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

Optimal multi-agent path planning for fast inverse modeling in UAV-based flood sensing applications

Mohamed Abdelkader, Mohammad Shaqura, Mehdi Ghommem, Nathan Collier,
Victor Calo and Christian Claudel
Department of Electrical Engineering and Computer Sciences, KAUST, Saudi Arabia
christian.claudel@kaust.edu.sa

***Abstract*— Floods are the most common natural disasters, causing thousands of casualties every year in the world. In particular, flash flood events are particularly deadly because of the short timescales on which they occur. Unmanned air vehicles equipped with mobile microsensors could be capable of sensing flash floods in real time, saving lives and greatly improving the efficiency of the emergency response. However, of the main issues arising with sensing floods is the difficulty of planning the path of the sensing agents in advance so as to obtain meaningful data as fast as possible. In this particle, we present a fast numerical scheme to quickly compute the trajectories of a set of UAVs in order to maximize the accuracy of model parameter estimation over a time horizon. Simulation results are presented, a preliminary testbed is briefly described, and future research directions and problems are discussed.**

I. Introduction

Floods are one of the most commonly occurring natural disasters, and caused more than 120,000 fatalities in the world between 1991 and 2005 [1]. They are a major problem in many areas in the world, and are expected to become worse due to global warming, which causes more extreme weather events around coastal areas. In 2010, floods were not only responsible for more than 4000 deaths worldwide but also caused considerable economic loss. The 2009 Jeddah floods claimed hundreds of lives and caused hundreds of millions of dollars of property damage, with for instance more than 10,000 vehicles lost. While these natural disasters are unavoidable, the loss of lives can be minimized by a proper warning system, which can also give emergency responders real time data to organize their operations.

Though rain monitoring systems have been used for flood prediction and estimation before [4], flood caused by extreme rains cannot be accurately predicted with these systems, as flood propagation models require a large number of parameters which are difficult to know beforehand. Similarly, fixed water level sensors are only adapted to river monitoring, and are unsuitable to desert environments in which the location and the extent of a flood cannot be estimated reliably. In a large scale coastal city such as Jeddah, the surface of the hydrological basin to monitor is in the orders of thousands of square kilometers, as illustrated in Figure 1, which makes a fixed monitoring infrastructure economically infeasible.

A recent effort to directly measure water levels in a river was investigated in [3], but this technology only applies to rivers and cannot monitor large hydrological basins in which new water channels that cannot be predicted in advance are formed during floods. Other such efforts involve the use of indirect measurements from rain stations or from meteorological data, combined with flood propagation models that attempt to predict flood parameters such as the extension of flooded areas, water velocities and levels. The main drawback with this approach is the lack of accurate model parameters. The inaccuracy of these model parameters can drastically change the predicted flood behavior, leading to a very unreliable flood warning system.

A new form of sensing known as *Lagrangian sensing* makes the use of mobile (floating) sensors, and has been recently investigated in the context of hydrological sensing in [24]. Lagrangian sensing is very promising for large scale sensing, or on demand sensing, as it requires minimal infrastructure. While operating costs during sensing operations are higher in Lagrangian sensors than in their *Eulerian* (fixed sensors) counterparts, the relatively rare occurrence of floods makes the Lagrangian sensing very suitable to flood monitoring. An added economic benefit is the possibility to redeploy the mobile sensors on demand.

The latest advances in unmanned aerial vehicles (UAVs) significantly improved their reliability and increased their application range [19]. As a result, an abundance of UAV-assisted projects has been witnessed recently. A Global Hawk UAV was deployed in Japan to assess the damage inside a nuclear plant in the aftermath of an earthquake [6]. A European project has been launched for the exploration and measurement of volcanic activities [18]. Adams et al. [2] presented a damage assessment-based UAV system for hurricane events. All of these recent applications leave no doubt that UAVs will play a major role in future remote sensing technology. This article contributes to the efforts of developing reliable real-time UAVs-based disaster monitoring systems.

This project proposes the use of UAVs as a platform for Lagrangian flood sensing using microsensors [5]. In this framework, a swarm of UAVs would drop small disposable wireless sensors over the areas to monitor. These wireless sensors would be buoyant, and would be carried away by the flood. UAVs will receive signals from these sensors, and will map the extent of the flood, transmitting back this data in real time to a fixed ground station for processing (in particular estimation, forecast or inverse modeling). This direct measurement data will provide the authorities with

Fig. 1. **Satellite view of Jeddah hydrological basin.** This picture represents a 70 km × 30 km area surrounding the city of Jeddah, Saudi Arabia (credit: Google Earth).

a real-time flood map and accurate short term forecasts of the flood propagation, allowing more efficient emergency response and saving lives. In addition, this direct measurement data will allow an accurate estimation of local flood parameters (inverse modeling), which will in turn increase the accuracy of flood prediction for future occurrences.

All real-time disaster monitoring and forecast systems operate until severe real-time constraints. In the context of UAV based sensing, the UAVs must fly as fast as possible to the area to monitor (as soon as a severe rainfall is forecasted). Once the location to monitor is reached, the UAVs need to quickly plan a sensing strategy. The problem of optimal sensor placement [17] has been extensively studied in the literature using multiple approaches including MDPs [21] in the context of ground target sensing, GPs in the context of wireless sensor networks in [17], [16] or EnKF based approaches in conjunction with machine learning in [15], [9]. However, the latter methods (which would be best suited to our system) can be very slow, as they require a large amount of complex matrix computations for each possible UAV trajectory. In the present case, we propose a different approach to the problem based on linear least squares applied to a polynomial chaos expansion of the ensembles. Using a database of such expansions, the trajectory computations could be done at the level of the UAV swarm itself for practical problems, greatly simplifying the architecture and improving the reliability of the system in case of temporary loss of communication.

