

# Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter

Andreas S. Stordal · Hans A. Karlsen · Geir Nævdal ·  
Hans J. Skaug · Brice Vallès

Received: 26 November 2009 / Accepted: 25 August 2010 / Published online: 17 September 2010  
© The Author(s) 2010. This article is published with open access at Springerlink.com

**Abstract** The nonlinear filtering problem occurs in many scientific areas. Sequential Monte Carlo solutions with the correct asymptotic behavior such as particle filters exist, but they are computationally too expensive when working with high-dimensional systems. The ensemble Kalman filter (EnKF) is a more robust method that has shown promising results with a small sample size, but the samples are not guaranteed to come from the true posterior distribution. By approximating the model error with a Gaussian distribution, one may represent the posterior distribution as a sum of Gaussian kernels. The resulting Gaussian mixture filter has the advantage of both a local Kalman type correction and the weighting/resampling step of a particle filter. The Gaussian mixture approximation relies on a bandwidth parameter which often has to be kept quite large in order to avoid a weight collapse in high dimensions. As a result, the Kalman correction is too large to capture highly non-Gaussian posterior distributions. In this paper, we have extended the Gaussian mixture filter (Hoteit et al., *Mon Weather Rev* 136:317–334, 2008) and also made the connection to particle filters more transparent. In particular, we introduce a tuning parameter for the importance weights. In the last part of the paper, we have performed a simulation experiment with the Lorenz40 model where our method has been

compared to the EnKF and a full implementation of a particle filter. The results clearly indicate that the new method has advantages compared to the standard EnKF.

**Keywords** Nonlinear filtering · Data assimilation · Ensemble Kalman filter · Particle filters

## 1 Introduction

The ensemble Kalman filter (EnKF) and its variants [8] are currently among the most popular approaches to data assimilation in high-dimensional dynamical systems [9]. The posterior distribution of the state variables is represented by a discrete ensemble of particles. The consensus has emerged that a small number of particles, typically 100, is sufficient, hence making application of the EnKF feasible in situations where the forward step of the data assimilation is computationally expensive. The main limitation of the EnKF is its tendency to provide approximations to the posterior that are too close to a Gaussian distribution.

Particle filters represent another ensemble-based approach to data assimilation, in which there is a weight associated with each ensemble member [6]. The weights are normalized so that they sum to one. At each assimilation step, the ensemble is updated, in a stochastic manner, according to Bayes' rule. The particle filter produces a sample from the exact posterior distribution. One limitation is that it is not applicable when the state vector is high dimensional due to a phenomenon known as the curse of dimensionality, i.e., as the dimension of the system increases, the largest of the sample weights converges to one in probability [3] causing a

---

A. S. Stordal (✉) · G. Nævdal · B. Vallès  
IRIS, P.O. Box 8046, 4068 Stavanger, Norway  
e-mail: Andreas.S.Stordal@iris.no

H. A. Karlsen · H. J. Skaug  
Department of Mathematics, University of Bergen,  
Johs. Brunsøgt 15, 5020 Bergen, Norway

filter degeneracy where the posterior distribution is represented by a single point in the state space.

Recently, several filters with both a Kalman update step and weighting correction have been developed such as the EnKF–sequential importance sampling (SIS) [17] and Gaussian mixture filters [4, 12, 14]. In the Gaussian mixture filters, the prior distribution is approximated by a mixture density [20] where each ensemble member forms the center of a Gaussian density function (a kernel). The mixture density, including the weights, is propagated through the dynamical system and updated according to Bayes' rule on the arrival of a new datum. Hence, weight degeneration is also a problem with the mixture filter, at least in its basic form.

The EnKF can be viewed as a mixture filter, obtained by using Gaussian kernels and forcing the weights to be uniform [11]. Our main proposal is to loosen up the latter requirement. We introduce a tuning parameter  $\alpha \in [0, 1]$  such that  $\alpha = 0$  gives the uniform weights of the EnKF and  $\alpha = 1$  gives the weights of the Gaussian mixture filter. The main idea is that by taking  $\alpha$  to be small, we reduce the weight degeneracy problem, but taking  $\alpha > 0$ , we improve upon the EnKF approximation of the posterior. In particular, we aim at better preserving non-Gaussian features of the marginal distributions. The proposed approach is adaptive, in the sense that an optimal  $\alpha$  is sought at each assimilation step resulting in an adaptive Gaussian mixture filter.

The Gaussian mixture filter is first applied to the Bernoulli equation [2] to demonstrate its flexibility and similarities with particle filters and the EnKF. The 40-dimensional chaotic Lorenz40 model [16] is then studied to show how the new parameter  $\alpha$  allows us to keep the linear Kalman update smaller than in the EnKF and the Gaussian mixture filter and at the same time avoid degeneracy of the weights. The nonlinear filter problem and some of its approximate solutions, including Gaussian mixture filters, are discussed in Section 2. In Section 3, we introduce the adaptive Gaussian mixture filter. The simulation studies are the content of Section 4 before a summary and discussion concludes the paper in Section 5.

## 2 The nonlinear filtering problem

We are interested in estimating sequentially in time the posterior distribution of the state vector,  $\mathbf{x}_t$ , of a high-dimensional dynamic system conditioned upon some noisy measurements. We restrict ourselves to the case where the state is Markovian with a linear measurement operator. The latter can always be achieved by

expanding the state space. For all  $t \geq 1$ , the system is described by the equations

$$\begin{aligned}\mathbf{x}_t &= \mathcal{M}_t(\mathbf{x}_{t-1}) + \eta_t, \\ \mathbf{y}_t &= \mathbb{H}_t \mathbf{x}_t + \epsilon_t,\end{aligned}\quad (2.1)$$

where  $\mathcal{M}_t$  is the nonlinear forward operator and  $\mathbb{H}_t$  is the linear observation operator. The measurement error,  $\epsilon_t$ , is assumed to be Gaussian white noise with covariance matrix  $\mathbb{R}_t$ , and  $\eta_t$  is the model error with density  $p(\eta_t)$ . The initial distribution is  $p(\mathbf{x}_0)$ . The optimal solution to the filtering problem is the posterior density  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ . That is, the conditional distributions of the state  $\mathbf{x}_t$  given  $\mathbf{y}_{1:t}$ , where  $\mathbf{y}_{1:t} = (\mathbf{y}_1, \dots, \mathbf{y}_t)$ , are all the observations up to and including time  $t$ .

Given the posterior density at time  $t - 1$ , the prior density at time  $t$  can be calculated as

$$p(\mathbf{x}_t | \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}. \quad (2.2)$$

When a new observation  $\mathbf{y}_t$  arrives, the posterior density is updated recursively via Bayes' rule:

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1})}, \quad (2.3)$$

with the normalizing constant given by

$$p(\mathbf{y}_t | \mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:t-1}) d\mathbf{x}_t.$$

In general, the optimal solution cannot be computed analytically. We therefore need approximative solutions such as Monte Carlo methods where one represents the prior and posterior densities with a random sample. We denote by the set of particles/ensembles  $\{\mathbf{x}_t^i\}_{i=1}^N$  a sample approximating the prior density (Eq. 2.2) and by the set of particles  $\{\hat{\mathbf{x}}_t^i\}_{i=1}^N$  a sample approximating the posterior density (Eq. 2.3).

### 2.1 The ensemble Kalman filter

The EnKF [8] is a sequential Monte Carlo method that solves the nonlinear filtering problem with an additional Gaussian assumption. Given a sample,  $\{\hat{\mathbf{x}}_{t-1}^i\}_{i=1}^N$ , from the posterior density  $p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$ , the EnKF then uses the Markov property of the system to obtain a sample from the prior Eq. 2.2 by sampling from the forward density  $p(\mathbf{x}_t | \hat{\mathbf{x}}_{t-1}^i)$  according to Eq. 2.1

$$\mathbf{x}_t^i = \mathcal{M}_t(\hat{\mathbf{x}}_{t-1}^i) + \eta_t^i, \quad i = 1, \dots, N. \quad (2.4)$$

By making the assumption that  $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$  is Gaussian with mean  $\mu_t$  and covariance matrix  $\mathbb{P}_t$ , it follows from the Gaussian observation error that  $p(\mathbf{x}_t|\mathbf{y}_{1:t})$  is Gaussian with mean  $\hat{\mu}_t$  and covariance matrix  $\hat{\mathbb{P}}_t$  given by the standard Kalman filter update,

$$\begin{aligned} \hat{\mu}_t &= \mu_t + \mathbb{K}_t(\mathbf{y}_t - \mathbb{H}_t\mu_t), \\ \hat{\mathbb{P}}_t &= (\mathbb{I} - \mathbb{K}_t\mathbb{H}_t)\mathbb{P}_t, \\ \mathbb{K}_t &= \mathbb{P}_t\mathbb{H}_t^T(\mathbb{H}_t\mathbb{P}_t\mathbb{H}_t^T + \mathbb{R}_t)^{-1}. \end{aligned} \tag{2.5}$$

The EnKF takes  $\mathbb{P}_t$  to be the empirical covariance matrix of  $\{\mathbf{x}_t^i\}_{i=1}^N$  and in updating particle  $i$  it puts  $\mu_t = \mathbf{x}_t^i$ ; thus, a sample from the posterior is provided by

$$\hat{\mathbf{x}}_t^i = \mathbf{x}_t^i + \mathbb{K}_t(\mathbf{y}_t - (\mathbb{H}_t\mathbf{x}_t^i + \epsilon_t^i)), \quad i = 1, \dots, N.$$

For a general nonlinear system, this scheme is biased in the sense that the ensemble members will not be a sample from the true posterior distribution even in the limit of an infinite number of ensemble members. However, it is robust meaning that it does not suffer from the dimensionality problem since every particle carry the same weight.

