

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. 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. A new sensor network hardware platform for gathering flow information is presented. We summarize the results of a field experiment designed to demonstrate the capabilities of our assimilation method with data gathered from the sensors. Validation of the results is performed by comparing them to an estimate derived from an independent set of static sensors.

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 [18]. Improving water use efficiency can help balance supply and demand [12] and relieve scarcity; this will require improved methods for modeling and monitoring the flow of freshwater through the hydrological cycle [20][16]. River hydraulics can be modeled with shallow water equations in one or two dimensions [3]. Data assimilation, the process of integrating measurements into a flow model, originated in meteorology and oceanography [4]. Techniques for data assimilation include variational methods [14], Kalman filtering and its extensions [6], optimal statistical interpolation [13], and Newtonian relaxation [17].

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). The trends of electronics miniaturization and availability of wireless communications have increased the interest in novel Lagrangian sensor systems. 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 implementations of Lagrangian data

assimilation are in oceanography or meteorology (see, for example, [8],[15],[14]); for hydraulic systems, Lagrangian data assimilation of shallow water flows to estimate the bottom topology was attempted in [9].

Our objective 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 PDE model of the river, for which, in general, we do not have knowledge of the *initial conditions* (ICs) or *boundary conditions* (BCs) for the model.

The contributions made in this article are described below:

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

The linearization and variational assimilation formulation presented herein were developed for shallow water flows in [21].

II. HYDRODYNAMIC MODEL

Our model of the hydrodynamics uses the SWE. We will present the equations, followed by linearization and discretization. For legibility we suppress the arguments for dependent variables. The governing hydrodynamic equations for the modeled system are [19], [24]:

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

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

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

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/s²; ν_t is the coefficient of turbulence diffusion, obeying the so called k-epsilon model [19]; and $F_x = F_x(x, y, t), F_y = F_y(x, y, t)$ are friction terms

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

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

where $\alpha = \alpha(x, y)$ 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.

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

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

as the linearized friction coefficient. For the boundary conditions, we imposed a condition that there be no velocity component perpendicular to the shoreline:

$$\vec{u} \cdot \vec{s} \Big|_{\partial\Omega_{\text{land}}} = 0 \quad (10)$$

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 ($\vec{u} \Big|_{\partial\Omega_{\text{land}}} = \vec{0}$) are also commonly used. 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.

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

$$\vec{u}(t) \Big|_{\partial\Omega_{\text{upstream}}} = \vec{u}(0) \quad (11)$$

$$h(t) \Big|_{\partial\Omega_{\text{downstream}}} = h(0) \quad (12)$$

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

The discretization scheme is implicit, using backward Euler for the time derivative and centered differences for the spatial derivatives.

$$\begin{aligned} \frac{u_{i,j}^{k+1} - u_{i,j}^k}{\Delta t} &= -U_{i,j}^0 \frac{u_{i+1,j}^{k+1} - u_{i-1,j}^{k+1}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &\quad - V_{i,j}^0 \frac{u_{i,j+1}^{k+1} - u_{i,j-1}^{k+1}}{\Delta y_{i,j-1} + \Delta y_{i,j}} \\ &\quad - g \frac{h_{i+1,j}^{k+1} - h_{i-1,j}^{k+1}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &\quad - g \frac{b_{i+1,j} - b_{i-1,j}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &\quad + C_{i,j} u_{i,j}^{k+1} \end{aligned} \quad (13)$$

