(C)$ , we use an iterative descent approach  $K$ -means algorithm [20, 22] to find the encoder. The  $K$ -means algorithm proceeds with the following two steps iteratively until a convergent encoder has been obtained:

Step 1. Minimize the cluster variance with respect to the cluster means  $\{u_j\}_{j=1}^K$ :

$$\min_{\{u_j\}_{j=1}^K} \sum_{j=1}^K \sum_{C(i)=j} \|x_i - u_j\|^2 \quad \text{for a given } C. \quad (5)$$

Step 2. Having computed the optimized cluster means in step 1, we next optimize the encoder as:

$$C(i) = \arg \min_{1 \leq j \leq K} \|x_i - u_j\|^2. \quad (6)$$

Then, we can build a radial basis function network [4, 24]

$$F(x) = \sum_{j=1}^K w_j \varphi(x, u_j), \quad (7)$$

where  $\varphi(x, u_j)$  is the Gaussian kernel function.  $w_j$  can be seen as an approximation of the temperature at the cluster center. Its value can be trained by a least mean square algorithm.

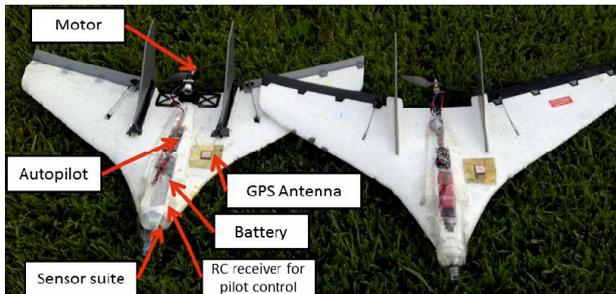
In the following section, the functionality of this algorithm is verified by a simulation of several aircraft flying through an artificial temperature field. The algorithm is also applied to experimental data of two aircraft equipped with humidity and temperature sensors.

## 4 Simulation and Experimentation

In the following we describe the experimental platform and use it to demonstrate certain properties of the SPH control scheme. We also verify the  $K$ -means error reduction technique using data obtained through simulation and apply the reduction technique to data from an experiment of two vehicles controlled by the SPH scheme. In order to accurately match the simulation to the physical world, the limited flight capabilities of the UAVs are imposed in the simulation through particle velocity and acceleration constrains. Also the inaccuracies of the real world sensor readings are modeled by adding noise to the sensor readings in simulation.

#### 4.1 UAV Hardware and Sensor Suite

A UAV equipped with a simple sensor suite and limited onboard computation capabilities is used for this study (shown in Figure 2). The aircraft has a wingspan of 0.8 m and weighs less than 0.5 kg. The airframe is a single piece of Styrofoam and can be purchased under the name F-27 Styker; additionally the ailerons, flaps, nose cone, propellers, etc. are all mass produced thus making the aircraft inexpensive compared to a custom design. This aircraft has been successfully used in several experiments in our research group and has proven itself robust to experimental mishaps, relatively simple and inexpensive to maintain. Additionally, the small size and simplicity of the aircraft allow for rapid deployment by a single pilot in areas that are not ideal for traditional UAVs. Most importantly, the Delta-Wing UAV is equipped with a custom autopilot to allow for implementation of custom control strategies. The CUPIC is a complete autopilot system developed at the University of Colorado at Boulder [28], and used in a large number of experiments in our (and other) research group(s). Several studies [2, 14] have shown that it is possible to achieve fully autonomous operation of a small UAV by means of this simple autopilot equipped with a limited number of sensors. The work of Floreano et. al [9, 16] has demonstrated the use of a similar fixed wing aircraft in swarm applications by using a different autopilot system. Additionally, Sensefly [1] is a Swiss company that provides a UAV that can be used to gather high resolution imaging using a similar hardware platform.



**Fig. 2** Resource constrained delta-wing UAVs used in experiments. The 0.8 m wingspan UAV is equipped with a GPS sensor, a roll rate sensor, a communication radio, and autopilot to control the craft during autonomous operation.

