International Journal of Control
Vol. 83, No. 8, August 2010, 1686–1700

Taylor & Francis
Taylor & Francis Group

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

Andrew Tinka[a]*, Issam Strub[b], Qingfang Wu[c] and Alexandre M. Bayen[b]

[a]*Department of Electrical Engineering, University of California, Berkeley 94720, USA;* [b]*Department of Systems Engineering, Civil and Environmental Engineering, University of California, Berkeley 94720, USA;* [c]*Department of Environmental Engineering, Civil and Environmental Engineering, University of California, Berkeley 94720, USA*

(*Received 26 October 2009; final version received 24 April 2010*)

We present a method for assimilating Lagrangian sensor measurement data into a shallow water equation model. The underlying estimation problem (in which the dynamics of the system are represented by a system of partial differential equations) relies on the formulation of a minimisation of an error functional, which represents the mismatch between the estimate and the measurements. The corresponding so-called variational data assimilation problem is formulated as a quadratic programming problem with linear constraints. For the hydrodynamics application of interest, data is obtained from drifting sensors that gather position and velocity. The data assimilation method refines the estimate of the initial conditions of the hydrodynamic system. The method is implemented using a new sensor network hardware platform for gathering flow information from a river, which is presented in this article for the first time. 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.

Keywords: estimation; variational data assimilation; partial differential equations; wireless sensors; freshwater hydrodynamics

1. Introduction

1.1 *Background and motivation*

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 urbanisation and water-intensive agriculture, will increase freshwater demand significantly over the next 50 years (Postel 2000). Improving water use efficiency can help balance supply and demand (Jackson et al. 2001) and relieve scarcity; this will require improved methods for modelling and monitoring the flow of freshwater through the hydrological cycle (Rodda, Pieyns, Sehmi, and Matthews 1993). River flows are particularly important to investigate, as they constitute the majority of available renewable freshwater (Oki and Kanae 2006).

River hydraulics can be modelled with *shallow water equations* (SWEs) in one or two dimensions (Chadwick, Morfett, and Borthwick 2004). SWEs are a standard constitutive model 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 which originated in meteorology and oceanography (Le Dimet and Talagrand 1986).

There are many different sensor systems for measuring flow fields. They can be categorised 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 stream gauges, which measure the height of the water at one location, which can be used to infer the stream velocity under certain conditions; satellite imaging (Smith 1997), in which the river height is estimated directly with radar or indirectly by observing the water/shore interface; and statically mounted *acoustic Doppler current profilers* (ADCPs), which measure the Doppler shift in a returning acoustic pulse due to velocity. ADCPs can also be mounted to moving vehicles, such as submarines or surface watercraft. Assuming that the vehicle is not

*Corresponding author. Email: tinka@berkeley.edu

ISSN 0020–7179 print/ISSN 1366–5820 online

DOI: 10.1080/00207179.2010.489621
http://www.informaworld.com

travelling at the same speed as the water, the ADCP sensor in this case is neither Eulerian nor Lagrangian.

The trends of electronics miniaturisation 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. Numerous water management applications require a system state estimate described in Eulerian coordinates (e.g. to estimate whether the water level at certain points in the system is high enough for agricultural users to draw water, or whether the water quality over a particular fish migration route is adequate). This need for estimates in Eulerian coordinates requires new data assimilation methods to bring the Lagrangian data into an Eulerian framework. Most known implementations of Lagrangian data assimilation are in oceanography or meteorology (see e.g. Gunson and Malanotte-Rizzoli 1996; Navon 1997; Nodet 2006); in the specific case of hydraulic systems, Lagrangian data assimilation of shallow water flows to estimate the bottom topology is attempted in Honnorat, Monnier, and Le Dimet (2009). Assimilation of Lagrangian data to estimate boundary conditions (BCs) in a tidally influenced river is described in Strub, Percelay, Stacey, and Bayen (2009), and assimilation into a 1D model for a network of channels is described in Wu, Rafiee, Tinka, and Bayen (2009). An assimilation technique on a simplified hydrodynamic model is presented in Tinka, Strub, Wu, and Bayen (2009); we extend this work with a more realistic model, as well as a complete treatment of the experimental method, numerical schemes and hardware platform.

Most existing data assimilation methods can be categorised as either variational assimilation methods or sequential assimilation methods (Ide, Courtier, Ghil, and Lorenc 1997); the method presented in this article belongs in the variational assimilation category. Sequential methods, including Kalman filtering and its extensions (Evensen 2007) and optimal statistical interpolation (Molcard, Piterbarg, Griffa, Özgökmen, and Mariano 2003), perform a series of update/analysis steps, blending the observational data into the state estimate one step at a time. By contrast, variational methods, such as the ones used in Navon (1997) and Kamachi and O'Brien (1995), perform a single optimisation using all the observational data at once to minimise a cost functional. Most variational methods use general optimisation techniques such as gradient descent algorithms to find their optimal solution. The major distinction between the method we present here and the methods in the variational data assimilation literature is that we have used a formulation that results in a *quadratic programming* (QP) optimisation problem, instead of a more general one. QP problems can be more efficiently solved than general nonlinear optimisation problems.

Our objective is the development of an integrated system and architecture, including hardware, software, communication and visualisation that is capable of performing data assimilation for shallow water flows using GPS measurements from passive, 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 BCs of the system.

The theoretical contributions made in this article are a linearisation of the SWE that can be used for formulating the optimisation problem with linear constraints, and an inversion algorithm, using QP, which takes Lagrangian measurements and uses them for reconstruction of the distributed state. The practical contributions documented herein are the construction of a hardware data gathering infrastructure, namely a floating sensor network used to gather Lagrangian flow data, presented for the first time in this article; and a field operational test in the Georgiana Slough, including our additional instrumentation deployment for validation purposes. Finally, this article includes a validation procedure and corresponding results.

An earlier version of the linearisation and variational assimilation formulation presented herein is developed for shallow water flows in Strub, Percelay, Tossavainen, and Bayen (2009), for a different mathematical and operational context.

This article is organised as follows. In Section 2, we describe the PDE used to model the hydrodynamical systems under study. In Section 3, we explain the QP variational assimilation method we use. In Section 4, we introduce the Lagrangian passive floating sensor platform for gathering experimental data. In Section 5, 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 6 we present the conclusions of our study and suggest future avenues for research.

1.2 *Nomenclature*

For convenience, we present here some brief definitions of terms which may be unfamiliar to readers.

- *Passive drifting sensor*: A floating sensor that is deployed in a body of water and allowed to move with the water current. Its form is designed so as to mimic the motion of an 'ideal particle' in

the water. 'Passive' sensors have no way of modifying their trajectory; 'active' sensors may occasionally modify their trajectory using some form of propulsion.

