

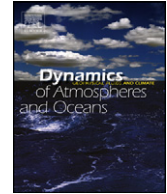


ELSEVIER

Contents lists available at ScienceDirect

Dynamics of Atmospheres and Oceans

journal homepage: www.elsevier.com/locate/dynatmoce



Fast data assimilation using a nonlinear Kalman filter and a model surrogate: An application to the Columbia River estuary

Sergey Frolov^{a,*,1}, António M. Baptista^a, Todd K. Leen^b,
Zhegdong Lu^b, Rudolph van der Merwe^{b,2}

^a NSF Science and Technology Center for Coastal Margin Observation and Prediction, Oregon Health & Science University, Beaverton, Oregon 97006, United States

^b Department of Computer Science and Electrical Engineering, Oregon Health & Science University, Beaverton, Oregon 97006, United States

ARTICLE INFO

Article history:

Available online 6 November 2008

Keywords:

Data assimilation
Columbia River estuary
Coastal margin circulation
Unstructured grid modeling

ABSTRACT

A fast and adjoint-free nonlinear data assimilation (DA) system was developed to simulate 3D baroclinic circulation in estuaries, leveraging two recently developed technologies: (1) a nonlinear model surrogate that executes forward simulation three orders of magnitude faster than a forward numerical circulation code and (2) a nonlinear extension to the reduced-dimension Kalman filter that estimates the state of the model surrogate. The noise sources in the Kalman filter were calibrated using empirical cross-validation and accounted for errors in model and model forcing.

The DA system was applied to assimilate *in situ* measurements of water levels, salinities, and temperatures in simulations of the Columbia River estuary. To validate the DA results, we used a combination of cross-validation studies, process-oriented studies, and tests of statistical and dynamical consistency. The validation studies showed that DA improved the representation of several important processes in the estuary, including nonlinear tidal propagation, salinity intrusion, estuarine residual circulation, heat balance, and response of the estuary to coastal winds.

© 2008 Elsevier B.V. All rights reserved.

* Corresponding author. Tel.: +1 831 775 1960; fax: +1 831 775 1646.

E-mail address: frolovs@mbari.org (S. Frolov).

¹ Now with Monterey Bay Aquarium Research Institute.

² Now with Advance Computation Group of Apple Inc®.

Nomenclature

k	index of the discrete time
$x(k)$	state vector in the full space
$u(k)$	vector of forcings
$y(k)$	vector of observations
$x_s(k)$	state vector in the reduced space
$\tilde{x}^{\Pi}(k)$	reconstruction of the vector $x_s(k)$ in the full space S
$\tilde{x}_s^{ms}(k)$	predictions of the vector $x_s(k)$ using model surrogate M^{ms}
$\hat{x}_s^f(k)$	optimal prediction of the state at time k conditioned on the observations up to time $k - 1$ (state forecast)
$\hat{x}_s^a(k)$	estimate of the reduced state at time k conditioned on the observations up to time k (state analysis)
$\hat{y}(k)$	optimal prediction of the observation at time k
X_k	time-lagged input to the model surrogate at time k (vector of past states)
U_k	time-lagged forcings of the model surrogate at time k (vector of past and future forcings)
Y_k	time-lagged output of the model surrogate at time k (vector of future states)
$M(\cdot)$	nonlinear dynamical model in the full space
$H(\cdot)$	nonlinear observation function in the full space
$M^s(\cdot)$	nonlinear model of the dynamics in the reduced space
$H^s(\cdot)$	nonlinear observation function for the reduced state x_s
$M^{ms}(\cdot)$	nonlinear model surrogate
$\nu, \nu_{ui}, \nu_x, \nu_{ms}, \text{ and } \nu_{xms}$	process, forcing, model, model surrogate, and combined model and model surrogate errors; and $Q_{ui}, Q_x, Q_{ms}, \text{ and } Q_{xms}$ are the respective error covariance matrixes
n_m	measurement (sensor) errors
n_o	vector of observation errors for the reduced space observation function $H^s(\cdot)$
S	full space \mathbb{R}^l of state vectors $x(k)$
S^{nl}	low-dimensional, possibly nonlinear, manifold of model trajectories
S^s	reduced space
\prod	EOF dimension-reduction operator
ε	reconstruction error
μ	mean of the state vector
σ_{var}	normalization constant for the components of the state vector
$E[\cdot]$	expectation operator
W	concatenated vector of unknown weights and bias terms in the neural network
λ	regularization parameter of the neural network
T_i	internal PCA operator of the neural network
P_{xx}^f	error covariance matrix for the state \hat{x}_s^f (forecast error covariance)
P_{xx}^a	error covariance matrix for the state estimate \hat{x}_s^a (analysis error covariance)
P_{yy}	covariance of innovation ($y(k) - \hat{y}^f(k)$)
K_k	Kalman gain at time k
$\alpha_u \text{ and } \alpha_x$	inflation factors for the forcing and model error covariance Q_u and Q_{xms}
$\hat{\sigma}_{\text{inov}}^2 \text{ and } \sigma_{\text{inov}}^2$	predicted and measured variance of the forecast error

1. Introduction

The future of oceanography is arguably being re-written through the development of modern observatories (Baptista, 2002; Clark and Isern, 2003; Martin, 2003), versions of which are being planned and implemented across the world's oceans and coastal margins. An increasing number of advanced observatories (ORION, 2005; Baptista, 2006) are conceptualized as end-to-end multi-purpose sys-

tems, integrating real-time observational networks, operational modeling systems, and integrative cyber-infrastructure.

In this context, the fusion of observations and simulations via data assimilation (DA) has enormous potential as a core technology for ocean observatories. However, to be able to fulfill this potential, DA methods must meet the operational needs of observatories. We envision the next generation of DA methods to be computationally fast, available in adjoint-free formulation, and adept at handling strong nonlinearities. To address the need for such DA methods, we recently developed the reduced-dimension Kalman filter (KF) (Lu et al., [submitted for publication](#))—a fast nonlinear extension to the KF. The computational efficiency of the new method comes, in part, from the use of neural network model surrogates (van der Merwe et al., 2007) that execute forward simulations three to four orders of magnitude faster than forward numerical circulation codes. This paper reports on the first realistic application of the new DA method to the 3D baroclinic circulation of an estuary, that of the Columbia River.

Process-wise, the Columbia River is an excellent testbed for developing and testing DA methods for coastal margin observatories. In particular, the well-documented nonlinear dynamics of the Columbia River estuary (Hamilton, 1990; Jay and Smith, 1990; Baptista et al., 2005; Chawla et al., 2008) are complex and likely to challenge assumptions at the core of a DA algorithm. An example is salinity intrusion, the length of which is determined by the balance of tidal mixing and the difference in density between ocean and river waters. Salinity intrusion is modulated by variations in astronomic tides, non-stationary river discharge, and coastal winds; its representation requires a nonlinear 3D baroclinic circulation model and knowledge of diverse external forcings (river discharge, tides, coastal winds, and ocean properties).

The infrastructure of an established coastal-margin observatory, CORIE, offers the support needed to the use of the Columbia River as a testbed. CORIE includes a real-time observation network, a modeling system for 3D baroclinic circulation in the Columbia River estuary and plume (Baptista et al., 2005), and integrative cyber-infrastructure. CORIE observational data and modeling products (Baptista et al., 2005; Baptista, 2006) are used to support the research and development of novel modeling techniques (Zhang et al., 2004; Zhang and Baptista, 2008), fisheries research (Bottom et al., 2005; Burla et al., 2007), and ecosystem management (USACE, 2001). The modeling products include both multi-annual simulation databases and real-time forecasts of 3D baroclinic circulation, and are produced by either of two unstructured-grid forward models: SELFE (Zhang and Baptista, 2008), a finite-element Eulerian–Lagrangian model, or ELCIRC (Zhang et al., 2004), SELFE's finite-volume predecessor. While CORIE simulations have achieved a useful representation of many Columbia River processes, DA is expected to further improve their skills by allowing DA to correct for remaining errors and uncertainties (both in external forcings and in numerical algorithms and resolution).

2. Data assimilation: context and goals

A plethora of DA algorithms is available in oceanography, with most falling into two broad categories: sequential and variational. The sequential algorithms are based on the KF (Kalman, 1960)—a two-step recursion that first forecasts a model state and error statistics and then estimates the best fit of the model state to measurements, using the least squares criterion. The direct implementation of the KF recursion is out of computational reach for high-resolution 3D nonlinear models of coastal ocean systems, principally due to the prohibitive computational cost of forecasting error statistics. As a result, a number of sub-optimal KF-solutions have emerged (Gandin, 1963; Cane et al., 1996; Lermusiaux and Robinson, 1999; Pham, 2001; Oke et al., 2002; Evensen, 2003).

One such sub-optimal solution is the ensemble KF (Evensen, 2003), which approximates the computationally expensive forecast of error-statistics using a limited (~ 100) ensemble of model states. Another sub-optimal solution is the optimal interpolation (Gandin, 1963), which approximates error statistics using statistical models that are stationary in time and/or space. In contrast to the sequential DA methods, the variational DA methods are based on solutions to Euler–Lagrange equations, where the forward model is used as a dynamic constraint. The leading variational algorithms include the 3DVAR (Le Dimet and Talagrand, 1986), the strong-constrain 4DVAR (Le Dimet and Talagrand, 1986),

and the representer method (Bennett, 1992). A variety of DA algorithms have been successfully applied to assimilate measurements into simulations of the coastal ocean (Martin, 2000; Heemink et al., 2002; Kurapov et al., 2002; Oke et al., 2002; Wilkin et al., 2005) and, to a lesser extent, of estuaries (Xu et al., 2002; Bertino et al., 2002).

However, the existing DA algorithms are not likely to satisfy the needs of coastal-margin observatories. Taking CORIE as a reference, the advanced DA algorithms are too computationally expensive to implement in the context of multi-annual hindcasts and real-time forecasts of the Columbia River estuary. Implementing DA algorithms that allow for time-varying error statistics (such as the ensemble KF, 4DVAR, and representer methods) may require a 100-fold increase in the computational resources available within CORIE. Also, the personnel costs of implementing variational methods (such as 4DVAR and representer methods) are very high, since these methods require development of a tangent-linear model and an adjoint model that are not available for emerging models such as SELFE. In contrast, while the computational and personnel costs of implementing optimal interpolation methods are relatively low, the stationary approximation of the error covariance in these methods is likely to be inadequate for modeling the statistics of a system as dynamic as the Columbia River estuary.

To overcome the limitations of existing DA algorithms, we have recently developed a number of enabling technologies (van der Merwe et al., 2007; Lu et al., submitted for publication) that rely on model surrogates to expedite the KF computations. A model surrogate (van der Merwe et al., 2007) is a very fast, nonlinear neural network that is trained to approximate a slower forward model. For typical CORIE simulations, we estimate that the model surrogate is 1000–10,000 faster than the forward model (van der Merwe et al., 2007). Because it is not computationally feasible to train the model surrogate in the high-dimensional space of the forward model, the surrogate is trained in the subspace spanned by the empirical orthogonal functions (EOF) of the forward model. In previous work (van der Merwe et al., 2007), we developed algorithms for training model surrogates that were able to successfully approximate a month-long simulation of the Columbia River estuary and plume.

To implement the KF recursion, we use a reduced-dimension KF (Lu et al., submitted for publication). Unlike similar filters (Cane et al., 1996; Hoteit and Pham, 2003) that used simplified models to propagate error statistics and the original forward model to propagate the central forecast, the reduced-dimension KF assimilates the observations directly into the model surrogate. The surrogates obviate the need to use the outputs of (much slower) forward models in the reduced-dimension KF. Because the state of the model surrogate is usually small (<100 degrees of freedom), it is possible to implement the reduced-dimension KF using a full-rank KF. In our experiments, we used sigma point KF (van der Merwe and Wan, 2003; van der Merwe, 2004), a nonlinear extension to the classical KF. The extension is based on using an ensemble of deterministically sampled model states (sigma points) that sample the covariance of the model, the forcing, and the observational errors. We demonstrated earlier (Lu et al., submitted for publication) the application of the reduced-dimension KF to a synthetic estuary, successfully estimating both the state of the synthetic estuary and the time-varying errors in forcing.

It is important to note that our reduced-dimension KF has similarities with existing reduced-space KF methods, such as the ensemble KF (Evensen, 2003), the singular evolutive interpolated KF (Pham, 2001), and the reduced-order extended KF (Cane et al., 1996). In particular, all of these methods assume that a model trajectory lies on a manifold of a lower dimension than the state-space of the numerical model. However, most existing methods use this assumption to justify the rank reduction for the error covariance, while preserving the complexity of the forward model (e.g., Pham et al., 1998; Evensen, 2003). In contrast, in our reduced-dimension KF framework we approximate the forward model with a model surrogate that already operates in the EOF space of a reduced dimension, which restricts the span of the error covariance to the same EOF subspace. It is to highlight this difference and to distinguish our KF framework from the existing reduced-space KFs, that we use the term reduced-dimension KF.

We report here on the first realistic DA application to the 3D baroclinic circulation in the Columbia River estuary, using model surrogates (van der Merwe et al., 2007) and the reduced-dimension KF (Lu et al., submitted for publication). The goal was a realistic year-long hindcast simulation. We assimilated *in situ* measurements of water levels, salinities, and temperatures into model surrogates, which we trained using 6 years of the high-resolution hindcast CORIE simulation databases. Our experiments showed that both an EOF and a model surrogate can be trained to perform well even outside the training period, thus allowing their application on unseen data. Our validation studies showed that

DA improves upon the non-assimilative model in representing several important processes in the Columbia River estuary, such as nonlinear tides, responses of the estuary to coastal winds, and salinity intrusion.

