Two-sample Kalman filter with HiRLAM and UKMO difference

Julius H. Sumihar*, Martin Verlaan and Arnold W. Heemink

Delft University of Technology, The Netherlands

*Corresponding author address: Julius H. Sumihar, Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands. Post: P.O. Box 5031, 2600 GA Delft, The Netherlands

E-mail: j.h.sumihar@ewi.tudelft.nl

Abstract

In this paper, an application of the two-sample Kalman filter algorithm is presented. The algorithm is implemented with an operational storm surge forecast model, the Dutch Continental Shelf Model (DCSM). Two different wind fields, HiRLAM and UKMO, are used to generate two different forecast samples for computing forecast error covariance matrix. Since the error in the wind fileds is usually believed to be correlated in time, the implementation of the algorithm requires some modification. First, the two samples are transformed in such a way to construct a system representation, which is driven by white noise. Second, since the transformation yeileds two dependent samples, the algorithm is slightly modified to work around this. The transformation and the modified algorithm are verified by implementing them with a simple one-dimensional wave model, where the results are found consistent with the known generated reference solution. At the end, the modified algorithm and the transformation are implemented with the DCSM. The experiments yield gain estimates which become smoother at each iteration and converge at the third step. Evaluation is performed by implementing the estimated gain in a data assimilation system to check if the assimilation has brought the model closer to measurements. Evaluation is also done by comparing the results at observation stations where data is not used in assimilation. The results show that the assimilation has succeeded in bringing the model closer to data in all stations.

1. Introduction

Most data assimilation algorithms require explicit specification of forecast and observation error statistics as primary quantities used to produce an analysis. The forecast and observation errors define the relative weight given to each observation. In practice it is very difficult to have an idea about the forecast error statistics, since the true state is never known. Without access to the true state, it is impossible to produce samples of forecast error.

There are two possible ways to approach the problem of specifying forecast error statistics. The first way is to attempt to disentangle information about the error statistics from the difference between forecast and observation. It requires a prior fitting of some stochastic parameters, such as correlation length or standard deviation. This approach is sometimes called the covariance matching method (e.g. Verlaan 1998). The second way is to try to find a surrogate quantity whose statistics are expected to be similar to those of the unknown system error statistics (Fisher 2003). The most popular choice of a surrogate quantity for forecast error in numerical weather prediction application is to use the difference between forecasts of different length that verify at the same time, frequently referred to as the NMC method (Parrish and Derber, 1992; Devenyi, et.al., 2004; Yang, et.al., 2006). The diffculty with this approach is that the statistics of any chosen surrogate quantity are likely to differ in some aspects from the true error statistcs. However, it has a practical advantage that once such forecast samples have been generated, it is straightforward to determine the forecast error covariance for the entire model domain, in terms of all model variables (Fisher 2003).

In oceanography, another increasingly popular choice to generate a surrogate quantity for forecast error is to use two different wind fields, produced by two different atmospheric models, as input to an oceanic model under study Borovikov et al. 2005, Alves and Robert 2005,

Leeuwenburgh 2007). In their works, the two different wind fields are used to produce estimates of forecast error covariances in the framework of ensemble Kalman filtering. Similar to them, we also use two different wind fields, as input to a storm surge forecast model, to generate proxies for forecast errors. The difference is that in our approach we use it in the context of sequential steady-state data assimilation.

This paper presents an application of the two-sample Kalman filter using an operational storm surge prediction model: the Dutch Continental Shelf Model (DCSM). In this study, the two forecast samples are generated by running the DCSM with two different wind forcings, namely HiRLAM and UKMO. Since error in the wind forcing is generally believed to be correlated in time, it is necessary to tranform the system representation such that it is driven by white noise. The transformation of the system representation as well as the necessary modification of the algorithm are explained in Section 2. Section 3 presents an experiment using a simple one-dimensional wave model to verify if the modified algorithm as well as the transformation work. The experiments using DCSM are presented and discussed in Section 4. The paper is concluded briefly in Section 5.

2. Estimating covariance using two samples of a random process

a. Introduction

The two-sample Kalman filter algorithm requires the system of interest to be driven by white noise process. However, in certain application, such as storm surge prediction, it is generally

believed that the noise process disturbing the input forcing is correlated in time. In that case, we need to transform the model representation such that it is driven only by white noise. In this section we describe briefly a method for such transformation in the case of when the noise is governed by an AR(1) process. Moreover, as we will see that this transformation produces two *dependent* samples, we need to modify slightly the algorithm described in Sumihar,et.al.[2008]. This section describes also the modification performed on the two-sample Kalman filter algorithm.

b. Samples transformation

Suppose that we have a dynamic system Φ driven by an uncertain input forcing $\hat{\mathbf{u}}(t_k) \in \mathcal{R}^q$ as the following

$$\mathbf{z}(t_{k+1}) = \Phi\mathbf{z}(t_k) + \mathbf{F}\hat{\mathbf{u}}(t_k) \tag{1}$$

$$\hat{\mathbf{u}}(t_k) = \mathbf{u}^t(t_k) + \hat{\mathbf{w}}(t_k) \tag{2}$$

where $\mathbf{z}(t_k) \in \mathcal{R}^p$ denotes the system state at time t_k, $\mathbf{u}^t(t_k) \in \mathcal{R}^q$ is the unknown *true* input vector and $\hat{\mathbf{w}}(t_k) \in \mathcal{R}^q$ is a vector representing the uncertainty in the input forcing, while $\mathbf{F}$ is the $p \times q$ input forcing matrix. Suppose also that the noise vector $\hat{\mathbf{w}}(t_k)$ in equation (2) is governed by an AR(1) process

$$\hat{\mathbf{w}}(t_k) = \mathbf{A}\hat{\mathbf{w}}(t_{k-1}) + \hat{\eta}(t_k) \tag{3}$$

where $\mathbf{A}$ is a time-invariant $p \times p$ diagonal matrix whose elements determine the correlation-time, and $\hat{\eta}(t_k) \in \mathcal{R}^p$ is a white noise vector with constant covariance $\hat{\mathbf{Q}} \in \mathcal{R}^{p\times p}$. Assuming that the system operates in steady-state, it is easy to show that the following properties hold:

$$\mathsf{E}[\hat{\mathbf{u}}(t_k)] = \mathbf{u}^t(t_k) \tag{4}$$

$$\mathsf{E}[\hat{\mathbf{w}}(t_k)] = \mathbf{0} \tag{5}$$

$$\hat{\mathbf{P}} = \mathbf{A}\hat{\mathbf{P}}\mathbf{A}^\top + \hat{\mathbf{Q}} \tag{6}$$

where $\hat{\mathbf{P}}$ is the steady-state covariance matrix of the noise vector $\hat{\mathbf{w}}(t_k)$, defined as $\hat{\mathbf{P}} = \mathsf{E}[\hat{\mathbf{w}}(t_k)\hat{\mathbf{w}}(t_k)^\top]$.

Note that the colored noise $\hat{\mathbf{w}}(t_k)$ is a means for modelling the deviation of the input forcing, $\hat{\mathbf{u}}(t_k)$, from its true value, $\mathbf{u}^t(t_k)$. We are considering the case where we actually do not have separate realizations of $\hat{\mathbf{w}}(t_k)$ and $\hat{\mathbf{u}}(t_k)$. In this case, what we have instead is the samples of $\hat{\mathbf{u}}(t_k)$, which contain the combination of the true value $\mathbf{u}^t(t_k)$ and the deviations $\hat{\mathbf{w}}(t_k)$.

The objective of the transformation presented in this section is to obtain another representation of the dynamics (1)-(2) such that it is driven by white noise process. The standard approach of working with a system driven by colored-noise is to augment the state vector $\mathbf{z}(t_k)$ with the noise vector $\hat{\mathbf{w}}(t_k)$. However, since we do not have the realizations of $\hat{\mathbf{w}}(t_k)$ separately from those of $\hat{\mathbf{u}}(t_k)$, we can not use this approach directly. In the following we propose a method to find an approximate of $\hat{\mathbf{w}}(t_k)$ by utilizing two samples of $\hat{\mathbf{u}}(t_k)$. These in turn will be used in state augmentation.