- *Data assimilation*: The process of estimating some part of the state of a system by combining (or 'integrating') physical measurements with a model of the system. Data assimilation is often distinguished from general estimation problems by the sparsity of observation data compared to the dimension of the model. Historically, the study of data assimilation was developed in oceanography and meteorology, independently from the study of estimation which developed in the literature of control theory.
- *Variational data assimilation*: Data assimilation methods can be broadly categorised into 'sequential' methods and 'variational' methods (Ide et al. 1997). The difference between these methods, and the use of the term 'variational', is discussed in Sections 1.1 and 3.1.
- *Implicit and explicit finite difference schemes*: Finite difference methods are a category of approximation methods for numerical solutions of PDEs. For most PDEs, the time derivative is approximated with terms from the state at the n and $n+1$ time step. These schemes can be classified by whether the approximation for the spatial derivative includes terms from the state at time step n only, in which case the scheme is 'explicit', or whether it includes terms from time step $n+1$ as well, in which case the scheme is 'implicit'.

2. Hydrodynamic model

2.1 *Shallow water equations*

In the following, we use the SWE as our constitutive hydrodynamic model. We will present the equations, followed by a specific linearisation and discretisation. For legibility we suppress the arguments for dependent variables. The governing hydrodynamic equations for the modelled system are (Vreugdenhil 1994; EDF 2003):

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

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

$$\frac{\partial h}{\partial t} + \vec{u}\cdot\nabla h + h\nabla\cdot\vec{u} = 0. \tag{3}$$

Equations (1) and (2) represent momentum balance, and Equation (3) represents mass balance. They can be obtained from fundamental principles (conservation of mass for an incompressible fluid, and Newton's second law) in a differential setting (Sturm 2010). The symbol ∇ is used to denote the gradient operator and $\nabla\cdot$ is used to denote the divergence operator. The variables (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 metres; $b = b(x, y)$ is elevation of bottom surface in metres; $\eta = h + b = \eta(x, y, t)$ is free surface elevation in metres; 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 (EDF 2003); and $F_x = F_x(x, y, t)$, $F_y = F_y(x, y, t)$ are friction terms

$$F_x = -\frac{1}{\cos\theta}\frac{gn^2}{h^{4/3}}u\sqrt{u^2+v^2}, \tag{4}$$

$$F_y = -\frac{1}{\cos\theta}\frac{gn^2}{h^{4/3}}v\sqrt{u^2+v^2}, \tag{5}$$

where $\theta = \theta(x, y)$ is the slope of the river bed; n is the Manning coefficient. The Manning coefficient is an empirical term that depends on the roughness of the channel bed. For this study we took the Manning coefficient to be 0.04 uniformly over the domain. This is a reasonable estimate for a major natural stream without significant brush obstruction on the banks (Sturm 2010). The friction terms are derived from the classic Gauckler–Manning hydraulic equation (Chow 1959), using the shallow water assumption (that the depth of the water is significantly smaller than the cross-sectional width of the channel).

Following common assumptions in fluvial hydraulics, our first simplification is to neglect the turbulence terms. We linearise the equations about a steady but non-uniform flow $U^0(x, y)$, $V^0(x, y)$, $H^0(x, y)$ that satisfies Equations (1)–(3):

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

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

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

with the choice of

$$C = \frac{1}{\cos\theta}\frac{gn^2}{H^{0^{4/3}}}\sqrt{U^{0^2} + V^{0^2}} \tag{9}$$

as the linearised friction coefficient.

2.2 *Non-orthogonal curvilinear grid*

For general geometries, the river region does not line up well with the Cartesian axes. Discretising using a Cartesian mesh would be inefficient; the grid size would have to be very fine in order to capture the spatial features properly. Generating a non-orthogonal grid to efficiently cover the domain while also representing the boundary is a standard problem with many approaches in the literature. For this study, the domain was manually decomposed into large, approximately rectangular regions that were then algorithmically subdivided into smaller quads. This is called the 'macro-element' or 'multi-block' approach, and was first described in Zienkiewicz and Phillips (1971). The interested reader is referred to Löhner (2008) for an overview of grid generation methods. The coordinate changes are characterised by the deviation of the local axes from the Cartesian axes, called α and β, respectively (Figure 1) (George 2007). 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 (Figure 1). Covariant velocity, denoted by u_{CL}, v_{CL} is used for the momentum balance equations (1) and (2), while contravariant velocity, denoted by u_{CV}, v_{CV} is used for the mass balance equation (3).

A summary of the derivations in George (2007) is presented here. In order to develop expressions for the curvilinear and contravariant velocity in terms of the Cartesian velocity components u and v, it is convenient to begin by expressing the velocity in polar coordinates R, θ with respect to the Cartesian axes. Expressions for the curvilinear velocity are derived by analysing triangle $\triangle OCV$ with the sine rule, then solving for u_{CL} and v_{CL}:

$$\frac{R}{\sin(\frac{\pi}{2}+\alpha-\beta)} = \frac{u_{\mathrm{CL}}}{\sin(\frac{\pi}{2}-\theta+\beta)} = \frac{v_{\mathrm{CL}}}{\sin(\theta-\alpha)},$$

$$u_{\mathrm{CL}} = R\cos(\theta-\beta)\sec(\alpha-\beta),$$

$$v_{\mathrm{CL}} = R\sin(\theta-\alpha)\sec(\alpha-\beta).$$

The process for the contravariant velocity is the same, analysing triangle $\triangle ODV$:

$$\frac{R}{\sin(\frac{\pi}{2}-\alpha+\beta)} = \frac{u_{\mathrm{CV}}}{\sin(\frac{\pi}{2}-\theta+\alpha)} = \frac{v_{\mathrm{CV}}}{\sin(\theta-\beta)},$$

$$u_{\mathrm{CV}} = R\cos(\theta-\alpha)\sec(\alpha-\beta),$$

$$v_{\mathrm{CV}} = R\sin(\theta-\beta)\sec(\alpha-\beta).$$

Using the sum and difference identities, we expand the $\{\cos, \sin\}(\theta\pm\{\alpha,\beta\})$ terms, and substitute $R\cos(\theta)=u$ and $R\sin(\theta)=v$. This leads to systems of equations which can be solved for the forward and inverse transforms between Cartesian velocity components and the two types of curvilinear components.

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

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\beta \\ \sin\alpha & \cos\beta \end{bmatrix}\begin{bmatrix} u_{\mathrm{CL}} \\ v_{\mathrm{CL}} \end{bmatrix},$$

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

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\alpha \\ \sin\beta & \cos\alpha \end{bmatrix}\begin{bmatrix} u_{\mathrm{CV}} \\ v_{\mathrm{CV}} \end{bmatrix},$$

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

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

All other variables have trivial transformations, and we will abuse notation by not distinguishing them from their original forms.

