An adaptive filter based innovation approach for state estimation in high-dimensional systems

HONG SON HOANG
SHOM
HOM/REC
42 Av Gaspard Corriolis
FRANCE
hhoang@shom.fr

REMY BARAILLE
SHOM
HOM/REC
42 Av Gaspard Corriolis
FRANCE
remy.baraille@shom.fr

Abstract: In this paper an adaptive filter (AF) based on innovation approach is described and its efficiency is compared with other estimation methods. Such AF is developed in the context of data assimilation problem in high-dimensional dynamical systems. The emphasis is put on the importance of innovation approach which is a basis for construction of the AF as well as the choice of a set of tuning parameters in the filter gain. It will be shown that the innovation representation for the initial dynamical system plays an essential role in providing stability of assimilation algorithms for stable and unstable system dynamics and allows to apply the economical and efficient optimization method known as stochastic simultaneous perturbation (SSP) algorithm. Numerical experiments will be given to illustrate the performance of the AF.

Key–Words: Dynamical system, innovation process, filter stability, minimum mean square prediction error, simultaneous stochastic perturbation

1 Introduction

Data assimilation [5] is a procedure which is aimed at estimating a best possible physical (ocean, atmospheric ...) state by combining a model forecast with observations available. The research on data assimilation methods has been increasing quickly during the last decade. This happens due to the exponential increase of computer power and memory. This progress allows to develop more and more sophisticated numerical models to simulate real physical processes. The latter, in its turn, results in gigantic growth of dimension of the numerical model state. At the present it is still out of question of possible application of optimal estimation methods to the design of optimal data assimilation schemes.

Mathematically, the data assimilation problem can be stated as the following : Let

$$x_{k+1} = \Phi x_k + w_k, \tag{1}$$

and we are given the observations

$$z_{k+1} = H x_{k+1} + v_{k+1}, k = 0, 1, 2, ... \tag{2}$$

here $x_k \in R^n$ is the system state at k instant, $\Phi \in R^{n\times n}$, $z_k \in R^p$ is observation vector, $H_k \in R^{p\times n}$, w_k, v_k are the model and observation uncorrelated noise sequences which are mutually uncorrelated and uncorrelated with x_0.

It is well known that the optimal in mean square error (MSE) estimate $\hat{x}_k$ based on the set of observations $Z[1,k] := \{z_1, ..., z_k\}$ is a filtered estimate and for the linear dynamical system (1)(2), its computation can be efficiently performed using the Kalman filter (KF) [11] which is a sequential procedure.

For the today ocean (or meteorological) numerical models, the system state x_k has the dimension lying in the range $[10^6 : 10^7]$ and there is uncertainty in statistics of the initial state, model and observational noises. Due to very large n, it is impossible to apply traditional optimal procedures to estimate the system state and for that reason there exist different approximation algorithms for solving this estimation problem. At the present and in the near future, the computer capacity, in both computational power and memory, is largely insufficient to implement the KF in real time to produce the filtered estimate and to make corresponding forecasts.

In this paper, the emphasis is put on the underlying principles in the construction of an AF, especially on the innovation approach as a basis for representing the initial dynamical system in a new innovation form for the input-output system (section 2). This innovation representation allows to consider the optimal AF as such which minimizes the prediction error (PE) for the system output and the choice of control variables from some pertinent parameters in the filter gain. To

see more the advantages of the AF, the classical variational method (VM) [16] will be briefly presented (section 3). A simple method based on stochastic simultaneous perturbation (SSP) for estimating a high-dimensional matrix and decomposing it into the product of two matrices in the class of matrices of the given rank, is presented in section 4 and section 5 (for numerical experiment). The high performance of the AF will be compared with different assimilation methods like the extended KF (EKF) [14] in the experiment with the Lorenz system (section 5). The assimilation experiment with the ocean model MICOM using sea surface height (SSH) observations to estimate the circulation in the North Atlantic configuration, is presented in section 6. Here the AF and three other sequential filters will be implemented to show the efficiency of the AF. The conclusions are given in Section 6.

2 Adaptive filtering (AF)

2.1 Innovation approach

The main underlying principle in the construction of the AF concerns the choice of the innovation representation for the original input-output system (1)(2) to formulate the optimization problem. It is well known that under standard conditions, the optimal in MSE estimate $\hat{x}_k$ can be obtained by the KF. As the innovation process for the system output in the KF forms a white sequence, Kailath [12] has developed an innovation approach, in an elegant way, to derive the optimal filter for more general linear systems like non-stationary, filtering problems with Markovian processes for the model and observation errors ... The innovation approach to linear least-squares approximation problems is first to "whiten" the output data and then to treat the resulting simpler white-noise observation problem. Consider the observation (output) sequence z_k. The innovation process, associated with z_k, is written as $\zeta_k = z_k - E[z_k|z_k^1]$ where $E[z_k|z_k^1]$ is conditional expectation, and under standard conditions (gaussianness, uncorrelated noise sequences ...), we have $E[z_k|z_{k-1}^1] = H\hat{x}_{k/k-1}$ hence

$$\zeta_k = z_k - H\hat{x}_{k/k-1}, \hat{x}_{k/k-1} = \Phi\hat{x}_{k-1}, \tag{3}$$

where $\hat{x}_{k/k-1}$ is an optimal in MSE one-step ahead prediction for x_k given z_{k-1}^1. Using ζ_k instead of z_k, under standard conditions, one can write out the formula for the estimate $\hat{x}_k$ and the KF

$$\begin{aligned} \hat{x}_k &= \Phi\hat{x}_{k-1} + K_k\zeta_k, \\ K_k &= M_k H^T[HM_kH^T + R_k]^{-1} \end{aligned} \tag{4}$$

where $M(k)$ is the ECM for the prediction $\hat{x}_{k/k-1}$. This matrix can be found recursively as a solution to the Algebraic Riccati equation (ARE) [11].