Suppose we have two independent realizations of the input forcing $\hat{\mathbf{u}}(t_k)$, namely $\hat{\mathbf{u}}_1(t_k)$ and $\hat{\mathbf{u}}_2(t_k)$. Using these two input forcing realizations, we introduce

$$\mathbf{u}(t_k) \equiv \frac{1}{2}(\hat{\mathbf{u}}_1(t_k) + \hat{\mathbf{u}}_2(t_k)) \tag{7}$$

$$\tilde{\mathbf{w}}_1(t_k) \equiv \sqrt{2}(\hat{\mathbf{u}}_1(t_k) - \mathbf{u}(t_k)) \tag{8}$$

$$\tilde{\mathbf{w}}_2(t_k) \equiv \sqrt{2}(\hat{\mathbf{u}}_2(t_k) - \mathbf{u}(t_k)). \tag{9}$$

The term $\mathbf{u}$ represents the average of the two input realizations. On the other hand, $\tilde{\mathbf{w}}_1$ and $\tilde{\mathbf{w}}_2$ can be shown to be resulting from linear combinations of the unknown $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$

$$\tilde{\mathbf{w}}_1(t_k) = \frac{1}{\sqrt{2}}(\hat{\mathbf{w}}_1(t_k) - \hat{\mathbf{w}}_2(t_k)) \tag{10}$$

$$\tilde{\mathbf{w}}_2(t_k) = \frac{1}{\sqrt{2}}(\hat{\mathbf{w}}_2(t_k) - \hat{\mathbf{w}}_1(t_k)) \tag{11}$$

and we call them the *transformed colored noise processes*. Another consequence of definition (8)-(9) is that the transformed colored noise processes evolve in time according to

$$\tilde{\mathbf{w}}_1(t_k) = \mathbf{A}\tilde{\mathbf{w}}_1(t_{k-1}) + \tilde{\eta}_1(t_{k-1}) \tag{12}$$

$$\tilde{\mathbf{w}}_2(t_k) = \mathbf{A}\tilde{\mathbf{w}}_2(t_{k-1}) + \tilde{\eta}_2(t_{k-1}) \tag{13}$$

which is similar to the unknown original colored noise process (3). We call $\tilde{\eta}_1(t_k)$ and $\tilde{\eta}_2(t_k)$ as the *transformed white noise processes*, which can be shown to be a combination of the unknown original $\hat{\eta}_1(t_k)$ and $\hat{\eta}_2(t_k)$

$$\tilde{\eta}_1(t_k) = \frac{1}{\sqrt{2}}(\hat{\eta}_1(t_k) - \hat{\eta}_2(t_k)) \tag{14}$$

$$\tilde{\eta}_2(t_k) = \frac{1}{\sqrt{2}}(\hat{\eta}_2(t_k) - \hat{\eta}_1(t_k)) \tag{15}$$

Using these new variables, we define the transformed input forcings as

$$\tilde{\mathbf{u}}_1(t_k) = \mathbf{u}(t_k) + \tilde{\mathbf{w}}_1(t_k) \tag{16}$$

$$\tilde{\mathbf{u}}_2(t_k) = \mathbf{u}(t_k) + \tilde{\mathbf{w}}_2(t_k) \tag{17}$$

It can be shown that these variables possess the following properties

$$\mathsf{E}[\tilde{\mathbf{u}}(t_k)] = \mathbf{u}^t(t_k) \tag{18}$$

$$\mathsf{E}[\tilde{\mathbf{w}}(t_k)] = \mathbf{0} \tag{19}$$

$$\mathsf{E}[\tilde{\mathbf{w}}(t_k)\tilde{\mathbf{w}}(t_k)^\top] = \mathsf{E}[\hat{\mathbf{w}}(t_k)\hat{\mathbf{w}}(t_k)^\top] = \hat{\mathbf{P}} \tag{20}$$

$$\mathsf{E}[\tilde{\eta}(t_k)\tilde{\eta}(t_k)^\top] = \mathsf{E}[\hat{\eta}(t_k)\hat{\eta}(t_k)^\top] = \hat{\mathbf{Q}} \tag{21}$$

which demonstrate that the statistics of the noise process are preserved under this transformation. This indicates that it is possible to work with $\tilde{\mathbf{u}}_1(t_k)$ and $\tilde{\mathbf{u}}_2(t_k)$, instead of to work with $\hat{\mathbf{u}}_1(t_k)$ and $\hat{\mathbf{u}}_2(t_k)$.

Since under this transformation the realizations of $\mathbf{u}(t_k)$ and $\tilde{\mathbf{w}}(t_k)$ are available separately, we can now use them in state augmentation. Denoting $\mathbf{x}(t_k) \equiv \begin{bmatrix} \mathbf{z}(t_k) \\ \tilde{\mathbf{w}}(t_k) \end{bmatrix}$, $\eta(t_k) \equiv \begin{bmatrix} \mathbf{0} \\ \tilde{\eta}(t_k) \end{bmatrix}$, $\mathbf{M} \equiv \begin{bmatrix} \Phi & \mathbf{F} \\ \mathbf{0} & \mathbf{A} \end{bmatrix}$, and $\mathbf{B} \equiv \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}$, we can rewrite the dynamical system (1)-(2) as

$$\mathbf{x}(t_{k+1}) = \mathbf{M}\mathbf{x}(t_k) + \mathbf{B}\mathbf{u}(t_k) + \eta(t_k) \tag{22}$$

Now we have a dynamical system representation, which is driven by white noise process.

c. Algorithm

The two-sample Kalman filter algorithm described in Sumihar, et.al. [2008] is an iterative procedure for computing forecast error covariance, which is used to define a steady-state Kalman gain. In computing the covariance, this algorithm makes use of two *independent* samples. If the two samples are dependent, the covariance will be either underestimated or overestimated depending on whether the samples are poisitively or negatively correlated. However, under the transformation described in Section b, since $\tilde{\eta}_1$ and $\tilde{\eta}_2$ are dependent, the two independent realizations of $\hat{\mathbf{u}}(t_k)$ become two *dependent* realizations of $\tilde{\mathbf{u}}(t_k)$. Therefore, we need to adapt slightly the algorithm to work in this situation. In this section, we explain the modification on the two-sample Kalman filter algorithm used in the study.

1) COMPUTING COVARIANCE USING ONE SAMPLE AND CENTRAL MODE

Consider a discrete random process $\mathbf{X}^f(t_k) \in \mathcal{R}^n$ with constant covariance but varying-in-time mean, from which the approximate of its central mode $\bar{\mathbf{x}}^f(t_k)$ and a sample $\mathbf{x}^f(t_k)$ are

available. We define $\mathbf{e}_{1,2}(t_k)$ as the difference between these two series:

$$\mathbf{e}_{1,2}(t_k) \equiv \mathbf{x}^f(t_k) - \bar{\mathbf{x}}^f(t_k) \tag{23}$$

From the definition above we can show that $\mathbf{e}_{1,2}(t_k)$ has the following properties:

$$\mathsf{E}[\mathbf{e}_{1,2}(t_k)] = 0 \tag{24}$$

$$\mathsf{E}[\mathbf{e}_{1,2}(t_k)\mathbf{e}_{1,2}^\top(t_k)] = \mathbf{P}^f \tag{25}$$

where $\mathbf{P}^f \in \mathcal{R}^{n\times n}$ is the covariance of random process $\mathbf{X}^f(t_k)$ defined as $\mathbf{P}^f = \mathsf{E}[(\mathbf{X}^f(t_k) - \mathsf{E}[\mathbf{X}^f(t_k)])(\mathbf{X}^f(t_k) - \mathsf{E}[\mathbf{X}^f(t_k)])^\top]$.