The linearised SWEs (6)–(8) are transformed into the curvilinear coordinates (George 2007):

$$\frac{\partial u_{\mathrm{CL}}}{\partial t} + U^0_{\mathrm{CL}}\frac{\partial u_{\mathrm{CL}}}{\partial x_{\mathrm{CL}}} + V^0_{\mathrm{CL}}\frac{\partial u_{\mathrm{CL}}}{\partial y_{\mathrm{CL}}} + \left(U^0_{\mathrm{CL}}\frac{\partial u_{\mathrm{CL}}}{\partial y_{\mathrm{CL}}} + V^0_{\mathrm{CL}}\frac{\partial u_{\mathrm{CL}}}{\partial x_{\mathrm{CL}}}\right)\sin(\alpha-\beta) = -g\frac{\partial h}{\partial x_{\mathrm{CL}}} - g\frac{\partial b}{\partial x_{\mathrm{CL}}} + Cu_{\mathrm{CL}}, \tag{10}$$

$$\frac{\partial v_{\mathrm{CL}}}{\partial t} + U^0_{\mathrm{CL}}\frac{\partial v_{\mathrm{CL}}}{\partial x_{\mathrm{CL}}} + V^0_{\mathrm{CL}}\frac{\partial v_{\mathrm{CL}}}{\partial y_{\mathrm{CL}}} + \left(U^0_{\mathrm{CL}}\frac{\partial v_{\mathrm{CL}}}{\partial y_{\mathrm{CL}}} + V^0_{\mathrm{CL}}\frac{\partial v_{\mathrm{CL}}}{\partial x_{\mathrm{CL}}}\right)\sin(\alpha-\beta) = -g\frac{\partial h}{\partial y_{\mathrm{CL}}} - g\frac{\partial b}{\partial y_{\mathrm{CL}}} + Cv_{\mathrm{CL}}, \tag{11}$$

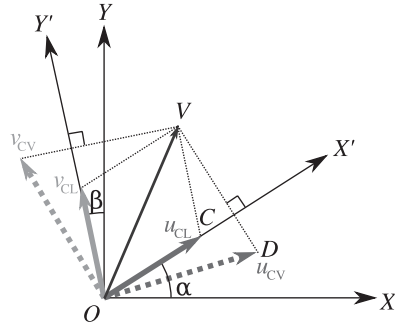

Figure 1. Example of non-orthogonal curvilinear axes. OX, OY: global Cartesian axes. OX', OY': local non-orthogonal curvilinear axes. OV: velocity vector. u_{CL}, v_{CL}, u_{CV}, v_{CV}: curvilinear and contravariant components of OV.

$$\frac{\partial h}{\partial t} + U^0_{\mathrm{CV}} \frac{\partial h}{\partial x_{\mathrm{CL}}} \sec(\alpha - \beta) + V^0_{\mathrm{CV}} \frac{\partial h}{\partial y_{\mathrm{CL}}} \sec(\alpha - \beta) + H^0 \left(\frac{\partial u_{\mathrm{CV}}}{\partial x_{\mathrm{CL}}} + \frac{\partial v_{\mathrm{CV}}}{\partial y_{\mathrm{CL}}} \right) \sec(\alpha - \beta) = 0. \quad (12).$$

These transformed equations are algebraically more involved, but from a practical perspective, simply add static trigonometric terms to the discretised scheme (16)–(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.

2.3 *Boundary conditions*

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

$$\vec{u} \cdot \vec{s}\big|_{\partial\Omega_{\mathrm{land}}} = 0, \quad (13)$$

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 nonlinear terms in the original momentum equations (1) and (2). This is a common BC in the hydrodynamics literature, and sometimes called the 'slip BC'; see, for example George (2007) and Honnorat et al. (2009). The implicit assumptions for the slip BC are that the bathymetry is steep enough at the shore that the water height will not significantly affect the location of the land boundary, and that there is no significant influx or outflux of water from or into the soil. This would be an inappropriate assumption, for example, for a shallow mud flat that wets and dries during the tidal cycle (Whitehouse 2000; Uchiyama 2004); but for steep banked river channels, such as the ones studied in this article, the assumption is justified.

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 BCs are implicitly defined as being equal to the value at the IC:

$$\vec{u}(t)\big|_{\partial\Omega_{\mathrm{upstream}}} = \vec{u}(0)\big|_{\partial\Omega_{\mathrm{upstream}}}, \quad (14)$$

$$h(t)\big|_{\partial\Omega_{\mathrm{downstream}}} = h(0)\big|_{\partial\Omega_{\mathrm{downstream}}}. \quad (15)$$

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

2.4 *Discretisation*

We use an implicit discretisation scheme, consisting of backward Euler for the time derivative and centred differencing for the spatial derivatives, also known as the *backwards time centred space (BTCS) method* (Hoffman 2001). This scheme was chosen for computational efficiency, since it is an *implicit*, *single step* scheme. The number of decision variables in the optimisation programme is proportional to the number of time steps in the discretisation, which will be seen in Section 3.1. A multi-step scheme (where intermediate states are computed for each time step) would result in a similar proportional growth of decision variables. Implicit methods, named for their *implicit* time update equation, are not constrained by the *Courant–Friedrichs–Lewy* (CFL) stability condition on the time step. Using an explicit method would require satisfying the CFL condition, which would require more time steps, which would increase the number of decision variables in the variational assimilation problem. As will be discussed in Section (3.1), however, the method developed in this article is applicable to all single and multi-step schemes, explicit or implicit.

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).

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

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

$$\begin{aligned}\frac{h_{i,j}^{k+1}-h_{i,j}^{k}}{\Delta t} = &-\sec(\gamma_{i,j})\sec(\gamma_{i,j})U_{\mathrm{CL}i,j}^{0}\frac{h_{i+1,j}^{k+1}-h_{i-1,j}^{k+1}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &-\sec(\gamma_{i,j})\tan(\gamma_{i,j})V_{\mathrm{CL}i,j}^{0}\frac{h_{i+1,j}^{k+1}-h_{i-1,j}^{k+1}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &-\sec(\gamma_{i,j})\tan(\gamma_{i,j})U_{\mathrm{CL}i,j}^{0}\frac{h_{i,j+1}^{k+1}-h_{i,j-1}^{k+1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &-\sec(\gamma_{i,j})\sec(\gamma_{i,j})V_{\mathrm{CL}i,j}^{0}\frac{h_{i,j+1}^{k+1}-h_{i,j-1}^{k+1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &-\sec(\gamma_{i,j})\sec(\gamma_{i,j})H_{i,j}^{0}\frac{u_{\mathrm{CL}i+1,j}^{k+1}-u_{\mathrm{CL}i-1,j}^{k+1}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &-\sec(\gamma_{i,j})\tan(\gamma_{i,j})H_{i,j}^{0}\frac{u_{\mathrm{CL}i+1,j}^{k+1}-u_{\mathrm{CL}i-1,j}^{k+1}}{\Delta x_{i-1,j}+\Delta x_{i,j}}\\ &-\sec(\gamma_{i,j})\tan(\gamma_{i,j})H_{i,j}^{0}\frac{u_{\mathrm{CL}i,j+1}^{k+1}-u_{\mathrm{CL}i,j-1}^{k+1}}{\Delta y_{i,j-1}+\Delta y_{i,j}}\\ &-\sec(\gamma_{i,j})\sec(\gamma_{i,j})H_{i,j}^{0}\frac{v_{\mathrm{CL}i,j+1}^{k+1}-v_{\mathrm{CL}i,j-1}^{k+1}}{\Delta y_{i,j-1}+\Delta y_{i,j}},\end{aligned} \tag{18}$$

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 $(i,j+1)$. $\gamma_{i,j}$ is an abbreviation for $\alpha_{i,j}-\beta_{i,j}$.