### 2.2 Sequential importance sampling and particle filters

SIS and particle filters [6, 22] are also discrete approximations of the optimal filter solution given by Eq. 2.3. They aim at building up sequentially a sample from the posterior density  $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ .

Assume that we have, at a given time  $t$ , a sample  $\{\hat{\mathbf{x}}_{0:t}^i\}_{i=1}^N$  from the joint posterior density  $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ . A sample  $\{\hat{\mathbf{x}}_t^i\}_{i=1}^N$  from the posterior density  $p(\mathbf{x}_t|\mathbf{y}_{1:t})$  is trivially obtained by omitting  $\hat{\mathbf{x}}_{0:t-1}^i$  in the vector  $(\hat{\mathbf{x}}_{0:t-1}^i, \hat{\mathbf{x}}_t^i)$ . As in ordinary importance sampling, we choose an importance function (a density)  $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ , not necessarily dependent on  $\mathbf{y}_{1:t}$ . Then we construct a sample from  $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$  by drawing  $N$  vectors  $\{\mathbf{x}_{0:t}^i\}_{i=1}^N$  from the importance function  $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$  and attach to each sample an associated weight,  $w_t^i$ , which is the ratio of the target density and the importance function

$$w_t^i = \frac{p(\mathbf{x}_{0:t}^i|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}^i|\mathbf{y}_{1:t})}. \tag{2.6}$$

This method is valid for almost any probability density function  $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$  as long as its support contains the support of  $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ . That is, as long as  $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) > 0$  implies that  $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) > 0$ . Otherwise, given  $\mathbf{y}_{1:t}$ , there exists a region of positive probability in the posterior

that never can be reached by sampling from  $q$  and this necessarily results in biased estimates.

Clearly, all the weights will be equal to one if we sample directly from the posterior distribution. Since we in general cannot sample from this, we want to find an importance function that is similar to the posterior to keep the weights as close to one as possible. If one for instance is interested in estimating the mean of the posterior distribution, the optimal choice of importance function would be the one that minimizes the variance of the estimator  $N^{-1} \sum_{i=1}^N w^i x^i$ . However, in dynamic systems, we are often limited to the forward model of the system due to lack of knowledge of the posterior and the fact that one needs an importance function that can be updated sequentially.

If we choose an importance function of the form

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}), \tag{2.7}$$

we see that a sample  $\{\mathbf{x}_{0:t}^i\}_{i=1}^N$  is obtained from  $\{\mathbf{x}_{0:t-1}^i\}_{i=1}^N$  by sampling  $\{\mathbf{x}_t^i\}_{i=1}^N$  from  $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$  and setting  $\mathbf{x}_{0:t}^i = (\mathbf{x}_t^i, \mathbf{x}_{0:t-1}^i)$ . The weights can also be updated sequentially up to a proportionality constant via Bayes' theorem

$$\begin{aligned} w_t^i &= \frac{p(\mathbf{x}_{0:t}^i|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}^i|\mathbf{y}_{1:t})} \\ &\propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)p(\mathbf{x}_{0:t-1}^i|\mathbf{y}_{1:t-1})}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_t)q(\mathbf{x}_{0:t-1}^i|\mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_t)} w_{t-1}^i. \end{aligned} \tag{2.8}$$

From Eqs. 2.7 and 2.8, we see that we can obtain our sample sequentially in time. This is of great importance as one does not have to restart the algorithm when new data arrive.

If we are only interested in the marginal posterior density of  $\mathbf{x}_t$ , we only need to store  $\{\hat{\mathbf{x}}_{t-1}^i, w_{t-1}^i\}_{i=1}^N$ . The EnKF can be viewed as an SIS algorithm restricting the weights to be uniform.

The SIS algorithm described above results in a filter degeneracy in the following sense. After a few time steps, most of the weights become numerically zero [10]. To avoid this filter degeneracy, an additional resampling step is introduced in the sequential importance resampling filter (SIR), which is the most common particle filter. It differs from SIS only by the additional resampling step. The standard SIR filter use  $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1})$  as the importance function and the corresponding importance

weights are updated according to Eq. 2.8 as  $\bar{w}_t^i = w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_t^i)$ . The weights have to be normalized as they are known only up to a constant,

$$w_t^i = \frac{\bar{w}_t^i}{\sum_{j=1}^N \bar{w}_t^j}. \tag{2.9}$$

A measure of the degree of (non) degeneracy is the effective ensemble size,  $N_{\text{eff}}$ , [13] estimated by

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w_t^i)^2}. \tag{2.10}$$

With all particles having equal weight,  $\hat{N}_{\text{eff}}$  is equal to  $N$ , while it is equal to one if all the weight is on one particle. If  $\hat{N}_{\text{eff}}$  is smaller than some critical value  $N_c$ , resampling is usually performed.

The resampling step is a selection procedure where the best fitted particles survive. There are several ways of performing the resampling, the most used being perhaps the multinomial resampling where  $N$  new particles are selected from a multinomial distribution with parameters  $(N, w_t^1, \dots, w_t^N)$ . After a resampling, the weights are reset to  $N^{-1}$ . Although resampling is necessary, it can be shown that it leads to underestimation of the posterior variance of the state variables [21]. Also since it is a stochastic procedure, resampling introduces more variance to the estimator itself. If the variance of the SIS mean estimator  $\hat{\mu}_{\text{SIS}} = \sum_{i=1}^N \mathbf{x}_i w_i$  is given by  $\text{Var}(\sum_{i=1}^N \mathbf{x}_i w_i)$ , it is easy to show that the mean estimator in the SIR with multinomial resampling is given by

$$\begin{aligned} \text{Var}(\hat{\mu}_{\text{SIR}}) &= \text{Var}(\hat{\mu}_{\text{SIS}}) \\ &+ N^{-1} \mathbb{E} \left[ \sum_{i=1}^N \mathbf{x}_i^2 w_i - \left( \sum_{i=1}^N \mathbf{x}_i w_i \right)^2 \right]. \end{aligned}$$

For high-dimensional systems, the main problem with the SIR filter is the curse of dimensionality: The number of particles should grow exponentially with the dimension of the system in order to avoid a filter collapse [3, 21]. The resampling step in the SIR was introduced to avoid the weights going to zero after a few time steps. However, when the dimension of the system becomes large, most of the weights will be zero after one time step and a resampling will not avoid degeneracy as it will select the few particles with nonzero weights several times. This makes applications to high-dimensional data assimilation practically infeasible as the forward integration of the model is often very time-consuming and hence puts an upper limit on  $N$ .

### 2.3 Improved SIR filter

Since the SIR filter originates from sequential importance sampling, we know that it is possible to improve the efficiency of our estimator by choosing a different importance function  $q$  [18]. Theoretically, the optimal importance function is [7]

$$q(\mathbf{x}_{0:t} | \mathbf{y}_{1:t})_{\text{opt}} = p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t) p(\mathbf{x}_{0:t-1} | \mathbf{y}_{1:t-1}).$$

This choice of  $q$  is optimal in the sense that it minimizes the variance of the weights conditioned on  $\mathbf{x}_{t-1}$  and  $\mathbf{y}_t$ . The weights,  $w_t^i$ , in the standard SIR were proportional to  $w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_t^i)$ , but with the optimal choice of  $q$ , the weights at time  $t$  are now, according to Eq. 2.8, proportional to  $w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_{t-1}^i)$ . These weights are more uniform since the variance of  $\mathbf{y}_t$  is larger conditioned upon  $\mathbf{x}_{t-1}^i$  than on  $\mathbf{x}_t^i$ .