The rest of this article is organized as follows. Section II introduces the concept of UAV-based Lagrangian sensing for real-time flood monitoring, which is modeled in section III. We then introduce the cooperative control scheme used for planning the trajectories of the UAVs over some time horizon, to minimize the uncertainty on the estimation of the unknown model parameters (inverse modeling). Using simulated flood data, we apply this cooperative control scheme on a practical example to compute the optimal UAV trajectories. An ongoing implementation of the concept using autonomous UAVs is then briefly presented.

II. Proposed flood sensing system

A. System overview and architecture

In our proposed system [5], a swarm of UAVs equipped with an array of droppable (disposable) microsensors is sent over the area to monitor, *i.e.* an hydrological basin on which a flood could occur. The swarm of UAVs would only be launched on the few days during which flooding could occur, based on rain forecasts over the region, or based on other direct measurements (for instance electronic rain gauges). The microsensors, also known as *Lagrangian microsensors* would emit a unique ID (similarly to an active RFID tag) periodically as soon as they are released from the aircraft, until battery exhaustion. After their fall, these microsensors would settle on the ground, and can either remain static, or be carried away by a flood. The microsensors are designed to be relatively insensitive to wind, having a relatively high mass to surface ratio. After dropping the transmitters, the UAVs would track their evolutions using a passive receiving antenna. Multiple UAVs can be used to collaboratively monitor a given area, enabling a map of these transmitters to be quickly established.

UAVs will also have radio links between themselves, and between a ground control station. They will share sensor location updates, as well as sensor drop events. UAVs will also receive updates of the current flood estimates (water level, water velocity, as well as uncertainty on both variables) from the ground control station, which will enable them to plan their response optimally, for instance by sensing in areas in which the uncertainty is high. Obviously, the initial sensing locations will be based on a map of probability of flood occurrence, which can be generated using historical flooding data. In return, UAVs will send to the ground station the current microsensor map, which will be used as input data (together with precipitation data) for the flood estimation and forecast algorithms.

Finally, the inter UAV communications will serve as a backup communication to the base station should the direct link fail. Note that this is expected to occur frequently as UAVs will patrol over valleys, and will thus not necessarily be in line of sight of the ground control station.

The complete system is illustrated in Figure 2.

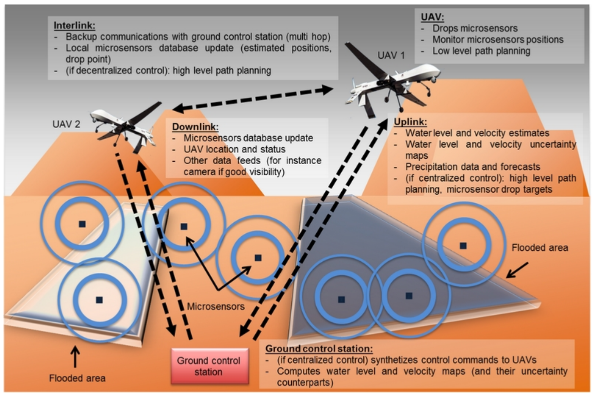

Fig. 2. **Proposed UAV sensing system architecture.** This figure represents the interactions between the UAVs and the ground control station during flood monitoring operations.

B. *UAV optimal path planning*

One of the issues arising when operating multiple UAVs for flood detection is the high-level path planning. In this article, our focus is to develop a numerical scheme that is as fast as possible, for multiple reasons:

- The set of all possible trajectories of all UAVs is increasing exponentially, therefore the computational power required to evaluate the cost of a set of trajectories (with respect to the optimization criterion chosen) is critical.
- Trajectories may have to be recomputed on the spot due to constraints such as wind changes affecting the reachable sets of the UAVs or new weather updates. Relying solely on a supercomputer for the computation of these trajectories could lead to system failures, specially since the UAVs monitor catastrophic events (floods) that could also affect the supercomputer (power loss, communication loss).

The objective of the proposed path planning algorithm is to rapidly compute the optimal trajectories using flood propagation models. Specifically, we want to solve the *inverse modeling problem* as quickly as possible, *i.e.* to estimate the value of the unknown model parameters affecting the solution within minutes. This will give a quick preliminary overview of the current flood characteristics (amplitude, approximate location, terrain parameters) which can be used by emergency responders instantly, even if the precise location of flooded areas and their evolution are still uncertain.

III. Flood modeling

Flooding is well modeled by the shallow water equations (SWE) [8], which are obtained by integrating the Navier Stokes equations in depth, assuming that the horizontal scales are much larger than the vertical scales. This assumption is valid during flood events, as the horizontal scales are in the order of kilometers, while the vertical scales are at most a few tens of centimeters.

