State estimation in large-scale open channel networks using sequential Monte Carlo methods: Optimal sampling importance resampling and implicit particle filters

Mohammad Rafiee,[1] Axel Barrau,[2] and Alexandre M. Bayen[3]

Received 10 November 2011; revised 24 October 2012; accepted 6 November 2012.

[1] This article investigates the performance of Monte Carlo-based estimation methods for estimation of flow state in large-scale open channel networks. After constructing a state space model of the flow based on the Saint-Venant equations, we implement the optimal *sampling importance resampling* filter to perform state estimation in a case in which measurements are available at every time step. Considering a case in which measurements become available intermittently, a random-map implementation of the *implicit particle filter* is applied to estimate the state trajectory in the interval between the measurements. Finally, some heuristics are proposed, which are shown to improve the estimation results and lower the computational cost. In the first heuristics, considering the case in which measurements are available at every time step, we apply the implicit particle filter over time intervals of a desired size while incorporating all the available measurements over the corresponding time interval. As a second heuristic method, we introduce a *maximum a posteriori* (MAP) method, which does not require sampling. It will be seen, through implementation, that the MAP method provides more accurate results in the case of our application while having a smaller computational cost. All estimation methods are tested on a network of 19 tidally forced subchannels and 1 reservoir, Clifton Court Forebay, in Sacramento-San Joaquin Delta in California, and numerical results are presented.

Citation: Rafiee, M., A. Barrau, and A. M. Bayen (2013), State estimation in large-scale open channel networks using sequential Monte Carlo methods: Optimal sampling importance resampling and implicit particle filters, *Water Resour. Res.*, *49*, doi:10.1029/2011WR011608.

1. Introduction

[2] Data assimilation is the process of integrating observations or measurements into a mathematical model of a physical system, to estimate some quantities of interest. Recently, data assimilation has provided rapid advances in geosciences such as meteorology, oceanography, and hydrology [*Brasseur and Nihoul*, 1994; *Haidvogel and Robinson*, 1989; *Malanotte-Rizzoli*, 1996; *Kalnay*, 2003; *Andreadis et al.*, 2007; *Biancamaria et al.*, 2010; *Durand et al.*, 2008; *Madsen and Skotner*, 2005; *Neal et al.*, 2009]. Different methods for assimilating data include variational data assimilation [*Navon*, 1997]; *Kaipio and Somersalo*, 2004; *Castaings et al.*, 2006; *Le Dimet and Talagrand*, 1986; *Wu et al.*, 2009], filtering-based methods [*Evensen*, 2009; *Kuznetsov et al.*, 2003; *Moradkhani et al.*, 2005; *Wang et al.*, 2009; *van Leeuwen*, 2010; *Weare*, 2009; *Tossavainen et al.*, 2008; *Neal et al.*, 2007], optimal statistical interpolation [*Molcard et al.*, 2003], or the Newtonian relaxation [*Ishikawa et al.*, 1996; *Paniconi et al.*, 1996].

[3] Open channels are examples of the so-called distributed parameter systems in which the dynamics of the system can be modeled by a set of *partial differential equations* (PDEs). For modeling the water flow in rivers and open channels, the Saint-Venant equations, which are a set of coupled first-order hyperbolic nonlinear PDEs, are commonly used [*Chow*, 1988; *Cunge et al.*, 1980; *Litrico and Fromion*, 2009]. Solving the PDEs requires an accurate knowledge of the boundary conditions, which are usually obtained from measurements of sensors installed at appropriate locations. Nevertheless, noise and inaccuracies in the measurements of the boundary conditions, as well as modeling assumptions (simplifications made to construct the mathematical model), can lead to a mismatch between the values computed by the model and the actual state of the system. When additional observations (measurements) of the system are available, it is desirable to incorporate these measurements into the model to reduce the mismatch between the values computed by the model and the actual system

[1] Department of Mechanical Engineering, University of California, Berkeley, California, USA.

[2] Département de Mathématiques Appliquées, Ecole Polytechnique, Palaiseau Cedex, France.

[3] Departments of Civil and Environmental Engineering and Electrical Engineering and Computer Science, University of California, Berkeley, California, USA.

Corresponding author: M. Rafiee, Department of Mechanical Engineering, University of California, Berkeley, CA 94710-1740, USA. (rafiee@berkeley.edu)

0043-1397/13/2011WR011608

throughout the whole domain of interest. Different state estimation methods can be used to estimate the state of the system with streaming data using the available observations obtained from the system. In distributed parameters systems, for which the system is typically high dimensional, it is important that state estimation methods with appropriate computational complexity are used so that real-time state estimation becomes tractable. An efficient real-time estimation method can be used in various applications such as flood monitoring, real-time flow studies, and emergency response in cases such as levee break or gate malfunctions.

[4] In the last decade, *sequential Monte Carlo methods*, also known as *particle filters*, have attracted a lot of attention among researchers and practitioners [*Gordon et al.*, 1993; *Liu and Chen*, 1998; *Arulampalam et al.*, 2002; *Doucet et al.*, 2000, 2001]. Particle filters and their variations have been extensively applied to estimation problems in geosciences, such as meteorology and hydrology, due to their generality and scalability. More specifically, particle filters can be applied to nonlinear systems, and they do not require Gaussianity assumptions on the noises. In *Giustarini et al.* [2011] and *Matgen et al.* [2010], particle filters have been used to assimilate water stage measurements of rivers obtained from the synthetic aperture radar (SAR) into hydraulic models. In *Moradkhani et al.* [2005], the authors use the particle filter to estimate parameters as well as the state in a hydrologic model. In particle filters, the posterior probability density function (pdf) is approximated by a number of particles with their corresponding weights. These particles are propagated forward, and their weights are updated at every time step. A larger number of particles result in more accurate results, while it increases the computational cost of the method. Nonetheless, particle filters have shown to encounter different problems when implemented on various systems. The most critical issue observed in implementations of particle filters is the degeneracy problem [*Doucet et al.*, 2000]. When degeneracy happens, almost all of the particle weights vanish after a number of iterations meaning that most of the samples get too far from the actual state of the system, and, consequently, they no longer contribute to approximating the posterior density function. Different methods have been developed to deal with the degeneracy problem among which *sampling importance resampling* (SIR) filter is a most commonly used approach [*Kitagawa*, 1996]. In the SIR filter, after each time step, the density function is resampled so that the samples with small weights are discarded and more probable samples are duplicated according to their weights.

[5] A more subtle issue that usually arises with particle filters is that, even with resampling, a lot of particles end up having small weights meaning that they correspond to low-probability regions. Hence, the number of particle contributing to the approximation of the posterior density function is usually smaller than the actual number of particles, and most of the computational effort is wasted on unlikely particles. To overcome this problem, *implicit particle filter* is introduced in *Chorin and Tu* [2009], *Chorin et al.* [2010], and *Morzfeld et al.* [2012]. Implicit particle filtering is a method to obtain high-probability samples from the density function. Implicit sampling requires solving an underdetermined equation for each particle at every time step when a measurement becomes available. While the cost of sampling can be higher in implicit filters, more accurate results may be obtained with a smaller number of particles as the particles belong to the high-probability region of the density function.

[6] In the current article, our goal is to evaluate the performance of different types of Monte Carlo methods for real-time state estimation of water flow in complex networks of open channels. In particular, we are interested to see how implicit particle filters, developed recently, perform in a practical application compared to particle filters in terms of both accuracy and computational complexity. After constructing a state space model for the network under consideration from the Saint-Venant equations, we apply a number of state estimation methods to incorporate some available measurements into the model in two situations: when measurements of the system are available at every time step and when the measurements become available intermittently. Given the observation model is linear and we assume the noises are Gaussian, we apply the optimal SIR filter to the system in which case the conditional density $p(x_k|x_{k-1}, z_k)$, which is chosen as the importance density, happens to be a Gaussian density function. We also consider a case in which measurements of the system become available intermittently; that is, we do not have measurements at every time steps, and we apply the implicit particle filter to incorporate these measurements to obtain estimates. We use the random-map method [*Morzfeld et al.*, 2012] to solve the implicit sampling equation. In this method, the samples are parametrized via a random map, and samples are obtained by solving an algebraic equation of one variable, for each sample, which is obtained from substituting the random map in the implicit sampling equation.

[7] We also consider a few different heuristics to improve the estimation results and to reduce the computational cost of the methods. We consider a case in which observations are available at every time step. However, we apply the implicit particle filter and do the sampling over time intervals of a desired length. We change the implicit filter equations slightly to incorporate all the available measurements during the interval. As will be seen, this heuristic method is very effective in the case of the system under consideration. We believe that this block-sampling implementation of implicit particle filter when measurements are available at every time step can be beneficial in the case of dynamic systems with *band-diagonal* structure, i.e., systems where the value of the state at each cell is determined by the value of the state at the neighboring cells at the previous time step. Note that most physical system lie in this category, since, in such systems, information propagates in space continuously with time. We will elaborate more on this in section 5.1.

[8] As a second heuristics, we propose a *maximum a posteriori* (MAP) estimation method to calculate the state trajectory over the time interval between two measurements at time steps $k+1$ and $k+r$, which maximizes the posterior density function in this interval. This requires the knowledge of the conditional pdf $p(x_k|z_{1:k})$ ($z_{1:k}$ represents the collection of all measurements up to time step k), which we approximate by a Gaussian density function, the mean and covariance of which are calculated from the estimation over the previous interval. As will be seen, this amounts to a probabilistic version of weak constraint 4D-Var [*Tremolet*, 2006, 2007], which is a variational data assimilation method used

in meteorology. In cases in which the posterior density is symmetric or almost symmetric and the approximation of conditional density $p(x_k|z_{1:k})$ with a Gaussian density is not too inaccurate, the MAP method may provide more accurate estimates while having a much lower computational cost than the implicit filter.

[9] The rest of this article is organized as follows: In section 2, the one- (1-D) Saint-Venant equations are presented, and a state space model of the flow in a network of open channels is constructed. In section 3, we review the *optimal* SIR filter and the implicit particle filters. The random-map implementation of the implicit particle filter and its application is reviewed in section 4. We propose two heuristic methods in section 5 to perform the data assimilation to improve performance and lower the computational cost in the case of system under consideration. Section 6 provides information about the implementation of methods in a network of open channels in Sacramento-San Joaquin Delta in California. In this section, the numerical results are presented, and the performance of different estimation methods is compared. Finally, we conclude this article in section 7.

2. Flow Model

2.1. Saint-Venant Model

[10] The Saint-Venant model is among the most common models used for modeling the flow in open channels and irrigation systems *Chow* [1988], *Cunge et al.* [1980]. In 1-D case, Saint-Venant equations are two coupled first-order hyperbolic PDEs derived from conservation of mass and momentum. For prismatic channels with no lateral inflow, these equations can be written as follows [*Strum*, 2001]:

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

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

for $(x,t)\in(0,L)\times\Re^+$, where L is the river reach (m), $Q(x,t)$ is the discharge or flow (m^3 s^{-1}) 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), $S_f(x,t)$ is the friction slope (m m^{-1}), S_b is the bed slope (m m^{-1}), g is the gravitational acceleration (m s^{-2}), and h_c is the distance of the centroid of the cross section from the free surface (m).

[11] The friction slope is empirically modeled by the Manning-Strickler formula [*Litrico and Fromion*, 2009]:

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

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

[12] In the case of subcritical flow, the boundary conditions are taken to be upstream flow $Q(0,t)$ and downstream stage $H(L,t)$ or downstream flow $Q(L,t)$ and upstream stage $H(0,t)$ [*Litrico and Fromion*, 2009].

[13] For channels with nonrectangular cross sections, three correction parameters, α, η, and γ can be introduced through the following equations: $A=\alpha TH$, $P=\eta(2T+H)$, and $h_c=\gamma H$. These parameters are calculated based on the average stage.

2.2. Discretization

[14] We use the Lax diffusive scheme [*Chaudhry*, 2008; *Strum*, 2001], which is a first-order explicit scheme to discretize the equations at internal grid points. Using f to represent the state variables, Q and H, the derivatives become

$$\frac{\partial f}{\partial t}=\frac{f_i^{k+1}-\frac{1}{2}(f_{i+1}^k+f_{i-1}^k)}{\Delta t}, \tag{4}$$

$$\frac{\partial f}{\partial x}=\frac{f_{i+1}^k-f_{i-1}^k}{2\Delta x}, \tag{5}$$

using traditional finite difference discretization notation, with subscript i for space and superscript k for time.

[15] Applying this scheme to equations (1) and (2), we obtain the following set of finite difference equations:

$$A_i^{k+1}=\frac{1}{2}\left(A_{i-1}^k+A_{i+1}^k\right)-\frac{\Delta t}{2\Delta x}\left(Q_{i+1}^k-Q_{i-1}^k\right) \tag{6}$$

$$\begin{aligned}Q_i^{k+1}=\frac{1}{2}\left(Q_{i-1}^k+Q_{i+1}^k\right)-\frac{\Delta t}{2\Delta x}&\left[\left(\frac{Q^2}{A}+gAh_c\right)_{i+1}^k\right.\\&\left.-\left(\frac{Q^2}{A}+gAh_c\right)_{i-1}^k\right]+\Delta t\left(\frac{\phi_{i+1}^k+\phi_{i-1}^k}{2}\right),\end{aligned} \tag{7}$$

where

$$\phi=gA\left(S_b-S_f\right). \tag{8}$$

[16] This scheme is stable, provided that the *Courant-Friedrich-Lewy* condition holds, i.e.,

$$\frac{\Delta t}{\Delta x}|V+C|\le 1, \tag{9}$$

where $C=\sqrt{gD}$ is the *wave celerity* with $D=A/T$ the hydraulic depth (m), and V is the average velocity.