$$\begin{aligned} \frac{v_{i,j}^{k+1} - v_{i,j}^k}{\Delta t} &= -U_{i,j}^0 \frac{v_{i+1,j}^{k+1} - v_{i-1,j}^{k+1}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &\quad - V_{i,j}^0 \frac{v_{i,j+1}^{k+1} - v_{i,j-1}^{k+1}}{\Delta y_{i,j-1} + \Delta y_{i,j}} \\ &\quad - g \frac{h_{i,j+1}^{k+1} - h_{i,j-1}^{k+1}}{\Delta x_{i,j-1} + \Delta x_{i,j}} \\ &\quad - g \frac{b_{i,j+1} - b_{i,j-1}}{\Delta x_{i,j-1} + \Delta x_{i,j}} \\ &\quad + C_{i,j} v_{i,j}^{k+1} \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{h_{i,j}^{k+1} - h_{i,j}^k}{\Delta t} &= -U_{i,j}^0 \frac{h_{i+1,j}^{k+1} - h_{i-1,j}^{k+1}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &\quad - V_{i,j}^0 \frac{h_{i,j+1}^{k+1} - h_{i,j-1}^{k+1}}{\Delta y_{i,j-1} + \Delta y_{i,j}} \\ &\quad - H_{i,j}^0 \frac{u_{i+1,j}^{k+1} - u_{i-1,j}^{k+1}}{\Delta x_{i-1,j} + \Delta x_{i,j}} \\ &\quad - H_{i,j}^0 \frac{v_{i,j+1}^{k+1} - v_{i,j-1}^{k+1}}{\Delta y_{i,j-1} + \Delta y_{i,j}} \end{aligned} \quad (15)$$

where the subscript indexes are for the x and y grid directions, respectively; the superscript index 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 $(i, j+1)$.

The spatial discretization is performed on a non-orthogonal curvilinear mesh as in [7]. From a practical perspective, this adds a few static trigonometric terms to the discretized scheme (13), (14), (15), and requires that the velocity components be transformed back and forth between Cartesian and curvilinear axes.

III. QUADRATIC PROGRAMMING ASSIMILATION

Our method uses the framework of *variational data assimilation* in discrete space and time. In the standardized notation set out in [10]:

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 :

$$\mathcal{J}^0(X_0) = (X_0 - X_B)^T B^{-1} (X_0 - X_B) + \sum_{n=0}^{n_{\max}} (Y_n - H_n[X_n])^T R_n^{-1} (Y_n - H_n[X_n]) \quad (16)$$

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. Introducing the background term both guarantees the existence of a unique minimum and provides a “first guess” to the assimilation problem. The background term could be derived from historical data, from forecasts, a previous assimilation, or from forward simulation based on boundary conditions (either observed or invented). 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 will 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 non-linear 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.

The search space for the variational data assimilation is the set of possible initial conditions 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

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

where E , F are matrices determined by the time and space difference schemes, and g is a vector capturing input terms like the friction coefficient. 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, like the Kalman Filter and variations, are restricted to explicit schemes by the nature of their update processes; the time step used in these methods is restricted by the CFL condition and can often be inconveniently short.

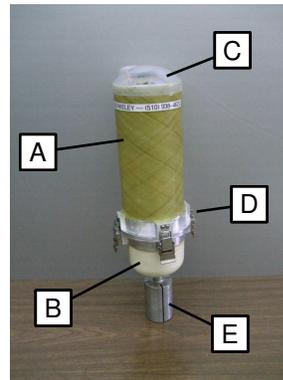
With a positive semi-definite quadratic cost function and linear constraints, the data assimilation problem can be posed as a QP problem

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

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 $X_0 \dots X_{n_{\max}}$, $\mathbf{A} \mathbf{x} = \mathbf{b}$ 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.

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.



- (A) Fiberglass pipe
- (B) Cast fiberglass
- (C) Polycarbonate cap
- (D) Clasp
- (E) Stand (not part of drifter)

Figure 1. Drifter hardware.

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 1, but will be modified into a flooded bay for water-facing sensors in generation 2.