The diffusive wave approximation to the shallow water equations (DSW,[8], [7], [23]) is a simplification of the shallow water equations, used to model flows where the vertical momentum is small relative to the horizontal. The strong form is obtained by assumptions which simplify the shallow water equations, leading to the following initial/boundary value problem on the spatial domain Ω for times $t \in [0,T]$

$$\begin{cases} \dot{u} - \nabla \cdot (\kappa(u, \nabla u)\nabla u) = f & \text{on } \Omega \times (0,T] \\ u = u_0 & \text{on } \Omega \times \{t=0\} \\ (\kappa(u, \nabla u)\nabla u) \cdot n = B_N & \text{on } \Gamma_N \times (0,T] \\ u = B_D & \text{on } \Gamma_D \times (0,T] \end{cases} \tag{1}$$

In the above equation, u denotes the water level relative to a datum, $\dot{u}$ represents its time derivative, f is a forcing function such as rainfall acting as a source or infiltration in the ground acting as a sink. The function $u_0(\cdot)$ represents the water level at the initial time, while B_N and B_D are respectively the Neumann and Dirichlet conditions.

The PDE constitutes a nonlinear diffusion problem where the diffusion coefficient κ is given by

$$\kappa(u, \nabla u) = \frac{(u-z)^{\alpha_M}}{C_f |\nabla u|^{1-\gamma_M}}.$$

In the above formula, the function z represents the elevation of the terrain, and $\alpha_M = \frac{5}{3}$ and $\gamma_M = \frac{1}{2}$ are constants which correspond to Manning's formula, used in the DSW derivation [10]. The constant function $C_f = 0.009$ represents Manning's coefficient. See [10] for details on the computational approach used.

IV. Trajectory planning for optimal flood sensing

A. *Motivations*

One of the greatest challenges arising when operating the UAV flood sensing system introduced before is the planning of the UAV paths. One of the possibilities to solve such a problem would be to start with the prior estimate of the EnKF (or any other data assimilation scheme) and send the UAVs in the areas for which the uncertainty is the largest. However, since UAVs have limited capabilities (maximal velocity, turning rate, etc), this strategy may not help much in practice as the time spent by the UAV flying between different locations with current maximal uncertainty could be high. In addition, once an UAV picks a location and sense the water velocity at this location, the location with the current maximal uncertainty may change unpredictably, which could be inefficient in terms of sensing.

Our idea is to work on inverse modeling (model parameter estimation) rather than data assimilation, as the area is typically not flooded when the UAVs are first deployed, therefore the initial conditions (initial water level) have no uncertainty. The uncertainty is thus only in the model parameters: terrain properties, rain rates and locations. For all these parameters, the system starts with a prior:

- The terrain properties (friction coefficients, runoff coefficients...) are tabulated. Their value is however not precisely known, with standard deviations on the order of half of the mean value of more [13] depending on the surface subtype.
- The rain parameters (rain rate as a function of time and space). The value of these parameters is inferred from short-term weather forecasts, obtained just before the UAVs are launched.

While technically an infinite of number of the above parameters is necessary to describe the evolution of the flood, in practice the friction and runoff coefficients can be taken as piecewise constant over large areas of the hydrological maps. Similarly, the rain rate dependency over time can be inferred from precipitation models, and the spatial rain distribution can be modeled for instance as a sum of Gaussians (whose centers correspond to the peak activity areas as detected by satellites or weather radars). Thus, the main features of the flood can be captured with a limited number of parameters. While the ultimate goal is to estimate the state of the system (water levels and velocities), our objective is first to estimate the flood parameters as early as possible. This data will help simulating a realistic scenario of the current flood, which can be sent to the public or to emergency responders.

B. Problem definition

Let us consider a discrete spatial domain

$$\mathcal{D} = [0, \ldots, n_{grid}\delta x] \times [0, n_{grid}\delta y]$$

on which we compute the solution to the diffusive shallow water equation (1). We assume that the solution to the diffusive shallow water equation depends on a finite number of parameters $(\alpha_1, ..., \alpha_k)$. These parameters can for instance represent friction coefficients or runoff coefficients of some areas of the terrain, or parameters of the rain rate (dependency with space and time).

Let us assume that a set of UAVs $\{UAV_1, \ldots, UAV_j\}$ is initially located at the respective initial positions $(x_1(0), y_1(0)), \ldots, (x_j(0), y_j(0))$. For simplicity, we assume a constant time step δ_T, and assume that each UAV drops a microsensor at each time step and measure its velocity with uncertainty $(\delta v_x(k), \delta v_y(k))$ during a sensing period τ.

Our objective is to determine the trajectories of each UAV so as to minimize the uncertainty of model parameter estimation at the time horizon $h\delta T$. Specifically, if P is a positive definite matrix, we want to minimize $\delta(\alpha)^T P \delta(\alpha)$ at the final time. In all subsequent simulations we considered $P = I$ for simplicity.

C. First order polynomial chaos expansion

To quantify the variations of the velocities due to uncertainties in the system parameters, we employ the polynomial chaos expansion (PCE). In this technique, the spectral representation of varying parameters is based on the decomposition of a random function (or variable) into separable deterministic and stochastic components [22], [14]. Here, for example, any generic variable α^* is expressed as:

$$\alpha^*(x, t, \xi) \simeq \sum_{i=0}^{P} \alpha_i(x, t)\Psi_i(\xi) \quad (2)$$

where $P+1 = \frac{(n+p)!}{n!p!}$ is the number of output modes, which is a function of the order of the polynomial chaos p and the number of random parameters n. $\alpha_i(t, x)$ is the deterministic component which is the amplitude of i^{th} fluctuation and $\Psi_i(\xi)$ is the random basis function corresponding to the i^{th} mode. It is worth pointing out that α^* is taken as a function of independent deterministic variables x and t, and the n-dimensional random variable vector $\xi = (\xi_1, ..., \xi_n)$ which has a specific probability distribution.

