ELSEVIER

Journal of Marine Systems 36 (2002) 101–127

JOURNAL OF MARINE SYSTEMS

www.elsevier.com/locate/jmarsys

A simplified reduced order Kalman filtering and application to altimetric data assimilation in Tropical Pacific

Ibrahim Hoteit *, Dinh-Tuan Pham*, Jacques Blum*

Laboratoire de Modélisation et Calcul, Tour IRMA, BP 53, F-38041 Grenoble, France

Received 4 September 2000; accepted 7 March 2002

Abstract

Several studies have demonstrated the effectiveness of the singular evolutive extended Kalman (SEEK) filter and its interpolated variant called singular evolutive interpolated Kalman (SEIK) filter in their capacity to assimilate altimetric data into ocean models. However, these filters remain expensive for real operational assimilation. The purpose of this paper is to develop degraded forms of the SEIK filter which are less costly and yet perform reasonably well. Our approach essentially consists in simplifying the evolution of the correction basis of the SEIK filter, which is the most expensive part of this filter. To deal with model instabilities, we also introduce two adaptive tuning schemes to control the correction basis evolution and adjust the variable forgetting factor. Our filters have been implemented in a realistic setting of the OPA model over the tropical pacific zone and their performance studied through twin experiments in which the observations are taken to be synthetic altimeter data sampled on the sea surface. The SEIK filter is used as a reference for comparison. Our new filters perform nearly as well as the SEIK, but can be 2–30 times faster. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Data assimilation; OPA model; Kalman filter; SEEK filter; SEIK filter; Forgetting factor

1. Introduction

In recent years, there has been an increasing interest in operational data assimilation schemes in oceanography. The need for such schemes arises from the purposes of improving short and mid-term weather prediction, developing climate prediction or from the specific objectives of the navies in various countries. Since the event of satellites launching, considerable progress has been made in applying concepts and techniques for statistical estimation and optimal control theories to the problem of oceanographical data assimilation (see for example Ghil and Malanotte-Rizzoli, 1991 for a review). The major challenge for the coming year is to design operational data assimilation schemes which can be operated in conjunction with realistic ocean models, with reasonable quality, and at acceptable cost as in meteorology.

One of the most widely used statistical assimilation schemes is the extended Kalman (EK) filter which is an extension of the common Kalman filter to nonlinear models. However, brute-force implementation of the EK filter in realistic ocean models is not possible in practice because of its prohibitive cost. To

* Corresponding authors. Tel.: +33-4-76514600; fax: +33-4-76631263.

E-mail addresses: Ibrahim.Hoteit@imag.fr (I. Hoteit), Dinh-Tuan.Pham@imag.fr (D.-T. Pham), Jacques.Blum@imag.fr (J. Blum).

0924-7963/02/$ - see front matter
PII: S0924-7963(02)00129-X

deal with this difficulty, different degraded forms of the EK filter, which basically reduce the dimension of the system n through some kind of projection onto a low dimensional subspace, have been proposed (Cane et al., 1995; Dee, 1990; Evensen, 1994; Fukumori and Malanotte-Rizzoli, 1995; Hoang et al., 1997).

With the same aim in view, the singular evolutive extended Kalman (SEEK) filter has been proposed by Pham et al. (1997). It essentially consists in approximating the error covariance matrix by a singular low rank r ($r \ll n$) matrix which leads to making correction only in the directions for which the error is not sufficiently attenuated by the system. Such directions evolve in time according to the system dynamics. Further, Pham et al. (1998) introduces a variant called singular evolutive interpolated Kalman (SEIK) filter in which the linearization used in the SEEK filter is replaced by a linear interpolation as this could result in less error for large deviations. These two filters have been applied in different realistic ocean frameworks with quite satisfactory results (Brasseur et al., 1999; Pham et al., 1997, 1998; Verron et al., 1998). The SEIK filter has strong similarities to the ensemble Kalman (ENK) filter introduced by Evensen (1994) (see also Burgers et al., 1998). Such a filter have been tested in realistic settings (Evensen, 1994; Houtekamer and Mitchell, 1998) and produces good results. Our filter however amounts to use, in an efficient way, the smallest number of ensemble member (or interpolating states), namely $r+1$, hence possesses an advantage in term of cost.

But the above filters remain expensive in real operational assimilation since the evolution equation of their correction basis requires model integration for each basis vector. Our aim is to reduce further their costs by simplifying the evolution of their correction basis, as it is the only way to achieve a significant reduction. This results in several degraded forms of the SEEK and SEIK filters which are much less costly and yet perform reasonably well. These filters generalize the fixed correction basis filter of Brasseur et al. (1999), derived from an empirical orthogonal function (EOF) analysis.

A number of studies have revealed rapid error growth when model instabilities appear (Farrell, 1989; Moore and Farrell, 1994; Trefethen et al., 1993). Our experiments confirm this finding in that the performances of our filters decline mostly in the presence of instability. To overcome this problem, we introduce two adaptive tuning schemes. The first is based on the control of the evolution of the correction basis and the second makes use of a variable forgetting factor.

This paper is organized as follows. The SEEK and SEIK filters are described in Section 2. Section 3 introduces some new degraded forms of the SEEK and SEIK filters. Section 4 discusses two approaches to overcome model instabilities. Finally, the performance of the new filters is illustrated in Section 5 with some simulation results based on a realistic setting of the OPA model over the tropical Pacific ocean.

2. The singular evolutive Kalman filters

We shall adopt the notation proposed by Ide et al. (1997). Consider a physical system described by

$$X^{\mathrm{t}}(t_k) = M(t_k, t_{k-1})X^{\mathrm{t}}(t_{k-1}) + \eta(t_k) \tag{1}$$

where $X^{\mathrm{t}}(t)$ denotes the vector representing the true state at time t, $M(t,s)$ is an operator describing the system transition from time s to time t and $\eta(t)$ is the system noise vector. At each time t_k, one observes

$$Y_k^{\mathrm{o}} = H_k X^{\mathrm{t}}(t_k) + \varepsilon_k \tag{2}$$

where H_k is the observational operator and ε_k is the observational noise. The noises $\eta(t_k)$ and ε_k are assumed to be independent random vectors with mean zero and covariance matrices Q_k and $\sigma^2 R_k$, respectively. Often R_k is assumed to be the identity matrix and then σ^2 represents the error covariance. The parameter σ^2 is introduced because the observation error matrix may be known only up to a constant factor, and even if it is known, we still prefer to estimate σ^2 from the data, for reasons explained later.

The sequential data assimilation consists in the estimation of the state of the system at each observation time, using only observations up to this time. In the linear case, this problem has been solved by the well known Kalman filter. In the nonlinear case, one often linearizes the model around the current estimated state vector, which yields to the so-called extended Kalman (EK) filter (see for example Ghil and Malanotte-Rizzoli, 1991 for details).

The SEEK filter is aimed to reduce the prohibitive cost of the EK filter (in meteorology and oceanography applications), arising from to the huge number (n) of the state variables. The main idea is to view the error covariance matrix as singular with a low rank $r \ll n$. This leads to a filter in which the errors correction is made only along certain directions parallel to a linear subspace of dimension r. These directions are those for which error is not sufficiently attenuated by the system dynamic.

This filter proceeds in two stages apart from an initialization stage. To initialize the filter, we generate a long sequence of state vectors according to our model and take $X^{a}(t_0)$ as the average of these vectors and $P^{a}(t_0)$ as the rank r approximation to their sample covariance matrix, obtained via an EOF analysis (see Pham et al., 1997 for details).

2.1. Forecast stage

At time t_{k-1} an estimate $X^{a}(t_{k-1})$ of the system state and its corresponding error covariance matrix $P^{a}(t_{k-1})$, in the factorized form $\sigma^2 L_{k-1} U_{k-1} L_{k-1}^{T}$ where L_{k-1} and U_{k-1} are of dimension $n \times r$ and $r \times r$, respectively, are available. The model (model (1)) is used to forecast the state as

$$X^{f}(t_k) = M(t_k, t_{k-1}) X^{a}(t_{k-1}). \tag{3}$$

The forecast error covariance matrix is given by

$$P^{f}(t_k) = \sigma^2 L_k U_{k-1} L_k^{T} + Q_k \tag{4}$$

where

$$L_k = \mathbf{M}(t_k, t_{k-1}) L_{k-1} \tag{5}$$

and $\mathbf{M}(t_k,t_{k-1})$ is the gradient of $M(t_k,t_{k-1})$ evaluated at $X^{a}(t_{k-1})$.

2.2. Correction stage

The new observation Y_k^{o} at time t_k is used to correct the forecast according to

$$X^{a}(t_k) = X^{f}(t_k) + G_k[Y_k^{o} - H_k X^{f}(t_k)], \tag{6}$$

where G_k is the gain matrix given by

$$G_k = L_k U_k L_k^{T} \mathbf{H}_k^{T} R_k^{-1}, \tag{7}$$

with $\mathbf{H}_k$ as the gradient of H_k evaluated at $X^{f}(t_k)$ and U_k is computed from

$$U_k^{-1} = \left[U_{k-1} + (L_k^{T} L_k)^{-1} L_k^{T} Q_k L_k (L_k^{T} L_k)^{-1} \right]^{-1} + L_k^{T} \mathbf{H}_k^{T} R_k^{-1} \mathbf{H}_k L_k. \tag{8}$$

The corresponding filter error covariance matrix is then equal to

$$P^{a}(t_k) = \sigma^2 L_k U_k L_k^{T}. \tag{9}$$

Eqs. (4) and (9) are only included for interpreting the results, but not needed in the algorithm. Thus the SEEK filter reduces drastically the computing cost with respect to the EK filter. Basically, it requires $r+1$ times the cost of the numerical integration of the model (to compute the evolution of L_k). One can also see from Eq. (6) that the correction of the forecast state is only done in the directions parallel to the linear space spanned by the columns of L_k. Therefore the collection of such vectors will be called the correction basis of the filter.

It has been shown in Pham et al. (1997) that in the case of a linear autonomous system and under the usual observability condition (which is clearly unavoidable for any data assimilation methods), the SEEK filter is stable provided that the initial covariance error matrix P_0 satisfies some "generic" condition and has rank at least equal to r^*, where r^* is the number of eigenvalues of the system matrix $\mathbf{M}$ (which is constant by assumption) of modulus not less than 1. Here by generic we mean that the set of values of P_0 for which the condition is not met has zero measure. The precise conditions for which stability holds can be found in Pham et al. (1997). It is shown there that under these conditions, the linear space spanned by the correction basis vectors[1] converges to that spanned by r^* eigenvectors of largest eigenvalues (in modulus)[2] of $\mathbf{M}$. It is then a simple matter to show that the filter is stable since any error in the

[1] Although these basis vectors (the columns of L_k) are not unique, the linear space spanned by them is.

[2] The proof in this paper actually is only valid in the case where the r^*-th largest eigenvalue (in modulus) is single, but it can be extended to cover the case of multiple eigenvalue.

direction parallel to the linear space spanned by the other eigenvectors would be attenuated by the system dynamic itself. Of course, in applications, the system is not linear, but one can hope that if the nonlinearity is weak enough, the evolutive correction basis L_k still eventually capture most of the errors which are not sufficiently attenuated by the system dynamics. Note that although theoretically the initial basis can be almost anything, in practice its choice is of great importance, since (i) the ideal conditions (no system noise and linear dynamics) under which our arguments are based are not met and (ii) even if they are met the convergence of the filter could be too slow to be of practical interest. We have however reasons (not detailed here and supported by experiments) to believe that the EOFs analysis provides a good choice.

The SEEK filter, as the EK filter, may produce instabilities, even divergence, due to strong model nonlinearities (see for example Evensen, 1992; Gauthier et al., 1993). To alleviate this problem, Pham et al. (1998) proposed a variant of the SEEK, called SEIK, in which linear interpolation is used instead of linearization. The interest of this filter is twofold: firstly it avoids the complex computation of the gradient required by the SEEK filter. Secondly it is more robust with respect to model nonlinearities, since interpolation results in less error for large deviations than linearization. The novelty in this filter lies in the choice of its interpolating states $X_1^{\mathrm{a}}(t_k),\ldots,X_I^{\mathrm{a}}(t_k)$ which are drawn randomly at every filtering step in such a manner as to represent the analysis state vector and its error covariance matrix by an ensemble of state vectors, i.e.

$$X^{\mathrm{a}}(t_k) = \frac{1}{I}\sum_{i=1}^{I} X_i^{\mathrm{a}}(t_k), \tag{10}$$

$$P^{\mathrm{a}}(t_k) = \frac{1}{I}\sum_{i=1}^{I} \left[X_i^{\mathrm{a}}(t_k) - X^{\mathrm{a}}(t_k)\right]\left[X_i^{\mathrm{a}}(t_k) - X^{\mathrm{a}}(t_k)\right]^T. \tag{11}$$

The use of these ensemble states is very similar to the ENK filter of Evensen (1994). However, here the drawing is constrained to satisfy Eqs. (10) and (11) and further the smallest possible number of interpolating states, equal to $r+1$, is used.

The SEIK filter proceeds in three stages apart from the initialization stage which is identical to that in the SEEK filter.

2.2.1. Drawing interpolating states: or "second order exact sampling"

At time t_{k-1}, an analysis state $X^{\mathrm{a}}(t_{k-1})$ and its corresponding error covariance matrix $P^{\mathrm{a}}(t_{k-1})$, in the factorized form $\sigma^2 L_{k-1}U_{k-1}L_{k-1}^{\mathrm{T}}$, are available. The Cholesky decomposition of U_{k-1}^{-1} to $C_{k-1}C_{k-1}^{\mathrm{T}}$ is performed which enables to write

$$P^{\mathrm{a}}(t_{k-1}) = \sigma^2 L_{k-1}(C_{k-1}^{-1})^{\mathrm{T}}\Omega_{k-1}^{\mathrm{T}}\Omega_{k-1}C_{k-1}^{-1}L_{k-1}^{\mathrm{T}} \tag{12}$$

where Ω_k is any $(r+1)\times r$ matrix with orthonormal columns and zero column sums. This matrix will be drawn randomly using the procedure described in Appendix A. Then one can take as interpolating states

$$X_i^{\mathrm{a}}(t_{k-1}) = X^{\mathrm{a}}(t_{k-1}) + \sigma\sqrt{r+1}L_{k-1}(\Omega_{k-1,i}C_{k-1}^{-1})^{\mathrm{T}}, \tag{13}$$

for $1 \le i \le r+1$, where $\Omega_{k-1,i}$ denotes the ith row of Ω_{k-1}. Note that formulas (12) and (13) ensures that the sample covariance matrix of the state is precisely $P^{\mathrm{a}}(t_{k-1})$.

