Floating Sensor Networks for River Studies

Andrew Tinka, *Student Member, IEEE,* Mohammad Rafiee, and Alexandre M. Bayen, *Member, IEEE*

Abstract—Free-floating sensor packages that take local measurements and track flows in water systems, known as drifters, are a standard tool in oceanography, but are new to estuarial and riverine studies. A system based on drifters for making estimates on a hydrodynamic system requires the drifters themselves, a communication network, and a method for integrating the gathered data into an estimate of the state of the hydrodynamics. This paper presents a complete drifter system and documents a pilot experiment in a controlled channel. The utility of the system for making measurements in unknown environments is highlighted by a combined parameter estimation and data assimilation algorithm using an extended Kalman filter. The performance of the system is illustrated with field data collected at the Hydraulic Engineering Research Unit, Stillwater, OK.

Index Terms—Data assimilation, hydrodynamics, Kalman filters, sensor systems and applications.

I. INTRODUCTION

A. Freshwater Systems

The majority of the renewable freshwater available for human use flows through rivers [1]. Human freshwater demand will increase significantly in the next 50 years, due mainly to population increase, urbanization, and increased use of water-intensive agriculture [2]. Modeling and monitoring the flow of freshwater, and the mixing and transport of constituents such as salt, can lead to improvements in water use efficiency and can help balance supply and demand [3]. Specific examples of environmental management scenarios requiring understanding of complex hydrodynamic systems include predicting the movement of silt disturbed during dredging and underwater construction operations, planning reservoir release and gate control policies to affect the intrusion of salt water based on specific local needs, and assessing vulnerabilities to contaminant spills or other unforeseen events in critical water resource regions. In each of these examples, high-quality hydrodynamic models, based on data gathered from the actual system, can be crucial for responsible environmental policy and decision-making.

B. Environmental and Mobile Sensing

The physical properties of large water systems can be measured using several different sensor types and modalities. Sensors are often 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 canonical Lagrangian sensor is a small floating package that transmits its location, and possibly other sensor measurements, as it is carried by the water current through the system. The oceanographic community calls such sensors *drifters*. While most infrastructural sensing in rivers and estuaries is implemented using Eulerian sensors, the evolution of wireless sensor network technology has increased the interest in novel Lagrangian sensor systems. The relative benefits of Lagrangian sensors compared to Eulerian sensors can be classified into two categories: *logistical* benefits and *information* benefits.

The logistical benefits of a Lagrangian sensor system derive from its flexibility and redeployable nature; in other words, intrinsic benefits of self-contained devices designed for autonomous operation. A fleet of drifters can be deployed, recovered, and redeployed in response to changing needs or new information. Their wireless communication allows them to be used in remote locations where power and communication infrastructure may not be available. These advantages are not inherent to the Lagrangian or Eulerian distinction; it would be possible to build an Eulerian sensor that was battery-powered, communicated using wireless networks, and could be easily redeployed. Rather, these logistical advantages are between Lagrangian systems as they *must* be implemented compared to Eulerian sensing as it is practised today.

The information benefits of mobile sensing, however, are unique to the Lagrangian or Eulerian split. By following the flow of water, Lagrangian sensors determine the *particle outcomes* of water in the system. An Eulerian sensor, observing the water as it flows past, can (normally) not infer anything about the water's *history*: where it came from, or where it will end up. Tracking movement of water is particularly important for studying the movement of contaminants or other constituents, especially in regions with complex topology, such as an estuary or a delta. Constituent transport is governed by processes including advection and diffusion [4]; the Lagrangian framework helps disambiguate the two, and allows investigation into the precise location of interfaces or rapid changes in concentration. One example of a hydrodynamic phenomenon of interest where Lagrangian drifters are relevant is *tidal trapping*, in which phase lags in tidal flow cause "dead

zones" where constituents can be trapped and released after a delay [5], [6].

Lagrangian sensors do have some disadvantages in river environments. Not all locations are suitable for deploying drifting sensors. Rapids and waterfalls have the potential to damage these devices. Rivers can contain obstacles that can capture drifters. The drifters must be retrieved at the end of a deployment, which can be a difficult procedure if they are scattered over a wide area (or snagged on different obstacles over a long stretch of river). The suitability of an environment for drifter studies must be assessed prior to drifter deployment.

C. Data Assimilation

River hydraulics can be modeled with shallow water equations in one or two dimensions [7]. Shallow water equations are a standard constitutive model used in the environmental engineering community and the hydraulics community to model river flow; they are commonly used for simulation and control. When dealing with experimental measurements, algorithms are required to incorporate them into a 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 [8].

Most data assimilation methods can be placed into the historically named categories of *variational* or *sequential* assimilation methods [9]. Variational assimilation methods perform a single optimization step on all the observed data to minimize a cost functional. By contrast, sequential assimilation methods, such as the Kalman filter and its extensions, perform a series of update and analysis steps, blending the observed data into the state estimate one step at a time. Several extensions of the Kalman filter are applicable to nonlinear systems. Examples include: the extended Kalman filter [10], which uses the Jacobian of the state update equation to update the estimate of the mean and covariance of the state, the ensemble Kalman filter [11], which tracks the evolution of a number of random samples in order to update the various estimates, and the unscented Kalman filter [12], which also tracks an ensemble of samples, but generates those samples using a deterministic technique in order to accurately track the mean and covariance with a minimal sample set.

This paper presents a data assimilation method based on the extended Kalman filter. Sequential assimilation methods are well suited to real-time assimilation, which is one of the future goals for this system. The extended Kalman filter is appropriate for nonlinear systems where the Jacobian is easy to compute, which will be seen in Section III.

D. Drifters in Oceanography and Hydrology

Although studies of *flotsam drift* (drawing inferences about currents from the observed movement of accidentally dropped material) can be found in antiquity, the first deliberate drifter study seems to be the work of G. Aimé circa 1845 [13]. His first drifters were *drift bottles*: sealed bottles containing a message asking the eventual recipient to report the date and location found. Drift bottle studies became a widely used technique in European oceanography around the beginning of the 20th century [14].

The first drifter that could actively communicate its position back to researchers was the "swallow float," invented by J. Swallow in 1955 [15]. It was a neutrally buoyant float that would drift approximately 1000 m underwater while transmitting acoustic pulses that would be received by researchers' hydrophones. Development of drifters with acoustic communication capabilities continued in the 1960s and 1970s [16]. In 1978, the introduction of the Argos satellite service [17] gave oceanographic researchers a global location and data uplink system, which lead to the development of oceanographic drifters that could communicate their position and sensor data during the mission. Examples of oceanographic drifters that leverage the Argos system include the Davis, i.e., coastal dynamics experiment drifter [18], the Ministar, i.e., world ocean climate experiment drifter [19], and the low cost tropical drifter [20], each developed in the mid-1980s.

Recent work in sensor networks for aquatic sensing missions included the AMOUR Project at the Massachusetts Institute of Technology (MIT), Cambridge [21], the NEPTUS framework of AUVs at the Laboratório de Sistemas e Tecnologia Subaquática, Porto, Portugal [22], submersible pneumatic drogues built at the University of California, San Diego [23], the Slocum underwater drifters at MBARI [24], and the SmartBay Sensor Network Project, Galway Bay, Ireland [25].

Although river and estuarine locations pose unique challenges, using drifting sensors in these environments is an emerging line of research. Other efforts include a low-cost floating GPS sensor [26] and a drogue carrying an acoustic profiler [27]. Deployment scenarios for drifting sensors usually involve deploying them at a specific location, allowing them to propagate through the environment with the water currents, and retrieving them at the end of the mission. The retrieval operation is usually assisted by the device transmitting its location to the research team.

The Floating Sensor Network (FSN) Project at the University of California, Berkeley (UC Berkeley) [28] designs and builds drifters for riverine and estuarine environments. An earlier generation with less developed capabilities was described in [29]. This paper gives a full system-level description of the second generation system; the improvements in this version include two reliable communication systems and new water-quality sensing capabilities. The system described herein is the first system built by the FSN Project that is capable of practical, unsupervised field deployments.

E. Purpose and Organization of This Paper

This paper will describe the design and implementation of a system for gathering data from multiple drifting sensors in a riverine or estuarine environment and assimilating it into a model that can be used to estimate the state of the natural environment. The design problem covers multiple domains, including the mechanical design of the drifting sensor, the selection and systems integration of the functional electronic components, an extensible and research-capable embedded computation capability on the individual sensors, the communication architecture for gathering the data from the field, and the software for the data assimilation problem on the back

end. The interdependencies between these domains make this a system design problem.

As outlined in Section I-D, drifter-based sensing has precedents in oceanography and other environments, but is a novel approach in the estuarine environment. The purpose of this paper is to describe the design problem and the approach taken for this implementation, and to demonstrate that a drifter-based system in riverine and estuarine environments is feasible and useful, based on a pilot study performed in a controlled environment.

The remainder of this paper is organized as follows. Section II describes the design and implementation of the second generation FSN, including the sensor hardware, sensor firmware, communication architecture, and central server software. Section III presents a deployment performed in November 2009 at the USDA-ARS Hydraulic Engineering Research Unit (HERU), Stillwater, OK. The data assimilation techniques used to integrate the gathered data into a model of the system, and the validation used to evaluate the system performance, are presented. Finally, Section IV presents conclusions and plans for future improvements to the FSN system.

II. System Description

A. Hardware (Sensor)

The overall form of the drifting sensor is a waterproof floating hull that encloses an electronics package.