Due to the very expensive computational burden in time stepping the ECM as well as insufficient memory storage, the KF is impractical for solving data assimilation problems in very high-dimensional setting. The idea of the AF is based on the fact that when the filter is optimal, the innovation ζ_k has a minimum variance. By assuming that the filter belongs to the class of parameterized gains, i.e.

$$K_k = K_k(\theta), \theta \in \Theta, \tag{5}$$

where θ is a vector of pertinent parameters (control vector), the optimal AF can be considered as that which minimizes the prediction error for the system output

$$\begin{aligned} J(\theta) = E[\Psi(\theta)] \rightarrow \min_{\theta\in\Theta}, \Psi(\zeta_k) = ||\zeta_k||^2, \\ ||\zeta_k||^2 :=< \zeta_k, \zeta_k > . \end{aligned} \tag{6}$$

In [10], the different classes of parameterized filters are found which belong to the class of stable reduced-order filters (ROF) [1], [10].

As an example for one class of ROFs, consider

$$K_k = P_r K_{e,k} \tag{7}$$

where $K_{e,k} : R^p \rightarrow R^{n_e}$ represents the gain, mapping the innovation vector from the observational space to the reduced space R^{n_e} of dimension $n_e \leq n$; P_r is mapping the reduced space R^{n_e} to the full space R^n. The choice of a reduced space is of primary importance since from it depends the main characteristics of the filter known as stability. As proved in [10], under detectability condition, stability of the filter is ensured by forming the columns of P_r from unstable and stable eigenvectors (or singular vectors, Schur vectors) of the fundamental matrix Φ and one can choose

$$K_e = H_e^T[H_eH_e^T(k) + I\alpha]^{-1}, H_e := HP_r, \alpha > 0. \tag{8}$$

One class of parameterized filters is (section 5.2.2, [10])

$$\begin{aligned} K(\theta) &= P_r\Lambda K_e(\theta), \\ \Lambda &= \text{diag}[\theta_1, ..., \theta_{n_e}], \\ 1 - 1/|\phi_i| < \epsilon_1(i) \leq \theta_i \leq \epsilon_2(i) < 1 + 1/|\phi_i|. \end{aligned} \tag{9}$$

if ϕ_i is an unstable eigenvalue of Φ. For the neutral ϕ_i, we have

$$0 < \epsilon_1(i) \leq \theta_i \leq \epsilon_2(i) < 2, \tag{10}$$

2.2 Implementation of AF

2.2.1 The choice of criteria (6)

The choice of (6) is important in many aspects in order to obtain a simple and efficient data assimilation algorithm. The idea lying behind (6) is to select the vector θ of some pertinent parameters as control variables for minimizing the mean of the cost function $\Psi(\zeta_k)$.

The solution to the problem (6) can be found iteratively using a stochastic optimization (SA) algorithm

$$\theta_{k+1} = \theta_k - a_k \nabla_\theta \Psi(\zeta_{k+1}) \tag{11}$$

where $\{a_k\}$ is a sequence of positive scalars satisfying some conditions to guarantee a convergence of the estimation procedure. The standard conditions are

$$a_k > 0, a_k \to 0, \sum_{k=1}^{\infty} a_k = \infty, \sum_{k=1}^{\infty} a_k^2 < \infty \tag{12}$$

The algorithm (11) is very simple : it requires, at the k^{th} assimilation instant, to compute only the gradient of the sample cost function $\Psi(\zeta_k)$. This gradient can be computed using the adjoint equation (AE) approach

$$[\delta\Psi(\zeta_{k+1})]_{\theta_k} = - < H\Phi P_r \delta\Lambda K_e \zeta_k, \zeta_{k+1} >= \\ < \delta\Lambda K_e \zeta_k, \zeta'_{k+1} >, \zeta'_{k+1} := -P_r^T \Phi^T H^T \zeta_{k+1} \tag{13}$$

Thus, minimization of (6) by gradient-based SA algorithm requires only one integration of the direct model Φ and one backward integration of the AE Φ^T. For the structure of the gain (9), the objective function Ψ is quadratic with respect to (wrt) θ hence one can find easily the optimal parameters.

A less computational burden can be achieved by measuring the sample objective function (but not based on a gradient formula): one approximates the gradient using the values of the cost function (on the basis of finite difference scheme (FDSA)). Traditionally, the i^{th} component of the gradient can be approximated by

$$\nabla_{\theta_i} \Psi(\theta_k) = g_i, \\ g_i = [\Psi(\theta_k + c_k e_i) - \Psi(\theta_k - c_k e_i)]/(2c_k) \tag{14}$$

where e_i is the unit vector with 1 in the i^{th} component, 0 otherwise. It is seen that FDSA algorithms do not require the formula for the gradient. However, for the high-dimensional systems, this algorithm is inapplicable due to component-wise derivative approximation: for approximation of each partial derivative, we need to make two integrations of the direct model.

Let us look at the class of simultaneous perturbation SA (SPSA) [15], [9]. The algorithm SPSA is of the same structure as that of FDSA (14), with the difference residing in the way to perturb stochastically and simultaneously all the components of θ. Concretely, let $\Delta_k = (\Delta_{k,1}, ..., \Delta_{k,n})^T$ be a random vector, $\Delta_{k,i}, i = 1, ..., n$ are Bernoulli independent identically distributed (iid). The gradient of the sample objective function is estimated as

$$\nabla_\theta \Psi(\theta_k) = g = (g_1, ..., g_n)^T, \\ g = [\Psi(\theta_k + c_k\Delta_k) - \Psi(\theta_k - c_k\Delta_k]\Delta_k^{-1}/(2c_k), \\ \Delta_k = (\Delta_{k,1}, ..., \Delta_{k,n})^T, \\ \Delta_k^{-1} := (1/\Delta_{k,1}, ..., 1/\Delta_{k,n})^T. \tag{15}$$

One sees here that all the directions in the parameter space are perturbed at the same time (the numerator is identical in all n components). Thus, SPSA uses only two (or three) times the direct integration of the model, independently on the dimension of θ which makes it possible to apply to high dimensional optimization problems. No development of the AE code is needed. Generally, SPSA converges in the same number of iterations as FDSA and it follows approximately the steepest descent direction, behaving like the gradient method [15]. On the other hand, SPSA, with the random search direction, does not follow exactly the gradient path. In average, though, it tracks the gradient nearly because the gradient approximation is an almost unbiased estimator of the gradient, as shown in [9].