2.2.2. Forecasting

One applies the model to bring the interpolating states $X_i^{\mathrm{a}}(t_{k-1})$ to $X_i^{\mathrm{f}}(t_k)$. The state forecast $X^{\mathrm{f}}(t_k)$ will be taken as the average of the $X_i^{\mathrm{f}}(t_k)$. The prediction error covariance matrix is approximated by

$$P^{\mathrm{f}}(t_k) = \frac{1}{r+1}\sum_{j=1}^{r+1}\left[X_i^{\mathrm{f}}(t_k) - X^{\mathrm{f}}(t_k)\right] \times \left[X_i^{\mathrm{f}}(t_k) - X^{\mathrm{f}}(t_k)\right]^{\mathrm{T}} + Q_k. \tag{14}$$

For later use, this matrix will be represented as

$$P^{\mathrm{f}}(t_k) = L_k[(r+1)T^{\mathrm{T}}T]^{-1}L_k^{\mathrm{T}} + Q_k, \tag{15}$$

with a correction basis

$$L_k = [X_1^{\mathrm{f}}(t_k) - X^{\mathrm{f}}(t_k)\cdots X_{r+1}^{\mathrm{f}}(t_k) - X^{\mathrm{f}}(t_k)]T = [X_1^{\mathrm{f}}(t_k)\cdots X_{r+1}^{\mathrm{f}}(t_k)]T \tag{16}$$

where T is a $(r+1)\times r$ full rank matrix with zero column sums. This property of T implies the second

equality in Eq. (16) and ensures that Eq. (15) is indeed the same as Eq. (14). The last assertion can be shown by noticing that any row vector v with components summing to 0, such as a row of the matrix in the right hand side of Eq. (16), can be expressed in the basis formed by the rows of T^{T} as $v = vT(T^{\mathrm{T}}T)^{-1}T^{\mathrm{T}}$. A convenient choice of T is the matrix

$$T = \begin{pmatrix} 1 & & 0 \\ \vdots & \ddots & \vdots \\ 0 & & 1 \\ 0 & \cdots & 0 \end{pmatrix} - \frac{1}{r+1} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix}. \quad (17)$$

2.2.3. Correction

As in the SEEK filter, the correction of the forecast state is done according to the formulas (6)–(8) except that the matrices $H_k L_k$ and U_{k-1} are replaced by $(HL)_k = [H_k X_1^{\mathrm{f}}(t_k) \cdots H_k X_{r+1}^{\mathrm{f}}(t_k)]T$ and $[\sigma^2(r+1)T^{\mathrm{T}}T]^{-1}$, respectively.

2.2.3.1. Observation error estimation. Theoretically, the variance σ^2 in Eq. (12) and especially in Eq. (13) should be the same as the factor σ^2 in the observation error $\sigma^2 R_k$. However, using this parameter when it is known is not recommended since this would generally leads to under estimating the filter error (especially when the observation error is small). A detailed explanation can be found in Pham (2001). It is better to estimate σ^2 from the data as follows. In the case where $R_k = I_d$, the identity matrix, we estimate σ^2 at time t_k by the quantity e_k^2/n_k^{o} where e_k^2 and n_k^{o} are updated recursively according to

$$e_k^2 = \rho e_{k-1}^2 + \| Y_k^{\mathrm{o}} - H_k X^{\mathrm{a}}(t_{k-1}) \|^2, \quad (18)$$

$$n_k^{\mathrm{o}} = \rho n_{k-1}^{\mathrm{o}} + \{\text{dimension of } Y_k^{\mathrm{o}} - r\} \quad (19)$$

where ρ is a forgetting factor (see Section 4.2). In our case, r is much smaller than the dimension of Y_k^{o} so we can drop it. (Also our procedure is only intend to be used in the case where this dimension is much larger than r.) This estimator actually estimates the spatial average of the squared prediction errors and its use in Eq. (12) would somewhat over-estimate the filter error covariance matrix. But we feel that it is better to over estimate rather than under estimate this matrix.

As the SEIK filter operates along the same principle as the SEEK filter, it possesses the same property for weekly nonlinear systems (in fact they coincide if the system is linear). It seems however that the SEIK filter is more robust with regard to strong nonlinearities than the SEEK filter.

3. Reduction of the cost of the SEIK filter

Our aim is to reduce the cost of the SEEK and SEIK filters. We will, however, consider only the SEIK filter since it is similar to the SEEK filter, and seems to perform better (in our experiments). Thus, we will propose several degraded forms of the SEIK filter, which are less costly and yet perform reasonably well. Our approach consists essentially in simplifying the way the correction basis of the SEIK filter evolves which is the most expensive part of the filter.

3.1. The singular fixed extended Kalman (SFEK) filter

Motivated by the fact that most of the error estimation in numerical experiments in Pham et al. (1997) was reduced immediately after the first correction, i.e. while the evolution of the EOF basis was not yet effective, Brasseur et al. (1999) proposed to keep the initial correction basis of the SEEK filter fixed in time. This can be justified by the fact that the state of the ocean evolves very slowly. Therefore, one can suppose that the transition operator of the ocean model is almost equal to identity and then write

$$\mathbf{M}(t_{k+1}, t_k) = I_d. \quad (20)$$

Under this approximation, the correction basis of the SEEK filter, according to Eq. (5), will always remain constant equal to the initial EOF basis L_0. This filter, called singular fixed extended Kalman (SFEK), operates in two stages exactly as for the SEEK but

without the evolution equation of the correction basis (Eq. (5)). It is thus almost $r+1$ times faster than the SEEK filter.

The SFEK filter highly depends on the representativeness of the EOF correction basis. It yields good results when this basis represents sufficiently well the model variability. Its appeal is its very low cost, thus providing a cheap way to test the relevance of the EOF basis. Actually, we find it useful to experiment with the SFEK filter to gain insight on the correction basis dimension r and the forgetting factor (see Section 4.2) to be used in the SEIK filter and its variants.

3.2. The singular intermittently evolutive interpolated Kalman (SIEIK) filter

Another approach to simplify the evolution of the correction basis of the SEIK filter is to use the convergence property of the error covariance matrix of the filtered state. It is well known that when the model model is linear and autonomous, this matrix, in the Kalman filter converges in time to a fixed matrix. Thus in the nonlinear but nearly linear (and still autonomous) case, one can expect that it tends quickly towards a semi-fixed mode in which it will evolve slowly. We therefore propose, after an initialization period with the SEIK filter, to let the correction basis evolve intermittently according to two modes:

- **Fixed mode:** Keep the matrices L_k, U_k and $(HL)_k$ fixed. The drawing of the interpolating states is no longer performed. Forecast and correction steps are done as for the SFEK filter.

- **Catch up mode:** After a certain time in the fixed mode, the matrices L_k, U_k and $(HL)_k$ will need to be updated. We start up again the SEIK filter to bring back these matrices to the semi-fixed mode. More precisely, we let L_k evolve as in the SEIK filter starting with its previous (fix) value and update U_k accordingly. This catch up mode actually can be very short (in our experiments, 1 day, that is 1 time step, can be enough).

This filter, called SIEIK, can be as stable as the SEIK filter but much less costly. For example, if one evolves the correction basis once every K filtering steps, one can easily see that the cost of the SIEIK filter is about K times less than the SEIK filter.

3.3. The singular semi-evolutive interpolated Kalman (SSEIK) filter

The main idea of this filter is to let only a few correction basis vectors evolve and keep the other fixed. The basis vectors, which do not evolve, are taken to be those which contribute the least to the error filter representation, as this would minimize the effect of keeping them fixed.

To construct this filter we start by representing the analysis error covariance $P^{\mathrm{a}}(t_{k-1}) = \sigma^2 L_{k-1} U_{k-1} L_{k-1}^{\mathrm{T}}$ as

$$P^{\mathrm{a}}(t_{k-1}) = \sigma^2 L_{k-1} (C_{k-1}^{-1})^{\mathrm{T}} \Theta \Theta^{\mathrm{T}} C_{k-1}^{-1} L_{k-1}^{\mathrm{T}} \tag{21}$$

where C_{k-1} is the Cholesky decomposition of U_{k-1}^{-1} and Θ is an arbitrary orthogonal matrix. This shows that there is a whole set of equivalent representations

$$P^{\mathrm{a}}(t_{k-1}) = \sigma^2 \tilde{L}_{k-1} \tilde{L}_{k-1}^{\mathrm{T}} \tag{22}$$

with the correction basis

$$\tilde{L}_{k-1} = L_{k-1} (C_{k-1}^{-1})^{\mathrm{T}} \Theta. \tag{23}$$

The idea is to exploit this degree of freedom to our advantage. One can split the representation (21) as

$$P^{\mathrm{a}}(t_{k-1}) = \sigma^2 \tilde{L}_{k-1}^{r_0} \tilde{L}_{k-1}^{r_0 \mathrm{T}} + \sigma^2 \tilde{L}_{k-1}^{r_1} \tilde{L}_{k-1}^{r_1 \mathrm{T}} \tag{24}$$

where $\tilde{L}_{k-1}^{r_0}$ and $\tilde{L}_{k-1}^{r_1}$ contain the first r_0 and the last $r_1 = r - r_0$ columns of $\tilde{L}_{k-1}$, respectively. Since Eq. (24) holds for any orthogonal matrix Θ, one can choose Θ such that the columns of $\tilde{L}_{k-1}$ are orthogonal and ranked according to their norm in increasing order (relative to some metric $\mathcal{M}$ in the state space, see Section 5.2.2). In this way, the matrix $\tilde{L}_{k-1}^{r_1}$ will contribute the most to the filter error representation. Such a matrix Θ can be easily constructed as the matrix whose columns are the eigenvectors of $C_{k-1}^{-1} L_{k-1}^{\mathrm{T}} \mathcal{M} L_{k-\mathrm{T}} C_{k-1}^{-1}$ ranked according to their eigenvalue in increasing order (for simplicity the notation $\mathcal{M}$ denotes both a metric in the state space and the corresponding positive definite matrix).

From the above considerations, we will let only the matrix $\tilde{L}_{k-1}^{r_1}$ evolve (in the same way as in the SEIK

filter) while keeping the matrix $\tilde{L}_{k-1}^{r_0}$ fixed in time. Specifically, we construct the interpolating states $X_1^a(t_{k-1}),\ldots,X_{r_1+1}^a(t_{k-1})$ with mean $X^a(t_{k-1})$ and covariance matrix $\sigma^2\tilde{L}_{k-1}^{r_1}\tilde{L}_{k-1}^{r_1\mathrm{T}}$ and let $\check{L}_{k-1}$ be the matrix $[\tilde{L}_{k-1}^{r_0},[X_1^a(t_{k-1})\cdots X_{r_1+1}^a(t_{k-1})]T]$ where T is as in Eq. (17) with r_1 instead of r. With this construction, we can write

$$P^a(t_{k-1})=\sigma^2\check{L}_{k-1}\begin{bmatrix} I_d & 0 \\ 0 & \sigma^2(r_1+1)T^{\mathrm{T}}T \end{bmatrix}^{-1}\check{L}_{k-1}^{\mathrm{T}}. \tag{25}$$

Then in the forecast stage, we compute the r_1 interpolating states $X_i^f(t_k)$ by applying the model on $X_i^a(t_{k-1})$ and approximate the forecast error covariance matrix by

$$P^f(t_k)=\sigma^2\tilde{L}_{k-1}^{r_0}\tilde{L}_{k-1}^{r_0\mathrm{T}}+\frac{1}{r_1+1}\sum_{i=1}^{r_1+1}[X_i^f(t_k)-X^f(t_k)]\times[X_i^f(t_k)-X^f(t_k)]^{\mathrm{T}}+Q_k. \tag{26}$$

Again, the above matrix can be rewritten as

$$P^f(t_k)=\sigma^2\check{L}_k\begin{bmatrix} I_d & 0 \\ 0 & \sigma^2(r_1+1)T^{\mathrm{T}}T \end{bmatrix}^{-1}\check{L}_k^{\mathrm{T}}+Q_k \tag{27}$$

where $X_i^f(t_k)=M(t_k,t_{k-1})X_i^a(t_{k-1})$ for $i=1,\ldots,r_1+1$, $X^f(t_k)$ is the average of the $X_i^f(t_k)$ and $\check{L}_k=[\tilde{L}_{k-1}^{r_0},[X_1^f(t_k)\cdots X_{r_1}^f(t_k)]T]$.

This representation is the same as Eq. (4) in the SEEK filter but with U_{k-1} replaced by the matrix

$$\begin{bmatrix} I_d & 0 \\ 0 & \sigma^2(r_1+1)T^{\mathrm{T}}T \end{bmatrix}. \tag{28}$$

Therefore, one may apply the same correction step as in the SEEK filter to get U_k and hence $P^a(t_k)$.

The cost of the SSEIK filter is about $(r_1+1)/(r+1)$ times the cost of the SEIK filter. Since the value of r_1 can be often taken equal to 1 or 2, this filter is much less expensive than the SEIK filter.