The design considerations for the hull form include the following.

1) Present a symmetric, nonrectifying drag to planar currents.
2) Expose modular sensor packages to the water while providing mechanical protection.
3) Minimize wind drag exposure while placing multiple antennas above water surface.
4) Allow easy opening and closing while still providing waterproofing.
5) Survive rough handling, collisions, being thrown from shore, bridges, boats.
6) Use materials and construction techniques to permit assembly in small numbers in a university laboratory at low cost.

The simplest way to satisfy 1) is to be as symmetric as possible about a vertical axis. A cylinder is a good choice because a commercially manufactured pipe can be used for the body, satisfying 6). A vertical cylinder can also easily satisfy goals 2) and 3).

The hull is manufactured at UC Berkeley using low-cost, small-run manufacturing techniques. The drifter has a vertical cylinder configuration in order to present a uniform profile to surface currents while also supporting the antennas a small distance above the waterline. The hull consists of four major components, shown in Fig. 1: (a) a hand-cast fiberglass lower hull; (b) machined aluminum parts for the watertight seal; (c) a commercially available fiberglass pipe for the upper hull; and (d) a vacuum-formed polycarbonate top cap. The lower hull is flooded so that water quality sensors mounted in the bulkhead may contact the water but also be mechanically protected.
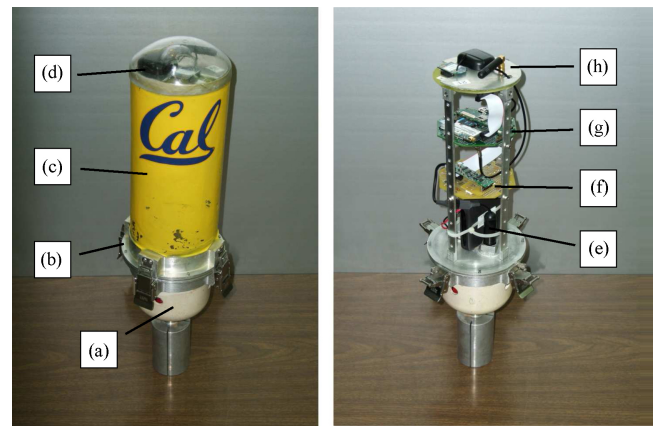

Fig. 1. Overview of the drifter hull. Left: closed. Right: open. (a), (c) Fiberglass components. (b) Aluminum components. (d) Polycarbonate top cap. (e) Battery and water quality sensor. (f) Sensor interface board. (g) Main electronics. (h) Antennas.
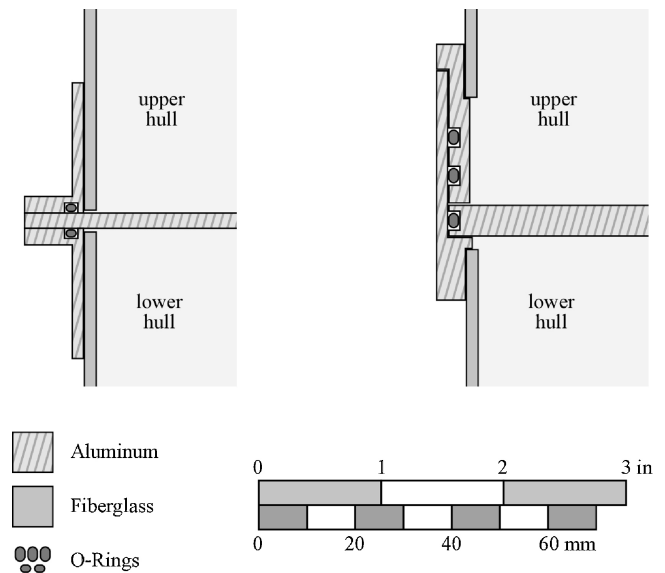

Fig. 2. Cross-sections of methods to seal two fiberglass cylinders at an aluminum bulkhead. Left: radial. Right: axial.

The waterproof seal is made using machined aluminum and rubber O-ring seals. Fig. 2 shows two sealing schemes that were tested with prototypes. The axial sealing scheme is generally preferred for two reasons: the O-rings are less likely to be damaged during the opening or closing operations, and redundant protection can be added by adding more O-rings in the major seal without major mechanical changes. However, the radial sealing scheme is more robust to damage and does not require tight tolerances in the aluminum parts. The prototype cylindrical sealing systems were prone to failure when the outer cylinder was knocked out of round through rough handling. The radial sealing system was the final selection.

Fig. 3 illustrates the position and mass of the major components in the drifter. When floating at the desired waterline, the drifter displaces 2.8 L of water. Before the battery, the mass of the drifter is 1.98 kg. This determines the mass of the ballast, which must be located as low as possible in order for the center of mass to be below the center of buoyancy
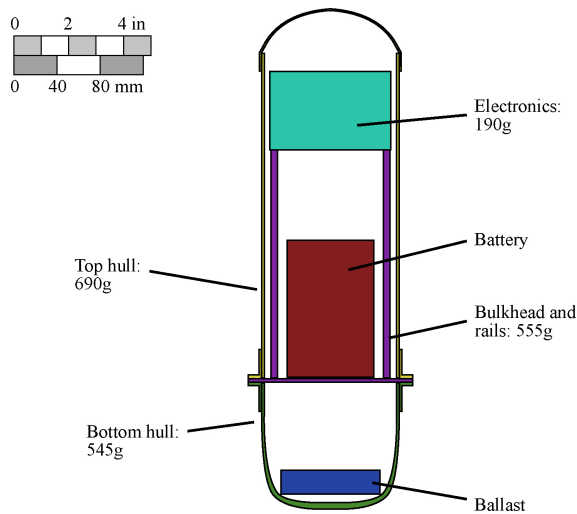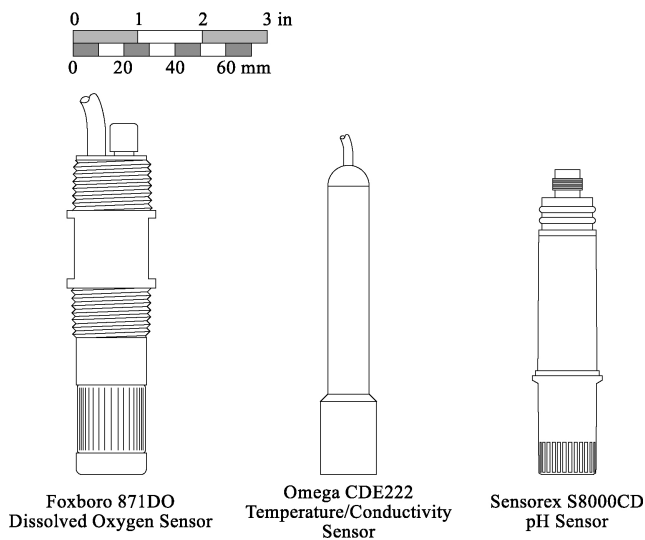
Fig. 3. Schematic of major mass components.

Fig. 4. Three examples of water quality sensors with the desired form factor. Foxboro 871DO [30], Omega CDH222 [31], Sensorex S8000CD [32].

Fig. 5. Module-level block diagram of drifter electronics.

(a necessary condition for the drifter to float in the desired vertical configuration). Two components make up the ballast: a lithium ion battery at the bottom of the dry hull, and an encapsulated cast lead weight at the bottom of the wet hull. The standard configuration uses a 200 g battery and a 600 g lead weight. The battery and water quality sensor are labeled (e) in Fig. 1.

One of the primary functions of the device is to carry water quality sensors. There are a wide variety of sensor modalities that are of interest to researchers; cost and mass limitations preclude carrying all possible sensors at once. The solution is to adopt a modular design that allows different sensors to be loaded into the body of the drifter.

Many commercially available sensors are available in form factors designed either for installation into process pipes or handhold laboratory use. Examples can be found in Fig. 4. In all three cases, the sensors can be roughly described as cylinders of diameter less than 25 mm and length of approximately 120 mm. This representative sensor size was used for the modular design. Two holes, 25.4 mm (1 in) diameter, with
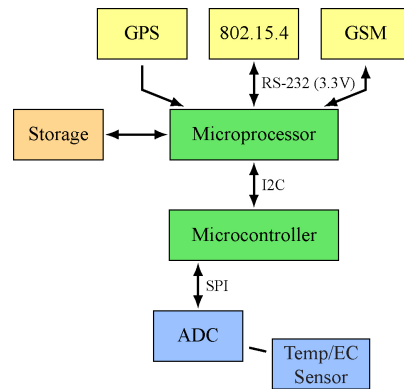
an NPT thread, are available in the main bulkhead. These holes can either be sealed with a plug or hold a bushing in which a desired water quality sensor is embedded. The contact side of the sensor is exposed to water in the flooded lower hull while still being protected from impact and debris by the lower fiberglass hull; the electronics side of the sensor is in the dry upper hull. The Omega CDH222 temperature or electroconductivity sensor was selected as the first sensor to be carried on the drifter.

The electronics are mounted near the top of the cylinder. See Fig. 5 for a block diagram of the major modules. The architecture was inspired by the Gumstix and Robostix products sold by Gumstix, Inc., Portola Valley, CA, in which low-level tasks are handled by a subordinate microcontroller while nonreal-time, higher-level tasks such as communications are governed by an embedded computer.