Shaw and Mohseni [30] showed that the CUPIC autopilot is also capable of demonstrating fully autonomous, distributed cooperative control of a team of UAVs. The CUPIC, in its most basic design, consists of an on-board processor, a single axis rate gyro to sense roll rates, an absolute pressure sensor for altitude sensing, and a GPS receiver for positioning. The autopilot controls the vertical location of the aircraft through pitch and thrust commands. The pitch and thrust commands are determined from the error between the desired altitude and the current altitude as well as the magnitude of the SPH acceleration from Equation 1. The horizontal

location of the aircraft is controlled by varying the roll angle. The desired roll angle is determined by the error between the direction of the SPH acceleration vector (from Equation 1) and the aircraft's current heading vector. Virtual saturation limits are employed in order to avoid commands which would result in aircraft stall or an excessive roll angle. The program has built-in routines to account for short term blackouts in the GPS signals and the noise and drifts in the sensors. The autopilot has been proven to be fully capable of stable autonomous flight on a wide variety of MAVs including Delta-wing aircrafts [2], warping-wing aircrafts, and gust-insensitive aircrafts [14].

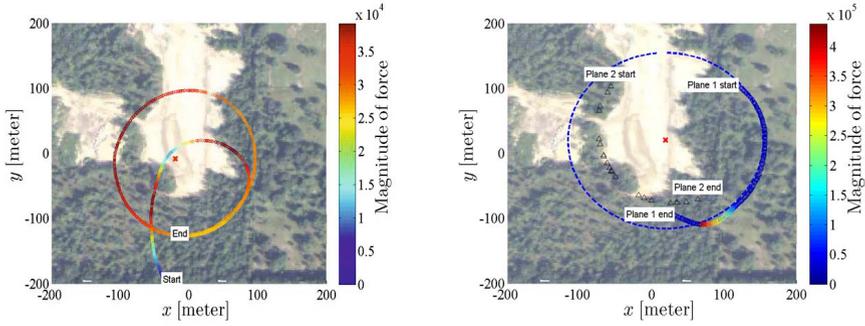
The CUPIC autopilot system also includes a complementary ground station which is comprised of a laptop running a MATLAB routine. The autopilot transmits telemetry data to the ground station and the ground station transmits commands and the location of the artificial attracting particle for the SPH control algorithm. The MATLAB routine includes a user interface that provides pilots and observers real time information of the aircraft's GPS position, physical state, sensor suite raw data, and autopilot commands. The graphical interface also includes the ability to alter the artificial particle's location, speed, and path. A detailed communication characterization of the communication scheme used in the autopilot systems is given in [30].

The autopilot was designed with the ability to interface up to 7 additional analog sensors through on board analog to digital converters. For this experiment a custom board was manufactured to house a HIH-5031 humidity sensor and a LM35 temperature sensor. The HIH-5031 humidity sensor and LM35 temperature sensor were chosen primarily due to size, simplicity and sensing range. The sensors raw output voltage is read by the autopilot at 10Hz and transmitted to the ground station along with the aircraft's most recent GPS position. The raw output voltage is then converted to percent relative humidity and temperature using equations found in the sensors respective datasheets.

## 4.2 Verification of SPH Control Scheme

In applications involving autonomous agents the two most important control aspects are agent guidance and collision avoidance. In this article, agent guidance is accomplished using a reduced density virtual particle which acts to attract agents to a specific region of space. Collision avoidance is accomplished by setting the SPH parameters of the agents such that they are repelled from each when they reach a certain distance. If the vehicle separation is greater than this distance the agents are only attracted to the reduced density particle.

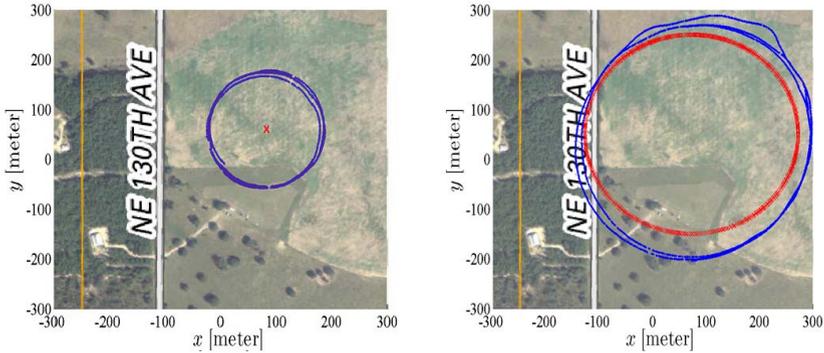
The guidance property is demonstrated by placing a stationary attractor particle at a location in the domain and engaging the autopilot with the UAV at some other point in the domain. Figure 3 shows the GPS position of the UAV demonstrating the guidance property. The marker color indicates the magnitude of the SPH force as calculated by the algorithm. The plane approaches the attracting particle (indicated by a red  $\times$ ), passes almost directly over the virtual particle and then begins to double back as the direction of the SPH force vector points opposite to the plane's heading.