3.4. Doubling the basis of the SEIK filter

Consider a physical system described by the differential equation

$$\frac{dX}{dt}=F(X(t),t), \tag{29}$$

which is quasi-autonomous in the sense that the gradient of F with respect to its first argument is independent of time (the second argument). Let $\mathbf{F}(\cdot)$ denote this gradient and $M(t_{k+1},t_k)$ be the transition operator from time t_k to t_{k+1}, associated with the system (29).

Recall that in the SEEK filter, the ith column $L_{k,i}$ of the correction basis L_k evolves according to the equation

$$L_{k+1,i}=\mathbf{M}(t_{k+1},t_k)L_{k,i} \tag{30}$$

where $\mathbf{M}(t_{k+1},t_k)$ is the gradient of $M(t_{k+1},t_k)$ evaluated at $X^a(t_k)$. Therefore, the vector function $v(t)=\mathbf{M}(t,t_k)L_{k,i}$ is a solution of the differential system

$$\begin{cases} dv/dt=\mathbf{F}(X^a(t))v(t) \\ v(t_k)=L_{k,i} \end{cases}. \tag{31}$$

Let $v_h(t)=v(t+h)$, we have

$$\frac{d(v_h-v)/h}{dt}=\frac{\mathbf{F}(X^a(t+h))-\mathbf{F}(X^a(t))}{h}v_h(t)+\mathbf{F}(X^a(t))\frac{v_h(t)-v(t)}{h}. \tag{32}$$

If the model is linear, then $\mathbf{F}$ is constant so that the first term in the right hand side of the above equation vanishes. If the model is nonlinear but close to linear, one may still neglect this term. Thus the limit $\mathbf{v}$ of $(v_h-v)/h$ as h tends to zero verifies

$$\frac{d\mathbf{v}}{dt}\approx\mathbf{F}(X^a(t_k))\mathbf{v}. \tag{33}$$

Since, by construction, $\mathbf{v}(t_k)$ and $\mathbf{v}(t_{k+1})$ are no other than $\mathbf{F}(X^a(t_k))L_{k,i}$ and $\mathbf{F}(X^a(t_{k+1}))L_{k+1,i}$ respectively, the above equation is equivalent to

$$\mathbf{F}(X^a(t_{k+1}))L_{k+1,i} \approx \mathbf{M}(t_{k+1},t_k)\mathbf{F}(X^a(t_k))L_{k,i}. \quad (34)$$

One recognizes in the last formula the evolution equation of the correction basis vectors (Eq. (30)), with $\mathbf{F}(X^a(t_k))L_{k,i}$ in place of $L_{k,i}$. Therefore, if $\mathbf{F}(X^a(t_k))L_{k,i}$ was a basis correction vector at time t_k then $\mathbf{F}(X^a(t_{k+1}))L_{k+1,i}$ is too at time t_{k+1}. This remark enables us to "double" the dimension of the correction basis of the SEEK filter without really increasing the computation cost. Indeed, if one initializes the correction basis by

$$\tilde{L}_0 = [L_{0,1}\cdots L_{0,r} \vdots \mathbf{F}(X^a(t_0))L_{0,1}\cdots\mathbf{F}(X^a(t_0))L_{0,r}], \quad (35)$$

then from the evolution equation (Eq. (5)) of the correction basis and using Eq. (34) repeatedly, the correction basis at time t_k is approximately

$$\tilde{L}_k = [L_{k,1}\cdots L_{k,r} \vdots \mathbf{F}(X^a(t_k))L_{k,1}\cdots\mathbf{F}(X^a(t_k))L_{k,r})]. \quad (36)$$

Thus, to compute the new correction basis $\tilde{L}_k$ at time t_k, one needs only to compute its first r columns and then deduce the last r columns via the pre-multiplication of the first ones by the matrix $\mathbf{F}(X^a(t_k))$.

• *On numerical computation*: Often $\mathbf{F}$ can be deduced analytically from the system equations. In this case, one can compute the $L_{k,i}$ by solving Eq. (31) directly so that one gets both $L_{k+1,i}$ and $\mathbf{F}(X^a(t_k))L_{k,i}$. More generally $L_{k+1,i}$ would be computed through an integration scheme involving the intermediate values $M(t_k+hv,t_k)L_{k,i}$, $v=1,\ldots,p$, $h=(t_{k+1}-t_k)/p$ (p being the number of time steps between two observations), then one may approximate $\mathbf{F}(X^a(t_k))L_{k,i}$ by $[M(t_k+h,t_k)L_{k,i}-L_{k,i}]/h$. Thus no extra calculation is required.

As we are more interested in the SEIK than the SEEK filter, we shall now develop the procedure for doubling the correction basis of the former. Recall that the correction basis of the SEIK filter is given in Eq. (16). Thus, based on the above considerations, we propose to double this basis to

$$\bar{L}_k = \left[[X_1^f(t_k)\cdots X_{r+1}^f(t_k)]T \vdots [\dot{X}_1^f(t_k)\cdots\dot{X}_{r+1}^f(t_k)]T\right] \quad (37)$$

where $\dot{X}_i^f(t_k)=F(X_i^f(t_k),t_k)$. Assume that the analysis error covariance matrix $P^a(t_k)$ can be represented in term of the above basis as $P^a(t_k)=\sigma^2\bar{L}_k\bar{U}_k\bar{L}_k^T$ where $\bar{U}_k$ is some $2r\times 2r$ positive definite matrix, we can bring it to a form similar to Eq. (12) as follows. As we usually work with the inverse of $\bar{U}_k$ instead of $\bar{U}_k$, we factorize the upper left $r\times r$ block of $\bar{U}_k^{-1}$ into $C_kC_k^T$ and put

$$\tilde{U}_k^{-1} = \begin{bmatrix} T^T(\Omega_{k-1}C_{k-1}^{-1}) & 0 \\ 0 & T^T(\Omega_{k-1}C_{k-1}^{-1}) \end{bmatrix} \bar{U}_k^{-1} \times \begin{bmatrix} (\Omega_{k-1}C_{k-1}^{-1})^T T & 0 \\ 0 & (\Omega_{k-1}C_{k-1}^{-1})^T T \end{bmatrix} \quad (38)$$

where Ω_k and T are as in the SEIK filter (that is they both have zero columns sum and Ω_k is random with orthonormal columns while T is fixed and needs only to be of full rank). Thus by construction, the upper left $r\times r$ block of $\bar{U}_k^{-1}$ is T^TT. Inverting the above equality, we can express $\bar{U}_k$ in term of $\tilde{U}_k$, which yields the representation $P^a(t_k)=\sigma^2\tilde{L}_k\tilde{U}_k\tilde{L}_k^T$ where

$$\tilde{L}_k = \bar{L}_k \begin{bmatrix} (\Omega_{k-1}C_{k-1}^{-1})^T T & 0 \\ 0 & (\Omega_{k-1}C_{k-1}^{-1})_T^T \end{bmatrix} \quad (39)$$

If the matrix $\tilde{U}_k^{-1}$ is block diagonal, the upper left $r\times r$ block of $\tilde{U}_k$ would be $(T^TT)^{-1}$. For simplicity, we shall proceed as it is the case (even if it is not) and take $(T^TT)^{-1}$ as an approximation to the upper left $r\times r$ block of $\tilde{U}_k$. Further, we shall neglect other blocks which leads to the representation $P^a(t_k)\approx\sigma^2\bar{L}_k(\Omega_kC_k^{-1})^T T(T^TT)^{-1}T^T\Omega_kC_k^{-1}L_k^T$. But since the rows of Ω_k has components summing to zero and thus can be expressed in the basis formed by the rows of T, we have $\Omega_k=T(T^TT)^{-1}T^T\Omega_k$ so that the factor

$T(T^{\mathrm{T}}T)^{-1}T^{\mathrm{T}}$ in the above representation can be dropped. By analogy with Eq. (12), this suggests constructing the interpolating state $X_1^{\mathrm{a}}(t_k),\ldots, X_{r+1}^{\mathrm{a}}(t_k)$ as in the SEIK filter, that is by the formula (13) with L_k now being the matrix formed by the first r columns of $\bar{L}_k$.

The above considerations only serve to motivate our construction of the interpolating states $X_i^{\mathrm{a}}(t_k)$. We will not actually use the above approximation to $P^{\mathrm{a}}(t_k)$. Instead we derive a much more accurate approximation which is exact when the model is linear. We begin by looking for an approximation to $\tilde{L}_k$ and since its first r columns are no other than those of $[X_1^{\mathrm{a}}(t_k)\ \ldots\ X_{r+1}^{\mathrm{a}}(t_k)]T$, we need only to consider its last r columns, which are those of $[\dot{X}_1^{\mathrm{f}}(t_k)\ldots\dot{X}_{r+1}^{\mathrm{f}}(t_k)]T(\Omega_k C_k^{-1})^{\mathrm{T}}T$. Observe that $X_i^{\mathrm{a}}(t_k)$ equals $X^{\mathrm{a}}(t_k)$ plus the ith column of $[X_1^{\mathrm{f}}(t_k)\ldots X_{r+1}^{\mathrm{f}}(t_k)]T(\Omega_k C_k^{-1})^{\mathrm{T}}$ and is thus a linear combinations of $X_1^{\mathrm{f}}(t_k),\ldots,X_{r+1}^{\mathrm{f}}(t_k)$ with coefficients summing to 1, since the same is true for $X^{\mathrm{a}}(t_k)$, as will be seen below, and each column of T has its elements summing to 0. Thus by approximating the function $F(\cdot,t_k)$ by its linear interpolation based on $X_1^{\mathrm{f}}(t_k),\ldots, X_{r+1}^{\mathrm{f}}(t_k)$, one can write $F(X_i^{\mathrm{a}}(t_k),t_k)$ as linear combinations of $F(X_1^{\mathrm{f}}(t_k),t_k),\ldots,F(X_{r+1}^{\mathrm{f}}(t_k),t_k)$ with the same coefficients. Therefore, letting $\dot{X}_i^{\mathrm{a}}(t_k)=F(X_i^{\mathrm{a}}(t_k),t_k)$ and noting that $F(X_i^{\mathrm{f}}(t_k),t_k)=\dot{X}_i^{\mathrm{f}}(t_k)$, we get

$$[\dot{X}_1^{\mathrm{a}}(t_k)\cdots\dot{X}_{r+1}^{\mathrm{a}}(t_k)] = F(X^{\mathrm{a}}(t_k),t_k) + [\dot{X}_1^{\mathrm{f}}(t_k)\cdots\dot{X}_{r+1}^{\mathrm{f}}(t_k)] \times T(\Omega_k C_k^{-1})^{\mathrm{T}} \tag{40}$$

Finally, we get the approximation

$$\tilde{L}_k = \left[[X_1^{\mathrm{a}}(t_k)\cdots X_{r+1}^{\mathrm{a}}(t_k)]T\ \vdots\ \dot{X}_1^{\mathrm{a}}(t_k)\cdots\dot{X}_{r+1}^{\mathrm{a}}(t_k)]T\right] \tag{41}$$

and $P^{\mathrm{a}}(t_k)=\sigma^2\tilde{L}_k\tilde{U}_k\tilde{L}_k{}^{\mathrm{T}}$.

The above representation has the same form as in the SEIK filer, therefore we can proceed in the same way. More precisely, the algorithm operates as follows.

(0) *Initialization*: As in the SEEK filter, one starts at the initial time t_0 from an analysis state $X^{\mathrm{a}}(t_0)$ and a low rank approximation $P^{\mathrm{a}}(t_0)$ of the error covariance matrix. The first $r+1$ interpolating states $X_i^{\mathrm{a}}(t_0)$ is drawn as usual by Eq. (13). The number of interpolating states is next doubled by considering the new $r+1$ interpolating states $\dot{X}_i^{\mathrm{a}}(t_0)=F(X_i^{\mathrm{a}}(t_0),t_0)$. One can then consider as initial correction basis the matrix $\bar{L}_0=[[X_1^{\mathrm{a}}(t_0)\cdots X_{r+1}^{\mathrm{a}}(t_0)]T\vdots[\dot{X}_1^{\mathrm{a}}(t_0)\cdots\dot{X}_{r+1}^{\mathrm{a}}(t_0)]T]$. The first correction can then be done as in the SEIK filter.

(1) *Drawing interpolating states*: At time t_k, one should have an analysis vector $X^{\mathrm{a}}(t_k)$ with an error covariance matrix $P^{\mathrm{a}}(t_k)=\bar{L}_k\bar{U}_k\bar{L}_k^{\mathrm{T}}$ with $\bar{L}_k$ of the form (37). Further $X^{\mathrm{a}}(t_k)$ is a linear combination of the $X_i^{\mathrm{f}}(t_k)$ with coefficient summing to 1. One factories the upper left $r\times r$ block of $\bar{\mathrm{U}}_k^{-1}$ into $C_kC_k^{\mathrm{T}}$, then draws the interpolating states $X_1^{\mathrm{a}}(t_k),\ldots, X_{r+1}^{\mathrm{a}}(t_k)$ according to Eq. (13), as in the SEIK filter but with the above C_k and with L_k the matrix formed by the first r columns of $\bar{L}_k$. Finally, one computes $\tilde{U}_k^{-1}$ by Eq. (38) and $\tilde{L}_k$ by Eq. (41).

(2) *Forecasting*: One applies the model operator to bring the $X_i^{\mathrm{a}}(t_k)$ to $X_i^{\mathrm{f}}(t_{k+1})$, then takes the average of the latter as forecast state $X^{\mathrm{f}}(t_k)$. Then one constructs the new correction basis $\bar{L}_{k+1}$ by Eq. (37) with $k+1$ in place of k. The forecast error covariance matrix may be obtained by $P^{\mathrm{f}}(t_{k+1})=\bar{L}_{k+1}\tilde{U}_k\bar{L}_{k+1}^{\mathrm{T}}+Q_{k+1}$, but it is not needed.