In practice, the series will be approximated by the finite-term summation in which the highest order terms of the polynomials are set according to the accuracy requirement. For the basis function, multi-dimensional Hermite polynomials are taken to span the n-dimensional random space which follows Gaussian probability distribution. It is well known that Hermite polynomials form an orthogonal set of basis functions in terms of Gaussian distribution [11]. Many other choices are possible for basis functions depending on the type of the probability distribution selected for the representation of variations in input parameters [25], [12], [20]. Any Hermite polynomial in the n-th dimensional random space is given by [11]

$$H_p(\xi_1, ..., \xi_n) = (-1)^p e^{\frac{1}{2}\xi^T\xi} \frac{\partial^p}{\partial(\xi_1)^{c_1} \ldots \partial(\xi_n)^{c_n}} e^{-\frac{1}{2}\xi^T\xi} \quad (3)$$

where $\sum_{k=1}^{n} c_k = p$, and c_ks are integers. These polynomials form a complete orthogonal set of basis functions in the random space.

An alternative way to obtain the coefficients of the polynomial expansion $\alpha(x, t)$ has been presented by Hosder et al. [14]. Such method is based on the following matrix equation

$$\begin{Bmatrix} \alpha_0^* \\ \alpha_1^* \\ \vdots \\ \alpha_N^* \end{Bmatrix} = \begin{bmatrix} \Psi_1(\xi^0) & \Psi_2(\xi^0) & \cdots & \Psi_P(\xi^0) \\ \Psi_1(\xi^1) & \Psi_2(\xi^1) & \cdots & \Psi_P(\xi^1) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_1(\xi^N) & \Psi_2(\xi^N) & \cdots & \Psi_P(\xi^N) \end{bmatrix} \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_P \end{Bmatrix} \quad (4)$$

which represents the discretised form of equation (2). If the number of samples N is selected to be equal to the number, $P+1$, of polynomials in the expansion, the matrix in equation (4) is square, and can be inverted to obtain the expansion coefficients α_i from the outputs α_i^*. However, as pointed out in [14], the solution obtained, when following this approach, is not unique due to the arbitrariness in the choice of the sampling vector. This fact might lead to errors in the estimation of the PCE coefficients α_is. Because in the present work, the number of samples N is higher than the number of polynomials in the expansion $P+1$, the system of equations (4) is solved by minimising the errors in the least squares sense.

Using a first order polynomial chaos expansion, we then write the quantities $v_x(i\delta x, j\delta y, k\delta T)$ and $v_x(i\delta x, j\delta y, k\delta T)$ as linear functions of α for all $(i,j,k) \in [1, n_{grid}] \times [1, n_{grid}] \times [1, h]$:

$$\begin{aligned} v_x(i\delta x, j\delta y, k\delta T) &= v_{x,0}(i,j,k) + c_x(i,j,k)^T\alpha \\ v_y(i\delta x, j\delta y, k\delta T) &= v_{y,0}(i,j,k) + c_y(i,j,k)^T\alpha \end{aligned} \quad (5)$$

This description of v_x and v_y allows us to capture the main effects of the unknown model parameters on the solution. While a higher order chaos expansion could be used, our algorithm leverages the existence of explicit solutions to linear least squares problems, which reduces the computational complexity of the optimization problem.

D. Formulation as combinatorial optimization

Let us now denote by $\mathcal{T}$ the set of possible trajectories of the UAVs. $\mathcal{T}$ is a subset of $\{0, \dots, n_{grid}\}^{2\cdot h\cdot j}$, and can be computed from the dynamical models of the UAVs, using classical reachability algorithms. Assuming that the trajectories $(traj_1, \dots, traj_j) \in \mathcal{T}$ are given, we obtain $2{\cdot}h{\cdot}j$ velocity measurements. From equation 5, and assuming that the measurements are known, the parameters α are solution to the following linear system:

$$\begin{bmatrix} c_x(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ c_y(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ \cdots \\ \cdots \\ c_x(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \\ c_y(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \cdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} v_x(i_{traj_1(1)}, j_{traj_1(1)}, 1)^{\text{meas}} \\ v_y(i_{traj_1(1)}, j_{traj_1(1)}, 1)^{\text{meas}} \\ \cdots \\ \cdots \\ v_x(i_{traj_j(h)}, j_{traj_j(h)}, h)^{\text{meas}} \\ v_y(i_{traj_j(h)}, j_{traj_j(h)}, h)^{\text{meas}} \end{bmatrix} \quad (6)$$

The above equation is overdetermined whenever $2j{\cdot}h > k$. In this situation, the vector of parameters $[\alpha_1, \dots, \alpha_k]$ can be obtained by an approximate solution to this linear system, for instance using linear least squares.

For trajectory planning considerations, we are interested in minimizing the uncertainty on parameter vector estimation. For this, we assume for simplification that the polynomial chaos expansion has no uncertainty, and therefore that $c_x(i,j,k)$ and $c_y(i,j,k)$ are perfectly determined for all (i,j,k).