[17] However, 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. Here, we use the *method of specified time intervals* to compute these variables [*Chaudhry*, 2008]. 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.

[18] Denoting the projected points corresponding to the upstream and downstream boundary conditions by subscripts L and R, respectively, the resulted update equations for upstream stage and downstream velocity read as follows:

$$H_1^{k+1} = H_L^k + \frac{C_L^k}{g}\left(V_1^{k+1} - V_L^k\right) + C_L^k \Delta t\left(S_{fL}^k - S_{b_1}\right), \tag{10}$$

$$V_N^{k+1} = V_R^k + g\frac{H_R^k - H_N^{k+1}}{C_R^k} - g\Delta t\left(S_{fR}^k - S_{b_N}\right), \tag{11}$$

where the velocity and wave celerity at projected points can be calculated as follows:

$$V_L^k = \frac{V_1^k + \beta\left(C_1^k V_2^k - C_2^k V_1^k\right)}{1 + \beta\left(-V_1^k + V_2^k + C_1^k - C_2^k\right)}, \tag{12}$$

$$C_L^k = \frac{C_1^k + \beta V_L^k\left(C_1^k - C_2^k\right)}{1 + \beta\left(C_1^k - C_2^k\right)}, \tag{13}$$

$$V_R^k = \frac{V_N^k + \beta\left(C_N^k V_{N-1}^k - C_{N-1}^k V_N^k\right)}{1 + \beta\left(-V_N^k + V_{N-1}^k + C_N^k - C_{N-1}^k\right)}, \tag{14}$$

$$C_R^k = \frac{C_N^k + \beta V_R^k\left(C_{N-1}^k - C_N^k\right)}{1 + \beta\left(C_N^k - C_{N-1}^k\right)}, \tag{15}$$

where $\beta = \Delta t/\Delta x$, C_i^k is the wave celerity at cell i at time k, and N is the number of cells.

2.3. Internal Conditions for Confluence in Channel Network

[19] To apply this model to a channel network, it is necessary to impose networked internal boundary conditions at every confluence in the channel network. Here, the internal boundary condition constraints are briefly described for a simple confluence as in Figure 1, which comprises three channels.

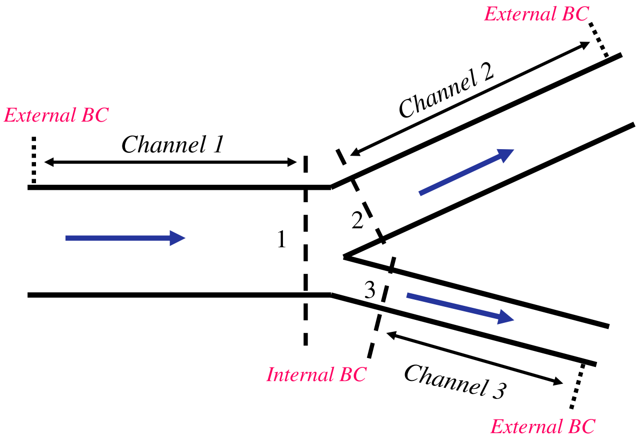

Figure 1. Illustration of internal and external boundary conditions for a channel network.

[20] The constraints corresponding to the internal boundary conditions of stage and discharge are as follows:

$$H_1 = H_2 = H_3,$$

$$Q_1 = Q_2 + Q_3,$$

where H_1, H_2, and H_3 represent the stages in the cross sections 1, 2, and 3, respectively, and Q_1, Q_2, and Q_3 are the discharges at the three cross sections. The first equation is simply consistency of stage in all channels at the junction, and the second equation is just the conservation of mass at the junction.

[21] The discretized equations obtained from the Lax scheme and the method of specified time intervals along with the internal boundary constraints assembled to obtain a state space model for the entire network of interest is written in a compact form as follows:

$$x_{k+1} = f(x_k, u_k), \tag{16}$$

where x_k is the state vector at time k, which consists of discharge and stage at all cells throughout the whole network excluding the external boundary condition variables, and the input vector u_k contains the external boundary conditions.

2.4. Stochastic State Space Model

[22] In the state space model (16), the uncertainties can be categorized as the modeling errors and the input (boundary conditions) uncertainties. The former resulted from modeling simplifications and the uncertainties in the parameters of shallow water equations, such as the Manning coefficient or uncertainties in the bathymetry data due to incomplete knowledge of them. For more detailed uncertainty analysis of the shallow water equations, see *Liu* [2009]. 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 equation (16) to obtain a stochastic equation:

$$x_{k+1} = f(x_k, u_k) + v_k. \tag{17}$$

[23] The noise v_k is usually assumed to be zero-mean white Gaussian and

$$E[v_k v_l^T] = Q_k \delta_{kl}, \tag{18}$$

where δ_{kl} is the Kronecker delta, which is one if k and l are equal and is zero otherwise.

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

$$x_0 = \mathcal{N}(\bar{x}_0, P_0), \tag{19}$$

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

[25] Similarly, the errors and uncertainties in the measurements can be taken into account by adding a noise term to the measurement model to obtain

$$z_k = g(x_k) + e_k, \tag{20}$$

where g is the function that relates the measurements to the state vector and e_k is the measurement noise of the sensors which is assumed to be zero-mean white Gaussian and

$$E[e_k e_l^T] = R_k \delta_{kl}. \tag{21}$$

We also assume that the process and measurement noises and the initial conditions are all uncorrelated.

3. Optimal SIR Filter

[26] In Bayesian estimation, the goal is to recursively calculate the conditional *pdf* $p(x_k|z_{1:k})$, where x_k is the state vector at time k and $z_{1:k}$ is the set of measurements obtained up to time step k. Assuming the initial state pdf $p(x_0)$ is known, the pdf $p(x_k|z_{1:k})$ may be calculated recursively in two steps, *prediction step* and *update step*. The prediction step uses the state space model to propagate the conditional pdf forward in time. In other words, it calculates $p(x_k|z_{1:k-1})$, given $p(x_{k-1}|z_{1:k-1})$ via the Chapman-Kolmogorov equation:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\mathrm{d}x_{k-1}. \tag{22}$$

[27] In the update step, when the measurements z_k become available, the conditional pdf is updated using the Bayes' rule:

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}, \tag{23}$$

where

$$p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1})\mathrm{d}x_k, \tag{24}$$

[28] While the above set of equations theoretically solves the Bayesian estimation problem, analytic solutions are tractable only in certain simplified cases, e.g., *Kalman* [1960] filter for linear systems with Gaussian noise. For more general cases, different approximate solutions have been devised. Extended Kalman filters [*Anderson and Moore*, 1979], approximate grid-based filters, unscented Kalman filters [*Julier and Uhlmann*, 1997; *Wan and van der Merwe*, 2000; *Haykin*, 2001], and particle filters [*Doucet et al.*, 2001] are the examples of these approximate methods.

[29] Particle filtering is a sequential Monte Carlo method that calculates approximate solutions to the above equations for a general case of nonlinear systems with arbitrary process and measurement noises. The basic idea behind particle filters is that the posterior pdf $p(x_{0:k}|z_{1:k})$, where $x_{0:k} = \{x_j, j = 0, \ldots, k\}$ is the set of all state vectors up to time k, is approximated using a number of particles or random samples with their corresponding weights (probabilities). In other words,

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta\left(x_{0:k} - x_{0:k}^i\right), \tag{25}$$

where $x_{0:k}^i$ is the ith sample, w_k^i is its corresponding weight (the weights are normalized so they sum to one), and N_s is the number of samples. The estimates are computed using the particles, and their associated weights and the weights are chosen using the principle of importance sampling [*Bergman*, 1999].

[30] A common problem with the sequential importance sampling particle filter is the degeneracy problem. However, a good choice of importance density can reduce the degeneracy of the particles. In this work, we choose the importance density $q\left(x_k|x_{k-1}^i, z_k\right)$ to be

$$\begin{aligned} q\left(x_k|x_{k-1}^i, z_k\right) &= p\left(x_k|x_{k-1}^i, z_k\right) \\ &= \frac{p\left(z_k|x_k, x_{k-1}^i\right)p\left(x_k|x_{k-1}^i\right)}{p\left(z_k|x_{k-1}^i\right)}. \end{aligned} \tag{26}$$

[31] It has been shown [*Doucet et al.*, 2000] that this choice of importance density minimizes the variance of *true weights*, w_k^{*i} defined as $w_k^{*i} = p\left(x_k^i|z_{1:k}\right)/q\left(x_k^i|x_{k-1}^i, z_k\right)$, which, in turn, maximizes the effective sample size defined as $N_{\mathrm{eff}} = \frac{N_s}{1+\mathrm{Var}\left(w_k^{*i}\right)}$. We assume that the process and measurement noises are mutually independent and *independent and identically distributed* Gaussian and

$$v_{k-1} \sim \mathcal{N}(\mathbf{0}, Q_{k-1}), \tag{27}$$

$$e_{k-1} \sim \mathcal{N}(\mathbf{0}, R_{k-1}). \tag{28}$$

[32] With this assumption, for systems with nonlinear dynamics and linear measurement model

$$x_k = f_k(x_{k-1}) + v_{k-1}, \tag{29}$$

$$z_k = H_k x_k + e_k, \tag{30}$$

it has been shown that $p\left(x_k|x_{k-1}^i, z_k\right)$ is Gaussian [*Doucet et al.*, 2000; *Del Moral*, 1998] and

$$p(x_k|x_{k-1}, z_k) = \mathcal{N}(m_k, \Sigma_k), \tag{31}$$

$$p(z_k|x_{k-1}) = \mathcal{N}\left(H_k f_k(x_{k-1}), Q_{k-1} + H_k R_k H_k^T\right), \tag{32}$$

with

$$\Sigma_k^{-1} = Q_{k-1}^{-1} + H_k^T R_k^{-1} H_k, \tag{33}$$

$$m_k = \Sigma_k\left(Q_{k-1}^{-1} f_k(x_{k-1}) + H_k^T R_k^{-1} z_k\right). \tag{34}$$

[33] With this choice of importance density, the weights update equation simplifies to [*Arulampalam et al.*, 2002]

$$w_k^i \propto w_{k-1}^i p\left(z_k|x_{k-1}^i\right), \tag{35}$$

$$= w_{k-1}^i \int p\left(z_k|x_k'\right)p\left(x_k'|x_{k-1}^i\right)\mathrm{d}x_k', \tag{36}$$

[34] When the measurements are obtained, equation (32) can be used to calculate $p\left(z_k|x_{k-1}^i\right)$ to update the weights using equation (35).

[35] When sample impoverishment still occurs even with the choice of optimal importance density, resampling can be applied whenever the effective sample size becomes too small [*Carpenter et al.*, 1999]. A pseudocode description of the resampling algorithm is provided in Algorithm 1 [*Arulampalam et al.*, 2002]. The algorithm generates a new set of particles $\{x_k^{j*}\}_{j=1}^{N_s}$ with equal weights $1/N_s$. In this resampling method, the particles with small weights, which correspond to the low-probability regions of the posterior density function, are discarded, and the particles with larger weights are replicated according to their weights to keep the sample size constant.

Algorithm 1: Resampling algorithm

```
$\left[\{x_k^{j*}, w_k^j, i^j\}_{j=1}^{N_s}\right] = \mathrm{RESAMPLE}\left[\{x_k^i, w_k^i\}_{i=1}^{N_s}\right]$
Initialize the CDF: $c_1 = 0$
for $i = 2$ to $N_s$ do
    Construct CDF: $c_i = c_{i-1} + w_k^i$
end for
Start at the bottom of the CDF: $i = 1$
Draw a starting point: $u_1 \sim \mathbb{U}\left[0, N_s^{-1}\right]$
for $j = 1$ to $N_s$ do
    Move along the CDF: $u_j = u_1 + \frac{j-1}{N_s}$
    while $u_j > c_j$ do
        $i = i + 1$
    end while
    Assign sample: $x_k^{j*} = x_k^i$
    Assign weight: $w_k^j = \frac{1}{N_s}$
    Assign parent: $i^j = i$
end for
```

[36] Algorithm 2 illustrates the particle filter with resampling.

Algorithm 2: Particle filter with resampling

```
$\left[\{x_k^{j*}, w_k^j\}_{j=1}^{N_s}\right] = \mathrm{PF}\left[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, z_k\right]$
for $i = 1$ to $N_s$ do
    Draw $x_k^i \sim q\left(x_k|x_{k-1}^i, z_k\right) = p\left(x_k|x_{k-1}^i, z_k\right)$ from (31)
    Assign the particle a weight, $w_k^i$, using (36).
end for
Calculate total weight: $t = \sum_{i=1}^{N_s} w_k^i$
for $i = 1$ to $N_s$ do
    Normalize: $w_k^i = \frac{w_k^i}{t}$
end for
Calculate $\hat{N}_{\mathrm{eff}} = \frac{1}{\sum_{i=1}^{N_s} \left(w_k^i\right)^2}$
if $\hat{N}_{\mathrm{eff}} < N_T$ then
    Resample using algorithm 1:
    $\left[\{x_k^i, w_k^i, -\}_{i=1}^{N_s}\right] = \mathrm{RESAMPLE}\left[\{x_k^i, w_k^i\}_{i=1}^{N_s}\right]$
end if
```

[37] It is worth noting that high effective sample size does not guarantee the effectiveness of the particle filter. Even in cases in which the degeneracy does not happen, i.e., a sufficient number of weights are not zero, the particle filter may not perform efficiently if all particles belong to a low-probability region. As an example of an extreme case, if all particles correspond to a low-probability region where the density function is almost uniform, the normalized weights will be all nonzero; however, the particles are not approximating the density function well. This motivates the implicit particle filter, which is the subject of the following section.