If we now make the assumption that the model error  $\eta_t$  is Gaussian white noise with covariance matrix  $\mathbb{Q}_t$ , it is possible to find the optimal importance function analytically. The density  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t)$  under the above assumption is Gaussian with mean  $\mu_t$  and covariance  $\mathbb{P}_t$  given as

$$\begin{aligned} \mu_t &= \mathcal{M}_t(\mathbf{x}_{t-1}) + \mathbb{K}_t(\mathbf{y}_t - \mathbb{H}_t \mathcal{M}_t(\mathbf{x}_{t-1})), \\ \mathbb{K}_t &= \mathbb{Q}_t \mathbb{H}_t^T (\mathbb{H}_t \mathbb{Q}_t \mathbb{H}_t^T + \mathbb{R}_t)^{-1}, \\ \mathbb{P}_t &= (\mathbb{I} - \mathbb{K}_t \mathbb{H}_t) \mathbb{Q}_t. \end{aligned} \tag{2.11}$$

To evaluate the weights, we need the density  $p(\mathbf{y}_t | \mathbf{x}_{t-1})$ . This can be found from Eq. 2.1

$$\mathbf{y}_t = \mathbb{H}_t \mathcal{M}_t(\mathbf{x}_{t-1}) + \mathbb{H}_t \eta_t + \epsilon_t.$$

Since  $\mathbb{H}_t$  is linear, it follows that  $p(\mathbf{y}_t | \mathbf{x}_{t-1})$  is Gaussian with mean  $\mathbf{b}_t$  and covariance  $\Sigma_t$  given by

$$\begin{aligned} \mathbf{b}_t &= \mathbb{H}_t \mathcal{M}_t(\mathbf{x}_{t-1}), \\ \Sigma_t &= \mathbb{H}_t \mathbb{Q}_t \mathbb{H}_t^T + \mathbb{R}_t. \end{aligned}$$

The additional term  $\mathbb{H}_t \mathbb{Q}_t \mathbb{H}_t^T$  of the covariance ensures that the weights are more uniform than in the standard SIR filter.

The improved SIR filter still suffers from degeneracy due to the fact that the model error covariance  $\mathbb{Q}_t$  is usually very small. In practice, this means that the Kalman update in Eq. 2.11 is small and the weights suffer from degeneracy in high dimensions as the increase in  $\Sigma_t$  is not sufficient to make the particles compatible with the measurements. One way to avoid this collapse is to force the weights to be more uniform by artificially increasing the uncertainty  $\mathbb{Q}_t$  associated with each particle so that they become compatible with the measurements. This can be obtained by smoothing each sample with a Gaussian density. That is, we take

each particle to be the center of a Gaussian density, thereby representing the distribution of the particles as a weighted sum of Gaussian density functions, a Gaussian mixture.

### 2.4 Gaussian mixture filters

Gaussian mixture filters [1, 4, 5, 14] are based on the fact that any density function  $f$  defined on  $\mathbb{R}^d$  can be approximated in  $L_1(\mathbb{R}^d)$  from a sample  $\{\mathbf{x}_i\}_{i=1}^N$  by a density of the form

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N w_i \Phi(\mathbf{x} - \mathbf{x}_i, \mathbb{P}),$$

where  $\Phi(\mathbf{x}, \mathbb{P})$  denotes a zero mean multivariate Gaussian density with covariance matrix  $\mathbb{P}$ .  $N$  is the number of particles and  $\{w_i\}_{i=1}^N$  are scalar weights so that  $\sum_{i=1}^N w_i = 1$  [23]. Note that each particle represents the mean of a Gaussian kernel and that the uncertainty associated with each particle is given by the covariance of that Gaussian kernel.

In filter theory, with known one-step Markov transitions for the state vector and known likelihood function, the only unknown quantity needed to update the posterior distribution at time  $t$  is the prior density  $p_t(\mathbf{x}_t | \mathbf{y}_{1:t-1})$  according to Eq. 2.3. The simple idea behind Gaussian mixture filters is that we at each time step,  $t$ , approximate the prior density  $p(\mathbf{x}_t | \mathbf{y}_{1:t-1})$  by a Gaussian kernel density estimator

$$p_N(\mathbf{x}_t | \mathbf{y}_{1:t-1}) = \sum_{i=1}^N w_{t-1}^i \Phi(\mathbf{x}_t - \mathbf{x}_t^i, \mathbb{P}_t), \tag{2.12}$$

where  $\{\mathbf{x}_t^i\}_{i=1}^N = \{\mathcal{M}_t(\hat{\mathbf{x}}_{t-1}^i)\}_{i=1}^N$ .

The choice of the kernel is theoretically not that important in terms of density estimation. However, since Gaussian mixtures are closed under linear measurement operator and Gaussian measurement error, that is a Gaussian mixture prior, updated against Gaussian linear measurements is again a Gaussian mixture (see below). A two-component Gaussian mixture before and after update is shown in Fig. 1. Note that the density is bimodal before the update, while it is unimodal after.

The numerical algorithm (“Appendix B”) starts by drawing  $N$  independent identically distributed (iid) particles,  $\{\mathbf{x}_0^i\}_{i=1}^N$ , from the initial density  $p(\mathbf{x}_0)$ . Note that there is no measurements at time zero so that the weights  $w_0^i = N^{-1}$  for all  $i = 1 \dots N$ , hence  $\{\hat{\mathbf{x}}_0^i\}_{i=1}^N = \{\mathbf{x}_0^i\}_{i=1}^N$ . The Gaussian mixture is then initialized by constructing  $N$  Gaussian kernels where kernel number  $i$  have mean  $\hat{\mathbf{x}}_0^i$  and covariance matrix  $\hat{\mathbb{P}}_0 = h^2 \mathbb{S}(\hat{\mathbb{X}}_0)$ ,

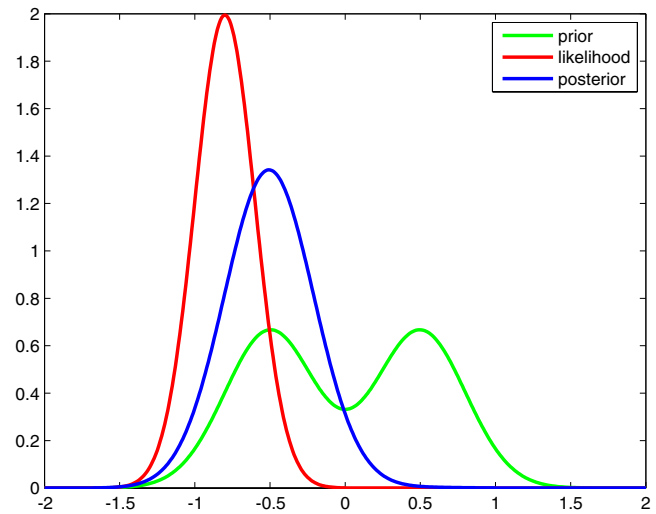


Fig. 1 Two component Gaussian mixture filter

where  $\mathbb{S}(\hat{\mathbb{X}}_0)$  is the sample covariance matrix of the ensemble  $\{\hat{\mathbf{x}}_0^i\}_{i=1}^N$  and  $h$  is the bandwidth parameter. The ensemble matrix, denoted by  $\mathbb{X}$ , contains all the ensemble members, where column number  $i$  represents ensemble member  $i$ .

Note that the sample covariance is factorized in a different way than in the standard EnKF. While the EnKF uses the ensemble matrix  $\mathbb{X}$  and its mean  $\bar{\mathbb{X}}$  to factor  $\mathbb{S}$  as  $(\mathbb{X} - \bar{\mathbb{X}})(\mathbb{X} - \bar{\mathbb{X}})^T / (N - 1)$ , the Gaussian mixture filter factor  $\mathbb{S}$  is  $\mathbb{L}\mathbb{U}\mathbb{L}^T$  (see “Appendix A”).

The total covariance matrix of the mixture distribution is  $(1 + h^2)\mathbb{S}(\hat{\mathbb{X}})$ . The excess variance can easily be removed by replacing  $\hat{\mathbf{x}}^i$  with  $\hat{\mathbf{x}}^i = a\hat{\mathbf{x}}^i + (1 - a)N^{-1} \sum_i \hat{\mathbf{x}}^i$ , where  $a = \sqrt{1 - h^2}$  [15]. This has not been applied in our work.

If we were only interested in the marginal properties of the individual components of  $\mathbf{x}_t$ , the optimal choice of the bandwidth  $h$  would be  $h_{\text{opt}} \propto N^{-1/5}$  [20]. However, the choice of the bandwidth parameter  $h$  determines the magnitude of the Kalman update step, described later, and it might be beneficial to choose  $h > h_{\text{opt}}$  in order to reduce the risk of filter divergence. We therefore treat  $h$  as a design parameter.