For SPSA algorithm, the conditions for the sequences of positive $\{a_k\}$ and $\{c_k\}$ are

$$a_k > 0, c_k > 0, \quad a_k \to 0, \quad c_k \to 0, \\ \sum_{k=1}^{\infty} a_k = \infty, \sum_{k=1}^{\infty} (a_k/c_k)^2 < \infty \tag{16}$$

2.2.2 On the operator P_r

As shown in [10], $span[P_r]$ - the subspace, spanned by the columns of P_r, must be chosen so that the filter gain K ensures a stability of the filter. Mention that even the KF may suffer from instability. To ensure filter stability, P_r is constructed from all unstable and neutral eigenvectors of the fundamental matrix Φ (or real Schur vectors (ScVs), singular vectors). In practice, we choose P_r to be consisting of the column-vectors of S

$$S = \Phi X \tag{17}$$

which are results of integration of leading ScVs (columns of X). The columns in S have the meaning of the PE for the system state and are used to

approximate the ECM. As to the real ScVs, they are preferred to eigenvectors (or singular vectors) because the ScVs are real and their computation is numerically stable. Mention that computation of singular vector requires also adjoint code. The ensemble of columns of S plays the same role as an ensemble PE samples in the ensemble-based filtering technique for approximating the background ECM [4]. In section 2.3 another method based on stochastic perturbation technique for estimation of the ECM will be addressed.

2.2.3 On separation of vertical and horizontal variables structure in ECM [8]

Let us consider the situation when the dynamical system (DS) is described by PDEs. The state vector of the numerical model at the time instant k is $x_k = x_k(i,j,l)$ where (i,j,l) represents a grid point in three dimensional space. Actually for high-dimensional systems, the number of generated PE samples is about 100 which is very small compared to the dimension of the system state. Evidently, the number of PE samples is insufficient to well approximate the true ECM. To better approximate the ECM, in [8] it is assumed that the estimated ECM is a member of the class of ECMs having a separation of vertical and horizontal variables structure (SeVHS). This hypothesis is not new and is used in modeling the ECM in meteorological data assimilation [3]. The optimal ECM is found as a solution to the minimization problem

$$J(\theta) = E||M_d - M_v(\theta_1) \otimes M_h(\theta_2)||_F^2,$$
$$||.||_F \text{denotes matrix Frobenious norm},$$
$$J(\theta) \rightarrow \min_\theta, \theta = (\theta_1^T, \theta_2^T)^T. \quad (18)$$

In (18) M_h is a "data" ECM which is obtained from an ensemble of PE samples. As the number of vertical layers in the today's numerical models is of order 10^1, all elements of the vertical ECM M_v (included in θ_1) can be considered as unknown to be estimated. As to M_h, it is often chosen in an analytical form (for example, the 1st or 2nd order autoregressive models). The parameters like correlation length ... can be selected as components of the control vector θ_2 in M_h.

2.3 A simple method for estimating a high-dimensional matrix

The minimization problem (18), in fact, gives us the way to decompose the "data" matrix M_d into a Kronecker product of two matrices $M_v(\theta_1)$ and $M_h(\theta_2)$. Unlike the traditional Nearest Kronecker Product problem (NKP) [6] where all elements of M_v and M_h have to be estimated, here the objective is to seek only some optimal parameters in these two matrices.

2.3.1 Estimation of a matrix

Suppose we are given the system $\Phi x = b$, $\Phi \in R^{m \times n}$. We want to find a Φ subject to the condition that for a given x', the product $y' = \Phi x'$ is known. This problem happens frequently in data assimilation when in (1) Φ is unknown, but there exists a computer code for computing $\Phi x'$ for some x'. Remark that for $b := (b_1, ..., b_m)^T$, the derivatives of b_i wrt to the vector x is defined as

$$db_i/dx = (\partial b_i/\partial x_1, ..., \partial b_i/\partial x_n) = (\phi_{i1}, ..., \phi_{in}), i = 1, ..., m$$

where ϕ_{ij} are the ij element of Φ. We can write then

$$db/dx = [(db_1/dx)^T, ..., (db_m/dx)^T]^T = \Phi$$

In what follows, we present a low-cost algorithm for approximating derivatives of b wrt x, independently of the dimension of x, and next to estimate Φ. The idea on estimation of high dimensional Φ on the basis of stochastic simultaneous perturbation (SSP) has been first briefly presented in [9].

Let $\bar{\Delta} := (\Delta_1, ..., \Delta_n)^T$, $\Delta_i, i = 1, ..., n$ be iid Bernoulli distributed variables assuming two values +/- 1 with equal probabilities 1/2. Introduce $[\bar{\Delta}]^{-1} := (1/\Delta_1, ..., 1/\Delta_n)^T$, $\bar{\Delta}_c := c\bar{\Delta}, c > 0$ is a small positive value.

In the context of estimating Φ, the proposed algorithm looks as follows :

Algorithm 1. Suppose it is possible to compute the product $\Phi x = b(x)$ for a given x. At the beginning let $l = 1$. Let the value u be assigned to the vector x, i.e. $x := u$, L be a (large) fixed integer number.