(3) *Correction*: One computes $\bar{U}_{k+1}^{-1}$ similarly as in the SEIK filter, that is by the right hand side of Eq. (8), with $k-1$ replaced by k and then U_k replaced by $\tilde{U}_k$ and L_{k+1} replaced by $\bar{L}_{k+1}$ but $\mathbf{H}_{k+1}\mathrm{L}_{k+1}$ is replaced by $(HL)_{k+1}$ as defined in the SEIK filter. The forecast is then corrected in the usual ways via formulas (6) and (7), again with the above replacements. As the columns of $\bar{L}_{k+1}$, hence those of G_{k+1} are linear combinations of $X_1^{\mathrm{f}}(t_{k+1}),\ldots,X_{r+1}^{\mathrm{f}}(t_{k+1})$ with coefficients summing to 0, $X^{\mathrm{a}}(t_{k+1})$ is indeed also a linear combination of these vectors, with coefficients summing to 1, as required.

• *Numerical computation of* $\dot{X}_i^{\mathrm{f}}(t_k)$: One can use the time differencing Euler scheme of the system (29) to compute

$$F(X_i^{\mathrm{f}}(t_k),t_k)=\frac{1}{\delta}[M(t_k+\delta,t_k)X_i^{\mathrm{f}}(t_k)-X_i^{\mathrm{f}}(t_k)], \tag{42}$$

where δ is the time step of the numerical model.

4. Tracking model instabilities

A large number of studies have made clear that rapid error growth in periods of baroclinic and barotropic instability is nonmodal (Farrell, 1989; Moore and Farrell, 1994; Trefethen et al., 1993). Cohn and Tolding (1996) noted that all their degraded EK filters failed to capture the instability, and generally diverge. The degraded forms of the SEIK filter presented in Section 3 cannot be an exception to this observation since the assumptions on which we have constructed them are no longer justified in unstable periods. To overcome this problem, we introduce two adaptive tuning schemes: the first controls the correction basis evolution and the second makes use of a variable forgetting factor.

4.1. Adaptive tuning of the correction basis evolution

In our experiments we noticed that the SEIK filter is more or less well-behaved in the presence of instability. This motivates us to let the correction basis evolve as in a degraded SEIK filter when the model is stable and as in the SEIK filter when the model is unstable.

To detect the periods of model instability, one can track the filter's state by computing an instantaneous average and a long term average of the prediction error variance, denoted by s_k and l_k, respectively. If $cs_k \leq l_k$ (c is a tuning constant), one may assume that steady conditions have been achieved and consider that the model is in a stable period (we would expect that s_k is close to l_k and if it is much less than l_k, we attribute this to chance). In this case, we let the correction basis evolve according to a degraded SEIK filter. If $cs_k > l_k$, this is an indication that the correction basis is not able to capture all the unstable modes of the dynamics and this is often associated with the fact that the model is in a period of rapid change or possesses too many unstable modes. Thus, it is then better to use the full SEIK evolution equation for the correction basis (in any case as the filter performance decreases, it is reason enough to switch to a better but more costly filter). The value of c is normally 1 but can be chosen slightly different than 1 to facilitate or to hinder somewhat the switch, this choice could be determined by experiments on a case by case basis. (Note that s_k and l_k are only estimates of the short and long term prediction errors variances, so that $s_k > l_k$ does not mean that their true values are in this order.)

The estimates of s_k and l_k are computed recursively as follows

$$s_k = \alpha s_{k-1} + (1-\alpha) \parallel Y_k^{\mathrm{o}} - H_k X^{\mathrm{f}}(t_k) \parallel^2, \tag{43}$$

$$l_k = \beta l_{k-1} + (1-\beta) \parallel Y_k^{\mathrm{o}} - H_k X^{\mathrm{f}}(t_k) \parallel^2 \tag{44}$$

where α and β are constants chosen such that $\beta \lesssim 1$ and $\alpha < \beta$.

4.2. Adaptive tuning of the forgetting factor

There are three reasons behind the use of the forgetting factor in the SEIK (and the SEEK) filter. Firstly, it limits the effective filter memory length by discarding old data. This will attenuate the error propagation and enable the SEIK filter to follow system changes. Secondly, it sets up the gain matrix to avoid the "blow up" phenomena (see Astrom and Wittenmark, 1995). Thirdly, it does not require any extra cost for its implementation: the filter equations remain unchanged except for the emergence of the forgetting factor ρ in the time propagation error covariance equation (see Pham et al., 1998). Specifically, the updating equation for U_k now changes to

$$\begin{aligned} U_k^{-1} = & \Big[[\rho\sigma^2 (r+1) T^{\mathrm{T}} T]^{-1} \\ & + (L_k^{\mathrm{T}} L_k)^{-1} L_k Q_k L_k (L_k^{\mathrm{T}} L_k)^{-1} \Big]^{-1} \\ & + (HL)_k^{\mathrm{T}} R_k^{-1} (HL)_k . \end{aligned} \tag{45}$$

With $\rho = 1$, all data have the same weight, but with $\rho < 1$, recent data are exponentially more weighted than old data.

However, the use of a too small forgetting factor when the system evolution is stable would degraded the filter performance especially when there is little information in the measurements. To maximize the benefit of the forgetting factor, we propose to use a variable one (Sorenson and Sacks, 1971): such factor should be close to 1 when the model is stable and much less then 1 when the model is unstable. Indeed, old data should be forgotten more in the last case to adapt the filter to the new model's mode. Therefore,

we will give the forgetting factor the value ρ_1^* or ρ_2^* according to the relative magnitudes of the instantaneous and long-term prediction errors s_k and l_k, i.e.

$$\rho_k = \begin{cases} \rho_1^* \lesssim 1 & \text{if} \quad cs_k < l_k, \\ \rho_2^* < \rho_1^* & \text{if} \quad cs_k \geq l_k. \end{cases}$$

Here, s_k and l_k are the same estimates for the short and the long term average prediction error variances, defined in Section 4.1, and c is again a tuning constant similar to the one used in this section (but need not be exactly the same). In this way, the adaptive scheme and the one in Section 4.1 can be implemented simultaneously. The goal of the present adaptive scheme is to choose ρ to optimize the filter performance. The long term estimated prediction error variance would be, by design, related to the average effect of this parameter. If our adaptive scheme is successful, this variance would be nearly optimal, that is, it cannot be significantly reduced. However, the short term estimated prediction error variance would be related to the current value of ρ and could rise significantly if the system enters a period of quick changes (instability). Thus it makes sense to reduce then the value of ρ to make the filtered state depending more on the observation and less on the model equation.

A different approach proposed in Dee and da Silva (1999) is to consider ρ as a unknown parameter of the forecast error covariance matrix, then estimate it adaptively (on-line) by the maximum likelihood method. Mitchell and Houtekamer (2000) follow a similar approach but parameterize directly the model error covariance matrix (which is more natural) in the context of the ENK filter. However, this can be much more costly than our simple adjustment of the forgetting factor.

5. Application to altimetric data assimilation in the OPA model of the tropical Pacific

In order to evaluate the performance of the degraded SEIK filters, we have implemented them in a realistic setting of the OPA model in the tropical Pacific ocean, under the assumption of a perfect model ($Q_k = 0$). The assimilation is based on the pseudo-observations, which are extracted from twin experiments. The SEIK filter is used as a reference to compare the performance of these new filters. The configuration and the characteristics of the model used in our experiments are presented in the next section.

5.1. OPA model in tropical pacific

In this section, we briefly present the model basics and a description of our configuration.

5.1.1. Model description

The OPA model (OPA for Océan PArallélisé) is a primitive equation ocean general circulation model which has been developed at the LODYC laboratory (Laboratoire d'Océanographie DYnamique et de Climatologie) to study large scale ocean circulation. It solves the Navier–Stokes equations which express the momentum balance, the hydrostatic equilibrium, the incompressibility, the heat and salt balance and a nonlinear realistic equation of state plus the rigid lid assumption and some hypothesis made from scale considerations. The system equations are written in curvilinear z-coordinates and discretized using the centered second order finite difference approximation on a three dimension generalized "C-grid Arakawa". In this scheme, the scalar variables are computed in the center of the cells and the vector variable in the center of cell faces (see Arakawa, 1972 for details). Time stepping is achieved by two time differencing schemes: a basic leap-frog scheme associated to an Asselin filter for the non-diffusive processes and a forward scheme for diffusive terms. The sub-grid scale physics is a tracer diffusive operator of second order on the vertical, the eddy coefficients being computed from a turbulent closure model (see Blanke and Delecluse, 1993). On the lateral, diffusive and viscous operators can be either of second or of fourth order. The reader is referred to the OPA reference manual (Madec et al., 1997) for more details.

5.1.2. Model configuration

The model domain covers the entire tropical Pacific basin extending from 120°E to 70°W and from 33°S to 33°N and the level depth varies from 0 at the sea surface to 4000 m. The number of horizontal grid points is 171×59 on 25 vertical levels. The model equations are solved on an isotropic horizontal grid with a zonal resolution 1° and a meridional

resolution maximal at the equator (0.5°) and goes down to 2° to the north and south boundaries. The vertical resolution is approximately 10 m from the sea surface to 120 m depth then decreases to 1000 m at the sea bottom. The time step is 1 h.

The bathymetry is relatively coarse. It was obtained from Levitus data's mask (Levitus, 1982). The forcing fields are interpolated from the ECMWF (European center for medium-range weather forecasts) reanalysis with monthly variability. It is composed of wind stress, heat and fresh water fluxes. Zero fluxes of heat and salt and non-slip conditions are applied at solid boundaries.

A second order horizontal friction and diffusion scheme for momentum and tracers is chosen with a coefficient of 2000 m^2/s in the strip 10°N–10°S and increase up to 10000 m^2/s at the north and south basins boundaries. The static instabilities are resolved in the turbulent closure scheme.

The model starts from rest (i.e. with zero velocity field). The salinity and the temperature stem from seasonal climatologic Levitus data (Levitus, 1982).

5.2. Experiments design

5.2.1. The state vector

The state vector is the set of prognostic model variables that must be initialized independently. Since the prognostic variables of the OPA model are the zonal U and meridional V velocities, the salinity S and the temperature T, one should consider the state vector

$$X^{\mathrm{t}} = (U, V, S, T)^{\mathrm{T}}. \tag{46}$$

However, the observed variable, which is the sea surface height SSH, is a diagnostic variable computed from the barotropic velocity by a complex nonlinear algebraic equation. Thus, the observation operator H, which relates this state vector to the observed variable, will be nonlinear and very difficult to compute. To avoid these difficulties, we will adopt a pseudo-state vector in our experiments which contains the true state vector augmented by the SSH, namely

$$X^{\mathrm{t}} = (U, V, S, T, \mathrm{SSH})^{\mathrm{T}}. \tag{47}$$

In this case, H will always be linear of the form $[0 \vdots I_d]$. Of course, the dimension of the state vector will increase but the increase is insignificant since the SSH is computed only on the sea surface. More precisely, the number of state variables is now $4 \times 171 \times 59 \times 25 + 171 \times 59 = 1\,018\,989$ instead of $4 \times 171 \times 59 \times 25 = 1\,008\,900$.

5.2.2. Filters initialization

Following the strategy explained in Pham et al. (1997), the choice of the initial state estimator flow field and the corresponding error covariance matrix is made through a simulation of the model itself. Note that this is only done once, and the results can be used to initialize all our degraded SEIK filters. In the present study, the data for the assimilation experiments is again simulated but in an unrelated way with the above simulation.

Thus, in a first experiment, the model has been spun up for 7 years from 1980 to 1986 with the aim to reach a statistically steady state of mesoscale turbulence. Next, another integration of 4 years is carried out from 1987 to 1990 to generate a historical sequence H_{S} of model realization. A sequence of 480 state vectors was retained by storing one state vector every 3 days to reduce the calculation since successive states are quite similar. Because the state variables in Eq. (47) are not of the same nature, we shall in fact apply a multivariate EOF analysis. We define a metric $\mathscr{M}$ in the state space to make the distance between state vectors independent from unit of measure. We choose $\mathscr{M}$ as the diagonal matrix with diagonal elements being the inverse of the spatial variances of each state variables, namely U, V, S, T and SSH, average over the grid points. The initial error covariance matrix $P^{\mathrm{a}}(t_0)$ is then estimated via an EOF analysis on the sample H_{S} (Pham et al., 1997).

Fig. 1 plots the number of EOFs and the percentage of variability (or inertia) contained in the sample H_{S} they explain. From this result, we have chosen to retain $r = 30$ EOFs in all assimilation experiments, as this achieve 85% of the inertia of the sample and this percentage is not much increased for higher value of r.

5.2.3. Data and filters validation

Twin experiments are used to assess the performances and the capabilities of our filters. A reference experiment is performed and the reference X^{t} saved to be later compared with the fields produced during the assimilation experiments. More precisely, a sequence

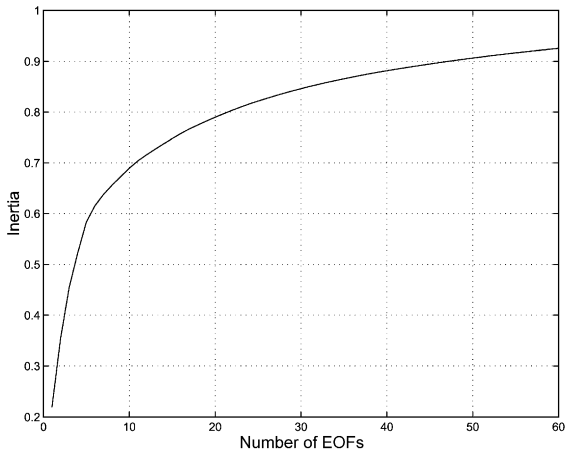

Fig. 1. Percentage of inertia versus the number of retained EOFs.

of 250 state vectors was retained every 24 h during the period of March 1, 1991 to November 10, 1991.

The assimilation experiments are performed using the pseudo-measurements which are extracted from the reference experiment. The SSH are assumed to be observed at every grid points of the model surface with a nominal accuracy of 3 cm. The observation error is simulated by adding randomly generated Gaussian noise to the synthetic observations of SSH. Note that in the assimilation interval, a period of very strong model instability occurs between July and September (see Fig. 2).

