Some Variants of the Singular Evolutive Extended Kalman (SEEK) Filter

*Dinh Tuan Pham** and *Ibrahim Hoteit***

*Laboratory of Modeling and Computation, IMAG, C.N.R.S.
Univ. of Grenoble, B.P. 53X, 38041 Grenoble Cedex, France
e-mail: Dinh-Tuan.Pham@imag.fr
**PORD, Scripps Institution of Oceanography
La Jolla, California
email: ihoteit@fjord.ucsd.edu

Some references

Hoteit, I. and Pham, D. T and Blum, J. (2002). A Simplified Kalman Filtering and Application to Altimetric Data Assimilation in the Tropical Pacific. *J. Marine Systems*, To appear

Hoteit, I., Pham, D. T. and Blum, J. (2001) A Semi-Evolutive Partially Local Filter for Data Assimilation. *Marine Pollution Bulletin*, **43**, 164–174.

Pham, D. T. (2001) Stochastic methods for sequential data assimilation in strongly nonlinear systems, *Monthly Weather Review*. **129**, 5, 1194–1207.

Pham, D. T., Verron, J. and Gourdeau, L. (1998) Filtres de Kalman Singuliers évolutif pour l'Assimilation de Données en Océanographie. *Comptes Rendus Aca. Sci.* Science de la terre et des planètres, **326**, 255–260

Pham, D. T., Verron, J. and Roubaud, M. C. (1998) A Singular Evolutive Extended Kalman Filter for Data Assimilation in Oceanography. *J. Marine Systems*, **16**, 3, 4, 323–340.

Aim

1. Low cost: implementable in real operational setting (number state variables of the order $10^5 - 10^7$)

2. Robust with respect to model nonlinearity and system instability.

3. Reasonably efficient in terms of performance (with respect to what can be achievable)

We shall focus on the sequential approach, based on the Kalman filter.

Ideally, a *nonlinear filter* should be used (goal 2 and 3), but is is *incompatible* with goal 1.

Our approach is to modify the Kalman filter to achieve goals $1 - 3$.

Notations and Models

$$
\begin{aligned}
\mathbf{x}^t(t_i) &= M(t_{i-1}, t_i)\mathbf{x}^t(t_{i-1}) + \eta_i, \\
\mathbf{y}_i^o &= H_i \mathbf{x}^t(t_i) + \epsilon_i
\end{aligned}
$$

$\mathbf{x}^t(t)$ = true state vector of the physical system at time t,
$M(s,t)$ = model transition operator, $\Rightarrow \mathbf{M}(s,t)$ = its gradient,
η_i = dynamic noise (or model error), covariance matrix $\mathbf{Q}_i$,
$\mathbf{y}_i^o$ = observation,
H_i = observation operator, $\Rightarrow \mathbf{H}_i$ = its gradient,
ϵ_i = observation noise, covariance matrix $\mathbf{R}_i$.

The η_i and ϵ_i are centered and independent. Further

$\mathbf{x}^f(t_i)$ = forecast state vector at t_i based on observations up to t_{i-1}
$\mathbf{P}_i^f$ = forecast error covariance matrix,
$\mathbf{x}^a(t_i)$ = analysis state vector at t_i based on observations up to t_i,
$\mathbf{P}_i^a$ = analysis error covariance matrix.

The Singular Evolutive Extended Kalman (SEEK) filter

It is really an extended Kalman filter based on a *singular low rank error covariance matrix*.

Instead of reducing the system state vector, we only reduce the rank of the filter error matrix. This way, *the dynamic of the system is respected, only the error propagation is approximated*.