When a new measurement arrives, the posterior density approximation is updated via Bayes’ theorem,

$$p_N(\mathbf{x}_t | \mathbf{y}_{1:t}) \propto p_N(\mathbf{x}_t | \mathbf{y}_{1:t-1}) \Phi(\mathbf{y}_t - \mathbb{H}_t \mathbf{x}_t, \mathbb{R}_t). \tag{2.13}$$

Since  $p_N(\mathbf{x}_t | \mathbf{y}_{1:t-1})$  is a weighted sum of Gaussian densities,  $p_N(\mathbf{x}_t | \mathbf{y}_{1:t})$  is a weighted sum of products of two

Gaussian densities where component number  $i$  is proportional to

$$w_{t-1}^i \Phi(\mathbf{x}_t - \mathbf{x}_t^i, \mathbb{P}_t) \Phi(\mathbf{y}_t - \mathbb{H}_t \mathbf{x}_t, \mathbb{R}_t). \tag{2.14}$$

By adding and subtracting  $\mathbb{H}_t \mathbf{x}_t^i$  in the first argument of  $\Phi(\mathbf{y}_t - \mathbb{H}_t \mathbf{x}_t, \mathbb{R}_t)$ , Eq. 2.14 may be re-written as

$$w_{t-1}^i \Phi(\mathbf{x}_t - \tilde{\mathbf{x}}_t^i, \tilde{\mathbb{P}}_t) \Phi(\mathbf{y}_t - \mathbb{H}_t \mathbf{x}_t^i, \Sigma_t),$$

where

$$\tilde{\mathbf{x}}_t^i = \mathbf{x}_t^i + \mathbb{K}_t(\mathbf{y}_t - \mathbb{H}_t \mathbf{x}_t^i),$$

$$\mathbb{K}_t = \mathbb{P}_t \mathbb{H}_t^T \Sigma_t^{-1},$$

$$\Sigma_t = \mathbb{H}_t \mathbb{P}_t \mathbb{H}_t^T + \mathbb{R}_t,$$

$$\tilde{\mathbb{P}}_t = (\mathbb{I} - \mathbb{K}_t \mathbb{H}_t) \mathbb{P}_t, \tag{2.15}$$

The updated weights  $\bar{w}_t = w_{t-1}^i \Phi(\mathbf{y}_t - \mathbb{H}_t \mathbf{x}_t^i, \Sigma_t)$  are normalized as

$$w_t^i = \frac{\bar{w}_t^i}{\sum_{j=1}^N \bar{w}_t^j},$$

and we see that the Eq. 2.13 is a Gaussian mixture

$$p_N(\mathbf{x}_t | \mathbf{y}_{1:t}) = \sum_{i=1}^N w_t^i \Phi(\mathbf{x}_t - \tilde{\mathbf{x}}_t^i, \tilde{\mathbb{P}}_t).$$

### 2.4.1 Resampling

It is necessary to resample if the effective sample size is too small, which means that most of the weights are allocated to a minor fraction of the particles. We perform a resampling step to multiply those particles with high weights and discard the ones with small weights. The multinomial resampling in the SIR filter is equivalent to sampling new particles  $\{\hat{\mathbf{x}}_t^i\}_{i=1}^N$  with replacement from the weighted empirical distribution defined by the particles and the weights. The extension to the Gaussian mixture filter is to sample the new particles  $\hat{\mathbf{x}}_t^i$  from the kernel density estimate

$$\hat{\mathbf{x}}_t^i \sim \sum_{i=1}^N w_t^i \Phi(\mathbf{x}_t - \tilde{\mathbf{x}}_t^i, \tilde{\mathbb{P}}_t).$$

After resampling, we reset the weights to  $1/N$  and  $\hat{\mathbb{P}}_t = h^2 \text{cov}(\hat{\mathbf{x}}_t^i)$  so that our Gaussian mixture approximation of the posterior distribution becomes

$$p_N(\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_t - \hat{\mathbf{x}}_t^i, \hat{\mathbb{P}}_t).$$

This resampling strategy is biased in the sense that the expected covariance of the Gaussian mixture produced is larger than the one before the resampling step, but as

we pointed out earlier, resampling generally leads to an underestimation of the posterior variance so we do not introduce any bias correction in our resampling scheme.

To sum up, the posterior density is approximated by

$$p_N(\mathbf{x}_t | \mathbf{y}_{1:t}) = \sum_{i=1}^N w_t^i \Phi(\mathbf{x}_t - \hat{\mathbf{x}}_t^i, \hat{\mathbb{P}}_t),$$

where  $\hat{\mathbf{x}}_t^i \sim \sum_{i=1}^N w_t^i \Phi(\mathbf{x}_t - \tilde{\mathbf{x}}_t^i, \tilde{\mathbb{P}}_t)$ ,  $w_t^i = N^{-1}$ , and  $\hat{\mathbb{P}}_t = \mathbb{S}(\hat{\mathbf{x}}_t^i)$  if a resampling is performed and  $\hat{\mathbf{x}}_t^i = \tilde{\mathbf{x}}_t^i$ ,  $w_t^i = w_{t-1}^i$ , and  $\hat{\mathbb{P}}_t = \tilde{\mathbb{P}}_t$  if no resampling is performed.

Since we are working with densities, the Gaussian mixture filter differs from the SIR and EnKF in the sense that we do not add any model noise in the forward integration nor in the Kalman update. By adding noise and setting  $h$  equal to zero, we would recover the standard SIR [11]. However, if we add noise and let  $h$  be equal to one, we are not able to capture the behavior of the EnKF entirely since the weights in the EnKF are equal by definition.

To estimate the posterior density precisely, we want to keep  $h$  small. However, this may lead to a degeneracy of the filter due to the collapse of the weights. On the other hand, if  $h$  is too large, the Kalman update is almost the same as in the EnKF and the weights will contain very little information, as they will be close to uniform. That is, the filter becomes too close to the EnKF and may not be able to capture highly non-Gaussian posterior distributions.

## 3 Adaptive Gaussian mixture filter

In order to keep  $h$  small and avoid degeneracy, we introduce a bias-variance tradeoff parameter  $\alpha$ . The idea behind  $\alpha$  is to force the weights to be more uniform without increasing the bandwidth parameter  $h$ . By decreasing the value of largest weights, we give new life into “dead” particles and avoid a filter degeneracy for small values of  $h$ .

### 3.1 Reducing divergence by weight interpolation

Consider a sample from a density  $g$  and the importance sampling estimator of  $\mathbf{E}_p f = \int f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$  where  $p$  is the target density and  $f$  is an arbitrary measurable function.

$$\hat{\mathbf{E}}_p f = N^{-1} \sum_{i=1}^N \frac{\bar{w}_i}{N^{-1} \sum_i \bar{w}_i} f(\mathbf{x}_i), \tag{3.1}$$

where  $\bar{w}_i = g(\mathbf{x}_i)/p(\mathbf{x}_i)$ . Although slightly biased, this estimator converges almost surely for almost any density  $g$ . However, if the ratio  $p/g$  is unbounded, the

weights will vary widely, giving too much importance on a few values  $\mathbf{x}_j$ . The result is that the estimator (Eq. 3.1) may change dramatically from one iteration to another, i.e., the estimator is unstable due to large variance. This scenario is typical if the tails of  $g$  is lighter than the tails of  $p$ . It has also been shown that this behavior occurs when the dimension of the variable  $\mathbf{X}$  is large [3]. For the Gaussian mixture filter, this behavior occurs when  $h$  is small. Thus, large range of the (un-normalized) weights results in large variance of the estimator in Eq. 3.1. The first problem can be avoided by replacing  $g$  with a mixture

$$\rho g(\mathbf{x}) + (1 - \rho)l(\mathbf{x}),$$

where  $l(\mathbf{x})$  is chosen for its heavy tails [18]. Here, we propose a different approach to handle the problem in high dimensions. To avoid too large range of the weights, we replace assign new weights to each variable by interpolating the original weight with the a uniform weight

$$w_\alpha^i = \alpha w^i + (1 - \alpha)N^{-1}. \tag{3.2}$$

This procedure will reduce the range of the (normalized) weights, hence reduce the variability of the estimator in Eq. 3.1. However, this procedure introduces bias in the estimator which asymptotically is equal to

$$\alpha \mathbf{E}_p f + (1 - \alpha)\mathbf{E}_g f - \mathbf{E}_p f = (\alpha - 1)(\mathbf{E}_p f - \mathbf{E}_g f). \tag{3.3}$$

As  $\alpha$  decreases, the filter solution moves further away from the correct posterior distribution, but it still captures some of the non-Gaussian behavior left out by the EnKF.

With this new parameter, we notice that we are able to recover the SIR filter by adding model noise in the forward integration and choosing  $\alpha = 1$  and  $h = 0$ . The particles can also behave exactly as the ones from the standard EnKF by adding model noise in the forward step, observation noise in the Kalman update step, and choosing  $\alpha = 0$  and  $h = 1$ .

For two different reasons, we seek to find an  $\alpha$  that varies in time. The first reason is for computational purposes, we do not want to run the model several times with different values for  $\alpha$ . The second reason is for filtering purposes in stationary models. As time increases, the Kalman update and resampling at previous time steps have pulled the particles into areas of higher likelihood decreasing the variability in the weights. This could make it possible to increase  $\alpha$  without causing degeneracy of the particles.

### 3.2 Selecting the tuning parameter

The tuning parameter  $\alpha$  is a tool that guarantees a lower bound for the weights which again implies that the variance of a posterior prediction given by Eq. 3.1 is under control. However, the payoff for the variance reduction is more bias. By carefully choosing  $\alpha$ , the goal is to minimize the total effect of variance and bias.