Given a sample $\mathbf{e}_{1,2}(t_k)$, $k = 1, ..., N$, and assuming that the time interval is sufficiently long that they are reasonably independent of each other, we can calculate the covariance by averaging the realization over time:

$$\mathbf{P}^f = \frac{1}{N-1}\sum_{i=1}^{N} \mathbf{e}_{1,2}(t_k)\mathbf{e}_{1,2}^\top(t_k) \tag{26}$$

2) PROPOSED ALGORITHM

Suppose we have a dynamic system (1) - (2), where two uncertain input forcing, $\hat{\mathbf{u}}_1(t_k)$ and $\hat{\mathbf{u}}_2(t_k)$, are available. The first step to be done is to compute and store the average of the input forcing $\mathbf{u}(t_k)$ using equation (7). The second step is to compute and store the approximate of white noise process $\tilde{\eta}_1(t_k)$ using equations (8) and (12) or $\tilde{\eta}_2(t_k)$ using equations (9) and (13). It is now possible to rewrite the dynamics into the form of equation (22). Suppose also that the observation $\mathbf{y}^o(t_k) \in \mathcal{R}^m$ of the dynamical system (22) is available and we assume that the observation is related to the unknown true state $\mathbf{x}^t(t_k)$ through

$$\mathbf{y}^o(t_k) = \mathbf{H}\mathbf{x}^t(t_k) + \mathbf{V}(t_k) \tag{27}$$

where $\mathbf{H} \in \mathcal{R}^{m\times n}$ is a linear observation operator and $\mathbf{V}(t_k) \in \mathcal{R}^m$ is a white noise vector with a constant covariance $\mathbf{R} \in \mathcal{R}^{m\times m}$ representing observation uncertainty. Combining the model and observation in the way which minimizes the error covariance, we can write the governing equation for $\mathbf{X}^f(t_k)$ as

$$\mathbf{X}^f(t_{k+1}) = \mathbf{M}\mathbf{X}^a(t_k) + \mathbf{B}\mathbf{u}(t_k) + \eta(t_k) \tag{28}$$

$$\mathbf{X}^a(t_{k+1}) = \mathbf{X}^f(t_{k+1}) + \mathbf{K}(\mathbf{y}^o(t_{k+1}) - \mathbf{H}\mathbf{X}^f(t_{k+1}) - \nu(t_{k+1})) \tag{29}$$

where $\mathbf{K}$ is a constant matrix given by

$$\mathbf{K} = \mathbf{P}^f\mathbf{H}^\top(\mathbf{H}\mathbf{P}^f\mathbf{H}^\top + \mathbf{R})^{-1} \tag{30}$$

The basic idea underlying the modification of the two-sample Kalman filter algorithm is to use one forecast sample $\mathbf{x}^f(t_k)$ and an approximate of central mode $\bar{\mathbf{x}}^f(t_k)$, instead of using two independent samples, of the dynamics (28)-(29) for computing forecast error covariance. The forecast sample is generated by running the model (28)-(29) for certain period and perturbing the run by either one of the approximate white noise processes $\tilde{\eta}_1(t_k)$ and $\tilde{\eta}_2(t_k)$. On the other hand, the central mode is obtained by running the model while simply setting all the noise terms to zero for all time t_k, since all the noise variable is assumed to have mean equal zero. Given one sample and the central mode, the covariance can be computed using equation (26). The forecast error covariance matrix is then used to compute the Kalman gain by using equation (30).

The algorithm consists of generating repeatedly one sample and the central mode of the dynamics (28)-(29), using them to compute the forecast error covariance $\mathbf{P}^f$, and using $\mathbf{P}^f$ to determine the Kalman gain $\mathbf{K}$. The algorithm should be initialized by an initial guess of $\mathbf{K}$. Basically, any initial guess of $\mathbf{K}$ can be chosen such that it stabilizes the overall dynamics.

However, for the applications that we foresee, we have found that $\mathbf{K} = \mathbf{0}$ is a good initial guess. For accordance with our previous work, we call throughout the step with $\mathbf{K} = \mathbf{0}$ as the open-loop step.

To make the description complete, we present below a brief summary of the algorithm:

- Initialization:
  - compute and store input forcing average using equation (7): $\mathbf{u}(t_k)$, $k = 1, ..., N$
  - compute and store approximate white noise process using equation (8) and (12) or equation(9) and (13): $\tilde{\eta}_1(t_k)$ or $\tilde{\eta}_2(t_k)$, $k = 1, ..., N$
  - specify initial guess of Kalman gain $\mathbf{K}$
- Closed-loop step:
  - insert $\mathbf{K}$ into equation (29)
  - generate one realization and the central mode of dynamics (28): $\mathbf{x}^f(t_k)$ and $\bar{\mathbf{x}}^f(t_k)$, $k = 1, ..., N$
  - estimate $\mathbf{P}^f$ using equation (26)
  - estimate $\mathbf{K}$ using equation (30)
  - repeat closed-loop step until $\mathbf{P}^f$, and therefore also $\mathbf{K}$, have converged

3. Experiments with a simple wave model

Two experiments are presented and discussed in this section. The first experiment is for demonstrating the problem which occurs if the two samples used for computing forecast error

covariance are not independent. On the other hand, the second experiment is for verifying if the transformation of the system representation as well as the modified algorithm work well. Both experiments are performed using a simple one-dimensional wave model. This model is a simplified version of the De St. Venant equation, which can be used for modelling a tidal flow in a long and narrow estuary. The governing equations of the one-dimensional wave is as follows:

$$\frac{\partial \xi}{\partial t} + D\frac{\partial u}{\partial x} = 0 \tag{31}$$

$$\frac{\partial u}{\partial t} + g\frac{\partial \xi}{\partial x} + \frac{cu}{D} = 0 \tag{32}$$

$$\xi(x = 0, t) = \xi_b(t) \tag{33}$$

$$\xi(x, t = 0) = 0 \tag{34}$$

$$u(x, t = 0) = 0 \tag{35}$$

$$u(x = L, t) = 0 \tag{36}$$

where $\xi(x, t)$ is the water level above the reference plane, $u(x, t)$ is the average current velocity over a cross section, t is the time, x is the position along the estuary, D is the water depth, g is the gravitational acceleration and c is a friction constant. For this study, the system of equations (31)-(36) is solved numerically using a Crank-Nicholson scheme with implicity factor $\theta = 0.4$ and the discretization in space is based on a staggered grid.

In this experiment, the following values for the parameters are used: D = 10 meter, $c = 0.0002s^{-1}$, L=60 km, time step Δt=1 minutes, and number of grid points N = 80 for the spatial discretization, hence $\Delta x = \frac{L}{N-1}$. Moreover, at the sea-estuary boundary we assume that the water level is periodic according to $\xi_b(t_k) = 0.5\sin(\frac{2\pi}{T}t_k)$, where T=3 hours.

The uncertainty of the model is assumed to be due to the inaccuracy of the set-up at the sea-estuary boundary. An additive colored noise modelled as an AR(1) process is used to represent

this uncertainty:

$$w_b(t_k) = \alpha_b w_b(t_{k-1}) + \epsilon(t_k) \tag{37}$$

where α_b determines the correlation time, which in this experiment is set to 2 hours and ϵ is a Gaussian white noise process with standard deviation 20 cm.

The experiment setup for illustrating the effect of sample dependence on the gain estimate using the original two-sample Kalman filter is as follows. One realization of the wave process is generated for a period of 600 hours simulation time to serve as the 'truth'. A series of artificial measurements of water level at the right-end of the estuary, i.e. at $x = 60$ km, is generated by adding a realization of a Gaussian white noise process to the 'truth' water level at this location. For this study, the standard deviation of the Gaussian white noise, used to represent the observation uncertainty, is 2 cm. This artificial observation is used in the closed-loop steps of the two-sample Kalman filter algorithm. At each iteration step, a pair of dependent samples are generated for the same period of 600 hours simulation time. The two samples are produced by generating two dependent realizations of the colored-noise $w_b(t_k)$ with correlation $\rho = 0.7$ and using them to drive the wave model. The original two-sample Kalman filter algorithm is applied with the two dependent samples to estimate the forecast error covariance and to use the covariance for computing the gain estimate. Moreover, another pair of independent realizations of the open-boundary noise are also generated at each step and used to compute a reference. It is already known that the two-sample Kalman filter algorithm with independent samples converges to the optimal solution for this model. Hence the gain estimates obtained by using two independent samples can serve as a reference to what we can compare the results obtained using dependent samples.

The gain estimates at each iteration step obtained by using both dependent and independent

samples are presented in Figure 1. The figures show that the gain estimates obtained using dependent samples are smaller than those obtained using independent samples. The positive correlation between the two samples causes the covariance to be underestimated. This in turn causes the gain to be underestimated as well. Figure 1 also shows that the iteration using both dependent and independent samples converges at the third step. However, the iteration using dependent samples converges to some suboptimal values. The same observation holds as well for the case of when the two samples are correlated negatively (the results are not shown). However, in this case, the covariance is overestimated and hence also the gain.

