Quadratic Programming based data assimilation with passive drifting sensors for shallow water flows

Andrew Tinka*‡, Issam Strub*, Qingfang Wu†, Alexandre M. Bayen*
* Systems Engineering, Department of Civil and Environmental Engineering
University of California, Berkeley, CA, 94720-1710
† Environmental Engineering, Department of Civil and Environmental Engineering
University of California, Berkeley, CA, 94720-1710
‡ email: tinka@berkeley.edu

***Abstract*— We present a method for assimilating Lagrangian sensor measurement data into a Shallow Water Equation model. Using our method, the variational data assimilation problem is formulated as a Quadratic Programming problem with linear constraints. Data obtained from drifting sensors that gather position and velocity information in the modeled system can then be used to refine the estimate of the initial conditions of the system. The algorithms are implemented using a new sensor network hardware platform for gathering flow information which is also presented in this article. The data assimilation method is used to integrate data collected using the hardware platform. Validation of the results is performed by comparing them to an estimate derived from an independent set of static sensors, some of which were deployed as part of our field experiments.**

I. Introduction

Renewable freshwater is a critical resource for human society. Human uses of freshwater include drinking, irrigation, fish production, transportation, hydroelectric power, and waste disposal; the growing world population, and societal shifts towards urbanization and water-intensive agriculture, will increase freshwater demand significantly over the next fifty years [19]. Improving water use efficiency can help balance supply and demand [13] and relieve scarcity; this will require improved methods for modeling and monitoring the flow of freshwater through the hydrological cycle [20]. As part of this system, river flows are particularly important to investigate, as they constitute the majority of available renewable freshwater [17].

River hydraulics can be modeled with *shallow water equations* (SWE) in one or two dimensions [3]. Shallow water equations are a standard tool used in the environmental engineering community and hydraulics community to model river flow; they are commonly used for simulation and control. When dealing with experimental measurements, techniques are required to incorporate them into the model. One such technique is data assimilation, which is the process of integrating measurements into a flow model, and originated in meteorology and oceanography [5]. Techniques for data assimilation include variational methods [15], Kalman filtering and its extensions [7], optimal statistical interpolation [14], and Newtonian relaxation [18].

There are many different sensor systems for measuring flow fields. They can be categorized as *Eulerian* or *Lagrangian* (using terminology from fluid mechanics) according to whether they observe the medium as it flows past a fixed location (Eulerian) or are embedded into the flow itself, measuring the medium while moving along a trajectory (Lagrangian).

Examples of Eulerian sensors for water velocity measurement are *Acoustic Doppler Current Profilers* (ADCPs), which measure the Doppler shift in a returning acoustic pulse due to velocity; satellite imaging [21], in which the river height is estimated directly with radar or indirectly by observing the water/shore interface; and stream gauges, which measure the height of the water at one location, which can be used to infer the stream velocity under certain conditions.

The trends of electronics miniaturization and availability of wireless communications have increased the interest in novel Lagrangian sensor systems, which sometimes provide an efficent way to complement static sensing infrastructure with mobile devices capable of sensing where fixed equipment cannot be deployed. The estimate of system state is usually more useful in Eulerian coordinates, however, which requires new data assimilation methods to bring the Lagrangian data into an Eulerian context. Most known implementations of Lagrangian data assimilation are in oceanography or meteorology (see for example [9],[16], and [15]); in the specific case of hydraulic systems, Lagrangian data assimilation of shallow water flows to estimate the bottom topology was attempted in [10]. Assimilation of Lagrangian data to estimate boundary conditions in a tidally influenced river was described in [22], and assimilation into a 1D model for a network of channels was described in [28]. An assimilation technique on a simplified hydrodynamic model was presented in [26]; we extend this work with a more realistic model, as well as a complete treatment of the experimental method, numerical schemes, and hardware platform.

Our objective in this article is the development of an integrated system, including hardware, software, communication, and visualization, that is capable of performing data assimilation for shallow water flows using GPS measurements from drifting, Lagrangian sensors. This sensor data is assimilated into a *partial differential equation* (PDE) model of the river, for which, in general, we do not have knowledge of the *initial*

conditions (ICs) or *boundary conditions* (BCs) of the system.

The contributions made in this article are described below:

- A linearization of the (SWE) that can be used for formulating the optimization problem with linear constraints,
- An inversion algorithm, using *Quadratic Programming* (QP), which takes Lagrangian measurements and uses them for reconstruction of the distributed state,
- The construction of a hardware data gathering infrastructure; a floating sensor network used to gather Lagrangian flow data, presented for the first time in this article,
- A field operational test in the Georgiana Slough, and our additional instrumentation deployment for validation purposes,
- Our validation procedure and results.

An earlier version of the linearization and variational assimilation formulation presented herein was developed for shallow water flows in [23], for a different mathematical and operational context.

This article is organized as follows: In Section II, we describe the PDE used to model the hydrodynamical systems under study. In Section III, we explain the QP variational assimilation method we use. In Section IV, we introduce the Lagrangian floating sensor platform for gathering experimental data. In Section V, we describe a field experiment performed in the Sacramento River in California, the results of our assimilation method, and the results of our validation procedure. Finally, in Section VI we present the conclusions of our study and suggest future avenues for research.

II. Hydrodynamic model

A. Shallow water equations

In the following, we use the SWE as our constitutive hydrodynamic model. We will present the equations, followed by a specific linearization and discretization. For legibility we suppress the arguments for dependent variables. The governing hydrodynamic equations for the modeled system are [6], [27]:

$$\frac{\partial u}{\partial t} + \vec{u}\cdot\nabla u = -g\frac{\partial \eta}{\partial x} + F_x + \frac{1}{h}\nabla\cdot(h\nu_t\nabla u) \quad (1)$$

$$\frac{\partial v}{\partial t} + \vec{u}\cdot\nabla v = -g\frac{\partial \eta}{\partial y} + F_y + \frac{1}{h}\nabla\cdot(h\nu_t\nabla v) \quad (2)$$

$$\frac{\partial h}{\partial t} + \vec{u}\cdot\nabla h + h\nabla\cdot\vec{u} = 0 \quad (3)$$

where (x, y) are space coordinates; t is time in seconds; $\vec{u} = (u(x, y, t), v(x, y, t))$ is depth-averaged water velocity in $^{m}/_{s}$; $h = h(x, y, t)$ is water depth in meters; $b = b(x, y)$ is elevation of bottom surface in meters; $\eta = h + b = \eta(x, y, t)$ is free surface elevation in meters; g is the acceleration of gravity in $^{m}/_{s^2}$; ν_t is the coefficient of turbulence diffusion, obeying the so called k-epsilon model [6]; and $F_x = F_x(x, y, t), F_y = F_y(x, y, t)$ are friction terms

$$F_x = -\frac{1}{\cos\theta}\frac{gm^2}{h^{4/3}}u\sqrt{u^2+v^2} \quad (4)$$

$$F_y = -\frac{1}{\cos\theta}\frac{gm^2}{h^{4/3}}v\sqrt{u^2+v^2} \quad (5)$$

where $\theta = \theta(x, y)$ is the slope of the river bed; m is the Manning coefficient. The Manning coefficient is an empirical term that depends on the roughness of the bed, affected by both vegetation and geology. For this study we took the Manning coefficient to be 0.04 uniformly over the domain.