Communication between modules uses a variety of serial busses: standard RS-232 serial communications, an inter integrated circuit (I2C) bus, and a serial peripheral interface (SPI) bus. The GPS receiver, GSM module, and embedded computer are on the main electronics printed circuit board, labeled (g) in Fig. 1. Antennas for the GPS and GSM modules, and a short-range 802.15.4 2.4 GHz radio, are located at the top of the hull (h). The GPS, 802.15.4, and GSM modules communicate with the embedded computer with individual RS-232 ports. A subordinate microcontroller for real-time tasks such as sensor management is located on a lower board (f). The microcontroller and embedded computer communicate with each other using the I2C bus. The temperature and electroconductivity sensors are interfaced to an analog-to-digital converter, which delivers digitized sensor readings to the microcontroller via the SPI bus. In every case except the I2C bus, the choice of communication protocols was determined by the modules.

The GPS receiver is the Magellan AC-12 OEM module. When uncorrected (not using differential correction, SBAS, or post-processing), its circular error probable range is 1.5 m. In addition to GPS coordinates, it also provides pseudorange and carrier phase data, which can be used for post-processing or differential GPS techniques [33].

Long-range communication with the server is performed using the Motorola G24 GSM module. In areas with GSM coverage, the General Packet Radio Service (GPRS) can be used to send TCP or UDP packets to servers on the Internet.

Data rates depend on the GSM base station configuration, but are at least 8.0 kbit/s upload and download [34].

Short-range communication between drifters, and between drifters and field personnel, is performed with the Digi XBee-PRO ZB module. Using the IEEE 802.15.4-2006 protocol [35], these devices can form ad hoc mesh networks. The PRO module can transmit with 50 mW (17 dBm) of power [36]; connectivity at distances of up to 1 km in river environments has been observed when using these modules.

The embedded computer is a Gumstix Verdex Pro XM4, a 20 mm × 80 mm single-board computer with a Marvell PXA270 400 MHz processor and 64 MB of RAM. The PXA270 is an applications processor designed around the ARMv5 architecture [37].

One relevant characteristic for designers of embedded sensor systems is that the PXA270 does not have the hardware floating point capability, which may make it difficult to efficiently implement intensive signal processing or other computations. The Verdex is developed to run an OpenEmbedded Linux distribution. For real-time tasks, such as collecting data from sensors, an Atmel ATmega128L microcontroller [38] is used. Following the Robostix architecture, an I2C link carries data between the embedded computer and the microcontroller. In retrospect, this was a poor choice for the drifter implementation, due to the large distance between the upper and lower boards. An RS-232 link would have been a better choice.

The design of any field-deployed sensor system must take power consumption into account. The size of the battery and the power consumption of the electronics set the maximum mission time. For this reason, the computational capacity of the device is in tension with the overall mission time of the system. The design of this system prioritized research flexibility and capabilities over long mission times; accordingly, a highly capable embedded computer was chosen, even though the basic functionality of a Lagrangian sensor could be accomplished with a much simpler, low-power microcontroller. The estimated mission lifetime under moderate power discipline is 48 h, which is sufficient for most estuarial environments. Section IV-C describes some of the advanced capabilities that might one day be possible with Lagrangian sensors with powerful onboard computation, which will hopefully be explored in successive studies.

B. Software (Sensor)

The Verdex embedded computer has vastly more computational power than is needed for a record-and-transmit mission. The long-term vision of the FSN project is for a fleet of sensors that not only measure the physical phenomenon but also perform real-time distributed computations with the data; the computational requirements of the device were laid out with this goal in mind. The major design criterion for the software architecture was reliability. The development cycle precluded using formal methodologies for verifying software correctness, and so the development team proceeded under the assumption that the software they developed would always contain uncorrected bugs. Their approach was to exploit the Linux operating system features for as many architectural features as possible.
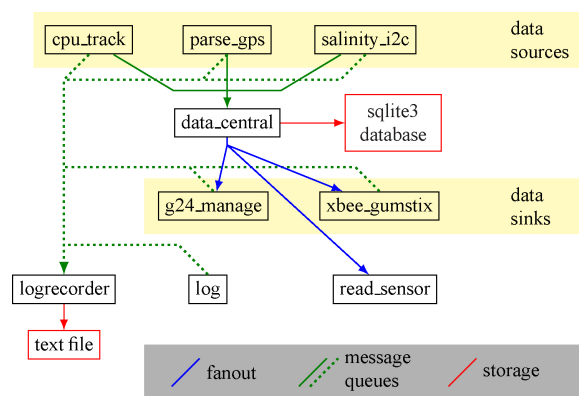

Fig. 6. Software modules. Data are produced by the sources, coordinated by the data_ central process, and transmitted by the data sinks. Other processes provide debugging and maintenance functions, as well as local storage.

The software was decomposed by function into software modules, as shown in Fig. 6. Each module was implemented as a separate process, written in C. Each process was designed to run indefinitely; each one was launched from a shell script that would restart it in the event that it did terminate (which would represent a bug). Many-to-one interprocess communication (IPC) (represented by green links in Fig. 6) was implemented using *message queues*, a seldom-used System V feature [39]. One-to-many IPC (represented by blue links in Fig. 6) was implemented using fanout first in, first outs (FIFOs), a custom kernel module [40]. Many modern embedded software projects in Linux environments use a Unix socket or TCP sockets for IPC [41]; message queues and fanout FIFOs, however, have one very attractive feature: messages are not lost if the receiving process crashes. In the event that the receiving process has terminated, messages are buffered (up to an OS-defined limit); if a new receiving process starts and connects to the message queue or fanout FIFO, it can recover all the queued messages that accumulated during the downtime.

Data are stored on a 1 GB MicroSD card installed on the Verdex. Data are stored in two formats: a text-based log that records events and debugging information, and an sqlite3 database [42] for random access during missions. Section II-C describes how the sensor data are transmitted in real time to back-end servers; the local storage in this case is merely a backup.

The following is a list of software modules or processes described in Fig. 6.

1) *logrecorder*: It receives generic event messages from other modules and records them on the flash storage.
2) *parse_gps*: It receives raw National Marine Electronics Association feed from the AC-12 GPS module, extracts useful information, performs basic conversions, and provides location information.
3) *data_central*: It receives data from any module that generates it, combines all data into a single stream, and exports it to the database and to any other module that uses information. If a publish or subscribe system [43] were used instead, this module would be the *dispatcher*.
4) *salinity_i2c*: It communicates with the Atmega microcontroller over the I2C bus, receives raw temperature and conductivity data from the water quality

sensor, processes it into salinity data, and exports it.

5) *xbee_gumstix*: It manages the XBee radio, periodically transmits important state information (see Section II-C), verifies incoming traffic, and pipes it to a shell.

6) *g24_manage*: It manages the G24 GSM module, connects and disconnects the GSM module from the network according to a mission-specific power management policy, opens TCP connections to a central server, and sends state information periodically.

7) *cpu_track*: It periodically gathers information on CPU state (utilization, free memory, and so on) and exports it.

8) *log*: It is a utility program to insert text manually into the log stream and is useful for recording events from scripts (such as program crashing and being restarted).

9) *read_sensor*: It is a utility program to print out all current sensor information for debugging purposes.

C. Communication Architecture

Sharing information between nodes in a network involves a process called *serialization*: converting structured data into a stream of bytes suitable for transmission over a digital communication link in a way that is efficient and can be unambiguously *deserialized* on the other end. In general, there are three approaches to serialization:

1) a custom, byte-compact binary format;
2) a generic framework such as XML [44], abstract syntax notation (ASN) [45], or Google protocol buffers [46];
3) a custom, textual, human-readable format.

These methods vary in data efficiency, ease of development, ease of debugging, and complexity of supporting software. For example, a custom byte-compact format will be very efficient in terms of the length of messages needed to represent data, but can be very difficult to debug. Generic frameworks with binary output, such as ASN, provide efficient messages, and the reliability of standardized packaged software, but may be too heavyweight for some applications. Given the relative simplicity of the data structures shared in this project, the developers decided upon the third option, a custom human-readable textual format. Messages were serialized in the following format

```
field_name/value/field_name/value
```

with newline terminating a message. The possible fields are:

1) **id**: drifter ID number;
2) **ts**: time stamp, seconds since Unix epoch;
3) **x_cm**: UTM *x* (easting) coordinate, cm;
4) **y_cm**: UTM *y* (northing) coordinate, cm;
5) **zn**: UTM zone;
6) **vel_x_cm**: velocity, *x* component, cm/s;
7) **vel_y_cm**: velocity, *y* component, cm/s;
8) **sats**: number of GPS satellites locked;
9) **sal**: salinity recorded;
10) **temp**: temperature recorded;
11) **cpu_1, cpu_5, cpu_15**: CPU utilization over a 1, 5, 15 min window;
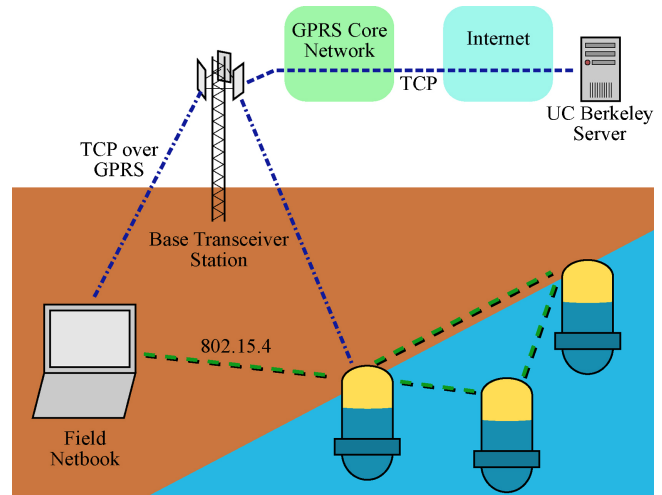12) **mem_free**: free memory, kB.

