

Article

# A Robust Non-Gaussian Data Assimilation Method for Highly Non-Linear Models

Elias D. Nino-Ruiz <sup>1,\*</sup> , Haiyan Cheng <sup>2</sup> and Rolando Beltran <sup>1</sup>

<sup>1</sup> Applied Math and Computational Science Laboratory, Department of Computer Science, Universidad del Norte, Barranquilla 080001, Colombia; rbeltran@uninorte.edu.co

<sup>2</sup> Department of Computer Science, Willamette University, 900 State Street, Salem, OR 97301, USA; hcheng@willamette.edu

\* Correspondence: enino@uninorte.edu.co; Tel.: +57-5-3509268

Received: 5 January 2018; Accepted: 20 March 2018; Published: 26 March 2018



**Abstract:** In this paper, we propose an efficient EnKF implementation for non-Gaussian data assimilation based on Gaussian Mixture Models and Markov-Chain-Monte-Carlo (MCMC) methods. The proposed method works as follows: based on an ensemble of model realizations, prior errors are estimated via a Gaussian Mixture density whose parameters are approximated by means of an Expectation Maximization method. Then, by using an iterative method, observation operators are linearized about current solutions and posterior modes are estimated via a MCMC implementation. The acceptance/rejection criterion is similar to that of the Metropolis-Hastings rule. Experimental tests are performed on the Lorenz 96 model. The results show that the proposed method can decrease prior errors by several order of magnitudes in a root-mean-square-error sense for nearly sparse or dense observational networks.

**Keywords:** ensemble Kalman filter; Gaussian Mixture Models; non-linear observation operator; Markov-Chain-Monte-Carlo

## 1. Introduction

Data Assimilation is the process of optimally combining the imperfect numerical forecast states and imperfect observations to better estimate the state  $\mathbf{x}^* \in \mathbb{R}^{n \times 1}$  of a system that evolves according to some model operator [1–3],

$$\mathbf{x}_{next}^* = \mathcal{M}_{t_{current} \rightarrow t_{next}}(\mathbf{x}_{current}^*), \quad (1)$$

where,  $\mathcal{M} : \mathbb{R}^{n \times 1} \rightarrow \mathbb{R}^{n \times 1}$  is an imperfect numerical model that evolves the states,  $n$  is the number of model components. One example of  $\mathcal{M}$  could be a model that mimics the ocean and/or the atmosphere dynamics. In traditional DA settings, prior errors are described by Gaussian distributions,

$$\mathbf{x} \sim \mathcal{N}(\mathbf{x}^b, \mathbf{B})$$

where  $\mathbf{x}^b \in \mathbb{R}^{n \times 1}$  and  $\mathbf{B} \in \mathbb{R}^{n \times n}$  is the background state and the background error covariance matrix, respectively. This assumption has plenty of computational benefits, for instance, by assuming Gaussian errors in the prior and the observations, Kalman-like updates can be performed in order to compute posterior (error) moments. However, in the context of geophysics, model dynamics can be highly non-linear and therefore, Gaussian Mixture Models (GMM) [4,5] can be used to capture our prior knowledge about the error dynamics:

$$\mathbf{x} \sim \sum_{k=1}^K \alpha_k^b \cdot \mathcal{N}(\mathbf{x}_k^b, \mathbf{B}_k), \quad \text{with} \quad \sum_{k=1}^K \alpha_k^b = 1 \quad (2)$$

where, for the  $k$ -th mixture component,  $\mathbf{x}_k^b \in \mathbb{R}^{n \times 1}$  is the mean,  $\mathbf{B}_k \in \mathbb{R}^{n \times n}$  is the background error covariance, and  $\alpha_k^b$  is the prior weight, for  $1 \leq k \leq K$ . These weights can be estimated, for instance, using the Expectation Maximization algorithm [6–8]. In sequential methods, it is common to assume Gaussian errors over observations  $\mathbf{y} \in \mathbb{R}^{m \times 1}$ ,

$$\mathbf{y} \sim \mathcal{N}(\mathcal{H}(\mathbf{x}), \mathbf{R}) \tag{3}$$

where  $m$  is the number of observed components from the model state,  $\mathbf{R} \in \mathbb{R}^{m \times m}$  is the data error covariance matrix, and  $\mathcal{H} : \mathbb{R}^{n \times 1} \rightarrow \mathbb{R}^{m \times 1}$  is the observation mapping operator. When the operator  $\mathcal{H}$  is linear, the posterior error distribution can be described by a GMM [9] as well:

$$\mathbf{x}|\mathbf{y} \sim \sum_{k=1}^K \alpha_k^a \cdot \mathcal{N}(\mathbf{x}_k^a, \mathbf{A}_k) \tag{4}$$

with updated weights  $\alpha_k^a$ , centroids  $\mathbf{x}_k^a$ , and covariances  $\mathbf{A}_k$ , for  $1 \leq k \leq K$ , which account for the observation (3). It can be noted that, for the  $k$ -th mixture component, the posterior mode  $\mathbf{x}_k^a \in \mathbb{R}^{n \times 1}$  can be obtained by means of a 3D-Var optimization problem:

$$\mathbf{x}_k^a = \arg \min_{\mathbf{x}} \mathcal{J}_k(\mathbf{x}) \tag{5}$$

where

$$\mathcal{J}_k(\mathbf{x}) = \frac{1}{2} \cdot \left\| \mathbf{x} - \mathbf{x}_k^b \right\|_{\mathbf{B}_k^{-1}}^2 + \frac{1}{2} \cdot \left\| \mathbf{y} - \mathcal{H}(\mathbf{x}) \right\|_{\mathbf{R}^{-1}}^2 \tag{6}$$

Under linear assumptions,  $\mathbf{x}_k^a$  can be estimated by means of, for instance, an EnKF updating formula. However, for non-linear observation operators, such expression can fail to obtain reasonable estimates of posterior modes and therefore, other alternatives such as sampling methods are employed. For instance, Monte-Carlo based methods are commonly utilized in order to relax the Gaussian assumption in the observational errors. Thus, we propose an efficient sampling method to draw samples from the posterior distribution (4) using cost functions of the form (6). In general, the method works as follows: for a fixed number of clusters, a GMM is fitted with the EM method, then, for each prior component, samples are drawn along steepest descent approximations of the 3D-Var cost function; these samples are accepted/rejected based on the Metropolis–Hastings rule. Besides, background error correlations are estimated based on a modified Cholesky decomposition in order to perform implicit localization and to reduce the impact of sampling errors, a major concern in the context of DA.

This paper is organized as follows: Section 2 discusses EnKF formulations and sampling methods for relaxing Gaussian assumptions in prior and observation errors. Section 3 presents an EnKF formulation based on GMM and MCMC (Markov-Chain-Monte-Carlo) for computing posterior modes together with error statistics; the computational cost of the method is estimated as well. In Section 4, experimental tests are performed on the Lorenz-96 model and non-linear observation operators; conclusions are finally stated in Section 5.

## 2. Preliminaries

### 2.1. Ensemble Kalman Filters Based on Modified Cholesky Decomposition

In sequential Data Assimilation (DA), under prior Gaussian assumptions, a well-known filter is the ensemble Kalman filter (EnKF) [10,11]. In EnKF, using an ensemble of model realizations,

$$\mathbf{X}^b = \left[ \mathbf{x}^{b[1]}, \mathbf{x}^{b[2]}, \dots, \mathbf{x}^{b[N]} \right] \in \mathbb{R}^{n \times N} \tag{7a}$$

the hyper-parameters of the error distribution,

$$\mathbf{x} \sim \mathcal{N}(\mathbf{x}^b, \mathbf{B}) \tag{7b}$$

are estimated as:

$$\mathbf{x}^b \approx \bar{\mathbf{x}}^b = \frac{1}{N} \cdot \sum_{e=1}^N \mathbf{x}^{b[e]} \in \mathbb{R}^{n \times 1} \tag{7c}$$

and

$$\mathbf{B} \approx \mathbf{P}^b = \frac{1}{N-1} \cdot \Delta \mathbf{X} \cdot \Delta \mathbf{X}^T \in \mathbb{R}^{n \times n} \tag{7d}$$

where  $\mathbf{x}^b \in \mathbb{R}^{n \times 1}$  is the background state,  $\mathbf{B} \in \mathbb{R}^{n \times n}$  is the background error covariance matrix,  $N$  is the ensemble size,  $\mathbf{x}^{b[e]}$  denotes the  $e$ -th ensemble member, for  $1 \leq e \leq N$ .  $\bar{\mathbf{x}}^b$  is known as the ensemble mean, and  $\mathbf{P}^b$  is the ensemble covariance matrix. The matrix of member deviations  $\Delta \mathbf{X} \in \mathbb{R}^{n \times N}$  reads

$$\Delta \mathbf{X} = \mathbf{X}^b - \bar{\mathbf{x}}^b \cdot \mathbf{1}^T \tag{7e}$$

and  $\mathbf{1}$  is a vector of consistent dimension whose components are all ones. When an observation  $\mathbf{y}$  becomes available, in the stochastic EnKF, the analysis ensemble can be estimated as follows:

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{P}^a \cdot \Delta \mathbf{Y} \in \mathbb{R}^{n \times N} \tag{7f}$$

where  $\Delta \mathbf{Y} = \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot [\mathbf{y} \cdot \mathbf{1}^T - \mathbf{H} \cdot \mathbf{X}^b + \mathbf{E}] \in \mathbb{R}^{n \times N}$  is the matrix of scaled innovations on the observations. The columns of  $\mathbf{E} \in \mathbb{R}^{m \times N}$  are formed by samples from a  $m$ -th dimensional standard Normal distribution.  $\mathbf{H} \in \mathbb{R}^{m \times n}$  is the linearized observation operator (with the linearization performed about the background state) also known as the Jacobian matrix of  $\mathcal{H}$  at the background state, and  $\mathbf{P}^a \in \mathbb{R}^{n \times n}$  is the analysis covariance matrix:

$$\mathbf{P}^a = \left[ [\mathbf{P}^b]^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} \right]^{-1} \tag{7g}$$