Let the error covariance matrix $\mathbf{P}_i^a$ be factorized as $\mathbf{L}_i\mathbf{U}_i\mathbf{L}_i$, the filter equations are

$$\mathbf{L}_i = \mathbf{M}(t_{i-1}, t_i)\mathbf{L}_{i-1}$$

$$\mathbf{U}_i^{-1} = \{\mathbf{U}_{i-1} + (\mathbf{L}_i^T\mathbf{L}_i)^{-1}\mathbf{L}_i^T\mathbf{Q}_i\mathbf{L}_i(\mathbf{L}_i^T\mathbf{L}_i)^{-1}\}^{-1} + \mathbf{H}_i^T\mathbf{L}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i\mathbf{L}_i$$

$$\mathbf{x}^f(t_i) = M(t_{i-1}, t_i)\mathbf{x}^a(t_{i-1}) \quad \text{forecast}$$

$$\mathbf{x}^a(t_i) = \mathbf{x}^f(t_i) + \mathbf{K}_i[\mathbf{y}_i^o - H_i\mathbf{x}^f(t_i)] \quad \text{Correction}$$

$$\mathbf{K}_i = \mathbf{L}_i\mathbf{U}_i\mathbf{L}_i^T\mathbf{H}_i^T\mathbf{R}_i^{-1} \quad \text{gain}$$

Discussion

The filter makes corrections only along a direction parallel to the linear space spanned by the columns of $\mathbf{L}_i$. We call it the correction space and $\mathbf{L}_i$ the correction basis.

To keep the error covariance matrix having the same low rank we have projected orthogonally the dynamic error vector orthogonally onto the correction space. But there might be better way.

Apart from the intrinsic error due to the observation and dynamic error, this filter introduce two kinds of (unaccounted for) error

1. The linearization error.

2. The truncation error: it comes from the fact that the part of error which lies outside the correction space will be corrected and the part of the dynamic error orthogonal to this space is ignored.

Over time these errors can accumulate and produce filter instability

The effect of the truncation error can be contained due to the evolutionary nature of the correction basis: $\mathbf{L}_i$ *evolves in time*.

It can be proved that under the ideal case of a linear system, the SEEK filter is stable (i.e. the error is contained) provided that the correction basis has rank *at least the number of unstable modes* of the system dynamic.

However, nothing can be said in the case of a (strongly) nonlinear system. In this respect a simple "trick" to stabilized the filter is the use of a forgetting factor

$$\mathbf{U}_i^{-1} = \rho \mathbf{U}_{i-1}^{-1} + \mathbf{L}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \mathbf{L}_{i-1}, \qquad 0 < \rho < 1$$

It amounts to inflating the analysis error covariance matrix $\mathbf{P}_{i-1}^a$ by a factor $1/\rho$. This error covariance matrix is in fact *underestimated by the filter equations*.

The Singular Evolutive Interpolated Kalman (SEIK) filter

Feature: Avoid calculation of gradient and introduce some degree of randomization.

Instead of factorizing $\mathbf{P}^a_{i-1}$ as $\mathbf{L}_{i-1}\mathbf{U}_{i-1}\mathbf{L}_{i_1}$, we represent it through $r+1$ "interpolating" states $\mathbf{x}^i_1(t_{i-1}), \ldots, \mathbf{x}^i_{r+1}(t_{i-1})$:

$$\mathbf{P}^a_{i-1} = \frac{1}{r+1}\sum_{j=1}^{r+1}[\mathbf{x}^i_j(t_{i-1}) - \mathbf{x}^a(t_{i-1})][\mathbf{x}^i_j(t_{i-1}) - \mathbf{x}^a(t_{i-1})]^T$$

$$\mathbf{x}^a(t_{i-1}) = \frac{1}{r+1}\sum_{j=1}^{r+1}\mathbf{x}^i_j(t_{i-1})$$

This is equivalent to taking

$$\mathbf{L}_{i-1} = [\mathbf{x}^i_1(t_{i-1}) \;\cdots\; \mathbf{x}^i_{r+1}(t_{i-1})]\mathbf{T}, \qquad \mathbf{U}_{i-1} = [(r+1)\mathbf{T}^T\mathbf{T}]^{-1}$$

where $\mathbf{T}$ be a $(r+1) \times r$ matrix with zero row sum, for ex.