As a measure of the degree of divergence in Eq. 3.1, we use

$$\frac{N^{-1} \sum_i w_i^2}{(N^{-1} \sum_i w_i)^2} = N \sum w_i^2 = \frac{N}{\hat{N}_{\text{eff}}}, \tag{3.4}$$

and the  $\alpha$  equivalent term is  $N \sum_i (w_\alpha^i)^2$ . The constant term  $\mathbf{E}_p f - \mathbf{E}_g f$  in Eq. 3.3 could be estimated by  $\sum_i (f(\mathbf{x}_i)(w^i - N^{-1}))$ , but this involves evaluation of the weights and is therefore very unreliable. Instead, we seek to minimize the square of the standardized bias,  $\text{bias}^2 / \text{max bias}^2 = (1 - \alpha)^2$ . Our suggestion for  $\alpha$  is therefor the value that minimizes the function

$$J(\alpha) = N \sum_i (w_\alpha^i)^2 + (\alpha - 1)^2. \tag{3.5}$$

The first term represents the variance while the second term reflects the bias. The second one could be thought of as a penalty term.

Finding the minimum of Eq. 3.5 gives the solution

$$\alpha_{\text{min}} = N^{-1} \hat{N}_{\text{eff}}. \tag{3.6}$$

The choice of  $\alpha$  is adaptive in the sense that it changes at each assimilation time according to how well our samples match the observations. However, it is not consistent. We see that  $\lim_{N \rightarrow \infty} \hat{N}_{\text{eff}} N^{-1} = 1 \mathbf{E}_p(w)$ . So that  $\lim_{N \rightarrow \infty} \alpha \leq 1$ . A consistent choice of  $\alpha$  would therefore be  $K\alpha$ , where  $K = \mathbf{E}_p(w)$ . However, we cannot find an analytical expression for  $K$  in general and since we require a large sample to estimate it, we suggest using  $K = 1$  as we want to be conservative when the sample size is small.

The parameter  $\alpha$  was introduced in order to avoid degeneracy of the weights. Degeneracy occurs when the effective ensemble size becomes so small that only a few particles represent the posterior distribution. With our choice of  $\alpha$ , we see that the new estimated effective ensemble size is

$$\begin{aligned} \hat{N}_{\text{eff}}^\alpha &= \frac{1}{\sum_{i=1}^N (w_\alpha^i)^2} = \frac{1}{\sum_{i=1}^N (\alpha w^i + (1 - \alpha) N^{-1})^2} \\ &= \frac{N^3}{\hat{N}_{\text{eff}}(N - \hat{N}_{\text{eff}}) + N^2}, \end{aligned}$$

which takes its minimum value of  $0.8N$  at  $\hat{N}_{\text{eff}} = 0.5N$ . That is, the effective ensemble size stays above 80% with our choice of  $\alpha$  and a filter degeneracy is impossible.

In the next section, we present some examples showing how the Gaussian mixture filters can estimate the correct posterior distribution for small systems and also why  $\alpha$  is advantageous for complex systems of higher dimensions.

### 4 Simulation studies

In the first example, we want to show the effect of the bandwidth parameter  $h$  in the Gaussian mixture filter in terms of the estimated posterior distributions. From the theory, we expect the Gaussian mixture filter to produce posterior distributions close to the one produced by the EnKF for large values of  $h$ . For small values of  $h$ , the results should be similar to the estimate obtained by the SIR filter. The model we choose is the univariate nonlinear Bernoulli differential equation used by Apte et al. [2].

#### 4.1 Bernoulli model

The Bernoulli equation is given by

$$\frac{d\mathbf{x}}{dt} - \mathbf{x} = -\mathbf{x}^3, \quad \mathbf{x}(0) = \mathbf{x}_0,$$

with analytical solution

$$\mathbf{x}(t) = \mathcal{M}(\mathbf{x}_0) = \mathbf{x}_0 \times (\mathbf{x}_0^2 + (1 - \mathbf{x}_0^2) e^{-2t})^{-\frac{1}{2}}. \quad (4.1)$$

In our example, the initial distribution of  $\mathbf{x}_0$  is Gaussian with mean  $-0.1$  and standard deviation  $0.2$ . The entire system is given by

$$\mathbf{x}(k\delta) = \mathcal{M}(\mathbf{x}((k-1)\delta))$$

$$\mathbf{y}_k = \mathbf{x}(k\delta) + \epsilon_k,$$

where  $\epsilon_k$  is a zero mean Gaussian noise with a standard deviation of  $0.8$ ,  $k = 1, \dots, 10$ , and  $\delta$  is the frequency of observations chosen as  $0.3$  in the following. Due to the functional relationship in Eq. 4.1, we are able to compute the true posterior density at each time step  $k$  [2]. The true value of  $\mathbf{x}_0$  is chosen as  $0.0001$ , which lies in the tail of the initial distribution. We calculate the posterior density at the final time step ( $k = 10$ ) and compare the result with the density estimates from the SIR filter, EnKF, and the Gaussian mixture filters.

All filters are carried out with  $N = 1,000$  ensemble members/particles which is small compared to the  $50,000$  in [2]. Resampling is performed when the num-

ber of effective members is less than half the ensemble size. For the Gaussian mixture filters,  $h$  is chosen as  $0.1, 0.2, 0.6$  and  $1$ .

Since the number of ensemble members are large and the problem is simple, this example is only included as a motivation to demonstrate the potential of Gaussian mixture filters.

#### 4.1.1 Results

Figure 2 shows the different density estimates of the filter solutions against the true posterior density (black). As expected, for small values of  $h$ , the Gaussian mixture filter is able to reproduce the skewness of the true posterior density. Similarly to the EnKF, it

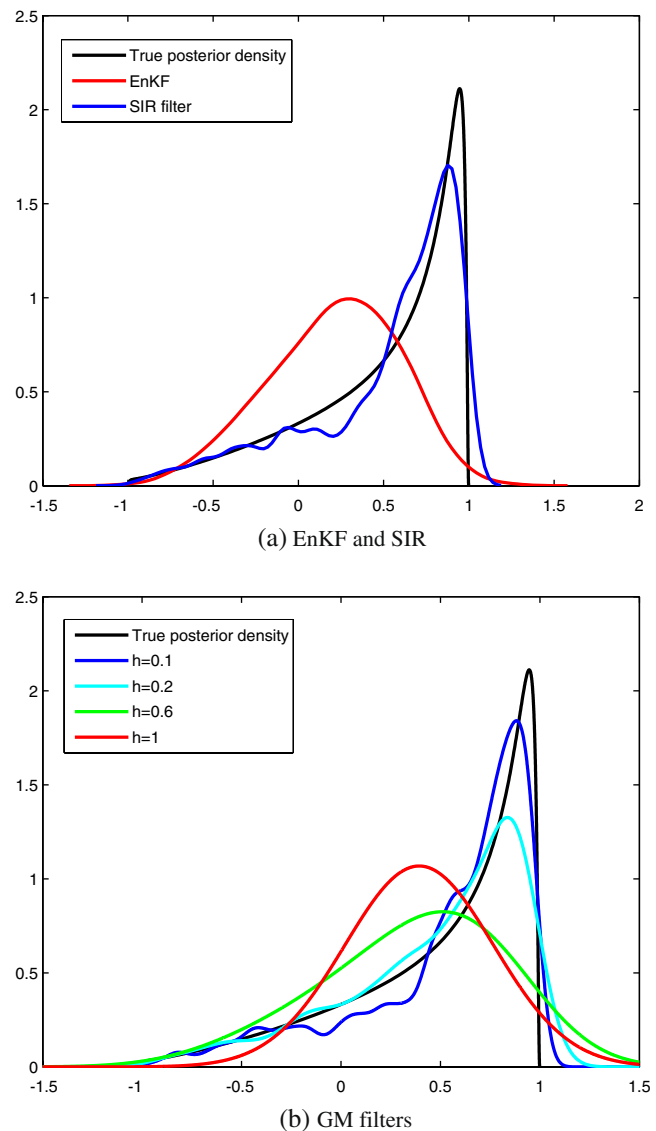


Fig. 2 Density estimates for the Bernoulli model



produces posterior densities that are too close to a Gaussian density for large values of  $h$ .

### 4.2 Lorenz40 model

To study the effect of simultaneously varying  $h$  and  $\alpha$  in a high-dimensional nonlinear system, we choose the Lorenz40 model [16] which is used as a reference example in Sakov and Oke [19].