In operational DA, ensemble sizes are several orders of magnitude smaller than the model dimensions ( $N \ll n$ ) and as a consequence, the covariance matrix  $\mathbf{P}^b$  is commonly rank-deficient. This implies that (7g) cannot be directly computed and even though equivalent updating formulas which avoid the  $[\mathbf{P}^b]^{-1}$  calculation can be found in the literature, sampling errors can still impact the quality of the analysis corrections. In practice, localization methods are often used to artificially increase the degrees of freedom of  $\mathbf{P}^b$  and to mitigate the impact of sampling errors [12–15]. An efficient EnKF implementation which accounts for implicit localization during the assimilation step is the EnKF method based on a modified Cholesky decomposition (EnKF-MC) [16]. In this filter, the vanilla covariance (7d) is replaced by the Bickel and Levina estimator [17]:

$$\mathbf{B}^{-1} \approx \hat{\mathbf{B}}^{-1} = \mathbf{L}^T \cdot \mathbf{D} \cdot \mathbf{L} \in \mathbb{R}^{n \times n} \tag{8a}$$

where  $\mathbf{L} \in \mathbb{R}^{n \times n}$  is a lower triangular matrix whose diagonal elements are all ones. Its non-zero sub-diagonal elements are computed by fitting models of the form:

$$\mathbf{x}_{[i]}^T = \sum_{j \in P(i,r)} \mathbf{x}_{[j]}^T \cdot \{-\mathbf{L}\}_{i,j} + \boldsymbol{\varsigma}_i \in \mathbb{R}^{N \times 1}, \text{ for } 1 \leq i \leq n \tag{8b}$$

In which  $P(i, r)$  denotes the predecessors of the  $i$ -th grid component for some labelling of model components,  $\mathbf{x}_{[i]} \in \mathbb{R}^{1 \times N}$  stands for the  $i$ -th row of the matrix (7e), and, as an assumption,  $\boldsymbol{\varsigma}_i \in \mathbb{R}^{N \times 1}$  follows a zero-mean Normal distribution with uncorrelated errors of unknown variance  $\sigma^2$ . Likewise,  $\mathbf{D} \in \mathbb{R}^{n \times n}$  is a diagonal matrix whose diagonal elements are given by the reciprocal variances of the residuals in (8b):

$$\{\mathbf{D}\}_{i,i} = \mathbf{var} \left( \mathbf{x}_{[i]}^T - \sum_{j \in \mathcal{P}(i,r)} \mathbf{x}_{[j]}^T \cdot \{-\mathbf{L}\}_{i,j} \right)^{-1} \approx \frac{1}{\sigma^2} \tag{8c}$$

Similar to (7f), the analysis ensemble can be built as,

$$\mathbf{X}^a = \mathbf{X}^b + \hat{\mathbf{A}} \cdot \Delta \mathbf{Y} \tag{8d}$$

where an estimate of the analysis covariance is:

$$\hat{\mathbf{A}} = \left[ \hat{\mathbf{B}}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} \right]^{-1} \in \mathbb{R}^{n \times n} \tag{8e}$$

Matrix-free implementations of the EnKF-MC are currently proposed in the literature, for instance, the Posterior EnKF (P-EnKF) [18,19] exploits the special structure of (8a) in order to estimate Cholesky factors of the posterior precision covariance (8e). Consider

$$\hat{\mathbf{A}}^{-1} = \hat{\mathbf{B}}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} = \hat{\mathbf{B}}^{-1} + \mathbf{Z} \cdot \mathbf{Z}^T = \hat{\mathbf{B}}^{-1} + \sum_{o=1}^m \mathbf{z}^{[o]} \cdot \left[ \mathbf{z}^{[o]} \right]^T \in \mathbb{R}^{n \times n} \tag{9}$$

where  $\mathbf{z}^{[o]} \in \mathbb{R}^{n \times 1}$ , for  $1 \leq o \leq m$ , is the  $o$ -th column of matrix  $\mathbf{Z} = \mathbf{H}^T \cdot \mathbf{R}^{-1/2} \in \mathbb{R}^{n \times m}$ , the updating process can be done via a sequence of rank-one updates over the prior Cholesky factors:

$$\begin{aligned} \hat{\mathbf{A}}^{(0)} &= \left[ \mathbf{L}^{(0)} \right]^T \cdot \mathbf{D}^{(0)} \cdot \left[ \mathbf{L}^{(0)} \right] = \mathbf{L}^T \cdot \mathbf{D} \cdot \mathbf{L} = \hat{\mathbf{B}}^{-1} \\ \hat{\mathbf{A}}^{(1)} &= \hat{\mathbf{A}}^{(0)} + \mathbf{z}^{[1]} \cdot \left[ \mathbf{z}^{[1]} \right]^T = \left[ \mathbf{L}^{(1)} \right]^T \cdot \mathbf{D}^{(1)} \cdot \left[ \mathbf{L}^{(1)} \right] \\ \hat{\mathbf{A}}^{(2)} &= \hat{\mathbf{A}}^{(1)} + \mathbf{z}^{[2]} \cdot \left[ \mathbf{z}^{[2]} \right]^T = \left[ \mathbf{L}^{(2)} \right]^T \cdot \mathbf{D}^{(2)} \cdot \left[ \mathbf{L}^{(2)} \right] \\ &\vdots \\ \hat{\mathbf{A}}^{(m)} &= \hat{\mathbf{A}}^{(m-1)} + \mathbf{z}^{[m]} \cdot \left[ \mathbf{z}^{[m]} \right]^T \\ &= \left[ \mathbf{L}^{(m)} \right]^T \cdot \mathbf{D}^{(m)} \cdot \left[ \mathbf{L}^{(m)} \right] = \hat{\mathbf{L}}^T \cdot \hat{\mathbf{D}} \cdot \hat{\mathbf{L}} = \hat{\mathbf{A}}^{-1} \end{aligned}$$

where  $\mathbf{L}^{(0)} \in \mathbb{R}^{n \times n}$  and  $\mathbf{D}^{(0)} \in \mathbb{R}^{n \times n}$  are the Cholesky factors of  $\hat{\mathbf{B}}^{-1}$ . Having a posterior precision in the form  $\hat{\mathbf{A}}^{-1} = \hat{\mathbf{L}}^T \cdot \hat{\mathbf{D}} \cdot \hat{\mathbf{L}}$ , an estimate of the posterior ensemble can be easily built:

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{Q} \in \mathbb{R}^{n \times N} \tag{10a}$$

where  $\mathbf{Q} \in \mathbb{R}^{n \times N}$  is given by the solution of a lower triangular linear system:

$$\left[ \hat{\mathbf{L}}^T \cdot \hat{\mathbf{D}}^{1/2} \right] \cdot \mathbf{Q} = \Delta \mathbf{Y} \tag{10b}$$

In spite of EnKF implementations well-recognized by the DA community, the Gaussian assumption (7b) can be easily broken when the numerical model dynamics (1) are highly non-linear. For this reason, one prefers to use a different model to describe the prior error distribution. A Gaussian Mixture Model is frequently used to relax the Gaussian assumption over the forecast distribution [20–23].

### 2.2. Gaussian Mixture Models Based Filters

In EnKF formulations based on Gaussian Mixture Models (GMM), the prior error distribution (7b) is replaced by a mixture of Gaussian distributions (2). Based on this general idea, many methods have been proposed in the current literature in order to deal with highly non-linear dynamics of,

for instance, operational DA models. These methods rely on the assumption that GMM models can approximate arbitrary density functions. GMM components are commonly fitted by using the Expectation Maximization (EM) algorithm [24]. As the name implies, this iterative method works in two recursive steps: Expectation (E) and Maximization (M). For a fixed number of clusters  $K$ , at iteration  $p$ , during the *E-step*, the probability of representativeness of each ensemble member is computed with regard to clusters:

$$w_{e,k}^{(p)} \propto \exp \left( -\frac{1}{2} \cdot \left\| \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right\|_{[\mathbf{B}_k^{(p)}]^{-1}}^2 \right), \text{ for } 1 \leq e \leq N \text{ and } 1 \leq k \leq K \tag{11}$$

which allows us to approximate the  $k$ -th weight for the prior mixture component:

$$\alpha_k^{b(p)} = \left( \sum_{r=1}^K \sum_{e=1}^N w_{e,r}^{(p)} \right)^{-1} \cdot \left( \sum_{e=1}^N w_{e,k}^{(p)} \right)$$

GMM components are updated during the M-step:

$$\mathbf{x}_k^{b(p+1)} = \eta_k^{(p)} \cdot \sum_{e=1}^N w_{e,k}^{(p)} \cdot \mathbf{x}^{b[e]} \in \mathbb{R}^{n \times 1}, \text{ and } \mathbf{B}_k^{(p+1)} = \eta_k^{(p)} \cdot \left[ \Delta \mathbf{X}_k^{(p)} \cdot \mathbf{\Gamma}_k^{(p)} \right] \cdot \left[ \Delta \mathbf{X}_k^{(p)} \cdot \mathbf{\Gamma}_k^{(p)} \right]^T \in \mathbb{R}^{n \times n}$$

where  $\eta_k^{(p)} = \left[ \sum_{e=1}^N w_{e,k}^{(p)} \right]^{-1} \in \mathbb{R}$ , the matrix of intra-cluster deviations reads

$$\Delta \mathbf{X}_k^{(p)} = \mathbf{X}^b - \mathbf{x}_k^{b(p)} \cdot \mathbf{1}^T \in \mathbb{R}^{n \times N} \text{ and}$$

$$\mathbf{\Gamma}_k^{(p)} = \mathbf{diag} \left\{ \sqrt{w_{1,k}^{(p)}}, \sqrt{w_{2,k}^{(p)}}, \dots, \sqrt{w_{N,k}^{(p)}} \right\} \in \mathbb{R}^{N \times N}$$