Following common assumptions in fluvial hydraulics, our first simplification is to neglect the turbulence terms. We linearize the equations about a steady flow $U^0(x, y)$, $V^0(x, y)$, $H^0(x, y)$ that satisfies equations (1), (2) and (3):

$$\frac{\partial u}{\partial t} + U^0\frac{\partial u}{\partial x} + V^0\frac{\partial u}{\partial y} = -g\frac{\partial h}{\partial x} - g\frac{\partial b}{\partial x} + Cu \quad (6)$$

$$\frac{\partial v}{\partial t} + U^0\frac{\partial v}{\partial x} + V^0\frac{\partial v}{\partial y} = -g\frac{\partial h}{\partial y} - g\frac{\partial b}{\partial y} + Cv \quad (7)$$

$$\frac{\partial h}{\partial t} + U^0\frac{\partial h}{\partial x} + V^0\frac{\partial h}{\partial y} + H^0\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0 \quad (8)$$

with the choice of

$$C = \frac{1}{\cos\theta}\frac{gm^2}{H^{0\,4/3}}\sqrt{U^{0\,2} + V^{0\,2}} \quad (9)$$

as the linearized friction coefficient.

B. Non-orthogonal curvilinear grid

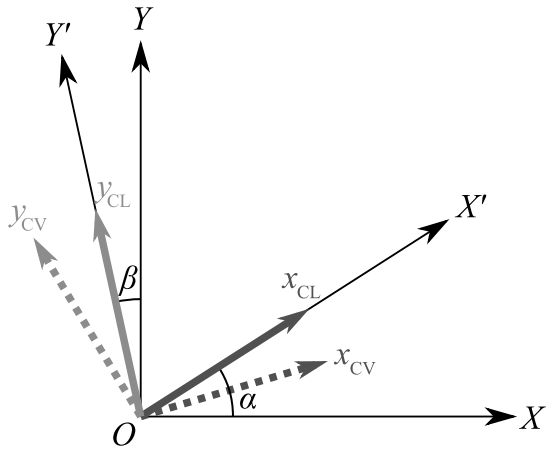


Figure 1. Example of non-orthogonal curvilinear axes. OX, OY: global Cartesian axes. OX',OY': local non-orthogonal curvilinear axes. $x_{CL}, y_{CL}, x_{CV}, y_{CV}$: unit vectors in the various curvilinear directions.

For general geometries, the river region does not line up well with the Cartesian axes. Discretizing using a Cartesian mesh would be inefficient; the grid size would have to be very fine in order to capture the spatial features properly. Fitting a non-orthogonal mesh to the river is a standard practice which provides better numerical performance. The coordinate changes are characterized by the deviation of the local axes from the Cartesian axes, called α and β, respectively (see figure 1) [8]. When working with water velocity in the curvilinear system, we must distinguish between the *curvilinear* (or *covariant*) velocity, whose components are

parallel with the local axes, and the *contravariant* velocity, whose components are perpendicular to their complementary axes (see figure 1). Covariant velocity, denoted by $u_{\mathrm{CL}}, v_{\mathrm{CL}}$ is used for the momentum-balancing equations (1) and (2), while contravariant velocity, denoted by $u_{\mathrm{CV}}, v_{\mathrm{CV}}$ is used for the mass-balancing equation (3).

The conversions between the three forms of velocity are easily computed:

$$\begin{bmatrix} u_{\mathrm{CL}} \\ v_{\mathrm{CL}} \end{bmatrix} = \sec(\alpha-\beta) \begin{bmatrix} \cos\beta & \sin\beta \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\beta \\ \sin\alpha & \cos\beta \end{bmatrix} \begin{bmatrix} u_{\mathrm{CL}} \\ v_{\mathrm{CL}} \end{bmatrix}$$
$$\begin{bmatrix} u_{\mathrm{CV}} \\ v_{\mathrm{CV}} \end{bmatrix} = \sec(\alpha-\beta) \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\alpha \\ \sin\beta & \cos\alpha \end{bmatrix} \begin{bmatrix} u_{\mathrm{CV}} \\ v_{\mathrm{CV}} \end{bmatrix}$$
$$\begin{bmatrix} u_{\mathrm{CV}} \\ v_{\mathrm{CV}} \end{bmatrix} = \begin{bmatrix} \sec(\alpha-\beta) & \tan(\alpha-\beta) \\ \tan(\alpha-\beta) & \sec(\alpha-\beta) \end{bmatrix} \begin{bmatrix} u_{\mathrm{CL}} \\ v_{\mathrm{CL}} \end{bmatrix}$$
$$\begin{bmatrix} u_{\mathrm{CL}} \\ v_{\mathrm{CL}} \end{bmatrix} = \begin{bmatrix} \sec(\alpha-\beta) & -\tan(\alpha-\beta) \\ -\tan(\alpha-\beta) & \sec(\alpha-\beta) \end{bmatrix} \begin{bmatrix} u_{\mathrm{CV}} \\ v_{\mathrm{CV}} \end{bmatrix}$$

All other variables have trivial transformations, and we will abuse notation by not distinguishing them from their original forms. For brevity, we will use γ as a shortcut for $(\alpha-\beta)$.

The linearized shallow water equations (6), (7) and (8) are transformed into the curvilinear coordinates [8]:

$$\begin{aligned} &\frac{\partial u_{\mathrm{CL}}}{\partial t} + U^0_{\mathrm{CL}}\frac{\partial u_{\mathrm{CL}}}{\partial x_{\mathrm{CL}}} + V^0_{\mathrm{CL}}\frac{\partial u_{\mathrm{CL}}}{\partial y_{\mathrm{CL}}} \\ &\quad + \left(U^0_{\mathrm{CL}}\frac{\partial u_{\mathrm{CL}}}{\partial y_{\mathrm{CL}}} + V^0_{\mathrm{CL}}\frac{\partial u_{\mathrm{CL}}}{\partial x_{\mathrm{CL}}}\right)\sin(\gamma) \\ &\quad = -g\frac{\partial h}{\partial x_{\mathrm{CL}}} - g\frac{\partial b}{\partial x_{\mathrm{CL}}} + Cu_{\mathrm{CL}} \end{aligned} \tag{10}$$