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

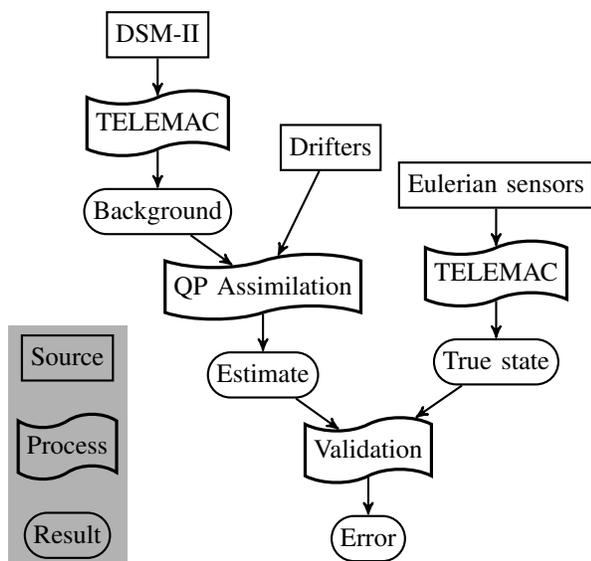


Figure 2. Data flow diagram used for the data assimilation using the hardware platform.

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 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 [11]. 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 [23].
- **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 [22].

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). The altitude signal from the GPS module is not accurate enough

to estimate the water height; we therefore use 2D position information only. 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 1D 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, which can be used to directly simulate the system. This additional data includes the following items (see Figure 3):

- **Acoustic Doppler Current Profiler (ADCP):** This Eulerian sensor was installed 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 on the Sacramento River and one on 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 equation (15)). 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 2 shows how the various data are used. Historical DSM-II data is used, with TELEMAC 2D [5] 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

Four 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. The flow in this region is tidally forced; our four experiments cover different tidal phases.

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 4 depicts an example of the drop points used in experiment 4 on November 16. The drifters were monitored as they travelled in the river. The GSM communication system was not fully implemented in time for the field test, so the drifters had to be monitored directly. 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.

Post-experiment analysis showed that several drifters did not record GPS data; in most cases this was traced to antenna connection problems. This reduced the number of operating drifters to between five and eight.

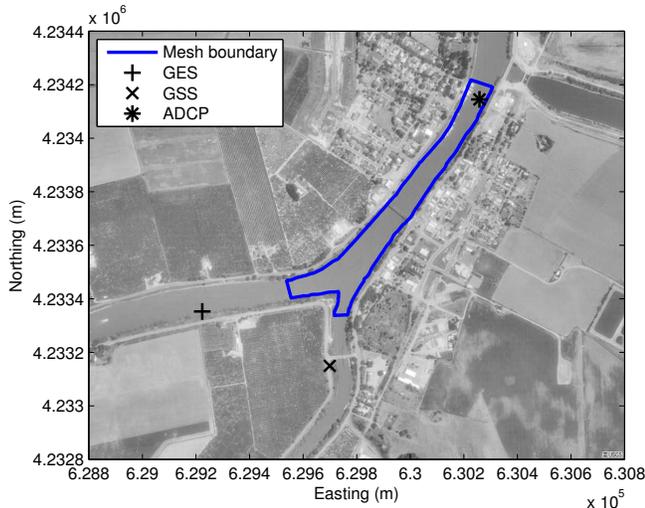


Figure 3. Sacramento River/Georgiana Slough with modelled area, ground stations. Image courtesy of USGS.

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

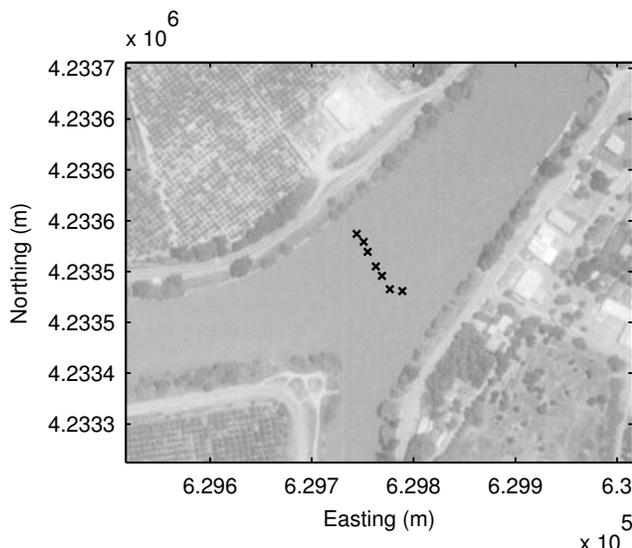


Figure 4. Example of drop points for drifter release in the final experiment.