Our Lorenz40 model consists of 40 coupled ordinary differential equations with cyclic boundary conditions and additive Gaussian noise:

$$\frac{d\mathbf{x}^{(i)}}{dt} = (\mathbf{x}^{(i+1)} - \mathbf{x}^{(i-1)})\mathbf{x}^{(i-1)} - \mathbf{x}^{(i)} + \delta, \quad i = 1, \dots, 40; \tag{4.2}$$

$$\mathbf{x}_0 = \mathbf{x}_{40}, \quad \mathbf{x}_{-1} = \mathbf{x}_{39}, \quad \mathbf{x}_{41} = \mathbf{x}_1. \tag{4.3}$$

The complete system is described by

$$\mathbf{x}(k\delta) = \mathcal{M}(\mathbf{x}((k-1)\delta)) + \eta_k$$

$$\mathbf{y}_k = \mathbf{x}(k\delta) + \epsilon_k, \quad k = 1, 2, \dots$$

where  $\mathcal{M}$  is the standard fourth order Runge Kutta solution of Eq. 4.2,  $\delta = 0.05$ ,  $\eta_i$  is a zero mean Gaussian vector with standard deviation 0.01, and  $\epsilon_i$  is a vector of 40 iid zero mean Gaussian variables with a standard deviation of one. The initial state and the initial ensemble

members are randomly drawn from a Gaussian prior with mean and covariance obtained from the sample mean and covariance of  $10^4$  states from a continuous integration selected at  $k = 1,000, 1,001, \dots, 11 \times 10^3$ . For this experiment, we use the standard number of 100 particles, a typical size in many EnKF applications. Even if the dimension of the system is smaller than the ensemble size, which is typically not the case in geophysical systems, we argue that it is large enough to demonstrate that Gaussian mixture filters is capable of dealing with problems where a standard particle filter collapse.

To reduce the sampling effects, we have run all experiments ten times each. The solutions are presented as the average over the ten runs.

Experiments are performed for different fixed values of the bandwidth  $h$  and the  $\alpha$  parameter. In addition, the adaptive method is implemented where the  $\alpha$  parameter is driven by the data at each assimilation step according to Eq. 3.6.

We also applied the improved SIR filter (see Section 2.2) with  $5 \times 10^5$  particles. The posterior distribution obtained from this run is considered as the true posterior.

#### 4.2.1 Results

As a first measure of the filter performance, we study the average root mean squared error (RMSE) of the

**Table 1** RMSE from the Gaussian mixture filters

$\alpha \backslash h$	0.4	0.5	0.6	0.7	0.8	0.9	1
0.0	3.837 (0.056)	0.739 (0.310)	<b>0.300</b> (0.011)	<b>0.294</b> (0.001)	<b>0.310</b> (0.001)	0.333 (0.001)	0.359 (0.001)
0.1	3.866 (0.060)	0.480 (0.360)	<b>0.290</b> (0.001)	<b>0.293</b> (0.001)	0.312 (0.001)	0.336 (0.001)	0.362 (0.001)
0.2	4.279 (0.040)	3.149 (0.985)	<b>0.299</b> (0.011)	<b>0.295</b> (0.001)	0.313 (0.001)	0.339 (0.001)	0.365 (0.001)
0.3	4.535 (0.014)	4.111 (0.053)	0.441 (0.179)	<b>0.300</b> (0.008)	0.317 (0.001)	0.341 (0.001)	0.367 (0.001)
0.4	4.661 (0.023)	4.386 (0.024)	3.267 (0.898)	0.324 (0.026)	0.318 (0.001)	0.343 (0.002)	0.369 (0.001)
0.5	4.765 (0.032)	4.551 (0.032)	4.150 (0.041)	0.451 (0.157)	0.318 (0.003)	0.343 (0.001)	0.371 (0.001)
0.6	4.765 (0.040)	4.668 (0.030)	4.344 (0.023)	2.019 (0.956)	0.318 (0.007)	0.345 (0.001)	0.370 (0.001)
0.7	4.893 (0.041)	4.725 (0.055)	4.516 (0.032)	3.658 (0.883)	0.323 (0.043)	0.347 (0.004)	0.372 (0.001)
0.8	4.938 (0.041)	4.801 (0.023)	4.694 (0.024)	4.286 (0.032)	1.375 (0.922)	0.360 (0.016)	0.373 (0.001)
0.9	4.977 (0.025)	4.866 (0.028)	4.693 (0.023)	4.429 (0.040)	3.187 (0.914)	0.382 (0.019)	0.375 (0.002)
1.0	5.006 (0.034)	4.907 (0.031)	4.780 (0.021)	4.568 (0.032)	3.589 (1.348)	0.475 (0.142)	0.386 (0.006)

The data in bold correspond to the ten lowest mean values, and the data in italics correspond to the ten largest standard deviations for all the filters

**Table 2** RMSE from the adaptive Gaussian mixture filters and the EnKF

$h$	0.4	0.5	0.6	0.7	0.8	0.9	1	EnKF
RMSE	3.667 (0.116)	0.378 (0.092)	<b>0.289</b> (0.004)	<b>0.294</b> (0.001)	0.313 (0.001)	0.336 (0.001)	0.362 (0.001)	0.644 (0.001)

The data in bold correspond to the ten lowest mean values

40 state estimates obtained from the different filters. The RMSE is calculated as the average over the 10,000 time steps of the root mean squared distance between the true states and the mean filter estimates of the 40 variables.

$$RMSE = \frac{1}{T} \sum_{k=1}^T \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2,$$

$$\hat{\mathbf{x}}_t = \sum_{i=1}^N w_t^i \mathbf{x}_t^i. \tag{4.4}$$

Since we have observations for all the states at each time step, the RMSE from the measurements is theoretically equal to 1. That is if we substitute  $\hat{\mathbf{x}}_t$  with  $\mathbf{y}_t$  in Eq. 4.4 the expectation is equal to one. This gives us a convergence criterion: If the RMSE is greater than one, we have not gained any additional information about the states from the filter. The mean and standard deviations (in parentheses) of the RMSE are presented in Tables 1 and 2, where the ten lowest mean values are marked in green and the ten largest standard deviations are marked in red.

As a measure of how well the filters reproduce the higher-order properties (skewness and kurtosis) of the marginal posterior distributions at  $k = 10^4$ , we calculate the Kullback–Leibler (KL) divergence from the filter approximations to the “true” marginal distributions obtained by the SIR filter with  $5 \times 10^5$  particles

In order to obtain a single index for each filter, we take the average KL divergence over the 40 variables and ten runs. For any of the 40 state variables, the KL divergence from one of the ten runs is calculated as

$$D_{KL} = \mathbf{E}_g \log(g/p)$$

where  $g$  is the posterior density obtained from any of the filters and  $p$  is taken as the true posterior obtained from the SIR filter.

The values were calculated numerically for all values of  $\alpha$  and  $h$  except  $h = 0.4$  since all the filters diverged for this value of  $h$ .

The results of the mean KL distance for the Gaussian mixture filters are presented in Tables 3 and 4 with standard deviations in parentheses below. The ten lowest mean values are marked in green and the ten largest standard deviations are marked in red.

**Table 3** Kullback–Leibler divergence for the Gaussian mixture filters

$\alpha \backslash h$	0.5	0.6	0.7	0.8	0.9	1
0.0	1.187 (0.180)	0.947 (0.222)	0.979 (0.126)	1.368 (0.106)	1.848 (0.134)	2.550 (0.264)
0.1	0.948 (0.258)	<b>0.776</b> (0.115)	0.901 (0.112)	1.319 (0.115)	1.821 (0.181)	2.542 (0.190)
0.2	12.71 (8.309)	<b>0.632</b> (0.092)	0.889 (0.125)	1.221 (0.084)	1.683 (0.105)	2.364 (0.147)
0.3	17.48 (1.885)	<b>0.679</b> (0.094)	0.806 (0.110)	1.097 (0.097)	1.530 (0.160)	2.210 (0.204)
0.4	17.76 (2.494)	11.93 (10.16)	<b>0.654</b> (0.107)	0.954 (0.146)	1.483 (0.067)	2.196 (0.158)
0.5	15.73 (2.244)	17.61 (2.752)	<b>0.656</b> (0.082)	0.908 (0.151)	1.416 (0.107)	2.012 (0.135)
0.6	11.48 (2.261)	15.95 (3.523)	2.644 (6.573)	<b>0.803</b> (0.136)	1.277 (0.193)	1.706 (0.170)
0.7	10.08 (2.273)	13.66 (1.448)	15.36 (8.145)	<b>0.773</b> (0.147)	1.081 (0.193)	1.790 (0.192)
0.8	9.680 (2.765)	11.65 (2.244)	16.74 (3.123)	<b>0.659</b> (0.117)	0.968 (0.101)	1.576 (0.141)
0.9	7.294 (3.394)	11.94 (3.476)	15.75 (2.244)	7.163 (8.905)	0.876 (0.143)	1.412 (0.164)
1.0	7.863 (2.582)	10.31 (0.104)	14.20 (2.514)	14.23 (7.652)	0.824 (0.149)	1.395 (0.255)

The data in bold correspond to the ten lowest mean values, and the data in italics correspond to the ten largest standard deviations for all the filters

**Table 4** Kullback–Leibler divergence for the adaptive Gaussian mixture filters and the EnKF

$h$	0.5	0.6	0.7	0.8	0.9	1	EnKF
KL	<b>0.781</b> (0.195)	<b>0.703</b> (0.104)	0.991 (0.164)	1.277 (0.090)	1.817 (0.148)	2.514 (0.216)	6.699 (0.438)

The data in bold correspond to the ten lowest mean values

The lowest RMSE was achieved with  $h = 0.6$  with the adaptive  $\alpha$ . However, the smallest KL distance, i.e., the posterior density closest to the “true” density, was obtained for  $h = 0.6$  and  $\alpha = 0.2$  with a value of 0.6322. With the adaptive  $\alpha$ , the smallest KL distance was 0.7028 for  $h = 0.6$ . The Gaussian mixture filter, i.e.,  $\alpha = 1$  achieved its minimum KL distance at 0.8238 for  $h = 0.9$ .