4. Implicit Particle Filter

[38] As discussed in the previous section, the main issue with the particle filters is the fact that many of the particles end up in low-probability regions, and, consequently, a considerable amount of computational effort is wasted in propagating these particles forward while they do not contribute to the approximation of the posterior density. Implicit particle filters have been developed to solve this problem [*Chorin and Tu*, 2009; *Chorin et al.*, 2010; *Morzfeld et al.*, 2012]. The basic idea behind implicit particle filters is to create samples with high probability. Although the computational cost per particle is, in general, higher in implicit filters, a better accuracy may be obtained by using a smaller number of particles than traditional particle filters. From another point of view, the implicit particle filter does not model the tails of the posterior. However, it should be noted that the tails of the posterior are not of statistical significance and particles corresponding to the tails of the posterior do not contribute to point estimates, which are normally taken as the mean of the density function.

[39] We consider a case in which intermittent observations are available. Suppose that we have a collection of N_s particles, $\{x_k^j, \text{ for} j = 1, \ldots, N_s\}$, and we have a set of observations $z_{0:k}$ up to time step k. We consider a case in which the next observation of the system, z_{k+r}, becomes available at time $k + r$. We can write the posterior as follows:

$$\begin{aligned} P\left(x_{0:k+r}^j|z_{0:k}, z_{k+r}\right) &= P\left(x_{0:k}^j|z_{0:k}\right) \\ &\times P\left(x_{k+1}^j|x_k\right) \cdots P\left(x_{k+r-1}^j|x_{k+r-2}^j\right) \\ &\times P\left(x_{k+r}^j|x_{k+r-1}^j\right) P\left(z_{k+r}|x_{k+r}^j\right)/Z_k, \end{aligned} \tag{37}$$

where Z_k is a normalization factor. The goal of implicit sampling is to obtain high-probability samples from $P\left(x_{0:k+r}^j|z_{0:k}, z_{k+r}\right)$. This is done by defining a function $F_j(x_{k+1:k+r})$ via *Morzfeld et al.* [2012]

$$\begin{aligned} \exp\left(-F_j\left(x_{k+1:k+r}^j\right)\right) &= P\left(x_{k+1}^j|x_k\right) \cdots P\left(x_{k+r-1}^j|x_{k+r-2}^j\right) \\ &\times P\left(x_{k+r}^j|x_{k+r-1}^j\right) P\left(z_{k+r}|x_{k+r}^j\right)/Z_k. \end{aligned} \tag{38}$$

A high-probability sample can be obtained by solving the following algebraic equation for $x_{k+1:k+r}^j$:

$$F_j\left(x_{k+1:k+r}^j\right) - \phi_j = \frac{1}{2}\xi_j^T \xi_j, \tag{39}$$

where ξ_j is a sample from a zero-mean Gaussian distribution, $P_\xi = \mathcal{N}(0, I)$, and $\phi_j = \min F_j(X)$. It is clear that, with this sampling method, we obtain a sample with high probability.

[40] The weights corresponding to the particles are calculated as follows:

$$w^j_{k+1} \propto w^j_{k+1}\exp(-\phi_j)J, \tag{40}$$

where $J = \left|\det\frac{\partial x^j_{k+1:k+r}}{\partial \xi_j}\right|$ is the Jacobian of the map between ξ_j and the particle trajectory $x^j_{k+1:k+r}$.

[41] In the special case of the system considered in this article in which the observation model is linear and the process noise and measurement noise are assumed to be Gaussian, the function F_j will have the following form:

$$\begin{aligned}F_j\left(x^j_{k+1:k+r}\right) = \log&\left(\frac{1}{(2\pi)^{(nr+m)/2}}\prod_{i=k}^{k+r}\frac{1}{\sqrt{\det Q_i}}\frac{1}{\sqrt{\det R_{k+r}}}\right)\\ &+\frac{1}{2}\sum_{i=k}^{k+r-1}(x_{i+1}-f(x_i,u_i))^T Q_k^{-1}(x_{i+1}-f(x_i,u_i))\\ &+\frac{1}{2}(z_{k+r}-H_{k+r}x_{k+r})^T R^{-1}_{k+r}(z_{k+r}-H_{k+r}x_{k+r}).\end{aligned} \tag{41}$$

[42] To implement the implicit particle filter for the system considered, the main tasks at each time step are calculating $\min F_j(X)$, the Jacobian of the implicit map J, and solving equation (39) for each particle to obtain a sample. In what follows, we elaborate on how we handle each one of these problems for our application.

4.1. Calculating the Minimum of Implicit Sampling Functions

[43] To calculate $\min F_j(X)$, we use the Newton method for which we need to calculate the Jacobian and Hessian of F_j. Note that the function F_j is in general a different function for each particle. Therefore, the Jacobian and Hessian must be calculated for each particle individually. This may make the implicit sampling procedure computationally expensive in cases in which the Jacobian and Hessian of F_j cannot be calculated analytically. Fortunately, in our case, we can calculate the partial derivatives for the interior grid points analytically. Moreover, we use a numerical method to calculate the partial derivatives at boundary points. This reduces the computational cost of the implicit sampling significantly.

[44] We can write the Jacobian, J^j, of F_j as follows:

$$J^j\left(x^j_{k+1:k+r}\right) = (A_{k+1},\ldots,A_{k+r}), \tag{42}$$

$$A_{k+1} = (X_{k+1}-f(x_k))^T Q_k^{-1}, \tag{43}$$

$$\begin{aligned}A_i &= (X_{i+1}-f(x_i))^T Q_i^{-1} - (X_{k+1}-f(x_k))^T Q_i^{-1} J_f(x_i)\\ &\text{for } i = k+2,\ \ldots,\ k+r-1,\end{aligned} \tag{44}$$

$$\begin{aligned}A_{k+r} = (X_{k+r+1}-f(x_{k+r}))^T Q^{-1}_{k+r-1}\\ +(Hx_{k+r}-z)R^{-1}_{k+r-1}H,\end{aligned} \tag{45}$$

where $J_f\left(x^j_k\right)$ is the Jacobian matrix of the system's state space function f, which we calculate analytically for interior grip points and numerically at the boundary points in each channel. The analytical partial derivatives of f for interior grid points are provided in Appendix A.

[45] The Hessian of F_j can be written as a band block-diagonal sparse matrix where the diagonal blocks are as follows:

$$H_F = \begin{pmatrix} D_1 & C_1 & 0 & \cdots & 0\\ C_1^T & D_2 & C_2 & \ddots & \vdots\\ 0 & C_2^T & \ddots & \ddots & 0\\ \vdots & \ddots & \ddots & D_{r-1} & C_{r-1}\\ 0 & \cdots & 0 & C_{r-1}^T & D_r \end{pmatrix}$$

with

$$\begin{aligned}C_i &= -J_f(x_{i+1})^T Q^{-1}_{i+1},\\ D_r &= Q_r^{-1} + H^T_{k+r}R^{-1}H_{k+r},\\ D_i &= Q_i^{-1} + J_f(x_i)^T Q^{-1}_{i+1}J_f(x_i) + L^l_i,\ i<r,\\ L^l_i &= (x_{i+1}-f(x_i))^T Q^{-1}_{i+1}H^l_f(x_i),\end{aligned}$$

where $\mathcal{H}^l_f(x_i) = \left[\frac{\partial^2 f_s}{\partial x_l \partial x_t}\right]_{s,t}$ is a slice of the Hessian matrix of the system's state space function f, which we calculate analytically for interior grid points and numerically at the boundary points in each channel. The analytical partial derivatives of f for interior grid points are provided in Appendix A. Note that f_s denotes the sth component of f.

[46] To obtain the initial seed for the Newton method in computation of $\min F_j(X)$, we run the forward simulation with the mean of particles as initial condition, and we run a few steps of Newton method to polish the initial guess of the minimizing solution. We use a line-search method to find a proper step size for the Newton method.

4.2. Solving the Sampling Equation via a Random Map

[47] To solve equation (39), we use the random-map method introduced by *Morzfeld et al.* [2012]. Assuming that the level sets of the function F_j are closed, the following map is considered as the map between ξ_j and $x^j_{k+1:k+r}$:

$$x^j_{k+1:k+r} = \mu_j + \lambda_j L_j^T \eta_j, \tag{46}$$

where $\eta_j = \xi_j/\sqrt{\xi_j^T\xi_j}$ is the direction of the sample ξ_j of the Gaussian reference variable $\xi_j \sim \mathcal{N}(0, I)$; μ_j is the location of the minimum of F_j, i.e., $F_j\left(\mu_j\right) = \phi_j$; and $L_j \in \mathbb{R}^{nr\times nr}$ is an invertible matrix chosen deterministically. It has been observed that taking L_j as the Cholesky factorization of the inverse of the Hessian, i.e., $H_j^{-1} = L_j L_j^T$ is a good choice for L_j specially if F_j is nearly quadratic.

[48] Note that, in equation (46), only λ_j is unknown after the reference Gaussian variable is sampled. By substituting equation (46) in equation (39), one obtains an algebraic equation for the single scalar λ_j, which can be easily solved.

This will yield a sample for the particle trajectory $x^j_{k+1:k+r}$. Assuming the level sets of function F_j are closed, solving equation (46) for λ_j corresponds to a solution of equation (39) in a specific direction, $L_j^T\eta_j$.

4.3. Calculating the Jacobian of the Implicit Sampling Function

[49] By differentiating equation (46), we obtain the following equation [*Morzfeld et al.*, 2012]:

$$\frac{\partial x^j_{k+1:k+r}}{\partial \xi_j} = L_j^T\left(\eta_j \frac{\partial \lambda_j}{\partial \xi_j}\right) + \lambda L_j^T \frac{\partial \eta_j}{\partial \xi_j}. \tag{47}$$

[50] After some manipulations, the Jacobian of the implicit map can be written as follows:

$$J = 2|\det L_j|\rho_j^{1-nr/2}\left|\lambda_j^{rn-1}\frac{\partial \lambda_j}{\partial \rho_j}\right|, \tag{48}$$

where $\rho_j = \xi_j^T\xi_j$ and $\frac{\partial \lambda_j}{\partial \rho_j}$ can be calculated numerically.

[51] A pseudocode description of the implicit particle filter is presented in Algorithm 3.

Algorithm 3: Implicit particle filter with resampling

```
$\left[\{x^j_k, w^j_k\}_{j=1}^{N_s}\right] = \text{Imp-PF}\left[\{x^j_{k-1}, w^j_{k-1}\}_{j=1}^{N_s}, z_k\right]$
for $j = 1$ to $N_s$ do
    Calculate $\phi_j = \min F_j(X)$ .
    Draw a sample $\xi_j \sim \mathcal{N}(0, I)$
    Calculate the Hessian of $F_j$, $H_j$ and set $L_j = \text{chol}(H_j^{-1})$
    Substitute the random map, equation (46) in the sampling equation (39) and solve for $\lambda_j$ to obtain a sample trajectory $x^j_{k+1:k+r}$.
    Calculate the Jacobian of the implicit map using (48).
    Update the particle weight, $w^j_k$, using equation (40).
end for
Calculate total weight: $t = \sum_{j=1}^{N_s} w^j_k$
for $j = 1$ to $N_s$ do
    Normalize: $w^j_k = \frac{w^j_k}{t}$
end for
Calculate $\hat{N}_{\text{eff}} = \frac{1}{\sum_{j=1}^{N_s}\left(w^j_k\right)^2}$
if $\hat{N}_{\text{eff}} < N_T$ then
    Resample using algorithm 1:
    $\left[\{x^j_k, w^j_k, -\}_{j=1}^{N_s}\right] = \text{RESAMPLE}\left[\{x^j_k, w^j_k\}_{j=1}^{N_s}\right]$
end if
```

5. Heuristics

5.1. Implicit Particle Filter With Block-Sampling

[52] In this section, we consider a case in which observations are available at every time step. It is known that, in the case of systems with linear observation model and Gaussian noises when measurements are available at every time step, the implicit particle filter becomes equivalent to the optimal SIR filter [*Morzfeld et al.*, 2012]. Nonetheless, as a heuristic method, we use the implicit particle filter to do the sampling every r time steps while using the measurements at all time steps instead of sampling at every time step. We believe that, for systems with band-diagonal structure, i.e., systems in which the value of each state at the next time step is determined by the current values of its neighboring states, this heuristic method may improve the estimation results. The shallow water model used in the current article is an example of such a system. As can be seen in the discretized Saint-Venant equations in section 2.2, the value of flow or stage at each cell is determined by values of flow and stage at its neighboring cells at the previous time step. In such systems, the information is propagated in space one cell every time step. Therefore, a measurement provides information about the value of the state at a cell, which is r cells far from the location of the measurement at r time steps before. This suggests that using all the measurements obtained over a block of time steps and performing the sampling over the block may improve the results. To implement this method, the implicit function needs to be slightly modified as follows so that the filter uses all the available measurements:

$$\begin{aligned} P\left(x^j_{0:k+r}|z_{0:k+r}\right) &= P\left(x^j_{0:k}|z_{0:k}\right) \\ &\times P\left(x^j_{k+1}|x_k\right)\cdots P\left(x^j_{k+r-1}|x^j_{k+r-2}\right)P\left(x^j_{k+r}|x^j_{k+r-1}\right) \\ &\times P\left(z_{k+1}|x^j_{k+1}\right)\cdots P\left(z_{k+r}|x^j_{k+r}\right)/Z_k, \end{aligned} \tag{49}$$

$$\begin{aligned} F_j\left(x^j_{k+1:k+r}\right) = \log&\left(\frac{1}{(2\pi)^{(nr+m)/2}}\prod_{i=k}^{k+r}\frac{1}{\sqrt{\det Q_i}}\frac{1}{\sqrt{\det R_{k+r}}}\right) \\ &+ \frac{1}{2}\sum_{i=k}^{k+r-1}\left(x_{i+1} - f(x_i, u_i)\right)^T Q_k^{-1}\left(x_{i+1} - f(x_i, u_i)\right) \\ &+ \frac{1}{2}\sum_{i=k+1}^{k+r}\left(z_i - H_{k+r}x_{k+r}\right)^T R_{k+r}^{-1}\left(z_i - H_{k+r}x_{k+r}\right). \end{aligned} \tag{50}$$

Moreover, the Jacobian and Hessian must be changed accordingly.