3. The Columbia River testbed

3.1. Dominant physical process in the Columbia River estuary

The Columbia River estuary, one of the world's classic river-dominated estuaries, is a highly energetic and dynamic system that responds quickly to changes in external forcing. We briefly review here three dominant physical processes in the estuary: tidal propagation, salinity intrusion, and response of estuarine temperatures to forcings.

Columbia River tides are driven by nonlinear interactions of astronomic tides with complex bathymetry and non-stationary river discharge (Jay et al., 1997). Tidal amplitudes during spring tides can reach 3.6 m at station *tpoint* (Fig. 1a). In addition, water levels are subject to oceanic sub-tidal influences, with seasonal and shorter-term fluctuations of as much as 1 m (Fig. 1b).

Compressed and often stratified, estuarine circulation is subject to extreme variations in salinity intrusion and stratification regimes (Hamilton, 1990; Jay and Smith, 1990; Baptista et al., 2005; Chawla et al., 2008). Fig. 1 shows the minimum (panel g) and the maximum (panel f) extent of the bottom salinity intrusion during the period of the study. The length of salinity intrusion (shown in Fig. 1c and defined along the navigation channel of the estuary) responds primarily to changes in river discharge and to the strength of the tidal mixing, but also to factors such as ocean upwelling. Salinity intrusion is the highest when river discharge is low, tidal mixing is low (neap tides), and ocean upwelling is strong.

Temperature variability in the estuary is determined by the mixing of two end-members: ocean waters with temperatures varying from $\sim 10^{\circ}\text{C}$ to $\sim 15^{\circ}\text{C}$ and river waters with temperature varying seasonally from $\sim 5^{\circ}\text{C}$ to $\sim 25^{\circ}\text{C}$. The seasonal trend in river temperature is present throughout the entire estuary, and can be seen in the time series of observed temperature at station *jetta* (Fig. 1d.) Shelf winds (through upwelling and downwelling regimes) combine with seasonality to determine the temperature of the oceanic sources to the estuary. During winter, temperatures stay consistently at

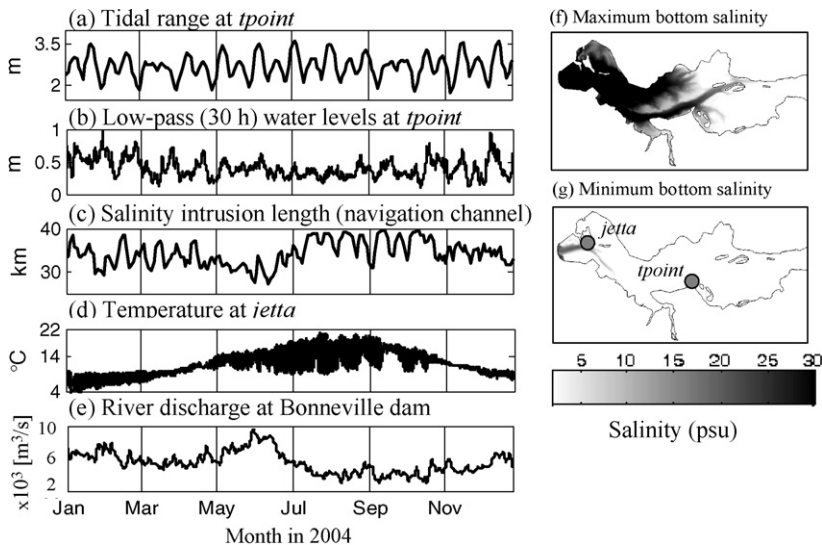


Fig. 1. Tidal variability and salinity intrusion in the Columbia River estuary. (a) Observed tidal range at station *tpoint*. (b) Observed low-pass (>30 h) elevations at station *tpoint*. (c) Simulated (DB16) salinity intrusion length. (d) Observed temperature at *jetta*. (e) Observed river discharge at Bonneville dam. (f) Simulated (DB16) maximum bottom salinity for the year. (g) Simulated (DB16) minimum bottom salinity for the year and location of stations *tpoint* and *jetta*.

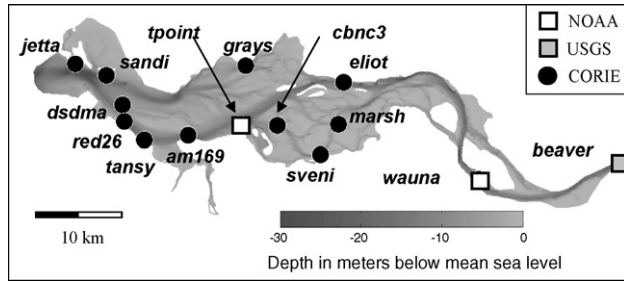


Fig. 2. Map of the bathymetry in the computational domain. Overlaid are the names and locations of the observation stations in our study.

$\sim 10^{\circ}\text{C}$. However, during summer the temperatures of the oceanic sources range from $\sim 10^{\circ}\text{C}$ (during prevailing upwelling regimes) to $\sim 15^{\circ}\text{C}$ (during downwelling events).

3.2. Observational data

In our DA experiments (Section 6), we assimilated measurements of water level, salinity, and temperature from 34 distinct sensors at the 13 stations within the estuary (Fig. 2). Most stations are part of the CORIE observation network; the exceptions are the tidal stations *tpoint* and *wauna*, which are operated by National Atmospheric and Oceanic Administration.³ Each CORIE station measures a variable combination of parameters, including water level, salinity, temperature, velocity profile, air temperature, and wind speed. Scalar sensors at most stations are deployed at a single depth, usually near the bottom. Exceptions are *red26* and *am169*, where measurements of salinity and temperature are available near bottom, mid-water and near surface. In addition, acoustic Doppler profilers are available at *red26*, *tansy*, and *am169*.

During the period of our study, there was a significant data loss in the observation network due to a combination of bio-fouling, telemetry drop-outs, and gaps in the deployment schedule of sensors. Of the 34 possible sensors, an average of only 23 sensors (70%) were available at any given time.

3.3. Simulation databases

A product of the CORIE modeling system is a set of multi-year simulation databases for the estuary and plume (Baptista et al., 2005). These simulations are conducted retrospectively, and their skill is carefully evaluated. At the time of our experiments, the most accurate simulation database for the estuary was DB16. Notwithstanding uncertainties in the boundary conditions and some persistent model errors, DB16 offers a realistic representation of the estuarine circulation and salinity propagation. We used the results of DB16 to compute the EOFs and to train the model surrogates described in Sections 5 and 6. Here we present only a short summary of modeling choices and an analysis of model results for DB16.

The simulation domain in our study (Fig. 2) started at the mouth of the estuary and extended 88 km upstream. Simulations were generated with SELFE (Zhang and Baptista, 2008)—a finite element, Eulerian–Lagrangian model that solves a set of nonlinear, baroclinic, shallow-water equations. Discretization was based on an unstructured horizontal grid ($\sim 16,000$ elements) and a sigma-coordinate vertical grid (26 levels). The primary prognostic variables were water levels and 3D velocities, salinities, and temperatures.

The modeled circulation of the Columbia River exhibited strong nonlinearity through baroclinicity, sharp velocity gradients, and shallow-water tides. The resolved spatial scales were limited by the resolution of the numerical grid and ranged from ~ 40 m in the main channel of the estuary to ~ 600 m

³ More information at <http://www.co-ops.nos.noaa.gov/>.

in the tidal flats. The resolved periodic timescales included nonlinear tides (4–6 h), astronomical tides (12–24 h), spring-neap tidal cycle (~ 15 days), and seasonal changes in surface heat-fluxes and river temperatures (~ 1 year). Nonperiodic timescales included weather changes (~ 2 –10 days) and inter-annual variations in ocean conditions (> 1 year).

4. Methods

4.1. Overview

The DA algorithm used in our DA experiments was based on the framework for the reduced-dimension KF (Lu et al., submitted for publication), where observations are assimilated into the model surrogate, instead of into the original forward model. This formulation led to the following four-step algorithm:

- (1) A long, statistically representative hindcast simulation of the system using the forward model SELFIE provided training samples for a model surrogate of the Columbia River estuary.
- (2) The training algorithms developed in (van der Merwe et al., 2007) were used to train the model surrogate. Because it was not computationally feasible to train the model surrogate in the high-dimensional space of the forward model, the surrogate was trained in the EOF subspace.
- (3) The state of the model surrogate was estimated using a KF of choice. Because the estimated state of the model surrogate was small (50 degrees of freedom), we choose to use the sigma point KF—a state-of-the-art, full-rank KF (van der Merwe and Wan, 2003; van der Merwe, 2004).
- (4) The output of the KF was analyzed by reconstructing the estimated state and the estimated model errors from the EOF subspace, where the model surrogate and the KF operated, to the full space of the model, where many analysis and visualization tools operated.

The four steps are grouped into two stages in Fig. 3: the off-line stage, which involves one-time, off-line training of the model surrogate (steps 1 and 2 above; left box in Fig. 3), and the on-line stage, which

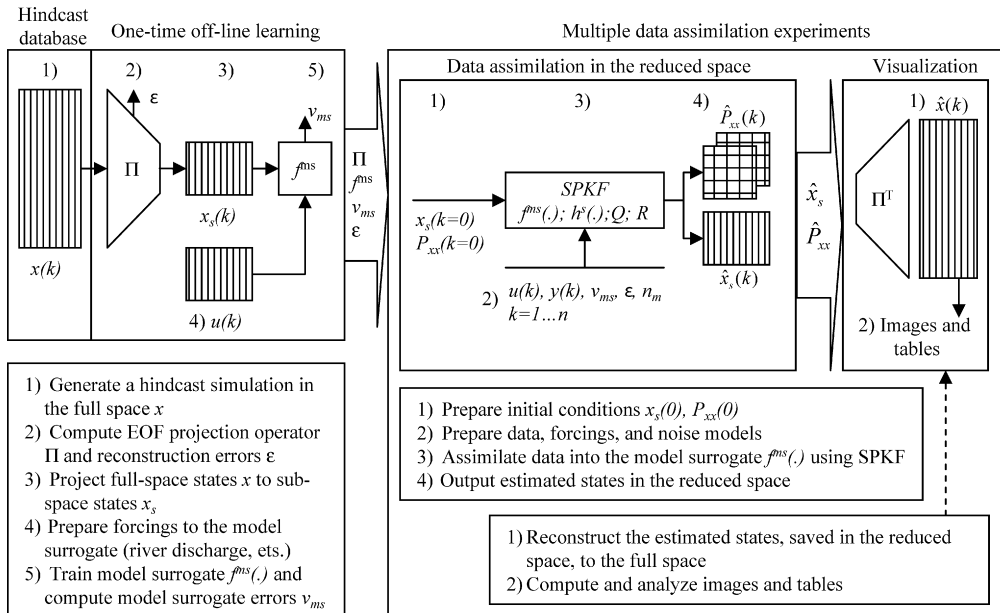


Fig. 3. Information flow in the data assimilation method.

involves multiple DA experiments (steps 3 and 4 above; right box in Fig. 3). Since all DA experiments (on-line stage, step 3 above) were performed in the reduced EOF subspace, the CPU, memory, and storage requirements for these DA experiments were very low. In the case of DB16, it took SELFE ~40 weeks to generate 6 years of the hindcast simulation using a single AMD Athlon® CPU. It took us ~4 weeks to compute the EOF basis and to train the model surrogate—and it took us only ~0.5 week to assimilate 6 years of observational data into the model surrogate. In practice, all these computations were distributed over multiple CPU processors, leading to shorter “wall-clock” times.

We present the description of our DA method for the Columbia River estuary in two parts. In the first part (Section 4), we present a concise overview of the concepts that comprise the reduced-dimension KF (Lu et al., submitted for publication), which include the EOF-based dimension reduction (Jolliffe, 1986), the model surrogate (van der Merwe et al., 2007), the classical KF (Kalman, 1960), and the sigma point KF (van der Merwe and Wan, 2003; van der Merwe, 2004). Each of these concepts is well established in the literature, and readers are referred to the original sources for further algorithmic and theoretical details. In the second part (Section 5), we document our implementation of the reduced-dimension KF for the Columbia River estuary.

4.2. State model in the full space

The forward model and the observational system, described in Sections 3.3 and 3.1, can be formalized as nonlinear dynamical and observation equations defined on the full space of the forward model:

$$x(k) = M(x(k-1), u(k), v(k)), \quad (1)$$

$$y(k) = H(x(k), n_m(k)), \quad (2)$$

where $x(k) \in \mathbb{R}^l$ is a state of the system (e.g., water level, salinity, temperature, and velocity) at discrete time k , $u(k) \in \mathbb{R}^p$ is a vector of forcings (e.g., wind, tides), $y(k) \in \mathbb{R}^m$ is a measurement vector, $M(\cdot) : \mathbb{R}^l \mapsto \mathbb{R}^l$ is a nonlinear dynamical model (e.g., SELFE), $H(\cdot) : \mathbb{R}^l \mapsto \mathbb{R}^m$ is a nonlinear measurement model (e.g., as described in Section 3.1), and $v(k) \in \mathbb{R}^v$, and $n_m(k) \in \mathbb{R}^n$ are the process- and measurement-noise sources. We further denote the space \mathbb{R}^l of state vectors $x(k)$, the full space S .

4.3. Dimension reduction

To enable training of the model surrogates, computationally impossible in the full space S , an EOF-based dimension reduction (also known as *principle component analysis* or *proper orthogonal decomposition*) was used. To justify the use of EOFs, we assumed that the trajectories of the system occupy a low-dimensional (possibly nonlinear) manifold S^{nl} embedded in the full state-space S . This nonlinear manifold is approximated by a linear, low-dimensional subspace S^s , $s \equiv \dim(S^s) \ll \dim(S)$. The EOF subspace is a good candidate for the linear subspace S^s , since, of all linear subspaces of dimension s , the EOF subspace captures the highest percentage of the state variance.