Tables 1 and 3 confirm what we discussed in Section 3, as  $\alpha$  increases the precision of the filter increases until  $\alpha$  becomes too large and the filter collapses due to the dimensionality of the problem. From Table 1, we see that for the smallest values of  $\alpha$  where the mean value is greater than one, the standard deviation is high. The reason is that the filter is unstable, i.e., some of the runs converge and some diverge.

The adaptive  $\alpha$  was constructed in such a way that a collapse is impossible, and this is reflected in the standard deviations in Table 4.

While the adaptive Gaussian mixture filter seems to slightly improve upon the Gaussian mixture filter, both these methods are superior to the standard EnKF which produced a Kullback–Leibler distance of 6.6986.

From the results, we see that with the new parameter  $\alpha$ , both fixed and adaptive, we are able to produce good results with a smaller  $h$  than the Gaussian mixture filter ( $\alpha = 1$ ), where we had to keep  $h$  as large as 0.9 in order to avoid divergence.

## 5 Summary

In this paper, we have tried to combine on a sliding scale the EnKF and particle filters. We have shown in numerical examples that the optimal filter, in the RMSE and KL sense, lies somewhere in between.

We added a new parameter to the Gaussian mixture filter and saw that if we add model noise in the forward run, we are able to completely recover the behavior of the particles in the SIR filter ( $\alpha = 1$ ,  $h = 0$ ) and by adding measurement noise in the update step, the filter is reduced to the ordinary EnKF ( $\alpha = 0$ ,  $h = 1$ ). The proposed filter can therefore be viewed as a synthesis of the particle filter and the EnKF. By making the tuning

parameter,  $\alpha$ , adaptive, the relative influence of both the methods at each time point is determined by the data.

The main limitation of the EnKF is its inability to estimate non-Gaussian posterior distributions. In Section 2.2 and 2.3, we discussed alternative filters to solve this problem in terms of sequential Monte Carlo methods. Although these methods are consistent, their estimates rely on importance weights. Unless the sample size grows exponentially with the dimension of the system, the largest importance weight converges to one in probability causing the filter to collapse [3].

To solve the curse of dimensionality problem, a Gaussian mixture approximation of the state distribution was made in Section 2.4 [12]. The covariance of the distribution, determining the size of the Kalman update, depends on a bandwidth parameter  $h$ . By increasing  $h$ , thereby making the assumption of a large model error, the Kalman update increased and the importance weights become more equal; hence, the filter behavior is closer to the EnKF.

We saw in the Bernoulli example that the posterior densities from the Gaussian mixture filter became too symmetric for large values of  $h$ . We therefore introduced a new parameter  $\alpha$  to the Gaussian mixture filter. The idea behind  $\alpha$  was to use smaller values for  $h$  and at the same time avoid filter divergence due to collapse of the weights. This was verified with the Lorenz40 model, where the filter showed good performance for  $h$  as small as 0.5. In comparison, with  $\alpha = 1$ , the filter converged only for  $h = 0.9$  and 1.

Instead of trying to find an optimal fixed  $\alpha$ , we decided to let  $\alpha$  adapt to the data at each assimilation time via the estimated effective sample size, hence the name adaptive Gaussian mixture filter.

The advantage of the adaptive  $\alpha$  is the computational time saved by running the filter only once, not having to test for different values of  $\alpha$ . The second advantage is the fact that the adaptive  $\alpha$  changes in time making it possible to increase the precision of the filter as more observations become available.

In addition, an adaptive tuning parameter also adjust to an increased number of observations which may ease the problem and thereby allow more influence from the particle filter part.

We showed that a filter degeneracy due to collapse of the weights was avoided as the estimated effective ensemble size was always above 80% with the adaptive  $\alpha$ . Thus, we have reduced variance of the filter by introducing bias.

However, to make this filter as efficient as the EnKF, we need to find a good way of choosing the design parameter  $h$ . Also, in order to conclude that this method improves the EnKF for high-dimensional filtering, we need to apply it to models where the dimension of the problem is much larger than the ensemble size. In the Lorenz40 example, we have a linear relation between the measurements and all the states as we observe all the states. The linear Kalman update will in this case give a good, although not correct, approximation of the state of the system. For nonlinear models where we do not observe all the states, the effect of a small linear update could be even more significant. Since Gaussian mixture filters include a resampling step, the results will be affected by how often resampling is performed. In our work, we resampled when the estimated effective ensemble size was below a threshold  $N_c$  which was fixed. It is also possible to resampling at fixed time instances. All the above will be investigated in future work.

**Acknowledgements** The authors would like to thank Dr. Dean Oliver, University of Oklahoma and Dr. Trond Mannseth, CIPR for valuable discussions and comments. A special thanks is given to Dr. Pavel Sakov, NERSC for Matlab scripts and samples used in the Lorenz40 example.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

**Appendix A: A low rank decomposition of the covariance matrices**

For high-dimensional systems, we need a low-rank decomposition of  $\mathbb{P}_t$  at each time  $t$  as calculations involving the associated full size matrix is impossible. Following [12], we decompose  $\mathbb{P}_0$  as  $\mathbb{L}_0\mathbb{U}_0\mathbb{L}_0^T$  in such a way that  $\mathbb{U}_0$  is a full rank square matrix with

$$\mathbb{L}_0 = \mathbb{X}_0\mathbb{T}$$

$$\mathbb{U}_0 = (\mathbb{T}^T\mathbb{W}^{-1}\mathbb{T})^{-1},$$

where  $\mathbb{X}_0$  is the vector of particles  $\mathbb{X}_0 = [\mathbf{x}_0^1 \dots, \mathbf{x}_0^N]$ ,  $\mathbb{W}$  is a diagonal matrix with the weights, and  $\mathbb{T}$  is a  $N \times N - 1$  matrix of the form

$$\mathbb{T} = \begin{bmatrix} \mathbb{1}_{N-1} \\ \mathbf{0} \end{bmatrix} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_{N-1}^T,$$

where  $\mathbf{1}_N$  is an  $N$  dimensional vector of ones. The  $\mathbb{T}$  matrix has zero columns sum and rank  $N - 1$ .

The prediction step consists of running each particle forward in time. The covariance matrix  $\mathbb{P}_t$  is updated so that the new Gaussian mixture approximation naturally follows from the dynamics of the system. As in the EnKF, the forward integration of the model is used to represent  $\mathbb{P}_t$  to avoid linearization of the model,

$$\begin{aligned} \mathbb{P}_t &= \mathcal{M}_t(\hat{\mathbb{X}}_{t-1})\mathbb{T}\mathbb{U}_{t-1}\mathbb{T}^T\mathcal{M}_t^T(\hat{\mathbb{X}}_{t-1}) + \mathbb{Q}_t \\ &= [\mathcal{M}_t(\hat{\mathbf{x}}_{t-1}^1) \dots \mathcal{M}_t(\hat{\mathbf{x}}_{t-1}^N)]\mathbb{T}\mathbb{U}_{t-1}\mathbb{T}^T \\ &\quad \times [\mathcal{M}_t(\hat{\mathbf{x}}_{t-1}^1) \dots \mathcal{M}_t(\hat{\mathbf{x}}_{t-1}^N)]^T + \mathbb{Q}_t \\ &= \mathbb{X}_t\mathbb{T}\mathbb{U}_{t-1}\mathbb{T}^T\mathbb{X}_t^T + \mathbb{Q}_t = \mathbb{L}_t\mathbb{U}_{t-1}\mathbb{L}_t^T + \mathbb{Q}_t. \end{aligned}$$

In order to keep the  $\mathbb{L}\mathbb{U}\mathbb{L}^T$  decomposition, we also need a low rank approximation of  $\mathbb{Q}_t$ . This can be done by an ensemble representation.