The experiment for verifying the transformation and the modified algorithm is done as follows. Using the same specifications used in the previous experiment, an artificial measurement is generated at each time step for 600 hours, but at $x = 0.3L$. At each iteration step, two independent realizations of the set-up at the sea-estuary boundary are generated. By propagating them into the dynamics, we obtain two independent samples of the system states. We call them as the *original* samples. At the same time, the two independent boundary set-up realizations are combined to produce two transformed boundary set-ups by using the method described in Section b. Propagating them into the dynamics model we have two *transformed* samples. Moreover, the approximate central mode is also generated at the same time by driving the model with the average of the two *original* boundary set-ups as input forcing and setting all white-noise terms to zero. Using the central mode and one of the *transformed* samples, the covariance of the *transformed* forecast error is computed utilizing the algorithm explained in this paper. On the other hand, using the *original* samples, the covariance of the *original* forecast error is computed utilizing the original two-sample Kalman filter algorithm. The covariance computed using the *original* samples serves as the reference to the *true* solution. The evalu-

ation is done by comparing the gain matrices obtained using the *original* samples to the ones using the *transformed* samples at each iteration step.

The gain matrices computed in this experiment are depicted in Figure 2. Pictures on the left-hand side panel exhibit the gain associated with water level, while those on the right-hand side show the gain associated with average current velocity. The dashed-line represents the gain matrices computed using the original samples, while the full-line using the transformed samples. Visual inspection on these pictures indicates that at each iteration step the gain matrices obtained using both the original and the transformed samples are very similar to each other. This should be expected from the theory. Since the model is linear, the effect of the input terms in the central mode and the *transformed* samples are cancelled out when we compute the difference between the two. This results in an error propagation model, which is driven by transformed noise. Because the transformed noise has the same statistics as the original one, the error covariance computed using both the transformed and original ones are the same. Hence, the gain matrices are also the same. Note that the gain estimates resulted from the open-loop step by using transformed samples is exactly equal with the ones from using original samples.

To make the presentation complete, we compare the gain estimated by using our algorithm to the one obtained by using RRSQRT Kalman filter (Verlaan and Heemink, 1997) as shown in Figure 2e. This gain is obtained by using 50 modes, which is enough to estimate the Kalman gain accurately. Comparing the gain estimated at the third closed-loop iteration step to the ones obtained using RRSQRT Kalman filter we see that they are similar. This shows that for this model, using the transformed sample and the algorithm proposed in this paper we can reproduce the optimal gain estimated by using steady-state Kalman filter.

4. Experiments with DCSM

a. Introduction

In this section we present and discuss some experiments, which are performed by using a real life application model, the Dutch Continental Shelf Model (DCSM). The DCSM will be described briefly and two different wind fields used to drive the DCSM to generate two model samples are also explained. The discussion continues with the estimates of some wind noise statistics. At the end, the performance of the assimilation using the method proposed in this paper is evaluated.

b. The Dutch Continental Shelf Model

The DCSM is a large-scale water level prediction and storm surge forecasting model that encloses the North West European shelf, including the British isles. It has been developed by the National Institute of Coastal and Marine Management (RIKZ). The DCSM uses the nonlinear 2D shallow water model as the governing equations written as follows

$$\frac{\partial \xi}{\partial t} + D\frac{\partial u}{\partial x} + D\frac{\partial v}{\partial y} = 0 \tag{38}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + g\frac{\partial \xi}{\partial x} - fv + \frac{gu\sqrt{u^2+v^2}}{C^2 H} - \frac{1}{\rho_w}\frac{\tau_x}{D} + \frac{1}{\rho_w}\frac{\partial p_a}{\partial x} = 0 \tag{39}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + g\frac{\partial \xi}{\partial y} + fu + \frac{gv\sqrt{u^2+v^2}}{C^2 H} - \frac{1}{\rho_w}\frac{\tau_y}{D} + \frac{1}{\rho_w}\frac{\partial p_a}{\partial y} = 0 \tag{40}$$

where:

x, y = cartesian coordinates in horizontal plane

u, v = depth-averaged currenct in x and y direction respectively

ξ = water level above the reference plane

D = water depth below the reference plane

g = gravity acceleration

f = coefficient for the Corriolis force

C = Chezy coefficient

$\tau_x, \tau_{\overline{y}}$ wind stress in x and y direction respectively

ρ_w = density of sea water

p_a = air pressure at the surface

These equations are discretized using an Alternating Directions Implicit (ADI) method and a staggered grid that is based on the method of Leenderstse (1967) and Stelling (1984), with spherical grid (e.g. Verboom et al. 1992). The DCSM covers the area of the north-east European continental shelf, i.e. 12^oW to 13^oE and 48^oN to 62^oN as shown in Figure (3). The resolution of the spherical grid is $1/8^o \times 1/12^o$, which is approximately 8×8 km.

The forecast error in DCSM is assumed to come from the uncertain wind input. To account for this, an error term should be added to the governing equations. There are three possible ways of doing this. We may try to add an error term to the wind velocity, to the wind stress or to the depth-average water current. The operational system represents this uncertainty as an error term added to the depth-average water current. The temporal correlatedness of the wind noise is accounted for by using an AR(1) model. Moreover, the noise is assumed to be isotropic, that is the spatial correlation length is independent from the flow direction.

In this study, AR(1) model is also used to represent the time correlatedness of the wind noise. However, we model the uncertain wind input as an error term added to the wind stress. Moreover, we do not assume anything about the structure of the spatial correlation. We do assume, however, that the difference between the two wind fields described below are characteristics of

the true error.

c. Wind input

The operational DCSM uses the forecasts of the Royal Netherlands Meteorological Institute (KNMI) high-resolution limited area model (HiRLAM) as input. Another wind field available as input to DCSM is the one from the UK Met Office (UKMO). In this study we used the wind fields from both sources and applied them to the algorithm explained in Section (c). The wind fields used in the study are the ones in the period 01 November 2004 until 28 January 2005.

Visual inspection on the two wind fields indicates that the magnitude of the UKMO wind stress tend to be smaller than that of HiRLAM. Figure (4) illustrates this condition. The picture on the left hand side shows a scattered plot of the wind stress of the two wind fields at a location above the sea area of DCSM. The full line in the picture shows the line of best fit, while the dashed line shows perfect positive correlation. Before the two wind fields can be used in the experiment, we need to make some corrections to reduce the systematic difference between the two wind fields. In our study, this is done by using linear regression on the UKMO wind stress in such a way that it has the same scale with that of HiRLAM. The picture on the right hand side in Figure (4) presents the scattered plot of wind stress at the same location after correction. This correction is done at all grid points on the wet area of DCSM. Two parameters are used for making the correction: slope of the line of best fit and bias, representing where the line intersects the y-axis. The slope and bias used for correcting the UKMO wind stresses are presented in Figure (5). The contour of the slope, both in u- and v-direction, shows that the biggest slope and the steepest transition reside around dry points. This may be due to the difference in schemes used for interpolating the wind fields into the grid points of the DCSM.

The UKMO wind fields are available in a coarser grid and in our study te interpolation is done without taking the land-sea difference into consideration.

In this study, the two wind fields are assumed to deviate from the true ones by a time-correlated noise term as in equation (2). Using HiRLAM and corrected UKMO wind fields, the average wind field is computed using (7) and the approximate of white noise realization is computed using equation (8) and (12). Figure (6) shows the estimate of the spatial correlation associated with the point located at Hoek van Holland. The spatial correlation varies smoothly over the model domain. Moreover, the structure shows local characteristic, where the highest correlation is at the point of reference and it decreases for the locations further away. We also observe that the correlation profile of the u-direction tends to stretch in the zonal direction, while that of the v-direction spread more in meridional direction. This shows that the noise has different characteristics from the ones in the operational system where the isotropic assumption is used.

d. Data assimilation experiments

The DCSM and wind data described previously have been used in data assimilation experiments using the algorithm described in Section (c). The experimental setup in this study is as follows. At each iteration step, the model is run twice for the same simulation period. The first run is to generate the central mode, which is the run where all noise terms are set to zero. The second run is to produce a forecast sample, which is perturbed by the approximate white noise representing model error. Added also to the second run is a realization of a Gaussian white noise process with standard deviation 10 cm for representing observation uncertainty. In both runs, the average of the HiRLAM and UKMO wind stress is used to drive the DCSM. The simulation

is run for the period 01 November 2007 until 28 January 2005. To eliminate the spin-up effect, the first three days simulation results are excluded in computing the covariance.