Fig. 7. Communication architecture. Drifters communicate amongst themselves and with field teams via 802.15.4; both the field laptop and the drifters can also communicate with the back-end servers using GPRS.

Any communication channel should make provision for *validation* of the data being transmitted. In this case, the underlying protocols (TCP/IP and the ZigBee protocol) have data integrity features. It was therefore unnecessary to include such features into the serialization protocol.

There are three ways in which data collected in the field make it back to the back-end servers for processing.

1) The drifter uses the GSM module to open a TCP connection during the mission directly to the server and delivers data.
2) A field team monitors the drifter via the ZigBee network during the mission and uploads the collected data to the server.
3) At the end of the experiment, the drifters are collected, and their logs and databases retrieved from their onboard MicroSD cards and uploaded.

Methods 1) and 2) are illustrated in Fig. 7. Method 3) will not be further discussed in this paper. The project goal is to build a system that delivers and processes data in real time; the end-of-experiment uploading is useful for verification and early research, but is not the project's direction.

A field team communicates with deployed drifters using a directional 2.4 GHz antenna connected to an XBee module. The field laptop runs a program called xbee_netbook, very similar to the xbee_gumstix running on the drifters. This program listens for the broadcasts of drifters, adds the messages to a local version of the database, and synchronizes the database with the home server via a GPRS/GSM connection.

Although the study described in Section III was performed over distances where ZigBee communications were easy, most studies in open river environments will often not permit short-range wireless communications. Both the distance between the base station and the drifting sensors, and the attenuation due to intervening obstacles such as vegetation or land masses, mean that the low powered ZigBee radios may not succeed at sending data. ZigBee radios are capable of mesh networking, meaning that a distant drifter could communicate with a base station through a multi-hop path of other drifters, but the same

factors that make single-hop communications difficult can also break connectivity over multiple hops. For this reason, the drifter system includes both the short-range ZigBee radio and the GSM module for connectivity over longer distances. The elevation of the GSM infrastructural tower, and the higher transmitting power, allow for GSM connectivity over much longer distances. Although this method of communication is dependent on the infrastructure of the GSM provider, in practical terms the GSM channel is a much more reliable method for transmitting data from the drifter for assimilation as well as logistical (i.e., retrieval) purposes.

Both the field laptop and the home server maintain a MySQL database [47] with all collected data from the drifters. A single message from a drifter consists of some or all of its sensor information for a particular time instant.

D. Back-End Architecture

Once the data have been uploaded from the field units (drifters or field netbook) to the MySQL database on the UC Berkeley server, it can be used for real-time or post-processing data assimilation and state estimation. The assimilation software is usually implemented on a different machine than the MySQL server. Remote query mechanisms are used to fetch new data from the MySQL server to the assimilation software. For the prototype implementation, the assimilation software (described further in Section III-B) was written in MATLAB and executed after the experiment was completed (post-processing). As will be seen in Section III-B, the dominant computational bottleneck is the inversion of an $n \times n$ matrix, where n is the assimilation state space. Therefore, the computational time should scale with the third power of the number of drifters plus the number of discretization points. Currently, the typical runtime for a complete assimilation job is 150 s to assimilate 450 s worth of data from six drifters, using a laptop with a 2.0 GHz Intel T2500 processor and 2 GB of RAM. This $O(n^3)$ scaling can easily be handled on larger systems using computation clusters. This method is therefore feasible for real-time assimilation of appropriately sized systems and drifter quantities.

Offline assimilation, as performed in the current system, is a valuable tool for analyzing the hydrodynamics of a region and for identifying the phenomena that govern the local environmental behavior. Online assimilation, where data are processed as it becomes available to inform a real-time model, has additional applications including forecasting and real-time monitoring. Section IV-C discusses details of the FSN Project's plans for the future scaling of the assimilation back-end, including development of online assimilation capabilities.

III. Sample Deployment

A. Mission Description

In November 2009, an experiment was performed at the USDA-ARS HERU, Stillwater, OK (see Fig. 8 for an overview). The HERU facility, located adjacent to Lake Carl Blackwell, has a gravity-fed supply canal that can have a controlled flow of up to 4.25 m³/s (150 ft³/s). The supply
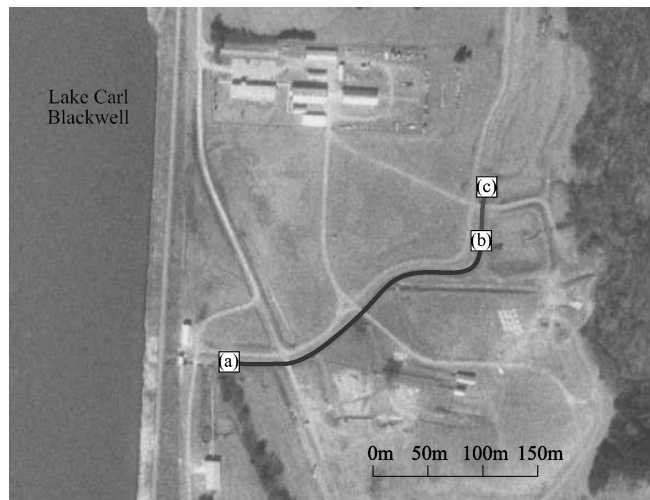

Fig. 8. HERU facility, with experimental channel annotated. Image courtesy of USGS. (a) Drifter release point. (b) Drifter recovery point. (c) Downstream gate.
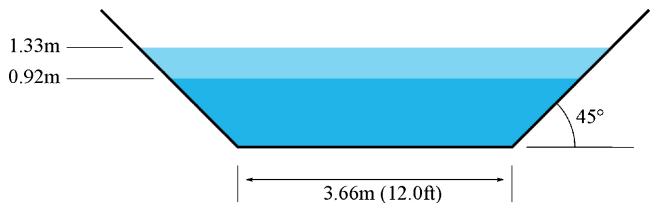

Fig. 9. Channel profile, including minimum and maximum water height.

canal feeds a number of experimental units that are normally used for investigations into levee reliability, reservoir safety, and spillway design [48]. For the experiment, drifters were deployed into the supply canal. The upstream boundary condition was the supply canal flow control, set to 1.42 m³/s (50 ft³/s); the downstream boundary condition was a gate that could be raised or lowered to restrict the flow out of the experimental region. Drifters were released at approximately 30 s intervals near the upstream boundary in Fig. 8(a). After traveling through the canal for approximately 400 s, they were individually retrieved in Fig. 8(b). Fig. 8(c) marks the location of the downstream control gate.

A total of 20 runs were performed, and divided into five cycles of four runs each. Each run in the cycle had a different operation of the downstream control gate. During the first run, the gate remained open for the entire run. During the second run, the gate was closed as soon as the sixth drifter was released. During the third run, the gate remained closed. Finally, during the fourth run the gate was opened as soon as the final drifter was released. The cycle was then repeated.

Fig. 9 shows the cross-section of the prismatic channel over most of its extent.

Because of the small experimental domain (and the low probability of losing a drifter), the GSM modules were not activated in this experiment. Instead, GPS position and velocity readings were stored on a 1 GB MicroSD card installed on the Verdex and simultaneously transmitted over the XBee radio to a nearby laptop, which uploaded them to the home server using a database synchronization protocol over a single GSM link.

Fig. 10. Two drifters in the HERU facility supply canal.

One disadvantage of this experiment was that there was no appreciable variation in water temperature, salinity, pH, or any other water quality factor. The only interesting state variables to measure and estimate were the velocity (equivalently, flow) and stage of the water. The Omega CDH222 salinity sensors were therefore not used in this experiment.

B. Assimilation Technique

The goal of the data assimilation is to incorporate the drifter position data into a model of the flow in order to estimate the state of the system, flow, and stage. The 1-D Saint–Venant equations are used as the model of the flow. After deriving a nonlinear state-space model from the Saint–Venant equations, the extended Kalman filter (EKF) is used to perform the estimation.

1) *Derivation of the State-Space Model:* The Saint–Venant model is among the most common models used for modeling the flow in open channels and irrigation systems [49], [50]. In the 1-D case, Saint–Venant equations are two coupled first-order hyperbolic partial differential equations derived from conservation of mass and momentum. For prismatic channels with no lateral inflow, these equations can be written as follows [51]:

$$T\frac{\partial H}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + \frac{\partial}{\partial x}(gh_cA) = gA(S_0 - S_f) \tag{2}$$

for $(x, t) \in (0, L) \times \Re^+$, where L is the river reach (m), $Q(x, t)$ is the discharge or flow (m³/s) across cross-section $A(x, t) = T(x)H(x, t)$, $H(x, t)$ is the stage or water depth (m), $T(x)$ is the free surface width (m), $D = A/T$ is the hydraulic depth m, $S_f(x, t)$ is the friction slope (m/m), S_b is the bed slope (m/m), g is the gravitational acceleration (m/s²), and h_c is the distance of the centroid of the cross-section from the free surface (m).