Under linear assumptions, GMM-EnKF formulations exploit the fact that a Gaussian distribution is conjugate of a GMM. Thus, posterior components can be computed as results of performing Kalman-like updates over prior weights, centroids, and covariances as follows:

$$\alpha_k^a \propto \alpha_k^b \cdot \exp \left( -\frac{1}{2} \cdot \left\| \mathbf{y} - \mathcal{H} \left( \mathbf{x}_k^b \right) \right\|_{\mathbf{Q}_k^{-1}}^2 \right), \text{ for } 1 \leq k \leq K \tag{12a}$$

$$\mathbf{x}_k^a = \mathbf{x}_k^b + \mathbf{A}_k \cdot \mathbf{H}^T \cdot \left[ \mathbf{y} - \mathcal{H} \left( \mathbf{x}_k^b \right) \right] \in \mathbb{R}^{n \times 1} \tag{12b}$$

and

$$\mathbf{A}_k = \left[ \mathbf{B}_k^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} \right]^{-1} \in \mathbb{R}^{n \times n} \tag{12c}$$

where  $\mathbf{Q}_k = \mathbf{H} \cdot \mathbf{B}_k \cdot \mathbf{H}^T + \mathbf{R} \in \mathbb{R}^{m \times m}$ , and  $\sum_{k=1}^K \alpha_k^a = 1$ . This computationally friendly property is exploited by filters such as the Non-linear Bayesian Estimator (N-BE) [25] and the Gaussian Mixture Ensemble Filter [26,27]. In [28] the N-BE is enhanced by using the Bayesian Information Criteria (BIC)

$$BIC = -2 \cdot \left( \sum_{e=1}^N \log \left( \sum_{k=1}^K \alpha_k^b \cdot \phi \left( \mathbf{x}^{b[e]}; \mathbf{x}_k^b, \mathbf{B}_k \right) \right) \right) + 2 \cdot (3 \cdot N - 1)$$

in order to choose the number of components for the GMM, where  $\phi \left( \mathbf{x}; \mathbf{x}_k^b, \mathbf{B}_k \right)$  denotes a Normal probability density function with parameters  $\mathbf{x}_k^b$  and  $\mathbf{B}_k$ , respectively. This is possible, as well, by other

means, for instance in the GMM-EnKF filter proposed by Smith in [29], the number of mixture components relies on the Akaike’s Information Criteria (AIC)

$$AIC = -2 \cdot \left( \sum_{e=1}^N \log \left( \sum_{k=1}^K \alpha_k^b \cdot \phi \left( \mathbf{x}^{b[e]}; \mathbf{x}_k^b, \mathbf{B}_k \right) \right) \right) + \log(N) \cdot (3 \cdot N - 1)$$

The BIC and the AIC are common methods for choosing the number of parameters in GMM but, information criteria based methods are poorly suited for selecting a model with a good out-of-sample fit in model-rich environments [30,31]. Besides, these methods are only valid for sample size  $N$  much larger than the number  $K$  of parameters, which can be difficult in the DA context.

Other methods such as Particle Filters (PFs) [32] are available in the literature in order to attack non-Gaussian DA problems. These methods are a good-choice from a theoretical point of view. Unfortunately, in practice, PFs do suffer from a degeneracy problem [33] (Section 1.4) and even more, many challenges have to be overcome before they can be considered under operational DA scenarios [34,35]. For these reasons, PFs are not considered any further in this paper.

### 3. Proposed Method

In this section, we develop an efficient Gaussian mixture ensemble Kalman filter implementation based on Markov-Chain-Monte-Carlo (GM-EnKF-MCMC) for non-Gaussian data assimilation. We describe in detail how hyper-parameters are computed as well as how observations are digested for the GM-EnKF-MCMC. Lastly, we briefly discuss how posterior ensembles are built.

#### 3.1. Estimation of Hyper-Parameters—EM Method

Consider an ensemble of model realizations

$$\mathbf{X}^b = [\mathbf{x}^{b[1]}, \mathbf{x}^{b[2]}, \dots, \mathbf{x}^{b[N]}] \in \mathbb{R}^{n \times N} \tag{13}$$

where  $\mathbf{x}^{b[e]} \in \mathbb{R}^{n \times 1}$  stands for the  $e$ -th ensemble member, for  $1 \leq e \leq N$ . To estimate the hyper-parameters of the prior error distribution, we use the EM method to fit a GMM with  $K$  components. Instead of estimating background error covariances  $\mathbf{B}_k$ , for  $1 \leq k \leq K$ , for the mixture components, we do prefer to estimate precision covariances  $\mathbf{B}_k^{-1}$  due to their computational benefits. For instance, in the context of Normal distributions, probabilities are computed based on precision covariances which would require the inversion of large matrices when background error covariances are fitted during EM steps. Besides, by using the modified Cholesky decomposition for computing the precision covariances, we can exploit their special structures in order to obtain huge savings in terms of memory usage and to reduce the computational cost of matrix-vector products among iterations.

The proposed EM method works as follows: for a given number of clusters  $K$ , we choose  $K$  random members from the ensemble (13) in order to set the initial mixture centroids while the initial precision covariances are all equal to the precision covariance of the ensemble (8a). During the E-step, we compute the representativeness of each ensemble member regarding each cluster

$$\hat{w}_{e,k}^{(p)} \propto \exp \left( -\frac{1}{2} \cdot \left\| \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right\|_{[\hat{\mathbf{B}}_k^{(p)}]^{-1}}^2 \right), \text{ for } 1 \leq e \leq N \text{ and } 1 \leq k \leq K \tag{14a}$$

where

$$[\hat{\mathbf{B}}_k^{(p)}]^{-1} = [\mathbf{L}_k^{(p)}]^T \cdot \mathbf{D}_k^{(p)} \cdot [\mathbf{L}_k^{(p)}] \in \mathbb{R}^{n \times n} \tag{14b}$$

The M-step updates the precision covariances as well as the background error states for each cluster. Consider the diagonal matrix of fuzzy weights,

$$\hat{\mathbf{\Gamma}}_k^{(p)} = \mathbf{diag} \left\{ \hat{w}_{1,k}^{(p)}, \hat{w}_{2,k}^{(p)}, \dots, \hat{w}_{N,k}^{(p)} \right\} \in \mathbb{R}^{N \times N} \tag{15a}$$

the GMM centroids are updated as follows:

$$\bar{\mathbf{x}}_k^{b(p)} = \hat{\eta}_k^{(p)} \cdot [\mathbf{X}^b \cdot \hat{\Gamma}_k^{(p)}] \in \mathbb{R}^{n \times 1} \tag{15b}$$

where  $\hat{\eta}_k^{(p)} = [\sum_{e=1}^N \hat{w}_{e,k}^{(p)}]^{-1} \in \mathbb{R}$ . Likewise, the non-zero elements of  $\mathbf{L}_k^{(p+1)} \in \mathbb{R}^{n \times n}$  (14b) are revised by fitting models of the form:

$$[\mathbf{x}_{[i]k}^{(p)}]^T = \sum_{q \in P(i,r)} [\mathbf{x}_{[q]k}^{(p)}]^T \cdot \{\mathbf{L}_k^{(p+1)}\}_{i,q} + \gamma_{i,k}, \text{ for } 1 \leq i \leq n \text{ and } 1 \leq k \leq K \tag{15c}$$

where  $\mathbf{x}_{[i]k}^{(p)} \in \mathbb{R}^{1 \times N}$  is the  $i$ -th row of matrix  $\Delta \mathbf{X}_k^{(p)} \in \mathbb{R}^{n \times N}$

$$\Delta \mathbf{X}_k^{(p)} = \mathbf{X}^b - \bar{\mathbf{x}}_k^{(p)} \cdot \mathbf{1}^T \tag{15d}$$

errors  $\gamma_{i,k} \in \mathbb{R}^{N \times 1}$  are assumed Normal distribution with zero mean and uncorrelated components of variance  $\sigma^2$ . Moreover, the diagonal entries of  $\mathbf{D}_k^{(p)} \in \mathbb{R}^{n \times n}$  are estimated via the residuals of model (15c):

$$\{\mathbf{D}_k^{(p)}\}_{i,i} = \mathbf{var} \left( [\mathbf{x}_{[i]k}^{(p)}]^T - \sum_{q \in P(i,r)} [\mathbf{x}_{[q]k}^{(p)}]^T \cdot \{\mathbf{L}_k^{(p+1)}\}_{i,q} \right)^{-1}, \text{ for } 2 \leq i \leq n, \text{ and } 1 \leq k \leq K \tag{15e}$$

with  $\{\mathbf{D}_k^{(p)}\}_{1,1} = \mathbf{var} (\mathbf{x}_{[1]k}^{(p)})^{-1}$ . Note that whenever covariance inflation is desired before assimilation steps, the inflation factors can be applied to the matrix of member deviations with regard to centroids (15d). In this manner, the estimated precision covariances come already inflated. Once the EM steps are concluded (i.e., when a maximum number of iterations is reached), the prior error distribution is described by the GMM:

$$\mathbf{x} \sim \sum_{k=1}^K \alpha_k^b \cdot \left( \bar{\mathbf{x}}_k^b \cdot [\mathbf{L}_k^T \cdot \mathbf{D}_k \cdot \mathbf{L}_k]^{-1} \right) \tag{16}$$

The special structure of the resulting precision covariances  $\hat{\mathbf{B}}_k^{-1} = \mathbf{L}_k^T \cdot \mathbf{D}_k \cdot \mathbf{L}_k$  can be exploited in order to reduce the computational effort of drawing samples from the posterior error distribution as is discussed in the next section.

### 3.2. Sampling Method—Approaching the Posterior

In order to approximate samples from the posterior distribution (4), for each mixture component  $1 \leq k \leq K$  in (16), we use a Markov-Chain-Monte-Carlo (MCMC) approximation. Traditionally, Normal distributions are good candidates for proposing states, for instance, in our context, starting with  $u = 0$  and  $\mathbf{x}^{(u)} = \bar{\mathbf{x}}_k^b \in \mathbb{R}^{n \times 1}$ , at iteration  $0 \leq u \leq v$ , where  $v$  is a user-defined number of iterations, a state can be proposed as follows:

$$\mathbf{z}^{(u)} \sim \mathcal{N} \left( \mathbf{x}^{(u)}, [\hat{\mathbf{B}}_k^{-1}]^{-1} \right)$$