$$\begin{aligned} &\frac{\partial v_{\mathrm{CL}}}{\partial t} + U^0_{\mathrm{CL}}\frac{\partial v_{\mathrm{CL}}}{\partial x_{\mathrm{CL}}} + V^0_{\mathrm{CL}}\frac{\partial v_{\mathrm{CL}}}{\partial y_{\mathrm{CL}}} \\ &\quad + \left(U^0_{\mathrm{CL}}\frac{\partial v_{\mathrm{CL}}}{\partial y_{\mathrm{CL}}} + V^0_{\mathrm{CL}}\frac{\partial v_{\mathrm{CL}}}{\partial x_{\mathrm{CL}}}\right)\sin(\gamma) \\ &\quad = -g\frac{\partial h}{\partial y_{\mathrm{CL}}} - g\frac{\partial b}{\partial y_{\mathrm{CL}}} + Cv_{\mathrm{CL}} \end{aligned} \tag{11}$$

$$\begin{aligned} &\frac{\partial h}{\partial t} + U^0_{\mathrm{CV}}\frac{\partial h}{\partial x_{\mathrm{CL}}}\sec(\gamma) + V^0_{\mathrm{CV}}\frac{\partial h}{\partial y_{\mathrm{CL}}}\sec(\gamma) \\ &\quad + H^0\left(\frac{\partial u_{\mathrm{CV}}}{\partial x_{\mathrm{CL}}} + \frac{\partial v_{\mathrm{CV}}}{\partial y_{\mathrm{CL}}}\right)\sec(\gamma) = 0 \end{aligned} \tag{12}$$

These transformed equations are algebraically more involved, but from a practical perspective, simply add static trigonometric terms to the discretized scheme (16), (17), (18), to be derived next. They require that the velocity components be transformed back and forth between Cartesian and curvilinear axes. In particular, linearity is preserved.

C. Boundary conditions

For the boundary conditions, we imposed a condition that there be no velocity component perpendicular to the shoreline:

$$\vec{u}\cdot\vec{s}\big|_{\partial\Omega_{\mathrm{land}}} = 0 \tag{13}$$

where $\vec{s} = \vec{s}(x,y)$ is a vector perpendicular to the shoreline, and $\partial\Omega$ is the boundary of the domain. No-slip conditions ($\vec{u}\big|_{\partial\Omega_{\mathrm{land}}} = \vec{0}$) are also commonly used, but are inappropriate for a linear scheme, since shear forces arise from the non-linear terms in the original momentum equations (1), (2). We assume that the bathymetry is steep enough at the shore that the water height will not significantly affect the location of the land boundary.

This constraint is enforced on the curvilinear mesh by forcing the u_{CV} or v_{CV} component of the water velocity at specific nodes to zero.

The upstream velocity and downstream height boundary conditions are implicitly defined as being equal to the value at the initial condition:

$$\vec{u}(t)\big|_{\partial\Omega_{\mathrm{upstream}}} = \vec{u}(0)\big|_{\partial\Omega_{\mathrm{upstream}}} \tag{14}$$
$$h(t)\big|_{\partial\Omega_{\mathrm{downstream}}} = h(0)\big|_{\partial\Omega_{\mathrm{downstream}}} \tag{15}$$

This is an appropriate assumption for assimilation over short times compared to the tidal cycle.

D. Discretization

We use an implicit discretization scheme, consisting of backward Euler for the time derivative and centered differencing for the spatial derivatives. We use the covariant velocity variables. The mass conservation equation (12) uses contravariant velocity, not covariant velocity, which means an additional transform is necessary, as can be seen in (18).

$$\begin{aligned} \frac{u^{k+1}_{\mathrm{CL}i,j} - u^{k}_{\mathrm{CL}i,j}}{\Delta t} = &- U^0_{\mathrm{CL}i,j}\frac{u^{k+1}_{\mathrm{CL}i+1,j} - u^{k+1}_{\mathrm{CL}i-1,j}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &- V^0_{\mathrm{CL}i,j}\frac{u^{k+1}_{\mathrm{CL}i,j+1} - u^{k+1}_{\mathrm{CL}i,j-1}}{\Delta y_{i,j-1} + \Delta y_{i,j}} \\ &- \sin(\gamma_{i,j})U^0_{\mathrm{CL}i,j}\frac{u^{k+1}_{\mathrm{CL}i,j+1} - u^{k+1}_{\mathrm{CL}i,j-1}}{\Delta y_{i,j-1} + \Delta y_{i,j}} \\ &- \sin(\gamma_{i,j})V^0_{\mathrm{CL}i,j}\frac{u^{k+1}_{\mathrm{CL}i+1,j} - u^{k+1}_{\mathrm{CL}i-1,j}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &- g\frac{h^{k+1}_{i+1,j} - h^{k+1}_{i-1,j}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &- g\frac{b_{i+1,j} - b_{i-1,j}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &+ C_{i,j}u^{k+1}_{\mathrm{CL}i,j} \end{aligned} \tag{16}$$

$$\begin{aligned}\frac{v^{k+1}_{\mathrm{CL}i,j}-v^{k}_{\mathrm{CL}i,j}}{\Delta t} =& -U^0_{\mathrm{CL}i,j}\frac{v^{k+1}_{\mathrm{CL}i+1,j}-v^{k+1}_{\mathrm{CL}i-1,j}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &- V^0_{\mathrm{CL}i,j}\frac{v^{k+1}_{\mathrm{CL}i,j+1}-v^{k+1}_{\mathrm{CL}i,j-1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &- \sin(\gamma_{i,j})U^0_{\mathrm{CL}i,j}\frac{v^{k+1}_{\mathrm{CL}i,j+1}-v^{k+1}_{\mathrm{CL}i,j-1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &- \sin(\gamma_{i,j})V^0_{\mathrm{CL}i,j}\frac{v^{k+1}_{\mathrm{CL}i+1,j}-v^{k+1}_{\mathrm{CL}i-1,j}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &- g\frac{h^{k+1}_{i,j+1}-h^{k+1}_{i,j-1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &- g\frac{b_{i,j+1}-b_{i,j-1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &+ C_{i,j}v^{k+1}_{\mathrm{CL}i,j} \end{aligned} \tag{17}$$