3. QP-based variational data assimilation

3.1 *Variational framework*

Our method uses *variational data assimilation*. The variables in the discretised equations (16)–(18) are concatenated into vectors, using the standardised framework set out in Ide et al. (1997), 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 improve well-posedness of the problem.
- Y_n Vector of observed variables at time t_n. No observations are taken at time 0.
- 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.

The method is referred to as *variational* data assimilation because the method estimates the optimum IC functions $u(x, y, 0)$, $v(x, y, 0)$ and $h(x, y, 0)$. The objective function expressed in (19) is a function of the discretised state, but in the original formulation the objective is actually a *functional* of the ICs. This terminology serves to distinguish the method from the broad category of *sequential* methods, which typically estimate the state at the observation times (Ide et al. 1997). It can be argued that, because of the discretisation, the objective is no longer a functional and that the method is no longer variational. The terminology is useful, however, for placing our method in context with the data assimilation literature.

Our data assimilation strategy is to search for the initial state X_0 that minimises 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:

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

X_B, the background term, is a 'first guess' about the state of the system; it is expected that it is inaccurate, and that the product of the assimilation will be a refined estimate of the state. The background term could be derived from historical data, from forecasts, from a previous assimilation, or from forward simulation based on BCs (either observed or artificially generated). The need for a background term is discussed further in Section 3.3.

The covariance matrices B and R_n affect the weight given to the background term and the observations. Choosing appropriate values for these covariances is discussed in Section 3.3. As a simplifying assumption we take these matrices to be a scalar times the identity matrix: $b\mathbb{I}$ and $r\mathbb{I}$, respectively.

General variational data assimilation schemes treat the observation operator H_n as a non-linear operator; however, as described in Section 4, 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 context of estimation on linear systems, H_n would be called the *observation matrix* (Stengel 1994). 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 synchronised with assimilation steps, then the values in the H_n matrix should reflect the polynomial approximation associated with the time discretisation 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 generalised to any linear multi-step method.

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

$$EX_{n+1} = FX_n + g, \tag{20}$$

where E and F are matrices determined by the time and space difference schemes (16)–(18), and g is a vector capturing source terms that do not depend on the state, such as the bottom elevation. This constraint formulation allows implicit discretisation schemes to be implemented, which broadens the applicability of the method significantly. Explicit schemes have a conceptually simpler process update (the E matrix is usually the identity matrix), but the time step used in these schemes is restricted by the CFL condition, and can often be inconveniently short.

3.2 *Quadratic programme*

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

$$\begin{aligned} \text{minimise} \quad & \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{G}\mathbf{x} \leq \mathbf{h} \\ & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned} \tag{21}$$

The variables in bold are from standard optimisation formulations (Boyd and Vandenberghe 2004) 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, as shown here:

$$\mathbf{P} = \begin{bmatrix} B^{-1} & & & & \\ & H_1^T R_1^{-1} H_1 & & & \\ & & H_2^T R_2^{-1} H_2 & & \\ & & & \ddots & \\ & & & & H_{n_{\max}}^T R_{n_{\max}}^{-1} H_{n_{\max}} \end{bmatrix}, \tag{22a}$$

$$\mathbf{q} = \begin{bmatrix} -X_B^T B^{-1} & Y_1^T R_1^{-1} H_1 & Y_2^T R_2^{-1} H_2 & \cdots & Y_{n_{\max}}^T R_{n_{\max}}^{-1} H_{n_{\max}} \end{bmatrix}. \tag{22b}$$

The equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ represents the flow dynamic constraints described in (20), and can be expanded as

$$\begin{bmatrix} -F & E & & & \\ & -F & E & & \\ & & \ddots & \ddots & \\ & & & -F & E \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{n_{\max}} \end{bmatrix} = \begin{bmatrix} g \\ g \\ \vdots \\ g \end{bmatrix}, \tag{23}$$

$\mathbf{G}$ and $\mathbf{h}$ are normally zero, although we may impose heuristic inequality constraints to reduce the search space, in particular for ICs and BCs.

3.3 *Background term and well-posedness of the problem*

It is a standard result in the study of hyperbolic PDEs, discussed in Drolet and Gray (1988), that the Cauchy problem associated with a system of SWEs (1)–(3), for *subcritical* flow, with ICs provided everywhere, and BCs (13)–(15), is *well-posed* as defined by Hadamard. The 'subcriticality' of the flow is a condition that can be formally defined, and has to do with the ratio of the velocity field and the eigenvalues of a matrix which appears when Equations (1)–(3) are written in conservation law form. In physical terms, it means that the velocity of the water is always less than the wave propagation speed. This is a reasonable assumption for our application. 'Well-posedness' means that (a) a solution exists, (b) the solution is unique and (c) the solution depends continuously on the ICs and BCs. We are interested in two of the analogous properties for the variational data assimilation problem (21); namely that

(1) there exists an optimal $\mathbf{x}$,
(2) the optimising $\mathbf{x}$ is unique.

A minimiser to (21) exists because of the convexity of the objective function ($\mathbf{P}$ is positive semidefinite) and the bounded, non-empty constraint set.

Observations are usually sparse compared to the number of mesh points; the H_i matrices are rank-deficient, and therefore the matrix $\mathbf{P}$ is positive semidefinite as opposed to positive definite. Thus, the uniqueness of an optimum $\mathbf{x}$ is not guaranteed; this is a common problem in oceanography and similar fields relying on data assimilation. However, we can argue that by incorporating the background term X_B, we restrict $\mathbf{x}$ to a set of smaller dimension than it would be otherwise.

Assume for argument that $\mathbf{x}^*$ is a feasible, optimal solution to the QP (21), and that $\frac{1}{2}\mathbf{x}^{*T}\mathbf{P}\mathbf{x}^* + \mathbf{q}^T\mathbf{x}^* = j^*$. For simplicity, we will drop the $\mathbf{G}\mathbf{x} \leq \mathbf{h}$ constraint. To investigate the uniqueness of $\mathbf{x}^*$, we consider the

related feasibility problem

$$\frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} = j^*, \quad \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{24}$$

We already have one feasible solution, so we consider $\mathbf{x} = \mathbf{x}^* + \mathbf{x}'$:

$$\frac{1}{2}\mathbf{x}'^T\mathbf{P}\mathbf{x}' + \left(\mathbf{x}^{*T}\mathbf{P} + \mathbf{q}^T\right)\mathbf{x}' = 0, \tag{25a}$$

$$\mathbf{A}\mathbf{x}' = 0, \tag{25b}$$

$\mathbf{x}^*$ is the *unique* optimising solution for (21) if and only if (25a) and (25b) admit a single solution in $\mathbf{x}'$, the zero vector.

The uniqueness of the solution to (21) then becomes a geometric question: whether the intersection of the quadratic hypersurface defined by (25a) and the linear subspace defined by (25b) contains just a single point (the zero vector) or multiple points; and if so, how those multiple points can be characterised. Since $\mathbf{P}$ is positive *semi*definite, the quadratic hypersurface defined by (25a) is degenerate, and so we might expect the set of solutions to include linear subspaces; however, the block structure of $\mathbf{P}$ and $\mathbf{A}$, when a background term is used, preclude this possibility.