Step 1. Generate $\bar{\Delta}^{(l)}$ whose components are l^{th} samples of the Bernoulli i.i.d. variables assuming two values +/- 1 with equal probabilities 1/2;

Step 2. Compute $\delta b^{(l)} = \Phi(u + \bar{\Delta}_c^{(l)}) - \Phi u$, $\bar{\Delta}_c^{(l)} = c\bar{\Delta}^{(l)}$, c is a small positive value;

Step 3. Compute $g_i^{(l)} = \delta b_i^{(l)} [\bar{\Delta}_c^{(l)}]^{-1}$, δb_i is the i^{th} component of δb, $g_i^{(l)}$ is the column vector consisting of derivative of $b_i(u)$ wrt to u, $i = 1, ..., m$.

Step 4. Go to Step 1 if $l < L$. Otherwise, go to Step 5.

Step 5. Compute

$$\hat{g}_i = \frac{1}{L} \sum_{l=1}^{L} g_i^{(l)}, i = 1, ..., m,$$
$$\hat{\Phi}(L) := D_x b = [\hat{g}_1, ..., \hat{g}_m]^T$$

Theorem 1 Consider **Algorithm 1** for estimation of the elements of the matrix Φ. Then the sequence $\hat{\Phi}(L)$ in **Algorithm 1** converges to the true Φ with MSE proportional to $(1/L)$ where L is the number of samples used in the estimation procedure.

2.3.2 Estimation of decomposition of Φ

For very high dimensional Φ, it is impossible to store all the elements of Φ. One way to overcome this difficulty is to approximate Φ by some matrix in a subspace of fewer dimensions (for example, the class of matrices of given rank). Let $\Phi \in R^{m\times n}, m \leq n$, $\text{rank}(\Phi) = m$. We want to find the best Φ among members of the class of matrices

$$\Phi_e = AB^T, A \in R^{m\times r}, B \in R^{n\times r},$$
$$A, B \text{ are matrices of dimensions}$$
$$m \times r, r \times n, r \leq m, \text{rank}(AB^T) = r. \quad (19)$$

Under the condition (19), the optimization problem is formulated as

$$J(A,B) = ||\Phi - \Phi_e||_F^2 =$$
$$||\Phi - AB^T||_F^2 \rightarrow \min_{(A,B)}, \quad (20)$$

where $||.||_F$ denotes the Frobenius matrix norm. Consider Φ and let $U\Sigma V^T$ be SVD (singular value decomposition) of Φ [6], i.e.

$$\Phi = U\Sigma V^T, U \in R^{m\times m}, V \in R^{n\times n}, \Sigma = [\Sigma_m|0],$$
$$\Sigma_m = \text{diag}[\sigma_1, ..., \sigma_m], \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m \geq 0. \quad (21)$$

Theorem 2 Suppose $A_oB_o^T$ is a solution to the problem (19)-(20). Then

$$J(A_o, B_o) = \sum_{k=r+1}^{m} \sigma_k^2 \quad (22)$$

Theorem 2 implies that $\Phi_e^o := A_oB_o^T$ is equal to the matrix formed by truncating the SVD of Φ (21) to its first r singular vectors and singular values.

In the same way (like **Algorithm 1**), one can write out the algorithm for estimating A and B based on SSP of all elements of A and B (denoted as **Algorithm 2**).

3 Variational method (VM) [16]

Return to the problem of estimating $\{x_k\}$ in (1)-(2). The VM consists of minimizing

$$J[x_0, ..., x_N] = e_0M_0^{-1}e_0 +$$
$$\sum_{k=1}^{N}(z_k - Hx_k)^TR^{-1}(z_k - Hx_k),$$
$$J[x_0, ..., x_N] \rightarrow \min_{[x_0,...,x_N]}, \quad (23)$$
$$\text{under the constraints } (1) \quad (24)$$

where $e_0 := x_0 - \bar{x}_0$. Thus, the VM seeks an optimal solution in the functional space (space of functions $\{x_k\}$). For systems of high dimension, this task is impossible to perform. The simplification is required. Suppose the system (1) is perfect, i.e. $w_k = 0$. Expressing all x_k as functions of the initial state x_0,

$$x_k = \Phi(k,0)x_0,$$
$$\Phi(k,l) = \Phi^{k-l}, (k > l), \Phi(k,k) = I, \quad (25)$$
I is the identity matrix of appropriate dimension

and substituting $x_k, \forall k$ (25) into (2), at each k^{th} observation instant we have

$$z_k = H_k^1x_0 + \epsilon_k', k = 1, 2, ..k = 1, 2, ... \quad (26)$$
$$H_k^1 := [(H_1\Phi(1,0))^T, ..., (H\Phi(k,0))^T]^T,$$
$$v_k^1 = [v_1^T, ..., v^T]^T.$$

The optimization problem (23)-(24) now is simplified,

$$J[x_0] \rightarrow \min_{[x_0]}, \quad (27)$$
$$J[x_0] := e_0^TM_0^{-1}e_0 +$$
$$\sum_{k=1}^{N}(z_k - H_k'x_0)^TR_k^{-1}(z_k - H_k'x_0), \quad (28)$$
$$H_k' := H\Phi(k,0).$$

We have now the unconstrained optimization problem (27)(28) with the control vector $\theta := x_0$ - the initial state. This problem can be solved using standard optimization techniques [6].

It is not hard to write out a solution to the problem (27)(28). For high-dimensional systems, the problem (27)(28) is solved iteratively to find θ,

$$\nabla_\theta J[\theta] = 0$$
$$\nabla_\theta J[\theta] := [\partial J/\partial\theta_1, ..., \partial J/\partial\theta_1]^T \quad (29)$$

Comment 1 Usually, finding a solution to (27)(28) requires about 20-30 iterations to reach a relatively good approximate solution.