Open-loop computation as well as up to the fifth closed-loop iteration step have been performed in this study. Figure (7) presents the contour plots of the water-level gain at each iteration step associated with assimilation station Hoek van Holland. The contour interval is 0.01. The pictures show that the estimate of the gain profile is smooth and has local characteristic. Near the assimilation station the gain has the biggest value and it decreases with distance. Moreover, the gain at locations far from the assimilation station is practically zero. This means that correction associated with certain assimilation station is propagated only to the nearby locations and that no correction will be imposed to the locations further away from the assimilation station. The pictures show also that before convergence, the gain computed at one iteration step becomes smoother and more localized than the ones computed at the previous steps. The open-loop gain has the biggest magnitude compared to the ones computed in the closed-loop steps. In the open-loop run, the difference between the two samples is the biggest. On the other hand, the assimilation of observation in the closed-loop run has brought both samples closer to the data. This in turn will reduce the covariance and hence the gain will become smaller. The gain profile does not change significantly starting from the third closed-loop step. It indicates that the filter converges, regardless of the fact that the model is nonlinear.

The evaluation of the results is also done by implementing the gain into a data assimilation system and checking if the assimilation system has brought the model state closer to the observation. In this study, the only objective data, which we have for evaluating the performance of the data assimilation system is the water-level data. Water level data from eight observation stations are assimilated in the first experiment. Five stations are located along the east coast of

the UK and three others are along the Dutch coast as shown in Figure (3).

The first check is done by comparing the water level analyses to the observation at assimilation stations, which are the observation stations where data is used in assimilation. Figure (8) shows the RMS of the water level difference between the analyses and the observation at assimilation stations. This figure demonstrates that the assimilation has succeeded in pulling the model closer to the observation in all assimilation stations. The biggest reduction in RMS innovation is obtained by data assimilation system using the gain computed in the open-loop step. This is due to the fact that the open-loop gain is also the biggest. This will put more weigh to the observation than to the model forecast, pulling the model closer to the observations. In the next iteration steps, the RMS error at the assimilation stations increases in the first iteration steps until it converges at the third closed-loop step.

The second check is by comparing the water level analyses to the observations at locations where data is not assimilated. We call such location here as validation station. The RMS residual at validation stations is presented in Figure (9). At all validation stations RMS water-level innovation of the analyses using open-loop gain is the biggest and it tends to decrease in the subsequent iteration steps before it converges. In the first five validation stations the RMS water-level innovation is smaller than that of the deterministic forecast, which is obtained by running the deterministic model without data assimilation. This indicates that the assimilation works also in these locations, although the data is not assimilated. This also may indicate that the covariance, hence the gain, has the proper structure that it distributes the correction in a consistent way to other locations. However, this is not true for the other three validation stations, which are located in the northern part of the Netherlands like Harlingen, Huibertgat, and Delfzijl. In these locations, the RMS water-level innovation obtained with data assimilation system

using open-loop gain is even bigger than that of the deterministic forecast without assimilation. It decreases over iteration-step until it converges to the value similar to that of the deterministic forecast.

To solve this problem we performed another experiment by adding one assimilation station in the northern part of the Netherlands. Station Huibertgat is chosen in this experiment to be the additional assimilation station. The gain is recomputed for this new set of assimilation stations. The RMS water-level residual obtained by the assimilation system with this new set of assimilation stations are shown in Figure (10) and (11) for the assimilation and validation stations respectively. These figures suggest that now the assimilation works well in all stations.

5. Conclusion

We have presented in this paper an application of the two-sample Kalman filter. The implementation is done with an operational storm surge prediction model, the Dutch Continental Shelf Model (DCSM). Two wind fields are available for driving the model, namely HiRLAM and UKMO. The two forecast samples required for computing the Kalman gain are generated by running the model twice, each run driven by either one of the two wind-fields. The basic assumption in this work is that the difference between the two wind fields is characteristics of the unknown true error. This work presents an attempt to use a surrogate quantity for the unknown true error, for estimating forecast error covariance used to define a steady-state Kalman gain.

It is generally believed that the error in the wind fields is correlated in time. Since the two-sample Kalman filter is developed based on the assumption that the system is driven by white noise, some modification is required before we can implement the algorithm. The first

modification is a transformation on the two wind fields. The objective of the transformation is to construct a new model representation, in which the driving error process becomes white noise. However, the transformation causes the two independent wind fields to be two dependent wind fields. Hence, the two-sample Kalman filter should be slightly modified. The underlying idea of the modification is to use a sample of the system state and an approximate central mode for computing forecast error covariance. The sample transformation and the modified algorithm are verified by implementing them on a simple one-dimensional wave model. The verification is done by comparing the results to a reference, obtained using the original algorithm. The experiments show that the results are consistent with the reference.

Finally the algorithm is implemented with the DCSM. Some care must be taken beforehand on a systematic difference between HiRLAM and UKMO wind fields. We perform some correction on the UKMO wind fields to eliminate this systematic difference. Using HiRLAM and corrected UKMO wind fields with the modified algorithm, we compute the estimate of Kalman gain. It turns out that the algorithm converged after three iteration. An evaluation is performed by implementing the gain estimates in a data assimilation system. We checked the performance of the assimilation system on both assimilation stations, where data is used for assimilation, and validation stations, where data is not used for assimilation. The experiments show that the assimilation system with the estimated gain is successful in bringing the model closer to data in all observation stations.

Acknowledgments.

The authors would like to acknowledge Bruce Hackett from Norwegian Meteorological Institute for initiating the idea studied in this paper.

References

Alves, O. and C. Robert, 2005: Tropical pacific ocean model error covariances from monte carlo simulations. *Q.J.R.Meteorol.Soc.*, **131**, 3643–3658.

Borovikov, A., M. Rienecker, C. Keppenne, and G.C.Johnson, 2005: Multivariate error covariance estimates by monte carlo simulation for assimilation studies in the pacific ocean. *Monthly Weather Review*, **133**, 2310–2334.

Fisher, M., 2003: Background error covariance modelling. ECMWF Seminar.

Leenderstse, J. J., 1967: *Aspects of a computational model for long-period free-surface flow modelling*. Report RM-5294-RR, Rand Corporation, Santa Monica, California, USA.

Leeuwenburgh, O., 2007: Validation of an enkf system for ogcm initialization assimilating temperature, salinity, and surface height measurements. *Monthly Weather Review*, **135**, 125–139.

Stelling, G. S., 1984: On the construction of computational methods for shallow water flow problems. Ph.D. thesis, Delft University of Technology, The Netherlands.

Verboom, G. K., J. G. de Ronde, and R. P. van Dijk, 1992: A fine grid tidal flow and storm surge model of the north sea. *Continental Shelf Research*, **12 (2/3)**, 213–233.

Verlaan, M., 1998: Efficient kalman filtering algorithms for hydrodynamics models. Ph.D. thesis, Delft University of Technology, The Netherlands.

Verlaan, M. and A. W. Heemink, 1997: Tidal flow forecasting using reduced rank square root filters. *Stochastic Hydrology and Hydraulics*, **11**, 349–368.