The friction slope is empirically modeled by the Manning–Strickler formula [52]

$$S_f = \frac{m^2 Q^2 P^{4/3}}{A^{10/3}} \tag{3}$$

with $Q(x, t) = V(x, t)A(x, t)$ the discharge across cross-section $A(x, t)$, P the wetted perimeter, i.e., the perimeter of the wetted portion of the cross-section, and m the Manning roughness coefficient ($sm^{-1/3}$).

The boundary conditions are usually taken to be the upstream flow $Q(0, t)$ and the downstream stage $H(L, t)$.

Applying the Lax diffusive scheme [51], [53] that is a first-order explicit scheme to discretize the equations to (1) and (2), the following set of finite difference equations are obtained:

$$A_i^{k+1} = \frac{1}{2}(A_{i-1}^k + A_{i+1}^k) - \frac{\Delta t}{2\Delta x}(Q_{i+1}^k Q_{i-1}^k) \tag{4}$$

$$Q_i^{k+1} = \frac{1}{2}(Q_{i-1}^k + Q_{i+1}^k)$$

$$-\frac{\Delta t}{2\Delta x}\left[\left(\frac{Q^2}{A} + gAh_c\right)_{i+1}^k - \left(\frac{Q^2}{A} + gAh_c\right)_{i-1}^k\right] \tag{5}$$

$$+\Delta t\left(\frac{\phi_{i+1}^k + \phi_{i-1}^k}{2}\right) \tag{6}$$

where

$$\phi = gA(S_b - S_f). \tag{7}$$

This scheme is stable provided that the Courant–Friedrich–Lewy condition holds, that is

$$\frac{\Delta t}{\Delta x}|V + C| \leq 1 \tag{8}$$

where $C = \sqrt{gD}$ is the wave celerity and V is the average velocity.

The equations above may only be used for interior grid points. At the boundaries, these equations cannot be applied since there is no grid point outside the domain. Therefore, another method needs to be used to compute the unknown variables at the boundaries. The method of specified time intervals is used to compute these variables [53]. In this method, after computing the characteristics, the boundary grid point is projected backward to the previous time step along its corresponding characteristic curve. After computing the variables at the projected point, which is usually done by using linear interpolation, the characteristic equations are used to compute the unknown variable at the boundary grid point at the next time step.

The discretized equations obtained above can be used to obtain a state-space model

$$x_{k+1} = f(x_k, u_k) \tag{9}$$

where x_k is the state vector at time k

$$x_k = (Q_2^k, \ldots, Q_N^k, H_1^k, \ldots, H_{N-1}^k)^T \tag{10}$$

and the input u_k contains the boundary conditions, i.e., the upstream flow and downstream stage

$$u_k = (Q_1^k, H_N^k)^T \tag{11}$$

where Q_i^k and H_i^k are the flow and stage at cell i at time $k\Delta t$, respectively, and N is the number of cells used for the discretization of the channel.

Assuming that all model parameters are known, when measurements of the flow other than the boundary conditions
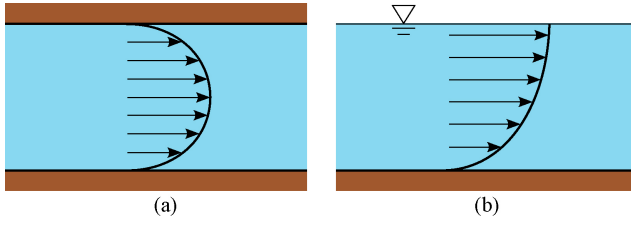
Fig. 11.   (a) Applied quartic function to the average velocity in the transverse. (b) Applied Von Karman log function to the average velocity in the vertical (right). Figures adapted from [55].

are available, these measurements can be incorporated into the state-space model using one of the standard nonlinear filters, e.g., the extended Kalman filter. However, in practice, it is sometimes impossible or expensive to obtain accurate values for one or more of these parameters. For instance, it is usually a difficult task to obtain an accurate value for the bed slope of a channel. As will be shown in Section III-C, the results of the model are very sensitive to the value of the bed slope. If parameters are critical but a direct measurement is not possible, proper experiments can be designed to obtain measurements of the system and these measurements may be used later to *identify* the unknown parameters. Nevertheless, it is sometimes not possible to carry out these types of experiments beforehand due to time constraints, lack of proper equipment, high costs, and so on.

In order to obtain estimates of the unknown parameters of the system, a vector of unknown parameters v_k is appended to the state vector; these parameters are considered to have zero dynamics ($v_{k+1} = v_k$). A nonlinear filter can then be applied to the augmented state-space model to simultaneously estimate the parameters and the actual state of the system. As the estimation of the state (that now includes the parameters as well) progresses, the estimates of the parameters get updated by incorporating the new measurements.

2) *Measurement Model:* The drifter velocity data provide local measurements of the flow velocity field. The relation between the drifter velocity and the flow at the corresponding cross-section relies on assumptions made about the profile of the water velocity. The surface velocity profile is assumed to be quartic, and the Von Karman logarithmic profile is assumed in the vertical direction [54] as shown in Fig. 11. For a given particle moving at a distance y from the center line and z from the surface, the particle's velocity $v_p(y, z)$ is related to the flow Q with the following equations:

$$v_p(y, z) = F_T(y) F_V(z) \frac{Q}{A} \quad (12)$$

with

$$F_T(y) = A_q + B_q \left(\frac{2y}{w} \right)^2 + C_q \left(\frac{2y}{w} \right)^4 \quad (13)$$

$$A_q + B_q + C_q = 0 \quad (14)$$

$$A_q + \frac{B_q}{3} + \frac{C_q}{5} = 1 \quad (15)$$

$$F_V(z) = 1 + \left(\frac{0.1}{\kappa} \right) \left(1 + \log \left(\frac{z}{d} \right) \right) \quad (16)$$

where w is the channel width, d is the water depth, and A_q, B_q and C_q are constants and $\kappa = 0.4$. A_q is commonly calculated experimentally and (14) and (15) are used to compute B_q and C_q. These equations are the constraint obtained from the facts that $F_T(y)$ is zero at the sides of the channel and the average $F_T(y)$ is equal to 1.

Denoting the collection of velocity measurements obtained from the drifters at time step k by y_k, the measurement model can be written as

$$y_k = g(x_k, k). \quad (17)$$

Note that the observation operator g is time-varying since the drifters are moving with the flow. Therefore, the cells at which the flow velocity is measured are changing over time.

3) *Stochastic State-Space Model and the Extended Kalman Filter:* The effect of modeling uncertainties, as well as inaccuracies in measurements of the inputs, is commonly considered as an additive noise term in the state equations (9) to obtain a stochastic equation

$$x_{k+1} = f(x_k, u_k, w_k). \quad (18)$$

The noise w_k is usually assumed to be zero-mean white Gaussian and

$$E[w_k w_l^T] = Q_k \delta_{kl}. \quad (19)$$

$x_0 \in \mathbb{R}^m$ is the initial state that is also assumed to be Gaussian and

$$x_0 = \mathcal{N}(\bar{x}_0, P_0) \quad (20)$$

where $\bar{x}_0$ and P_0 are the initial guesses for state and error covariance.

Note that in the case of combined state-parameter estimation, x_k is the augmented state, i.e., the concatenation of the actual state vector and the vector of unknown parameters.

Similarly, the errors and uncertainties in the measurements can be taken into account by adding a noise term to the measurement model (17) to obtain

$$y_k = g(x_k, e_k, k) \quad (21)$$

where e_k is the measurement noise of the sensors which is assumed to be zero-mean white Gaussian and

$$E[e_k e_l^T] = R_k \delta_{kl}. \quad (22)$$

The process and measurement noises and the initial conditions are all assumed to be independent.

In the EKF, the states of the system are approximated by a Gaussian random variable and are propagated through a linearized approximation of the state equations. The *prior* mean of the state is fed into the state equations to yield the prediction of the state. The *posterior* covariance matrices are calculated for a linear model that is obtained from linearizing the state equations around the current estimate [10].
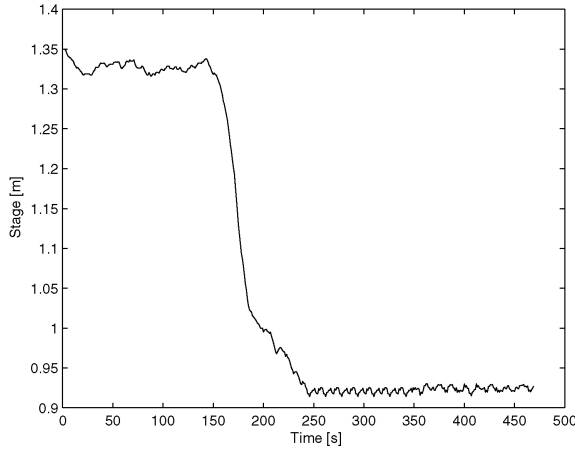
Fig. 12. Downstream stage (m).

Fig. 13. (a) Flow and (b) stage at the tenth cell for $S_b = 0.000$ (solid), $S_b = 0.001$ (dashed), $S_b = 0.002$ (dot-dashed).