The EOF subspace is characterized by an orthonormal projection operator Π that maps the vectors of the full space $x \in S$ to the vectors of the reduced space $x_s \in S^s$:

$$x_s = \Pi(x - \mu), \quad (3)$$

and that reconstructs (embeds) the reduced space vectors $x_s \in S^s$ into the full space as $\tilde{x} \in S$:

$$\begin{aligned} \tilde{x} &= \Pi^T x_s + \mu \\ x &= \tilde{x} + \varepsilon = [\Pi^T \Pi(x - \mu) + \mu] + \varepsilon \end{aligned} \quad (4)$$

where μ is the mean of the dataset and ε is the reconstruction error.

The EOF projection operator Π is characterized by the r leading eigen vectors (φ_i) of the covariance matrix \mathbf{C} for state vectors $x(k)$:

$$\Pi = [\varphi_1, \dots, \varphi_r]. \quad (5)$$

In practice, a singular vector decomposition (Golub and Van Loan, 1996) was used to compute the operator Π efficiently, without computing the covariance matrix \mathbf{C} explicitly (Bai et al., 2000).

4.4. Nonlinear model surrogate

The projection and embedding equations (Eqs. (3) and (4)) enabled an easy interchange between the full space S and the reduced space S^s representation of the system state. However, for the purpose of simulation, a representation of the dynamics $M(\cdot)$ (Eq. (1)) in the EOF basis (Eq. (5)) was also needed. In the case of linear dynamics, the forward model $M(\cdot)$ can be expressed as a matrix operator and projected to the space S^s using the projection operator Π (Eq. (3)) (Cane et al., 1996). In a nonlinear case, the Galerkin method can be used to project the nonlinear equations of the dynamics to the EOF basis (Holmes et al., 1996). However, such nonlinear projection is non-trivial to implement. To simplify our task, we used a model surrogate to approximate the dynamics of the forward model (Eq. (1)) in the reduced basis (Eq. (5)).

In the past, linear model surrogates were used to expedite DA algorithms. For example, Hoteit et al. (Hoteit and Pham, 2003) used a linear, autoregressive model to expedite the DA of the tropical Pacific. In our preliminary experiments, we also trained a linear model surrogate using the autoregressive model with exogenous inputs (ARX) and the robust least-squares optimization method (Nabney, 2004). However, we found that the trained surrogate was often unstable and led to the exponential growth of the ARX predictor response over time. To address the problem of the unstable linear ARX predictor, we trained a nonlinear model surrogate using a neural network. Output of the neural network is bound by design and cannot yield the exponential response that the linear ARX model yields.

A detailed discussion of the network architecture and of the network's training and validation algorithms is presented in (van der Merwe et al., 2007). The implementation details and the analysis of the accuracy are presented in Section 5.4 of the present paper. Below, we present a brief review of the concepts and terminology that underlie model surrogate training, many of which are common to the neural network literature (Bishop, 1995).

4.4.1. Model surrogate

A model surrogate is a time-lagged, nonlinear neural network designed for iterative prediction:

$$Y_k = M^{ms}(X_{k-1}, U_k) + v_{ms} = \begin{bmatrix} \tilde{x}_s^{ms}(k+p') \\ \vdots \\ \tilde{x}_s^{ms}(k) \end{bmatrix} = M^{ms} \left(\begin{bmatrix} x_s(k-1) \\ \vdots \\ x_s(k-n) \end{bmatrix}, \begin{bmatrix} u(k+p') \\ \vdots \\ u(k-p) \end{bmatrix} \right) + v_{ms} \quad (6)$$

where Y , X , and U are the vectors of the time-lagged outputs, inputs, and forcings; $\tilde{x}_s^{ms}(k)$ is a model surrogate prediction of the subspace state $x_s(k)$; v_{ms} is the error of the surrogate prediction; $M^{ms}(\cdot)$ is a neural network implementation of the model surrogate; k is the index of the current time step; and n , p , and p' are time-embedding indexes. After training is complete, we discard the predictions of the future steps and only use the prediction of the first time step $\tilde{x}_s(k)$ in the iterative prediction loop, which leads to the following simplification in the notation for the model surrogate (Eq. (6)):

$$\tilde{x}_s(k) = M^{ms}(X_{k-1}, U_k) + v_{ms}. \quad (7)$$

Intuitively, the reduced-space representation x_s captures the spatial frequencies of the modeled system, while time-lag in the input and the forcing vectors X and U captures the temporal frequencies and improves the accuracy of the network prediction. Time-lag in the output vector Y_k helps train models for more accurate iterative predictions.

Although the dependence of the predicted state $x_s(k)$ on the future forcings $[u(k+1): \dots : u(k+p)]$ in Eq. (6) may seem counterintuitive, this dependency should be viewed as an artifact of the training procedure. As training methods for model surrogates mature, the time-lag of prediction targets and future forcings may become obsolete, for example as in the algorithm SUR2 in (van der Merwe et al., 2007).

4.4.2. Neural network

We implemented the model surrogate (Eq. (6)) with a standard feed-forward, multi-layer perceptron network. The network had a single hidden layer (with hyperbolic tangent activation functions), a linear output layer, and input pre-processing with an internal PCA (principle component analysis). Such networks are very well-equipped for modeling nonlinear relations among high-dimensional inputs and outputs where large datasets are available for model fitting (training). Where there is significant nonlinearity, the performance of the multi-layer perceptron network far exceeds that of traditional linear models, such as ARX, ARMA, ARMAX, and GLMs (Bishop, 1995).

A mathematical expression for our network architecture is the following:

$$\begin{aligned} Y_k &= M^{ms}(X_{k-1}, U_k) + v_{ms} = \sum_i^m w_{2i} \tanh \left(w_{1i}^T T_I \begin{bmatrix} X_{k-1} \\ U_k \end{bmatrix} + b_1 \right) + b_2 + v_{ms}, \\ W &= [w_1(:); b_1(:); w_2(:); b_2(:)], \\ T_I &= N_{I2} \Pi_I N_{I1} \end{aligned} \quad (8)$$

where $\tanh(\cdot)$ is a hyperbolic tangent function; m is the size of the hidden layer; w_1 , b_1 and w_2 , b_2 are the weight and bias vectors for the first and second layer of the network; and W is a concatenated vector of all unknown parameters w_1 , b_1 , w_2 , and b_2 .

Inputs to the network are pre-processed using matrix T_I —a dimension-reduction operator that consists of the orthonormal projection operator Π_I (computed using PCA of the network inputs) and two diagonal normalization matrixes N_{I1} and N_{I2} . The matrixes N_{I1} and N_{I2} normalize the inputs and outputs of the internal PCA operator T_I , by dividing each input and output with their respective standard deviations. After normalization, the inputs and outputs of the operator T_I have unit variances. The pre-processing with matrix T_I reduces the number of unknown parameters in the network, improves the scaling of the network's cost function, and, as a result, leads to the faster convergence of the optimization algorithm.

The vector of unknown parameters W in Eq. (8) was optimized using a scaled conjugate-gradient algorithm (Møller, 1996) that minimizes a regularized cost function:

$$J(W) = \sum_{i=1}^N \|t_i - M^{ms}(X_i, U_i)\|_2^2 + \lambda \|W\|_2^2 \quad (9)$$

where X_i and U_i are the input and forcing vectors that correspond to the training target t_i , N is the number of available targets (data points), $\|\cdot\|_2^2$ is the square of the 2-norm, and λ is the weight-decay parameter determined through cross-validation. The cost function (Eq. (9)) is a balance between the first term that penalizes the high prediction error v_{ms} of the neural network and a weight-decay regularizer (Bishop, 1995) that penalizes large weights in the neural network and reduces model variance. The regularization of the network is critical to prevent over-fitting to data, which is a consequence of having too few data and too many free parameters. We determined the magnitude of the weight-decay parameter λ through cross-validation. For details on the optimization and cross-validation algorithms, see (van der Merwe et al., 2007) and Section 5.3 of this paper.

4.4.3. Errors in the model surrogate

We assume that errors of a well-trained model surrogate are time-independent, zero-mean Gaussian variables. We estimate the covariance of errors as

$$Q_{ms} = E[v_{ms} v_{ms}^T] = E[(t - M^{ms}(X, U))(t - M^{ms}(X, U))^T]. \quad (10)$$

4.5. Kalman filter in the reduced space

4.5.1. State equations in the subspace

To estimate the state of the dynamical system in the reduced space (Eq. (5)), we defined a reduced-dimension KF that assimilates data into the model surrogate (Eq. (7)) (instead of the original dynamic

model (Eq. (1)). We started by defining a dynamical equation in the reduced space:

$$x_s(k) = M^s(X_{k-1}, U_k, v(k)) \quad (11)$$

where $M^{ms}(\cdot)$ is a reduced space model implemented with the model surrogate (Eq. (7)); X_i and U_i are the time-lagged input and forcing vectors; and $v(k)$ is a process-noise vector that concatenates all error sources in the dynamical model $M^s(\cdot)$, including the model surrogate error v_{ms} . We expanded the definition of the noise vector $v(k)$ for our application in the implementation Section 5.6.

To redefine the observation function $H(\cdot)$ for the reduced state x_s , we employ the embedding equation (Eq. (4)):

$$y(k) = H(x(k), n_m(k)) = H(\Pi^T x^s(k) + \mu + \varepsilon, n_m(k)) = H^s(x^s(k), n_o(k)), \quad n_o(k) = [\varepsilon; n_m(k)] \quad (12)$$

where $H^s(\cdot)$ is the observation function for reduced state x_s , ε is the reconstruction error from Eq. (4), and $n_o(k)$ is the concatenation of the measurement n_m and of the reconstruction error ε . We expand on implementation of the noise source $n_o(k)$ in Section 5.6.

4.5.2. Kalman filter recursion

The following recursive minimum, mean-square-error estimate⁴ for the state $\hat{x}_s(k)$ in Eq. (11), conditioned on the observational data $y(k)$ in Eq. (12), can be derived (Kalman, 1960):

$$\begin{aligned} \hat{x}_s^a(k) &= \hat{x}_s^f(k) + K_k(y(k) - \hat{y}^f(k)) \\ \mathbf{P}_{xx}^a &= \mathbf{P}_{xx}^f - K_k \mathbf{P}_{yy} K_k^T \end{aligned} \quad (13)$$

where $\hat{x}_s^f(k)$ is the optimal prediction of the state at time k conditioned on the observations up to time $k-1$ (state forecast), $\hat{y}^f(k)$ is the optimal prediction of the observation at time k , \mathbf{P}_{xx}^f is the error covariance matrix for the state $\hat{x}_s^f(k)$ (the forecast error covariance), \mathbf{P}_{yy} is the covariance of innovation ($y(k) - \hat{y}^f(k)$), and K_k is the Kalman gain at time k . The optimal terms in this recursion are given by:

$$\begin{aligned} \hat{x}_s^f(k) &= E[M^s(X_{k-1}, U_k, v(k))] \\ \hat{y}^f(k) &= E[H^s(\hat{x}_s^f(k), n_o(k))] \\ \mathbf{P}_{xx}^f &= E[(x_s(k) - \hat{x}_s^f(k))(x_s(k) - \hat{x}_s^f(k))^T] \\ \mathbf{P}_{yy} &= E[(y(k) - \hat{y}^f(k))(y(k) - \hat{y}^f(k))^T] \\ \mathbf{P}_{xy} &= E[(x_s(k) - \hat{x}_s^f(k))(y(k) - \hat{y}^f(k))^T] \\ K_k &= \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1} \end{aligned} \quad (14)$$

The recursion (Eq. (13)) is linear in K ; however, optimal terms (Eq. (14)) do not assume linearity of the model $M^s(\cdot)$ or measurement operator $H^s(\cdot)$. For linear state equations (Eqs. (11) and (12)), the solution for the optimal terms in the recursion (Eq. (14)) is the well-known linear KF (Kalman, 1960).

4.6. Sigma point Kalman filter

To compute the optimal terms (Eq. (14)) for nonlinear state equations (Eqs. (11) and (12)), we used the sigma point KF (van der Merwe and Wan, 2003; van der Merwe, 2004)—a full-rank nonlinear extension to the classical KF. We used the full-rank sigma point KF instead of reduced-rank KF algorithms, such as (Pham et al., 1998; Lermusiaux and Robinson, 1999; Pham, 2001; Evensen, 2003), since the estimated state of the model surrogate was very small (50 degrees of freedom) and it was not necessary to use a reduced-rank approximation to the state covariance matrix \mathbf{P}_{xx} .

⁴ For a Gaussian prior and linear state Eqs. (11) and (12), the posterior distribution on the state $\hat{x}_s(k)$ is Gaussian. In this case, the mean of the posterior distribution is an optimal estimate under the minimum mean-square error, the maximum likelihood, and the maximum *a posteriori* criteria. In the case of a non-Gaussian noise source or nonlinear state equations, the KF is only optimal under the minimum mean-square error criterion.

We introduce the sigma point KF algorithm by comparing it to KF commonly used in oceanography, such as the ensemble KF (Evensen, 2003), the singular evolutive interpolated KF (SEIK) (Pham, 2001), and the singular evolutive extended KF (SEIK) (Pham et al., 1998). Like other ensemble methods (e.g., the ensemble KF and the SEIK), the sigma point KF represents model statistics through a set of ensemble members (sigma points). As a result, the sigma point KF avoids the need for a tangent-linear approximation used by the variants of the extended KF, such as the SEEK algorithm as described in (Pham et al., 1998). Similarly to the SEIK, the sigma point KF algorithm uses the deterministic sampling of the prior covariance, unlike the ensemble KF that uses the Monte Carlo sampling, which is a more computationally expensive sampling method. Another advantage of the sigma point KF is that it directly supports nonlinear observation function and nonlinear observation- and process-noise models, which are only partially supported by the ensemble KF and the SEIK filter.