Or simply,

$$\mathbf{z}^{(u)} = \mathbf{x}^{(u)} + \boldsymbol{\nu} \in \mathbb{R}^{n \times 1} \tag{17}$$

where  $\boldsymbol{\nu} \in \mathbb{R}^{n \times 1}$  is given by the solution of an upper triangular linear system:

$$[\mathbf{L}^T \cdot \mathbf{D}^{1/2}] \cdot \boldsymbol{\nu} = \boldsymbol{\epsilon} \in \mathbb{R}^{n \times 1}$$

and  $\epsilon \in \mathbb{R}^{n \times 1}$  follows a standard Normal distribution. Nevertheless, in high-dimensional spaces such as those found in the context of operational DA, high-probability zones of posterior error distributions can be reached after a huge number of iterations. Thus, to overcome this situation, we proceed as follows: we linearize the observation operator about the current state  $\mathbf{x}^{(u)}$ ,

$$\mathcal{H}(\mathbf{x}) \approx \mathcal{G}_u(\mathbf{x}) = \mathcal{H}(\mathbf{x}^{(u)}) + \mathbf{H}_{\mathbf{x}^{(u)}} \cdot [\mathbf{x} - \mathbf{x}^{(u)}]$$

where  $\mathbf{H}_{\mathbf{x}^{(u)}} \in \mathbb{R}^{m \times n}$  is the Jacobian matrix of  $\mathcal{H}(\mathbf{x})$  at  $\mathbf{x}^{(u)}$ , the cost function (6) can then be approximated by the quadratic cost function:

$$\hat{\mathcal{J}}_k(\mathbf{x}) = \frac{1}{2} \cdot \|\mathbf{x} - \bar{\mathbf{x}}_k^b\|_{\mathbf{B}_k^{-1}}^2 + \frac{1}{2} \cdot \|\mathbf{y} - \mathcal{G}_u(\mathbf{x})\|_{\mathbf{R}^{-1}}^2 \tag{18a}$$

whose gradient reads,

$$\nabla \hat{\mathcal{J}}_k(\mathbf{x}) = \hat{\mathbf{B}}_k^{-1} \cdot [\mathbf{x} - \bar{\mathbf{x}}_k^b] - \mathbf{H}_{\mathbf{x}^{(u)}}^T \cdot \mathbf{R}^{-1} \cdot [\mathbf{y} - \mathcal{G}_u(\mathbf{x})] \in \mathbb{R}^{n \times 1} \tag{18b}$$

By using this gradient, the proposal distribution (17) can be modified in such a manner that, samples along the (approximated) steepest descent direction  $-\nabla \hat{\mathcal{J}}_k(\mathbf{x}^{(u)})$  of (6) have high probability of occurrence, that is,

$$\mathbf{z}^{(u)} = \mathbf{x}^{(u)} + \psi \cdot \left[ -\frac{\nabla \hat{\mathcal{J}}_k(\mathbf{x}^{(u)})}{\|\nabla \hat{\mathcal{J}}_k(\mathbf{x}^{(u)})\|} \right], \text{ with } \psi \sim \mathcal{U}(0, \beta) \tag{19}$$

where  $\mathcal{U}(0, \beta)$  stands for an Uniform distribution on the interval  $(0, \beta)$ . The value of  $\beta$  can be tuned according to the degree of the observation operator. For instance, for a linear observation operator,  $\beta$  can be set as  $\|\nabla \hat{\mathcal{J}}(\mathbf{x}^{(u)})\|$  since the gradient of (6) and (18a) are the same. When observation operators are highly non-linear, since Taylor based approximations suffer from myopia, a small step of such gradient must be taken and therefore, a good choice under this consideration is 1. Thus, an intuitive range for  $\beta$  can be

$$\beta \in [1, \|\nabla \hat{\mathcal{J}}(\mathbf{x}^{(u)})\|]$$

In the absence of prior information about  $\beta$ , one can choose 1. The main motivation for using gradient approximations is that, the subset of samples along the descent direction (19) can provide states which potentially maximize the posterior probability. Computationally speaking, no matrix inversion is needed in this context in order to propose states. The acceptance/rejection rule for the states (19) rely on the Metropolis-Hastings criterion which can be adapted as follows:

$$\mathbf{x}^{(u+1)} = \begin{cases} \mathbf{x}^{(u)} & , y > \frac{\mathcal{J}_k(\mathbf{x}^{(u)})}{\mathcal{J}_k(\mathbf{z}^{(u)})}, \text{ for } y \sim \mathcal{U}(0, 1) \\ \mathbf{z}^{(u)} & , \text{ otherwise} \end{cases} \tag{20}$$

The observation operator is then linearized about  $\mathbf{x}^{(u+1)}$  and the overall process is repeated until some number of iterations is satisfied (or any other numerical condition). Putting it all together, the sampling procedure to compute the posterior modes is in the following:

- Step 1 Let  $k = 1$ , set  $u = 0$ , go to step 2.
- Step 2 Set  $\mathbf{x}^{(u)} = \bar{\mathbf{x}}_k^b$ .
- Step 3 Linearize  $\mathcal{H}$  about  $\mathbf{x}^{(u)}$  and compute the direction (18b).
- Step 4 Compute  $\mathbf{z}^{(u)}$  via Equation (19).

- Step 5 Set  $\mathbf{x}^{(u+1)}$  according to Equation (20).
- Step 6 If  $u \leq v$  set  $u = u + 1$  and go to step 3, set  $\bar{\mathbf{x}}_k^a = \mathbf{x}^{(v)}$  and go to step 7 otherwise.
- Step 7 If  $k \leq K$  go to step 1, go to step 8 otherwise.
- Step 8 The posterior mode approximations read  $\{\bar{\mathbf{x}}_k^a\}_{k=1}^K$ .

Note that, unlike MCMC methods based on Random-Walk, for each prior mode, this procedure does not return a chain but only the last state in  $\mathbf{x}^{(v)}$ . This sampling step can be replaced by an optimization method such as Trust Region [36–38] or Line Search [39,40]. However, in order for it to work, some regularities must be satisfied by the gradient approximation (18b) [41–45], for instance, smoothness, which in practice is not necessarily the case.

### 3.3. Building the Posterior Ensemble

Once the posterior modes are computed, posterior covariances within clusters can be estimated as follows:

$$\hat{\mathbf{A}}_k = \left[ \hat{\mathbf{B}}_k^{-1} + \mathbf{H}_{\bar{\mathbf{x}}_k^a} \cdot \mathbf{R}^{-1} \cdot \left[ \mathbf{H}_{\bar{\mathbf{x}}_k^a} \right]^T \right]^{-1} \in \mathbb{R}^{n \times n} \tag{21a}$$

Therefore, the analysis members can be computed as:

$$\mathbf{x}^{a[e]} \sim \mathcal{N} \left( \bar{\mathbf{x}}_k^a, \hat{\mathbf{A}}_k \right), \text{ with probability } \alpha_k^a, \text{ for } 1 \leq e \leq N \tag{21b}$$

where  $\alpha_k^a$  is estimated via the likelihood ratio:

$$\alpha_k^a = \left[ \sum_{j=1}^K \phi \left( \mathbf{y}; \mathcal{H} \left( \mathbf{x}_j^a \right), \mathbf{R} \right) \right]^{-1} \cdot \phi \left( \mathbf{y}; \mathcal{H} \left( \mathbf{x}_k^a \right), \mathbf{R} \right)$$

### 3.4. Computational Complexity

In this section, we estimate the number of long computations of the proposed method in order to assess its computational effort. We detail the number of computations of the GMM-EnKF-MCMC below, we avoid the use of iteration indexes for ease of reading:

1. During the E-Step, the computations of weights (14a) depend on the calculation:

$$\begin{aligned} -\frac{1}{2} \cdot \left\| \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right\|_{\left[ \hat{\mathbf{B}}_k^{(p)} \right]^{-1}}^2 &= -\frac{1}{2} \cdot \left[ \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right]^T \cdot \hat{\mathbf{B}}_k^{-1} \left[ \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right] \\ &= -\frac{1}{2} \cdot \left[ \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right]^T \cdot \mathbf{L}_k^T \cdot \mathbf{D}_k \cdot \mathbf{L}_k \cdot \left[ \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right] \\ &= -\frac{1}{2} \cdot \underbrace{\left[ \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right]^T}_{\mathbf{s}_k} \cdot \mathbf{L}_k^T \cdot \mathbf{D}_k^{T/2} \cdot \mathbf{D}_k^{1/2} \cdot \mathbf{L}_k \cdot \underbrace{\left[ \mathbf{x}^{b[e]} - \mathbf{x}_k^{b(p)} \right]}_{\mathbf{s}_k} \\ &= -\frac{1}{2} \cdot \mathbf{s}_k^T \cdot \mathbf{L}_k^T \cdot \mathbf{D}_k^{T/2} \cdot \mathbf{D}_k^{1/2} \cdot \mathbf{L}_k \cdot \mathbf{s}_k \\ &= \left[ \mathbf{D}_k^{1/2} \cdot \underbrace{\mathbf{L}_k \cdot \mathbf{s}_k}_{\hat{\mathbf{s}}_k} \right]^T \cdot \mathbf{D}_k^{1/2} \cdot \underbrace{\mathbf{L}_k \cdot \mathbf{s}_k}_{\hat{\mathbf{s}}_k} \\ &= \left[ \underbrace{\mathbf{D}_k^{1/2} \cdot \hat{\mathbf{s}}_k}_{\tilde{\mathbf{s}}_k} \right]^T \cdot \underbrace{\mathbf{D}_k^{1/2} \cdot \hat{\mathbf{s}}_k}_{\tilde{\mathbf{s}}_k} = \tilde{\mathbf{s}}_k^T \cdot \tilde{\mathbf{s}}_k = \|\tilde{\mathbf{s}}_k\|_2^2 \end{aligned}$$

From this step, given the special structure of  $\mathbf{L}_k$ ,  $\tilde{\mathbf{s}}_k \in \mathbb{R}^{n \times 1}$  can be computed with no more than  $\mathcal{O}(\theta^2 \cdot n)$  long computations where  $\theta$  denotes the maximum number of non-zero elements across all rows in  $\mathbf{L}_k$  with  $\theta \ll n$ . Likewise, the number of long computations in order to obtain  $\tilde{\mathbf{s}} \in \mathbb{R}^{n \times 1}$  is bounded by  $\mathcal{O}(n)$  since  $\mathbf{D}_k$  is diagonal. Thus, since there are  $K$  clusters, each E-step has the following operation counting:

$$\mathcal{O}\left(K \cdot \left[n + \theta^2 \cdot n\right]\right) \tag{22a}$$

- During the M-step, updating the centroids (15b) can be performed with no more than  $\mathcal{O}(N^2 \cdot n)$  since  $\mathbf{D}_k$  has only  $n$  components different from zero (the diagonal ones), the least square solution of (15c) is bounded by  $\mathcal{O}(\theta^2 \cdot n)$  calculations since there are  $n$  model components, and the cost of (15e) is bounded by  $\mathcal{O}(n \cdot \theta^2)$  since the multiplication of coefficients and model components is constrained to the neighbourhood of each model component. The computational effort of this method is then estimated as follows:

$$\mathcal{O}\left(K \cdot \left[\theta^2 \cdot n + N^2 \cdot n\right]\right) \tag{22b}$$