With the stochastic state-space model given in the previous section and the following notations:

$$\hat{x}_{k|k-1} = \mathbb{E}[x_k | y_0, y_1, \ldots, y_{k-1}] \tag{23}$$

$$\hat{x}_{k|k} = \mathbb{E}[x_k | y_0, y_1, \ldots, y_k] \tag{24}$$

$$P_{k|k-1} = \mathbb{E}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T | y_0, y_1, \ldots, y_{k-1}] \tag{25}$$

$$P_{k|k} = \mathbb{E}[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T | y_0, y_1, \ldots, y_k]. \tag{26}$$

The iterations of the EKF can be summarized as follows:
Time update

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}, 0) \tag{27}$$

$$P_{k|k-1} = \Phi_{k-1} P_{k-1|k-1} \Phi_{k-1}^T + B_{k-1} Q_{k-1} B_{k-1}^T. \tag{28}$$

Measurement update

$$K_k = P_{k|k-1} G_k^T (G_k P_{k|k-1} G_k^T + D_k R_k D_k^T)^{-1} \tag{29}$$

$$\hat{y}_k = G_k \hat{x}_{k|k-1} \tag{30}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_k) \tag{31}$$

$$P_{k|k} = (I - K_k G_k) P_{k|k-1} \tag{32}$$

where

$$\Phi_{k-1} = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{k-1|k-1}, u_{k-1}} \qquad B_{k-1} = \left. \frac{\partial f}{\partial w} \right|_{\hat{x}_{k-1|k-1}, u_{k-1}}. \tag{33}$$

C. Numerical Results

This section presents the results of the implementation of the data assimilation method on the data collected from the experiment performed at the USDA-ARS Hydraulic Engineering Research Unit, Stillwater, OK, in November 2009. The measurements used for data assimilation are the positions and velocities of the drifters.

Fig. 12 shows the stage at the downstream end of the channel corresponding to a run used for evaluating the method. As can be seen in this figure, the downstream stage is initially 1.33 m and it starts to decrease as the downstream gate is opened until it becomes 0.92 m.

The discretization is done by dividing the channel into 60 cells, each of approximately 5 m length. The temporal step size is chosen as 1 s. Since data about the bottom elevation of the
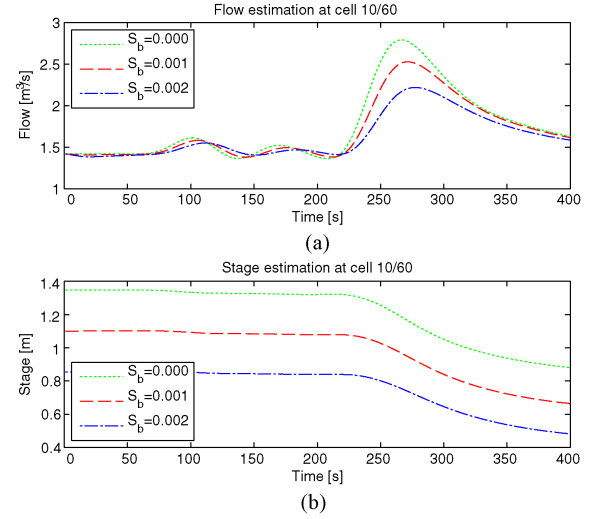
channel are unavailable, the bed slope of the channel cannot be calculated. In order to determine the sensitivity of the model with the given boundary conditions to the value of the bed slope, the forward simulation is run with three different values of bed slopes. In each case, the initial condition is chosen to be the backwater curve (steady state) that is computed using the following equations:

$$\frac{\partial Q}{\partial x} = 0 \tag{34}$$

$$\frac{\partial H}{\partial x} = \frac{gA(S_0 - S_f)}{-Q^2 \dfrac{T_b + 2H}{H^2(T_b + H)^2} + g(T_b H + H^2)} \tag{35}$$

where T_b is the bottom width.

Fig. 13 shows the flow and stage at the tenth cell, as a representative cell, for the three values of the bed slope. It is not surprising to see that the results of forward simulation vary significantly with different values of the bed slope.

To implement the data assimilation method, the measurements obtained from the five drifters are used. Next, the velocity of the sixth drifter is estimated using the estimated flow that is compared with its actual value obtained from the sixth drifter. Two methods are implemented: the extended Kalman filter with, and without, estimating the bed slope. Figs. 14 and 15 show the flow and stage at a few different cells predicted by the forward simulation (i.e., state-space model) assuming that the bed slope is zero, estimated flow and stage by performing the data assimilation method while the bed slope is assumed to be zero, and estimated flow and stage by performing the data assimilation method and estimating the bed slope as an unknown parameter. As can be seen in Fig. 12, the downstream stage starts to decrease at around time step 150 due to the gate opening. As can be seen in Fig. 14, the flow increases as a result of opening the gate. It can be seen in Fig. 15 that the stage reduction caused by opening the downstream gate propagates backward through the channel. However, in the case of assuming the bed slope as an unknown parameter, this reduction stage is
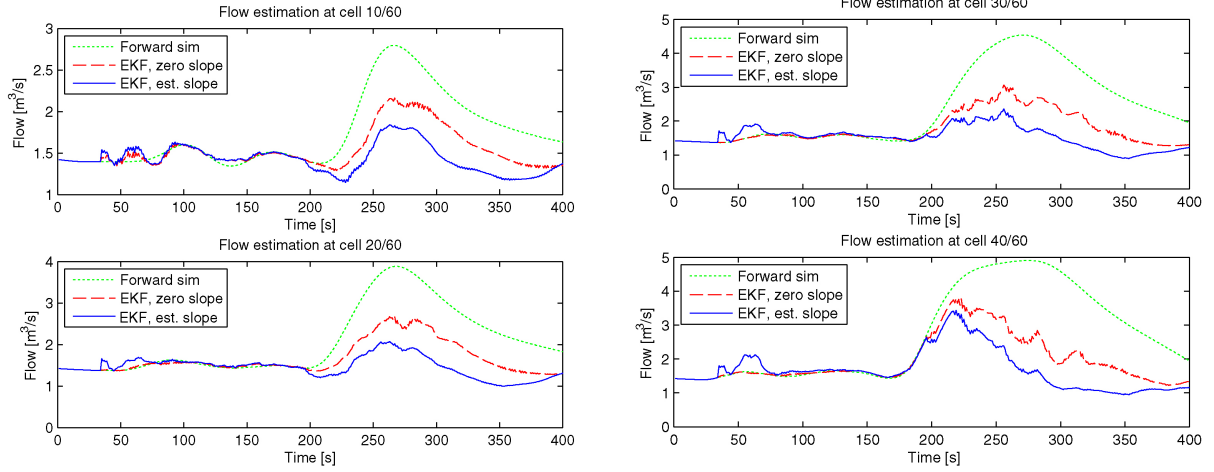
Fig. 14. Flow (m^3/s) at the 10th, 20th, 30th, 40th cells, forward simulation (dot-dashed), EKF with zero bed slope (dashed), and EKF with estimating bed slope (solid).
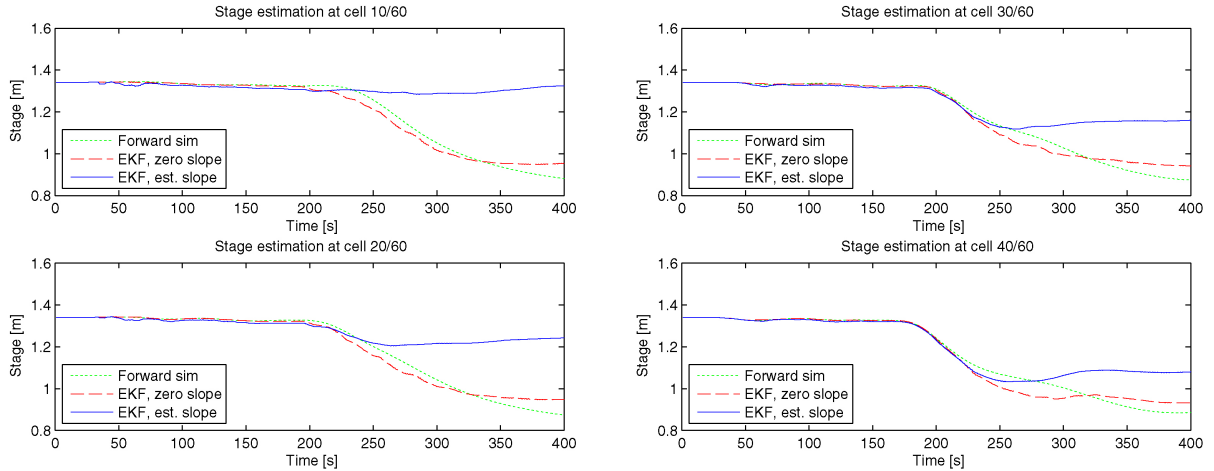

Fig. 15. Stage (m) at the 10th, 20th, 30th, 40th cells, forward simulation (dot-dashed), EKF with zero bed slope (dashed), EKF with estimating bed slope (solid).

more moderate. In particular, at cell 10, which is close to the upstream end of the channel, no decrease in the stage is seen. This is due to the fact that for a nonzero bed slope, the backwater curve (steady state) is not uniform. Since the initial estimate of the bed slope is taken to be equal to zero, the extended Kalman filter is initialized by a uniform steady state corresponding to a zero bed slope. However, as the estimated bed slope deviates from zero, the steady state of the system deviates from the uniform steady state accordingly. The time evolution of the estimated bed slope is illustrated in Fig. 16. While the values of flow and stage estimated by the data assimilation methods seem physically more reasonable, it is not possible to formally evaluate the performance of the method by looking at these figures. In order to obtain a more quantifiable assessment of the method, the velocity of the sixth drifter is calculated using the estimated flow at the corresponding cell. The same velocity profiles on the surface and along the depth as described in Section III-B2 are used to calculate the drifter velocity from the estimated flow. Fig. 17 shows the velocity of the sixth drifter predicted by the forward simulation, and both data assimilation methods and its actual

value. The peak in the measurement graph corresponds to when the sixth drifter is thrown into the water from the channel bank with an initial speed. As can be seen in this figure, the data assimilation methods significantly improve the estimation results. Also, it can be seen that considering the bed slope as an unknown parameter and using the measurements to estimate it improves the estimation results. In order to quantify the performance of the methods, we calculate the relative error of the estimated velocity of the sixth drifter at each time step using the following formula:

$$E(k) = \sqrt{\frac{(\hat{v}_k - v_k)^2}{(v_k)^2}} \times 100\% \qquad (36)$$

where v_k and $\hat{v}_k$ are the true and estimated values of the velocity of the sixth drifter at time step k.