From a technical viewpoint, the sigma point KF algorithm addresses the problem of computing the optimal terms $\hat{x}_s^f(k)$, $\hat{y}^f(k)$, and K_k (Eq. (14)) in the KF recursion (Eq. (13)). In the sigma point KF, the state distribution is modeled as a Gaussian and is represented by a set of carefully chosen, weighted sample points (the sigma point transformation of the random variable). Fig. 4 illustrates the sigma point transformation for the mean and the covariance of a highly nonlinear function $y = g(x)$ and of a Gaussian random variable x . First, the prior distribution is sampled using $2n + 1$ sigma points drawn according to one of the deterministic methods (the central-difference method here). The mean and the covariance of the sigma points is exactly that of the prior. Second, the sigma points are propagated through a nonlinear function (a random neural network here). Lastly, the posterior mean and covariance are computed as a weighted sample mean and covariance of the updated sigma points, which capture the posterior mean and the covariance to the second order of accuracy (van der Merwe, 2004).

Various implementations of the sigma point KF algorithm, including hybridization with the particle filter, are discussed in (van der Merwe, 2004). The details of the algorithm used in our experiment are presented in Section 5.5. For further description of the sigma point KF algorithm please refer to the original publication (van der Merwe, 2004) and to the ReBel (van der Merwe, 2002–2006)—the software library that we used to implement the sigma point KF equations.

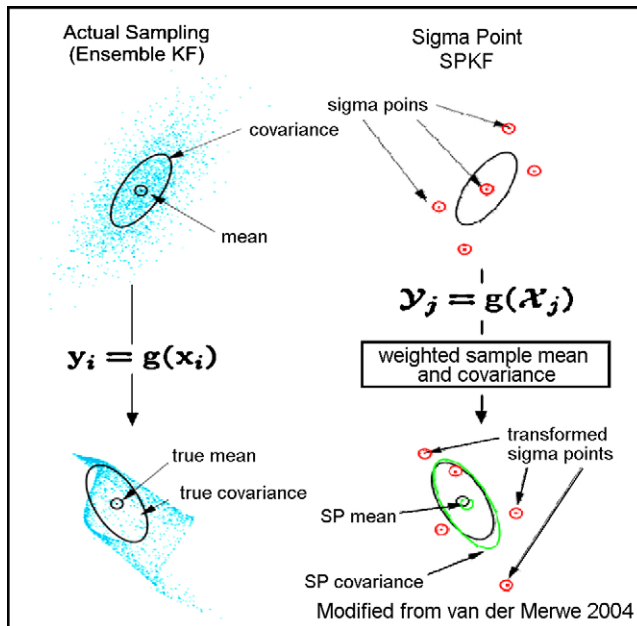


Fig. 4. Illustration of the sigma point transformation for an arbitrary nonlinear function $y = g(x)$ and a Gaussian random variable x . (Image adapted from van der Merwe (2004) with the permission of the author).

5. Implementation

5.1. State vector and model forcings

Two representations of the state vector were used in our experiments:

- (1) The full-space state vector $x(k)$ in Eq. (1), which described the state of DB16 and provided training samples for estimating the EOF dimension-reduction operator (Eq. (3)); and
- (2) The reduced space state vector $x_s(k)$ in Eq. (11), which was used for training the model surrogate (Eq. (6)) and for estimating the state in the KF (Eq. (13)).

The full-space state vector consisted of concatenated model variables defined at the nodes of the computational grid, namely: water level, salinity, temperature, and u and v components of velocity. The total length of the state vector was $\sim 10^6$.

Prior to concatenation, each model variable was normalized with the inverse of its standard deviation:

$$\sigma_{\text{var}} = \sqrt{\frac{1}{l} \sum_{j=1}^l \left[\frac{1}{k} \sum_{i=1}^k (x_{\text{var}}(j, i) - \mu_{\text{var}}(j))^2 \right]}, \quad (15)$$

where i is the time index from 1 to k , j is the index over grid nodes from 1 to l , the subscript var indicates the variable type (e.g., salinity or temperature), x_{var} is the state vector for this variable type, and μ_{var} is the mean vector. Only one normalization constant was used for each variable type.

To represent the extensive wetting and drying of tidal flats in the estuary, for which the model variables were set to missing values by the forward model, we used the following procedure to handle the missing values in the EOF algorithm. For water levels, we substituted the missing values with the values of the bathymetry at these locations. For the rest of the variables, we substituted the missing values with the mean values at these locations. During the reconstruction of the full state from the reduced state, we masked the nodes as dry at the locations where the water column was shallower than 0.1 m, following the definition of dry areas used in the SELFE model (Zhang and Baptista, 2008).

To describe the forcing vector of the model surrogate, we concatenated the original forcings of the forward model (DB16), including the time series of the ocean, river, and atmospheric boundary conditions. The boundary conditions were pre-processed using an EOF dimension reduction, resulting in the forcing vector with 29 degrees of freedom.

5.2. EOF dimension-reduction

To compute the EOF dimension-reduction operator for a large hindcast database of the Columbia River estuary (~ 4 Tb of data for 6.5 years of simulation), we used the divide-and-conquer EOF algorithm (DC-EOF). The DC-EOF algorithm is similar to the algorithm used in (Xue et al., 1997) (details of our implementation described in (Frolov, 2007)). The DC-EOF algorithm operates in two stages:

- (1) A large EOF problem is first divided into smaller sub-problems (e.g., by variable type or by region); for each subproblem, an approximation to the EOF basis is computed from a small, random sub-sample of the original dataset.
- (2) The solutions to these sub-problems are then combined into a single, global solution using the secondary EOF decomposition.

In our experiments, we used 1000 randomly drawn samples to compute the first stage of the DC-EOF algorithm, and we used half-hourly data ($\sim 100,000$ states) in the second stage of the DC-EOF algorithm. To train model surrogates, we retained 50 EOF modes, which captured 98% of the total variance in the test dataset.

5.3. Model surrogate

To train and to validate the model surrogates, we used the training and the cross-validation algorithms described in (van der Merwe et al., 2007). The software implementation of these algorithms was developed by (van der Merwe et al., 2007) and was based on Netlab (Nabney, 2004)—a free network optimization library for MATLAB[®]. The neural-network cost function (Eq. (9)) was optimized using the scaled conjugate gradient algorithm (Møller, 1996), with the network inputs normalized using the internal PCA (Eq. (8)). A two-fold cross-validation, using a simple random permutation,⁵ estimated the value of the weight-decay parameter $\lambda = 5000$.

To improve the accuracy of a model surrogate in the recursive prediction, we introduced training algorithm SUR3, which modifies algorithm SUR1 originally described in (van der Merwe et al., 2007). The original SUR1 algorithm trained a model surrogate to predict only a single, next state of the system ($p' = 0$ in Eq. (6)), which was inconsistent with our intended use of the model surrogate for the recursive prediction. To resolve this inconsistency, van der Merwe (van der Merwe et al., 2007) proposed the algorithm SUR2 that recursively trains the network to predict several steps in the future. Unfortunately, the algorithm SUR2 proved to be too computationally expensive for the intended application in the Columbia River estuary. Instead, we introduced algorithm SUR3, which predicts a short history of the future states ($p' > 0$ in Eq. (6)). The SUR3 algorithm removes the computationally expensive recursion from the training procedure, yet it informs the network of the long-term dynamics in the system.

Using a simple, trail-and-error calibration, we found the following values for the time-embedding parameters in Eq. (6): the time step for states and forcings $\Delta t = 0.5$ h, the past history of states $n = 12$ h, the past history of forcings $p = 24$ h, and the future history of state targets and forcings $p' = 6$ h. In the internal PCA pre-processing (with operator T_i in Eq. (8)), we retained 99% of the variance, which reduced the input layer of the network from 2911 inputs to 548 inputs.

5.4. Accuracy of EOF subspace and the model surrogate

The accuracy of the model surrogate and of the EOF basis is critical to the performance of the reduced-dimension KF. In our analysis, we distinguished between two aspects of accuracy: accuracy in reproducing data that was seen in training (the training error) and accuracy in reproducing new, unseen data (the generalization error). In a hindcast DA, a small training error is sufficient, since it is always possible to overlap the time interval of a hindcast DA experiment with the time interval used in training of the model surrogate. However, a low training error is not sufficient in a real-time assimilative nowcast, which also requires a low generalization error on unseen data. In the case of a high generalization error, a cumbersome retraining of the EOF basis and of the model surrogate may become necessary (however, this was not necessary in our experiments; see Section 6.3).

To determine whether the EOF subspace and the model surrogate, which were trained using DB16, generalized well beyond their training interval, we trained the model surrogate MS_B and the basis EOF_B . Both were trained using the first 5 years of DB16 and were tested using the remaining 1.5 years. The model surrogate MS_B and the basis EOF_B can generalize well if they have similar errors on training and test sets. To eliminate the possibility that the difference between the training and the test error was due to the temporal variability of the error, we compared the test errors for the EOF_B and the MS_B with the errors for the globally trained EOF_A and MS_A . The EOF_A and MS_A were trained on the entire 6.5 years of the DB16 data.

An analysis of the EOF-reconstruction error and the model-surrogate prediction error, presented in Fig. 5, showed that both the EOF_B and the model surrogate MS_B had low training errors (2.1% for EOF_B and 1.9% for MS_B). However, we found that only the EOF basis EOF_B was able to generalize well (errors of 2.1% on the training set and 2.0% on the test set). The model surrogate MS_B had exhibited a small

⁵ The term *random shuffle* cross-validation is from (van der Merwe et al., 2007).

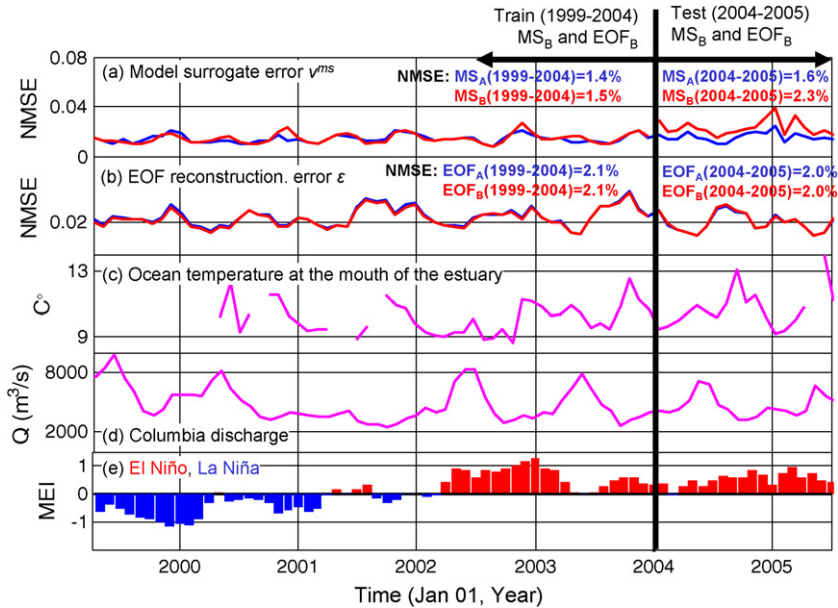


Fig. 5. Monthly averaged time series of model surrogate (a) and EOF errors (b) compared with monthly averaged time series of Columbia River estuarine forcings. The model surrogate error: (a) was measured for a continuous, 6.5-year feedback integration of the model surrogate, using the initial condition and the model forcings from DB16. The mean square errors in (a) and (b) were normalized by the variance of the state vector in the full space (NMSE). Training and testing periods for MS_B and EOF_B are indicated by the black vertical line. (c) Ocean temperature at the mouth of the estuary, estimated from a composite of available observations. (d) River discharge was observed by USGS at the Bonneville Dam. (e) Multivariate ENSO Index (MEI) as computed by Klaus Walter at NOAA (<http://www.cdc.noaa.gov/people/klaus.wolter/MEI/>).

discrepancy between errors on the training set and the test set (1.5% vs. 2.3%), which were confirmed to be a generalization error, determined after we compared the errors of the model surrogate MS_A and the MS_B. It is likely that the generalization error for the MS_B can be reduced by using more sophisticated, cross-validation strategies for calibrating the weight-decay parameter λ in Eq. (9); see (van der Merwe et al., 2007) for such strategies. Section 6.1.1 presents further analysis of how this generalization error affected the overall performance of the DA system and, hence, the prospect of deploying our DA system as a real-time nowcast.