- During the sampling procedure, the gradient (18b) can be efficiently computed as follows:

$$\begin{aligned} \nabla \hat{\mathcal{J}}_k(\mathbf{x}) &= \hat{\mathbf{B}}_k^{-1} \cdot \underbrace{\left[ \mathbf{x} - \bar{\mathbf{x}}_k^b \right]}_{\mathbf{g}_k} - \mathbf{H}_{\mathbf{x}^{(u)}}^T \cdot \mathbf{R}^{-1} \cdot \underbrace{\left[ \mathbf{y} - \mathcal{G}_u(\mathbf{x}) \right]}_{\mathbf{f}_k} = \mathbf{L}_k^T \cdot \mathbf{D}_k \cdot \underbrace{\mathbf{L}_k \cdot \mathbf{g}_k}_{\hat{\mathbf{g}}_k} - \mathbf{H}_{\mathbf{x}^{(u)}}^T \cdot \mathbf{R}^{-1} \cdot \underbrace{\mathbf{f}_k}_{\hat{\mathbf{f}}_k} \\ &= \mathbf{L}_k^T \cdot \underbrace{\mathbf{D}_k \cdot \hat{\mathbf{g}}_k}_{\tilde{\mathbf{g}}_k} - \underbrace{\mathbf{H}_{\mathbf{x}^{(u)}}^T \cdot \hat{\mathbf{f}}_k}_{\tilde{\mathbf{f}}_k} = \mathbf{L}_k^T \cdot \tilde{\mathbf{g}}_k - \tilde{\mathbf{f}}_k = \bar{\mathbf{g}}_k - \mathbf{f}_k \end{aligned}$$

Given the special structure of  $\mathbf{L}_k$ ,  $\mathbf{g}_k \in \mathbb{R}^{n \times 1}$  and  $\bar{\mathbf{g}}_k \in \mathbb{R}^{n \times 1}$  can be computed with no more than  $\mathcal{O}(\theta^2 \cdot n)$ , the computations in  $\tilde{\mathbf{g}}_k \in \mathbb{R}^{n \times 1}$  are bounded by  $\mathcal{O}(n)$  since  $\mathbf{D}_k$  is diagonal. Thus, since this sampling method is performed  $v$  times, the computational effort of the sampling procedure reads,

$$\mathcal{O}\left(v \cdot \left[n + \theta^2 \cdot n\right]\right) \tag{22c}$$

- The posterior ensemble can be built (Section 3.3 [19]) with no more than

$$\mathcal{O}\left(n \cdot \theta^2 + m \cdot N\right), \tag{22d}$$

long computations.

Assuming that the number of clusters and the number of iterations in the sampling process are much lower than the model dimension, based on Equation (22) the computational effort of the GMM-EnKF-MCMC reads,

$$\mathcal{O}\left(n \cdot \theta^2 + m \cdot N\right)$$

which is linear with regard to the model resolution  $n$ .

### 3.5. Comparison of GM-EnKF-MCMC with Other Sampling Methods

In this section, we briefly compare the GM-EnKF-MCMC method with well-known filters from the literature: the Cluster Sampling Filter (CSF) [21], the Cluster Monte Carlo Implementation (CMCI) [46], and the Cluster Ensemble Kalman Filter (CEnKF) [29].

The CSF exploits the evolution of a system under Hamiltonian dynamics, for instance, based on Newton’s law it is possible to describe the dynamics of particles. Each particle is fully described by two components: the position and the velocity coordinates which are associated with model states

$\mathbf{x} \in \mathbb{R}^{n \times 1}$  and momentums  $\mathbf{p} \in \mathbb{R}^{n \times 1}$  (auxiliary variables), respectively. The performance of CSF relies on the proper choice of a mass matrix  $\mathbf{M} \in \mathbb{R}^{n \times n}$  in order to describe the probability distribution of  $\mathbf{p}$  during the sampling process. Besides, numerical integrators which preserve Liouville’s Theorem are a must in order to obtain a proposal state  $(\mathbf{p}^*, \mathbf{x}^*)$ . Some of those are the Verlet integrator [47] and the Leap-frog scheme [48]. A main difference between the GMM-EnKF-MCMC and the CSF is that, no auxiliary  $n$ -th dimensional vectors are required in our method during the sampling process. This can be convenient, for instance, under current operational DA systems. Besides, during the sampling process, the only parameter to be tuned in the GMM-EnKF-MCMC is  $\beta$  in (17) while in the CSF method, as we mentioned before, proper choices of  $\mathbf{M}$  are required in order to speed-up the convergence of MCMC towards high-probability zones of the posterior. Besides, extra parameters may need to be tuned depending on the numerical integrator chosen in order to propose states during the sampling process.

In the CMCI, the background error distribution is approximated by a summation of Gaussian Kernels. These are selected using the method of Fukunaga [49] to form a continuous approximation to the random ensemble. By assuming linear observation operators, the posterior ensemble is Gaussian as well. Posterior members are drawn by sampling each of the components/kernels based on a set of calculated weights which account for the observation. As is pointed out by the authors, the CMCI is a very promising filter but, additional work is needed in order to extend its capabilities to more realistic scenarios. Moreover, this filter has been developed under Gaussian assumptions on the observations; such assumption is not required in the GMM-EnKF-MCMC.

The CEnKF is developed by assuming linear observation operators. Prior error distributions are described by GMM and its components are fitted by using the EM-method. This filter can fail to obtain reasonable estimates of posterior ensembles when non-linear observation operators are present during the assimilation of observations, as is typical in practice.

#### 4. Experimental Settings

In this section, numerical tests are performed to assess the accuracy of the proposed filter. We use the Lorenz-96 model [50] as our surrogate model. The Lorenz-96 model is described by the following set of ordinary differential equations [51]:

$$\frac{dx_j}{dt} = \begin{cases} (x_2 - x_{n-1}) \cdot x_n - x_1 + F & \text{for } j = 1 \\ (x_{j+1} - x_{j-2}) \cdot x_{j-1} - x_j + F & \text{for } 2 \leq j \leq n - 1 \\ (x_1 - x_{n-2}) \cdot x_{n-1} - x_n + F & \text{for } j = n \end{cases} \quad (23)$$

where  $F$  is external force and  $n = 40$  is the number of model components. Periodic boundary conditions are assumed. When  $F = 8$  the model exhibits chaotic behavior, which makes it a relevant surrogate problem for atmospheric dynamics [52,53]. One time unit in the Lorenz-96 represents 7 days in real case. The experimental settings are described below:

- Starting with an initial random solution, a 4th order Runge Kutta method is employed in order to integrate it over a long time period from which initial condition  $\mathbf{x}_{-2}^* \in \mathbb{R}^{n \times 1}$  is obtained.
- A perturbed background solution  $\tilde{\mathbf{x}}_{-2}^b$  is formed at time  $t_{-2}$  by drawing a sample from the Normal distribution,

$$\tilde{\mathbf{x}}_{-2}^b \sim \mathcal{N}(\mathbf{x}_{-2}^*, 0.05^2 \cdot \mathbf{I})$$

This solution is then integrated for 10 time units (equivalent to 70 days) in order to obtain a background solution  $\mathbf{x}_{-1}^b$  consistent with the dynamics of the numerical model.

- An initial perturbed ensemble is built about the background state by taking samples from the distribution,

$$\hat{\mathbf{x}}_{-1}^{b[\hat{e}]} \sim \mathcal{N}(\mathbf{x}_{-1}^b, 0.05^2 \cdot \mathbf{I}), \text{ for } 1 \leq \hat{e} \leq \hat{N}$$

In order to make them consistent with the model dynamics, the ensemble members are propagated for 10 time units, from which the initial ensemble members are obtained. We create the initial pool  $\widehat{\mathbf{X}}_0^b$  of  $\hat{N} = 10^4$  members. The actual solution is integrated over 20 more time units in order to place it at the beginning of the assimilation window.

- Two assimilation windows are proposed for the tests, both of them consist of  $M = 15$  observations. In the first assimilation window, observations are taken every 16 h (time step of 0.1 time units) while in the last one, observations are available every 50 h (time step of 0.3 time units). We denote by  $\delta t \in \{16, 50\}$  the elapsed time between two observations.
- The observational errors are described by the probability distribution,

$$\mathbf{y}_\ell \sim \mathcal{N}(\mathcal{H}_k(\mathbf{x}_\ell^*), [\epsilon^o]^2 \cdot \mathbf{I}), \text{ for } 1 \leq \ell \leq M$$

where the standard deviations of observational errors  $\epsilon^o = 10^{-2}$ , and  $\ell$  should be interpreted as time index. Random observation networks are formed at the different assimilation cycles. The space between observations will depend on the step size, for instance, in the first step size observations are available every 0.1 time units (16 h) while in the last one, observations are taken every 0.3 time units (50 h).

- We consider the non-linear observation operator [32]:

$$\{\mathcal{H}(\mathbf{x})\}_j = \frac{\{\mathbf{x}\}_j}{2} \cdot \left[ \left( \frac{|\{\mathbf{x}\}_j|}{2} \right)^{\gamma-1} + 1 \right] \tag{24}$$

where  $j$  denotes the  $j$ -th observed component from the model state.  $\gamma \in \{1, 3, 5\}$ .

- We consider two percentages of observed components  $s$  from the model state  $s \in \{70\%, 100\%\}$ .
- The radius of influence is set to  $r = 1$  while the inflation factor is set to 1.02 (a typical value).
- We propose two ensemble sizes for the benchmark  $N \in \{20, 80\}$ . These members are randomly chosen from the pool  $\widehat{\mathbf{X}}_0^b$  for different experiments in order to form the initial ensemble  $\mathbf{X}_0^b$  for the assimilation window. Evidently,  $\mathbf{X}_0^b \subset \widehat{\mathbf{X}}_0^b$ .
- The  $L - 2$  norm of errors are utilized as a measure of accuracy at the assimilation step  $\ell$ ,

$$\lambda_\ell = \sqrt{[\mathbf{x}_\ell^* - \mathbf{x}_\ell^a]^T \cdot [\mathbf{x}_\ell^* - \mathbf{x}_\ell^a]} \tag{25}$$

where  $\mathbf{x}_k^*$  and  $\mathbf{x}_k^a$  are the reference and the analysis solutions, respectively. The analysis state is obtained by a weighted combination of posterior centroids via the likelihood ratio (21b) in lieu of the posterior mean.