The relative error is calculated for all cases, and Table I provides the average relative error per time step corresponding to each case.
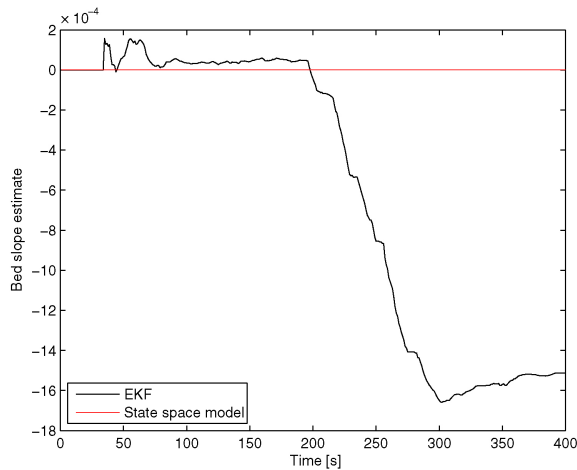
Fig. 16.   Estimated bed slope (black) compared to the assumed bed slope (dashed).
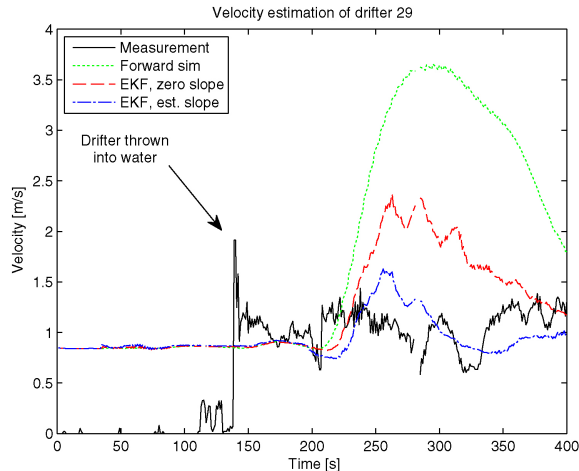

Fig. 17.   Velocity of the sixth drifter: forward simulation (dot-dashed), EKF with zero bed slope (dashed), EKF with estimating bed slope (dotted), and the actual drifter measurements (solid).

TABLE I

ERROR SUMMARY FOR CROSS-VALIDATION STUDY

Method	Relative Error
Forward simulation	139.2%
EKF with zero bed slope	53.5%
EKF with est. bed slope	22.9%

IV. CONCLUSION

A. *Feasibility of Real-Time Assimilation Using Drifters in Rivers*

The FSN Project has designed, implemented, and tested a complete system for gathering data from estuarial and riverine systems using independent passive floating sensor packages known as "drifters" and assimilating these data into a hydrodynamic model of the system being studied. In addition to the drifters, the system incorporates the communication infrastructure to transfer the information gathered by the drifters to the field team and to databases on the back-end server. The system also incorporates an extended Kalman filter-based assimilation algorithm for incorporating the data into a 1-D shallow water model. While the system as currently implemented performs the data assimilation after the deployment phase of the experiment is complete, the EKF algorithm is designed to process data incrementally, and is fast enough to permit a real-time assimilation on the data as it is being gathered.

These pilot studies show that the mobile sensor technique for studies in river channels is feasible and can yield useful information about the state of the system. Moreover, the use of a simultaneous parameter estimation and data assimilation technique demonstrates that this method is feasible for environments where the parameters of the system (channel geometry, bed slope, friction constants, and so on) have not yet been measured or estimated. This is a strong advantage when studying new hydrodynamic systems for the first time, and supports the known advantage of Lagrangian sensing techniques in terms of flexibility and ease of deployment in remote, unexplored, and unfamiliar regions.

B. *Limitations of Drifters as Sensors*

The major limitation of using a drifting sensor package to estimate the velocity of the water is the variation in the velocity of the water at different positions within the channel (the "profile" of the velocity). The 1-D shallow water equations model the cross-sectional average of the water velocity; the raw velocity observed by a drifter will be different from the average value. Determining the best possible estimate of the *average* velocity based on the drifter's *local* velocity is nontrivial. Some aspects of the profile are easy to work with; for example, it is known that the drifter is floating at the water surface, and logarithmic depth profiles are a reasonable assumption. Dealing with the lateral position of the drifter within the channel is more challenging.

C. *Future Work*

The future plans for the FSN Project include improvements to the sensor hardware as well as to the data assimilation back-end. The next generation of the hardware will feature a propulsion system that will allow some modifications of the drifter's trajectory. Naturally, a propelled drifter is no longer an ideal Lagrangian sensor, so the use of this feature will have to be carefully considered. On the software side, the next generation system will be capable of performing real-time assimilation; the data coming in from the sensors via GSM will be integrated within minutes to provide a live estimate of the state of the hydrodynamic system. The domains of future experiments will have more complicated topologies; networks of branching and merging channels will require more sophisticated models than the 1-D shallow water equation, and the assimilation techniques will need to be refined to match.

The problem of extracting the average water velocity from the local drifter measurements will be investigated in future work. Possible approaches include making profile parameters part of the estimation process, using the GPS position to estimate the lateral position of the drifter, and placing several drifters close together in the channel to provide a diversity of measurements.

The future version of the data assimilation system will include an online server that will assimilate the data in real time as it arrives; this system will be implemented on a large parallel computational cluster. This future system will assimilate the data into the river, estuary, and land model, a new 2-D model of the San Francisco Bay and Sacramento-San Joaquin Delta currently under development by the California Department of Water Resources and the Lawrence Berkeley National Laboratory [56]. An even more ambitious plan for future assimilation work is to use the onboard computation of the drifter sensors to perform a distributed data assimilation, as opposed to transmitting the sensor data to back-end servers for centralized assimilation.

REFERENCES

[1] T. Oki and S. Kanae, "Global hydrological cycles and world water resources," *Science*, vol. 313, no. 5790, pp. 1068–1072, 2006.

[2] S. L. Postel, "Entering an era of water scarcity: The challenges ahead," *Ecol. Applicat.*, vol. 10, no. 4, pp. 941–948, 2000.

[3] R. B. Jackson, S. R. Carpenter, C. N. Dahm, D. M. McKnight, R. J. Naiman, S. L. Postel, and S. W. Running, "Water in a changing world," *Ecol. Applicat.*, vol. 11, no. 4, pp. 1027–1045, 2001.

[4] K. R. Dyer, *Estuaries: A Physical Introduction*. New York: Wiley, 1973.

[5] R. Smith and C. F. Scott, "Mixing in the tidal environment," *J. Hydraulic Eng.*, vol. 123, no. 4, pp. 332–340, Apr. 1997.

[6] L. MacVean and M. Stacey, "Estuarine dispersion from tidal trapping: A new analytical framework," *Estuaries Coasts*, vol. 34, no. 1, pp. 45–59, Jan. 2011.

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

[8] F.-X. Le Dimet and O. Talagrand, "Variational algorithms for analysis and assimilation of meterological observations: Theoretical aspects," *Tellus*, vol. 38A, no. 2, pp. 97–110, Mar. 1986.

[9] K. Ide, P. Courtier, M. Ghil, and A. Lorenc, "Unified notation for data assimilation: Operational, sequential and variational," *J. Met. Soc. Japan*, vol. 75, no. 1B, pp. 71–79, 1997.

[10] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Upper Saddle River, NJ: Prentice-Hall, 1979.

[11] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*. Berlin, Germany: Springer, 2006.

[12] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. Int. Symp. Aerospace/Defense Sensing Simul. Controls*, vol. 3. 1997, pp. 182–193.

[13] J. N. Carruthers, "Some oceanography from the past," *J. Navigation*, vol. 16, no. 2, pp. 180–188, 1963.

[14] J. N. Carruthers, "Further investigation upon the water movements in the English channel: Drift-bottle experiments in the summers of 1927, 1928, and 1929, with critical notes on drift-bottle experiments in general," *J. Marine Biol. Assoc. United Kingdom*, vol. 17, no. 1, pp. 241–275, 1930.

[15] J. C. Swallow, "A neutral-buoyancy float for measuring deep currents," *Deep Sea Res. (1953)*, vol. 3, no. 1, pp. 74–81, 1955.

[16] W. J. Gould, "From swallow floats to argo: The development of neutrally buoyant floats," *Deep Sea Research Part II: Topical Stud. Oceanography*, vol. 52, nos. 3–4, pp. 529–543, 2005.

[17] D. D. Clark, "Overview of the Argos system," in *Proc. OCEANS*, vol. 3. Sep. 1989, pp. 934–939.

[18] R. E. Davis, "Drifter observations of coastal surface currents during CODE: The method and descriptive view," *J. Geophys. Res.*, vol. 90, no. C3, pp. 4741–4755, 1985.