Assume for contradiction that $\mathbf{x}'$ is a non-zero member of a linear subspace that satisfies (25a) and (25b); in other words, $\lambda\mathbf{x}'$ also satisfies (25a) and (25b) for any scalar λ. Because (25a) is the sum of a linear and quadratic term, it is then necessarily true that $\mathbf{x}'^T\mathbf{P}\mathbf{x}' = 0$. Recalling the block structure of $\mathbf{P}$ from (22a),

$$\mathbf{x}'^T\mathbf{P}\mathbf{x}' = X_0'^T B^{-1} X_0' + \sum_{i=1}^{n_{\max}} X_i' T H_i^T R_i^{-1} H_i X_i',$$

B^{-1} is positive definite, unlike all the other blocks of $\mathbf{P}$, so $X_0' = 0$. But the block structure of $\mathbf{A}$, shown in (23), means that if $X_0' = 0$, all the other X_i' vectors must be zero as well. This contradicts our assumption that $\mathbf{x}'$ is non-zero; and so the feasibility problem (25a) and (25b) does not admit any non-trivial linear subspaces.

Unfortunately, the 'true' uniqueness of $\mathbf{x}^*$ is not guaranteed. Equations (25a) and (25b) still admit solutions where $\mathbf{x}'^T\mathbf{P}\mathbf{x}' \neq 0$. But these solutions are not linear subspaces; for any non-zero $\mathbf{x}'$ satisfying (25a) and (25b), the only scalar values of λ for which $\lambda\mathbf{x}'$ also satisfies (25a) and (25b) are 0 and 1.

If we did not use a background term for our data assimilation problem, $\mathbf{P}$ would have no strictly positive definite block, and the contradiction proof above would not work. The set of optimal solutions to (21) could include linear subspaces as well as non-degenerate quadratic hypersurfaces. By adding a background term, we exclude all linear subspaces from the set of optimal solutions.

3.4 *Choice of the covariance matrices*

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 discussed above, we have assumed that $B = b\mathbb{I}$ and $R_i = r\mathbb{I}$ for simplicity. A reasonable choice for r can be made from the accuracy specifications of the GPS module. Choosing b is less clear. If b is too small, and the eigenvalues of B^{-1} become large, the assimilation process will over-emphasise the background term, and the observations will not significantly affect the final estimate. If b is too large, the regularising effect of the background term will be insufficient to improve the convergence of the algorithm.

There is also an important relationship between the choice of B and the accuracy of the X_B term. If we had a trusted estimate of the covariance of the error between the background term and the true state, then the correct choice of B is obvious; however, this is not practically useful, since the background term is almost always a 'guess'. Estimating the error of the background term, when the background term is generated from guessing, simulations or historical data, is an extremely non-trivial task.

Clearly, it would be advantageous to be able to use 'bad' background terms and still get accurate assimilation results. In many ways, the background term encodes assumptions about qualitative properties of a valid solution: for example, the overall direction of water movement, the relative uniformity of water height, the fact that the water velocity is locally roughly parallel, etc. If the background term violates these assumptions, a valid solution is extremely unlikely. For example, if we used a background term with water flowing in the wrong direction, we would be asking the optimisation algorithm to find a valid PDE solution with no external sources that somehow performed a complete flow reversal in the short time period between the initial time and the first observation. The result would depend on the weighting of the B and R matrices, but would probably be some intermediate value, inconsistent with both the background and the observations. In this sense, we see that the background term must be *qualitatively* accurate. Establishing the requirements for *quantitative* accuracy, and the corresponding choice of B, is an open problem.

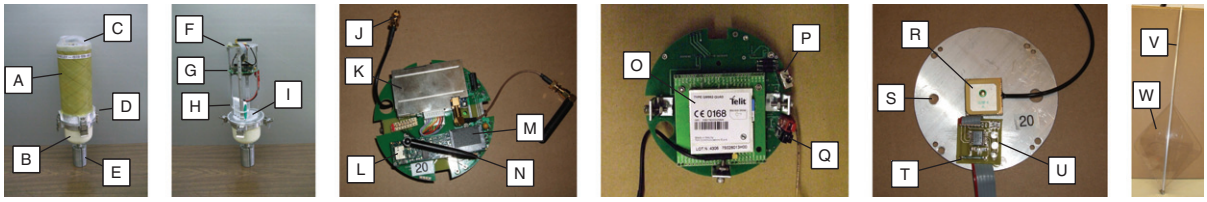

Figure 2. Drifter hardware. (A) Fibreglass pipe (B) Cast fibreglass (C) Polycarbonate cap (D) Clasp (E) Stand (not part of drifter) (F) Antenna plate (G) Electronics (H) Battery (I) Bulkhead (J) GSM antenna connector (K) GPS module (L) Gumstix module (M) MMC card (N) Bluetooth antenna (O) GSM module (P) Power switch (Q) Battery connector (R) GPS antenna (S) Hole for GSM antenna (not shown) (T) Magnetic switch (U) Status LEDs (V) Aluminium tube (W) Polycarbonate plate.

3.5 *Convergence of QP solutions to valid PDE solutions*

A feasible solution to the quadratic programme (21) will, by construction, satisfy the finite difference equation system (16)–(18).

Hence, in the limit case in which the discretisation size converges to zero, the solution of the discretised system (16)–(18) converges to the solution of the linearised PDE system (6)–(8). This is a known result of the Lax equivalence theorem, applied to the BTCS method, which is unconditionally stable (Hoffman 2001).

The linearised PDE system (6)–(8) is an approximation of the original PDE system (1)–(3). These linearisations are standard in fluid mechanics, and obtained procedurally as follows. A nominal flow of the nonlinear PDE is constructed, around which a first-order Taylor expansion of the PDE system is derived, which leads to the *linear* PDE system. Accordingly, the error between the two models is of order $O(u^2)$, $O(v^2)$, $O(h^2)$, as well as the neglected terms. Extending properties of linearised hyperbolic conservation PDE systems to their nonlinear counterparts has examples in the literature; for example, Coron, d'Andréa-Novel, and Bastin (2007) showed that stability properties for a 1D linearised shallow water system under boundary control can be extended to the nonlinear case. Extending similar properties for the estimation problem under the conditions of this article (arbitrary geometry, variable BCs) is beyond the mathematical scope of this study. However, it is likely that specific results could be proven for simpler geometries and idealised conditions.

Our case for the applicability of the solutions derived from our assimilation method is thus a three-step argument: (1) the QP solution is an exact solution for the discretised system; (2) the solution to the discretised system converges to the solution to the linearised PDE as the step sizes go to zero; (3) the error in the linearised model is of quadratic order compared to the fully nonlinear PDE model.

4. Hardware platform

We now present the passive 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 10 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 h 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 fibreglass pipe. The top cap is vacuum-formed polycarbonate. The lower shell is hand-cast fibreglass. The top hull and bottom hull are joined with epoxy to machined aluminium 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 aluminium 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^2, 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.