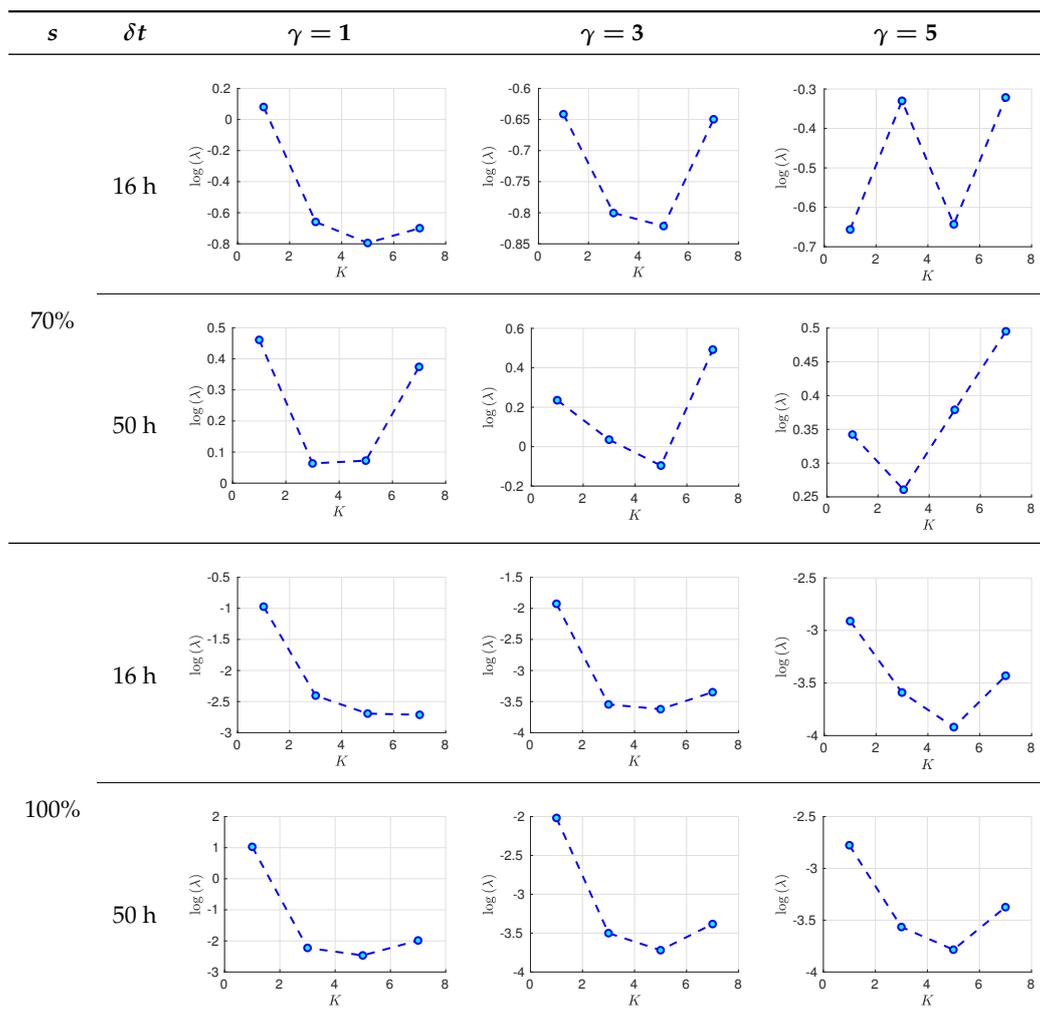
- The Root-Mean-Square-Error (RMSE) is used as a measure of performance. On average, on a given assimilation window,

$$\lambda = \sqrt{\frac{1}{M} \cdot \sum_{\ell=1}^M \lambda_\ell^2} \tag{26}$$

Results for  $\delta t = 16$  h, different ensemble sizes, and different values of  $\gamma$  are shown in Figure 1 in terms of L-2 error norms  $\lambda_\ell$  per assimilation step (25). Notoriously, the performance of the filter is improved as long as prior errors are described by a GMM ( $K > 1$ ). This can be expected owing to the non-linear dynamics of the Lorenz-96 model. In addition, the performance of the GMM-EnKF-MCMC is not impacted by the degree  $\gamma$  of the observation operator. Both features make the proposed formulation attractive for DA systems where non-linear observation operators are the bridge for

digesting observations and besides, non-linear dynamics are encapsulated in the numerical model. Moreover, for  $K > 1$ , there are less posterior errors than in prior and observations; this of course must be expected for full observational networks. In some cases, filter divergence is possible for  $K = 1$  (a single mode prior distribution). Assuming that spurious correlations are not impacting the analysis corrections, in this set of experiments, filter divergence can be seen as a consequence of one of two things: prior errors are not well fitted by a Gaussian distribution (the actual error distribution is multimodal) or inflation factors are not properly tuned. Regardless of the main cause, filter convergence is evident when a GMM describes the background error distribution. In Figure 2, results for  $\delta t = 50$ , different ensemble sizes, and different values of  $\gamma$  are shown as well. Under this configuration, the accuracy of the GMM-EnKF-MCMC is slightly impacted by increments in  $\gamma$ . Filter convergence is rapidly achieved when the number of prior modes is larger than one. Despite the fact that filter divergence can be still for  $K = 1$ , interestingly, the proposed method does a reasonable estimation of the actual state of the system. In the linear case, this is not surprising, EnKF formulations are widely utilized in practice wherein Gaussian assumptions are commonly broken during assimilation steps. For non-linear observation operators  $\gamma > 1$ , and Gaussian assumptions on the prior  $K = 1$ , the accuracy of the filter relies on the sampling procedure. As can be seen, under such assumptions, the proposed method is capable of obtaining good estimates of posterior error modes.

**Table 1.** Experimental results with the Lorenz-96 model. Mean of RMSE values are reported across the 10 different experimental configurations for different configuration of parameters  $\delta t$ ,  $s$ ,  $K$ , and  $\gamma$ . The number of ensemble members equals  $N = 20$ .



RMSE values ( $\lambda$ ) (26) in the log-scale are shown in the Tables 1 and 2 for  $N = 20$  and  $N = 80$ , respectively. We group RMSE values by elapsed time between observations  $\delta t$  and number of observed components  $s$ . Notice that the performance of the filter can be improved when the number of prior modes is larger than one. This aligns with the highly non-linear (and chaotic) dynamics exhibited by the Lorenz 96 model which makes a single mode distribution insufficient to encapsulate all prior information. However, the GMM-EnKF-MCMC can be impacted when a large number of mixture components is utilized and in some cases, overfitting is possible, for instance, in Table 1, for  $\gamma = 5$  and  $N = 20$ .

Yet another interesting analysis is the behaviour of the sampling process. In Figure 3, we show a two dimensional projection of our sampling steps for some particular choices of parameters  $\gamma$ ,  $N$ ,  $s$ , and  $K$ . We use the two leading components of the space generated by the accepted samples among iterations. Note that, in all cases, the sampling method obtains good estimates of the actual state of the system  $\mathbf{x}^*$ , even more, assuming  $N = K$  (which is equivalent to assume Gaussian errors for each ensemble member), the sampling steps converges to  $\mathbf{x}^*$ .

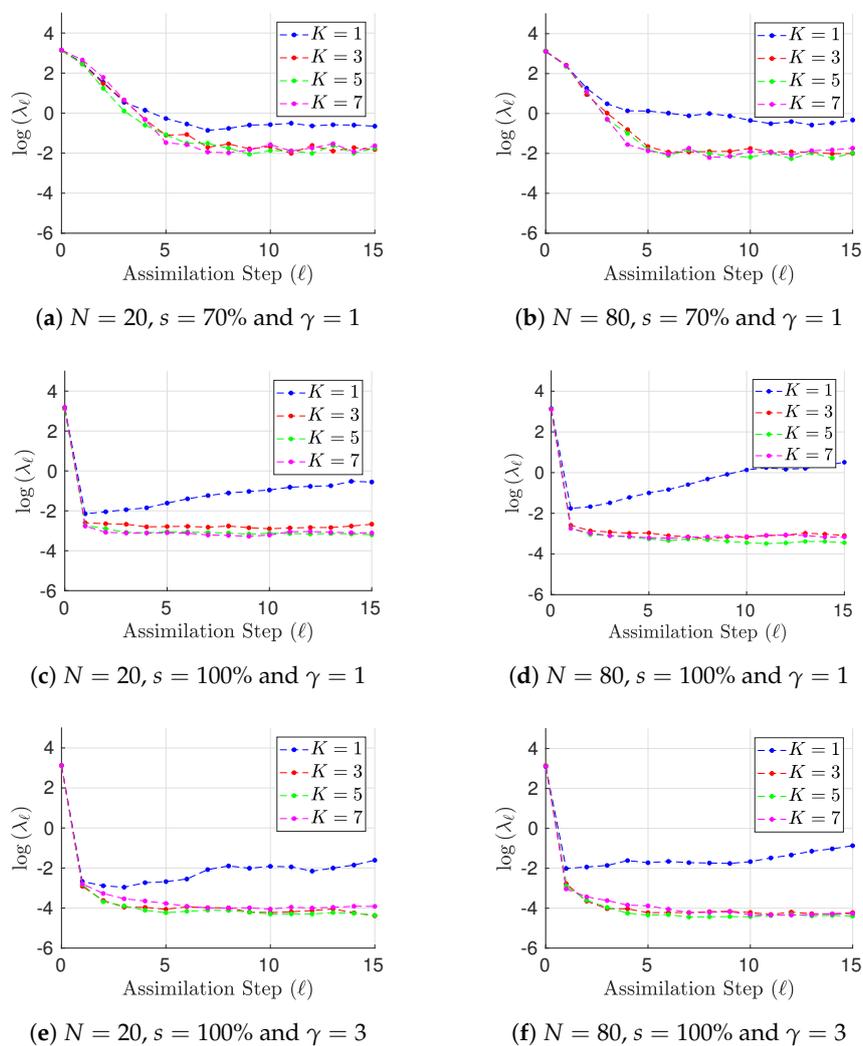
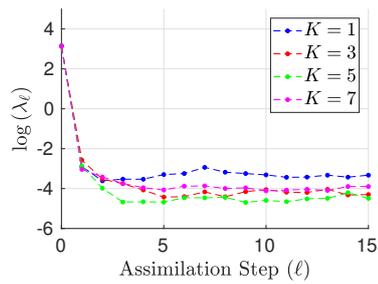
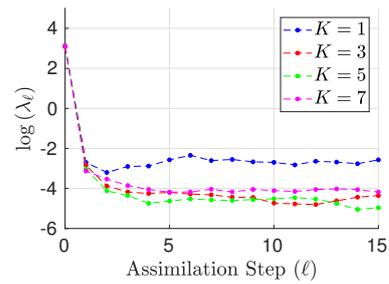


Figure 1. Cont.