$$\begin{aligned}\frac{h^{k+1}_{i,j}-h^{k}_{i,j}}{\Delta t} =& -\sec(\gamma_{i,j})\sec(\gamma_{i,j})U^0_{\mathrm{CL}i,j}\frac{h^{k+1}_{i+1,j}-h^{k+1}_{i-1,j}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &- \sec(\gamma_{i,j})\tan(\gamma_{i,j})V^0_{\mathrm{CL}i,j}\frac{h^{k+1}_{i+1,j}-h^{k+1}_{i-1,j}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &- \sec(\gamma_{i,j})\tan(\gamma_{i,j})U^0_{\mathrm{CL}i,j}\frac{h^{k+1}_{i,j+1}-h^{k+1}_{i,j-1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &- \sec(\gamma_{i,j})\sec(\gamma_{i,j})V^0_{\mathrm{CL}i,j}\frac{h^{k+1}_{i,j+1}-h^{k+1}_{i,j-1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &- \sec(\gamma_{i,j})\sec(\gamma_{i,j})H^0_{i,j}\frac{u^{k+1}_{\mathrm{CL}i+1,j}-u^{k+1}_{\mathrm{CL}i-1,j}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &- \sec(\gamma_{i,j})\tan(\gamma_{i,j})H^0_{i,j}\frac{u^{k+1}_{\mathrm{CL}i+1,j}-u^{k+1}_{\mathrm{CL}i-1,j}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &- \sec(\gamma_{i,j})\tan(\gamma_{i,j})H^0_{i,j}\frac{u^{k+1}_{\mathrm{CL}i,j+1}-u^{k+1}_{\mathrm{CL}i,j-1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &- \sec(\gamma_{i,j})\sec(\gamma_{i,j})H^0_{i,j}\frac{v^{k+1}_{\mathrm{CL}i,j+1}-v^{k+1}_{\mathrm{CL}i,j-1}}{\Delta y_{i,j-1}+\Delta y_{i,j}} \end{aligned} \tag{18}$$

where the subscript indexes i and j are for the x and y grid directions, respectively; the superscript index k is the time index; Δt is the time step; $\Delta x_{i,j}$ is the distance between node (i,j) and $(i+1,j)$; and $\Delta y_{i,j}$ is the distance between node (i,j) and $(i,j+1)$.

III. Quadratic Programming based variational data assimilation

Our method uses *variational data assimilation*. The variables in the discretized equations (16), (17), and (18) are concatenated into vectors, using the standardized framework set out in [11], as follows:

- X_n Concatenated vector of state variables (u,v,h) for all mesh points at time t_n.
- X_B Background term vector to guarantee well-posedness of the problem.
- Y_n Vector of observed variables at time t_n.
- B Covariance matrix of the background error (the vector difference between the initial state X_0 and the background term X_B).
- R_n Covariance matrix of the observation error at time t_n.
- H_n Observation operator, which projects the state vector X_n into the observation subspace containing Y_n.

Our variational data assimilation strategy is to search for the initial state X_0 that minimizes the ℓ^2 norm of the difference between the state and observation variables and the difference between the initial state and the background term X_B:

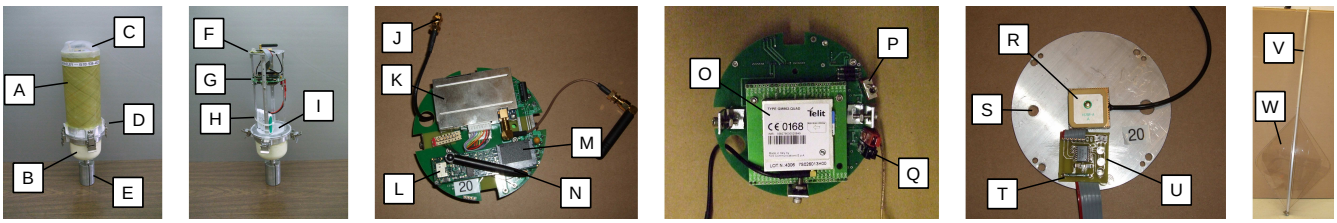
$$\begin{aligned}\mathcal{J}^0(X_0) &= (X_0-X_B)^T B^{-1}(X_0-X_B)\\ &+\sum_{n=0}^{n_{\max}}(Y_n-H_n[X_n])^T R_n^{-1}(Y_n-H_n[X_n])\end{aligned} \tag{19}$$

Without the background term, the problem is likely to be ill posed, as the number of points in the mesh far outnumbers the number of measurements. With the background term influencing the solution at the initial time, and the PDE constraints linking each point in the solution to its antecedents, the problem is well-posed. The background term could be derived from historical data, from forecasts, from a previous assimilation, or from forward simulation based on boundary conditions (either observed or artificially generated). The covariance matrices B and R_n affect the weight given to the background term and the observations. In the absence of second-order statistics, they can be approximations representing the assumed reliability of the different sources of information. As a simplifying assumption we take these matrices to be a scalar times the identity matrix: $b\mathbb{I}$ and $r\mathbb{I}$, respectively.

The observation operator H_n is often treated as a nonlinear operator; however, as described in Section IV, our observations come with both location and velocity information. Our assimilation method is *a posteriori*, so our knowledge of the observation positions can be used to represent the observation operator as a time-varying matrix. In the simplest case, where the assimilation time steps match the observation times, the H_n matrix would be a $\{0,1\}$ matrix, with element $(i,j)=1$ if the drifter associated with measurement i was in the cell associated with state variable j at time n. If drifter measurements are not synchronized with assimilation steps, then the values in the H_n matrix should reflect the polynomial approximation associated with the time discretization scheme. For example, for a single step method such as the backward Euler scheme, a drifter observation would be mapped into two H_n matrices using linear interpolation. This mapping can be generalized to any linear multi-step method.

The search space for the variational data assimilation is the initial condition of the solution to the linearized, discrete PDE; by the implicit definition of the boundary conditions, we are at the same time searching for the upstream velocity and downstream height boundary conditions. Appropriate choices for B and R_n mean that the cost function can be represented as a positive semidefinite quadratic term. The discretized dynamics of the flow are represented as a series of linear constraints of the form

$$EX_{n+1} = FX_n + g$$



(A) Fiberglass pipe (B) Cast fiberglass (C) Polycarbonate cap (D) Clasp (E) Stand (not part of drifter) (F) Antenna plate (G) Electronics (H) Battery (I) Bulkhead (J) GSM antenna connector (K) GPS module (L) Gumstix module (M) MMC card (N) Bluetooth antenna (O) GSM module (P) Power switch (Q) Battery connector (R) GPS antenna (S) Hole for GSM antenna (not shown) (T) Magnetic switch (U) Status LEDs (V) Aluminum tube (W) Polycarbonate plate

Figure 2. Drifter hardware.

where E, F are matrices determined by the time and space difference schemes (16), (17) and (18), and g is a vector capturing source terms that doe not depend on the state, such as the bottom elevation. There is no requirement that E be invertible, which means that implicit schemes can be implemented in this formulation. This broadens the applicability of the method significantly; implicit methods are not constrained by the *Courant-Friedrichs-Lewy* (CFL) stability condition on the time step. Sequential assimilation methods, such as the Kalman filter and variations, are restricted to explicit schemes by the nature of their update process. The time step used in these methods is restricted by the CFL condition and can often be inconveniently short. In addition, it is difficult to assimilate initial conditions (which are part of the unknown search space) using sequential methods.