To determine whether the 1.5 years of test data were dynamically different from the 5 years of training data, we compared the time series of the EOF and model surrogate errors with the time series of estuarine and climatic forcings (Fig. 5). To represent estuarine forcings, we chose Columbia River discharge at the Bonneville Dam and temperature of the ocean water at the mouth of the estuary, shown in Fig. 5c and d. To represent climatic forcings, we chose a multivariate ENSO index (Wolter and Timlin, 1993, 1998), shown in Fig. 5e. Both the estuarine and the climatic forcings showed significant inter-annual variability in the seasonal cycle. For example, during the test period, spring freshets in the river discharge were smaller than most freshets in the training period, but not as small as the anomalously low spring freshet of 2001. For ocean temperature, the test period was marked by slightly higher maximum temperatures (14 °C vs. 13 °C), which were due to an unseasonably strong downwelling event in the early summer of 2005. Unfortunately, since neither a strong El Niño (MEI > 1) nor a strong La Niña (MEI < -1) event was observed during 6.5 years of DB16 coverage, we were not able to test how well the EOF and the model surrogate generalize during a period of a strong climatic forcing. The inter-annual variabilities in the estuarine and climatic forcings suggest that the selected test period was sufficiently different to test the accuracy of the EOF basis and the model surrogate during a typical forcing year. However, a longer simulation period is needed, to test how

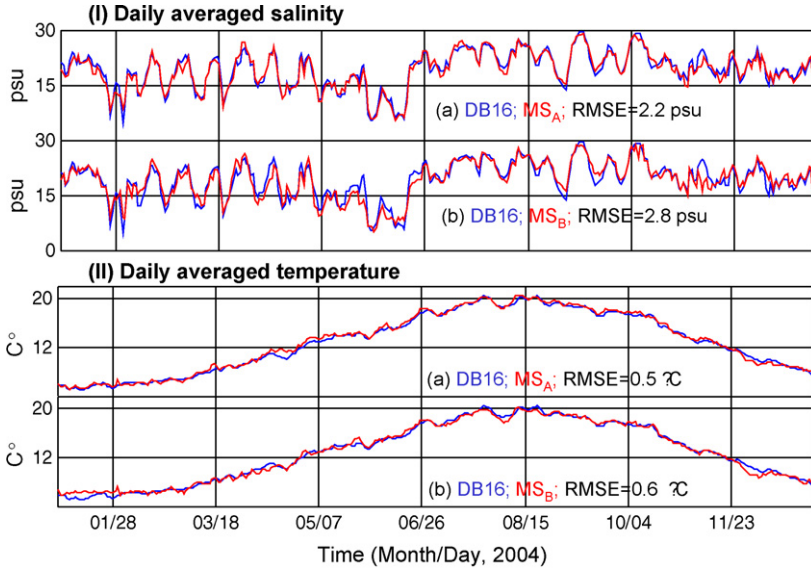


Fig. 6. Daily averages for salinity (I) and temperature (II) at *am169* (bottom sensor). (a) Model surrogate MS_A , and (b) Model surrogate MS_B . The RMSE are between DB16 (blue line) and the model surrogates (red line).

the EOF basis and the model surrogate generalize during a period of extreme climatic or estuarine forcings.

To illustrate how little the generalization error for the model surrogate MS_B affected the predicted values of salinity and temperature in the Columbia River estuary, we compared the daily averaged salinity (Fig. 6I) and the daily averaged temperature (Fig. 6II) for the model surrogates MS_A and MS_B . The comparisons were for the bottom sensor at station *am169*. Fig. 6 shows that both MS_A (sub-panel a) and MS_B (sub-panel b) tracked the solution of DB16 well. For example, the differences between DB16 and MS_A and MS_B , as measured by the root mean square (RMS) error, were very small: 2.2 psu vs. 2.8 psu for salinity and 0.5 °C vs. 0.6 °C for temperature. We did not find any systemic differences between DB16 and model surrogates.

5.5. Sigma point Kalman filter

To estimate the state of the system, we used a square root central-difference formulation of the sigma point KF (van der Merwe and Wan, 2003; van der Merwe, 2004). The square root version of the sigma point KF propagates the covariance matrix \mathbf{P}_{xx} in a numerically more stable and computationally more efficient square-root form. Central difference is the sampling algorithms that we used to draw sigma points from the prior distribution. To specify the spread of the sigma points in the central difference algorithm, we used scaling parameter ζ of $\sqrt{3}$, which is optimal for Gaussian distributions. Our preliminary experiments showed that state estimates were not sensitive to the choice of the sampling algorithm and the scaling parameter. Our implementation was based on ReBel (van der Merwe, 2002–2006)—a well-tested, freely available Matlab® package for the recursive Bayesian estimation.

To ensure that the measurements of different types (e.g., water level and salinity) were weighted similarly in the update equations for the KF (Eq. (13)), we normalized the measurements the same way in which we normalized the state vector of the system (Eq. (15)). The observations were assimilated by the sigma point KF every half-hour.

5.6. Noise models

5.6.1. Process noise

We designed the following process-noise model to characterize the errors in the model forcings, the errors of the model surrogate prediction, and the errors of the original forward model:

$$\begin{aligned}\tilde{x}_s^{ms}(k) &= M^s(X_{k-1}, U_k, v(k)) = \\ &= \sum_i^m w_{2i} \tanh \left(w_{1i}^T T_l \begin{bmatrix} X_{k-1} \\ U_k \end{bmatrix} + b_1 + v_{ui} \right) + b_2 + v_x + v_{ms} \\ v_{xms} &= v_x + v_{ms} \\ v(k) &= [v_{ui}; v_{xms}]\end{aligned}\quad (16)$$

where $\tanh(\cdot)$ is a hyperbolic tangent function; m is the size of the hidden layer; w_1 , b_1 and w_2 , b_2 are the trained weights and biases vectors for the model surrogate (Eq. (8)) (truncated to provide only one-step-ahead prediction); and T_l is the internal PCA operator of the trained model surrogate.

The noise sources in Eq. (16) are:

$$\begin{aligned}v_{ui} &\sim N(\mathbf{0}, \mathbf{Q}_{ui}) \\ v_{xms} = v_x + v_{ms} &\sim N(\mathbf{0}, \mathbf{Q}_{xms}) = N(\mathbf{0}, \alpha_x \mathbf{Q}_{ms})\end{aligned}\quad (17)$$

where v_{ui} , v_x , and v_{ms} are the forcing, the forward model, and the model surrogate errors, with the error covariance \mathbf{Q}_{ui} , \mathbf{Q}_x , and \mathbf{Q}_{ms} , respectively. The prediction errors for the model surrogate v_{ms} and for the forward model v_x are combined into a single noise source v_{xms} , with the combined covariance \mathbf{Q}_{xms} .

To characterize the combined prediction error v_{xms} , we approximated the error covariance \mathbf{Q}_{xms} with a scaled variance of the model surrogate error v_{ms} :

$$\mathbf{Q}_{xms} = \alpha_x \mathbf{Q}_{ms} = \alpha_x * \text{diag}(\text{var}[v_{ms}])\quad (18)$$

where α_x is the tunable scaling parameter.

To characterize the forcings error covariance \mathbf{Q}_{ui} , we used a scaled version of the forcing variance that was projected onto the subspace of the network weights using the internal PCA operator T_l :

$$\begin{aligned}\mathbf{Q}_{ui} &= T_l \begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_u \end{bmatrix} T_l^T \\ \mathbf{Q}_u &= \alpha_u * \text{diag}(\text{var}[U])\end{aligned}\quad (19)$$

where \mathbf{Z} is a square zero matrix with a size equal to the length of past states X , and α_u is a scaling parameter that requires tuning. The projection (Eq. (19)) accomplished two objectives: (i) it expedited the sigma point KF computation by reducing the dimension of the process-noise vector, and (ii) it introduced the temporal correlations in the forcing noise, which were encoded in the operator T_l .

The KF in this study estimated only the current time step $\hat{x}_s(k)$, with the uncertainty in the initial conditions entering a single past-time-step $x_s(k-1)$. The remaining past states in the model surrogate $[x_s(k-2); \dots; x_s(k-n)]$ were treated as a bookkeeping variable. The KF in our study is unlike the original reduced-dimension KF (Lu et al., submitted for publication) that estimated the entire history of the present and the past states $[\hat{x}_s(k); \dots; \hat{x}_s(k-n)]$, which led to a fix-lag formulation of the Kalman smoother.

To summarize, uncertainty (noise) entered our modeling system (Eq. (11)) in several ways. Uncertainty in the initial conditions entered into the past state $x_s(k-1)$ through the covariance matrix \mathbf{P}_{xx} . Uncertainty in forcings entered the entire time-embedded history of forcings U_k as a white Gaussian noise, projected to the subspace of the network weights. Finally, uncertainty in the accuracy of the modeling system entered as a scaled version of the model surrogate error v_{ms} .

5.6.2. Observation noise

In Section 4, we used a generic, nonlinear observation function (Eq. (12)) in the definition of the reduced-dimension KF. In our experiments, the nonlinearity in observation operator was due to vertical interpolation on the sigma grid. Assuming that the reconstruction error ε was small in comparison to the reconstructed state \tilde{x} , we used the following approximation to Eq. (12):

$$y(k) = H^s(x^s(k), n_o(k)) = H(\tilde{x} + \varepsilon) + n_m \approx H(\tilde{x}) + H_\mu(\varepsilon) + n_m \quad (20)$$

where $\varepsilon \sim N(0, R_\varepsilon)$ and $n_m \sim N(0, R_m)$ are the reconstruction and the measurement errors, and $H_\mu(\varepsilon)$ is the observation of the reconstruction error for the mean state μ . The covariance of the reconstruction error R_ε was estimated as

$$R_\varepsilon = \text{diag}[\text{var}(\varepsilon)]. \quad (21)$$

We made the following best guesses for the values of the standard deviation for the measurement error n_m for: water level, 0.01 m; salinity, 0.5 psu; and temperature, 0.1 °C.

5.7. Calibration of process-noise models through cross-validation

To calibrate the meta-parameters α_x and α_u of the noise model (Eqs. (16)–(19)), we used three optimization criteria:

- (1) The n -fold cross-validation (Bishop, 1995);
- (2) The maximum likelihood criterion of (Dee, 1995); and
- (3) The consistency criterion described in Section 6.5.1 (Eq. (24)).

Unlike the cross-validation criterion, which optimizes DA to accurately estimate the system-state at a rotating set of validation stations, the maximum likelihood and the consistency criteria optimize DA in order to accurately predict the statistics of the model error. That is, they adjust the parameters of the noise model until the predicted forecast error covariance matches the observed forecast error covariance, e.g., as measured at the locations of the observations.

The results of the cross-validation experiment are presented in Fig. 7, with the estimated values of $\alpha_x = 20$ and $\alpha_u = 1$. To fit the noise model parameters α_x and α_u , we used a fourfold cross-validation technique. The errors from different sensors were normalized similar to Section 5.5. We first searched for the optimal value of α_x and then for the value of α_u . To optimize calibration period for data availability and to save computational resources, we ran cross-validation experiments using 3 weeks of observational data (05/07/2004–05/28/2004).

The calibration results based on the maximum likelihood and the consistency criteria showed good agreement with the results obtained using the cross-validation criterion, suggesting that the noise

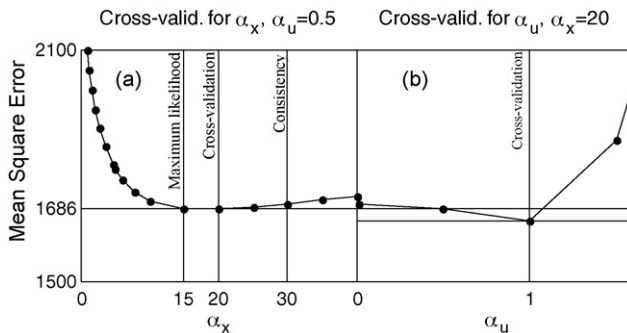


Fig. 7. Cross-validation curves for α_x (a) and α_u (b).

model (Eqs. (16)–(19)) was of high quality. Under the maximum likelihood criterion, the calibrated value of α_x was 15; and for the consistency criterion (Eq. (24)), α_x was 30. As shown by the cross-validation curve for α_x (Fig. 7a), there was no significant difference between the values of two estimates ($\alpha_x = 15$ and $\alpha_x = 30$), since the quality of the state estimate did not change much between the two values of α_x . In the DA-experiments (Section 6), we used the parameters optimized under the cross-validation criterion; hence, we traded a sub-optimal forecast of the model uncertainty for a better quality of the state estimates.

6. Validation experiments

6.1. Strategy

To validate the DA algorithm described in Sections 4 and 5, we examined how DA can improve the simulated variability of the Columbia River estuary. Specifically, we evaluated the improvements at locations other than the locations of assimilated measurements and the improvements to the long-term hindcast simulation of the estuary. The assimilated data included water level, salinity, and temperature measurements, from the fixed network of sensors described in Section 3.1. In the absence of extensive observational data from sensors independent of the fixed network, such as sensors deployed on mobile or remote sensing platforms, we used a cross-validation procedure discussed in detail in Section 6.1.1. This procedure is similar to the one that we used to calibrate the process-noise model in Section 5.7.

We focused on answering four specific questions regarding the skill of the DA system that we developed for the Columbia River estuary:

- (1) *Does our DA method improve the simulated variability of the Columbia River estuary?* To evaluate the improvement, we used a leave-one-out cross-validation procedure to estimate the reduction of model errors across the Columbia River estuary. The details of the procedure are presented in Section 6.1.1.
- (2) *Is it possible to apply our DA method to a real-time nowcast, without retraining the model surrogate and the EOF basis?* To study this possibility, we used the leave-one-out cross-validation procedure to compare the error reduction in two DA experiments that used different model surrogates. The first model surrogate, MS_A , was used in a prototype of the hindcast DA experiment since MS_A was partially trained during the cross-validation study. The model surrogate MS_B was used in a prototype of the nowcast DA experiment since it was trained outside the cross-validation period.
- (3) *Does our DA method improve the accuracy of a long hindcast simulation?* To evaluate such improvements, we conducted a year-long experiment, where the accuracy of the DA experiment was validated at key stations in the Columbia River estuary. Specifically, we evaluated whether DA could improve the representation of tidal and sub-tidal variability of water levels, the response of salinity intrusion to changes in river and ocean conditions, and the representation of the seasonal cycle of heating and cooling in the Columbia River.
- (4) *Are the DA-estimates of the Columbia River estuary statistically and dynamically consistent?* To study the statistical consistency of the DA estimates, we checked whether the error-statistics predicted by the DA were consistent with the observed error-statistics. To study the dynamical consistency of the DA estimates, we checked whether the estimates obeyed mass- and volume-conservation laws.

To distinguish between different DA experiments, we introduced the following nomenclature. The baseline DA experiment, which used all available data, was denoted *DA-all*. For validation DA experiments, we used a three-part abbreviation *DA-(abbreviated variable name)-(station name)* to indicate the nature of data that were withheld for validation. For example, the name *DA-wl-tpoint* denoted the validation experiment where the water level (*wl*) data from the station *tpoint* were withheld for validation.