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 cost function, and the curvilinear, discretized, linearized PDE equations as linear constraints, as described in Section III. The R_n matrix, assumed to be of the form $r\mathbb{I}$, was determined from the GPS module documentation (the accuracy of the 2D position data); the B matrix, in the form $b\mathbb{I}$, was determined by estimating the accuracy of the background state. 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 (16). 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. Since these sensors do not measure height, the observation operator H_n in (16) does not map the estimated height into the cost function.

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. 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:

$$\epsilon_E(k) = \sqrt{\frac{\left(\sum_j (u_{Tj} - u_j)^2 + (v_{Tj} - v_j)^2\right)}{\left(\sum_j u_{Tj}^2 + v_{Tj}^2\right)}} \quad (17)$$

where j is the node index.

E. Results

Figure 5 shows the initial condition assimilated by the QP algorithm for one of the experiments. Only one experiment is shown for space constraints. Figure 6 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

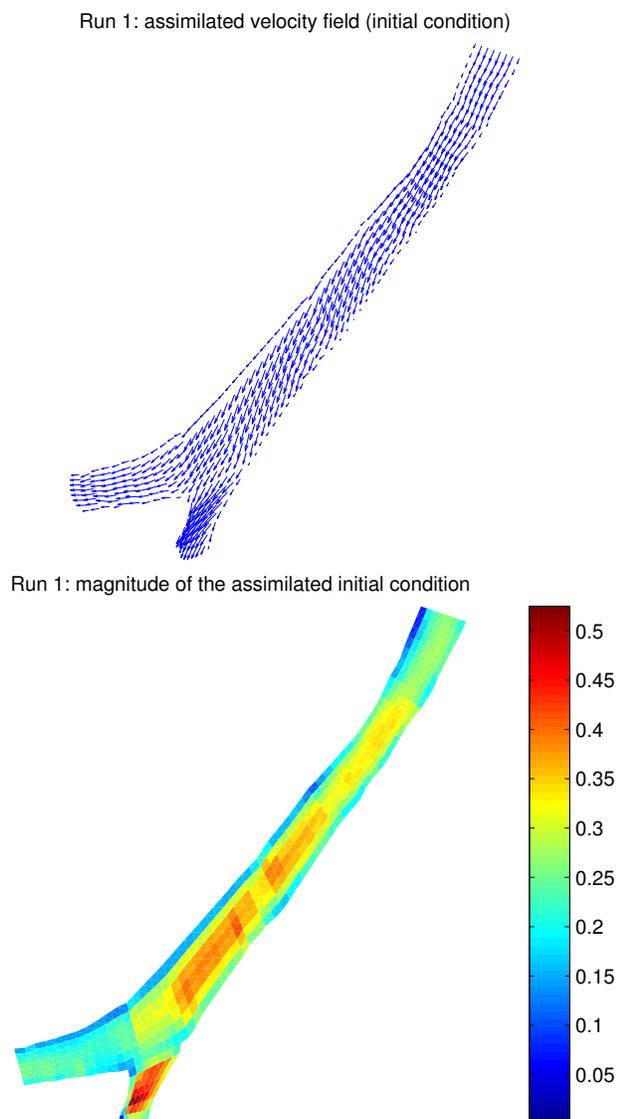


Figure 5. Assimilated velocity field initial condition for experiment 1 (vectors and magnitude). Color scale is in m/s .