With a positive semi-definite quadratic cost function and linear constraints, the data assimilation problem can be posed as a QP problem

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{G}\mathbf{x} \leq \mathbf{h} \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned}$$

The variables in bold are from standard optimization formulations [2] and should not be confused with the variables used in the rest of this article. In particular, note that $\mathbf{x}$ is the vertical concatenation of all state vectors $X_0 \ldots X_{n_{\max}}$, $\mathbf{P}$ and $\mathbf{q}$ are found by expanding all the terms in (19) and combining into a single quadratic expression. The equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ represents the flow dynamic constraints described above. $\mathbf{G}$ and $\mathbf{h}$ are normally zero, although we may impose heuristic inequality constraints to reduce the search space, in particular for initial and boundary conditions.

IV. Hardware platform

We now present the floating sensor network hardware platform that was developed to gather Lagrangian flow data in shallow water environments and used to gather the data presented later in this article. Interior and exterior photos of the drifter device are shown in Figure 2.

The drifter fleet, consisting of ten units, was designed and manufactured in the Lagrangian Sensor Systems Laboratory at UC Berkeley. Design goals included low cost, ease of manufacture and service with in-house techniques, 48 hour mission autonomy, stable hydrodynamic configuration, rotationally symmetric profile, and an internal volume sufficient for electronics and future water sensors.

Drifter housing. The housing of the drifter is based around a 11 cm ID fiberglass pipe. The top cap is vacuum-formed polycarbonate. The lower shell is hand-cast fiberglass. The top hull and bottom hull are joined with epoxy to machined aluminum flanges, which seal against the main bulkhead with O-Rings and spring-loaded clamps. The bottom hull is watertight in generation one, but will be modified into a flooded bay for water-facing sensors in generation two.

Drifter drogue. A 1.3 m aluminum tube is attached to a lug in the lower hull with a cotter pin. At the opposite end of the tube, two polycarbonate plates, 40 cm square, are mounted diagonally. This puts a large drag component 1.0 m below the drifter hull, which makes the drifter be driven primarily by the current below the surface as opposed to the wind-mixed layer that may be present at the surface.

Electronics. The main challenge of the electronics design was the selection and integration of the various modules. Cost, power consumption, voltage compatibility, communication protocols, and mechanical footprint were the main selection criteria. Harness wiring was kept to a minimum by integrating the three major modules (CPU, GPS, and GSM) onto a single printed circuit board, which also provided mechanical support.

- Power: The drifter carries a 10.4 amp-hour, 3.7 volt lithium ion battery.
- Gumstix: The main computational unit is a Basix 400-BT from Gumstix Inc. This embedded module contains a 400 MHz Marvell XScale PXA255 processor capable of running an embedded Linux distribution [12]. It has a 1 GB MMC card.
- GPS: The GPS receiver is a Magellan AC-12 OEM module. It has a CEP of 1.5 m, and can record pseudo-range and carrier phase data for post-processing [25].
- Cell Phone: A Telit GM-862 GSM module is used for communication. TCP connections can be made with home base servers via AT&T's GPRS service [24].

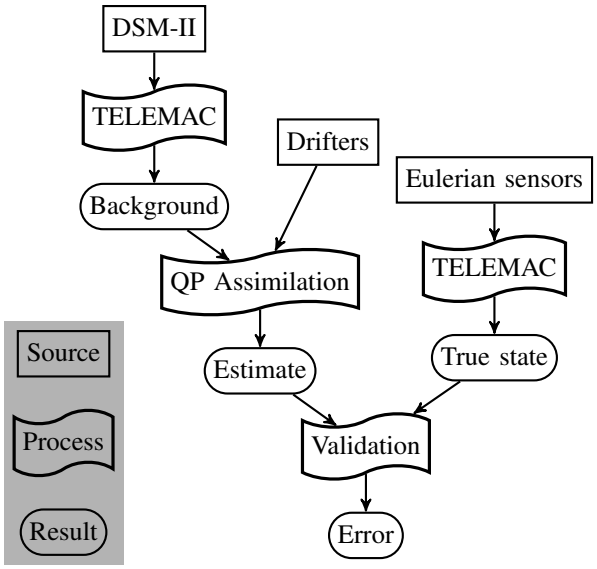


Figure 3. Data flow diagram used for the data assimilation using the hardware platform.

V. Field test

A. Available data

The proposed architecture and platform is designed to solve practical problems for which several types of information are available. The following is a list of the data sources used by the data assimilation method:

- *Drifters*: The Lagrangian sensors record their position with GPS as they advect through the water. They also record a GPS velocity signal, which we use directly (as opposed to deriving velocity from the successive positions using a finite difference scheme). We built ten drifters; in the experiments presented here, up to eight were deployed at a time.
- *DSM-II Historical Data*: DSM-II [1] is a one dimensional model of the entire Sacramento/San Joaquin Delta. It was used to generate historical flow and height values for the background.

For validation purposes, we also gathered Eulerian data at the boundaries of the region of interest, using sensors described below. This data was used as the boundary conditions for a forward simulation using TELEMAC, a commercial hydrodynamics simulator. TELEMAC is essentially a specialized PDE solver for the shallow water equations; given the initial conditions and boundary conditions, it finds the velocity at all points in the mesh through a forward simulation of the equation. Since actual measurement of the initial conditions was unavailable, we used the standard technique of starting with an arbitrary initial condition, holding the boundary conditions steady, and running the simulation for a long time, essentially "washing away" the arbitrary initial condition. This technique is only appropriate for systems that are close to a steady state, which is a reasonable assumption for the slowly-changing river.

The Eulerian data includes the following items (see Figure 4):

- *Acoustic Doppler Current Profiler* (ADCP): This Eulerian sensor was installed by our group near the upstream boundary of the region of interest. It sits on the bottom of the river and measures the water velocity in the vertical column over it. This data allows estimation of the upstream flow boundary condition.
- *USGS Gauge Stations*: These Eulerian sensors measure flow and height. One sensor in the Sacramento River and one in the Georgiana Slough provide information about the downstream boundaries.

The list of data sources must also include the bathymetry and Manning parameters. The bathymetry is used in the QP assimilation (see equations (6), (7) and (8)). The TELEMAC forward simulations that generate the background term and validation data use both the bathymetry and the Manning parameters.

The data flow diagram in Figure 3 shows how the various data are used. Historical DSM-II data is used, with TELEMAC 2D [6] forward simulations, to generate the background term for the QP process. The estimate of the state of the system is generated by assimilating the drifter data. The Eulerian sensors are used with TELEMAC to generate a separate state estimate that is used for Eulerian validation.

B. Experimental strategy

Eight drifter deployments were performed from November 12 to November 16, 2007, at the junction of the Georgiana Slough and Sacramento River in California. This location was chosen for the USGS field gauges which could be used for Eulerian validation.