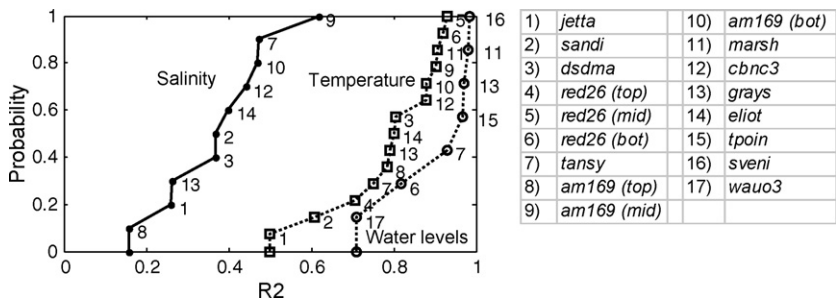


Fig. 8. Empirical cumulative distribution function (ECDF) for the percentage of the error variance at validation stations (list on the right) as explained by the assimilation stations. X-axis: R^2 - the percentage of the error variance at the validation station as explained by the assimilation stations. Y-axis: Probability that an explained error variance at a validation station is less than x.

6.1.1. Using cross-validation to validate DA experiments

Leave-one-out cross-validation (Bishop, 1995) is a validation procedure that effectively reuses observational data for assimilation and validation by estimating the accuracy of a DA solution across model variables and throughout a computational domain. This reuse is achieved by averaging the validation errors over multiple DA experiments. In each experiment, data from only a single sensor are withheld for the validation, hence leading to a minimal impact on the estimate of the DA accuracy. Cross-validation was first used to estimate the accuracy of a DA solution by Cane (Cane et al., 1996).

A concern with using the cross-validation criterion to measure DA accuracy is that errors at the validation stations may be strongly correlated with errors at assimilation stations; hence providing poor candidates for independent validation of DA accuracy. To illustrate how the error-correlation strength varied between different validation stations, we used the R^2 statistics, which measured what percentage of the error-variance at a validation station was explained by the cross correlation with errors at assimilation stations. Fig. 8 presents the variability in R^2 values as an empirical, cumulative distribution function.

The results in Fig. 8 show that our observational array had the strongest error correlation for water level measurements (average $R^2 = 0.93$) and the weakest error correlation for salinity measurements (average $R^2 = 0.37$). The R^2 values varied strongly among different stations in the Columbia River estuary. In the case of salinity measurements, the R^2 value varied from 0.16 for the surface sensor at *am169* to 0.62 for the mid-water sensor at *am169*. The strong error correlation for water levels, as compared to the weak error correlation for salinity, indicated an under-sampled salinity-error field, as compared to the well-sampled water-level-error field.

6.2. Overall improvement in simulated variability of the Columbia River estuary

To evaluate the overall accuracy of our DA system, we applied the leave-one-out cross-validation procedure to a 3-week-long time interval (05/07/2004–05/28/2004), which was chosen to simplify analysis of the DA accuracy by minimizing the impact of gaps in observational data. The results of the cross-validation study are presented in two Tables 1 and 2. Table 1 presents the results in terms of RMS errors, while Table 2 presents the errors as a percentage of the data variance. The cross-validation results are presented for the locations of assimilation stations (columns D and F in both tables) and for the locations of validation stations (columns E and G). The error at validation stations was averaged over all validation experiments. In this section, we present the results of the hindcast DA experiment (columns D and E), which was based on the model surrogate MS_A described in Section 5.4. For a comparison between the hindcast and the nowcast experiment, see Section 6.3.

The cross-validation study showed that DA effectively reduced errors in water levels, salinities, and temperatures at both the assimilation stations and the validation stations. At the assimilation stations (column D), the average error (over all stations and time) was reduced from 0.14 m to 0.06 m for water levels, from 4.3 psu to 1.2 psu for salinity, and from 1.2 °C to 0.5 °C for temperature. When the errors

Table 1
Root mean square errors (RMSE) for the cross-validation experiments.

Variables	Observation noise		RMSE				
	n_m	n_o	DB16	Hindcast		Nowcast	
	A	B		Assim.	Valid.	Assim.	Valid.
			C	D	E	F	G
Water levels (m)	0.01	0.04	0.14	0.06	0.10	0.06	0.10
Salinity (psu)	0.5	1.5	4.3	1.2	3.0	1.2	3.0
Temperature (°C)	0.1	0.3	1.2	0.5	0.7	0.5	0.7
Velocity magnitude (m/s)	n/a	n/a	0.274	n/a	0.268	n/a	0.270
Along-channel residual velocity magnitude (m/s)	n/a	n/a	0.099	n/a	0.070	n/a	0.070

were expressed as the percentage of the data variance at stations, the errors for water levels were reduced from 2.5% to 0.5%, for salinity from 13% to 1.1%, and for temperature from 0.7% to 0.1%. As expected, the errors after DA at the assimilation stations were at or below the combined observation error n_o , which varied for water levels from 0.03 m to 0.06 m, for salinity from 0.5 psu to 2.0 psu, and for temperature from 0.12 °C to 0.34 °C. At the validation stations (column E), the reduction in errors was smaller than that at the assimilation stations (column D), but was still substantial: for water levels from 0.15 m to 0.10 m (from 2.5% to 1.2%), for salinity from 4.3 psu to 3.0 psu (from 13.0% to 6.1%), and for temperature from 1.2 °C to 0.7 °C (from 0.7% to 0.3%).

Velocity magnitudes showed a substantial improvement (from 14.7% to 7.4%) in along-channel residual velocities ($T > 30$ h), but only marginal improvement (from 30.1% to 28.8%) in instantaneous velocities. Because velocity measurements were not assimilated, we attribute the improvement in residual velocities to a multivariate correction from assimilating salinity and temperature. Assimilating salinity and temperature improves density structure in the estuary, which is a driver of the residual circulation. For example, Fig. 9 shows how the profiles of residual velocity (panel b) responded to improvements in the density structure (shown as salinity profiles in panel a) at stations *red26*, *tansy*, and *am169*. The estimated residual velocities were most accurate at stations *tansy* and *am169*, where salinity data were assimilated. Further investigation (not shown) suggests that DB16 errors in instantaneous velocities are primarily associated with lateral (cross-channel) circulation. Likely causes are local effects, associated with the geometry of either tidal flats or channel banks. Our sensor network is too sparse to allow assimilation of scalar variables to correct for such localized errors in velocity.

6.3. Comparison of error reduction in the nowcast and hindcast experiments

To evaluate the ability of our DA algorithm to assimilate data in a real-time nowcast, we compared the performance of DA experiments based on two model surrogates, described in Section 5.4. The

Table 2
Errors in cross-validation experiments as a percentage of data variance at validation stations.

Variables	DB16	Hindcast		Nowcast	
		Assim.	Valid.	Assim.	Valid.
	C	D	E	F	G
Water levels (%)	2.5	0.5	1.2	0.5	1.2
Salinity (%)	13.0	1.1	6.1	1.0	6.2
Temperature (%)	0.7	0.1	0.3	0.1	0.3
Velocity magnitude (%)	30.1	n/a	28.8	n/a	29.3
Along-channel residual velocity magnitude (%)	14.7	n/a	7.4	n/a	7.3

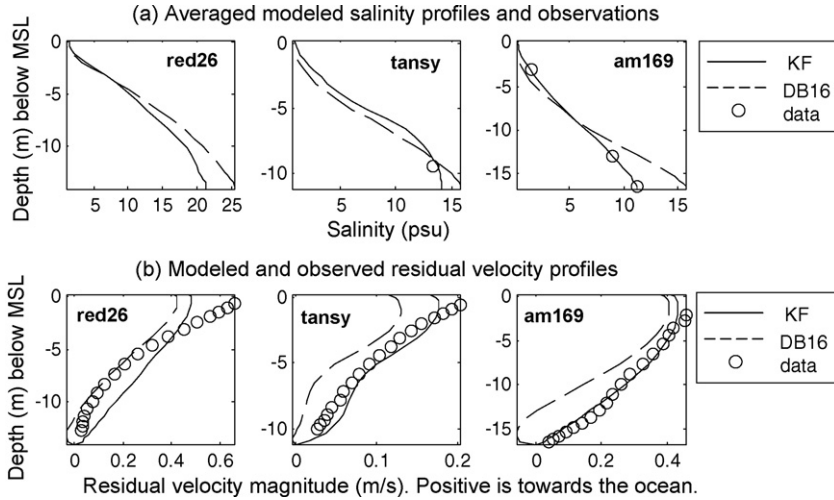


Fig. 9. Average salinity and residual velocity profiles at stations *red26*, *tansy*, and *am169*. (a) Average salinity profiles and observations. (b) Along channel residual ($T > 30$ h) velocities. Averages were computed for the experiment *DA-all* during from 05/07/2004 to 05/28/2004.

model surrogate MS_A was used in a prototype of the hindcast DA experiment since MS_A was partially trained during the cross-validation study. The model surrogate MS_B was used in a prototype of the nowcast DA experiment since it was trained outside the cross-validation period.

The comparison of cross-validation errors in Tables 1 and 2 suggests that the nowcast DA experiment (columns F and G) was as accurate as the hindcast DA experiment (columns D and E). This result is significant for the operational applications of our DA method since this result shows that the reduced-dimension KF can be successfully applied in a real-time nowcast of the Columbia River estuary without the need to retrain the model surrogate and the EOF on a new set of data.

6.4. Improvements in the simulated variability of a long-term hindcast experiment

To determine whether our DA improved the simulated variability of a long-term hindcast simulation, we analyzed the accuracy of the simulated tides, salinity intrusion, and temperature in a 1-year-long hindcast simulation of the Columbia River estuary.

6.4.1. Tidal variability

Tidal variability in the Columbia River estuary is nonlinear and non-stationary (Jay et al., 1997); it is governed by the nonlinear interactions between several external forcings, such as ocean tides, ocean set-up due to atmospheric influences, bottom-friction forces, and river discharge. The uncertainties in these forcings can lead to errors in the simulated tides. To determine whether DA can reduce the tidal errors, we performed the DA experiment *DA-wl-tpoint*, where the water level data were assimilated at all stations, but the station *tpoint*, which was used for an independent validation.

The results from the validation experiment *DA-wl-tpoint* showed that the DA improved the representation of the tides in the estuary, as seen in Fig. 10, which compares the maximum daily errors at the validation station *tpoint* for DB16 and the DA experiment *DA-wl-tpoint*. The water-level errors were reduced from 0.14 m to 0.07 m for the all-pass band, from 0.10 m to 0.04 m for the tidal band ($T < 30$ h), and from 0.10 m to 0.06 m for the sub-tidal band ($T > 30$ h). While the error reduction in the tidal band was consistent throughout the experiment, the reduction in the sub-tidal band was most prominent during winter storms (January–March and October–December). The exceptions to this effective error-reduction were two short periods, marked with arrows in Fig. 10, when most of

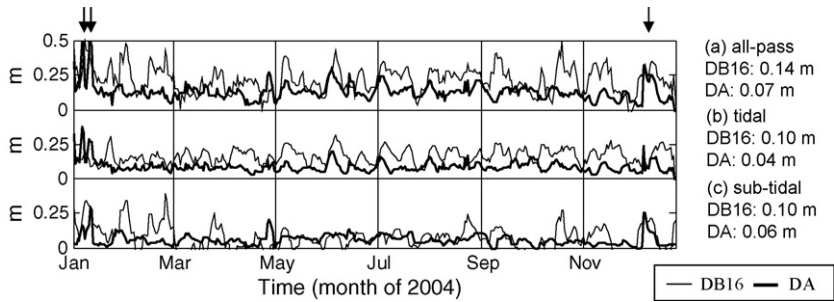


Fig. 10. Maximum daily error in simulated water levels for DB16 (dashed) and DA experiment *DA-wl-tpoint* (solid). (a) All-pass errors, (b) errors in tidal band, and (c) errors in sub-tidal band. Separation into tidal and sub-tidal bands was with a butterworth filter ($T = 30$ h). Values of the RMSE are listed to the left of panels. Vertical arrows indicate periods when most water-level sensors in the estuary failed.

the water-level sensors failed and when the errors in the DA experiment were similar to the errors of DB16.

6.4.2. Salinity intrusion

To determine whether our DA method improved upon the simulated salinity in DB16, we studied the errors in simulated salinity at four distinct regions of the Columbia River estuary: the mouth of the estuary, the mid-estuary, the upper estuary, and a small lateral channel in the estuary. Salinity variability in these four regions were exemplified by measured salinities at stations *jetta*, *am169* (bottom sensor), *cbnc3*, and *eliot*, respectively. For each region, the results of two DA experiments were analyzed. In the first experiment, all available data were assimilated (*DA-all*). In the second experiment, the data at the validation stations were withheld for validation. To present the long-term variability of salinity at the validation stations, we used the daily salinity-means as proxy statistics in Fig. 11. The daily-averages were computed over a tidal day of 30 h.

The results of the validation experiments showed that DA improved the simulated salinity at three regions of the estuary: in the mouth of the estuary (Fig. 11I), in the mid-estuary (Fig. 11II), and in the small lateral channel (Fig. 11III). When all data were used in assimilation, the simulated variability improved substantially in all three regions, including a reduction in the RMS errors, a reduction in model biases, and the better representation of episodic events. When the validation data were withheld from the assimilation, improvements at these validation stations were modest. The strongest improvement was at the mid-estuary station *am169*, suggesting that the observation network was more informative in the middle of the estuary than in the mouth or in the lateral channel of the estuary.

In the fourth region—the upper estuary (Fig. 11IV)—the DA method failed to improve the simulated salinity, possibly due to the limitation of the EOF subspace, where our KF was defined. In fact, the EOF space, trained on the forward model, encoded the upper estuary as permanently fresh, since the forward model (DB16) failed to propagate any salinity to the upper estuary. As a result, the KF, defined in this subspace, also failed to estimate nonzero salinity in the upper estuary. However, the region of the upper estuary where the KF failed was relatively small in comparison to the rest of the estuary, where the KF succeeded in improving simulations of salinity.