5.2. MAP Estimation

[53] The idea investigated in this section is to propagate the maximum-probability trajectory obtained in the minimization step of the implicit particle filter forward and to use that as the estimates. This significantly reduces the computational cost of the method, as there will be no need to create particles. Moreover, the minimization problem needs to be solved only once instead of once for each particle. The simplest thing to do would be to calculate $\max_{x_{k+1:k+r}} P(x_{k+1}, \ldots, x_{k+r}|x_k, z_{k+r})$ using x_k as a deterministic vector obtained from the maximization over the previous time interval. However, this does not take into account the uncertainty in the value of x_k. To account for the uncertainty in x_k, one can consider x_k as a random variable. Considering x_k as a random variable, we aim to find the MAP estimate of the state trajectory over the interval between time steps $k+1$ and $k+r$, i.e., the solution of the following optimization problem:

$$\hat{x}_{k:k+r} = \arg\max_{x_{k:k+r}} P(x_k, \ldots, x_{k+r}|z_{1:k+r}). \tag{51}$$

The cost function in equation (51) can be factored as follows:

$$\begin{aligned} P(x_{k:k+r}|z_{1:k+r}) &= \frac{P(x_{k:k+r}|z_{1:k})P(z_{k+r}|z_{1:k},x_{k:k+r})}{P(z_{k+r}|z_{1:k})} \\ &= \frac{P(x_k|z_{1:k})P(x_{k+1:k+r}|x_k)P(z_{k+r}|x_{k+r})}{P(z_{k+r}|z_{1:k})}. \end{aligned} \tag{52}$$

Defining the function $\tilde{F}$ as

$$\exp\left(-\tilde{F}(x_{k:k+r})\right) = P(x_k|z_{1:k})P(x_{k+1:k+r}|x_k)P(z_{k+r}|x_{k+r}), \tag{53}$$

we have

$$\hat{x}_{k:k+r} = \arg\min_{x_{k:k+r}} \tilde{F}(x_{k:k+r}). \tag{54}$$

Now, we explain how we solve the above optimization problem. It can be readily seen that

$$\tilde{F}(x_{k:k+r}) = F(x_{k+1:k+r}) - \log(P(x_k|z_{1:k})), \tag{55}$$

where F is the function defined in equation (38). The posterior density function $P(x_k|z_{1:k})$ is not generally available, but we approximate it with a Gaussian density, $\mathcal{N}(\hat{x}_k, K_k)$. The mean, $\hat{x}_k$, is the estimate of x_k obtained from solving the MAP estimation over the previous interval. To find the covariance, we approximate $P(x_{k-r:k}|z_{1:k})$ with a Gaussian density with covariance being equal to the inverse of the Hessian of $P(x_{k-r:k}|z_{1:k})$, which is already available to us from solving the MAP estimation over the previous time interval. We then obtain the covariance of the marginal density $P(x_k|z_{1:k})$, which is the desired covariance matrix K_k.

[54] For the case of the system under consideration, we can write

$$\begin{aligned} \tilde{F}(x_{k:k+r}) = \log&\left(\frac{1}{(2\pi)^{(nr+m)/2}}\prod_{i=k}^{k+r}\frac{1}{\sqrt{\det Q_i}}\frac{1}{\sqrt{\det R_{k+r}}}\right) \\ &+ \frac{1}{2}\sum_{i=k}^{k+r-1}(x_{i+1}-f(x_i,u_i))^T Q_k^{-1}(x_{i+1}-f(x_i,u_i)) \\ &+ \frac{1}{2}(z_{k+r}-H_{k+r}x_{k+r})^T R_{k+r}^{-1}(z_{k+r}-H_{k+r}x_{k+r}) \\ &+ \frac{1}{2}(x_k-\hat{x}_k)^T K_k^{-1}(x_k-\hat{x}_k). \end{aligned} \tag{56}$$

The optimization problem (54) can be solved using the Newton method with the forward simulation predicted state trajectory as the initial seed to obtain an optimal solution. Note that $\tilde{F}(x_{k:k+r})$ has one additional quadratic term than $F(x_{k+1:k+r})$ and the calculated Jacobian and Hessian of F need to be slightly changed to obtain those of $\tilde{F}(x_{k:k+r})$.

[55] It is not hard to see that the above approach is similar to the weak constraint 4D-Var method, where a cost function consisting of quadratic terms accounting for model error, measurement error, and error in the initial conditions is minimized [*Tremolet*, 2006, 2007].

[56] Figure 2 shows a flowchart of the MAP estimation algorithm presented in this section.

[57] Note that in cases in which the posterior density $P(x_{k+1}, \ldots, x_{k+r}|x_k, z_{k+r})$ is symmetric, the MAP estimates will be more accurate than the estimates obtained from the weighted average of the particles, and as the number of particles converges to infinity, the weighted average estimate converges to the MAP estimate, which is the trajectory that maximizes the posterior density. In cases in which the posterior density $P(x_{k+1}, \ldots, x_{k+r}|x_k, z_{k+r})$ is close to

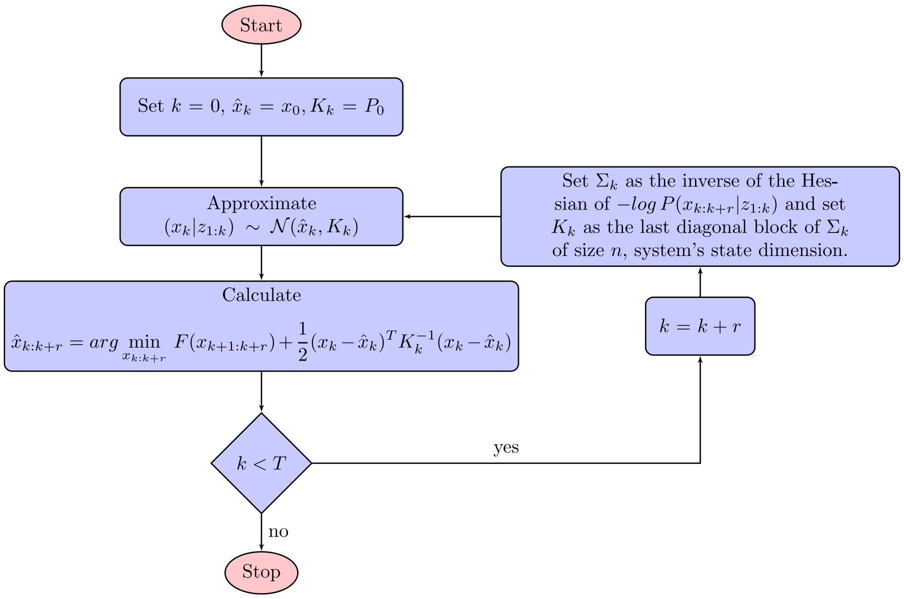
Start
Set $k = 0$, $\hat{x}_k = x_0, K_k = P_0$
Approximate $(x_k|z_{1:k}) \sim \mathcal{N}(\hat{x}_k, K_k)$
Calculate
$\hat{x}_{k:k+r} = arg \min_{x_{k:k+r}} F(x_{k+1:k+r}) + \frac{1}{2}(x_k - \hat{x}_k)^T K_k^{-1}(x_k - \hat{x}_k)$
$k < T$
yes
no
Stop
$k = k + r$
Set Σ_k as the inverse of the Hessian of $-log\, P(x_{k:k+r}|z_{1:k})$ and set K_k as the last diagonal block of Σ_k of size n, system's state dimension.

Figure 2. Flowchart of the MAP estimation heuristic method where T is the desired length of the experiment.

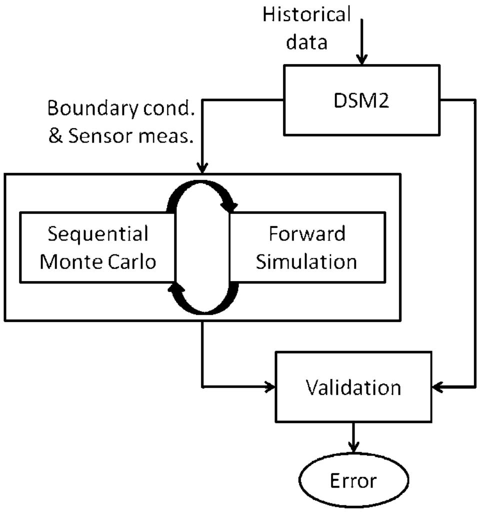

Figure 3. A flow diagram of the experiment.

symmetric and the approximation of $P(x_k|z_{1:k})$ with a Gaussian density is reasonable, we expect that the MAP method introduced above to be a very appealing estimation method, as it may provide more accurate estimates with a much lower computational cost than the optimal SIR and the implicit particle filter. As we apply the method for the system under consideration, we will see in section 6 that this is the case for this system. In other words, we obtain more accurate results with a smaller computational cost by using the MAP method.

[58] As a side remark, note that the implicit particle filter and the MAP method do not require the linearity of the measurement model and can be applied to systems with nonlinear measurement model without any changes. However, the optimal SIR filter is only valid for systems with Gaussian noise and linear observation model. Other versions of the particle filters must be used for systems with nonlinear observation model.

6. Implementation

6.1. Experiment Setup

[59] We consider a network of 19 subchannels and 1 reservoir in the southern part of Sacramento-San Joaquin to implement the data assimilation methods. The Sacramento-San Joaquin Delta, in northern California, is the hub of California's water system. This complex network covers 738,000 acres interlaced with more than 1150 km of tidally influenced channels and sloughs, and approximately 50% of California's average annual streamflow flows to the delta. Figure 4 shows a map of the delta.

[60] The network considered for implementation consists of the Clifton Court Forebay and its surrounding channels, which are parts of the Old River, the Italian Slough, the Mendota Canal, the West Canal, the Victoria Canal, and the Grant Line Canal, and is located on the northern side of Tracy. Clifton Court Forebay is a regulated reservoir at the head of the State Water Project's California Aqueduct. There are five radial gates, which control the flow into the forebay. The inflow is measured indirectly. The California *Department of Water Resources* (DWR) Delta Field Division calculates the difference in expected storage from the actual measured storage, which is used to calculate the inflow into the forebay. Figure 5 shows a satellite picture of the area and a map of the network with the channel configuration of the network, considered in building a 1-D model of the flow. As can be seen in Figure 5, this network consists of 1 reservoir, 19 subchannels, and 10 junctions. The total length of the channels in the network is 38,420 m. Table 1 presents the name and some geometry information about the channels. The channels are represented by their parent nodes from Figure 5.

[61] Figure 3 shows a flow diagram of the experiment. We use the Delta Simulation Model II (DSM2) to obtain measurements for boundary conditions and data assimilation as well as to evaluate our results. DSM2 is a 1-D mathematical model, developed in the DWR, for dynamic simulation of hydrodynamics, water quality, and particle tracking in the delta. DSM2 can calculate stages, flows, velocities, transport of individual particles, and mass transport processes for conservative and nonconservative constituents, including salt, water temperature, dissolved oxygen, and dissolved organic carbon. We use DSM2 since there are not enough U.S. Geological Survey (USGS) sensor stations in the area of interest to obtain boundary conditions and additional flow measurements needed for the data assimilation. DSM2 is one of the reference models used by the DWR for operations and will be considered in this work as the ground truth. Figure 6 compares the discharge and stage at the upstream of Old River (node 4 in Figure 5) generated by DSM2 with the USGS sensor measurements for 10–12 June 2006 and illustrates the applicability of DSM2 as a modeling tool to compute discharge and stage at locations where we need them for boundary conditions or as virtual sensor measurements. Note that DSM2 is a model of the whole delta and node 4 is one of its internal grid points. The motivation for using a different model for producing the measurements is to avoid committing the so-called inverse crime (which is the act of using the same model as both the forward and inverse models in a data assimilation experiment [*Kaipio and Somersalo*, 2004]) and assess the performance of the data assimilation methods in a more realistic setting.

6.2. Numerical Results

[62] We perform an experiment for a period of 25 h using historical data corresponding to 12 June 2006. We obtain the boundary conditions and measurements used for data assimilation from DSM2. As boundary conditions, discharge is imposed at nodes 1, 7, 15–17, and stage is imposed at nodes 4 and 12. Figure 7 shows the boundary conditions. The Tracy Pumping Plant is located at node 15, and, as can be seen in Figure 7b, there is a constant outflow soon after the experiment starts and there is no outflow at node 1 for the period of the experiment. To model the measurement noise, we add a zero-mean Gaussian noise with a variance of 5 $\mathrm{ft}^6\ \mathrm{s}^{-2}$ and 0.05 ft^2 to the boundary conditions for discharge and stage measurements, respectively.