Owing to the linear relationship between parameters and measurements in (6), one can explicitly write the least squares solution to this problem as:

$$\begin{bmatrix} \alpha_1 \\ \cdots \\ \alpha_k \end{bmatrix} = \left(\begin{bmatrix} c_x(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ c_y(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ \cdots \\ \cdots \\ c_x(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \\ c_y(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \end{bmatrix}^T \times \begin{bmatrix} c_x(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ c_y(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ \cdots \\ \cdots \\ c_x(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \\ c_y(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \end{bmatrix} \right)^{-1} \begin{bmatrix} c_x(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ c_y(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ \cdots \\ \cdots \\ c_x(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \\ c_y(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \end{bmatrix} \times \begin{bmatrix} v_x(i_{traj_1(1)}, j_{traj_1(1)}, 1)^{\text{meas}} \\ v_y(i_{traj_1(1)}, j_{traj_1(1)}, 1)^{\text{meas}} \\ \cdots \\ \cdots \\ v_x(i_{traj_j(h)}, j_{traj_j(h)}, h)^{\text{meas}} \\ v_y(i_{traj_j(h)}, j_{traj_j(h)}, h)^{\text{meas}} \end{bmatrix} \quad (7)$$

Since the right hand side of equation (6) is uncertain, the set of least squares solution to (6) for all values of v^{meas} within the uncertainty bound

$$b_{\min} \leq \begin{bmatrix} v_x(i_{traj_1(1)}, j_{traj_1(1)}, 1)^{\text{meas}} \\ v_y(i_{traj_1(1)}, j_{traj_1(1)}, 1)^{\text{meas}} \\ \cdots \\ \cdots \\ v_x(i_{traj_j(h)}, j_{traj_j(h)}, h)^{\text{meas}} \\ v_y(i_{traj_j(h)}, j_{traj_j(h)}, h)^{\text{meas}} \end{bmatrix} \leq b_{\max}$$

is the polyhedron:

$$\begin{aligned} &\{x \in \mathbb{R}^k | (A^TA)^{-1}A^Tx \leq b_{\max}\} \\ \cap &\{x \in \mathbb{R}^k | (A^TA)^{-1}A^Tx \geq b_{\min}\} \end{aligned} \quad (8)$$

where $A = \begin{bmatrix} c_x(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ c_y(i_{traj_1(1)}, j_{traj_1(1)}, 1)^T \\ \cdots \\ \cdots \\ c_x(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \\ c_y(i_{traj_j(h)}, j_{traj_j(h)}, h)^T \end{bmatrix}$

To quantify the uncertainty on the model parameters $[\alpha_1, \dots, \alpha_k]$, we choose to use the weighted norm of $(A^TA)^{-1}A^T(b_{\max} - b_{\min})$, though other functions of the size of (8) could be utilized. Note that $b_{\max} - b_{\min}$ is usually known beforehand (even if b is not), as it corresponds to the vector of uncertainties in the velocity measurements. The error range when measuring a velocity is assumed to be absolute, *i.e.* it is not a function of the measured velocity.

The trajectory optimization problem can therefore be formulated as:

$$\min_{(traj_1, \dots, traj_k) \in \mathcal{T}} || \left(A(traj_1, \dots, traj_k)^T A(traj_1, \dots, traj_k) \right)^{-1} \times A(traj_1, \dots, traj_k)^T (b_{\max} - b_{\min}) || \quad (9)$$

In practice, the implementation follows the procedure described in Figure 3 below.

E. Simulation

We now simulate the performance of the proposed algorithm on a practical flooding scenario over the terrain described in Section III. We assume that a single cloud is

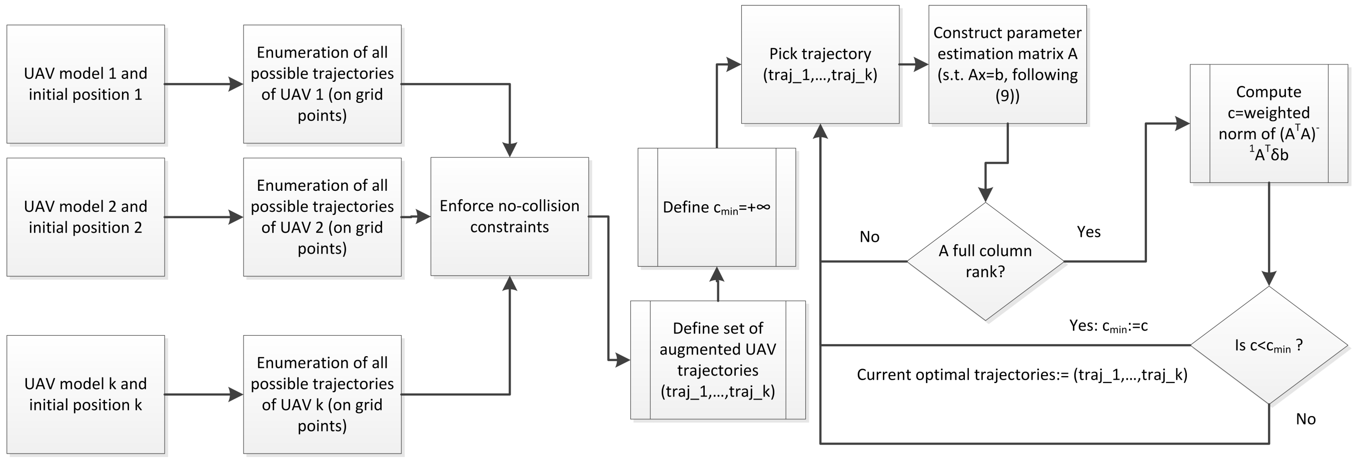

Fig. 3. **Trajectory planning principle.** This figure summarizes the method used by the trajectory planner.