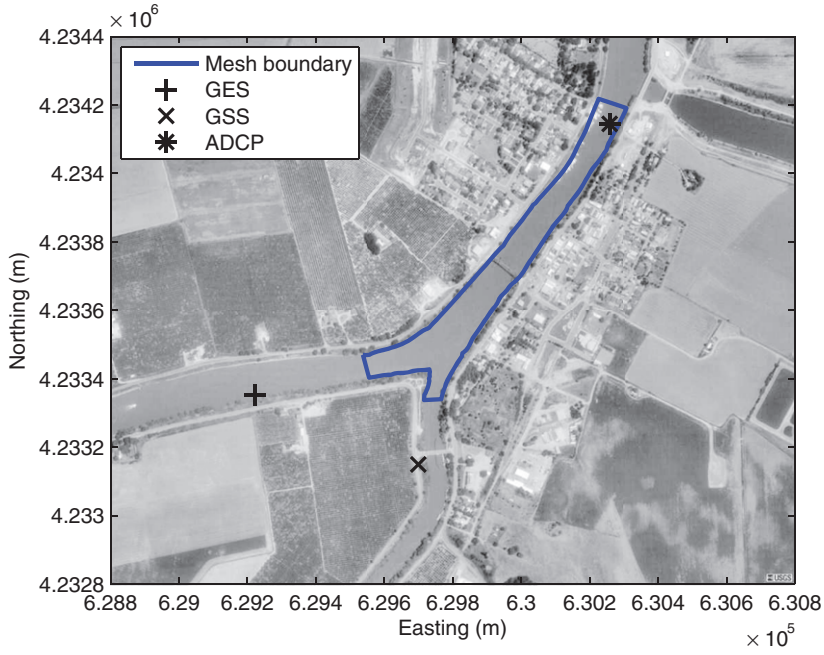

Figure 3. Sacramento River/Georgiana Slough with modelled area, ground stations and sensor deployment locations. Image courtesy of USGS.

- Power: The drifter carries a 10.4 A-h, 3.7 V 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 (Intel Corporation 2003). 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 pseudorange and carrier phase data for post-processing (Thales Navigation 2005).
- 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 (Telit Communications 2006).

5. Field test

5.1 *Available data*

The proposed 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 10 drifters; in the experiments presented here, up to eight were deployed at a time.
- *DSM-II historical data*: DSM-II (Anderson and Mierzwa 2002) 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 at the boundaries of the region of interest, using sensors described below. This data was used as the BCs for a forward simulation using TELEMAC, a commercial hydrodynamics simulator. TELEMAC is essentially a specialised PDE solver for the SWEs; given the ICs and BCs, it finds the velocity at all points in the mesh through a forward simulation of the equation. Since actual measurement of the ICs was unavailable, we used the standard technique of starting with an arbitrary IC, holding the BCs steady, and running the simulation for a long time, essentially 'washing away' the arbitrary IC. 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 (Figure 3):

- 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 BC.

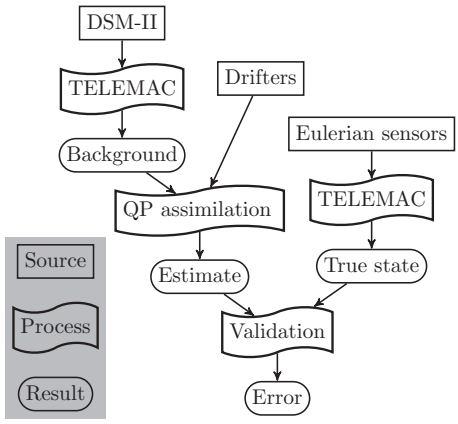

Figure 4. Data flow diagram used for the data assimilation using the hardware platform.

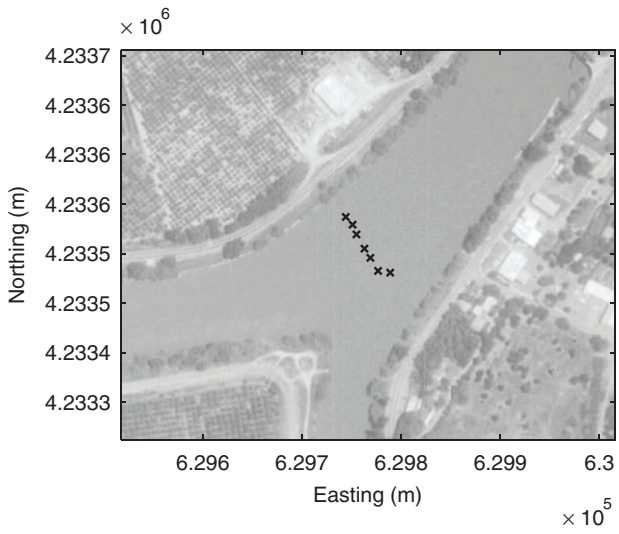

Figure 5. Example of drop points for drifter release in the final experiment.

- *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)–(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 4 shows how the various data are used. Historical DSM-II data is used, with TELEMAC 2D (EDF 2003) 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.

5.2 *Experimental strategy*

Eight drifter deployments were performed from 12 November to 16 November 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 7 and 10 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 centre 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 16 November. The drifters were monitored as they travelled in the river. Each experiment was planned to last between 45 and 60 min; 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 5 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 h, generated by the Parks–McClellan optimum filter algorithm (IEEE 1979)). To better show the length of the experiments, and their relationship to the tidal cycle, the filtered velocity signal for all 5 days was superimposed in Figure 7, referenced to the minor maximum of the velocity.

Post-experiment analysis showed that several drifters did not record sufficient amounts of GPS data (e.g. less than 10% of the expected amount of data); 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.

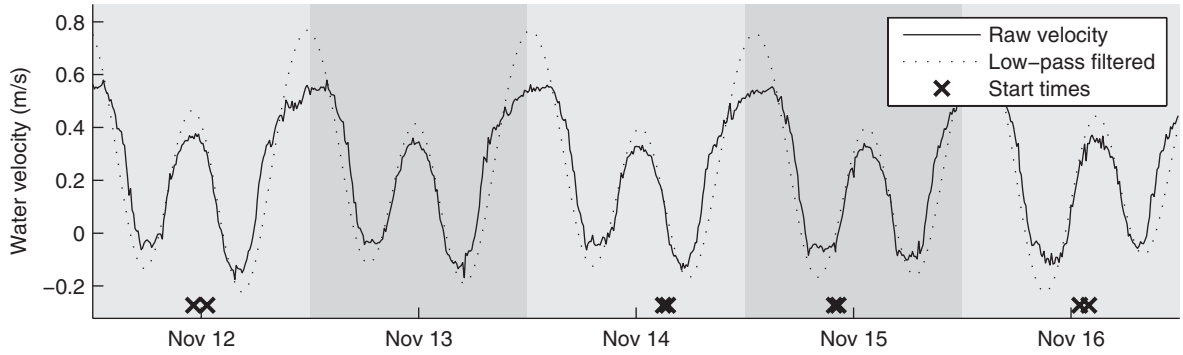

Figure 6. Overview of experimental timeframe.