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} - \frac{1}{r+1} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

However, $\mathbf{L}_{i-1}$ would now be updated as

$$\mathbf{L}_i = [\mathbf{x}_1^i(t_i-) \ \cdots \ \mathbf{x}_{r+1}^i(t_i-)]\mathbf{T}, \qquad \mathbf{x}_j^i(t_i-) = M(t_{i-1}, t_i)\mathbf{x}_j^i(t_{i-1}),$$

U_i being updated as before. Then $\mathbf{P}_i^a = \mathbf{L}_i \mathrm{U}_i \mathbf{L}_i$ would be again factored similarly to obtained the new interpolating states $\mathbf{x}_1^i(t_i)$, $\ldots$, $\mathbf{x}_{r+1}^i(t_i)$. Note however that, in computing the new $\mathbf{x}^a(t_i)$,

- $\mathbf{x}^f(t_i)$ is computed as $(r+1)^{-1} \sum_{j=1}^{r+1} \mathbf{x}_j^i(t_i-)$ and not through the model equations.

- the gain matrix $\mathbf{K}_i$ is computed as $\mathbf{L}_i \mathrm{U}_i (\mathbf{HL})_i^T \mathbf{R}_i^{-1}$ where

$$(\mathbf{HL})_i = [H_i \mathbf{x}_1^f(t_i-) \quad \cdots \quad H_i \mathbf{x}_{r+1}^f(t_i-)].$$

Discussion

The novelty of this filter is that the updating of the correction basis is made not through the linear tangent model but through the interpolation of the model operator, based on $r+1$ interpolating states. The observation operator is also not linearized but interpolated.

Interpolation could yield better results than linearization for moderately nonlinear systems: the tangent linear model would be adequate only for small deviation (from the current estimate); for larger deviation (i.e. filter error) linear interpolation may be better.

Another point is that the representation of $\mathbf{P}_i^a$ in terms of the $\mathbf{x}_j^i(t_i)$ is not unique. We take advantage of this to draw them randomly according to the Gaussian distribution of mean $\mathbf{x}^a(t_i)$ and covariance matrix $\mathbf{P}_i^a$, with the constraint that *the sample means and covariance matrix of the drawn states* *match exactly* *the theoretical values*.

SEIK filter versus EnKF (Ensemble Kalman Filter)

Through the use of Monte-Carlo drawing, the SEIK filter is similar to the EnKF.

There is however an important difference: the draws must be such that their sample means and covariance matrix must match their theoretical values $\mathbf{x}^a(t_i)$ and $\mathbf{P}^a(t_i)$, computed through the *Kalman correction equations*

This can be achieved by a technique called second order exact sampling.

$\Rightarrow$ the SEIK coincides with the Kalman filter in the case of a linear (possibly time varying) system (with no dynamical error). Thus it *preserves the optimality* of the Kalman filter in this case.

It is also possible to draw more Monte-Carlo samples than necessary ($> r+1$), then reduce the rank of the covariance matrix to r through an eigenvector analysis. (c.f. Mon. Wea. Rev. **129**).

The singular semi-evolutive interpolated Kalman (SSEIK) filter

In our experiences, we find that the initial correction basis constructed through an EOF (empirical orthogonal function) analysis is often quite good: a large percentage of error can be corrected at once.

Further, by keeping this basis fix, the filter still performs acceptably (Brasseur *et Al.*, 1999), except that it deteriorates in the long run.

The above observations suggest that *although the correction basis need to evolve, it doesn't not need to evolves much*.

Our idea 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 the filter we start by representing $\mathbf{P}^a_{i-1}$ as

$$\mathbf{P}^a_{i-1} = \mathbf{L}_{i-1}(\mathbf{C}^{-1}_{k-1})^T \Theta \Theta^T \mathbf{C}^{-1}_{i-1} L^T_{i-1}$$

where $\mathbf{C}_{k-1}$ is the Cholesky decomposition of $\mathbf{U}^{-1}_{i-1}$ and Θ is an *arbitrary orthogonal matrix*. Thus there is a whole set of equivalent representations, depending on the choice of Θ:

$$\mathbf{P}^a_{i-1} = \tilde{\mathbf{L}}_{i-1}\tilde{\mathbf{L}}^T_{i-1}, \qquad \tilde{\mathbf{L}}_{i-1} = \mathbf{L}_{i-1}(\mathbf{C}^{-1}_{i-1})^T \Theta,$$