List of Figures

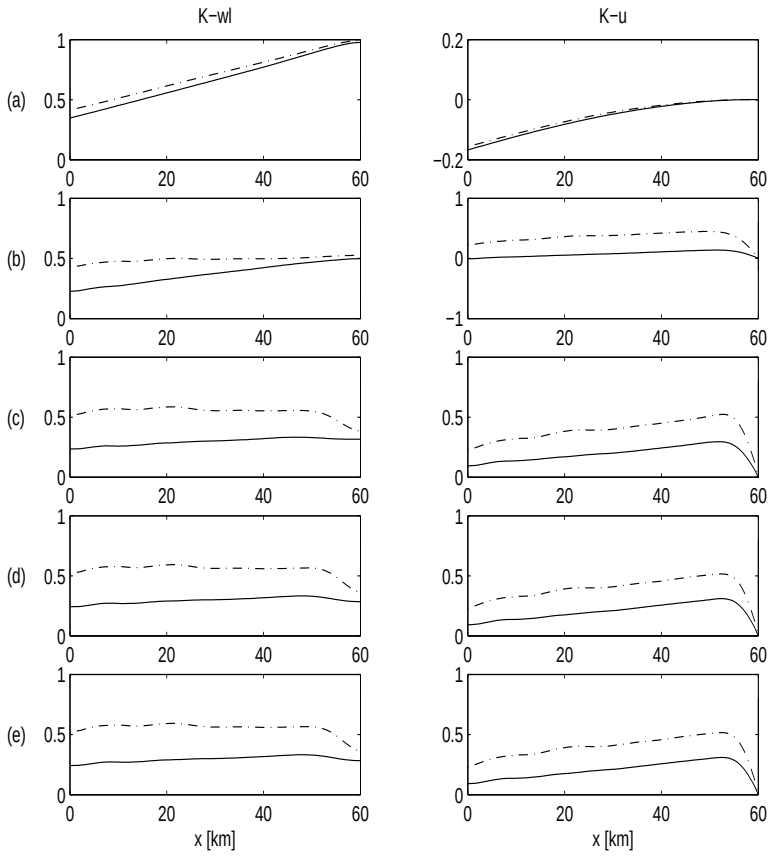


FIG. 1. Gain estimates with dependent (full-lines) and independent (dashed-lines) samples, (a) open-loop, (b)-(e) 1st-4th closed-loop estimates. The left-hand panel shows the gain associated with water-level ξ and the right-hand panel associated with velocity u. The location of the assimilation station is at the right-end of the estuary, $x = 60$ km.

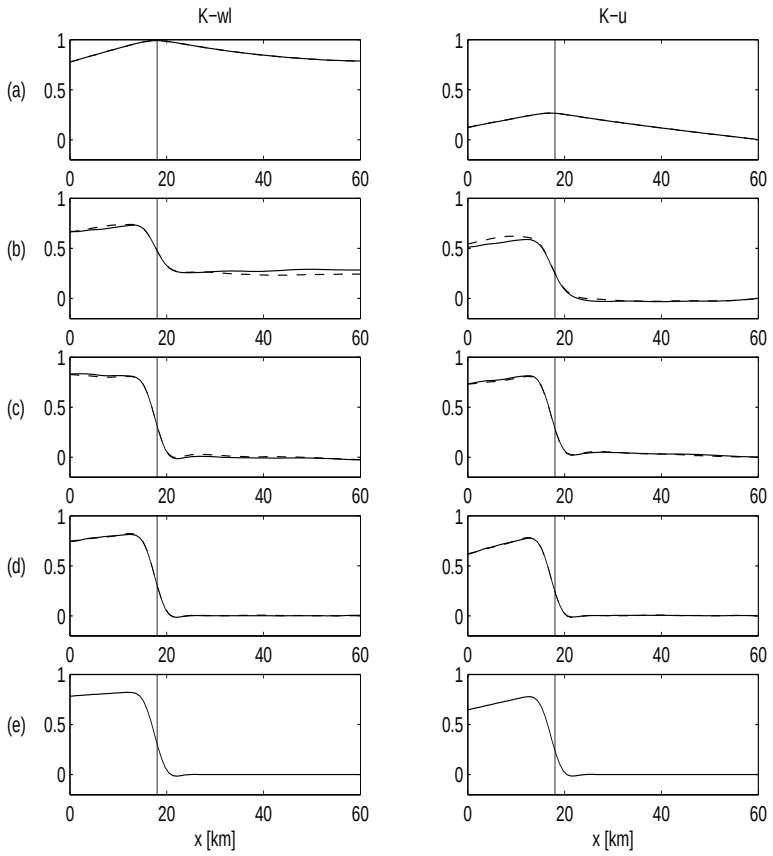


FIG. 2. Estimated gain of water-level ξ (left-hand panels) and velocity u (right-hand panels): (a)open-loop, (b)-(d)closed-loop estimates, 1st-3rd iteration steps, (e) RRSQRT with 50 modes. The full-line represents the gain computed by using the transformed samples, the dashed-line using the original samples, and the vertical line shows the location of the assimilation station.

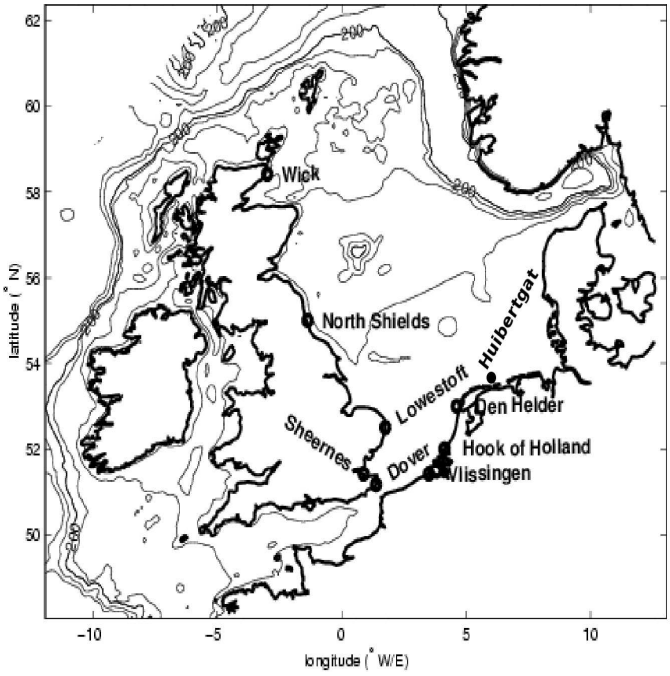


FIG. 3. DCSM area with some water-level observation stations

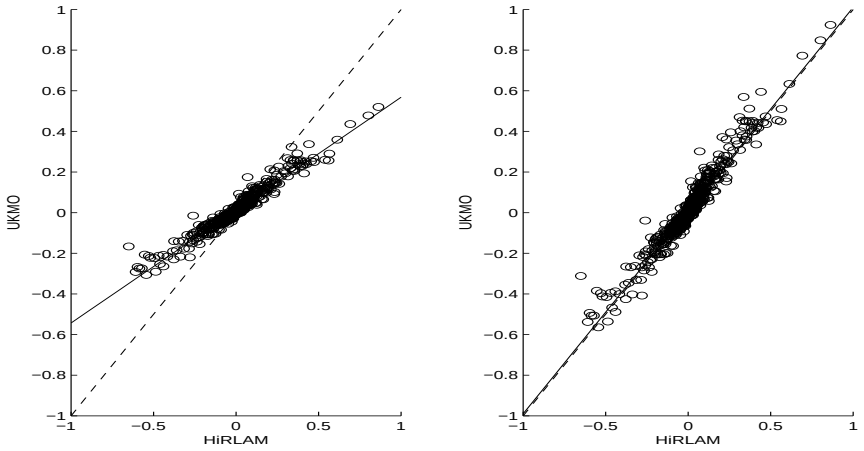


FIG. 4. Scattered-plot of wind stress HiRLAM vs UKMO at Hoek van Holland, (left) original and (right) after correction. Full line shows best fit of the dots, while dashed line gives reference to perfect positive correlation