Finally, the performance of all our filters is evaluated by comparing the relative root mean square (RRMS) error for each state variable, in each layer or in the whole domain. The RRMS is defined as

$$\mathrm{RRMS}(t_k) = \frac{\|X^{\mathrm{t}}(t_k) - X^{\mathrm{a}}(t_k)\|}{\|X^{\mathrm{t}}(t_k) - \bar{X}\|}, \qquad (48)$$

where X^{a} is the analysis state obtained by the filter, $\bar{X}$ is the mean state of the sample H_{S} and $\|\cdot\|$ denotes the Euclidean norm. Note that the error is relative to the free-run error since the denominator represents the error when there is no observation and the analysis vector is simply taken as the mean state vector.

5.3. Results of assimilation experiments

We first present the results of the SEIK filter and compare them to those of the SEEK filter. Next, we implement the degraded SEIK filters to study their performances with respect to the SEIK filter. Finally, we investigate the effect of our two adaptive tuning schemes presented in Section 4 on the performance of our filters.

5.3.1. SEIK and SEEK filters

The SEIK filter has been implemented with a fixed forgetting factor equal to $\rho = 0.8$. It can be seen in Figs. 3–5 that it performs very well both in the upper layers and in the lower layers. Although its performance appears to degrade somewhat in the presence of instability, the filter behaves satisfactorily during this period. It may appear that the meridional velocity V is not sufficiently well assimilated because the assimilation error is only reduced to almost a half. However, since the velocity field of the tropical Pacific Ocean is essentially zonal, the meridional velocity fields are generally well approached by the average meridional velocity. This means that the free run error is already low and therefore it would be hard to reduce it much further.

We have presented the results of our experiments for the SEIK filter in both the upper and lower layers for completeness. But we have noticed that, for our new filters, the difference between their RRMS and that of the SEIK filter computed in all the layers are quite similar to that computed on each layer. There-

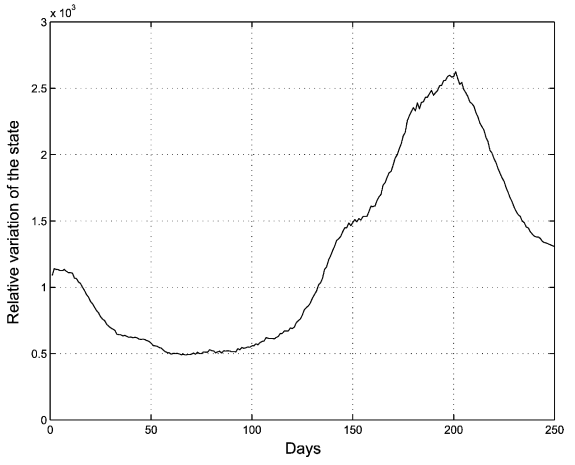

Fig. 2. Relative variation of the state vector in the assimilation period.

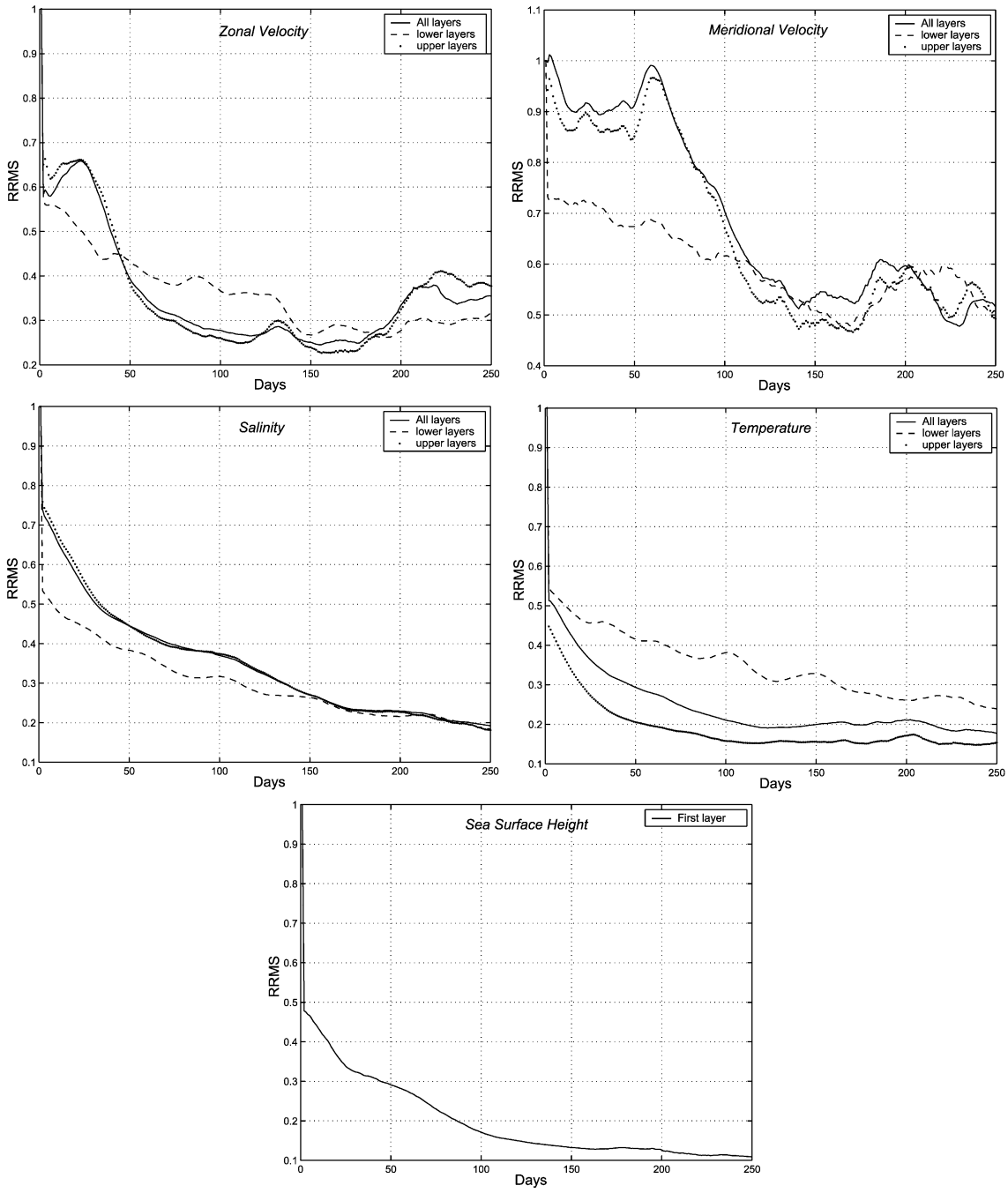

Fig. 3. Evolution in time of the RRMS for the SEIK filter on the whole model domain, on the (mean of 5) upper and the (mean of 5) lower layers.

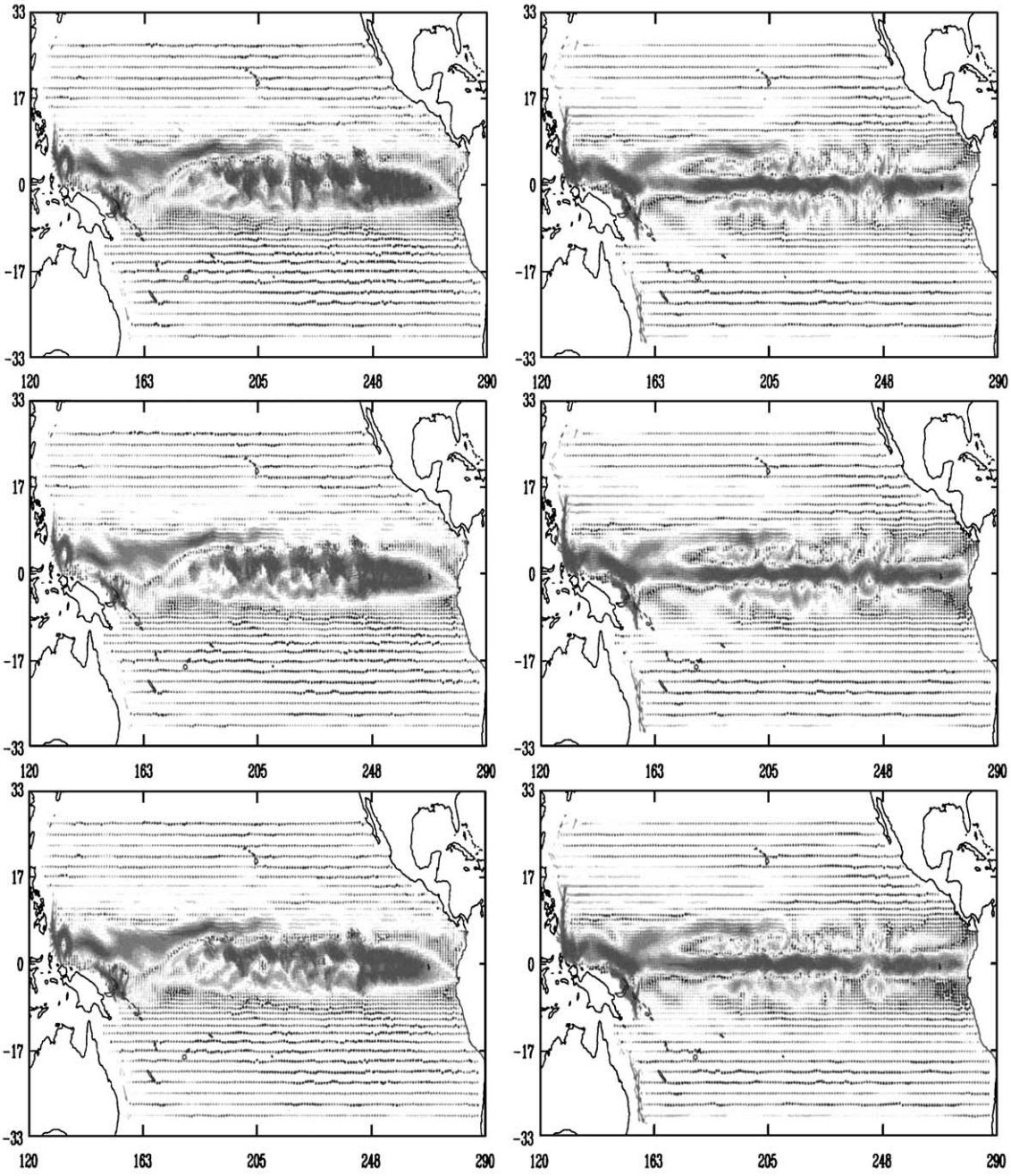

Fig. 4. Maps of ocean velocity on September 1st 90 in the uppermost (left) and the 17th (right) layers: from the SEIK filter (top); from the reference (middle); and from the SEEK filter (bottom).

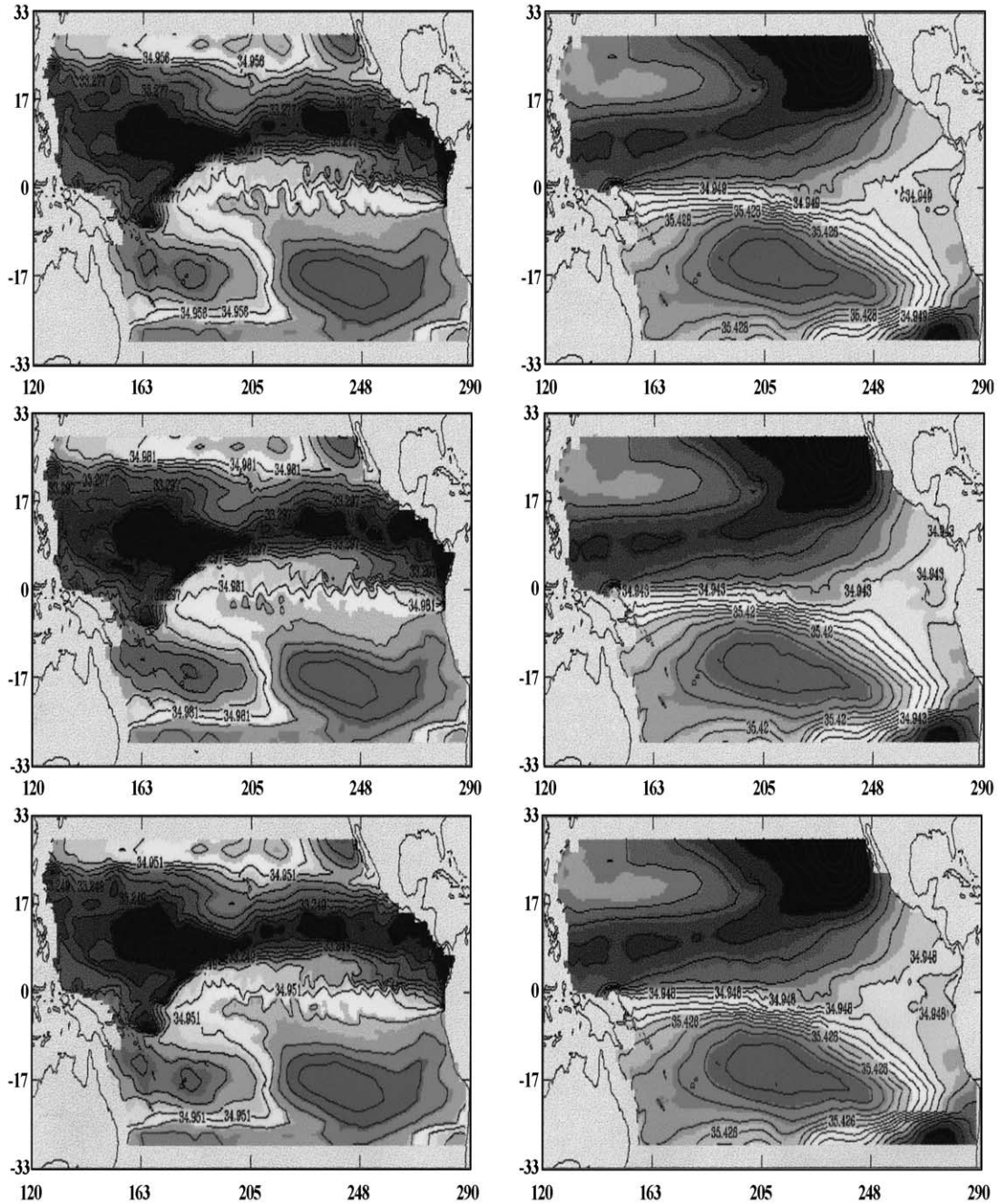

Fig. 5. Maps of sea salinity on September 1st 90 in the uppermost (left) and the 17th (right) layers: from the SEIK filter (top); from the reference (middle); and from the SEEK filter (bottom).