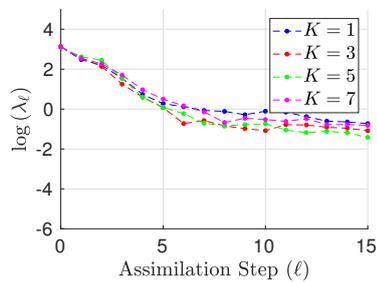


(g)  $N = 20, s = 100\%$ , and  $\gamma = 5$

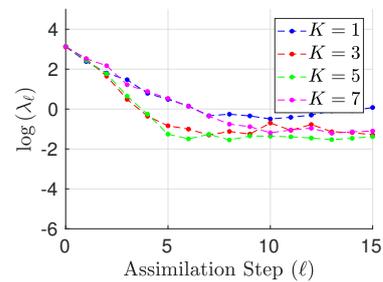


(h)  $N = 80, s = 100\%$ , and  $\gamma = 5$

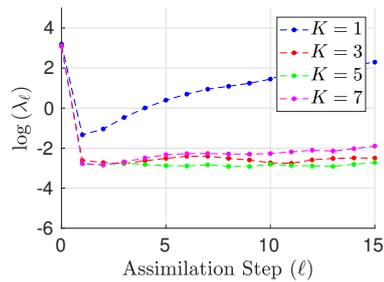
**Figure 1.** Experimental results with the Lorenz-96 model (29). The time evolution of mean analysis errors and their standard deviation across the 10 different experimental configurations are reported for  $\gamma \in \{1, 3, 5\}$ , and  $\delta t = 16$ .



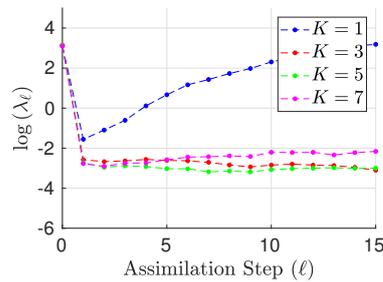
(a)  $N = 20, s = 70\%$  and  $\gamma = 1$



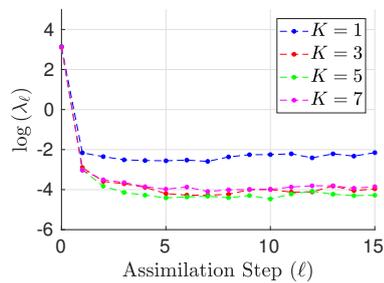
(b)  $N = 80, s = 70\%$  and  $\gamma = 1$



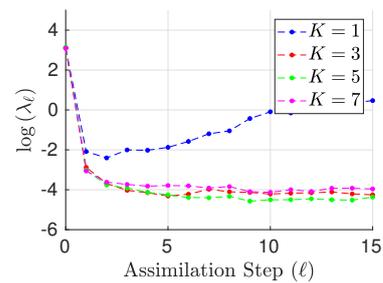
(c)  $N = 20, s = 100\%$  and  $\gamma = 1$



(d)  $N = 80, s = 100\%$  and  $\gamma = 1$

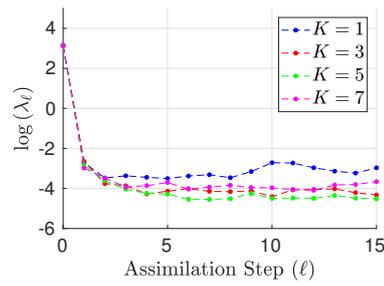


(e)  $N = 20, s = 100\%$  and  $\gamma = 3$

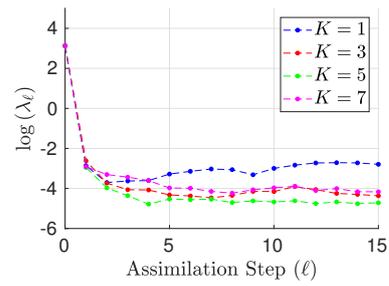


(f)  $N = 80, s = 100\%$  and  $\gamma = 3$

**Figure 2.** Cont.



(g)  $N = 20, s = 100\%$ , and  $\gamma = 5$

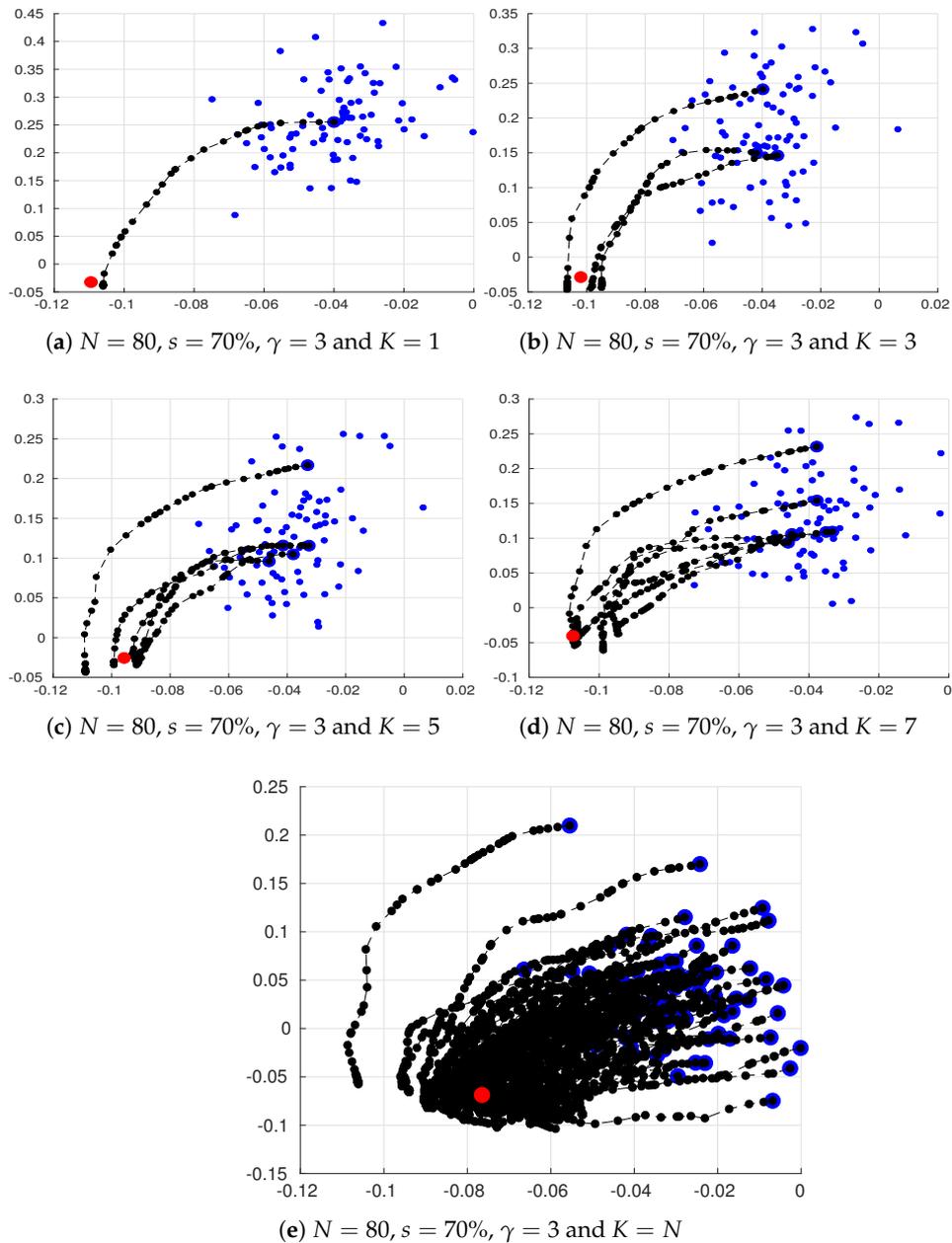


(h)  $N = 80, s = 100\%$ , and  $\gamma = 5$

**Figure 2.** Experimental results with the Lorenz-96 model (29). The time evolution of mean analysis errors and their standard deviation across the 10 different experimental configurations are reported for  $\gamma \in \{1, 3, 5\}$ , and  $\delta t = 50$ .

**Table 2.** Experimental results with the Lorenz-96 model. Mean of RMSE values are reported across the 10 different experimental configurations for different configuration of parameters  $\delta t, s, K$ , and  $\gamma$ . The number of ensemble members equals  $N = 80$ .

$s$	$\delta t$	$\gamma = 1$	$\gamma = 3$	$\gamma = 5$
70%	16 h			
	50 h			
100%	16 h			
	50 h			



**Figure 3.** Two dimensional projections of the steps performed by the sampling method. Blue dots denote prior ensemble members  $x^{b[e]}$ , large blue dots stand for centroids  $\bar{x}_k^b$ , dashed black lines together with black dots denote accepted states  $x^{(u)}$ , and the red dot stands for the actual state of the system  $x^*$ . Even for  $K = N$ , the method is able to obtain reasonable estimates of the actual state of the system. The variance explained in such plots is larger than 90%.

### 5. Conclusions

In this paper we propose an EnKF implementation based on a modified Cholesky decomposition and a Markov-Chain-Monte-Carlo (MCMC) Method, the GMM-EnKF-MCMC. During the assimilation of observations, the method proceeds as follows: prior error distributions are fitted by using Gaussian Mixture Models. Prior components are estimated making use of the Expectation Maximization algorithm. Based on MCMC, posterior components are individually estimated. This is done by sampling states along an approximated steepest descent direction of the well-known Three Dimensional Variational cost function. Experimental tests are performed on the Lorenz-96 model.

