NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Shrinked $(1-\alpha)$ ensemble Kalman filter and α particle filter

by

Rezaie, J. and Eidsvik, J.

PREPRINT
STATISTICS NO. 6/2011

NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL http://www.math.ntnu.no/preprint/statistics/2011/S6-2011.pdf
Javad Rezaie has homepage: http://www.math.ntnu.no/∼rezaie
E-mail: rezaie@stat.ntnu.no
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Shrinked $(1-\alpha)$ ensemble Kalman filter and α particle filter

Javad Rezaie and Jo Eidsvik
Department of Mathematical Sciences, NTNU, Norway

State estimation in high dimensional systems remains a challenging part of real time analysis. The ensemble Kalman filter addresses this challenge by using Gaussian approximations constructed from a number of samples. This method has been a large success in many applications. Unfortunately, for some cases, Gaussian approximations are no longer valid and the filter does not work so well. In this paper we use the idea of the ensemble Kalman filter together with the more theoretical particle filter. We outline a Gaussian mixture approach based on shrinking the predicted samples to overcome sample degeneracy, while maintaining non-Gaussian nature. A tuning parameter determines the degree of shrinkage. The computational cost is similar to the ensemble Kalman filter. We compare several filter methods on three different cases, a target tracking model, the Lorenz 40 model, and a reservoir simulation example conditional on seismic and electromagnetic data.

Keywords: Sequential updating, Filtering, EnKF, reservoir simulation, statistics

1 Introduction

State estimation is an important problem in engineering and science. If we represent the system dynamics (differential or difference equations) in state space form, the measurements are transformed, noisy and an incomplete representation of the system state. Filtering methods extract the probability distribution of the state at every time point, given all measurements until that time. For dynamic systems it is natural to perform the estimation process as soon as new observations arrive. Thus, recursive Bayesian estimation algorithms are the most powerful tool for dealing with filtering problems. This consists of sequentially going forward in time according to two-step routine: i) a forward propagation step using the system dynamics, and ii) an updating step when the new data gets available. Step i) is known as the *prediction problem*, while step ii) is the *filtering problem*.

The celebrated Kalman filter (KF) is the optimal solution to the recursive estimation challenge under certain model restrictions [10]. These assumptions include a linear dynamical and observation model, Gaussian initial conditions, and independent Gaussian process and observation noise. For nonlinear systems, we can use linearization and apply the KF for the resulting system. This is known as the extended Kalman filter (EKF) [8]. If system nonlinearity is high, the EKF may diverge. Sigma point Kalman filters was proposed to overcome some of the weak points of linearized filters [9], [7], [13] and [17]. They have very good performance for small to medium size systems, but for high dimensional systems the computational burden becomes too large. The ensemble Kalman filter (ENKF) uses Monte Carlo realizations along with a Gaussian approximation in the updating step. It has shown very good results for high dimensional systems [3] and [4].

All the algorithms mentioned above have a Gaussian approximation as the key ingredient. Even though the algorithms have been successful for high dimensional non-linear problems, one cannot really justify a Gaussian approximation in these problems, apart from using computational convenience as an argument. Noteably, there are no asymptotic results saying that the ENKF converges to the optimal filtering distribution when the number of samples goes to infinity. The Particle filter (PF) is a Monte Carlo based algorithm which approximates the posterior distribution with weighted samples [1]. Under very weak regularity conditions, the PF converges to the optimal filtering distribution when the sample size goes to infinity. It works well for small dimension systems with general non-Gaussian and non-linear models, but for high dimensional systems it suffers from sample degeneracy, i.e. all samples collapse to one sample. In theory one can overcome this problem by increasing the number of Monte Carlo samples, but this has to increase faster than the system dimension, and for most practical purposes the computational burden becomes too large.

In this paper we outline a method between the ENKF and the PF. We are interested in maintaining the robust properties of the ENKF, while encouraging some good theoretical properties as the sample size increases. The filtering method we propose here fits a Gaussian mixture distribution to the predictive distribution. This mixture is constructed by shrinking the particles towards the overall mean. We use a tuning parameter α to control the degree of shrinkage. The extreme cases are the PF ($\alpha = 1$) and the ENKF ($\alpha = 0$). The covariance of the predictive distribution is controlled by scaling the elementwise covariances of the mixture components. If the main computational cost is the forward propagation, which is often the case in high dimensional applications such as fluid flow simulation, the computation time of our method is at the order of the ENKF.