For each experiment, between seven and ten drifters were placed in the water by personnel in a small motorcraft. The initial positions were in a roughly straight line across the river, with approximately even spacing, but in the center of the river to avoid obstacles and shallow areas on the sides. Figure 5 depicts an example of the drop points used in experiment 4 on November 16. The drifters were monitored as they travelled in the river. Each experiment was planned to last between 45 and 60 minutes; in practice, some of the experiments were terminated earlier. Reasons for terminating the experiment included (*i*) drifters travelling past the junction, eliminating line of sight, (*ii*) drifters spacing out too far, making them difficult to monitor, (*iii*) miscellaneous logistical concerns.

With the development of short-range and long-range wireless communication capabilities on the drifters, the hardware infrastructure is designed to let the drifters operate autonomously, without direct line of sight supervision, allowing for experiments with expanded domains in space and time.

Figure 6 shows the water velocity at the ADCP versus time over the five day experimental period. The start times of the eight experiments are shown with "x" marks. The velocity time series was processed with a low-pass filter (zero-phase, cutoff frequency 7.85×10^{-5}Hz, corresponding to a period of 3.54 hour, generated by the Parks-McClellan optimum filter algorithm [4]). To better show the length of the experiments, and their relationship to the tidal cycle, the

filtered velocity signal for all five days was superimposed in figure 7, referenced to the minor maximum of the velocity.

Post-experiment analysis showed that several drifters did not record GPS data all the time; in most cases this was traced to antenna connection problems. This reduced the number of operating drifters at a given time to between five and eight. Only four of the eight experiments had enough data to proceed with the assimilation method.

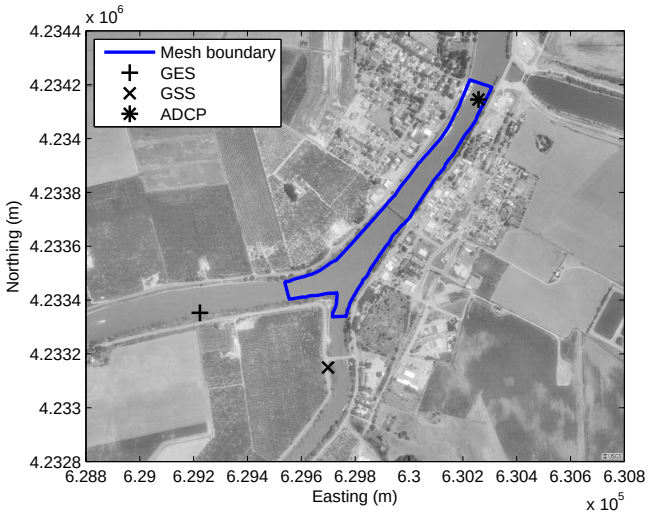


Figure 4. Sacramento River/Georgiana Slough with modelled area, ground stations, and sensor deployment locations. Image courtesy of USGS.

C. Implementation of the algorithm

The drifter measurements were sampled at 30s. Each drifter measurement was assigned according to its GPS location to a specific cell of the curvilinear mesh, and the GPS velocity was converted to curvilinear coordinates. The DSM-II historical data was then used to generate boundary conditions for a TELEMAC forward simulation to generate the background term. A QP problem was formulated using the drifter measurements and the background term for the

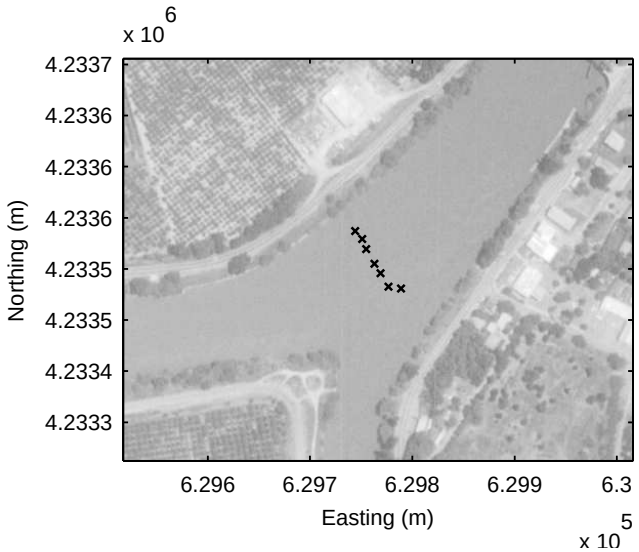


Figure 5. Example of drop points for drifter release in the final experiment.

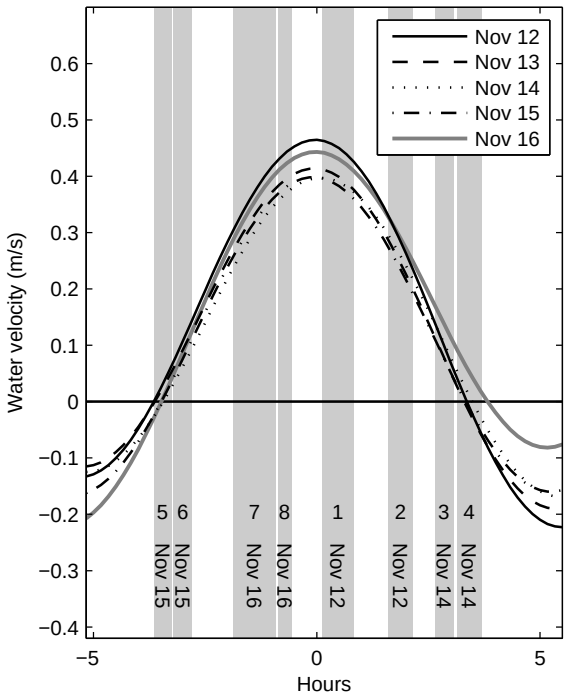


Figure 7. Experiment times shown relative to minor velocity peak.

cost function, and the curvilinear, discretized, linearized PDE equations as linear constraints, as described in Section III. The drifters do not gather information about the water height. The friction source term was set to zero. The QP problem was expressed using the optimization modeling language AMPL and solved with CPLEX. The optimal initial condition was extracted from the CPLEX solution, and the curvilinear velocity field was converted back to the Cartesian grid.

One feature of the QP formulation is that the number of sensors can vary with time, simply by adding or removing the necessary terms from the cost function (19). This is advantageous, because in practice there are often gaps in the GPS tracks of the drifters (as they pass underneath bridges, or experience similar signal loss). Instead of trying to patch the holes in the record with some sort of interpolation, the data can be passed as-is to the QP assimilation process.