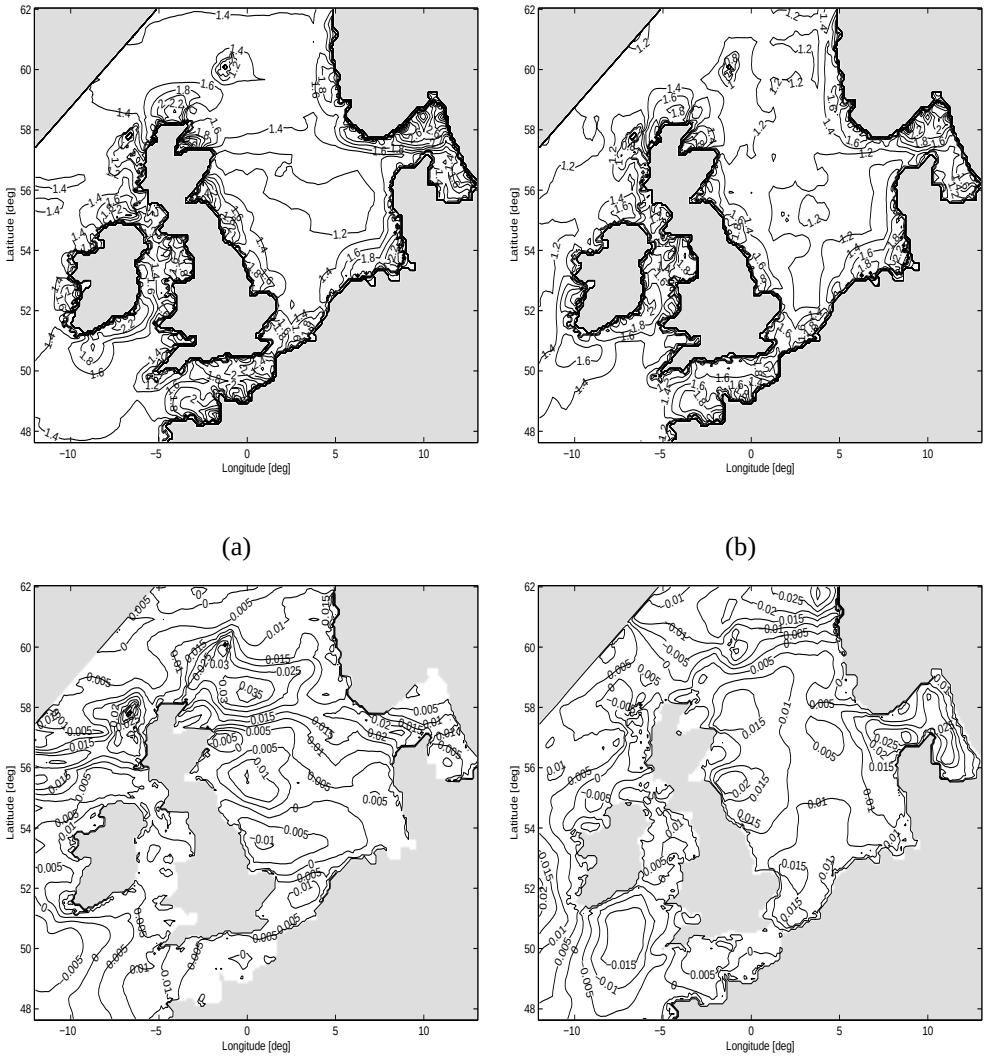
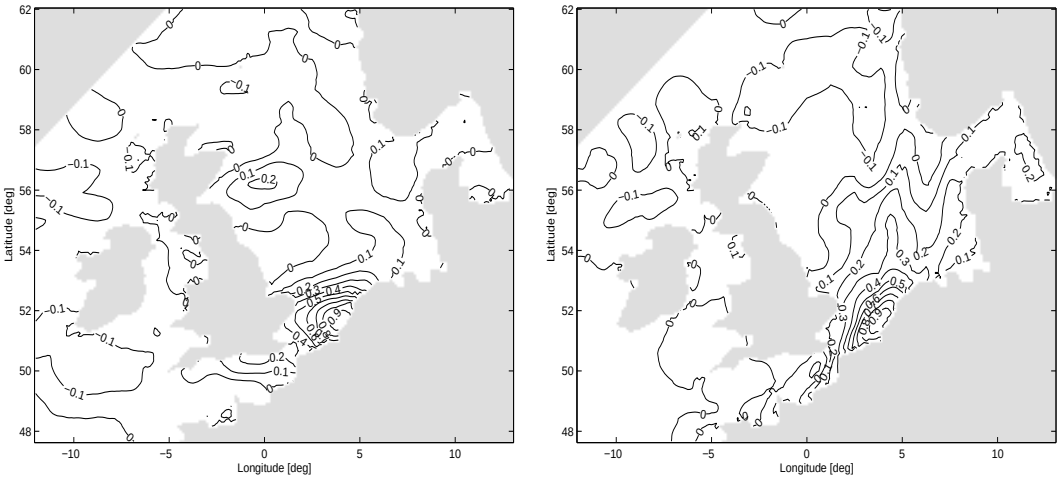


FIG. 5. Contours of (a)-(b) slope and (c)-(d) bias for u- and v-direction respectively used for correcting UKMO wind stresses. Contour interval is 0.2 for slope and 0.005 for bias.



(a) (b)

FIG. 6. Spatial correlation of the wind noiseassociated with Hoek van Holland, (a) u-direction and (b) v-direction

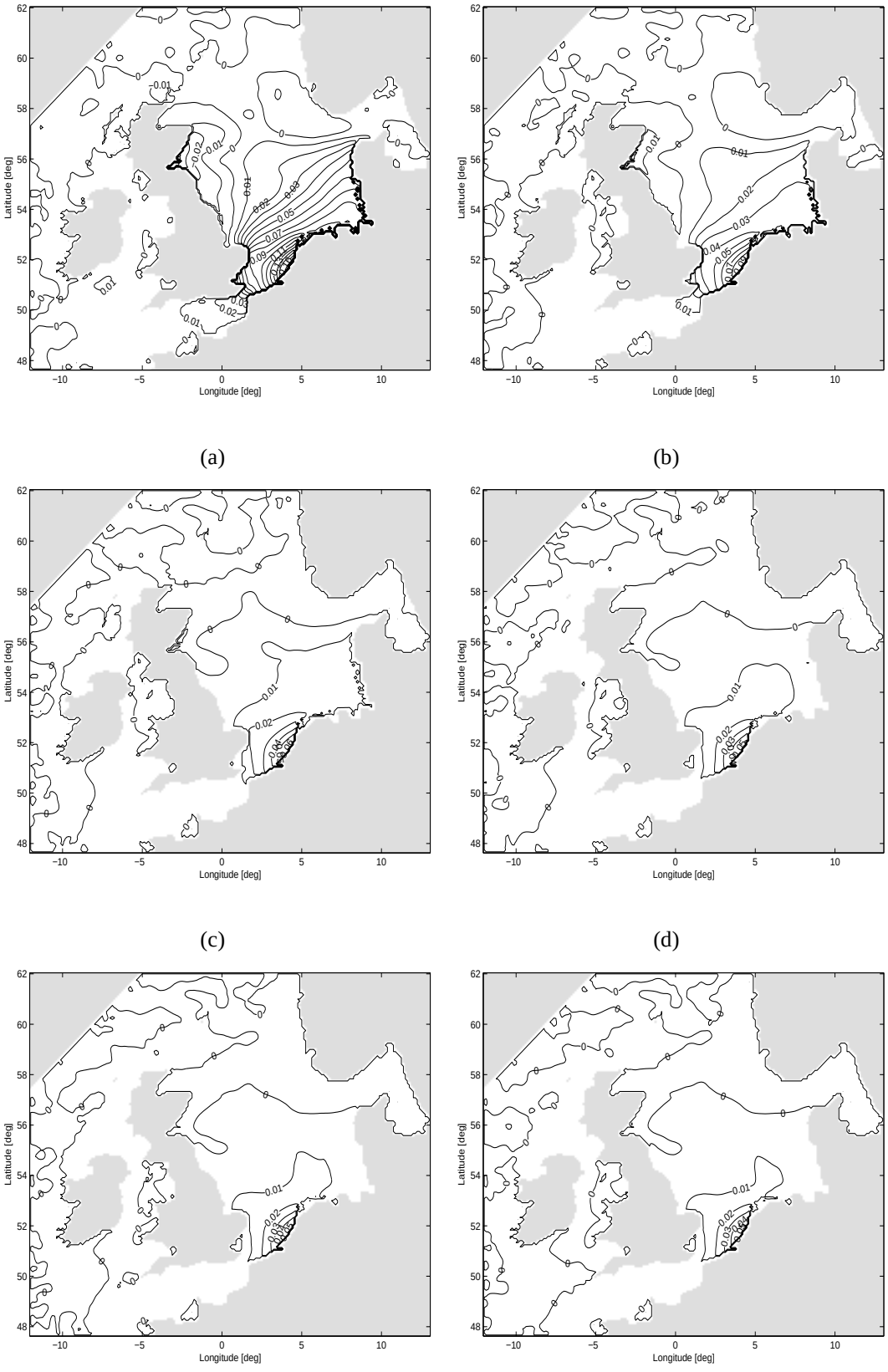


FIG. 7. Water-level gain for assimilation station Hoek van Holland: (a) Open-loop estimate, (b)-(f) Closed-loop estimates, 1st-5th iteration-steps.