We take advantage of this degree of freedom to choose Θ such that the columns of $\tilde{\mathbf{L}}_{i-1}$ are orthogonal and ranked according to their norm in increasing order. Then

$$\mathbf{P}^a_{i-1} = \tilde{\mathbf{L}}^{r_0}_{i-1}\tilde{\mathbf{L}}^{r_0\,T}_{i-1} + \tilde{\mathbf{L}}^{r_1}_{i-1}\tilde{\mathbf{L}}^{r_1\,T}_{i-1}$$

where $\tilde{\mathbf{L}}^{r_0}_{i-1}$ and $\tilde{\mathbf{L}}^{r_1}_{i-1}$ contain the first r_0 and the last $r_1 = r - r_0$ columns of $\tilde{\mathbf{L}}_{i-1}$, respectively.

The matrix $\tilde{\mathbf{L}}^{r_0}_{i-1}$ contribute the least to the filter error representation. The idea is to keep it fix and let only the matrix $\tilde{\mathbf{L}}^{r_1}_{i-1}$ evolve.

Specifically, we draw r_1+1 interpolating states $\mathbf{x}_1^i(t_{i-1}), \ldots, \mathbf{x}_{r_1+1}^i(t_{i-1})$ with sample mean $\mathbf{x}^a(t_{i-1})$ and covariance matrix $\tilde{\mathbf{L}}_{i-1}^{r_1}\tilde{\mathbf{L}}_{i-1}^{r_1 T}$ and let

$$\check{\mathbf{L}}_{i-1} = [\tilde{\mathbf{L}}_{i-1}^{r_0}, [\mathbf{x}_1^i(t_{i-1}) \quad \cdots \quad \mathbf{x}_{r_1+1}^i(t_{i-1})]\mathbf{T}]$$

Then the analysis error covariance matrix can be represented as

$$\mathbf{P}_{i-1}^a = \check{\mathbf{L}}_{i-1} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & (r_1+1)\mathbf{T}^T\mathbf{T} \end{bmatrix}^{-1} \check{\mathbf{L}}_{k-1}^T.$$

The filter can then be constructed similarly to the SEIK filter. One applies $M(t_{i-1}, t_i)$ on the $\mathbf{x}_j^i(t_{i-1})$ to obtain $\mathbf{x}_j^i(t_i-)$. The last states will be used to obtain the forecast $\mathbf{x}^f(t_i)$ (as their average) and

$$\mathbf{L}_i = [\tilde{\mathbf{L}}_{i-1}^{r_0}, [\mathbf{x}_1^i(t_i-) \quad \cdots \quad \mathbf{x}_{r_1+1}^i(t_i-)]\mathbf{T}]$$

The updating of $\mathbf{U}_i$ and the correction step is similar to that of the SEIK filter. Then $\mathbf{L}_i$ is transformed in to $\tilde{\mathbf{L}}_i$ and the new interpolating states $\mathbf{x}_1^i(t_i), \ldots, \mathbf{x}_{r_1+1}^i(t_i)$ are drawn.

The Semi-Evolutive Partially Local Kalman (SEPLK) Filter

This filter combines the feature of the previous semi-evolutive Kalman filter and the use of an additional *local* correction basis.

Local correction basis

The basis which have been in used in our filter are of global nature.

This means that a observation at the spatial point (x, y, z) would induces a correction of the system state at any other spatial point (x', y', z') no matter how far is this point from (x, y, z).

Intuitively this correction should be negligible when the point (x', y', z') is too far from (x, y, z) and it makes sense not to make correction at all at such a point.

This idea can be implemented through the use of a local correction basis. The basis functions of such a basis, by definition, vanish outside a small sub-domain (which is its support).

In general the use of a local basis is less efficient in the sense that a higher number of basis functions is necessary (as their support must at least cover the whole physical domain).

But one can afford a much larger number of basis functions, since calculations on them (scalar products, multiplication by a matrix, ...) need to be done only on the small region where they do not vanish, and hence are vastly less costly.