Comment 2 Writing out $\nabla_\theta J[\theta]$ shows that solving (29) requires implementation of the operation

$$(H_k')^Ty = \Phi_k^T\Phi_{k-1}^T...\Phi_1^TH^Ty$$

for some y. The AE method is used then for computing the product Φ_k^Ty. In the next section we see that the SPSA can also be used to solve this problem at a much lower cost.

4 Differences between AF and VM

4.1 Differences between VM and AF

We list here the main differences between two approaches VM and AF from which it becomes clear what are the advantages of the AF over the VM.

(D1) Dynamical system (DS): if in (23)(24), the DS is the initial system (1), in (6) the DS is the filtering equation (3)(4). This difference has an interesting consequence : if in practice, there is very little known about statistics of w_k, the sequence ζ_k is observed and hence it is possible to estimate its statistics.

(D2) w_k in (1) is white, while ζ_k in (10) is a white only if the filter is optimal : This allows to apply different statistical tests for verifying the optimality of the assimilation procedure.

(D3) Control variable x_0 in the VM is the initial state, whereas the control variable in (6) is the gain parameters.

This difference has an important consequence : as x_0 has to be of precise physical meaning (depending, for example, on the ocean domain of interest), the structure for the guess $\theta_0 := \hat{x}_0^0$ - initial state, as well as its correction $\delta\hat{x}_0^\nu$, must be chosen carefully so that $\hat{x}_0^\nu$, $\hat{x}_0^\nu = \hat{x}_0^{\nu-1} + \delta\hat{x}_0^\nu$, must be of physically admissible structure. That is not an easy task. As to the AF, the parameters usually are immaterial (see θ in (10)) hence the choice of structure for θ is of no importance.

(D4) Suppose (1) is unstable. It implies the error growth in estimating x_0 during integration of the direct and AE. As for the AF, by its construction, the filtering system remains stable. This can be seen by representing the filtering Eqs (3)(4) through its fundamental matrix L_k,

$$\begin{aligned} \hat{x}_k &= L_k\hat{x}_{k-1} + K_k v_k \\ L_k &= (I - K_k H)\Phi_k. \end{aligned} \tag{30}$$

As shown in [10], the filter (30) is stable under the conditions (9)(10). It means that the filtering error is bounded during model integration since the parameters θ_i are lying in the interval guaranteeing a stability of the filter (18).

(D5) Return to the objective function (23)(24). Taking the derivative of (23)(24) wrt x_0 yields

$$\begin{aligned} \nabla_{x_0} J[\hat{x}_0^\nu] = M_0^{-1}e_0^\nu + \\ -\sum_{k=1}^{N} \Phi^T(k,0)H_k^T R_k^{-1}(z_k - H\Phi(k,0)x_0^\nu) = \\ -\sum_{k=1}^{N} \Phi^T(k,0)H_k^T R_k^{-1}(H\Phi(k,0)e_0^\nu + v_k), \\ e_0^\nu := x_0 - \hat{x}_0^\nu. \end{aligned} \tag{31}$$

Table 1: Number of elementary arithmetic operations

VM	$(n^2(N^2+N) + N(2np+p^2) + n^2)N_{it_o}$
AF	$[(n^2 + 2np + p^2)N_{it} + 2(np+n^2)]N$
ROF	$[CN_{it} + n^2 + np + n_e(n_e + p + 1)]N$
	$C = n_e^2 + 2n_e p + p^2$

One sees that Eq. (31) requires computation of N terms (without counting for the term $M_0^{-1}e_0^\nu$). The k^{th} term is associated with the assimilation instant k and one needs to compute first $\mu_k := \Phi(k,0)e_0^\nu$ - i.e. to integrate k times the direct model Φ^k and next to integrate backward (k times also) the AE Φ^T, i.e. to compute $\Phi^T(k,0)H_\kappa^T R_\kappa^{-1}(H\mu_k + v_k)$. The larger k, the bigger amplification of the initial error e_0^ν and the observation error v_k. The error e_0^ν is amplified doubly since it is integrated by the direct and adjoint models. But the amplification of v_k (and w_k when $w_k \neq 0$) is most worrying since it is integrated in the gradient estimate, making the gradient direction to be, possibly, completely erroneous.

4.2 Computational comparison between VM and AF

We give here a brief comparison (computational burden) between the VM and AF algorithms based on AE approach. Here the AF has the gain structure (7)-(9). Table 1 shows the number of elementary arithmetic operations required for the implementation of the VM and AF filtering algorithms based on AE approach. These numbers are related to gradient computations since they represent the most computational burdens in the implementation of VM and AF algorithms. Here the AF means that the ECM $M \in R^{n\times n}$ is given. The numbers of operations are rounded up to the dimensions of the entry matrices. Here N_{it_o} is a number of iterations required to solve the minimization problem (23)-(24), N_{it} is a number of iterations required to solve the equation of the type $\Xi y = \zeta_k, \Xi := [HMH + R]$.

In the MV algorithm, the computation of M_0^{-1}, R^{-1} is not taken into account. In Table 1 there is also the number of operations required for the ROF, when the ECM M is given in the form $M = P_r P_r^T$, $P_r \in R^{n\times n_e}$.

Comment 3 Looking at Table 1 one sees that the dominant numbers n_d of operations in VM and NAF are $n_d(VM) = n^2N^2N_{it_o}$ and $n_d(NAF) = n^2NN_{it}$. If we assume that $N_{it_o} \approx N_{it}$ the number $n_d(VM)$ is N times larger than $n_d(NAF)$.