[63] The number of cells in each subchannel is chosen in a way that the spatial step size in the subchannel is close to and not smaller than 900 ft, and the temporal step size is chosen to be 15 s. This choice of spatial step size results in 204 cells for the full network. Since, at each internal cell,

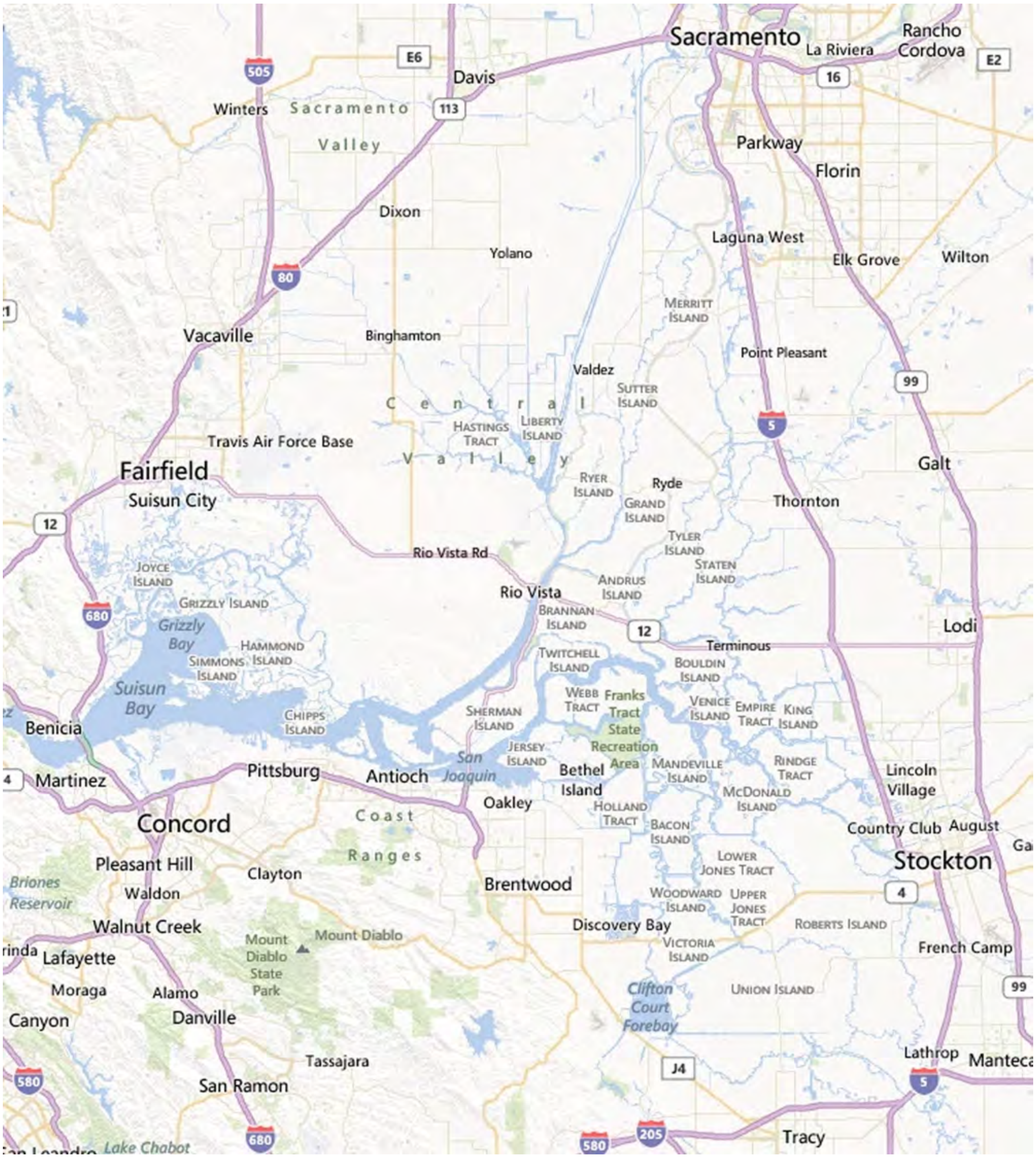

Figure 4. The Sacramento-San Joaquin Delta, image adapted from *Lund et al.* [2007]. The small box on the southern part of the delta is the network considered for implementation of the methods proposed in the current article.

there are two states, discharge and stage, and there is one state at the boundary cells, we will have a 401-dimensional system. We run DSM2 with a spatial step size of 5000 ft and a temporal step size of 15 min. We run DSM2 starting 1 day earlier so that the effects of inaccurate initial conditions are washed away and the DSM2 results are close to reality from the beginning of the experiment.

[64] Figure 8 shows the stage at two different locations computed by our model, the *forward simulation*, compared to the stage at the corresponding locations obtained from DSM2. As can be seen in Figure 8, the stage obtained from the forward simulation and DSM2 match closely. This is true at all locations throughout the network. However, there is a discrepancy between the discharge computed by the

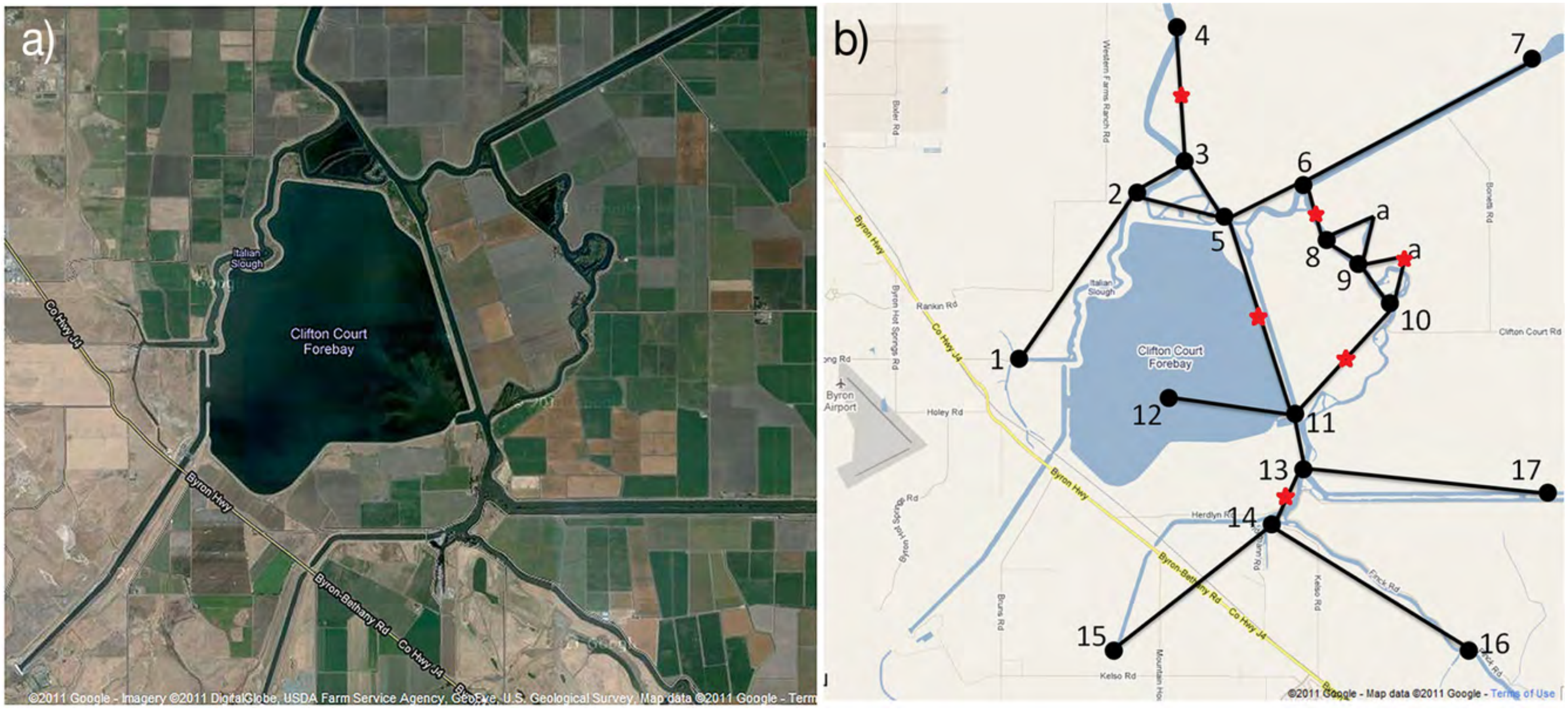

Figure 5. (a) Satellite image of the channel network around the Clifton Court Forebay used for the experiment. (b) The network connectivity of the channel network used for the experiment consisting of 19 subchannels, 10 junctions, and 1 reservoir. As boundary conditions, discharge is imposed at nodes 1, 7, 15, 16, and 17 and stage is imposed at nodes 4 and 12, where node 12 represents the reservoir. The red stars in Figure 5b show the locations from which flow measurements are obtained for data assimilation.

Table 1. The Names and Geometry Information of the Subchannels in the Open Channel Network in Sacramento-San Joaquin Delta in California Used for the Implementations

Channel	River	Length	Average Width	Average Depth
1–2	Italian Slough	14,198	234.0	14.0
2–3	Italian Slough	2723	203.4	16.4
2–5	Italian Slough	3227	437.5	10.3
3–4	Old River	4754	351.3	21.8
3–5	Old River	5022	351.2	22.0
5–6	Old River	4313	238.4	13.0
5–11	West Canal	10,041	253.0	28.0
6–7	Victoria Canal	8760	386.5	18.3
6–8	Old River	2722	276.3	15.2
8–9	Old River	2793	109.0	11.7
8a9	Old River	5347	157.1	9.0
9–10	Old River	2456	109.0	11.7
9a10	Old River	5062	157.1	9.0
10–11	Old River	7744	198.4	12.4
11–13	Old River	2609	266.0	19.0
13–14	Old River	3857	245.0	17.8
14–16	Old River	12,089	176.0	10.0
14–15	Mendota Canal	12,500	196.0	18.0
13–17	Grant Line Canal	15,831	404.0	16.0

forward simulation and DSM2. This is due to the fact that the bathymetry used for the forward simulation is different than that of DSM2. DSM2 computes the water surface (relative free surface elevation with respect to a datum), and we use the same bathymetry used for the forward simulation to convert the water surface to stage. This decreases the sensitivity of stage to bathymetry, which explains the lower discrepancy in stage computed by DSM2 and the forward simulation.

[65] To perform data assimilation, we only focus on assimilating discharge to improve the forward simulation results. We use six discharge measurements at the middle of channels 3–4, 6–8, 9a10, 10–11, 5–11, and 13–14, the locations of which are shown by red stars in Figure 5b. The process and measurement noises are assumed to be zero-mean white Gaussian noise. The measurement noise is assumed to be zero-mean Gaussian, as it is an assumption commonly made in accordance with the central limit theorem. The covariance of the noise is approximated by calculating the sample covariance obtained from historical data. We assume that the noises on different measurements are uncorrelated. At each cell, the process noise on discharge is assumed to be correlated with the discharge at its four neighboring cells from each side. The variance on discharge at each cell is taken to be 25 ft^6 s^{-2}, and the correlations are taken to be 20 ft^6 s^{-2}, 12 ft^6 s^{-2}, 8 ft^6 s^{-2}, 4 ft^6 s^{-2} with the discharge at the neighboring cells, respectively. These approximations of the noise variance were chosen based on the sample variances of the process noise at a few locations in the network, and other variances and correlations were approximated accordingly. The sensor measurements are obtained from DSM2, and a zero-mean Gaussian noise with a variance of 50 ft^6 s^{-2} is added to these measurements to simulate the uncertainty in the measurements.

[66] Using these flow measurements, the optimal SIR filter is applied. Figure 9 shows the results of the forward simulation and the SIR filter with 1000 particles compared with the corresponding results obtained from DSM2. The discharges at six locations in the network are illustrated for the period of the experiment. As can be seen in this figure, the SIR filter improves the model results significantly. To quantify the performance of the methods rigorously, we calculate the relative error throughout the whole domain at each time step using the following formula:

$$E(k) = \sqrt{\frac{\sum_{i=1}^{N_{\text{cells}}} \left(\hat{Q}_i^k - Q_i^k\right)^2}{\sum_{i=1}^{N_{\text{cells}}} \left(Q_i^k\right)^2}}, \tag{57}$$

where Q_i^k and $\hat{Q}_i^k$ are the true value of the flow and the estimated flow at cell i and time step k, respectively.

[67] The implicit particle filter is also applied for two cases, a case with sparse measurements and a case in which measurements are available at every time step, but we apply the first heuristic method to perform the sampling over intervals of 10 time steps while using all the available measurements. Figure 10 shows the time evolution of the relative error corresponding to the forward simulation, the SIR filter with 1000 particles, and the implicit particle filter with 50 particles while using measurements at every time step and performing block sampling. It can be seen in this figure that the relative error of the forward simulation has large peaks corresponding to more than 60% error. The SIR filter reduces the relative error to below 20% almost at all times. It can also be seen that using the implicit particle filter with the block-sampling heuristic method discussed in section 5.1 improves the results further. The relative error corresponding to the implicit particle filter with block sampling remains below 10% most of the time. The authors believe the fact that the decrease in average relative error is small as the number of particles increases is due to the fact that the posterior density is close to symmetric and is unimodal. The point estimates are taken as the mean of the posterior, which is approximated by the weighted average of the particles. In the case of implicit particle filter, we think that the average relative error will converge to a value smaller or equal to the one obtained from the MAP method, as the number of particles tends to infinity.