[19] P. P. Niiler, R. E. Davis, and H. J. White, "Water-following characteristics of a mixed layer drifter," *Deep Sea Res. Part A. Oceanographic Res. Papers*, vol. 34, no. 11, pp. 1867–1881, 1987.

[20] D. S. Bitterman and D. V. Hansen, "The design of a low cost tropical drifter buoy," in *Proc. Marine Data Syst. Int. Symp. (MDS)*, 1986, pp. 575–581.

[21] C. Detweiller, I. Vasilescu, and D. Rus, "An underwater sensor network with dual communications, sensing, and mobility," in *Proc. OCEANS 2007 Eur.*, 2007, pp. 1–6.

[22] P. B. Sujit, J. Sousa, and F. L. Pereira, "UAV and AUVs coordination for ocean exploration," in *Proc. OCEANS 2009 Eur.*, 2009, pp. 1–7.

[23] Y. Han, R. A. de Callafon, J. Cortés, and J. Jaffe, "Dynamic modeling and pneumatic switching control of a submersible drogue," in *Proc. 7th Int. Conf. Informatics Control Autom. Robot.*, 2010, pp. 89–97.

[24] P. Bhatta, E. Fiorelli, F. Lekien, N. Leonard, D. Paley, F. Zhang, R. Bachmayer, R. Davis, D. Fratantoni, and R. Sepulchre, "Coordination of an underwater glider fleet for adaptive ocean sampling," in *Proc. Int. Workshop Underwater Robot.*, 2005 [Online]. Available: http://publications.iot.nrc.ca/documents/IR/IR-2005-44.pdf

[25] J. Ryan, N. Aonghusa, and E. Sweeney, "SmartBay, Ireland: Design and planning for a cabled ocean observatory off the west coast of Ireland," in *Proc. OCEANS*, 2008, pp. 1–4.

[26] J. Austin and S. Atkinson, "The design and testing of small, low-cost GPS-tracked surface drifters," *Estuaries*, vol. 27, no. 6, pp. 1026–1029, Dec. 2004.

[27] C. M. Schacht and C. J. Lemckert, "A new Lagrangian-acoustic Drogue (LAD) for monitoring flow dynamics in an estuary: A quantification of its water-tracking ability," *J. Coastal Res.*, vol. 50, pp. 420–426, 2007.

[28] J. Martinez. (2010). *Floating Sensor Network* [Online]. Available: http://float.berkeley.edu

[29] A. Tinka, I. Strub, Q. Wu, and A. M. Bayen, "Quadratic programming based data assimilation with passive drifting sensors for shallow water flows," *Int. J. Control*, vol. 83, no. 8, pp. 1686–1700, Aug. 2010.

[30] *871DO-C Dissolved Oxygen Sensors and Sensor Accessories: Product Specifications*, document PSS 6-9B1 A, Invensys Systems, Inc., Ashburn, VA, 2005.

[31] *CDH222 Conductivity/TDS Meter Manual*, 0212-YK-22CT, Omega Engineering, Inc., Stamford, CT, 2010.

[32] *S8000 Series Outline and Dimensions*, Sensorex Corporation, Garden Grove, CA, 2008.

[33] *A12, B12, & AC12 Reference Manual*, 630871 rev. D, Thales Navigation, Tulsa, OK, 2005.

[34] *Motorola G24 Developers' Guide: Module Hardware Description*, 6889192V27-G, Motorola, Schaumburg, IL, 2007.

[35] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std., 2006.

[36] *XBee/XBee-PRO ZB RF Modules*, 90000976_C, Digi International, Minnetonka, MN, 2009.

[37] *Intel PXA27x Family Design Guide*, 280001-002, Intel Corporation, Santa Clara, CA, May 2005.

[38] *ATmega128/ATmega128L Datasheet*, 2467R-AVR-06/08, Atmel Corporation, San Jose, CA, 2008.

[39] W. R. Stevens, *Advanced Programming in the UNIX Environment*. Reading, MA: Addison-Wesley, 1992.

[40] B. Smith, J. Hardin, G. Phillips, and B. Pierce, *Linux Appliance Design: A Hands-on Guide to Building Linux Appliances*. San Francisco, CA: No Starch Press, 2007.

[41] J. Gowdy, "A qualitative comparison of interprocess communications toolkits for robotics," Robotics Instit., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-00-16, Jun. 2000.

[42] M. Owens, *The Definitive Guide to SQLite*. New York: Apress, 2006.

[43] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003.

[44] C. F. Goldfarb and P. Prescod, *The XML Handbook*. Upper Saddle River, NJ: Prentice-Hall, 1998.

[45] *Data Networks and Open System Communications OSI Networking and Systems Aspects: Abstract Syntax Notation One (ASN.1): Information Technology, ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, International Telecommunication Union, Geneva, Switzerland, 1995.

[46] Google, Inc. (2010, Oct. 2). *Protocol Buffers Developers' Guide* [Online]. Available: http://code.google.com/apis/protocolbuffers/docs/overview.html

[47] M. Kofler, *The Definitive Guide to MySQL 5*. New York: Apress, 2005.

[48] S. L. Britton, G. J. Hanson, and D. M. Temple, "A historic look at the USDA-ARS hydraulic engineering research unit," in *Henry P. G. Darcy and Other Pioneers in Hydraulics: Contributions in Celebration of the 200th Birthday of Henry Philibert Gaspard Darcy*, G. O. Brown, J. D. Garbrecht, and W. H. Hager, Eds. Reston, VA: American Soc. Civil Engineers, 2003, pp. 263–276.

[49] V. Chow, *Open-Channel Hydraulics*. New York: McGraw-Hill, 1988.

[50] J. Cunge, F. Holly, and A. Verwey, *Practical Aspects of Computational River Hydraulics*. London, U.K.: Pitman, 1980.

[51] T. Strum, *Open Channel Hydraulics*. New York: McGraw-Hill, 2001.

[52] X. Litrico and V. Fromion, *Modeling and Control of Hydrosystems*. Berlin, Germany: Springer, 2009.

[53] M. Chaudhry, *Open-Channel Flow*, 2nd ed. Berlin, Germany: Springer, 2008.

[54] G. V. Bogle, "Stream velocity profiles and longitudinal dispersion," *J. Hydraulic Eng.*, vol. 123, no. 9, pp. 816–820, 1997.

[55] Office of State Water Project Planning, "Methodology for flow and salinity estimates in the Sacramento-San Joaquin Delta and Suisun Marsh," California Dept. Water Resources, Sacramento, CA, Tech. Rep. 21, 2000 [Online]. Available: http://baydeltaoffice.water.ca.gov/modeling/deltamodeling/annualreports.cfm

[56] E. Ateljevich, P. Colella, D. T. Graves, T. J. Ligocki, J. Percelay, P. O. Schwartz, and Q. Shu, "CFD modeling in the San Francisco Bay and Delta," in *Proc. 4th SIAM Conf. Math. Ind.*, 2009, pp. 99–107.

Mohammad Rafiee received the B.Sc. degree in mechanical engineering from the Sharif University of Technology, Tehran, Iran, in 2005, and the M.Sc. degree in mechanical engineering and the M.A. degree in mathematics from the University of California, Berkeley (UC Berkeley), in 2008 and 2011, respectively, and the Ph.D. degree from the Department of Mechanical Engineering, UC Berkeley, in May 2012.

His current research interests include machine learning, time series analysis, Monte Carlo methods and particle filters, data assimilation and inverse problems, distributed parameter systems, and convex optimization.

Andrew Tinka (M'04–S'06) received the B.A.Sc. degree in engineering physics from the University of British Columbia, Vancouver, BC, Canada, in 2002, and the M.S. degree in civil and environmental engineering (systems engineering) from the University of California, Berkeley, in 2008, where he is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical Engineering and Computer Sciences, focusing on the design and applications of the floating sensor network.

He has been with Powis Parker, Inc., Berkeley, CA, where he worked on embedded systems engineering, and with the Center for Collaborative Control of Unmanned Vehicles, University of California, Berkeley, where he worked on systems engineering. His current research interests include Lagrangian sensor design, multivehicle control and planning, and data assimilation.

Alexandre M. Bayen (S'02–M'04) received the Engineering degree from École Polytechnique, Palaiseau, France, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA.

He is currently an Associate Professor with the Department of Electrical Engineering and Computer Sciences and the Department of Civil and Environmental Engineering, University of California, Berkeley (UC Berkeley). He was a Visiting Researcher with the NASA Ames Research Center, Moffett Field, CA, from 2000 to 2003. He has been the Research Director of the Autonomous Navigation Laboratory, LRBA, Ministere de la Defense, Vernon, France, where he has held the rank of Major. He has authored one book and over 100 articles in peer-reviewed journals and conferences.

Dr. Bayen was the recipient of the Ballhaus Award from Stanford University in 2004, the CAREER Award from the National Science Foundation in 2009, and was a NASA Top 10 Innovator on Water Sustainability in 2010. His projects Mobile Century and Mobile Millennium received the Best of ITS Award for Best Innovative Practice at the ITS World Congress in 2008, and the TRANNY Award from the California Transportation Foundation in 2009. He was the recipient of the Presidential Early Career Award for Scientists and Engineers from the White House in 2010. Mobile Millennium has been featured more than 100 times in the media, including TV channels and radio stations (CBS, NBC, ABC, CNET, NPR, KGO, BBC), and in the popular press (Wall Street Journal, Washington Post, LA Times).