5 Numerical experiments

5.1 Lorenz system

The experiment is performed for the following Lorenz attractor [13],

$$\frac{dy_1}{dt} = -\sigma(y_1 - y_2), \frac{dy_2}{dt} = \rho y_1 - y_2 - y_1 y_3, \quad \frac{dy_3}{dt} = y_1 y_2 - \beta y_3, \quad (32)$$

where σ is called the Prandtl number and ρ is called the Rayleigh number. All σ, β, $\rho > 0$, but usually σ = 10, β = 8/3 and ρ is varied. The system exhibits chaotic behavior for ρ = 28 but displays knotted periodic orbits for other values of ρ.

In the experiments the parameters σ, ρ, β are chosen to have the values 10, 28 and 8/3 for which the "butterfly" attractor exists.

The numerical model is obtained by applying the Euler method (first-order accurate method) to approximate (32). The corresponding model time step is $\delta t = 0.005$, and the interval between two observations is $\Delta T_k = 1$. Thus we are given the sequence of observations $z(k) := z(T_k), k = 1, ..., N_o$ where $z(k)$ consists of the first and third components x_1, x_3 observed at each time instant $T_k, k = 1, ..., 100$. The observation error is zero mean uncorrelated sequence with the covariance $R = 2I_2$. As to the model error, the zero mean uncorrelated sequence with variances 2, 12.13 and 12.13 is added to the numerical model at each instant T_k. The true system state x^* is simulated subject to the initial condition $x_0^* = (1.508870, -1.531271, 25.46091)^T$.

The problem considered in this experiment is to apply the Extended KF (EKF), non-adaptive filter (NAF) and adaptive filter (AF) to estimate the true system state using the observations z_k, $k = 1, 2, ..., N_o$ and to compare their performances. In all the filters the initial state is $\hat{x}_0 = (5, 10, 27)^T$. Here the NAF is the Prediction Error Filter (PEF) described in [7].

Figure 1 shows the evolution of the prediction errors resulting from three filters: NAF(L1), AF(L1) and EKF. The symbol $L1$ in the NAF signifies that the gain in the NAF is estimated on the basis of an ensemble of samples for the first Schur vector (or Krylov vectors). It is of no surprise that the NAF has produced the estimates with larger estimation error. By adaptation, however, it is possible to design an AF (on the basis of the NAF) which improves significantly NAF performance and even behaves better than the EKF.

Mention that the VM is much less appropriate for assimilating the observations in the Lorenz model due

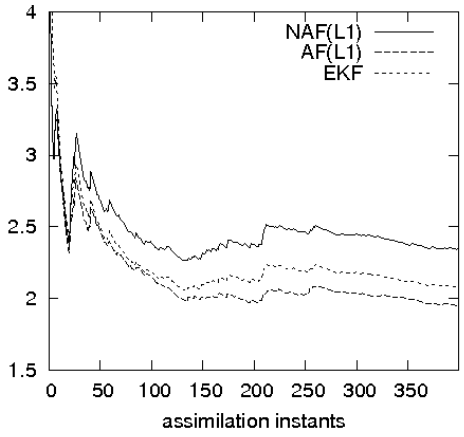

Figure 1: Prediction errors resulting from three filters: Nonadaptive filter (NAF), Adaptive filter (AF) and EKF

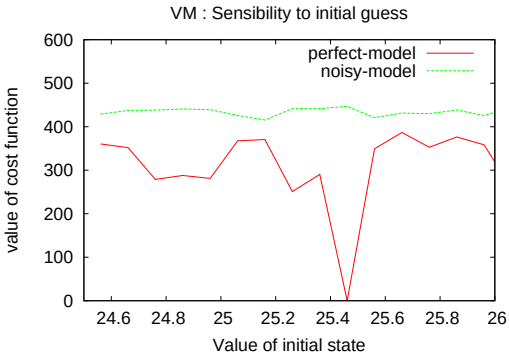

Figure 2: Time averaged variance between the true trajectory and model trajectory in the VM as a function of perturbed third component of the initial state. The global minimum is attained at the true initial condition, but there is no guarantee for the VM to approach the true initial state even in the perfect model case. For the noisy model, the global minimum is not attained at the true initial state. The curve "noisy-model" is scaled by the factor $C = 1/15$.

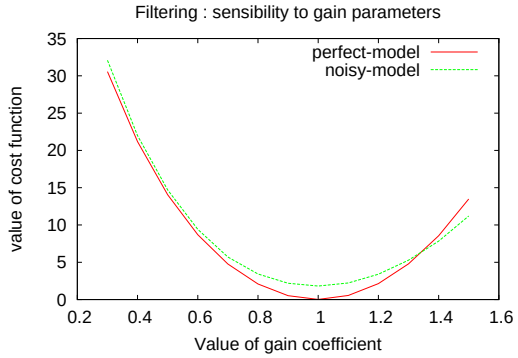

Figure 3: Cost function in the PEF as a function of perturbed third gain parameter θ_3. It is seen that in the PEF the cost function is quadratic wrt to the gain parameter in both situations of the perfect and noisy models. The curve "noisy-model" is scaled by the factor $C = 1/50$.

to the choice of the initial state as a control vector. For simplicity, let three components be observed, $H = I_3$.

In Figure 2 we show the cost function (time averaged variances of the difference between the true trajectory and model trajectory, denoted as $AV(x^*, \hat{x})$), resulting from varying the third component of the initial state for two situations of the perfect model and noisy model. Namely, we initialize the model by the initial state, which is the same as the true one, with the difference, that the third component $x_3(0)$ is varying in the interval $[24.5 : 26.5]$. For the perfect model, the global minimum is attained at $x_0^*(3) = 25.46091$ as expected. However, if the system is initialized by the estimate in a vicinity, even not so far from the true $x_3^*(0)$, there is no guarantee that the VM can approach the true initial state and the resulting estimation error may be very large. For the noisy model, the global minimum is not attained at $x_0^*(3)$.