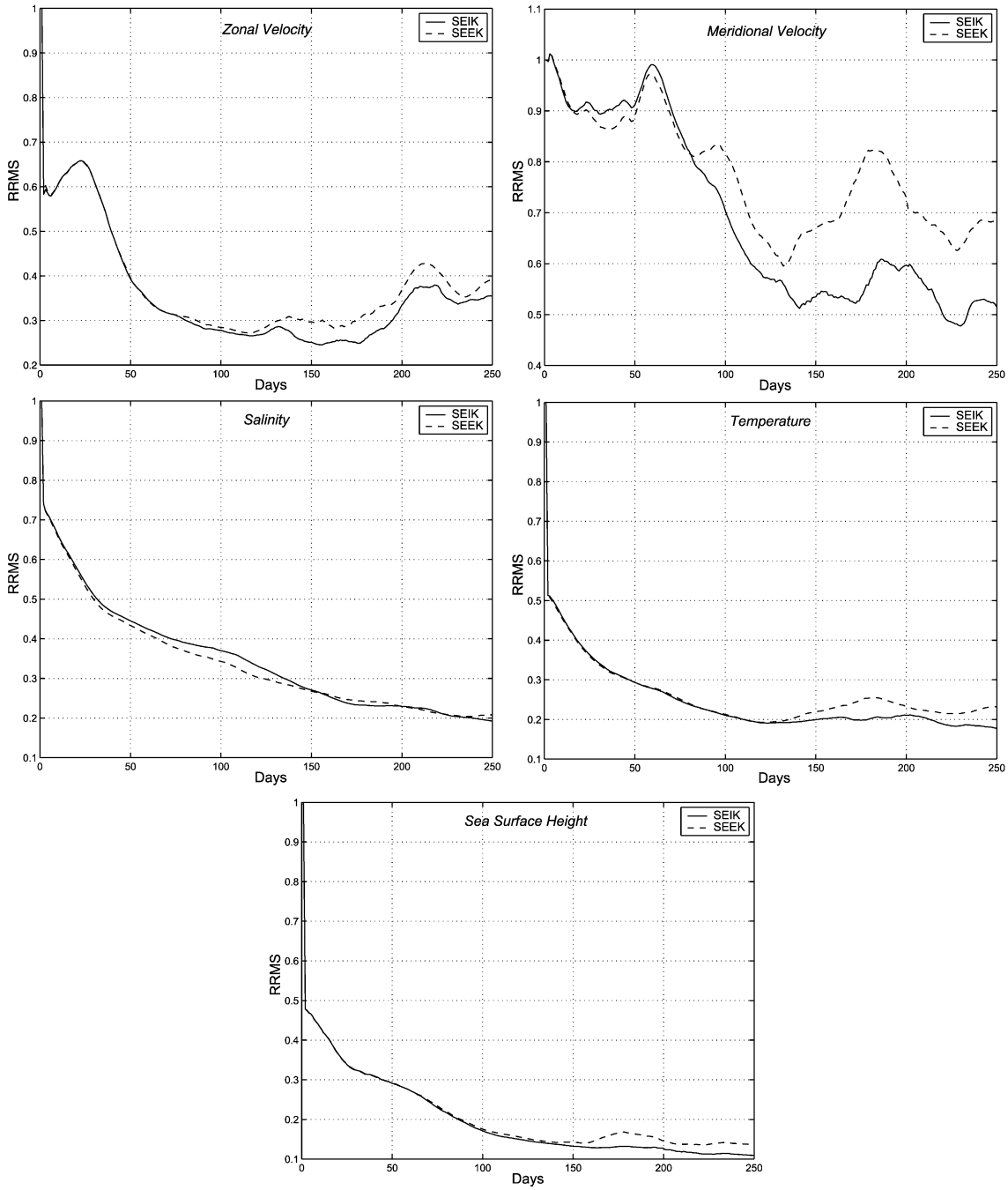

Fig. 6. Evolution in time of the RRMS for the SEIK and SEEK filters.

fore, in the sequel we will only present results in all layers, to save space.

We have also conducted a preliminary experiment with the SEEK filter, which confirms the advantage of the interpolation in the SEIK filter over the linearization used in the SEEK filter. Fig. 6 plots the assimilation results of these two filters using the same setup as before. Clearly, the SEIK filter performs better, particularly with regard to the velocity components U and V, in the unstable period. The SEIK filter thus seems to be more robust than the SEEK filter, in the sense that it can support more deviation from linearity without being broken. Therefore in the sequel we will be interested only in the variants of the SEIK filter and shall evaluate their performance against this filter.

5.3.2. Degraded SEIK filters

In order to study the numerical performance of the degraded SEIK filters with respect to the SEIK filter, we have implemented them under the same previous setup. For the SSEIK filter, only one vector out of 30 of the correction basis evolves. For the SIEIK filter, the whole basis of 30 vectors was to evolve once every two filtering steps after an initialization period of 10 steps (using the SEIK filter). Concerning the SDEIK filter, an evolutive basis of 15 vectors was "doubled" as in Eq. (37) to yield a basis of 30 vectors. Thus, the SFEK, SSEIK, SIEIK and the SDEIK filters are respectively about 30, 15, 2 and 2 times faster than the SEIK filter (Table 1 provides a summary of the cost and performance for all the filters that we have implemented in this section with respect to the SEIK filter). Results of the experiments are plotted in Fig. 7.

The SFEK filter is relatively well-behaved in the stable period. Its performance degrades when model instability appears. This suggests that the EOF analysis has failed to capture a large part of the ocean variability, particularly those occurring in the unstable periods. Indeed the performance of this filter depends highly on the representativeness of the correction basis obtained from the EOF analysis.

The results of the SIEIK filter are very encouraging. It seems to be reasonably well behaved in the unstable period thanks to the update of the correction basis. The good performance of this filter seems to suggest the existence of the semi-fixed mode for an autonomous nearly linear model. We have also noticed, in other experiences not presented here, that the results of this filter depend mostly on the fixed and the keep up modes length noted K_f and K_u, respectively. Its performance is a decrease function of K_f and an increased function of K_u, as one would expect.

Table 1
Summary of the filters costs and performances with respect to the SEIK filter (the cost of which is slightly higher than $(r+1)$ times a model integration, where r is the dimension of the correction basis)

Filter	Cost ratio relative to the SEIK filter	Performance with respect to the SEIK filter
SFEK	1/30	Reasonable when the model is stable and bad when instabilities appear
SFEK-adaptive	1/30	Acceptable
SSEIK	1/15	Good when the model is stable and bad when instabilities appear
SSEIK-adaptive	1/15	Good performance
SIEIK (1/2)	1/2	Very good performance
SIEIK-adaptive	2/5	Very good performance
SDEIK	1/2	Very good when the model is stable and very bad when instabilities appear
SDEIK-adaptive	3/5	Very good performance

Concerning the results of the SSEIK filter, it performs very well in the stable period. However, its performances degrade quickly during the unstable period.

Finally, with regard to the results of the SDEIK filter which are in some way a bit disappointing, we have started this filter by using the SEIK to let the correction basis settle into a "more stable" mode. Despite a very good behavior in the stable period, this filter diverges somewhat especially in the upper layers, during the unstable period (see Fig. 12). The poor performance of the SDEIK filter on the upper layers let us presume that it is very sensible to the forcing fields. Note that the results of this experiment was plotted together with the results of the SDEIK filter with the adaptive tuning correction basis evolution scheme (presented in Section 4) in order to show the usefulness of this scheme.

5.3.3. Adaptive tuning of the forgetting factor

We discuss here the results of several experiments conducted to study the sensibility of our filters with

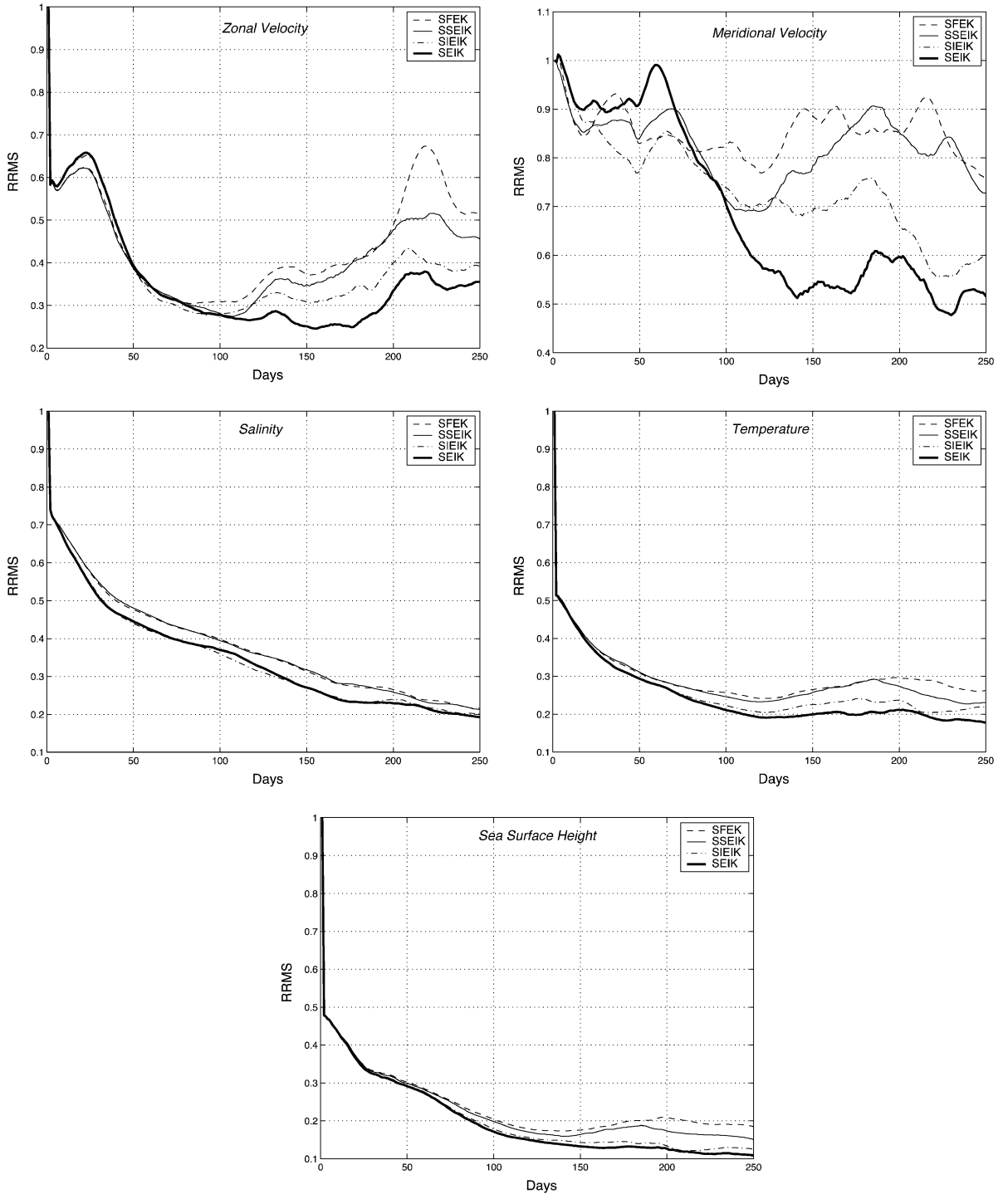

Fig. 7. Evolution in time of the RRMS for the SEIK, SFEK, SSEIK and SIEIK filters.