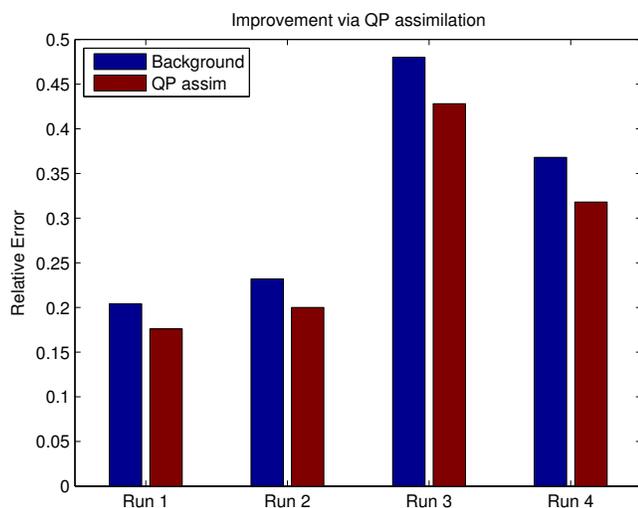


Figure 6. Change in relative error for each of four experiments.

in shallow water flows as a QP optimization problem with linear constraints. A major advantage of the QP formulation is that the constraints can express the model PDE 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.

Like many variational data assimilation methods, the QP assimilation method relies on a background term, 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).

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 PDEs 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 QP solution to “almost steady” flows would reduce the search space, which could allow for greater weight on the measurements, and could also allow reintroduction of the h equation (15). 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, 2004.

- [3] A. Chadwick, J. Morfett, and M. Borthwick. *Hydraulics in Civil and Environmental Engineering*. Spon Press, 2004.
- [4] F. X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus*, 38A:97–110, 1986.
- [5] Telemac 2d. version 5.2. Technical report, EDF, 2003.
- [6] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag, 2007.
- [7] K. George. A depth-averaged tidal numerical model using non-orthogonal curvilinear co-ordinates. *Ocean Dynamics*, 57(4–5):363–374, October 2007.
- [8] J. R. Gunson and P. Malanotte-Rizzoli. Assimilation studies of open-ocean flows. *J. Geophys. Res.*, 101:28457–28472, 1996.
- [9] M. Honnorat, J. Monnier, and F.-X. Le Dimet. Lagrangian data assimilation for river hydraulics simulations. In *European Conference on Computational Fluid Dynamics*, Netherlands, 2006. ECCOMAS CFD.
- [10] 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.
- [11] Intel. *Intel PXA255 Processor Design Guide*, 2003.
- [12] R. Jackson, S. Carpenter, C. Dahm, D. McKnight, R. Naiman, S. Postel, and S. Running. Water in a changing world. *Ecological Applications*, 11(4):1027–1045, 2001.
- [13] A. Molcard, L.I. 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.
- [14] I. M. Navon. Practical and theoretical aspects of adjoint parameter estimation and identification in meteorology and oceanography. *Dyn. Atmos. Oceans*, 27:55–79, 1997.
- [15] M. Nodet. Variational assimilation of lagrangian data in oceanography. *Inverse Problems*, 22:245–263, 2006.
- [16] T. Oki and S. Kanae. Global hydrological cycles and world water resources. *Science*, 313:1068–1072, 2006.
- [17] 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.
- [18] S. Postel. Entering an era of water scarcity: The challenges ahead. *Ecological Applications*, 10(4):941–948, 2000.
- [19] Report EDF. *TELEMAC 2D. Version 5.2 – Principle note*, 2002.
- [20] J. Rodda, S. Pieyns, N. Sehmi, and G. Matthews. Towards a world hydrological cycle observing system. *Hydrological sciences journal*, 38:373, 1993.
- [21] I. Strub, J. Percelay, O.-P. Tossavainen, and A. Bayen. Comparison of two data assimilation algorithms for shallow water flows. *Networks and heterogenous media*, 4(2):409–430, June 2009.
- [22] Telit Communications. *GM862-QUAD/GM862-QUAD-PY Hardware User Guide*, 2006.
- [23] Thales Navigation. *A12, B12, & AC12 Reference Manual*, 2005.
- [24] C.B. Vreugdenhil. *Numerical Methods for Shallow Water Flow*. Kluwer Academic Publishers, 1994.