The results reveal that the proposed method is able to handle non-linearities produced by the model dynamics as well as those generated by the non-linear observation operator and even more, in a root-mean-square-error sense, the accuracy of the filter is not highly impacted as the degree of the observation operator is increased.

**Acknowledgments:** This work was supported by the Applied Math and Computer Science Laboratory at Universidad del Norte, Barranquilla, COL.

**Author Contributions:** Elias D. Nino-Ruiz and Haiyan Cheng conceived and designed the experiments; Elias D. Nino-Ruiz and Rolando Beltran performed the experiments; Elias D. Nino-Ruiz and Haiyan Cheng analyzed the data; Rolando Beltran contributed analysis tools; Elias D. Nino-Ruiz and Haiyan Cheng wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hoar, T.; Anderson, J.; Collins, N.; Kershaw, H.; Hendricks, J.; Raeder, K.; Mizzi, A.; Barré, J.; Gaubert, B.; Madaus, L.; et al. *DART: A Community Facility Providing State-Of-The-Art, Efficient Ensemble Data Assimilation for Large (Coupled) Geophysical Models*; AGU: Washington, DC, USA, 2016.
2. Clayton, A.; Lorenc, A.C.; Barker, D.M. Operational implementation of a hybrid ensemble/4D-Var global data assimilation system at the Met Office. *Q. J. R. Meteorol. Soc.* **2013**, *139*, 1445–1461.
3. Fairbairn, D.; Pring, S.; Lorenc, A.; Roulstone, I. A comparison of 4DVar with ensemble data assimilation methods. *Q. J. R. Meteorol. Soc.* **2014**, *140*, 281–294.
4. Reynolds, D. Gaussian mixture models. *Encycl. Biom.* **2015**, 827–832, doi:10.1007/978-1-4899-7488-4\_196.
5. Nguyen, T.M.; Wu, Q.J.; Zhang, H. Bounded generalized Gaussian mixture model. *Pattern Recogn.* **2014**, *47*, 3132–3142.
6. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann: Burlington, MA, USA, 2016.
7. Mandel, M.I.; Weiss, R.J.; Ellis, D.P. Model-based expectation-maximization source separation and localization. *IEEE Trans. Audio Speech Lang. Proc.* **2010**, *18*, 382–394.
8. Vila, J.P.; Schniter, P. Expectation-maximization Gaussian-mixture approximate message passing. *IEEE Trans. Signal Process.* **2013**, *61*, 4658–4672.
9. Dovera, L.; Della Rossa, E. Multimodal ensemble Kalman filtering using Gaussian mixture models. *Comput. Geosci.* **2011**, *15*, 307–323.
10. Evensen, G. *Data Assimilation: The Ensemble Kalman Filter*; Springer: Secaucus, NJ, USA, 2006.
11. Evensen, G. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dyn.* **2003**, *53*, 343–367.
12. Greybush, S.J.; Kalnay, E.; Miyoshi, T.; Ide, K.; Hunt, B.R. Balance and ensemble Kalman filter localization techniques. *Mon. Weather Rev.* **2011**, *139*, 511–522.
13. Buehner, M. Evaluation of a Spatial/Spectral Covariance Localization Approach for Atmospheric Data Assimilation. *Mon. Weather Rev.* **2011**, *140*, 617–636.
14. Anderson, J.L. Localization and Sampling Error Correction in Ensemble Kalman Filter Data Assimilation. *Mon. Weather Rev.* **2012**, *140*, 2359–2371.
15. Nino-Ruiz, E.D.; Sandu, A. Efficient parallel implementation of DDDAS inference using an ensemble Kalman filter with shrinkage covariance matrix estimation. *Clust. Comput.* **2017**, 1–11, doi:10.1007/s10586-017-1407-1.
16. Nino-Ruiz, E.D.; Sandu, A.; Deng, X. A parallel implementation of the ensemble Kalman filter based on modified Cholesky decomposition. *J. Comput. Sci.* **2017**, in press, doi:10.1016/j.jocs.2017.04.005.
17. Bickel, P.J.; Levina, E. Regularized estimation of large covariance matrices. *Ann. Statist.* **2008**, *36*, 199–227.
18. Nino-Ruiz, E.D.; Mancilla, A.; Calabria, J.C. A Posterior Ensemble Kalman Filter Based on A Modified Cholesky Decomposition. *Procedia Comput. Sci.* **2017**, *108*, 2049–2058.
19. Nino-Ruiz, E.D. A Matrix-Free Posterior Ensemble Kalman Filter Implementation Based on a Modified Cholesky Decomposition. *Atmosphere* **2017**, *8*, 125.
20. Attia, A.; Sandu, A. A hybrid Monte Carlo sampling filter for non-gaussian data assimilation. *AIMS Geosci.* **2015**, *1*, 41–78.

21. Attia, A.; Moosavi, A.; Sandu, A. Cluster Sampling Filters for Non-Gaussian Data Assimilation. *arXiv* **2016**, arXiv:1607.03592.
22. Kotecha, J.H.; Djuric, P.M. Gaussian sum particle filtering. *IEEE Trans. Signal Process.* **2003**, *51*, 2602–2612.
23. Rings, J.; Vrugt, J.A.; Schoups, G.; Huisman, J.A.; Vereecken, H. Bayesian model averaging using particle filtering and Gaussian mixture modeling: Theory, concepts, and simulation experiments. *Water Resour. Res.* **2012**, *48*, doi:10.1029/2011WR011607.
24. Moon, T.K. The expectation-maximization algorithm. *IEEE Signal Process. Mag.* **1996**, *13*, 47–60.
25. Alspach, D.; Sorenson, H. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. Autom. Control* **1972**, *17*, 439–448.
26. Frei, M.; Künsch, H.R. Mixture ensemble Kalman filters. *Comput. Stat. Data Anal.* **2013**, *58*, 127–138.
27. Tagade, P.; Seybold, H.; Ravela, S. Mixture Ensembles for Data Assimilation in Dynamic Data-driven Environmental Systems I. *Procedia Comput. Sci.* **2014**, *29*, 1266–1276.
28. Sondergaard, T.; Lermusiaux, P.F. Data assimilation with Gaussian mixture models using the dynamically orthogonal field equations. Part I: Theory and scheme. *Mon. Weather Rev.* **2013**, *141*, 1737–1760.
29. Smith, K.W. Cluster ensemble Kalman filter. *Tellus A* **2007**, *59*, 749–757.
30. Hansen, P.R. *A Winner's Curse for Econometric Models: On the Joint Distribution of In-Sample Fit and Out-Of-Sample Fit and Its Implications for Model Selection*; Research Paper; Stanford University: Stanford, CA, USA, 2010; pp. 1–39.
31. Vrieze, S.I. Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychol. Methods* **2012**, *17*, 228.
32. Van Leeuwen, P.J.; Cheng, Y.; Reich, S. *Nonlinear Data Assimilation*; Springer: Cham, Switzerland, 2015; Volume 2.
33. Bannister, R. A review of operational methods of variational and ensemble-variational data assimilation. *Q. J. R. Meteorol. Soc.* **2017**, *143*, 607–633.
34. Snyder, C.; Bengtsson, T.; Bickel, P.; Anderson, J. Obstacles to high-dimensional particle filtering. *Mon. Weather Rev.* **2008**, *136*, 4629–4640.
35. Rebeschini, P.; Van Handel, R. Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab.* **2015**, *25*, 2809–2866.
36. Moré, J.J.; Sorensen, D.C. Computing a trust region step. *SIAM J. Sci. Stat. Comput.* **1983**, *4*, 553–572.
37. Conn, A.R.; Gould, N.I.; Toint, P.L. *Trust Region Methods*; SIAM: Philadelphia, PA, USA, 2000; Volume 1.
38. Nino-Ruiz, E.D. Implicit Surrogate Models for Trust Region Based Methods. *J. Comput. Sci.* **2018**, in press, doi:10.1016/j.jocs.2018.02.003.
39. Wächter, A.; Biegler, L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **2006**, *106*, 25–57.
40. Grippo, L.; Lampariello, F.; Lucidi, S. A nonmonotone line search technique for Newton's method. *SIAM J. Numer. Anal.* **1986**, *23*, 707–716.
41. Kramer, O.; Ciaurri, D.E.; Koziel, S. Derivative-free optimization. In *Computational Optimization, Methods and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 61–83.
42. Conn, A.R.; Scheinberg, K.; Vicente, L.N. *Introduction to Derivative-Free Optimization*; SIAM: Philadelphia, PA, USA, 2009; Volume 8.
43. Rios, L.M.; Sahinidis, N.V. Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Glob. Optim.* **2013**, *56*, 1247–1293.
44. Jang, J.S.R. Derivative-Free Optimization. In *Neuro-Fuzzy Soft Computing*; Prentice Hall: Englewood Cliffs, NJ, USA, 1997; pp. 173–196.
45. Nino-Ruiz, E.D.; Ardila, C.; Capacho, R. Local search methods for the solution of implicit inverse problems. In *Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1–14.
46. Anderson, J.L.; Anderson, S.L. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Mon. Weather Rev.* **1999**, *127*, 2741–2758.
47. Grubmüller, H.; Heller, H.; Windemuth, A.; Schulten, K. Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Mol. Simul.* **1991**, *6*, 121–142.
48. Van Gunsteren, W.F.; Berendsen, H. A leap-frog algorithm for stochastic dynamics. *Mol. Simul.* **1988**, *1*, 173–185.

49. Koontz, W.L.; Fukunaga, K. A nonparametric valley-seeking technique for cluster analysis. *IEEE Trans. Comput.* **1972**, *100*, 171–178.
50. Lorenz, E.N. Designing Chaotic Models. *J. Atmos. Sci.* **2005**, *62*, 1574–1587.
51. Fertig, E.J.; Harlim, J.; Hunt, B.R. A comparative study of 4D-VAR and a 4D ensemble Kalman filter: Perfect model simulations with Lorenz-96. *Tellus A* **2007**, *59*, 96–100.
52. Karimi, A.; Paul, M.R. Extensive chaos in the Lorenz-96 model. *Chaos* **2010**, *20*, 043105.
53. Gottwald, G.A.; Melbourne, I. Testing for chaos in deterministic systems with noise. *Physica D* **2005**, *212*, 100–110.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).