5.3 *Implementation of the algorithm*

The drifter measurements were sampled at 30 s. 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 BCs 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, discretised, linearised PDE equations as linear constraints, as described in Section 3. 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 optimisation modelling language AMPL and solved with CPLEX. The optimal IC 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 a form of interpolation, the data can be passed as-is to the QP assimilation process.

5.4 *Validation*

A forward simulation of the region of interest was performed using the data from Eulerian sensors. This data was used as the BCs for an 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 IC 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:

$$\epsilon_E = \sqrt{\frac{\left(\sum_j (u_{Tj} - u_j)^2 + (v_{Tj} - v_j)^2\right)}{\left(\sum_j \left(u_{Tj}^2 + v_{Tj}^2\right)\right)}}, \qquad (26)$$

where j is the node index.

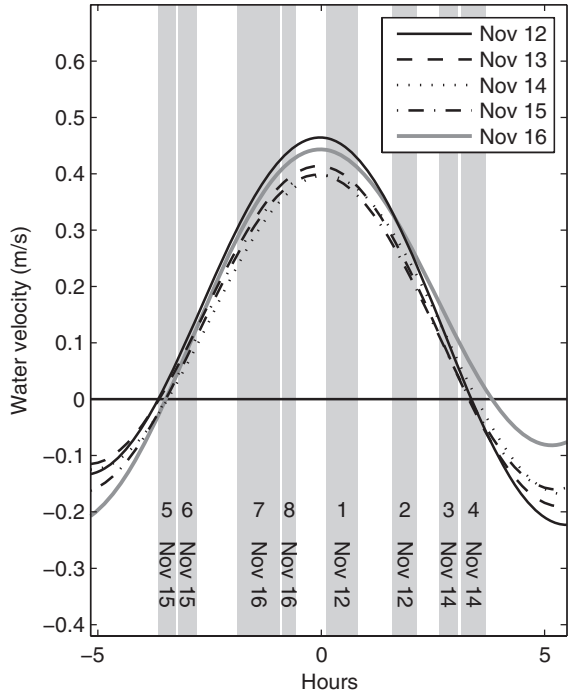

Figure 7. Experiment times shown relative to minor velocity peak.

5.5 *Results*

Figure 8 shows the initial flow field condition assimilated by the QP algorithm for one of the experiments. The height variable is very smooth (differing by only a

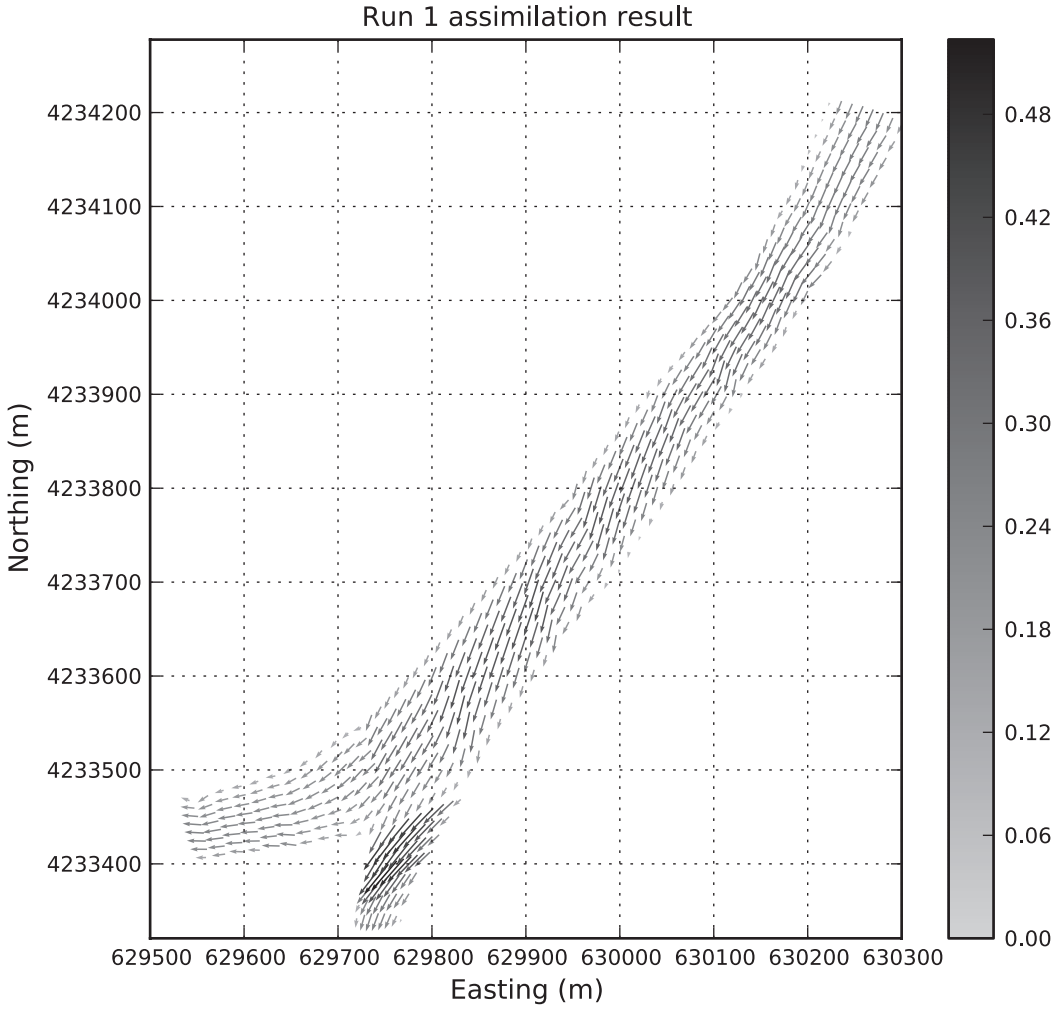

Figure 8. Assimilated velocity field IC for experiment 1. Scale is in m/s. Decimated by two for legibility.

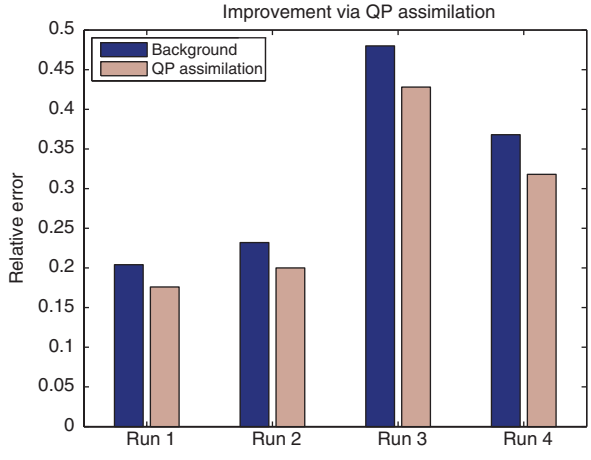

Figure 9. Change in relative error for each of four experiments.

few centimetres 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), performing a QP assimilation of 30 min worth of drifter data (from 5 to 8 drifters) takes approximately 10 min on a single 2.0 GHz processor.