located above the simulated area, with some uncertainty on its position and its corresponding rain rate, and study the flooding problem over a period of 30 minutes. The flood propagation model is described by four main parameters:

- rain rate (average value 290 mm/hour, with standard deviation of 15 %)
- x position of the cloud (average value of 1000 meters, with standard deviation of 15 %)
- y position of the cloud (average value of 750 meters, with standard deviation of 15 %)
- Manning's coefficient (average value of 0.03, with standard deviation of 15 %)

To reduce the computational complexity, we describe the domain by 2500 grid points (uniform 50×50 rectangular grid) and 20 ensembles. We assume that the UAVs can reach any location in a circle of radius $v(\delta T - \tau)$, where $v = 10m/s$, $\delta T = 90s$ and $\tau = 60s$. The parameter τ physically represents the duration required by the UAV to estimate the velocity of the Lagrangian sensor before moving to the next sensing location. Note that more complex reachable sets could be used (for instance to take into account the wind effects) without adversely affecting the computational time, which only depends upon the number of points that can be reached by the UAV within a time step.

The results were simulated on a `Macbook Pro` with `Intel i5 2.5 GHz` processor in `Matlab`. The computation of each ensemble took 15 minutes in average, while the polynomial chaos expansion of the resulting ensembles took 18 minutes. We simulated multiple scenarios for single and multiple agent problems, outlined in Figures 4 and 5 below.

In the case described in Figure 5, the weighted norm of the model parameter decreases from 0.071 (left) to 0.027 (center), and then to 0.011 (right), illustrating the fact that the estimation of the model parameters improves with additional sensing. The computed trajectories of the UAVs are also within the flood prone area (canyon), since the upper areas are not likely to be flooded.

V. CURRENT IMPLEMENTATION

A. UAV system overview

We chose for this project a Tiansheng C-17 Globemaster remote controlled aircraft, with four electric ducted fans illustrated in Figure 6. We equipped this airframe with an `Ardupilot Mega` (APM) 2.0 microcontroller, as well as different sensors: Pitot tube, barometer, magnetometer, inertial measurement unit (IMU) and ultrasound ground proximity sensor. To enable more advanced computations, we also equipped the airframe with a `Gumstix Overo Earth` computer on module, connected to the APM 2.0 using USB.

The sensor release system is currently implemented using 3D printing, and the microsensors themselves are 3D printed to accelerate the prototyping phase. Based on the simulations in `Matlab` described above, the `Gumstix Overo Earth` should be capable of determining the optimal trajectories of a set of two UAVs on a 3 minute horizon in real time. Since the cost if inverting a matrix is polynomial in its size, the complete trajectory planning algorithm is polynomial in the number of parameters of the model, which should enable the resolution of more complex problems involving a large number of model parameters (for instance the runoff coefficients of multiple areas of the computational domain and/or the size of multiple clouds) in real time on commercial embedded platforms.

VI. CONCLUSION

In this article, we presented a fast inverse modeling method applicable to UAV-based flood sensing. Using a flood propagation model with a finite set of unknown parameters, we first computed the evolution of a set of ensembles, and applied first order polynomial chaos expansion to compute the dependency of the solution with respect to the parameters at each point and for all time steps. We then show that the trajectory optimization problems for the multi-agent system boils down to solving a linear least squares problem for each possible trajectory. This algorithm is then validated on simulated flood data. Preliminary results show that practical

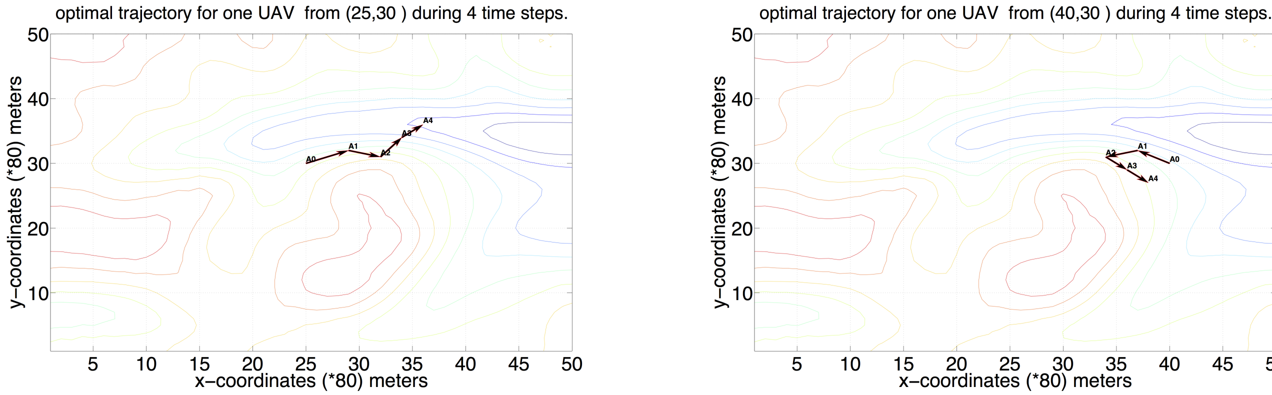

Fig. 4. **Trajectory computation for a single agent.** This figure shows the optimal trajectory for a single UAV starting at location (25,30) (left) and time t=22.5 min, and a single UAV starting at location (40,30) (right) at t=22.5 min.