D. Validation

A forward simulation of the region of interest was performed using the data from Eulerian sensors. This data was used as the boundary conditions for a SWE simulation, to generate what we will call the "true state" velocity field. This forward simulation *does* include the river bed friction term. The relative error between the true state, (u_T, v_T), and the estimated initial condition velocity field from the QP process, (u, v), was computed by dividing the ℓ^2 norm of the difference by the magnitude of the simulated field:

$$\epsilon_E(k) = \sqrt{\left(\sum_j \begin{matrix}(u_{Tj}-u_j)^2 \\ +(v_{Tj}-v_j)^2\end{matrix}\right) \Big/ \left(\sum_j \left(u_{Tj}^2 + v_{Tj}^2\right)\right)} \tag{20}$$

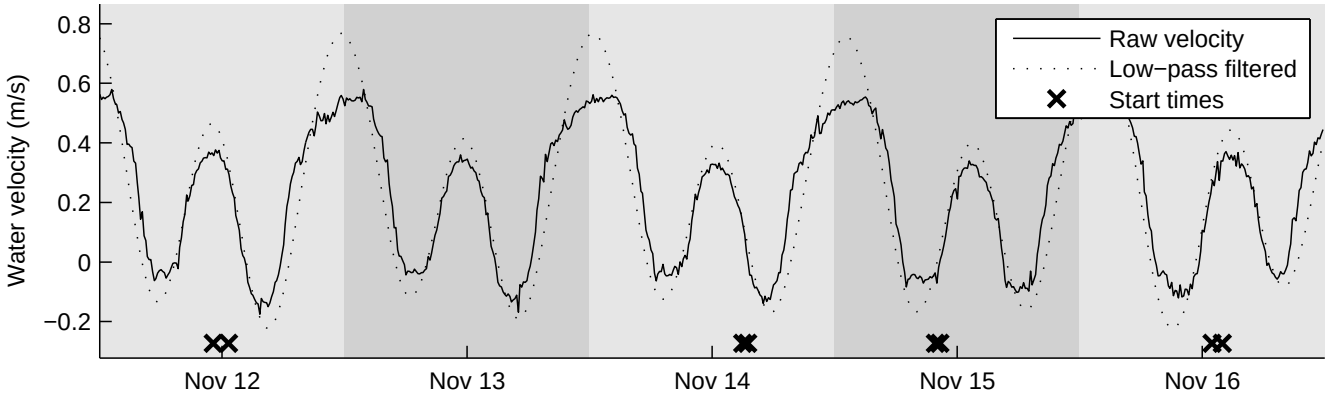


Figure 6. Overview of experimental timeframe.

where j is the node index.

E. Results

Figure 8 shows the initial flow field condition assimilated by the QP algorithm for one of the experiments. The velocity information is displayed both in vector and magnitude plots. The height variable is very smooth (differing by only a few centimeters over the region) and not interesting to plot. Only one experiment is shown for space constraints. Figure 9 shows how the QP assimilated velocity field is closer to the true state (generated by forward simulation from Eulerian sensors) than the background term.

Using a 2000-point mesh (16 cells across the river, 118 cells down the reach of the Sacramento segment), each QP assimilation takes approximately 10 minutes on a single 2.0GHz processor.

VI. Conclusion

In this article, we present a method for formulating the variational data assimilation problem for Lagrangian sensors in shallow water flows as a quadratic programming optimization problem with linear constraints. A major advantage of the quadratic programming formulation is that the constraints can express the model partial differential equation discretized with an implicit scheme. This differentiates our method from sequential methods such as the Kalman filter, and allows our method to use longer time steps than explicit methods. Our method also assimilates on the initial conditions, in contrast to many sequential methods.

The quadratic programming assimilation method relies on a background term, as many variational data assimilation methods do, both to guarantee well-posedness and to provide a "first guess" to the system. The metric used to evaluate the assimilation performance is the improvement made in relative error versus a true state. Care was taken to ensure that the true state used distinct information; the assimilation process relied on historical data (for the background term) and Lagrangian sensor data, while the true state was simulated from local Eulerian sensors. (Both sides use the same bathymetry and Manning parameter data, but this is not a major issue).

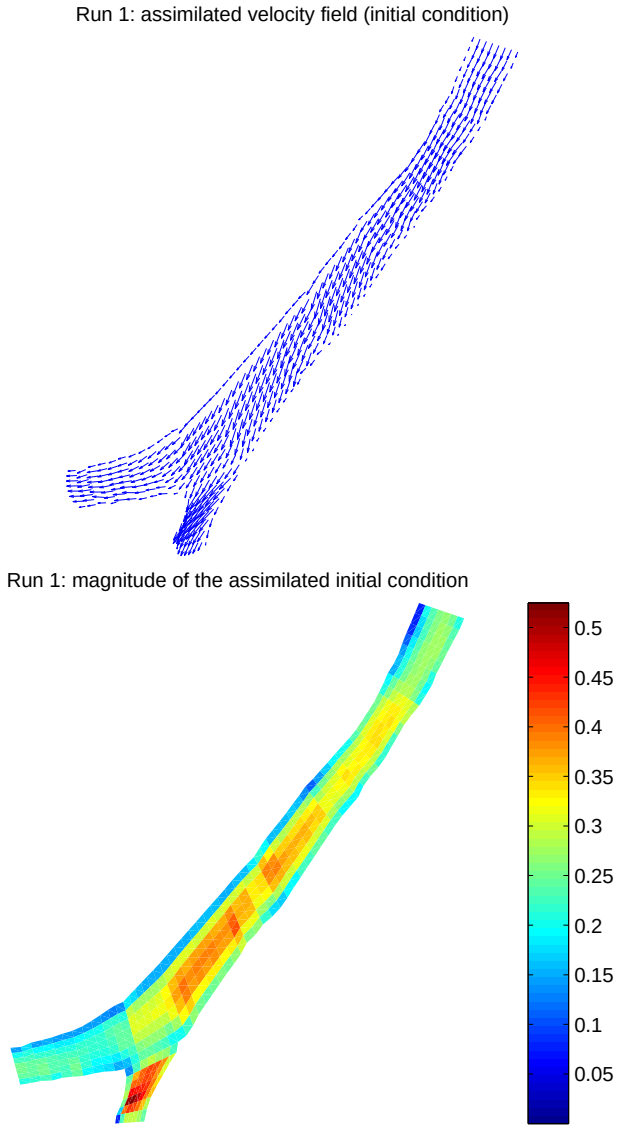


Figure 8. Assimilated velocity field initial condition for experiment 1 (vectors and magnitude). Color scale is in m/s.