Another advantage of using a local basis is its flexibility: there is a wide range of options to choose: (i) the number of sub-domains, (ii) their size and shape, and (iii) the number of the local basis elements in each sub-domains.

Over all, a local basis can provide better representativeness with less cost.

Construction of a local correction basis

For this purpose we make use of the well known EOF analysis.

But instead of performing a single EOF analysis, the idea is to define a set of sub-domains on each of which a separate EOF analysis will be applied.

Specifically, we consider a *partition of unity*, which is a set of positive functions $\{\pi^{(j)}, j = 1, \ldots, J\}$ (of the spatial coordinates x, y, z), whose sum equals 1 identically. Typically, $\pi^{(j)}$ has support a small sub-domain.

A state vector $\mathbf{x} = X(x, y, z)$ can thus be written as

$$X(x, y, z) = \sum_{j=1}^{J} X(x, y, z)\pi^{(j)}(x, y, z) = \sum_{j=1}^{J} X^{(j)}(x, y, z)$$

We then carries out separately an EOF analysis on each local field $X^{(j)}$, $1 \leq j \leq J$, to obtain a set of basis functions. The collection of all such functions constitute our local basis.

Semi-evolutivity

The main difficulty with the local EOF basis is that we can not make it to evolve (with the model) without losing its locality.

To overcome this difficulty, we introduce a *mixed basis, consisting a a global and a local part and allow the global part to evolve*.

Another reason to consider mixed basis is that *purely local basis does not well represent "long range" phenomena*. Adding some global basis functions can thus greatly enhance representativeness of the basis.

Practically, *the mixed basis is constructed by performing first a global EOF analysis, the performing local EOF analysis (as above) on the residuals of the first analysis.*

The SEPLK filter operates quite similarly to the SSEIK filter, with the only constraint that the fix part of the correction basis contains all local basis functions.

Simulation

Model:

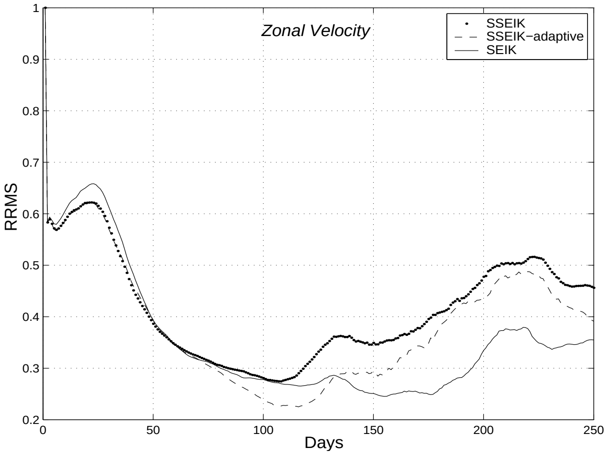
OPA (Océan Parallésisé) for the tropical Pacific. It is a primitive equation model developed at the LODYC laboratory.

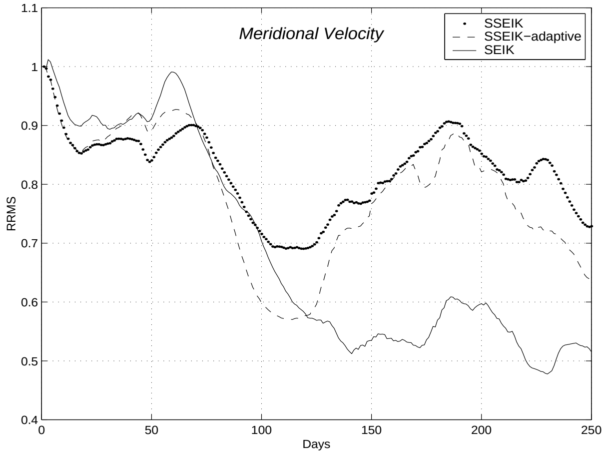
Configuration:

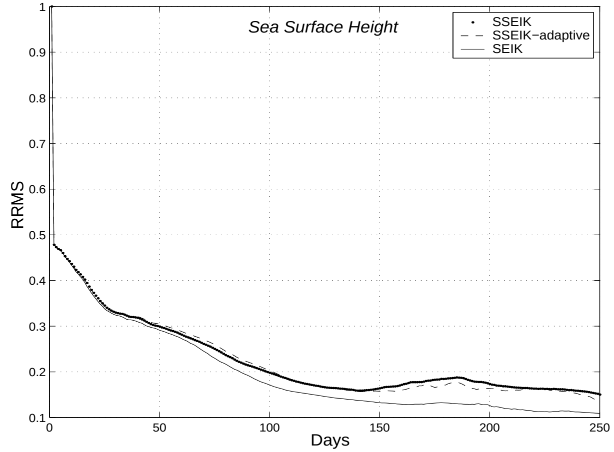
The domain covers the entire tropical Pacific basin extending from 120°E to 70°W and from 33°S to 33°N. The number of grid points is 171×59 on 25 vertical levels.

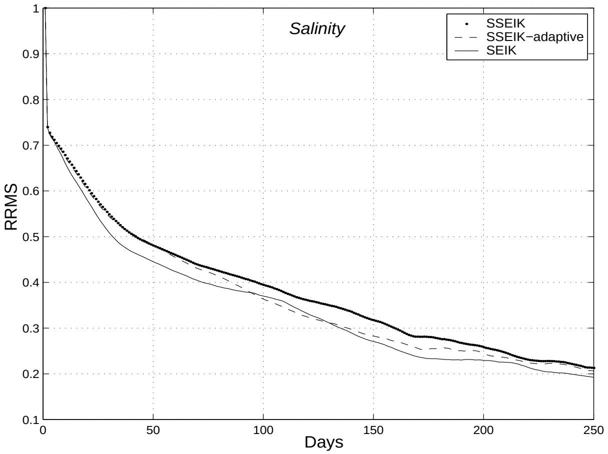
The resolution is: zonal 1°, meridional: 0.5° (equator) $\rightarrow$ 2° (North and South boundaries); vertical: 10m (sea surface – 120m) $\rightarrow$ 1000m (sea bottom). The time step is one hour.

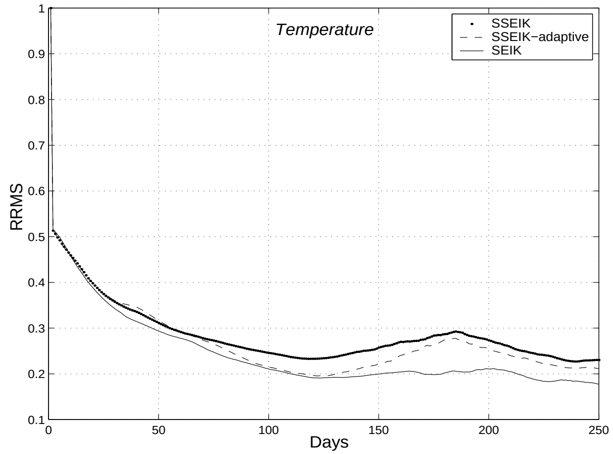
The bathymetry was obtained from Levitus data's mask and the forcing fields are interpolated from the ECMWF (European center for medium-range weather forecasts). The salinity and the temperature are stem from seasonal climatologic Levitus data.