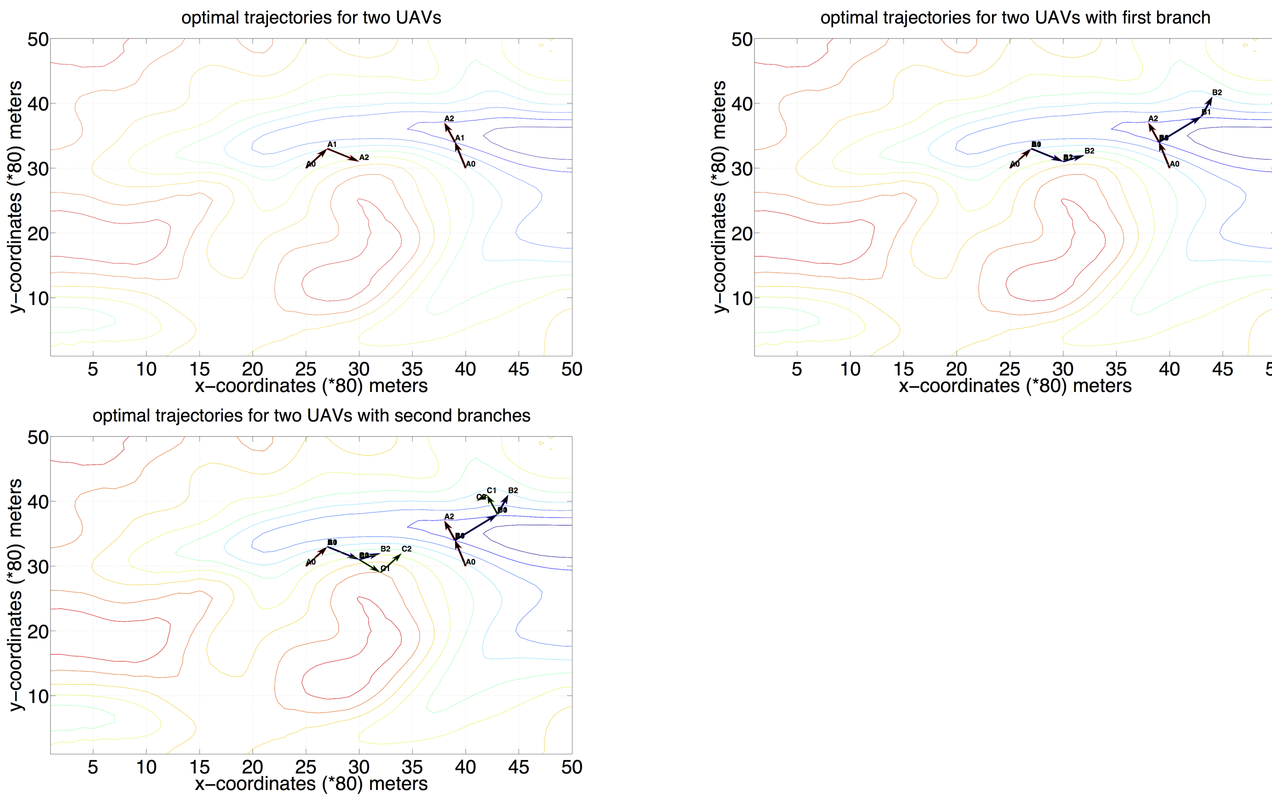

Fig. 5. **Trajectory computation for two agents.** This figure show the trajectories of two UAVs sensing the area collaboratively. Each trajectory is computed at every time step (90s) over a two time step horizon (180s). The two step optimization problem can be computed in real time (3 minutes) on a `Macbook Pro` laptop.

problems can be solved almost in real time on regular laptops, which implies that the UAVs themselves could carry out the computations, greatly enhancing the reliability of the system in case of server, communication or power failures. Future work will deal with the full use of the Lagrangian sensing capabilities of the system, in which previously deployed sensors are re-tracked along their path as part of the sensing activities. This would greatly speed up the estimation process, though it would require the position of the deployed sensors to be inferred based on the currently available flood parameter estimates.

REFERENCES

[1] ISDR disaster statistics, `http://www.unisdr.org/`.

[2] S. Adams, C. Friedland, and M. Levitan. Unmanned Aerial Vehicle Data Acquisition for Damage Assessment in Hurricane Events. In *8th International Workshop on Remote Sensing for Disaster Management*, Tokyo, Japan, 2010.

[3] E. Basha, S. Ravela, and D. Rus. Model-based monitoring for early warning flood detection. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 295–308. ACM, 2008.

[4] L. Li, Y. Hong, J. Wang, R. Adler, F. Policelli, S. Habib, D. Irwin, T. Korme, and L. Okello. Evaluation of the real-time TRMM-based multi-satellite precipitation analysis for an operational flood prediction system in Nzoia Basin, Lake Victoria, Africa. *Natural hazards*, 50(1):109–123, 2009.