To further illustrate the ability of the DA to control the length and the shape of the salt wedge, we compared in Fig. 12 transects of model salinities from DB16 (panel I) and the DA experiment *DA-all* (panel 2) and *DA-salt-am169* (panel 3). These transects were taken along the navigation channel of the Columbia River estuary, during a period (20–27 September, 2004) of higher model error. In DB16, the simulated estuary was too stratified and too saline at station *am169*, as seen in Fig. 12Ia. After DA (shown in Fig. 12II and III), the stratification at *am169* was corrected, resulting in a more mixed estuary. Experiment *DA-salt-am169* shows that our salinity array had enough information to correct

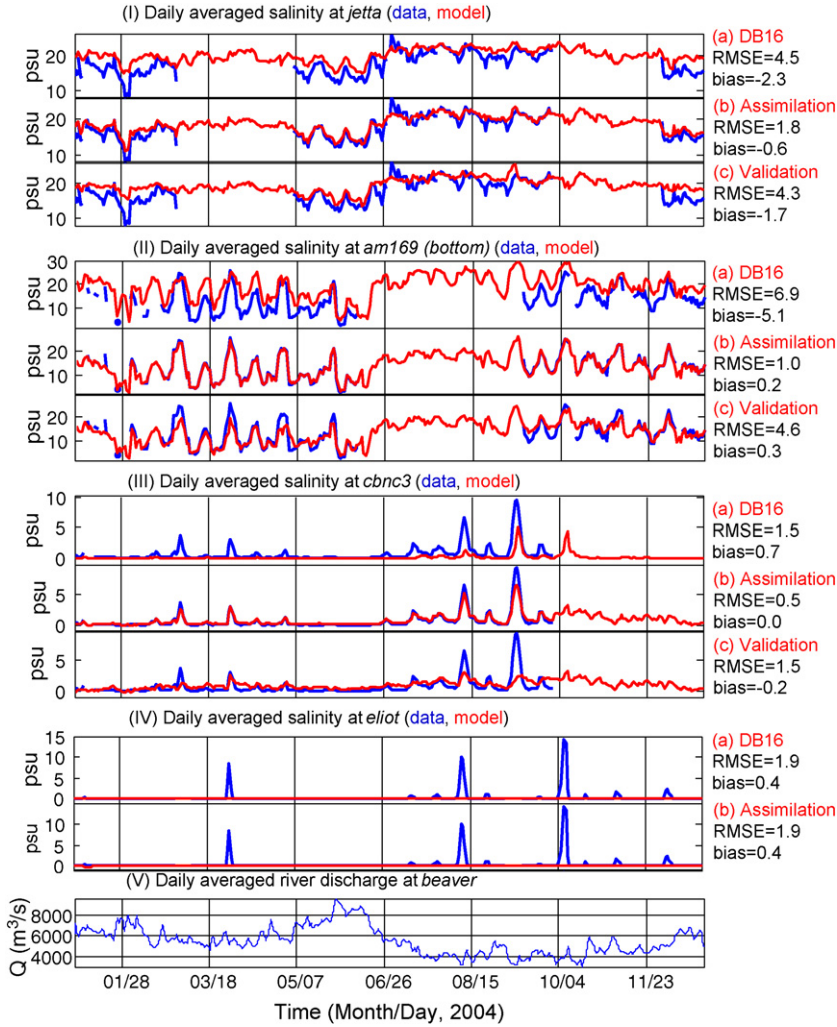


Fig. 11. Daily averaged salinities at stations *jetta* (I), *am169* (II), *cbnc3* (III), and *eliot* (IV). Observed, daily averaged river discharge (V). Observational data are in blue, and model data are in red (I, II, and III). For each station, except for *eliot*, three results are presented: (a) no data were assimilated (DB16), (b) data from this station (*DA-all*) were used for assimilation, and (c) data from this station were withheld for validation. Values of a RMSE and a model bias are listed on the right of each panel.

stratification at station *am169* even when the data at station *am169* were withheld for validation; see transect and time series shown in Fig. 12III.

The problem of salinity intrusion in the estuary is further investigated in separate publications (Frolov, 2007; Frolov et al., in press), where we compare the length of salinity intrusion along the main channel of the estuary against field observations and non-assimilated model simulations.

6.4.3. Temperature variability

To determine whether DA improved upon the simulated temperature in DB16, we studied three regions of the Columbia River estuary: the mouth of the estuary, the mid-estuary, and a small lateral channel of the estuary. Temperature variability in these three regions were exemplified by tempera-

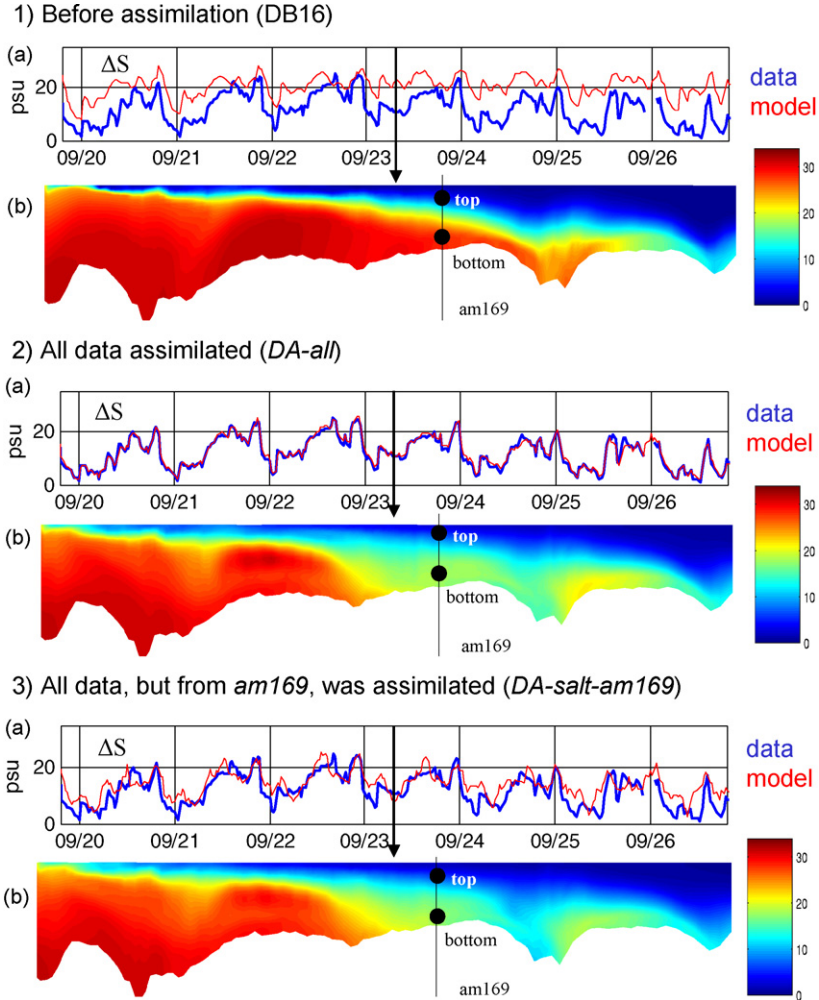


Fig. 12. Salinity stratification at *am169* (shown on panels [a]), compared to the transect of salinity along the navigation channel (shown on panels [b]). In [a], observations are in blue and model data are in red. Stratification was computed as the difference in salinities between the bottom and top sensors. Transect was measured on September 23, 2004 at 07:15 (solid vertical line in [a]). Panel (I) is before data assimilation, panel (II) is with all data assimilated, and panel (III) is with all data, but *am169*, assimilated.

tures at stations *jetta*, *am169* (bottom sensor), and *cbnc3*, respectively. Fig. 13 shows the analysis of temperature estimates in these three regions, following the methods described in Section 6.4.2.

The results of the validation studies presented in Fig. 13 showed that DA reduced RMS errors and improved the temperature representation at all three regions of the estuary. In the mouth of the estuary and in the mid-estuary (Fig. 13I and II), DA improved the representation of cold water intrusion events. The timing of these events, shown with green arrows in Fig. 11I and IV, correlated with the relaxation of the upwelling-favorable winds, as seen in the time series of the North-South winds measured off-shore of the Columbia River mouth (Fig. 13IV). In the small lateral channel of the estuary (Fig. 13III), DA improved the representation of the annual temperature trend, which was misrepresented in DB16 possibly due to errors in heating and cooling of the waters in estuarine tidal flats.

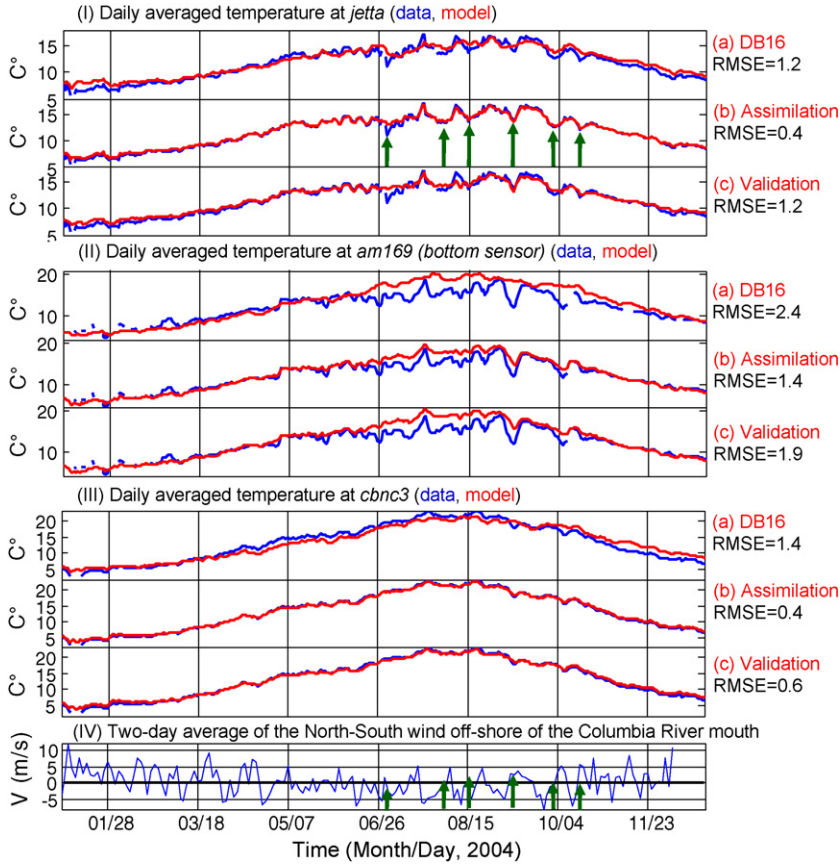


Fig. 13. Daily averaged temperatures at stations *jetta* (I), *am169* (II), and *cbnc3* (III). The 2-day average of the North-South wind measured off-shore of the Columbia River mouth (NOAA buoy 46029) (IV). Observational data are in blue and model data are in red (I, II, and III). For each station, three results are presented: (a) no data were assimilated (DB16), (b) data from this station (DA-all) were used for assimilation, and (c) data from this station were withheld for validation. Values of RMSE and a model bias are listed to the right of each panel. Green arrows highlight upwelling episodes.

6.5. Consistency of assimilated results

6.5.1. Statistics of the innovation errors

To determine whether the estimate of error statistics predicted by the DA was statistically consistent with the measured error statistics, we compared the predicted error variance for the state forecast $\hat{x}_s^f(k)$

$$\hat{\sigma}_{inov}^2(k) = \text{trace} \left[\frac{\mathbf{P}_{yy}(k)}{m} \right] \quad (22)$$

with the measured variance of the innovation errors

$$\sigma_{inov}^2(k) = \text{var}[y(k) - \hat{y}^f(k)] \quad (23)$$

where $y(k)$ are the observational data, $\hat{y}^f(k)$ are the observations of the state forecast (Eq. (14)), \mathbf{P}_{yy} is the covariance of the innovation (Eq. (14)) predicted by the KF, and m is the number of measurements. A consistent KF should on average satisfy the following equality (Bar-Shalom and Fortmann, 1988):

$$\hat{\sigma}_{inov}^2(k) = \sigma_{inov}^2(k). \quad (24)$$

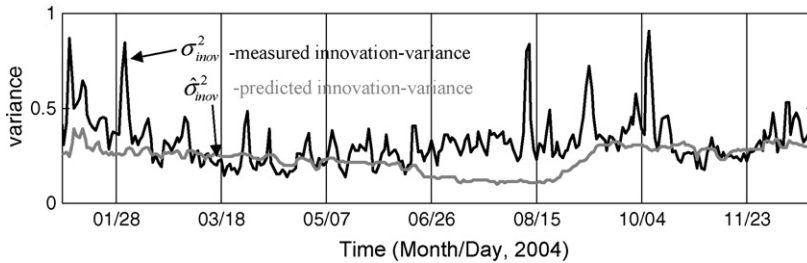


Fig. 14. Measured (black line) and expected (gray line) variance of the innovation signal for the DA experiment DA-all. The low-pass ($T > 30$ h) filter was used to improve the visual representation of the time series.

Fig. 14 compares the time series of the measured σ_{inov}^2 (black line) and the predicted $\hat{\sigma}_{inov}^2$ (gray line) error variances for the DA experiment DA-all.

On average, KF was able to track the magnitude of the measured error variance well. The average measured error variance σ_{inov}^2 was 0.3 and the predicted error variance $\hat{\sigma}_{inov}^2$ was 0.2. However, our KF failed to represent the high-frequency variability in measured error, warranting further tuning of the error models in the designed KF.