**Fig. 3** Experimental demonstration of SPH control properties using small, resource constrained UAVs. (Left) Experimental results of a single vehicle (colored  $\circ$ ) approaching an attractor particle (red  $\times$ ) from the south and then beginning a loiter pattern around the attractor particle. Color of the  $\circ$  represents the magnitude of SPH force calculated by the vehicle. Results taken from a small portion of large flight experiment. (Right) Experimental results of a single vehicle (colored  $\circ$ ) avoiding another aircraft (black  $\triangle$ ) while loitering an attracting particle (red  $\times$ ). The blue dashed line indicates the previous loiter circle achieved by plane 1 prior to interaction with plane 2. Color of the  $\circ$  represents the magnitude of SPH force calculated by the vehicle.

The plane then enters into a loiter circle which is a result of a balance between the SPH force magnitude and physical limitations (i.e. turning radius) of the aircraft. For this and all following experiments, aircraft were given an  $h$  value of 30 and the attractor particle was given an  $h$  value of 200; the average of the two  $h$  values were used to determine the SPH force per Equation 1.

The collision avoidance property is demonstrated with two flying aircraft. Figure 3 plots the information received from a single vehicle (plane 1):  $\circ$  represent the vehicle's GPS position colored by the calculated SPH force magnitude, the red  $\times$  represents the location of the attractor particle,  $\triangle$  represent the GPS position of the other vehicle (plane 2) as known by plane 1. The dashed line represents the loiter circle achieved by plane 1 prior to interaction with plane 2. Plane 1 loitered in a clockwise direction while plane 2 loitered in a counter clockwise direction. As the two planes approach each other, both planes make corrections to their respective courses avoiding potential collision as evident in the course correction of plane 1. The SPH force magnitude range is greater for collision avoidance than attraction due to the fact that the interaction between planes results in a higher repulsive force than the attraction force experienced between a plane and the attracting particle. Simulations have been previously conducted showing the smooth collision avoidance property of the SPH control technique [27] and these experimental results correspond well to the simulations. Although collision avoidance may not be guaranteed in real-world situations where packet loss and location error play a role, incorporating a safety factor into the inter-vehicle spacing provides high confidence that collisions will be avoided.



**Fig. 4** Two ways to achieve loiter circles using SPH control scheme and UAVs. (Left) GPS coordinates of a single vehicle (blue ●) in a series of loiter circles around a stationary attractor particle (red ×). The loiter circles are a result of a balance between the SPH force and the physical limitations of the aircraft. (Right) Experimental results of a single vehicle (blue ●) following a moving attracting particle (red ×) demonstrating a different method of achieving a loiter.

Due to the fact that fixed wing aircraft must maintain a forward velocity to stay aloft, a loiter circle is a common technique employed in experiments involving fixed wing aircraft. One way of achieving a loiter circle using SPH control is with a stationary attractor particle and imposed acceleration and velocity constraints on the moving particle. The moving particle will move directly towards the attracting particle until it passes the particle. Due to specified constraints, the moving particle will then bank one way and eventually find and maintain an equilibrium balancing the SPH force and the imposed constraints resulting in a loiter. This property was experimentally verified and the results can be seen in Figure 4 which shows the GPS location for a single aircraft (blue ●) loitering around a stationary attractor (red ×).

While a stationary particle allows for a loiter circle, the radius of the circle is a function of several parameters and thus the radius of the loiter is not easy to predict. It is difficult to maintain a uniform loiter since any disturbance to the vehicle's path could result in the plane taking a more direct approach over the attractor particle as evident in Figure 3. A more robust way to achieve a loiter circle is with a moving attractor particle. If the velocity and acceleration of the attracting particle are within the physical limitations of the aircraft, the vehicle will follow closely behind the attractor. This technique is shown in Figure 4 with the GPS location of the aircraft shown as blue ● and the attractor location as a red ×. A moving particle allows for a loiter circle of varying radius as well as more complex paths.