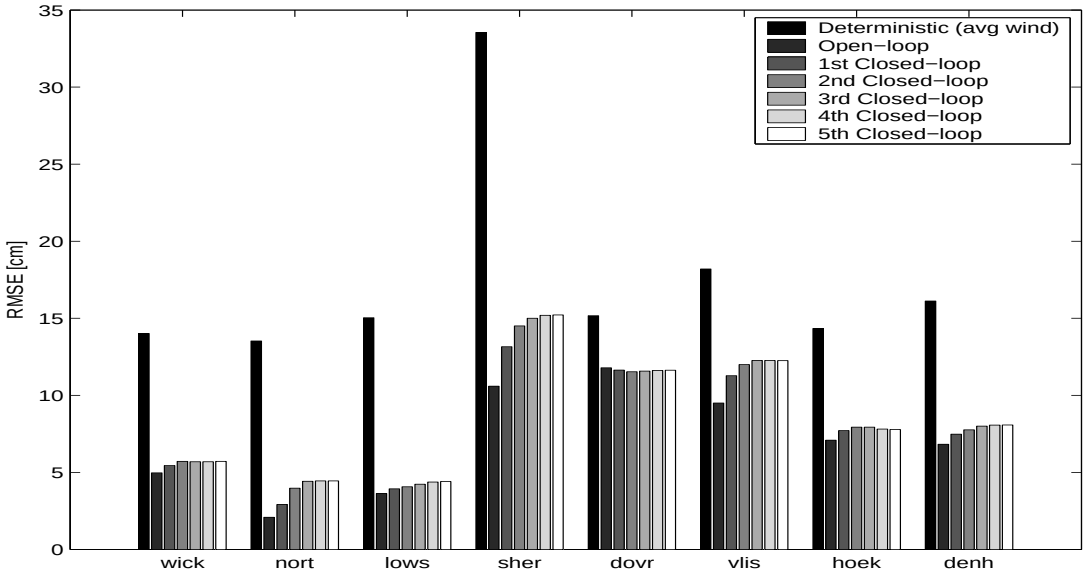


FIG. 8. RMS error at assimilation stations with eight assimilation stations

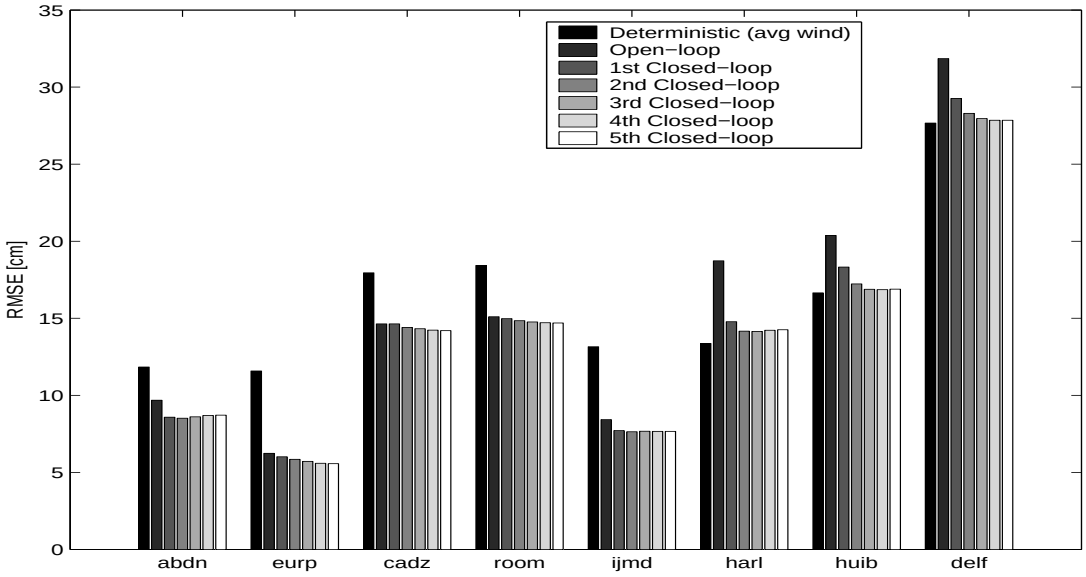


FIG. 9. RMS error at validation stations with eight assimilation stations

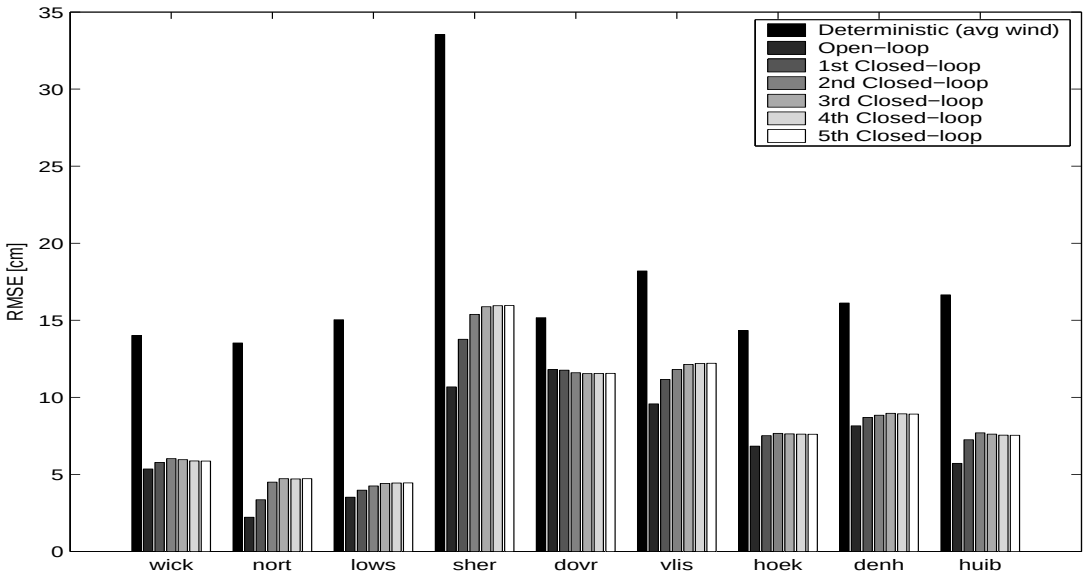


FIG. 10. RMS error at assimilation stations with nine assimilation stations

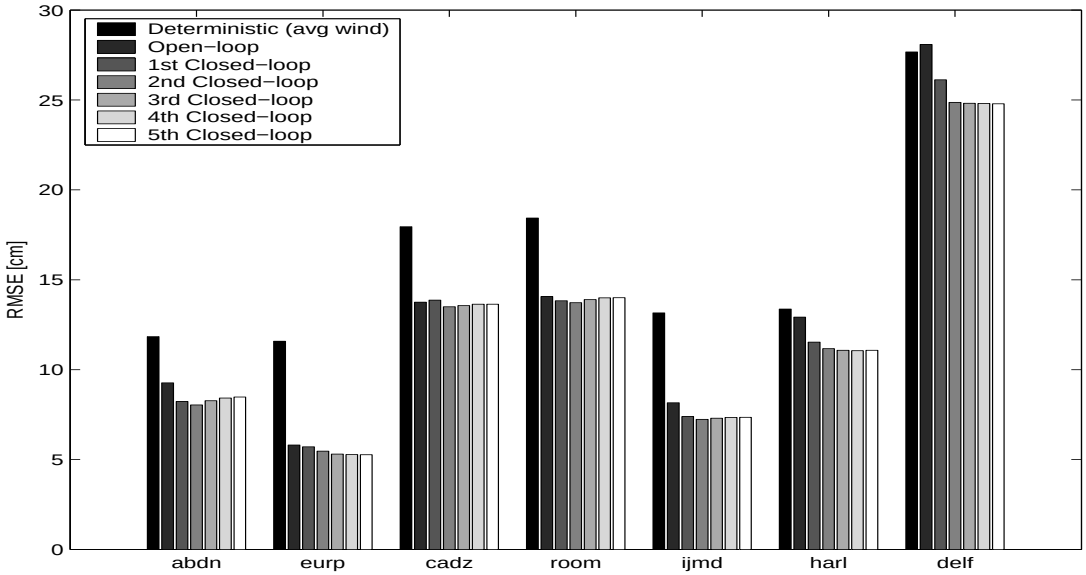


FIG. 11. RMS error at validation stations with nine assimilation stations