Some recent publications are similar, but different to the current paper. Sætrom et. al changed ENKF updating schemes based on shrinkage methods known from multivariate linear regression such as partial least square regression and principal component regression and consequently they reduced effects caused by collinear ensemble members [16]. Stordal et. al also combine ENKF with PF using the shrinkage idea on weights in order to increase ENKF performance [15]. Dovera and Della Rossa modify classical ENKF for dealing with multimodal distributions by considering Gaussian mixture models [2]. In [16] they increase ENKF performance but the approach could still experience problems in non-Gaussian distributions. The proposed method in [15] is good for low to medium size but for high dimensional systems they do not propose a method for tuning the filter that works as effectively as ENKF. [2] just consider multimodal problems. We approximate the distributions with a mixture of Gaussian in order to deal with general non-Gaussian cases. Besides by defining an algorithm for tuning the filter parameter we can deal with high dimensional systems.

In Section 2 we define the model assumptions used in this paper. Section 3 outlines the α shrinked ensemble-particle filter. Section 4 provides examples from target tracking, the Lorenz 40 model and a reservoir simulation example.

2 Notation and modeling assumptions

Denote the state variable at time t by $\boldsymbol{x}_t$, and let $\boldsymbol{X}_t = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ be the collection of the state variables from time 1 to the current time t. Further, the observations at time t are denoted $\boldsymbol{y}_t$, and $\boldsymbol{Y}_t = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$ is the collection of observations at this current time step. We assume continuous state and observation variables, i.e. $\boldsymbol{x}_t \in \mathcal{R}^n$ and $\boldsymbol{y}_t \in \mathcal{R}^m$, where the dimensions n and m tend to get large in most modern applications.

We use a state space formulation with the usual conditional independence assumptions. This means that the conditional probability density function of $\boldsymbol{x}_t$, given all previous state and observation variables, only depends on the previous state. Moreover, the conditional distribution of observation $\boldsymbol{y}_t$, given the state at that time and all previous states and observations, only depends on the state at the current time. Mathematically these two assumptions entail that the conditional distributions are $\pi(\boldsymbol{x}_t|\boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1}) = \pi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $\pi(\boldsymbol{y}_t|\boldsymbol{X}_t, \boldsymbol{Y}_{t-1}) = \pi(\boldsymbol{y}_t|\boldsymbol{x}_t)$, respectively. The joint density of $\boldsymbol{Y}_t$ and $\boldsymbol{X}_t$ can then be factorized using the conditional independence assumptions:

$$\pi(\boldsymbol{Y}_t, \boldsymbol{X}_t) = \prod_{i=1}^{t} \pi(\boldsymbol{y}_i|\boldsymbol{x}_i) \prod_{i=2}^{t} \pi(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})\pi(\boldsymbol{x}_1). \tag{1}$$

Here, $\pi(\boldsymbol{x}_1)$ is the specified probability density function of the initial state variable. The joint model is defined once we have specified this initial distribution, along with the density for the dynamical propagation model $\pi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and the likelihood model $\pi(\boldsymbol{y}_t|\boldsymbol{x}_t)$. Depending on the dynamical model and the information content in the data, there will often be a transient phase for small t, revealing the initial conditions.

We next specify our particular assumptions about the forward propagation and the likelihood model. Generically, we let $N(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the Gaussian probability density function of random variable $\boldsymbol{x}$, with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We model the dynamics of the system in the following way:

$$\pi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = N(\boldsymbol{x}_t; \boldsymbol{g}_t(\boldsymbol{x}_{t-1}), \boldsymbol{P}), \tag{2}$$

where the expectation term is defined by a non-linear function $\boldsymbol{g}_t(\cdot)$. This function is usually the computationally hard part, involving a forward propagation of complex physical phenomena. For instance, in a reservoir simulation application, this function consists of numerical solutions of the partial differential equations for fluid flow in porous media. The covariance matrix $\boldsymbol{P}$ provides a correction term for non-modeled physics, and may be a result of using coarse scales in a physical simulator or a compensation for simplified physics, such as a treating some physical properties as fixed in the dynamical model.