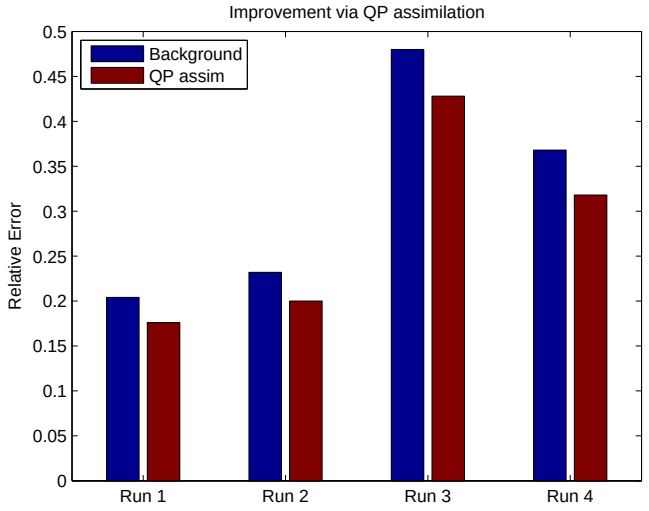


Figure 9. Change in relative error for each of four experiments.

We also present a new sensor network platform for gathering Lagrangian flow information. The drifters described herein provide an inexpensive flow measurement capability. With the appropriate assimilation techniques to process their data, they open up new possibilities for modeling and understanding shallow water systems in regions where Eulerian sensing is too expensive or otherwise unavailable.

Our new hardware platform was demonstrated and validated in a set of experiments that gathered flow information in a river junction environment. The assimilation procedure demonstrated an improved relative error to the assumed ground truth.

Further work will focus on demonstrating larger relative error improvements using refinements of the technique. The partial differential equations used in our model are appropriate for unsteady as well as steady flows; the flows we study in this experiment are technically unsteady (due to the tidal influence), but the rate of change is very slow. Heuristically adding constraints to restrict the quadratic programming solution to "almost steady" flows would reduce the search space, which could allow for greater weight on the measurements. Ultimately, we hope to demonstrate a method that can produce useful assimilations even when the background term is severely different from the true state.

VII. Acknowledgements

The authors would like to thank Prof. Mark Stacey for providing the deployable Eulerian sensors used for validation, and Maureen Downing-Kunz and Julie Percelay for assisting with their deployment. Julie Percelay also provided valuable assistance with the TELEMAC forward simulations used herein. The hardware development project relies on the hard work and ingenuity of many undergraduates and interns, including Andrew Spencer, Jason Wexler, Jonathan Ellithorpe, Jean-Severin Deckers, Nahi Ojeil, Tarek Ibrahim, and Anwar Ghoche.

References

[1] J. Anderson and M. Mierzwa. DSM2 tutorial, an introduction to the Delta Simulation Model II (DSM2). Technical report, State of California, Department of Water Resources, 2002.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[3] A. Chadwick, J. Morfett, and M. Borthwick. *Hydraulics in Civil and Environmental Engineering*. Spon Press, London, 2004.

[4] D. S. P. Committee, editor. *Programs for digital signal processing*. IEEE Press, New York, 1979.

[5] F. X. L. Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meterological observations: theoretical aspects. *Tellus*, 38A:97–110, 1986.

[6] EDF. Telemac 2D. version 5.2. Technical report, EDF, 2003.

[7] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag, New York, 2007.

[8] K. George. A depth-averaged tidal numerical model using non-orthogonal curvilinear co-ordinates. *Ocean Dynamics*, 57(4–5):363–374, 2007.

[9] J. R. Gunson and P. Malanotte-Rizzoli. Assimilation studies of open-ocean flows. *J. Geophys. Res.*, 101:28457–28472, 1996.

[10] M. Honnorat, J. Monnier, and F.-X. L. Dimet. Lagrangian data assimilation for river hydraulics simulations. In *European Conference on Computational Fluid Dynamics*. ECCOMAS CFD, Egmond aan Zee, The Netherlands, 2006.

[11] K. Ide, P. Courtier, M. Ghil, and A. Lorenc. Unified notation for data assimilation: Operational, sequential and variational. *J. Met. Soc. of Japan*, 75(1B):71–79, 1997.

[12] Intel. *Intel PXA255 Processor Design Guide*, 2003.

[13] R. B. Jackson, S. R. Carpenter, C. N. Dahm, D. M. McKnight, R. J. Naiman, S. L. Postel, and S. W. Running. Water in a changing world. *Ecological Applications*, 11(4):1027–1045, 2001.

[14] A. Molcard, L. Piterbarg, A. Griffa, T. Özgökmen, and A. Mariano. Assimilation of drifter observations for the reconstruction of the eulerian circulation field. *J. Geophys. Res.*, 108:3056, 2003.

[15] I. M. Navon. Practical and theoretical aspects of adjoint parameter estimation and identification in meteorology and oceanography. *Dyn. Atmos. Oceans*, 27:55–79, 1997.

[16] M. Nodet. Variational assimilation of lagrangian data in oceanography. *Inverse Problems*, 22:245–263, 2006.

[17] T. Oki and S. Kanae. Global hydrological cycels and world water resources. *Science*, 313:1068–1072, 2006.

[18] C. Paniconi, M. Marrocu, M. Putti, and M. Verbunt. Newtonian nudging for a richards equation-based distributed hydrological model. *Adv. Water Resources*, 26:161–178, 2003.

[19] S. L. Postel. Entering an era of water scarcity: The challenges ahead. *Ecological Applications*, 10(4):941–948, 2000.

[20] J. Rodda, S. Pieyns, N. Sehmi, and G. Matthews. Towards a world hydrological cycle observing system. *Hydrological sciences journal*, 38:373, 1993.

[21] L. C. Smith. Satellite remote sensing of river inundation area, stage, and discharge: a review. *Hydrological Processes*, 11(10):1427–1439, 1997.

[22] I. S. Strub, J. Percelay, M. T. Stacey, and A. M. Bayen. Inverse estimation of open boundary conditions in tidal channels. *Ocean Modelling*, 29(1):85–93, 2009.

[23] I. S. Strub, J. Percelay, O.-P. Tossavainen, and A. M. Bayen. Comparison of two data assimilation algorithms for shallow water flows. *Networks and Heterogenous Media*, 4(2):409–430, 2009.

[24] Telit Communications. *GM862-QUAD/GM862-QUAD-PY Hardware User Guide*, 2006.

[25] Thales Navigation. *A12, B12, & AC12 Reference Manual*, 2005.

[26] A. Tinka, I. Strub, Q. Wu, and A. M. Bayen. Quadratic programming based data assimilation with passive drifting sensors for shallow water flows. In *Proceedings of the IEEE Conference on Decision and Control*. 2009. Accepted.

[27] C. Vreugdenhil. *Numerical Methods for Shallow Water Flow*. Kluwer Academic Publishers, 1994.

[28] Q. Wu, M. Rafiee, A. Tinka, and A. M. Bayen. Inverse modeling for open boundary conditions in channel network. In *Proceedings of the 47th IEEE Conference on Decision and Control*. 2009. Accepted.