As for the PEF, the function $AV(x^*, \hat{x})$ is quadratic wrt to the gain parameter, for both situations of the perfect and noisy models, as seen in Figure 3 : here the sample cost function is averaged over all assimilation period, by varying the third parameter θ_3 in the gain (related to the third observed component of the system state). From Figure 2 and Figure 3 it is seen that the estimation error in the VM is about 10 times larger than that produced by the filtering algorithm for the perfect model case.

5.2 Estimation of matrix

Let us consider the problem of estimating elements of the following matrix $\Phi \in R^{4\times 5}$ (see https://en.wikipedia.org/wiki/Singular-value-decomposition)

$$\Phi := A = [a_{i,j}] = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{pmatrix} \quad (33)$$

In the SVD-decomposition the matrix A has 3 non-zero singular values $\sigma_1 = 4$, $\sigma_2 = 3$, $\sigma_3 = \sqrt{5}$. In the experiment, first we have applied **Algorithm 1** to obtain the estimate A_e and next decomposed this matrix **Algorithm 2** into the product of two matrices.

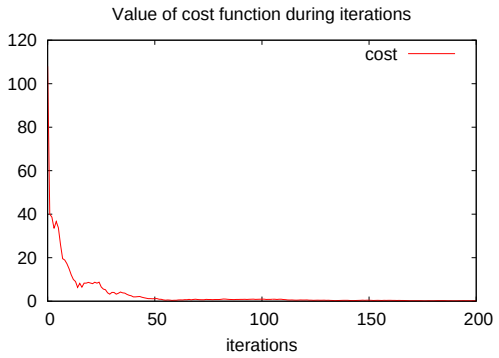

Figure 4. Cost function resulting from **Algorithm 1** : convergence is observed after about 50 iterations

Figure 4 displays the cost function during the estimation process by **Algorithm 1**. It is seen that the process converges after about 50 iterations.

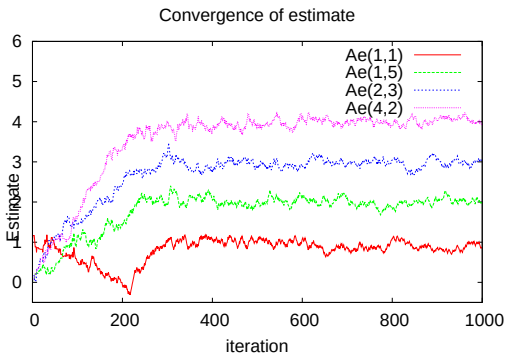

Figure 5. Estimates of matrix elements by three-dimensional subspace approximation : All elements $a_{4,2} = 4, a_{2,3} = 3, a_{1,5} = 2, a_{1,1} = 1$

Next we solve the decomposition problem (20) (by **Algorithm 2**, section 2.3.2) subject to $\Phi_e := AB^T, A \in R^{m\times 3}, B \in R^{n\times 3}$. rank $(AB^T) = r = 3$. Fig. 5 shows the estimates for four elements $A_e(1,1), A_e(1,5), A_e(2,3), A_e(4,2)$ produced by the

decomposition problem. One sees that all the coefficients converge to their true values, but compared to Figure 3, the decomposition algorithm requires more iterations to reach a convergence. A more quick convergence is obtained for $r = 4$ (not shown here).

6 Assimilation in high dimensional ocean model MICOM

In this section first we apply the results in section 2 (**Algorithm 1**) to obtain the "data" ECM M^d_{ssp} based on an ensemble of PE samples generated by the SSP method (denoted as $En(SSP)$). The decomposition algorithm (**Algorithm 2**) will be implemented to estimate the unknown parameters of the ECM $\hat{M}_{ssp}$ represented in the form of the Kronecker product as shown in (20). Once having $\hat{M}_{ssp}$ we can construct the filter PEF(SSP).

The other "data" ECM M^d_{sch} is formed on the basis of an ensemble of PE samples generated by the sampling procedure in [7] (denoted as $En(SCH)$. In the same way, the problem (20) is solved to estimate the second ECM $\hat{M}_{sch}$. We have then the second filter PEF(SCH)). The efficiency of these two PEFs as well as that of the CHF (Cooper-Haines Filter, [2]) will be compared in the context of the experiment with sea surface height (SSH) data assimilation with the ocean model MICOM. Mention that the CHF is a filter widely used in the SSH data assimilation in oceanography. It is constructed on the basis of the principle of lowering and lifting the water column.

6.1 Ocean model MICOM

The ocean model used here is the MICOM (Miami Isopycnal Ocean Model) which is identical to that described in [7]. The model configuration is a domain situated in the North Atlantic from 30^0 N to 60^0 N and 80^0 W to 44^0 W; for the exact model domain and some main features of the ocean current (mean, variability of the SSH, velocity ...) produced by the model, see [7]. The state of the model is $x := (h, u, v)$ where $h = h(i, j, lr)$ is the thickness of lr^{th} layer, $u = u(i, j, lr), v = v(i, j, lr)$ are two velocity components. The layer stratification is made in the isopycnal coordinates, i.e. stratification characterized by a constant potential density of water. Thus, with three variables $x := (h, u, v)$, the state of the numerical model has the dimension $n = 302400$.

To be closer to realistic situations with the observations available only at along-track grid points, only the SSH values from the true state situated at the points $i = 1, 11, ..., 131$, $j = 1, 11, ..., 171$ are taken as observations. They are noise-free.