### 4.3 Sensing Using Multiple Vehicles

Next we simulate a more complicated situation involving multiple vehicles taking measurements over a large domain to determine the temperature field. The results

are shown in Figure 5. In this simulation, a group of 10 vehicles begins in the lower left corner of the domain and travels back and forth across the domain, finishing in the upper right. Each vehicle records a temperature every 0.2 seconds and sends data back to the base station. The temperature is generated by the function

$$T(x, y) = 75 + 3[\sin(x/50) + \cos(y/42)\cos(x/100) + 3 \tan^{-1}((x+20)/10) + \cos(\sqrt{x^2+y^2}/40)]. \quad (8)$$

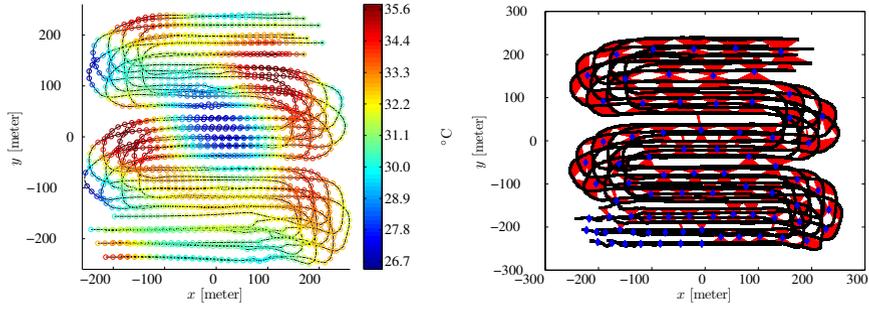
To create a closer approximation of reality and consider the limited accuracy of onboard temperature sensors, white Gaussian noise with a standard deviation of 0.78 is added to the temperature data. Every 5th point of the temperature data is plotted in Figure 5 as a colored circle. The vehicle paths are shown as black curves and the end vehicle positions are shown as black dots. The SPH controller maintains an even vehicle spacing throughout the trajectories. By using this well spaced group of multiple vehicles flying at 15 m/s, a large amount of data is collected over this 250,000 sq. m domain in only 160 seconds.

This procedure produces a large amount of data (8000 data points in 160 s) that can be used to reduce the noise in the resulting interpolations. By using the aforementioned  $K$ -means algorithm we are able to interpolate the data over the interior of the domain and reduce the error. The  $K$ -means clusters for this example are shown in Figure 5. The error in the final temperature approximation can be computed as

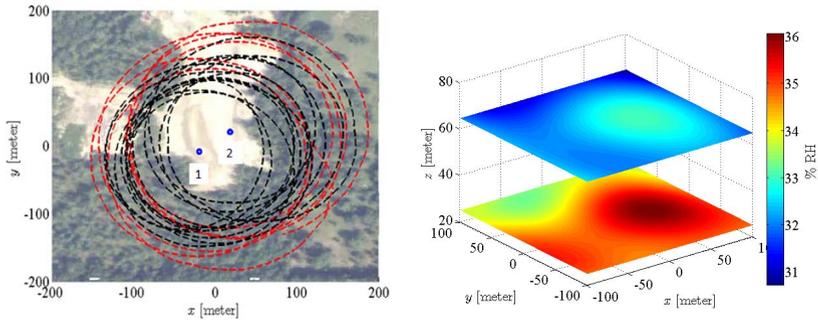
$$E_\infty = \|T - T_{\text{estimate}}\|_\infty \text{ or } E_{\text{RMS}} = \sqrt{\frac{\sum_i^n (T(\mathbf{x}_i) - T_{\text{estimate}}(\mathbf{x}_i))^2}{n}}$$

where  $n$  is the number of spatial grids,  $T$  is the true temperature (given by Equation 8) and  $T_{\text{estimate}}$  is the temperature estimated by the  $K$ -means algorithm. We find that  $E_\infty = 2.32$  and  $E_{\text{RMS}} = 0.23$  after applying  $K$ -means and interpolating. This is an improvement over the raw noisy data that had  $E_\infty = 2.97$  and  $E_{\text{RMS}} = 0.79$ . The largest errors occur at the edge of the domain where interpolations are less likely to be valid due to the limited data in these regions.

In the real world experiments, we measured humidity and temperature using the aforementioned sensor suite placed on the two aircraft used in demonstrating the SPH control scheme. An AcuRite digital humidity and temperature monitor was used to determine the temperature and percent relative humidity at the ground station location as 36 °C and 57% respectively. All the data was collected by two UAVs flying in a series of overlapping loiter circles centered at the location of a stationary attractor particle as seen in Figure 6. The path of UAV 1 is shown as a dashed black line, the path of UAV 2 is shown as a red dash-dot line, and the attractor particle is shown as a blue  $\circ$ . The loiter circles are a result of a balance between the minimum velocity enforced by the autopilot to keep the plane aloft and the attraction to the artificial particle. After each plane completed at least 3 loiter circles the location of the attractor particle was moved approximately 70 meters to the northeast resulting in the behavior shown in Figure 6. The motion of the aircraft in the horizontal plane was determined by the SPH control law while the vertical location of the aircraft