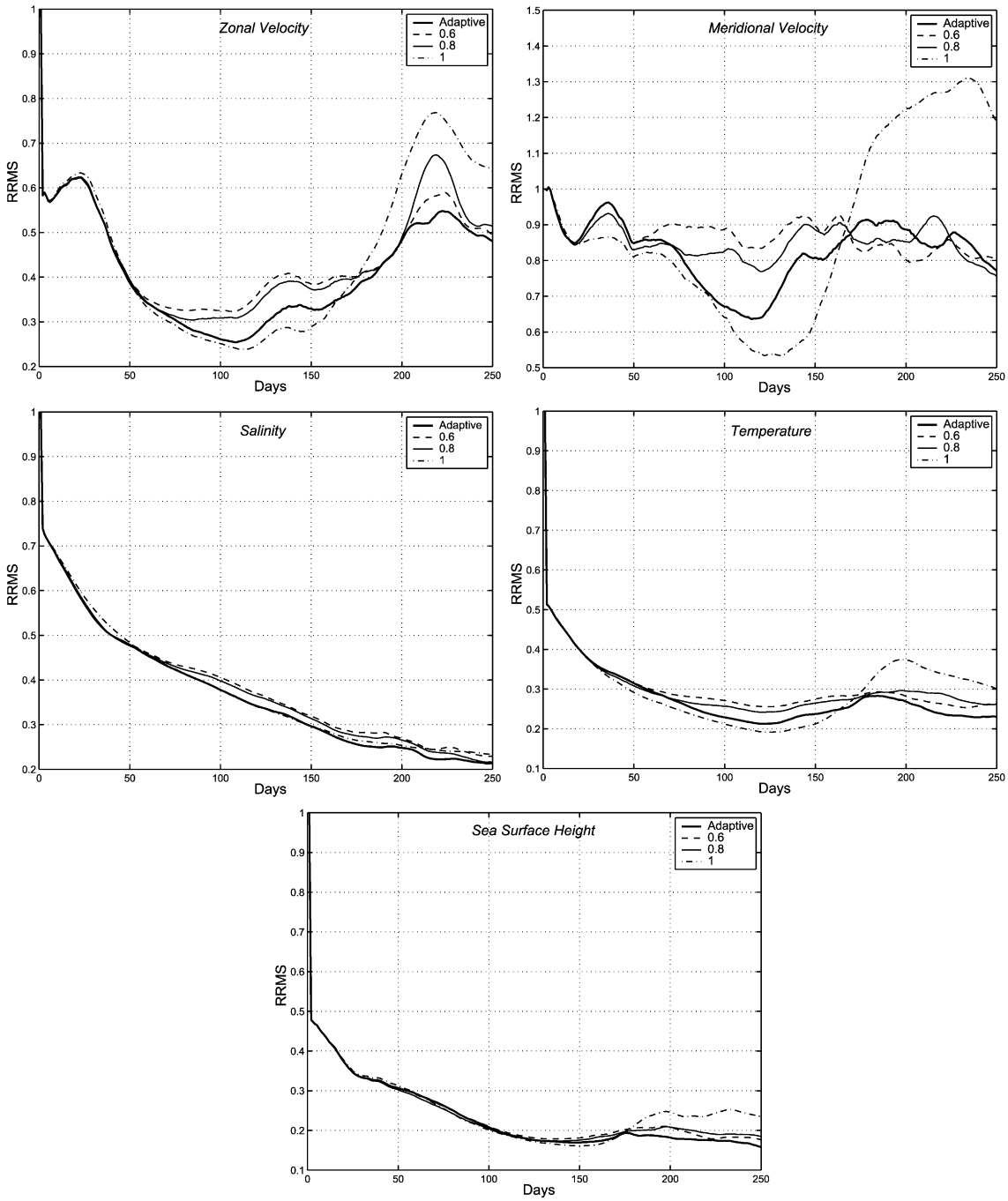

Fig. 8. Evolution in time of the RRMS for the SFEK filter with different values of the forgetting factor and with adaptive forgetting factor.

respect to the forgetting factor ρ and to test the usefulness of the adaptive tuning scheme on ρ described in Section 4. Since one can expect that the effects of the forgetting factor on different degraded SEIK filters are quite similar, we have only conduct experiment with the SFEK filter to save the computation cost. Therefore, we consider, on the one hand a fixed forgetting factor with three different values 1, 0.8 and 0.6 and on the other hand, a variable forgetting factor which takes value 1 or 0.6 according to the relative magnitudes of the short-term and long-term prediction error s_k and l_k. The initial values s_0 and l_0 were taken as $\| Y_0^o - H_k X^f(t_0) \|^2$ to make sure that ρ takes the value 0.6 during early assimilation period. The values of the constants c, α and β were chosen as 1.001, 0.95 and 0.9, respectively.

Fig. 8 shows the RRMS error for these experiments: as expected, the SFEK filter diverges for $\rho = 1$ when model instability appears. For $\rho = 0.8$, the SFEK performs relatively well but its results are not as good as for $\rho = 1$ in the stable period. Finally, the results of this filter are improved in the unstable period for $\rho = 0.6$, however, its performance degraded in the stable period. We conclude from these experiments that the nearer to 1 the value of ρ is, the better the SFEK performs in the stable period but we observe the opposite phenomena in the unstable period (under some limit on the minimum value of ρ). This confirms our argument on how to adapt the forgetting factor ρ. Furthermore, these results show the efficiency of our adaptive tuning scheme for the forgetting factor and, as can be seen form Fig. 9, for the detection of the unstable periods.

In other assimilation experiments, we study the usefulness of a variable forgetting factor on SSEIK, SIEIK and SDEIK filters under the same previous setting. The results for the SSEIK filter are plotted in Fig. 10. It can be seen that the performance of the SSEIK filter is significantly enhanced, particularly in the unstable period. A similar, but not as significant, improvement has been observed on the SIEIK filter. However, this scheme has no significant effect on the SDEIK filter.

5.3.4. Adaptive tuning of the correction basis evolution

To test the relevance of the adaptive tuning scheme of the correction basis L_k evolution, we first compare

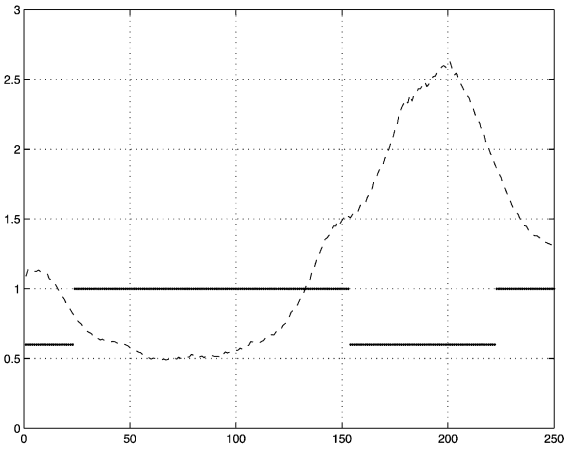

Fig. 9. Evolution in time of the forgetting factor (solid line) to show its relation with the relative variation of the state vector (the curve in Fig. 2 is reproduced in dotted line, rescaled by a factor of 10^3).

the performance of the SIEIK filter, with and without this adaptation scheme, in the same situation as before. We chose to let the correction basis L_k evolve once every four filtering steps and the forgetting factor was also adapted, taking the values 1 and 0.8. Note that we have increased the second value of the forgetting factor because there is no need to use a small value when the correction basis also evolves as in the SEIK filter, since the last is sufficiently stable during the unstable period. It can be seen from Fig. 11 that the adaptive tuning of the evolution of L_k noticeably improves the performance of the SIEIK filter.

The same previous set up has been used in experiments with SDEIK, SFEK and SSEIK filters. In Fig. 12, one can clearly see by comparing the results of the SDEIK filter with and without this adaptation scheme that this filter was completely stabilized in the first case. Concerning the SFEK and the SSEIK filter and particularly for the last one, this adaptation scheme has made no significant improvement. This can be explained by the fact that the convergence of the correction basis towards the amplification error directions was very slow during the unstable period because of the initial value of L_k (just before the evolution) which badly represents the unstable periods of the model.

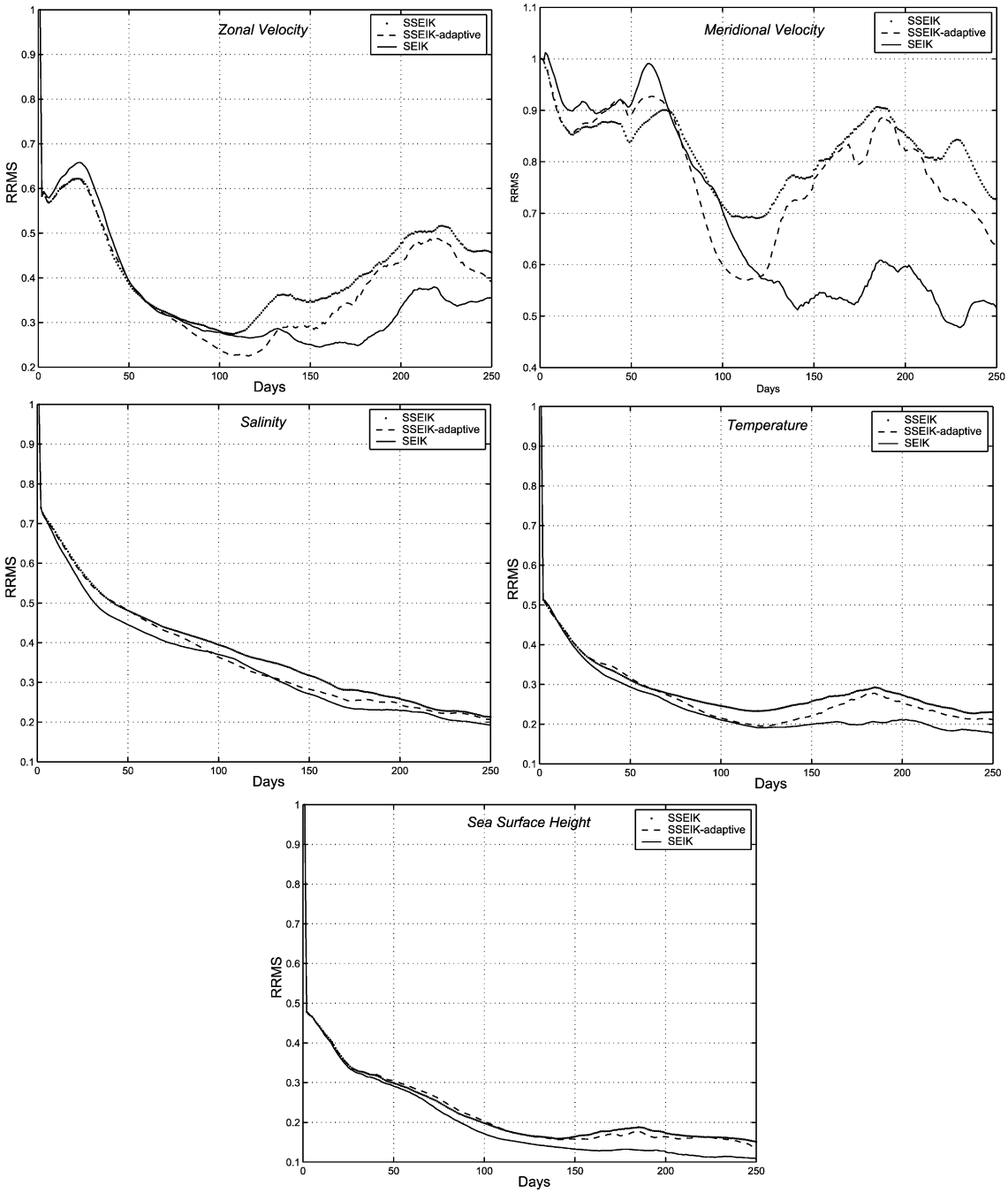

Fig. 10. Evolution in time of the RRMS for the SSEIK filter with and without adaptation of the forgetting factor and the SEIK filter.

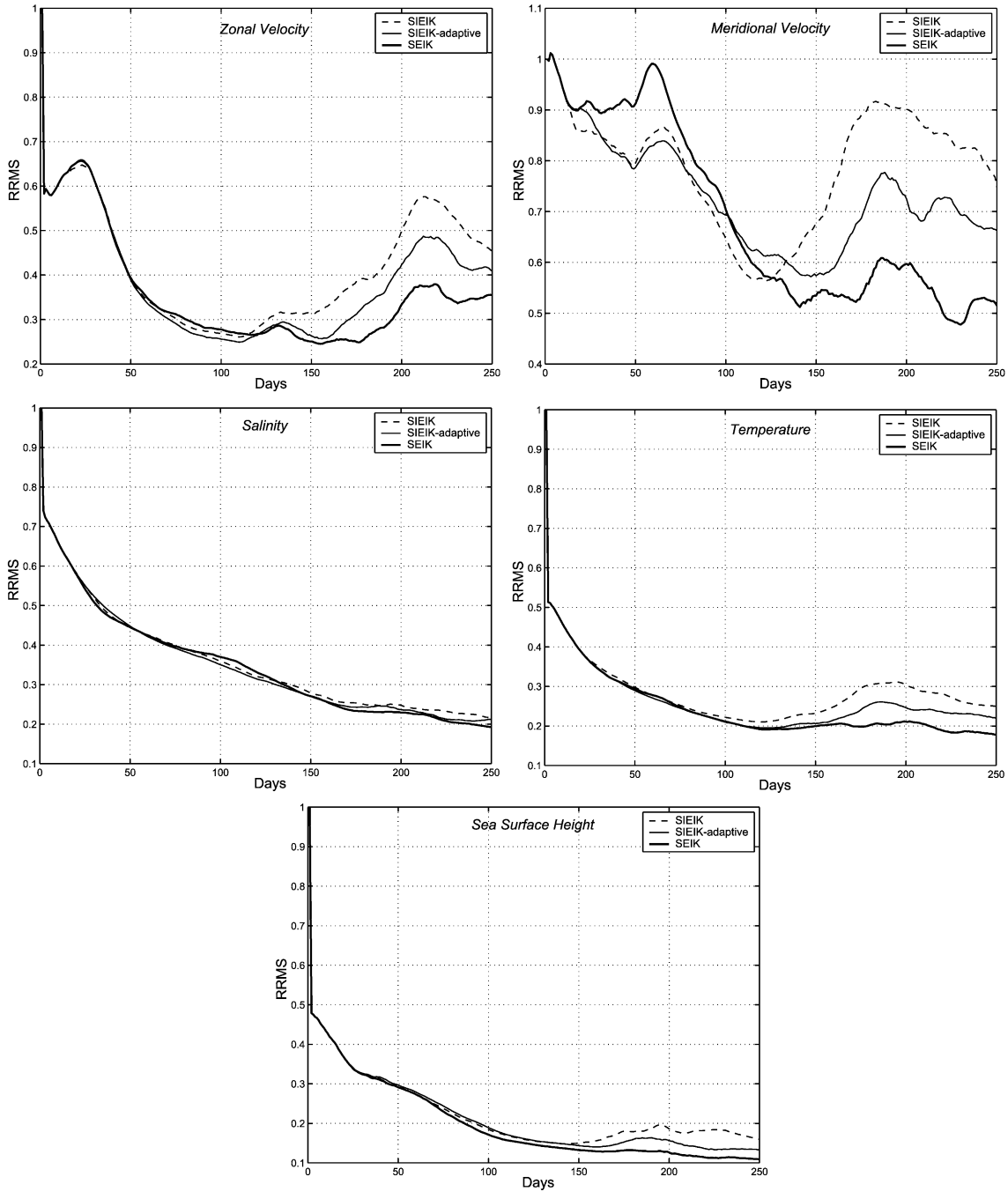

Fig. 11. Evolution in time of the RRMS for the SIEIK filter with and without adaptive tuning schemes (of the basis evolution and the forgetting factor) and the SEIK filter.