[68] The MAP estimation method proposed in section 5.2 is also implemented for both cases of sparse measurements and measurements available at every time step. Figure 10 compares the time evolution of the relative error of the implicit particle filter with block sampling with 50 particle and the MAP method for the case where measurements are available at every time step. As can be seen in this figure, the MAP method improves the relative error compared to the implicit particle filter almost at all times.

[69] We implement different estimation methods discussed in previous sections for different scenarios and to compare the performance of different methods in different cases, we calculate the average of the relative error per time step over the period of the experiment as follows:

$$\overline{E} = \sqrt{\frac{\sum_{k=1}^{T}\sum_{i=1}^{N_{\text{cells}}} \left(\hat{Q}_i^k - Q_i^k\right)^2}{\sum_{k=1}^{T}\sum_{i=1}^{N_{\text{cells}}} \left(Q_i^k\right)^2}}. \tag{58}$$

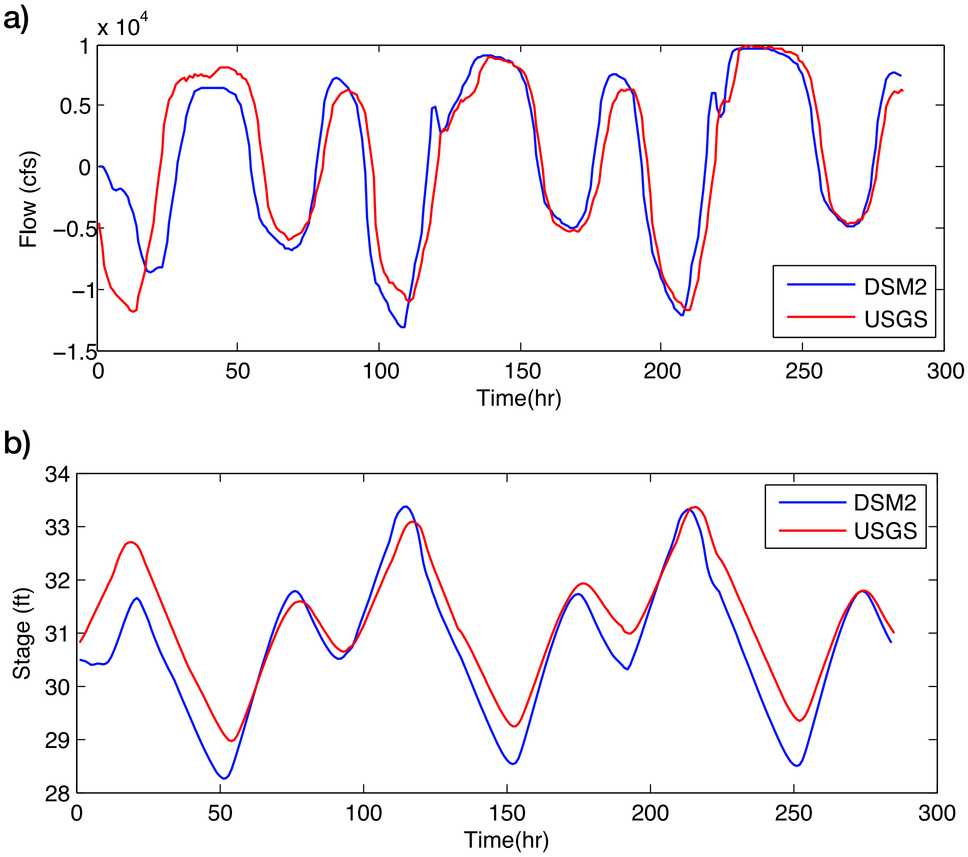

Figure 6. A comparison of the DSM2 prediction of (a) flow and (b) stage at node 4 in Figure 5 and USGS sensor measurements, corresponding to 10–12 June 2006.

[70] In the cases of SIR filter and the implicit particle filter, the relative error is a random variable due to the randomness in propagation of the particles. Therefore, we perform the assimilation 10 times for each scenario in these cases to calculate the mean and variance of the relative error.

[71] In Table 2, the average relative error is provided for the forward simulation, the optimal SIR, and implicit particle filters with block-sampling for a few different number of particles, as well as, the MAP method. As can be seen in this table, performing data assimilation using the optimal SIR filter reduces the average error of the model from about 23% to around 10%, which is a significant improvement. More interestingly, using the implicit particle filter with block sampling reduces the average error to about 8%. As discussed before, this is due to the fact that using more measurements before sampling results in more accurate samples, specially in cases where information propagates in space continuously with time. Also, note that increasing the number of particles improves the results slightly. Unfortunately, we were not able to run the methods for larger number of particles due to the required computational time. For instance, implementing the implicit particle filter with 50 particles for the period of the experiment on a 2.6 GHz dual-core desktop takes a few days. Finally, note that the MAP method improves the results further and reduces the average relative error to about 7%. It is remarkable that the MAP method provides the best estimate considering the fact that the computational cost of the MAP method is also less than other methods. In the MAP method, the minimization is solved only once at every time interval in contrast to once for every particle in the case of the implicit particle filter, and there is also no need to solve the sampling equation.

[72] Note that the particle filters even with one particle decrease the model error significantly by taking advantage of the six additional discharge measurements used for the data assimilation. In other words, when the particle filters are applied, extra discharge measurements are incorporated into the model even with one particle as opposed to the forward simulation, which only takes advantage of the boundary conditions. In the optimal SIR case, the importance density is chosen as $p\left(x_k|x_{k-1}^i, z_k\right)$, and the measurements not only affect the weight updates but also affect the importance density and hence the particles. In the case of the implicit particle filter, the measurements directly affect the location of the minimizer of $F_j(X)$.

[73] We also consider a case in which sparse measurements are available. We assume that measurements become available at every 10 time steps, and we implement the implicit particle filter and the MAP method to incorporate the sparse measurements into the model. Note that it is not necessary to assume the time interval between the measurements to be fixed, although the implementation is done with this assumption. Table 3 shows the relative error in the case of sparse measurements resulted from using the implicit particle filter and the MAP method. As can be seen

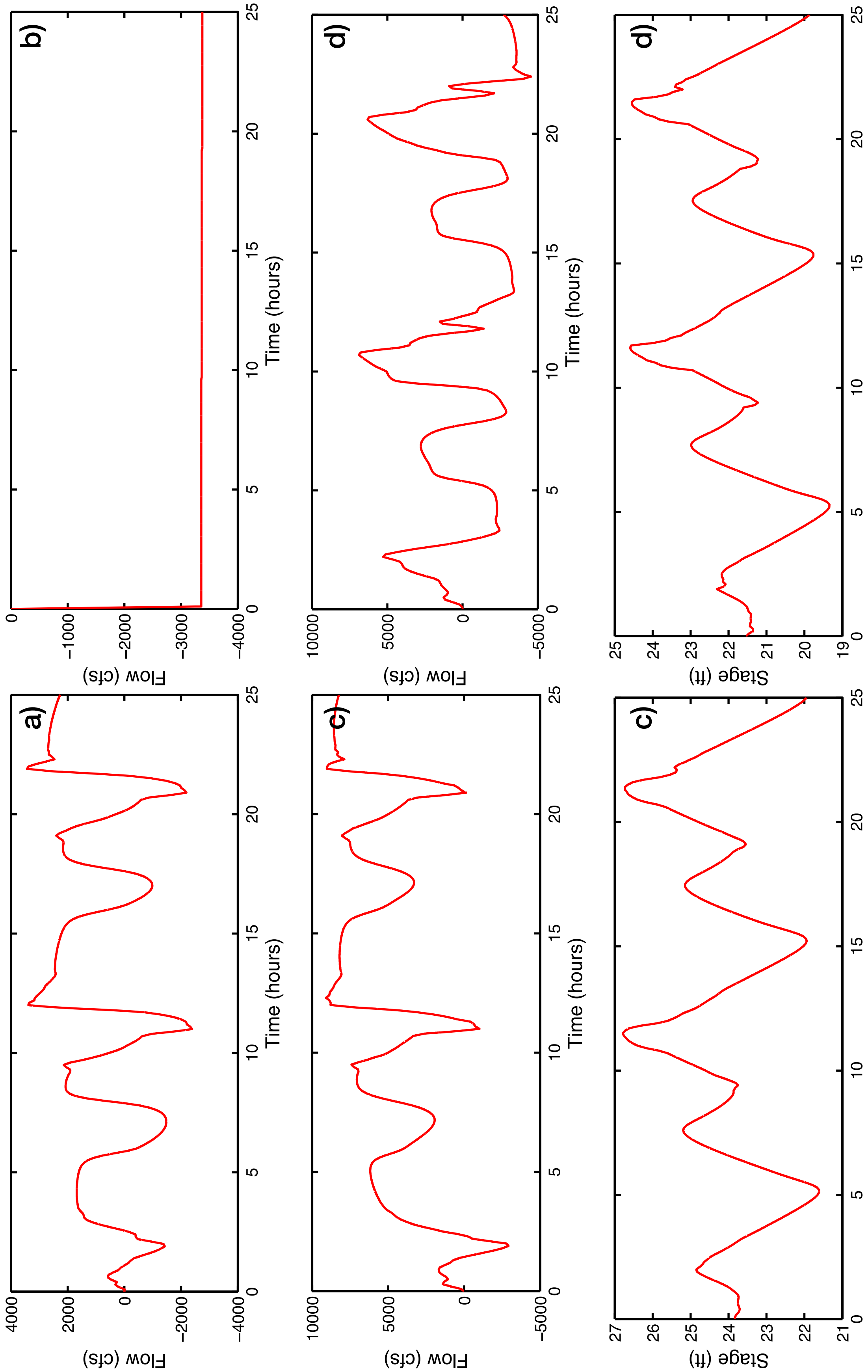

Figure 7. The boundary conditions, discharge at nodes (a) 16, (b) 15, (c) 17, and (d) 7, and stage at nodes (e) 4 and (f) 12.

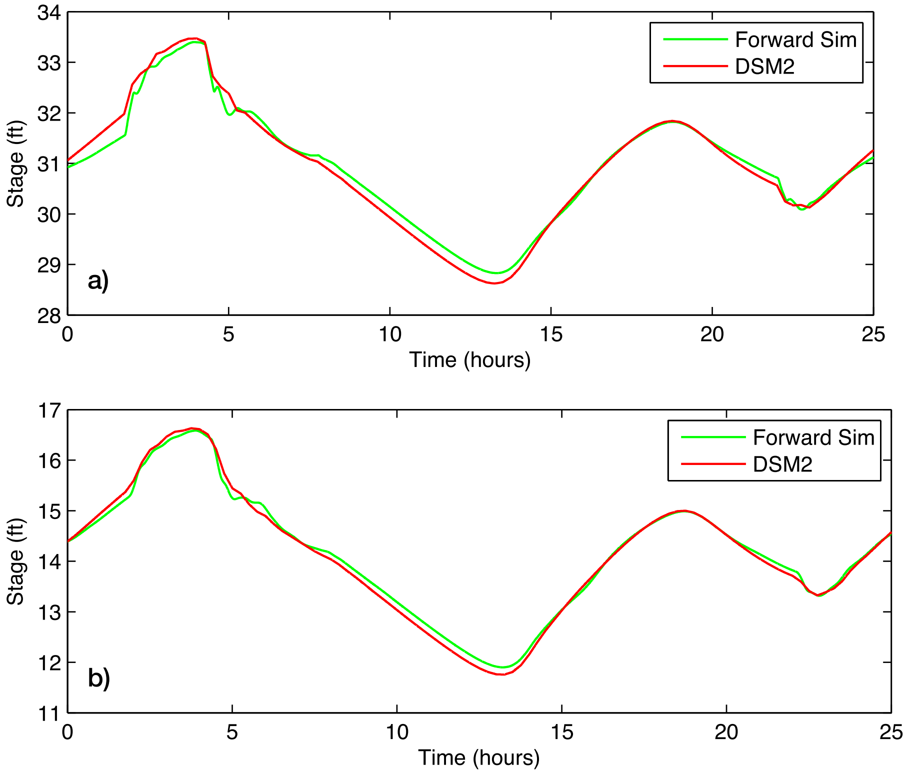

Figure 8. The stage computed by the forward simulation compared to DSM2 results at channels (a) 5–11 and (b) 8–9.

in this table, the MAP method provides a smaller average error than the implicit filter.

[74] Finally, in the case in which measurements are available at every time step, we investigate the effect of r, i.e., the length of the sampling time interval, on the performance of the methods for the case of the implicit particle filter with block sampling and the MAP method. Table 4 provides the relative error corresponding to four different values of r. Note that when r is taken to be 1, the implicit particle filter has the same error as the optimal SIR filter. This is consistent with the fact that, in the case of a linear observation model and Gaussian noise, the implicit particle filter recovers the optimal SIR filter. It is also interesting to see that $r = 5$ provides the least error in both cases. We think that this is for two reasons. First, the dimension of the problem increases with r. More precisely, the dimension of the state trajectory over the interval increases linearly with r. Therefore, larger inaccuracies will result from numerical processes, e.g., solving the minimization problem, as the dimension of the problem increases. Second, as r becomes large, some of the measurements used for the estimation of the state trajectory may become irrelevant, i.e., the measurements obtained at the end of the interval may not contain relevant information about the state at the beginning of the interval if the length of the interval, r, is too large.