The expression for the posterior covariance given by Eq. 2.15 can be decomposed into  $\tilde{\mathbb{L}}_t\mathbb{U}_t\tilde{\mathbb{L}}_t^T$  [12], where

$$\tilde{\mathbb{L}}_t = \tilde{\mathbb{X}}_t\mathbb{T} \tag{A.1}$$

$$\mathbb{U}_t^{-1} = \mathbb{B}_t^T\bar{\mathbb{U}}_t^{-1}\mathbb{B}_t, \tag{A.2}$$

$$\bar{\mathbb{U}}_t = [\mathbb{U}_{t-1} + (\mathbb{H}_t\mathbb{L}_t)^T\mathbb{R}_t^{-1}\mathbb{H}_t\mathbb{L}_t]^{-1}, \tag{A.3}$$

$$\mathbb{B}_t = \mathbb{1}_{N-1} + \bar{\mathbb{U}}_t(\mathbb{H}_t\mathbb{L}_t)^T\mathbb{R}_t^{-1}[\mathbf{y}_t - \mathbb{H}_t\mathbf{x}_t^1, \dots, \mathbf{y}_t - \mathbb{H}_t\mathbf{x}_t^N]\mathbb{T}. \tag{A.4}$$

**Appendix B: Algorithm for the adaptive Gaussian mixture filter**

Initialization  $t = 0$

Select  $h$  and  $N_c$ .

$$\mathbb{T} = \begin{bmatrix} \mathbb{1}_{N-1} \\ \mathbf{0} \end{bmatrix} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_{N-1}^T.$$

Define  $\mathbb{X}_0$  by  $\hat{\mathbf{x}}_0^i, \quad i = 1 \dots N$ .

$$\hat{\mathbb{L}}_0 = \hat{\mathbb{X}}_0\mathbb{T}.$$

$$w_0^i = N^{-1}, \quad i = 1 \dots N$$

$$\mathbf{w}_0 = (w_0^i)_{i=1}^N.$$

$$\mathbb{W}_0 = \text{diag}(\mathbf{w}_0\mathbf{1}^T).$$

$$\mathbb{U}_0 = (\mathbb{T}^T\mathbb{W}_0^{-1}\mathbb{T})^{-1}.$$

$$\hat{\mathbb{P}}_0 = h^2\hat{\mathbb{L}}_0\mathbb{U}_0\hat{\mathbb{L}}_0^T.$$

Evolution  $t \geq 1$

$$\mathbb{X}_t = \mathcal{M}(\hat{\mathbb{X}}_{t-1}).$$

$$\mathbb{L}_t = \mathbb{X}_t \mathbb{T}.$$

$$\mathbb{P}_t = \mathbb{L}_t \mathbb{U}_{t-1} \mathbb{L}_t^T.$$

$$\Sigma_t = \mathbb{H}_t \mathbb{P}_t \mathbb{H}_t^T + \mathbb{R}_t.$$

$$\mathbb{K}_t = \mathbb{P}_t \mathbb{H}_t^T \Sigma^{-1}.$$

$$\tilde{\mathbb{X}}_t = \mathbb{X}_t + \mathbb{K}_t (\mathbf{y}_t \mathbf{1} - \mathbb{H}_t \mathbb{X}_t).$$

$$\tilde{\mathbb{L}}_t = \tilde{\mathbb{X}}_t \mathbb{T}.$$

$$\tilde{\mathbb{U}}_t = [\mathbb{U}_{t-1} + (\mathbb{H}_t \mathbb{L}_t)^T \mathbb{R}_t^{-1} \mathbb{H}_t \mathbb{L}_t]^{-1}.$$

$$\mathbb{B}_t = \mathbb{I}_{N-1} + \tilde{\mathbb{U}}_t (\mathbb{H}_t \mathbb{L}_t)^T \mathbb{R}_t^{-1} [\mathbf{y}_t \mathbf{1} - \mathbb{H}_t \mathbb{X}_t] \mathbb{T}.$$

$$\mathbb{U}_t = (\mathbb{B}_t^T \tilde{\mathbb{U}}_t^{-1} \mathbb{B}_t)^{-1}.$$

$$\tilde{\mathbb{P}}_t = \tilde{\mathbb{L}}_t \mathbb{U}_t \tilde{\mathbb{L}}_t^T.$$

$$\tilde{w}_t^i = w_{t-1}^i \Phi(\mathbf{y}_t - \mathbb{H}_t \mathbf{x}_t^i, \Sigma_t), \quad i = 1 \dots N.$$

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{j=1}^N \tilde{w}_t^j}, \quad i = 1 \dots N.$$

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w_t^i)^2}.$$

$$\alpha = \hat{N}_{\text{eff}} N^{-1}.$$

$$w_t^i = \alpha w_t^i + (1 - \alpha) N^{-1}.$$

**if**  $\hat{N}_{\text{eff}} < N_c$  **then**

$$\mathbf{I} = (1 \dots N).$$

$$\mathbf{w}_t = (w_t^i)_{i=1}^N$$

$\mathbf{J} = \text{randsample}(\mathbf{I}, \mathbf{w}_t, \text{replace} = \text{TRUE}).$

$$\xi_t^i \sim \mathcal{N}(0, 1), \quad i = 1 \dots N.$$

$$\hat{\mathbf{x}}_t^i = \tilde{\mathbf{x}}_t^{j(i)} + \tilde{\mathbb{P}}_t^{1/2} \xi_t^i, \quad i = 1 \dots N$$

$$\mathbb{U}_t = \mathbb{U}_0.$$

$$\hat{\mathbb{L}}_t = \hat{\mathbb{X}}_t \mathbb{T}.$$

$$\hat{\mathbb{P}}_t = h^2 \hat{\mathbb{L}}_t \mathbb{U}_t \hat{\mathbb{L}}_t^T.$$

**else**

$$\hat{\mathbb{X}}_t = \tilde{\mathbb{X}}_t.$$

$$\hat{\mathbb{P}}_t = \tilde{\mathbb{P}}_t.$$

**end if**

$$t = t + 1.$$

The **randsample** function samples a new set of indices from the set  $1 \dots N$  with replacement such that  $P(j = i) = w^i$ . This corresponds to sampling integers from a multinomial distribution.

### References

1. Anderson, B.D., Moore, J.B.: Optimal Filtering. Dover, New York (2005)
2. Apte, A., Hairer, M., Stuart, A.M., Voss, J.: Sampling the posterior: an approach to non-Gaussian data assimilation. *Physica D* **230**, 50–64 (2007)

3. Bengtsson, T., Bickel, P., Li, B.: Curse-of-dimensionality revisited: collapse of particle filter in very large scale systems. *IMS Collect.* **2**, 316–334 (2008)
4. Bengtsson, T., Snyder, C., Nychka, D.: Toward a nonlinear ensemble filter for high-dimensional systems. *J. Geophys. Res.* **108**, 35–45 (2003)
5. Chen, R., Liu, J.S.: Mixture Kalman filters. *J. R. Stat. Soc., Ser. B Stat. Methodol.* **60**, 493–508 (2000)
6. Doucet, A., de Freitas, N., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer, New York (2001)
7. Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **10**, 197–208 (2000)
8. Evensen, G.: Data Assimilation: The Ensemble Kalman Filter. Springer, New York (2007)
9. Gneiting, T., Raftery, A., Westveld III, A., Goldman, T.: Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Mon. Weather Rev.* **133**(5), 1098–1118 (2005)
10. Gordon, N.: Bayesian Methods for Tracking. Ph.D. thesis, University of London (1993)
11. Hoteit, I., Pham, D.-T., Korres, G., Triantafyllou, G.: Particle Kalman filtering for data assimilation in meteorology and oceanography. In: 3rd WCRP International Conference on Reanalysis, p. 6. Tokyo, Japan (2008)
12. Hoteit, I., Pham, D.-T., Triantafyllou, G., Korres, G.: A new approximative solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. *Mon. Weather Rev.* **136**, 317–334 (2008)
13. Kong, A., Liu, J., Wong, W.: Sequential imputations and Bayesian missing data problems. *J. Am. Stat. Assoc.* **89**(425), 278–288 (1994)
14. Kotecha, J.H., Djurić, P.M.: Gaussian sum particle filtering. *IEEE* **51**(10), 2602–2612 (2003)
15. Liu, J., West, M.: Sequential Monte Carlo Methods in Practice, pp. 197–223. Springer, New York (2001)
16. Lorenz, E.N., Emanuel, K.A.: Optimal sites for supplementary weather observations: simulations with a small model. *J. Atmos. Sci.* **55**, 399–414 (1998)
17. Mandel, J., Beezley, J.D.: An ensemble Kalman-particle predictor-corrector filter for non-Gaussian data assimilation. In: ICCS 2009: Proceedings of the 9th International Conference on Computational Science, pp. 470–478. Springer, Berlin (2009). ISBN 978-3-642-01972-2
18. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer, New York (2004)
19. Sakov, P., Oke, P.R.: Implications of the form of the ensemble transformation in the ensemble square root filters. *Mon. Weather Rev.* **136**, 1042–1053 (2008)
20. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York (1986)
21. Snyder, C., Bengtsson, T., Bickel, P., Anderson, J.: Obstacles to high-dimensional particle filtering. *Mon. Weather Rev.* **136**, 4629–4640 (2008)
22. Stordal, A.: Sequential Monte Carlo Methods for Nonlinear Bayesian Filtering. M.Sc. thesis, Department of Mathematics, University of Bergen (2008)
23. Titterton, D.: Statistical Analysis for Finite Mixture Distributions. Wiley, Chichester (1985)