Table 2: rms of prediction error for ssh, and u, v velocity components

rms	CHF	PEF(SCH)	PEF(SSP)
ssh(fcst) (cm)	6.455	4.091	3.704
u(fcst) (cm/s)	7.501	5.255	4.966
v(fcst) (cm/s)	7.618	5.599	5.331

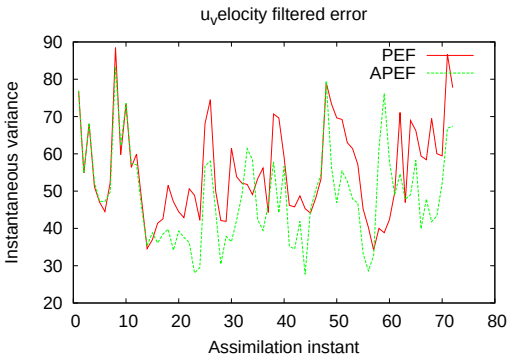

Figure 4: Filtered errors for the u-velocity component estimate, resulting from PEF(SCH) and APEF (SCH) (AF based on PEF(SCH)). Optimization is performed by SPSA. By tuning the parameters in the filter gain, one can improve considerably the performance of the PEF

6.2 Assimilation results

In Table 2 the performances of the three filters are displayed. The errors are averaged (spatially and temporally) root mean square (rms) of prediction error for the SSH and two velocity components u and v.

The results in Table 2 show that two filters PEF(SCH) and PEF(SSP) largely outperform the CHF, with a slightly better performance for the PEF(SSP). It may be explained by the fact that the best theoretical performance of PEF(SCH) can be obtained only if the model is linear, stationary and the number of PE samples in $En(SCH)$ at each iteration must be large enough. The ensemble size of $En(SCH)$ in the present experiment is too small compared with the dimension of the MICOM model. As to $En(SSP)$, the samples are obtained by stochastically perturbing all directions of the state, hence probably they capture more variability of the prediction error.

To see the effect of adaptation, Figure 4 displays the filtered estimation errors for the u-velocity component at the surface, produced by the PEF (i.e.

PEF(SCH)) and by its adaptive version (APEF) respectively. Here the free parameters of the gain are optimized by the SPSA method. For the choice of tuning parameters in the gain, see [10]. From the computational point of view, the SPSA requires much less time integration and memory storage compared with the traditional the AE method. At each assimilation instant, we have to make only two integrations of the MICOM for approximating the gradient vector. From Figure 4 one sees that adaptation can serve as an useful tool to reduce significantly the estimation error in the non-adaptive filter.

7 Conclusion

In this paper the AF based on innovation approach is presented and its performance is compared with other estimation methods. The optimality of such AF is defined as minimal PE for the system outputs, with the control variables chosen as pertinent parameters in the gain. This choice is very beneficial : it allows to formulate and solve the optimization problem for high-dimensional AF on the basis of the SPSA approach. We show further how one can estimate the elements of an unknown matrix of high dimension and decompose it into the product of two matrices, in a simple way, on the basis of SSP technique, under the condition that only the matrix-vector product is accessible. By this way there is no difficulty in manipulating matrices of high dimensions as well as generating an informative ECM for the construction of the filter gain.

Numerical experiments in section 5 (low and moderate systems) and section 6 (high dimensional system) well illustrate the theoretical results and the efficiency of the proposed algorithms.

References:

[1] Y. Baram and G. Kalit, Order reduction in linear state estimation under performance constraints. *IEEE Trans. Autom. Contr.*, AC-32, No. 11, 1987, pp. 983-989.

[2] M. Cooper and K. Haines, Altimetric assimilation with water property conservation. *J. Geophys. Res.*, 101, 1996, pp. 1059-1077.

[3] R. Daley, The effect of serially correlated observation and model error on atmospheric data assimilation. *Monthly Weather Review*, 120, 1992, pp. 165-177

[4] G. Evensen, The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 2003, pp. 343-367.

[5] M. Ghil and P. Manalotte-Rizzoli, Data assimilation in meteorology and oceanography. *Adv. Geophys.*, 33, 1991, pp. 141-266.

[6] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, 1996.

[7] H.S. Hoang H.S. and R. Baraille, Prediction error sampling procedure based on dominant Schur decomposition. Application to state estimation in high dimensional oceanic model *J. App. Math. and Comput.*, V. 218, Iss. 1, Dec., 2011, pp. 3689 - 3709

[8] H.S. Hoang H.S. and R. Baraille, A Low Cost Filter Design for State and Parameter Estimation in Very High Dimensional Systems. *Proc. of the* 19^{th} *IFAC Congress*, August, 2014, pp. 3256-3261.

[9] H.S. Hoang and R. Baraille, Stochastic Simultaneous Perturbation as Powerful Method for State and Parameter Estimation in High Dimensional Systems. In : Albert R. Baswell, editors, *Advances in Mathematics Research*, V. 20, Nova Scie. Pub.; 2015, pp. 117-148.

[10] H.S. Hoang, O. Talagrand, R. Baraille, On the design of a stable adaptive filter for high dimensional systems. *Automatica*, 37, 2001, pp. 341-359.

[11] R.E. Kalman, A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82, 1960, pp. 35-45.

[12] M. Gevers and T. Kailath, An Innovations Approach to Least-Squares Estimation, Pt. VI: Discrete-Time Innovations Representations and Recursive Estimation, *IEEE Trans. Automatic Control*, 18(6), December, 1973, pp. 588-600.

[13] E.N. Lorenz, Deterministic non-periodic flow. *J. Atmos. Sci.*, 20, 1963, pp. 130-141.

[14] D. Simon, *Optimal State Estimation*, Hoboken, NJ: John Wiley and Sons, 2006.

[15] J.C. Spall, Accelerated Second-Order Stochastic Optimization Using Only Function Measurements. *Proc. 36th IEEE Conf. on Decision and Control*, 1997, pp. 14171424.

[16] O. Talagrand and P. Courtier, Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Quart. J. Roy. Meteor. Soc.*, 113, 1987, pp. 1311-1328.