6. Conclusion

In this article, we present a method for formulating the variational data assimilation problem for Lagrangian sensors in shallow water flows as a QP optimisation

problem with linear constraints. A major advantage of the QP formulation is that the constraints can express the model PDE discretised with an implicit scheme. This allows our method to use longer time steps than explicit methods. Our method also assimilates on the ICs, in contrast to many sequential methods.

The QP 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.)

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 modelling 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. Ultimately, we hope to demonstrate a method that can produce useful assimilations even when the background term is severely different from the true state. Further research into the accuracy relationships between background, observations, and results will also shed light on the role of the number of sensors and sensor accuracy on the results. The question of how many sensors are required for an accurate estimate of the system is a very interesting open problem.

Acknowledgements

The authors 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

Anderson, J., and Mierzwa, M. (2002), 'DSM2 Tutorial, an Introduction to the Delta Simulation Model II (DSM2)', Technical Report, State of California, Department of Water Resources.

Boyd, S., and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge: Cambridge University Press.

Chadwick, A., Morfett, J., and Borthwick, M. (2004), *Hydraulics in Civil and Environmental Engineering*, London: Spon Press.

Chow, V.T. (1959), *Open-Channel Hydraulics*, New York: McGraw-Hill.

Coron, J.M., d'Andréa-Novel, B., and Bastin, G. (2007), 'A Strict Lyapunov Function for Boundary Control of Hyperbolic Systems of Conservation Laws', *IEEE Transactions on Automatic Control*, 52, 2–11.

Drolet, J., and Gray, W.G. (1988), 'On the Well Posedness of Some Wave Formulations of the Shallow Water Equations', *Advances in Water Resources*, 11, 84–91.

EDF. (2003), 'TELEMAC 2D. Version 5.2', Technical Report, EDF.

Evensen, G. (2007), *Data Assimilation: The Ensemble Kalman Filter*, New York: Springer-Verlag.

George, K. (2007), 'A Depth-averaged Tidal Numerical Model using Non-orthogonal Curvilinear Co-ordinates', *Ocean Dynamics*, 57, 363–374.

Gunson, J.R., and Malanotte-Rizzoli, P. (1996), 'Assimilation Studies of Open-ocean Flows', *Journal of Geophysical Research*, 101, 28457–28472.

Hoffman, J.D. (2001), *Numerical Methods for Engineers and Scientists* (2nd ed.), New York, NY: Marcel Dekker.

Honnorat, M., Monnier, J., and Le Dimet, F.X. (2009), 'Lagrangian Data Assimilation for River Hydraulics Simulations', *Computing and Visualization in Science*, 12, 235–246.

Ide, K., Courtier, P., Ghil, M., and Lorenc, A. (1997), 'Unified Notation for Data Assimilation: Operational, Sequential and Variational', *Journal of the Meteordogical Society of Japan*, 75, 71–79.

IEEE. (1979), *Programs for Digital Signal Processing*, New York: IEEE Press.

Intel Corporation. (2003), *Intel PXA255 Processor Design Guide*.

Jackson, R.B., Carpenter, S.R., Dahm, C.N., McKnight, D.M., Naiman, R.J., Postel, S.L., and Running, S.W. (2001), 'Water in a Changing World', *Ecological Applications*, 11, 1027–1045.

Kamachi, M., and O'Brien, J.J. (1995), 'Continuous Data Assimilation of Drifting Buoy Trajectory into an

Equatorial Pacific Ocean Model', *Journal of Marine Systems*, 6, 159–178.

Le Dimet, F.X., and Talagrand, O. (1986), 'Variational Algorithms for Analysis and Assimilation of Meterological Observations: Theoretical Aspects', *Tellus*, 38A, 97–110.

Löhner, R. (2008), *Applied CFD Techniques* (2nd ed.), New York: John Wiley & Sons.

Molcard, A., Piterbarg, L., Griffa, A., Özgökmen, T., and Mariano, A. (2003), 'Assimilation of Drifter Observations for the Reconstruction of the Eulerian Circulation Field', *Journal of Geophysical Research*, 108, 3056–3076.

Navon, I.M. (1997), 'Practical and Theoretical Aspects of Adjoint Parameter Estimation and Identification in Meteorology and Oceanography', *Dynamics of Atmospheres and Oceans*, 27, 55–79.

Nodet, M. (2006), 'Variational Assimilation of Lagrangian Data in Oceanography', *Inverse Problems*, 22, 245–263.

Oki, T., and Kanae, S. (2006), 'Global Hydrological Cycles and World Water Resources', *Science*, 313, 1068–1072.

Postel, S.L. (2000), 'Entering an Era of Water Scarcity: The Challenges Ahead', *Ecological Applications*, 10, 941–948.

Rodda, J.C., Pieyns, S.A., Sehmi, N.S., and Matthews, G. (1993), 'Towards a World Hydrological Cycle Observing System', *Hydrological Sciences Journal*, 38, 373–378.

Smith, L.C. (1997), 'Satellite Remote Sensing of River Inundation Area, Stage, and Discharge: A Review', *Hydrological Processes*, 11, 1427–1439.

Stengel, R.F. (1994), *Optimal Control and Estimation*, New York: Dover.

Strub, I.S., Percelay, J., Stacey, M.T., and Bayen, A.M. (2009), 'Inverse Estimation of Open Boundary Conditions in Tidal Channels', *Ocean Modelling*, 29, 85–93.

Strub, I.S., Percelay, J., Tossavainen, O.P., and Bayen, A.M. (2009), 'Comparison of Two Data Assimilation Algorithms for Shallow Water Flows', *Networks and Heterogenous Media*, 4, 409–430.

Sturm, T.W. (2010), *Open Channel Hydraulics* (2nd ed.), Boston: McGraw-Hill.

Telit Communications, (2006), *GM862-QUAD/GM862-QUAD-PY Hardware User Guide*.

Thales Navigation. (2005), *A12, B12, & AC12 Reference Manual*.

Tinka, A., Strub, I., Wu, Q., and Bayen, A.M. (2009), 'Quadratic Programming Based Data Assimilation with Passive Drifting Sensors for Shallow Water Flows', in *Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, P.R. China, pp. 7614–7620.

Uchiyama, Y. (2004), 'Wetting and Drying Scheme for POM and its Applications to San Francisco Bay', in *Hydrodynamics VI: Theory and Applications: Proceedings of the 6th International Conference on Hydrodynamics*, eds. L. Cheng, and K. Yeow, Perth, Western Australia, pp. 293–299.

Vreugdenhil, C. (1994), *Numerical Methods for Shallow Water Flow*, Dordrecht: Kluwer Academic Publishers.

Whitehouse, R. (2000), *Dynamics of Estuarine Muds*, London: Thomas Telford.

Wu, Q., Rafiee, M., Tinka, A., and Bayen, A.M. (2009), 'Inverse Modeling for Open Boundary Conditions in Channel Network', in *Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, P.R. China, pp. 8258–8265.

Zienkiewicz, O.C., and Phillips, D.V. (1971), 'An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by Isoparametric Co-ordinates', *International Journal for Numerical Methods in Engineering*, 3, 519–528.