[5] M. Abdelkader, M. Shaqura, C. Claudel, and W. Gueaieb. A UAV

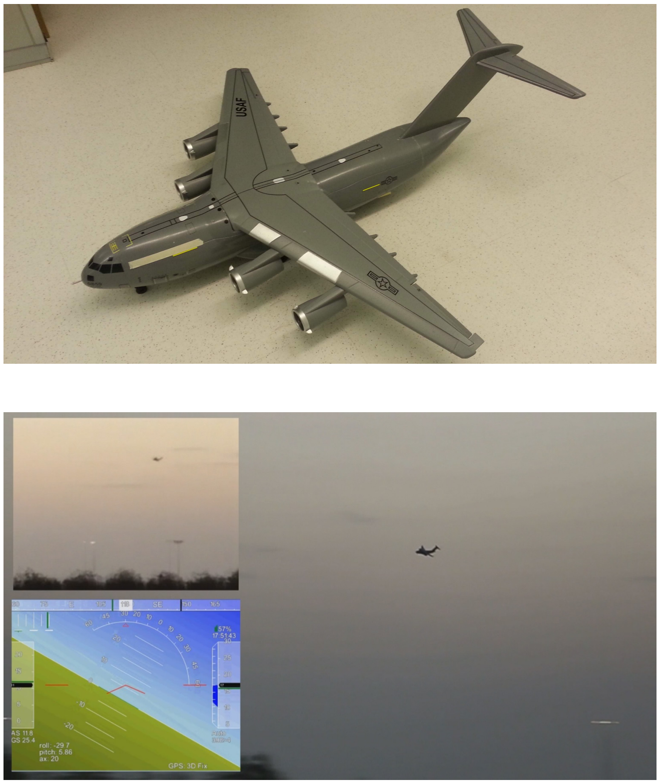

Fig. 6. **Upper figure:** a picture of the UAV model used. **Lower figure:** UAV during autonomous flight at 11 m/s, during a flight test aimed at validating the possibility of dynamically rerouting the UAV.

based system for real time flash flood monitoring in desert environments using Lagrangian microsensors. In *ICUAS, 2013*, pages 25–34, 2013.

[6] E. Ackerman. Japan Earthquake: Global Hawk UAV May Be Able to Peek Inside Damaged Reactors. *IEEE Spectrum*, 17 March, 2011.

[7] R. Alonso, M. Santillana, and C. Dawson. On the difussive wave approximation of the shallow water equations. *J. Appl.Math.*, 19(5):575–606, 2008.

[8] R. J. Alonso, M. Santillana, and C. Dawson. Analysis of the diffusive wave approximation of water equations. *J. Appl.Math.*, 2007.

[9] H.-L. Choi and J. P. How. A multi-uav targeting algorithm for ensemble forecast improvement. In *AIAA Guidance, Navigation, and Control Conference*, 2007.

[10] N. Collier, H. Radwan, L. Dalcin, and V. M. Calo. Diffusive wave approximation to the shallow water equations: Computational approach. *Procedia Computer Science*, 4(0):1828 – 1833, 2011.

[11] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Dover Publications, Inc., 2003.

[12] M. Ghommem, M. R. Hajj, C. L. Pettit, and P. S. Beran. Stochastic modeling of incident gust effects on aerodynamic lift. *Journal of Aircraft*, 47:1720–1727, 2010.

[13] L. Gottschalk and R. Weingartner. Distribution of peak flow derived from a distribution of rainfall volume and runoff coefficient, and a unit hydrograph. *Journal of Hydrology*, 208(34):148 – 162, 1998.

[14] S. Hosder, R. W. Walters, and P. Rafael. A non-intrusive polynomial chaos method for uncertainty propagation in cfd simulations. AIAA Paper 2006-891, AIAA, 2006. 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.

[15] J. P. How, N. Roy, H.-L. Choi, and S. Park. Learning covariance dynamics for path planning of uav sensors in a large-scale dynamic environment. 2013.

[16] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 2–10. ACM, 2006.

[17] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[18] G. Muscato, E. Bonaccorso, L. Cantelli, D. Longo, and C. Melita. Volcanic environments: Robots for exploration and measurement. *IEEE Robotics & Automation Magazine*, 19(1):40–49, 2012.

[19] E. Pereira, C. M. Kirsch, R. Sengupta, and J. B. de Sousa. Bigactors-a model for structure-aware computation. In *4th International Conference on Cyber-Physical Systems. ACM/IEEE*, 2013.

[20] C. L. Pettit, M. R. Hajj, and P. S. Beran. A stochastic approach for modeling incident gust effects on flow quantities. *Probabilistic Engineering Mechanics*, 25:153–162, 2010.

[21] S. Ragi and E. K. Chong. Uav path planning in a dynamic environment via partially observable markov decision process. *Aerospace and Electronic Systems, IEEE Transactions on*, 49(4):2397–2412, 2013.

[22] M. Reagan, H. N. Najm, R. G. Ghanem, and O. M. Knio. Uncertainty quantification in reacting flow simulations through non-intrusive spectral projection. *Combustion and Flame*, 132:545–555, 2003.

[23] M. Santillana and C. Dawson. A numerical approach to study the properties of solutions of the diffusive wave approximation of the shallow water equations. *Journal of Computational Geosciences*, 2009.

[24] A. Tinka, I. Strub, Q. Wu, and A. Bayen. Quadratic programming based data assimilation with passive drifting sensors for shallow water flows. *International Journal of Control*, 83(8):1686–1700, 2010.

[25] D. Xiu and G. E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187:137–167, 2003.