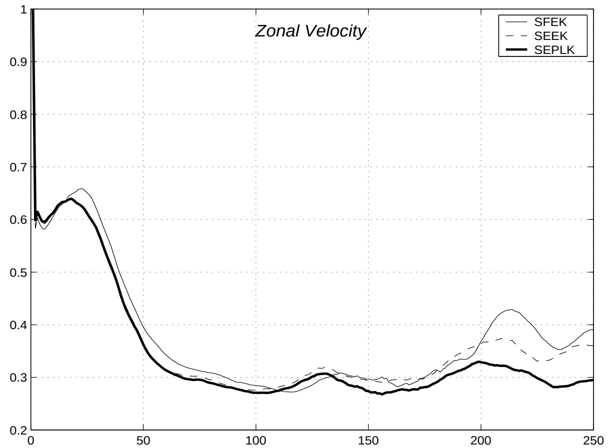


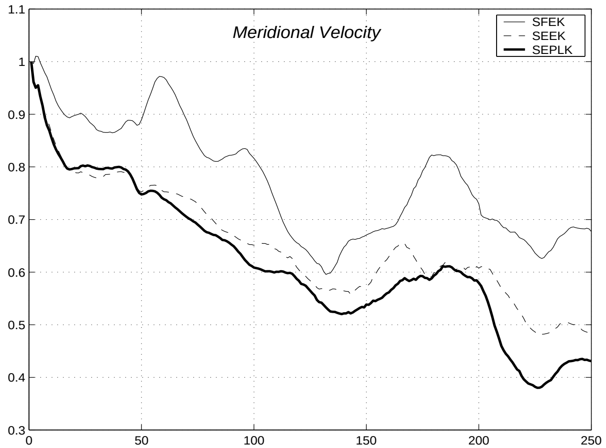


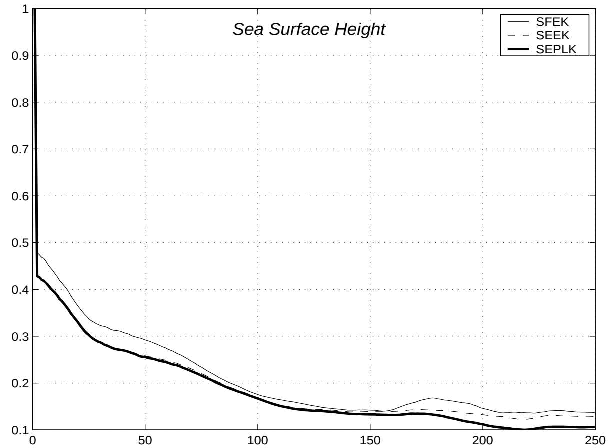


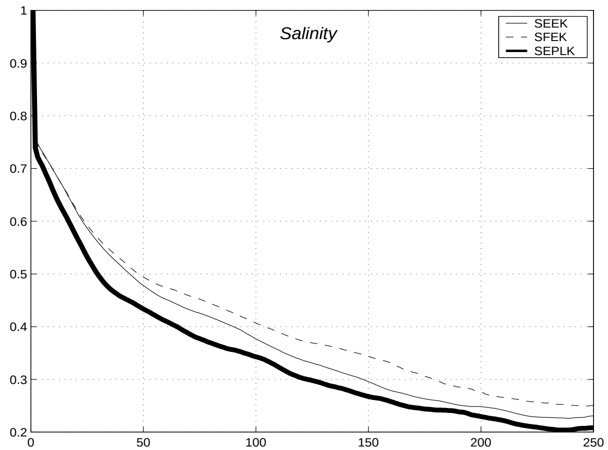


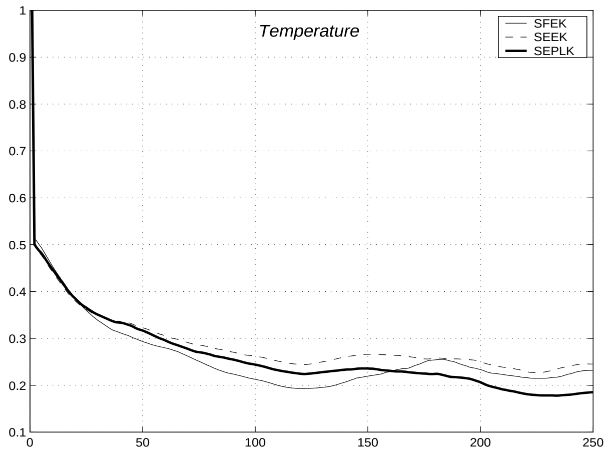
Evolution in time of the $RRMS$ for the SSEIK filter with and without adaptation of the forgetting factor and the SEIK filter.











Evolution in time of the $RRMS$ for the SFEK (with the mixed EOF basis), SEPLK and SEEK filters (forgetting factor $0,8$).