[75] Finally, to compare the computational complexity of each of the methods discussed, we provide the average computational time per time step for each case, while implemented on a 2.6 GHz dual-core desktop computer, in Table 5. It should be emphasized that the programs are not currently written in an efficient way specially in the case of the implicit particle filter. As can be seen in this table, the computational time of the implicit filter with 10 particles is almost the same as that of the optimal SIR with 500 particles. Nonetheless, as seen in Table 2, the implicit filter with 10 particles produces an average error of about 8%, while the optimal SIR produces an average error of about 10%. Also, note that the MAP method has a lower average computational time than the implicit filter. In fact, it can be seen that the computational time of the MAP method is almost that of the implicit filter divided by the number of particles. This is because, in the MAP method, the minimization, which is the most costly step of the implicit filtering, is done once as opposed to once for every particle in the case of implicit filters.

[76] Note that, in all cases of particle filters, the algorithms can be easily parallelized by distributing the particle among the machines. Each machine can be responsible for the propagation of a certain number of particles, and as the number of machines increases, larger number of particles can be handled in the same amount of time, which can result in a better accuracy.

7. Conclusion

[77] We investigated the performance of different Monte Carlo methods applied for the estimation of water flow in an open channel network consisting of 19 subchannels and 1 reservoir in Sacramento-San Joaquin Delta in California. Starting from 1–D Saint-Venant equations, we constructed a state space model of the flow in the network of interest.

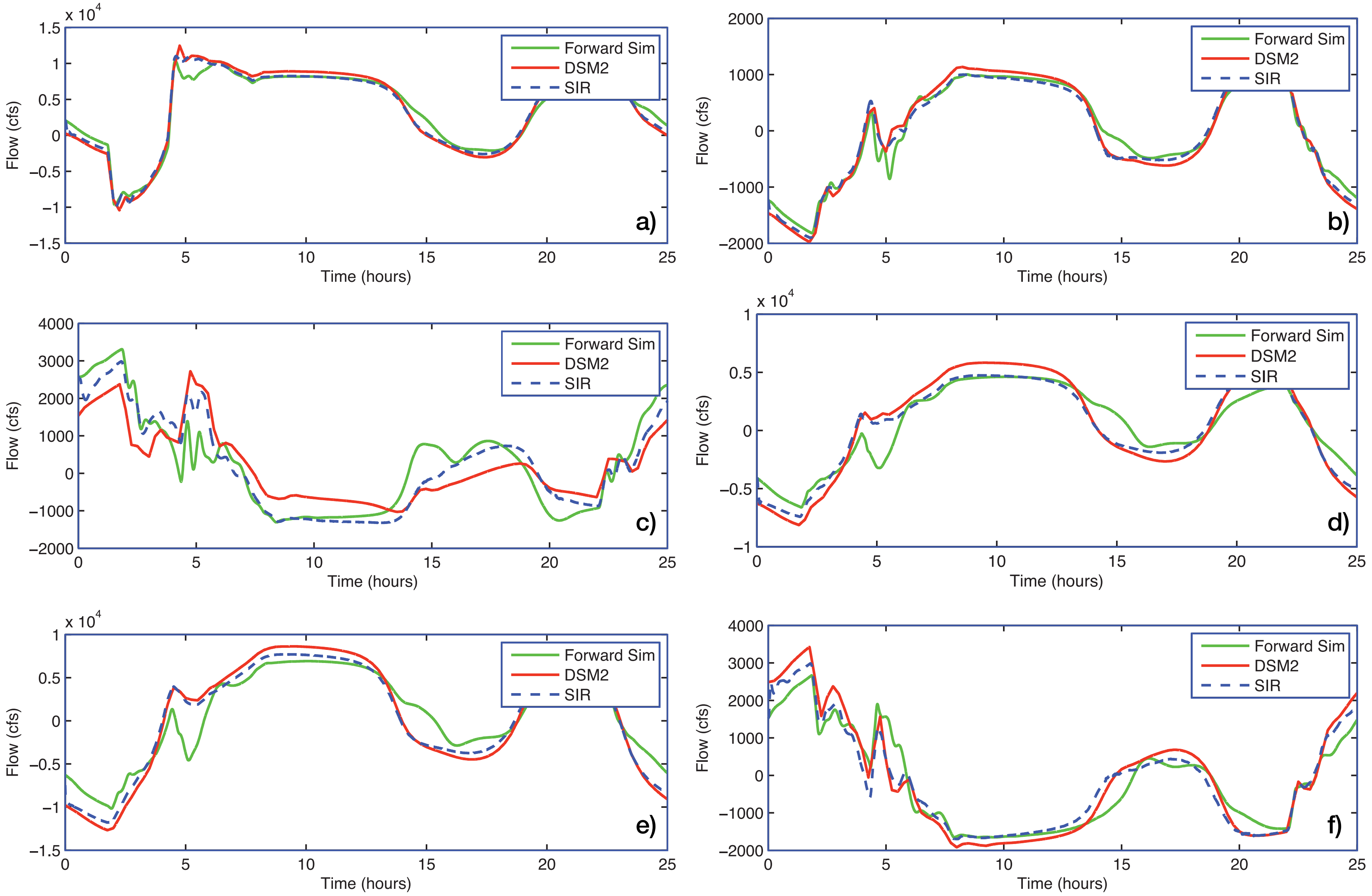

Figure 9. Time evolution of discharge at six channels in the network: (a) channels 11–13, (b) channels 8–9, (c) channels 5–6, (d) channels 3–5, (e) channels 3–4, and (f) channels 2–5, obtained from the forward simulation, the optimal SIR filter compared with the ground truth, i.e., DSM2 results.

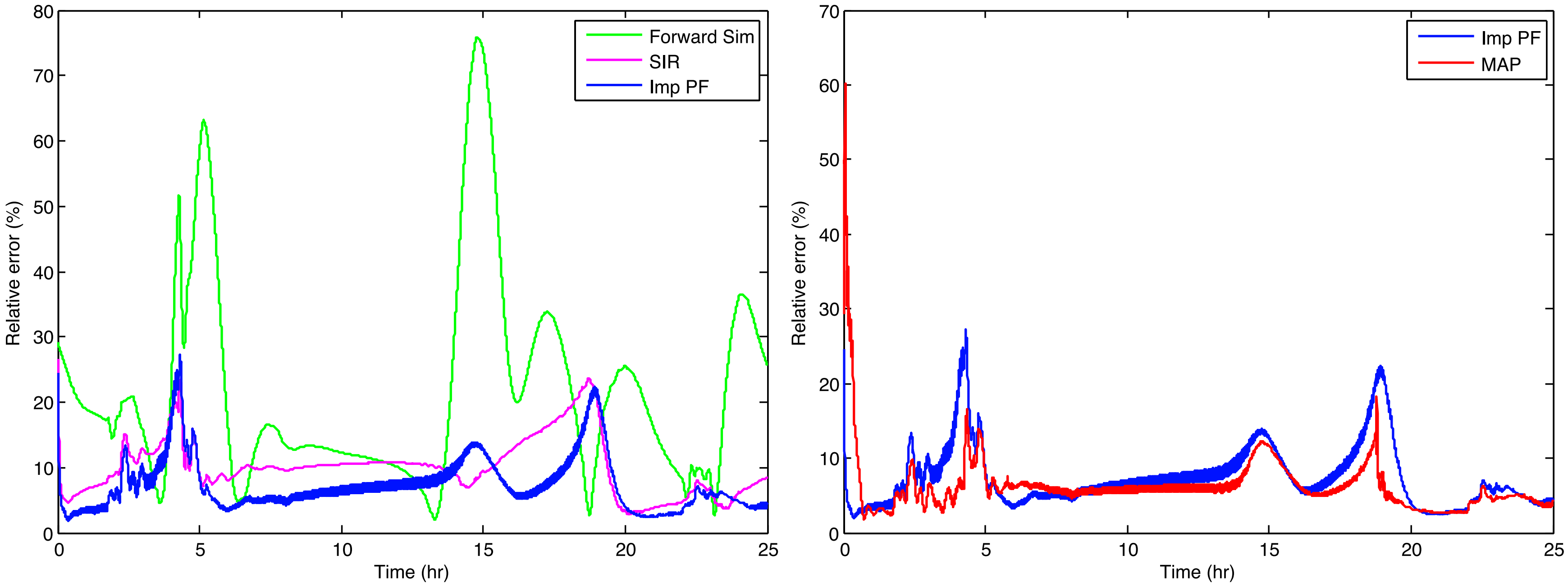

Figure 10. Time evolution of relative error for (a) forward simulation, optimal SIR, and implicit particle filter with block sampling and (b) the implicit particle filter with block sampling and the MAP method.

Table 2. Average Relative Error Corresponding to the Forward Simulation, the Optimal SIR, the Implicit Particle Filter with Block-Sampling (IMP PF with BS) for a Few Different Number of Particles, and the MAP Method, for a Case Where Measurements Are Available at Every Time Step

Method	Number of Particles	Error (%): Mean/Variance
Forward simulation	–	23.36
SIR	1	10.87/0.63
SIR	10	10.41/0.48
SIR	100	10.28/0.39
SIR	1000	10.15/0.30
IMP PF with BS	1	8.03/0.41
IMP PF with BS	10	7.74/0.26
IMP PF with BS	50	7.55/0.21
MAP	–	6.92

Considering a case in which measurements are available at every time step, we implemented the optimal SIR filter. We also considered a case in which sparse measurements are available, i.e., measurements become available at every r time steps, where r could be a random integer in general (although we considered a fixed r in this work). We implemented a random-map implementation of the implicit particle filter implemented to incorporate the sparse measurements into the model.

[78] In the case in which measurements are available at every time step, we proposed a heuristic method to obtain better estimates than the optimal SIR filter. It is known that, for systems with linear observation model and Gaussian noise when $r = 1$, the implicit particle filter is equivalent to the optimal SIR filter. However, in such a case, we implemented the implicit particle filter over time intervals of a desired length, and we incorporated all the measurements available within the interval to obtain an estimate of the state trajectory over the interval. It was seen through implementation that this implementation of implicit particle filter with block sampling provides a more accurate estimate than the optimal SIR. We think that this will be the case in systems with a band-diagonal dynamic structure, for which the value of each state component is determined by the value of its neighboring components at the previous time step. In such systems, the information propagates one cell in space per one time step. As a result, a measurement at a given location contains information about the value of state r components away r time steps ago. Therefore, performing estimation over time intervals using all the available measurements may create more accurate samples.

Table 3. Average Relative Error Corresponding to the IMP PF, for 10 and 50 Particles, and the MAP Method, for a Case in Which Measurements Are Available at Every 10 Time Steps

Method	Number of Particles	Error (%): Mean/Variance
IMP PF	1	14.31/0.47
IMP PF	10	13.88/0.32
IMP PF	50	13.68/0.27
MAP	–	11.72

Table 4. Average Relative Error Corresponding to the IMP PF with BS, for 10 Particles, and the MAP Method, for a Case in Which Measurements Are Available at Every Time Step, for Four Different Values of r

r	IMP PF with BS	MAP
1	10.89/0.48	10.38
5	7.52/0.22	6.20
10	7.74/0.26	6.92
20	9.83/0.35	8.96

[79] As a second heuristics, we posed the estimation problem as a MAP estimation in which the goal is to maximize the posterior density of the state trajectory over the interval, given all measurements. We believe that, in cases in which the posterior density is symmetric or close to symmetric, this method may provide more accurate estimates than the implicit filtering. By approximating the conditional density of the state at the beginning of the interval with a Gaussian density, we were able to solve the corresponding optimization problem with the same computational cost as the optimization problem in the case of implicit filters. It was observed that, for the case of estimation problem considered in this article, this method provides more accurate estimates than the implicit particle filter while having a much lower computational cost.

[80] We think the application presented in this article can help the practitioners to assess the utility of additional measurements of the system, which can be used in data assimilation such as the bones presented to improve the model outputs. It was seen that in the implementation considered in the current article, using six internal discharge measurements led to an improvement of the error from over 20% to less than 10%.

[81] The implicit particle filter and the proposed MAP method can be applied in many practical problems in geosciences and hydraulics. The traditional particle filters have been applied for different estimation problems in hydrology, for instance, to assimilate the SAR data into hydrologic models to estimate the water level [*Giustarini et al.*, 2011; *Madsen and Skotner*, 2005; *Matgen et al.*, 2010] and discharge [*Neal et al.*, 2009] in rivers and for flood modeling. We believe that the implicit particle filter can produce more accurate results in such applications. Applying the implicit particle filter to such problems to which the particle filters have been previously applied and performing a quantitative comparison between the two is a topic of future research.