**Fig. 5** Simulation of data gathering using UAVs equipped with a temperature sensor. (Left) Simulation of 10 vehicles (black  $\circ$ 's) that were guided through a  $500 \times 500$  domain by a single attracting particle. The black curves show the full vehicle paths and the colored  $\circ$ 's denote individual temperature measurements (in  $^{\circ}\text{C}$ ). (Right) The  $K$ -means clusters for the 10 vehicle simulation. Cluster points are shown in blue, data points are shown in black and the data-cluster connections are shown in red.



**Fig. 6** Flight data of two UAVs demonstrating the SPH control scheme to gather the humidity distribution of an environment. (Left) UAV 1 path in black dashed line, and UAV 2 path in red dash-dot line. An attractor particle (blue  $\circ$ ) was placed at  $(0,0)$  and then moved to the northeast to approximately  $(50,50)$ . (Right) Humidity field results from two aircraft. The aircraft were flown approximately 40 meters apart (in altitude). The field results show higher relative humidity at a lower altitude and also a region of higher humidity at approximately  $x = 50$  m,  $y = 0$  m.

was maintained by an altitude controller. Regardless of the altitude difference, the aircraft maintain a safe horizontal separation thus avoiding collision if the UAVs altitude's were the same. The aircraft were flown at an altitude separation of 40 m in order to obtain three dimensional data about the temperature and humidity fields.

The noise of the sensor readings was rather high (approximately 10%), in order to approximate the entire field and minimize error we implemented the  $K$ -means method to reduce the noise. The results from the  $K$ -means approximations for the two aircraft are then used to find humidity and temperature as a smooth function of

space. Figure 6 shows the resulting humidity fields obtained from the two aircraft at their respective average altitude.

The test environment was a dry retention pond surrounded by woods. The results show a higher relative humidity centered at approximately ( $x = 50$  m,  $y = 0$  m) which is at the edge of the wooded region. While more tests are required to make conclusive remarks, the higher relative humidity over a region of trees is likely indicative of increased evaporation over this region compared to the dry retention pond.

## 5 Conclusions

An SPH based controller has been successfully implemented that includes a reduced density virtual attracting particle for vehicle guidance of autonomous UAVs with limited processing capabilities. The temperature and humidity data collection capabilities of multiple UAVs were demonstrated. Additionally, these UAVs experimentally verified several desirable properties of the SPH control scheme. Although the examples presented here are two-dimensional, all the techniques used are valid in three dimensions, but have been artificially restricted to constant altitude for simplicity and an added level of safety.

Additional simulations have been implemented to demonstrate the multi-vehicle capabilities of the SPH controller as pertinent to the data collection opportunities made possible by swarms of sensor equipped UAVs that can quickly collect data over a large two or three dimensional region. This is in contrast to the commonly used data collection methods available today (i.e. remote sensing, dropsondes, weather balloons, etc.). Furthermore, uncertainties in the sensor readings are mitigated using a *K*-means algorithm that is well suited to process and interpolate the data over a desired region to minimize errors. The data processing algorithm was implemented on a set of humidity and temperature data gathered by a pair of aircraft equipped with sensors flying in several loiter patterns as governed by the SPH control.

## References

1. Sensefly autonomous ultralight uav for professionals@ONLINE (October 2012)
2. Allred, J., Hasan, A., Pisano, B., Panichsakul, S., Gray, P., Han, R., Lawrence, D., Mohseni, K.: SensorFlock: A mobile system of networked micro-air vehicles. In: The ACM SenSys 2007: The 5th ACM Conference on Embedded Networked Sensor Systems, Sydney, Australia, November 6-9 (2007)
3. Amzajerjian, F., Pierrottet, D., Petway, L., Hines, G., Roback, V.: Lidar systems for precision navigation and safe landing on planetary bodies, p. 7 (2011)
4. Buhmann, M.D.: Radial Basis Functions: Theory and Implementations. Cambridge University (2003)
5. Campbell, J.: Introduction to Remote Sensing. The Guilford Press (1996)