6.5.2. Conservation laws

To determine whether the DA estimates were dynamically consistent, we evaluated the conservation properties of the DA estimates. Obeying conservation laws, such as mass, volume, momentum, and transport conservation, is an important attribute of a numerical algorithm and is crucial for the conservative transport of scalar tracers such as salinity, nutrients, and pollutants. The conservation of mass is enforced in a forward model, such as SELFE, by the deliberate design of the numerical algorithms. However, at the time of our experiments, neither the EOF projection, nor the model surrogate, nor the DA algorithm were designed to deliberately enforce conservation laws.

To quantify the conservation errors in the DA estimates, we concentrated on volume conservation and transport conservation laws. A numerical algorithm is called *volume-conservative* when the flux through the boundary of a control volume is equivalent to the change in the volume of a control volume (via movement of the free surface). A numerical algorithm is called *transport-conservative* when the flux of a tracer (e.g., salinity) through the boundary of a control volume is equivalent to the change in the tracer occurring inside a control volume. To evaluate the conservation quantities in the year-long DA experiment DA-all, we used the entire domain of DB16 as the control volume.

Volume conservation (with a low⁶ conservation error of 0.7%) was well preserved in SELFE. Unlike the states in the SELFE model, the states reconstructed from the EOF projection and from the DA estimate had a slightly larger error: 1.6% for the EOF and 6% for the DA experiment. The larger error for the DA experiment was a likely indicator of inconsistencies between the water levels that experienced strong corrections from the DA and the velocities that experienced little change from the DA.

Transport conservation was also well preserved in SELFE, leading to a small conservation error of 1.3%. The states reconstructed from the EOF projection had a similar error of 1.4%, possibly indicating a very low reconstruction error for the velocity and the salinity fields. Unlike DB16 and the EOF, the DA results had a larger conservation error of 15%, possibly suggesting that some physical processes in the estuary were poorly resolved in DB16, such as influence of salinity storage in tidal flats and oscillations due to internal waves.

⁶ The conservation error was even lower when it was computed using model variables internal to the SELFE numerics (e.g., the velocities defined at side centers and the water levels at element centers). However, only the external representation of these variables was available for our experiments (e.g., defined on the nodes of the computational grid), which led to a higher conservation error.

7. Conclusions

In applying the reduced-dimension KF to a realistic simulation of the Columbia River estuary, we determined that:

- (1) The novel framework of the reduced-dimension KF can be applied to estimate the state of a complex, nonlinear coastal circulation, such as the Columbia River estuary.
- (2) In the Columbia River estuary, our DA method improved upon the non-assimilative model in simulating the long-term variability of tides, salinity, and temperature.
- (3) The reduced-dimension KF was successful in a real-time nowcast of the Columbia River estuary, without the need to retrain the model surrogates and the EOF basis.
- (4) Implementing our DA method using the reduced-dimension KF led to substantial computational and logistical economies. Computationally, our DA method was ~ 100 times faster than the forward model and ~ 100 – $10,000$ times faster than most existing DA algorithms. Logistically, our DA method did not require a tedious development of the tangent linear and adjoint models.

The results to date show the robustness of our tools and their potential for applications in other ocean settings. Successful implementation of the reduced-dimension KF in other ocean settings will require the training of an accurate model surrogate and a representative EOF space, just as we trained them in the Columbia River estuary. A forward model with a high level of skill in representing the statistics of the modeled system will be necessary to provide training samples. Linear theory suggests that training samples from a forward model should cover the range of forcings expected during applications of the trained model surrogates, however extensions that will accommodate nonlinear dynamics of the ocean may be necessary (van der Merwe et al., 2007).

Acknowledgments

We thank the anonymous reviewers, whose comments helped us improve the quality of this paper. We thank Michael Wilkin for his work on maintaining the CORIE observational network, Dr. Yinglong Zhang for his work on the SELFE model and DB16, Dr. Ibrahim Hoteit for extended suggestions on the early version of the manuscript, and Dr. Sandra Oster for her editing of the text. SF is grateful to the members of his thesis committee—Drs. Antônio Baptista, Todd Leen, Robert Miller, Mike Foreman, and John Kindle—for their guidance, support, and constructive feedback.

The National Science Foundation (ACI-0121475, OCE-0424602) and National Oceanic and Atmospheric Administration (AB133F-04-CN-0033) provided financial support for this research. Any statements, opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or policies of the federal sponsors, and no official endorsement should be inferred.

References

- Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H. (Eds.), 2000. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia.
- Baptista, A.M., 2002. Environmental observation and forecasting systems. In: Meyers, R.A. (Ed.), *Encyclopedia of Physical Science and Technology*, vol. 5. Academic Press, pp. 565–581.
- Baptista, A.M., 2006. CORIE: the first decade of a coastal margin collaborative observatory. In: *Oceans'06 MTS/IEEE*, Boston, MA.
- Baptista, A.M., Zhang, Y.L., Chawla, A., Zulauf, M., Seaton, C., Myers III, E.P., Kindle, J., Wilkin, M., Burla, M., Turner, P.J., 2005. A cross-scale model for 3D baroclinic circulation in estuary-plume-shelf systems. II. Application to the Columbia River. *Continental Shelf Research* 25, 935–972.
- Bar-Shalom, Y., Fortmann, T.E., 1988. *Tracking and Data Association*. Academic Press, New York, NY, USA.
- Bennett, A.F., 1992. *Inverse Methods in Physical Oceanography*. Cambridge University Press, NY.
- Bertino, L., Evensen, G., Wackernagel, H., 2002. Combining geostatistics and Kalman filtering for data assimilation in an estuarine system. *Inverse Problems* 18, 1–23.
- Bishop, C., 1995. *Neural Networks for Pattern Recognition*. Oxford University Press.
- Bottom, D.L., Simenstad, C.A., Burke, J., Baptista, A.M., Jay, D.A., Jones, K.K., Casillas, E., Schiewe, M.H., 2005. *Salmon at River's End. The Role of the Estuary in the Decline and Recovery of Columbia River Salmon*. NOAA Technical Memorandum NMFS-NWFSC-68, U.S. Dept. Commer.

- Burla, M., Baptista, A.M., Zhang, Y.-L., Casillas, E., Bottom, D.L., Simenstad, S.A., 2007. Salmon habitat opportunity in the Columbia River estuary: modeling the physical environment to inform management decisions. In: *Proceedings of the Coastal Zone 07*, Portland, OR.
- Cane, M.A., Kaplan, A., Miller, R.N., Tang, B., Hackert, E., Busalacchi, A.J., 1996. Mapping tropical Pacific sea level: data assimilation via a reduced state space Kalman filter. *Journal of Geophysical Research* 101 (C10), 22599–22617.
- Chawla, A., Jay, D.A., Baptista, A.M., Wilkin, M., Seaton, C., 2008. Seasonal variability and estuary–shelf interactions in circulation dynamics of a river-dominated estuary. *Estuaries and Coasts* 31, 269–288, doi:10.1007/s12237-007-9022-7.
- Clark, H.L., Isern, A., 2003. The OOI and the IOOS—Can they be differentiated? An NSF perspective. *Oceanography* 16, 20–21.
- Dee, D.P., 1995. On-line estimation of error covariance parameters for atmospheric data assimilation. *Monthly Weather Review* 123, 1128–1144.
- Evensen, G., 2003. The Ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics* 53 (4), 343–367.
- Frolov, S., 2007. Enabling technologies for data assimilation in a coastal-margin observatory. Ph.D. Thesis. Oregon Health & Science University, Environmental & Biomolecular Systems, also available at <http://drl.ohsu.edu/cgi-bin/showfile.exe?CISOROOT=/etd&CISOPTR=262&filename=66281815112007.200711.frolov.sergey.pdf>.
- Frolov, S., Baptista, A.M., Zhang, Y.-L., Seaton, C., in press. Estimation of ecologically significant circulation features of the Columbia River estuary and plume, using a reduced-dimension Kalman filter. *Continental Shelf Research*.
- Gandin, L., 1963. *Objective Analysis of Meteorological Fields*. Hydrometeoizdat, Leningrad.
- Golub, G.H., Van Loan, C.F., 1996. *Matrix Computations*, 3rd ed. Johns Hopkins University Press, Baltimore, MD.
- Hamilton, P., 1990. Modelling salinity and circulation for the Columbia River estuary. *Progress in Oceanography* 25, 113–156.
- Heemink, A.W., Mouthaan, E.E.A., Roest, M.R.T., Vollebregt, A.E.H., Robaczewska, K.B., Verlaan, M., 2002. Inverse 3D shallow water flow modelling of the continental shelf. *Continental Shelf Research* 22, 465–484.
- Holmes, P., Lumley, J., Berkooz, G., 1996. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry* (Cambridge Monographs on Mechanics). Cambridge University Press, Cambridge.
- Hoteit, I., Pham, D.T., 2003. Evolution of the reduced state space and data assimilation schemes based on the Kalman filter. *Journal of Meteorological Society Japan* 81, 21–39.
- Jay, D.A., Geyer, W.R., Uncles, R.J., Vallino, J., Largier, J., Boynton, W.R., 1997. A review of recent developments in estuarine scalar flux estimation. *Estuaries* 20 (2), 262–280.
- Jay, D.A., Smith, J.D., 1990. Circulation, density distribution and neap-spring transition in the Columbia River Estuary. *Progress in Oceanography* 25, 81–112.
- Jolliffe, I.T., 1986. *Principal Component Analysis*. Springer-Verlag.
- Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering* 82 (D), 35–45.
- Kurapov, A.L., Egbert, G.D., Miller, R.N., Allen, J.S., 2002. Data assimilation in a baroclinic coastal ocean model: ensemble statistics and comparison of methods. *Monthly Weather Review* 130, 1009–1025.
- Le Dimet, F.-X., Talagrand, O., 1986. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A* 38, 97–110.
- Lermusiaux, P.F.J., Robinson, A.R., 1999. Data assimilation via error subspace statistical estimation, part I: theory and schemes. *Monthly Weather Review* 127 (8), 1385–1407.
- Lu, Z., Leen, T.K., van der Merwe, R., Frolov, S., Baptista, A.M., submitted for publication. Sequential data assimilation with sigma-point Kalman filter on low-dimensional manifold. *Ocean Modeling*. Also available as a technical report at <http://www.stccmop.org/files/CMOP-TR-07-001.pdf>.
- Martin, D.L., 2003. The National Oceanographic Partnership Program, Ocean US, and real movement towards an integrated and sustained ocean observing system. *Oceanography* 16, 13–19.
- Martin, P.J., 2000. Description of the Navy Coastal Ocean Model.
- Møller, M., 1996. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6 (4), 525–533.
- Nabney, I.T., 2004. *NETLAB: Algorithms for Pattern Recognition*. Springer, London, New York.
- Oke, P.R., Allen, J.S., Miller, R.N., Egbert, G.D., Kosro, P.M., 2002. Assimilation of surface velocity data into a primitive equation coastal ocean model. *Journal of Geophysical Research* 107 (C9), 3122.
- ORION, 2005. *Ocean Observatories Initiative Science Plan*, Washington, DC.
- Pham, D.T., 2001. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review* 129 (5), 1194–1207.
- Pham, D.T., Verron, J., Roubaud, M.C., 1998. A singular evolutive extended Kalman filter for data assimilation in oceanography. *Journal of Marine Systems* 16, 323–340.
- USACE, 2001. *Columbia River Channel Improvement Project: 2001 Biological Assessment*. U.S. Army Corps of Engineers.
- van der Merwe, R., 2002–2006. *ReBEL: Recursive Bayesian Estimation Library*. Available at <http://choosh.ece.ogi.edu/rebel/>.
- van der Merwe, R., 2004. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. Ph.D., OHSU.
- van der Merwe, R., Leen, T.K., Lu, Z., Frolov, S., Baptista, A.M., 2007. Fast neural network surrogates for very high dimensional physics-based models in computational oceanography. *Neural Networks* 20, 462–478, doi:10.1016/j.neunet.2007.04.023.
- van der Merwe, R., Wan, E., 2003. Sigma-point kalman filters for probabilistic inference in dynamic state-space models. In: *Workshop on Advances in Machine Learning*, Montreal, Canada.
- Wilkin, J.L., Arango, H.G., Haidvogel, D.B., Lichtenwalner, C.S., Glenn, S.M., Hedström, K.S., 2005. A regional ocean modeling system for the long-term ecosystem observatory. *Journal of Geophysical Research* 110, C06S91.
- Wolter, K., Timlin, M.S., 1993. Monitoring ENSO in COADS with a seasonally adjusted principal component index. In: *Proceedings of the 17th Climate Diagnostics Workshop*. NOAA/N MC/CAC, NSSL, Oklahoma Clim. Survey, CIMMS and the School of Meteor., Univ. of Oklahoma, Norman, OK, pp. 52–57.
- Wolter, K., Timlin, M.S., 1998. Measuring the strength of ENSO events—how does 1997/98 rank? *Weather* 53, 315–324.

- Xu, J., Shenn-Yu, C., Raleigh, H.R., Wang, H.V., Boicourt, W.C., 2002. Assimilating high-resolution salinity data into a model of a partially mixed estuary. *Journal of Geophysical Research* 107 (C7), 11–111.
- Xue, Y., Cane, M.A., Zebiak, S.E., 1997. Predictability of a coupled model of ENSO using singular vector analysis. Part I. Optimal growth in seasonal background and ENSO cycles. *Monthly Weather Review* 125, 2043–2056.
- Zhang, Y.-L., Baptista, A.M., 2008. SELFE: a semi-implicit Eulerian–Lagrangian finite-element model for cross-scale ocean circulation. *Ocean Modeling* 21 (3–4), 71–96, doi:10.1016/j.ocemod.2007.11.005.
- Zhang, Y.-L., Baptista, A.M., Myers, E.P., 2004. A cross-scale model for 3D baroclinic circulation in estuary-plume-shelf systems. I. Formulation and skill assessment. *Continental Shelf Research* 24, 2187–2214.