Table 5. Average Computational Time of the Methods Per Time Step, for the Optimal SIR, the IMP PF with BS, and the MAP Method

Method	Number of Particles	Average Computation Time/Time Step (s)
SIR	10	0.14
SIR	50	0.54
SIR	500	5.16
IMP PF with BS	10	5.24
IMP PF with BS	50	25.57
MAP	–	0.55

Appendix A

[82] The nonzero terms of the Jacobian matrix of the state space function corresponding to interior grid points are as follows:

$$\frac{\partial Q_{i+1}^{k+1}}{\partial Q_i^k} = \frac{1}{2} + \frac{\Delta t}{\Delta x}\frac{Q_i^k}{\alpha_i T_i H_i^k} \tag{A1}$$

$$\frac{\partial Q_{i+1}^{k+1}}{\partial Q_{i-1}^k} = \frac{1}{2} + \frac{\Delta t}{\Delta x}\frac{Q_i^k}{\alpha_i T_i H_i^k} \tag{A2}$$

$$\frac{\partial H_{i+1}^{k+1}}{\partial Q_i^k} = \frac{\Delta t}{2\alpha_{i+1} T_{i+1} H_{i+1}^k \Delta x} \tag{A3}$$

$$\frac{\partial H_{i+1}^{k-1}}{\partial Q_i^k} = -\frac{\Delta t}{2\alpha_{i-1} T_{i-1} H_{i-1}^k \Delta x} \tag{A4}$$

$$\frac{\partial Q_{i+1}^{k+1}}{\partial H_i^k} = \frac{\Delta t}{2\Delta x}\left[\left(-\frac{Q^2}{\alpha T H}\right)_i^k + 2g\alpha_i T_i \gamma_i H_i^k\right] \tag{A5}$$

$$\frac{\partial Q_{i-1}^{k+1}}{\partial H_i^k} = -\frac{\Delta t}{2\Delta x}\left[\left(-\frac{Q^2}{\alpha T H}\right)_i^k + 2g\alpha_i T_i \gamma_i H_i^k\right] \tag{A6}$$

$$\frac{\partial H_{i+1}^{k+1}}{\partial H_i^k} = \frac{\alpha_i T_i}{2\alpha(i+1)T(i+1)} \tag{A7}$$

$$\frac{\partial H_{i-1}^{k+1}}{\partial H_i^k} = \frac{\alpha_i T_i}{2\alpha(i-1)T(i-1)} \tag{A8}$$

[83] The nonzero terms of the Hessian matrix of the state space function corresponding to the interior grid points are as follows:

$$\frac{\partial^2 Q_i^{k+1}}{\partial Q_{i-1}^k \partial H_{i-1}^k} = \frac{\Delta t}{\Delta x}\left(\frac{-Q_{i-1}^k}{\alpha_{i-1} T_{i-1} \left(H_{i-1}^k\right)^2}\right) \tag{A9}$$

$$\frac{\partial^2 Q_i^{k+1}}{\partial Q_{i+1}^k \partial H_{i+1}^k} = \frac{\Delta t}{\Delta x}\left(\frac{-Q_{i+1}^k}{\alpha_{i+1} T_{i+1} \left(H_{i+1}^k\right)^2}\right) \tag{A10}$$

$$\frac{\partial^2 Q_i^{k+1}}{\partial \left(Q_{i-1}^k\right)^2} = \frac{\Delta t}{\Delta x}\frac{1}{\alpha_{i-1} T_{i-1} H_{i-1}^k} \tag{A11}$$

$$\frac{\partial^2 Q_i^{k+1}}{\partial \left(Q_{i+1}^k\right)^2} = \frac{\Delta t}{\Delta x}\frac{1}{\alpha_{i+1} T_{i+1} H_{i+1}^k} \tag{A12}$$

$$\frac{\partial^2 Q_i^{k+1}}{\partial \left(H_{i-1}^k\right)^2} = \frac{\Delta t}{\Delta x}\left(\frac{\left(Q_{i-1}^k\right)^2}{\alpha_{i-1} T_{i-1} \left(H_{i-1}^k\right)^3} + g\alpha_{i-1}\gamma_{i-1} T_{i-1}\right) \tag{A13}$$

$$\frac{\partial^2 Q_i^{k+1}}{\partial \left(H_{i+1}^k\right)^2} = \frac{\Delta t}{\Delta x}\left(\frac{\left(Q_{i+1}^k\right)^2}{\alpha_{i+1} T_{i+1} \left(H_{i+1}^k\right)^3} + g\alpha_{i+1}\gamma_{i+1} T_{i+1}\right) \tag{A14}$$

[84] **Acknowledgments.** The authors thank Alexandre Chorin from the Mathematics Department at UC Berkeley for his guidance and help through the project and for reviewing this article and giving us feedback. The authors also thank Matthias Morzfeld from the Mathematics Department at Lawrence Berkeley National Laboratory (LBNL) for fruitful discussions about the implicit particle filter and his useful comments about this article. The authors are grateful to Eli Ateljevich with the DWR for helping us with DSM2 from which the data of simulations were extracted.

References

Anderson, B., and J. Moore. (1979), *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, N. J.

Andreadis, K. M., E. A. Clark, D. P. Lettenmaier, and D. E. Alsdorf (2007), Prospects for river discharge and depth estimation through assimilation of swath-altimetry into a raster-based hydrodynamics model, *Geophys. Res. Lett.*, *34*, L10403, doi:10.1029/2007GL029721.

Arulampalam, M., S. Maskell, N. Gordon, and T. Clapp (2002), A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Trans. Signal Process.*, *50*(2), 174–188.

Bergman, N. (1999), Recursive Bayesian estimation: Navigation and tracking applications, Ph.D. dissertation, Linkoping Univ., Linkoping, Sweden.

Biancamaria, S., M. Durand, K. Andreadis, P. Bates, A. Booneg, N. Mognard, E. Rodriguez, D. Alsdorf, D. Lettenmaier, and E. Clark. (2010), Assimilation of virtual wide swath altimetry to improve arctic river modeling, *Remote Sens. Environ.*, *115*(2), 373–381, doi:10.1016/j.rse.2010.09.008.

Brasseur, P., and J. Nihoul (1994), Data assimilation: Tools for modelling the ocean in a global change perspective, in *NATO ASI Series, Series I: Global Environmental Change*, vol. 19, edited by P. Brasseur and J. Nihoul, p. 239, Springer, Berlin, New York.

Carpenter, J., P. Clifford, and P. Fearnhead (1999), Improved particle filter for nonlinear problems, *Proc. Inst. Electr. Eng. Radar Sonar Navig.*, *146*, 2–7.

Castaings, W., D. Dartus, M. Honnorat, F. Le Dimet, Y. Loukili, and J. Monnier (2006), Automatic differentiation: A tool for variational data assimilation and adjoint sensitivity analysis for flood modeling, *Springer Lect. Notes Comput. Sci. Eng.*, *50*, 249–262.

Chaudhry, M. H. (2008), *Open-Channel Flow*, Springer, New York.

Chorin, A., and X. Tu. (2009), Implicit sampling for particle filters, *Proc. Natl. Acad. Sci. U. S. A.*, *106*(41), 17,249–17,254.

Chorin, A., M. Morzfeld, and X. Tu (2010), Implicit particle filters for data assimilation, *Comm. Appl. Math. Comp.*, *5*(2), 221–240.

Chow, V. (1988), *Open-Channel Hydraulics*, McGraw-Hill, New York.

Cunge, J., F. Holly, and A. Verwey (1980), *Practical Aspects of Computational River Hydraulics*, Pitman. London.

Del Moral, P. (1998), Measure valued processes and interacting particle systems, Application to nonlinear filtering problems, *Ann. Appl. Probab.*, *8*, 438–495.

Doucet, A., S. Godsil, and C. Andrieu (2000), On sequential Monte Carlo methods for Bayesian filtering, *Stat. Comput.*, *10*, 197–208.

Doucet, A., N. de Freitas, and N. Gordon (Eds.) (2001), *Sequential Monte Carlo Methods in Practice*, vol. 1, Springer-Verlag, New York.

Durand, M., K. Andreadis, D. Alsdorf, D. Lettenmaier, D. Moller, and M. Wilson. (2008), Estimation of bathymetric depth and slope from data assimilation of swath altimetry into a hydrodynamic model, *Geophys. Res. Lett.*, *35*, L20401, doi:10.1029/2008GL034150.

Evensen, G. (2009), *Data Assimilation: The Ensemble Kalman Filter*, Springer-Verlag, Berlin Heidelberg.

Giustarini, L., et al. (2011), Assimilating SAR-derived water level data into a flood model: A case study, *Hydrol. Earth Syst. Sci.*, *15*, 2349–2365.

Gordon, N. J., D. J. Salmond, A. F. M. Smith, and T. Clapp (1993), Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *Radar Signal Process. IEE Proc. F*, *140*(2), 107–113.

Haidvogel, D., and A. Robinson (1989), Special issue on data assimilation, *Dyn. Atmos. Oceans*, *13*, 171–517.

Haykin, S. S. (2001), *Kalman Filtering and Neural Networks*, John Wiley & Sons Inc., New York.

Ishikawa, Y., T. Awaji, K. Akitomo, and B. Qiu (1996), Successive correction of the mean sea surface height by the simultaneous assimilation of drifting buoy and altimetric data, *J. Phys. Oceanogr.*, *26*, 2381–2397.

Julier, S. J., and J. K. Uhlmann (1997), A new extension of the Kalman filter to nonlinear systems, in *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulations and Controls*, International Society for Optics and Photonics, *AeroSense'97*, pp. 182–193.

Kaipio, J., and E. Somersalo (2004), *Statistical and Computational Inverse Problems*, vol. 160, Springer, New York.

Kalman, R. (1960), A new approach to linear filtering and prediction problems, *J. Basic Eng.*, *82*, 35–45.

Kalnay, E. (2003), *Atmospheric Modeling, Data Assimilation and Predictability*, Cambridge Univ. Press, Cambridge.

Kitagawa, G. (1996), Monte Carlo filter and smoother for non-Gaussian non-linear state space models, *J. Comput. Graph. Stat.*, *5*(1), 1–25.

Kuznetsov, L., K. Ide, and C. Jones. (2003), A method for assimilation of Lagrangian data, *Mon. Weather. Rev.*, *131*(10), 2247–2260.

Le Dimet, F., and O. Talagrand (1986), Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects, *Dyn. Meteorol. Oceanogr.*, *38*, 97–110.

Litrico, X., and V. Fromion (2009), *Modeling and Control of Hydrosystems*, Springer, London.

Liu, D. (2009), Uncertainty quantification with shallow water equations, Ph.D. dissertation, Univ. of Braunschweig, Braunschweig, Germany.

Liu, S., and R. Chen (1998), Sequential Monte Carlo methods for dynamic systems, *J. Am. Stat. Assoc.*, *93*(443), 1032–1044.

Lund, J., E. Hanak, W. Fleenor, R. Howitt, J. Mount, and P. Moyle (2007), *Envisioning Futures for the Sacramento-San Joaquin Delta*.

Madsen, H., and C. Skotner (2005), Adaptive state updating in real-time river flow forecasting—A combined filtering and error forecasting procedure, *J. Hydrol.*, *308*, 302–312.

Malanotte-Rizzoli, P. (Ed.) (1996), *Modern Approaches to Data Assimilation in Ocean Modeling, Elsevier Oceanography Series*, Elsevier, Amsterdam.

Matgen, P., M. Montanari, R. Hostache, L. Pfister, L. Hoffmann, D. Plaza, V. Pauwels, G. De Lannoy, R. De Keyser, and H. Savenije (2010), Towards the sequential assimilation of SAR-derived water stages into hydraulic models using the particle filter: Proof of concept, *Hydrol. Earth Syst. Sci.*, *14*, 1773–1785.

Molcard, A., L. Piterbarg, A. Griffa, T. Ozgokmen, and A. Mariano (2003), Assimilation of drifter observations for the reconstruction of the Eulerian circulation field, *J. Geophys. Res.*, *108*(C3), 3056, doi:10.1029/2001JC001240.

Moradkhani, H., K. Hsu, H. Gupta, and S. Sorooshian (2005), Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter, *Water Resour. Res.*, *41*, W05012, doi:10.1029/2004WR003604.

Morzfeld, M., X. Tu, E. Atkins, and A. Chorin (2012), A random map implementation of implicit filters, *J. Comput. Phys.*, *231*, 2049–2066.

Navon, I. (1997), Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography, *Dyn. Atmos. Oceans*, *27*, 55–79.

Neal, J., P. Atkinson, and C. Hutton (2007), Flood inundation model updating using an ensemble Kalman filter and spatially distributed measurements, *J. Hydrol.*, *336*, 401–415.

Neal, J., G. Schumann, P. Bates, W. Buytaert, P. Matgen, and F. Pappenberger (2009), A data assimilation approach to discharge estimation from space, *Hydrol. Process.*, *23*, 3641–3649.

Paniconi, C., M. Marrocu, M. Putti, and M. Verbunt (1996), Newtonian nudging for a Richards equation-based distributed hydrological model, *Adv. Water. Resour.*, *26*(2), 161–178.

Strum, T. (2001), *Open Channel Hydraulics*, McGraw-Hill, New York.

Tossavainen, O.-P., J. Percelay, A. Tinka, Q. Wu, and A. Bayen (2008), Ensemble Kalman filter based state estimation in 2D shallow water equations using Lagrangian sensing and state augmentation, in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 1783–1790, Cancun, Mexico.

Tremolet, Y. (2006), Accounting for an imperfect model in 4D-Var, *Q. J. R. Meteorol. Soc.*, *102*, 2483–2504.

Tremolet, Y. (2007), Model error estimation in 4D-Var, *Q. J. R. Meteorol. Soc.*, *133*, 1267–1280.

van Leeuwen, P. (2010), Nonlinear data assimilation in geosciences: An extremely efficient particle filter, *Q. J. R. Meteorol. Soc.*, *136*, 1991–1999.

Wan, E., and R. van der Merwe. (2000), The unscented Kalman filter for nonlinear estimation, in *Proceedings of the IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*, Lake Louise, Alberta, Canada.

Wang, D., Y. Chen, and X. Cai (2009), State and parameter estimation of hydrologic models using the constrained ensemble Kalman filter, *Water Resour. Res.*, *45*, W11416, doi:10.1029/2008WR007401.

Weare, J. (2009), Particle filtering with path sampling and an application to a bimodal ocean current model, *J. Comput. Phys.*, *228*, 4312–4331.

Wu, Q., M. Rafiee, A. Tinka, I. Strub, and A. Bayen (2009), Inverse estimation of open boundary conditions in channel network via quadratic programming, in *Proceedings of the 48th IEEE Conference on Decision and Control*, IEEE, Shanghai, China.