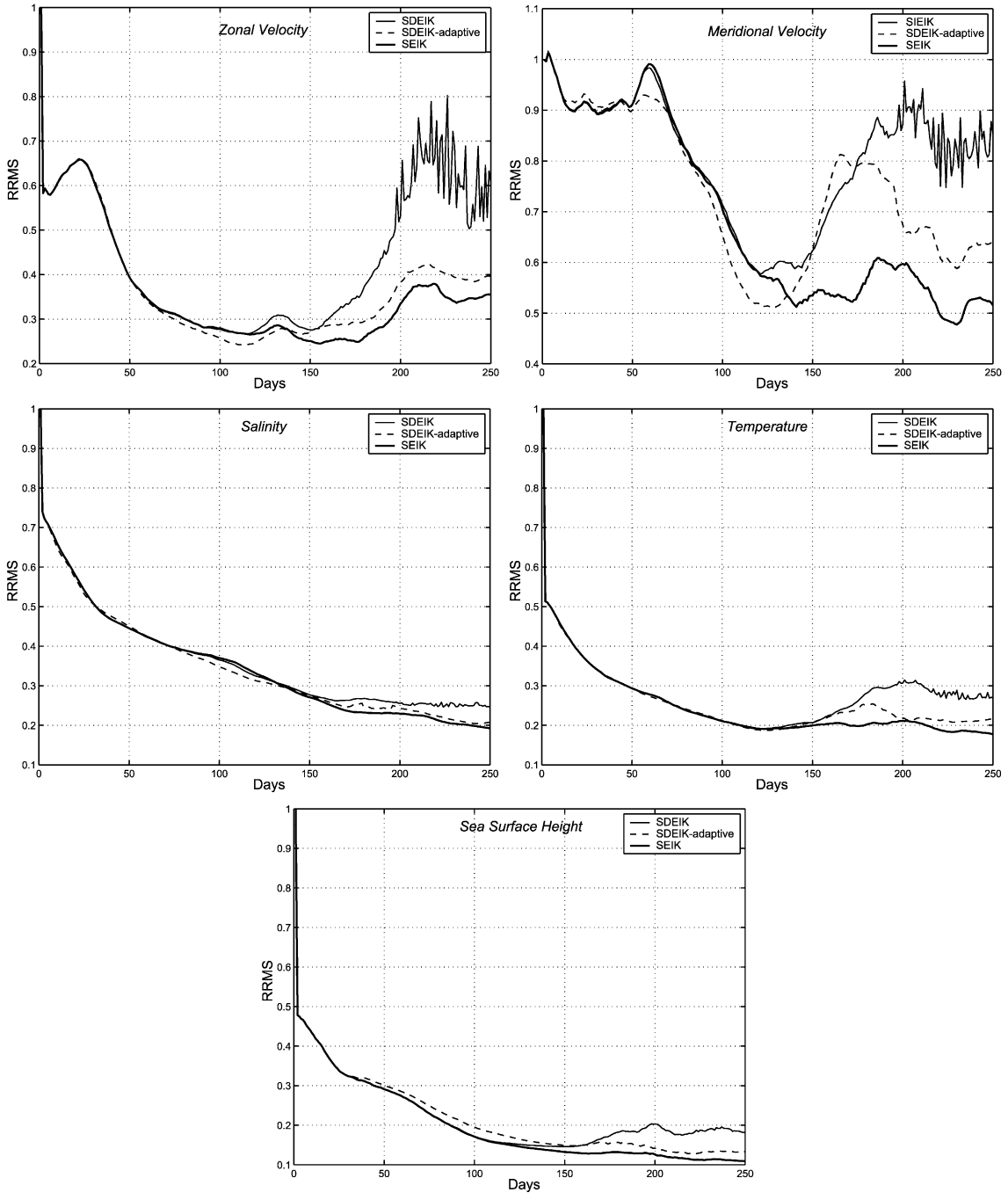

Fig. 12. Evolution in time of the RRMS for the SDEIK filter with and without adaptive tuning schemes (of the basis evolution and the forgetting factor) and the SEIK filter.

6. Conclusions

New data assimilation schemes that derived from SEEK and SEIK filters have been developed and discussed. The motivation was to reduce the cost of the SEEK and SEIK filters. Our approach was to simplify the way the evolution of the correction basis, which is the most expensive part of these filters. To deal with model instabilities, we have proposed two adaptive schemes. The first consists in letting the correction basis evolves as in the SEIK filter during the unstable periods. The second scheme is to consider a variable forgetting factor. Moreover, a method to detect periods of instability was suggested. It essentially consists of tracking the filter state by computing an instantaneous and a long-term prediction error norm. Finally, a series of twin experiments was conducted to assess the feasibility of the new filters and to evaluate their performances in comparison with SEIK. Our main conclusions are as follows:

(1) When model is stable, the degraded SEIK filters perform nearly as well as the SEIK filter, but can be 2–30 times faster. When model instabilities appear, the assumptions on which the degraded SEIK filters have been constructed are no longer justified and their performances degrade.

(2) Without any adaptive scheme, comparison between all degraded SEIK filters shows the superiority of the SIEIK filter. With a very low cost, the SSEIK filter is well behaved but only on one condition: the value of the forgetting factor is well tuned. Concerning the instability of the SDEIK filter, one can remedy to this by combining this filter with the SEIK filter. The resulting filter seems to be very effective in practice. The performance of the SFEK filter is relatively poor which is understandable, however, it can be used in a preliminary study to get some idea for the tuning of the parameters of the SEIK filter and its variants, because of its very low cost.

(3) By tracking prevision errors, one can obtain information about the filter state and then adapt the filter parameters to the present situation.

(4) Adaptive tuning of the forgetting factor considerably enhances the performance of all degraded SEIK filters. Moreover, its very low cost makes it particularly attractive.

(5) In most cases, and especially with the SDEIK filter, adaptive tuning of the evolution of the correction basis is useful; in few cases, it makes no significant improvement.

In twin experiments, our degraded SEIK filters was found to be fairly effective in assimilating of surface-only pseudo-altimeter data. Further works will consider more realistic situations, like the addition of the model error, or the use of more realistic observations (according to satellite tracks and real data from satellite). However, these preliminary twin experiment applications were necessary steps before realistic applications and provide us with encouraging as regard to that purpose.

Acknowledgements

This work was carried out within the framework of the IDOPT project, which is a joint project between INRIA, CNRS, University Joseph Fourier and INPG. The authors would like to thank Dr. Eric Blayo for interesting discussions during part of this work. Thanks are also given to Laurent Debreu and Laurent Parent for assistance with the numerical model.

Appendix A. Drawing a random orthogonal matrix

We present here an algorithm to generate a uniform orthogonal matrix Ω with r columns orthonormal in $\mathbb{R}^{r+1}$ and orthogonal to $(1,\ldots,1)^{\mathrm{T}}$. In order to do this, we will construct an iterative sequence of orthogonal matrix Ω_l, for l varies from 1 to r, as follows

A.1. Initialization

Take $\Omega_1 = w_{1,1}$ where $w_{1,1}$ is a random variable taking value 1 or -1 with a probability 1/2.

A.2. Iteration

For $l = 1,\ldots,r$, we compute Ω_l from Ω_{l-1} with the formula

$$\Omega_l = \begin{pmatrix} & & w_{l,1} \\ H(w_l)\Omega_{l-1} & & \vdots \\ & & w_{l,l} \end{pmatrix}, \tag{49}$$

where $w_l=(w_{l,1},\ldots,w_{l,l})^{\mathrm{T}}$ is a random vector distributed uniformly on the unit sphere of $\mathbb{R}^l$ and $H(w_l)$ is the Householder matrix associated to the vector w_l, namely

$$H(w_l)=\begin{pmatrix}1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 1\\ 0 & \cdots & 0\end{pmatrix}-\frac{1}{|w_{l,l}|+1}\times\begin{pmatrix}w_{l,1}\\ \vdots\\ w_{l,l-1}\\ w_{l,l}+\operatorname{sign}(w_{l,l})\end{pmatrix}(w_{l,l},\cdots,w_{l,l-1}).$$

Note that the columns of $H(w_l)$ are orthonormal and orthogonal with the vector w_l. One can then show that the matrices Ω_l, $l=1,\ldots,r$ are random orthogonal matrices distributed uniformly. Finally, to obtain Ω, just multiply Ω_r with the Householder matrix associated to the normed vector $\frac{1}{\sqrt{r+1}}(1,\ldots,1)^{\mathrm{T}}$.

References

Arakawa, A., 1972. Design of the UCLA general circulation model. Numerical integration of weather and climate, Rep. 7. Dept. of Meteorology, University of California, 116 pp.

Astrom, K.J., Wittenmark, B., 1995. Adaptive Control. Addison Wesley, New York, NY, 580 pp.

Blanke, B., Delecluse, P., 1993. Variability of the tropical Atlantic ocean simulated by a general circulation model with two different mixed layer physics. J. Phys. Oceanogr. 23, 1363–1388.

Brasseur, P., Ballabrera-Poy, J., Verron, J., 1999. Assimilation of altimetric observations in a primitive equation model of the gulf stream using a singular evolutive extended Kalman filter. J. Mar. Sys. 22 (4), 269–294.

Burgers, G., Van Leeuwen, P.J., Evensen, G., 1998. Analysis scheme in the ensemble Kalman filter. Mon. Weather Rev. 126, 1719–1724.

Cane, M.A., Kaplan, A., Miller, R.N., Tang, B., Hackert, E.C., Busalacchi, A.J., 1995. Mapping tropical Pacific sea level: data assimilation via a reduced state Kalman filter. J. Geophys. Res. 101 (C10), 599–617.

Cohn, S.E., Tolding, R., 1996. Approximate Kalman filters for stable and unstable dynamics. J. Meteorol. Soc. Jpn. 74, 63–75.

Dee, D.P., 1990. Simplification of the Kalman filter for meteorological data assimilation. Q. J. R. Meteorol. Soc. 117, 365–384.

Dee, D.P., da Silva, A.M., 1999. Maximum-likelihood estimation of forecast and observation error covariance parameters. Part I: Methodology. Mon. Weather Rev. 127, 1822–1834.

Evensen, G., 1992. Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model. J. Geophys. Res. 97 (C11), 17905–17924.

Evensen, G., 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. J. Geophys. Res. 99 (C5), 10143–10162.

Farrell, B.F., 1989. Optimal excitation of baroclinic waves. J. Atmos. Sci. 46, 1193–1206.

Fukumori, I., Malanotte-Rizzoli, P., 1995. An approximate Kalman filter for ocean data assimilation: an example with an idealized gulf stream model. J. Geophys. Res. 100 (C4), 6777–6793.

Gauthier, P., Courtier, P., Moll, P., 1993. Assimilation of simulated wind lidar data with a Kalman filter. Mon. Weather Rev. 121, 1803–1820.

Ghil, M., Malanotte-Rizzoli, P., 1991. Data assimilation in meteorology and oceanography. Adv. Geophys. 33, 141–266.

Hoang, H.S., De Mey, P., Tallagrand, O., Baraille, R., 1997. Adaptive filtering: Application to satellite data assimilation in oceanography. J. Dyn. Atmos. Oceans 27, 257–281.

Houtekamer, P.L., Mitchell, H.L., 1998. Data assimilation using an ensemble Kalman filter technique. Mon. Weather Rev. 126, 796–811.

Ide, K., Bennett, A.F., Courtier, P., Ghil, M., Lorenc, A.C., 1997. Unified notation for data assimilation: operational, sequential and variational. J. Meteorol. Soc. Jpn. 75 (1B), 181–189.

Levitus, S., 1982. Climatological atlas of the world ocean. NOAA Prof. Paper No. 13. US Government Printing Office, Washington, DC, 173 pp.

Madec, G., Delecluse, P., Imbard, M., Levy, C., 1997. Ocean General Circulation Model Reference Manual. Technical report, University Pierre and Marie Curie, Paris VI.

Mitchell, H.L., Houtekamer, P.L., 2000. An adaptive ensemble Kalman filter. Mon. Weather Rev. 128, 416–433.

Moore, A.M., Farrell, B.F., 1994. Using adjoint models for stability and predictability analysis. Data assimilation: tools for modeling the ocean in a global change perspective. In: Brasseur, P.P., Nihoul, J.C.J. (Eds.), NATO ASI Series, vol. 19, 253 pp.

Pham, D.T., 2001. Stochastic methods for sequential data assimilation in strongly nonlinear systems. Mon. Weather Rev. 129, 1194–1207.

Pham, D.T., Verron, J., Roubaud, M.C., 1997. Singular evolutive Kalman filter with EOF initialization for data assimilation in oceanography. J. Mar. Syst. 16, 323–340.

Pham, D.T., Verron, J., Gourdeau, L., 1998. Singular evolutive Kalman filters for data assimilation in oceanography. C. R. Acad. Sci. Paris 326, 255–260.

Sorenson, H.W., Sacks, J.E., 1971. Recursive fading memory filtering. Inform. Sci. 3, 101–119.

Trefethen, L.N., Trefethen, A.E., Reddy, S.C., Driscoll, T.A., 1993. Hydrodynamic stability without eigenvalues. Science 261, 571–584.

Verron, J., Gourdeau, L., Pham, D.T., Murtugudde, R., Busalacchi, A.J., 1998. An extended Kalman filter to assimilate satellite altimeter data into a nonlinear numerical model of the tropical pacific: method and validation. J. Geophys. Res. 104 (C3), 5441–5458.