6. Corrigan, C.E., Roberts, G.C., Ramana, M.V., Kim, D., Ramanathan, V.: Capturing vertical profiles of aerosols and black carbon over the Indian Ocean using autonomous unmanned aerial vehicles. *Atmospheric Chemistry & Physics Discussions* 7, 11429–11463 (2007)
7. Hardin, P., Jensen, R.: *Small-scale unmanned aerial vehicles in environmental remote sensing: Challenges and opportunities* (2011)
8. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer (2009)
9. Hauert, S., Leven, S., Varga, M., Ruini, F., Cangelosi, A., Zufferey, J., Floreano, D.: Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5015–5020. IEEE (2011)
10. Holland, G., Webster, P., Curry, J., Tyrell, G., Gauntlett, D., Brett, G., Becker, J., Hoag, R., Vaglianti, W.: The aerosonde robotic aircraft: A new paradigm for environmental observations. *Bulletin of the American Meteorological Society* 82(5) (May 2001)
11. Huhn, S., Mohseni, K.: Cooperative control of a team of AUVs using smoothed particle hydrodynamics with restricted communication. In: *Proceedings of the ASME 28th International Conference on Ocean, Offshore and Arctic Engineering*, Honolulu, HA, May 31–June 5, OMAE 2009-79869 (2009)
12. Jadbabaie, A., Lin, J., Morse, A.S.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control* 48(6) (2003)
13. Jensen, J.R.: *Remote Sensing of the Environment: An Earth Resource Perspective*, 2nd edn. Prentice Hall (May 2006)
14. Lawrence, D., Frew, E., Pisano, W.: Lyapunov vector fields for autonomous unmanned aircraft flight control. *Journal of Guidance, Control, and Dynamics* 31(5), 1220–1229 (2008)
15. Pimenta, R.M.L.C.A., Mendes, M.L., Pereira, G.: Fluids in electrostatic fields: An analogy for multirobot control. *IEEE Transactions on Magnetics* 43(4) (2007)
16. Leven, S., Zufferey, J.-C., Floreano, D.: Dealing with Mid-Air Collisions in Dense Collective Aerial Systems. *Journal of Field Robotics* 28(3), 405–423 (2011)
17. Lipinski, D., Mohseni, K.: Cooperative control of a team of unmanned vehicles using smoothed particle hydrodynamics. AIAA paper 2010-8316, AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, Canada, August 2-5 (2010)
18. Lipinski, D., Mohseni, K.: A master-slave fluid cooperative control algorithm for optimal trajectory planning. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 3347–3351. IEEE (2011)
19. Liu, G., Liu, M.: Smoothed particle hydrodynamics: a meshfree particle method. World Scientific (2003)
20. Lloyd, S.P.: Least square quantization in pcm. *IEEE Transactions on Information Theory* 28(2), 129–137 (1982)
21. MacKay, D.: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, New York (2003)
22. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297 (1967)
23. Monaghan, J.: Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30, 543–574 (1992)
24. Moody, J., Darken, C.J.: Fast learning in networks of locally tuned processing units. *Neural Computation* 1, 281–294 (1989)

25. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control* 51(3), 401–420 (2006)
26. Pimenta, L., Michael, N., Mesquita, R., Pereira, G., Kumar, V.: Control of swarms based on hydrodynamic models. In: *IEEE International Conference on Robotics and Automation* (May 2008)
27. Pimenta, L., Michael, N., Mesquita, R., Pereira, G., Kumar, V.: Control of swarms based on hydrodynamic models. In: *IEEE International Conference on Robotics and Automation*, pp. 1948–1953 (May 2008)
28. Pisano, W.J., Lawrence, D.A.: Autonomous UAV control using a 3-sensor autopilot. *AIAA paper 2007-2756*, American Institute of Aeronautics and Astronautics, Infotech Aerospace Conference, Rohnert Park, CA (May 2007)
29. Richards, M., Scheer, J., Holm, W.: *Principles of Modern Radar Basic Principles*. SciTECH Publishing Inc. (2010)
30. Shaw, A., Mohseni, K.: A fluid dynamic based coordination of a wireless sensor network of unmanned aerial vehicles: 3-d simulation and wireless communication characterization. *IEEE Sensors Journal* 11(3), 722–736 (2011)
31. Spiess, T., Bange, J., Buschmann, M., Vörsmann, P.: First application of the meteorological Mini UAV M2AV. *Meteorologische Zeitschrift* 16(2), 159–169 (2007)
32. van den Kroonenberg, A., Martin, S., Beyrich, F., Bange, J.: Spatially-Averaged Temperature Structure Parameter Over a Heterogeneous Surface Measured by an Unmanned Aerial Vehicle. *Boundary-Layer Meteorology* 142, 55–77 (2011)