We assume a linear (or linearized) likelihood model, with additive Gaussian noise, i.e.

$$\pi(\boldsymbol{y}_t|\boldsymbol{x}_t) = N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{x}_t, \boldsymbol{R}). \tag{3}$$

The matrix $\boldsymbol{H}_t$ is defined by the data acquisition of the problem, while $\boldsymbol{R}$ is the covariance matrix of the measurement noise. Whereas the dynamical model could be very non-linear, the likelihood is assumed linear or close to linear. In a physical application this assumption for the likelihood entails that a measurement equation is well known and can be solved analytically. In the simplest case we have $m = n$ and $\boldsymbol{H}_t = \boldsymbol{I}_n$ indicating that we measure the state directly, with additive noise. Note that we let the function $\boldsymbol{g}_t$ depend on the time variable, and the same holds for the expectation operator $\boldsymbol{H}_t$ in the likelihood. For simplicity, specify the covariance matrices $\boldsymbol{P}$ and $\boldsymbol{R}$ as fixed over time, but this is easily generalized.

The filtering task consists of sequential propagation and updating as we obtain new observations. At time t-1, consider that we have the updated (filtering) distribution of the state given all observations until that time, denoted by the density $\pi(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1, ..., \boldsymbol{y}_{t-1}) = \pi(\boldsymbol{x}_{t-1}|\boldsymbol{Y}_{t-1})$. The one-step prediction density is constructed from the dynamical model

$$\pi(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}) = \int \pi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})\pi(\boldsymbol{x}_{t-1}|\boldsymbol{Y}_{t-1})d\boldsymbol{x}_{t-1}, \tag{4}$$

where the model assumptions simplify the integrand according to $\pi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{Y}_{t-1}) = \pi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. When the new observation $\boldsymbol{y}_t$ is available, we combine the system dynamics and the likelihood in Bayes rule for the updating:

$$\pi(\boldsymbol{x}_t|\boldsymbol{Y}_t) = \frac{\pi(\boldsymbol{y}_t|\boldsymbol{x}_t)\pi(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1})}{\pi(\boldsymbol{y}_t|\boldsymbol{Y}_{t-1})} \propto \pi(\boldsymbol{y}_t|\boldsymbol{x}_t)\pi(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}), \tag{5}$$

where the conditional independence assumption of the data $\boldsymbol{y}_t$ is used. This recursive Bayesian method gives the exact solution to the general filtering problem, but for practical applications we cannot implement it for large systems because we must calculate multi dimensional complicated integrals. Thus, some simplified conditions on the system dynamics and observations have to be considered, inducing some consistent approximations.

A sampling approximation of the filtering distribution can be established using Monte Carlo realizations[5]. Suppose we have B independent and identically distributed samples $x_{t-1}^1, ..., x_{t-1}^B$ from $\pi(\boldsymbol{x}_{t-1}|\boldsymbol{Y}_{t-1})$. The sample approximation to the filtering distribution at time $t-1$ is then

$$\pi(\boldsymbol{x}_{t-1}|\boldsymbol{Y}_{t-1}) = \frac{1}{B}\sum_{b=1}^{B}\delta(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^b), \tag{6}$$

with the Dirac function $\delta(\boldsymbol{x}) = 1$ if $\boldsymbol{x} = 0$, and $\delta(\boldsymbol{x}) = 0$ otherwise. For the prediction step, all B samples are run through the dynamical model, i.e. $\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b)$, $b = 1, ..., B$. In several applications the $\boldsymbol{g}_t(\cdot)$ evaluation is so time consuming that this dynamical model can only be run about $B \sim 100$ times. In the next section we discuss various approaches for recursive updating of this size B sample approximation of the filtering distributions.

3 Shrinked Gaussian mixture filters

The algorithms presented below differ in the construction of a predictive density, and the induced filtering density. We first present a particular PF, which is exact when $B \to \infty$. The PF is represented by a marginalized Gaussian mixture distribution, and denoted the Gaussian mixture Monte Carlo (GMMC) filter. In the practical situation, B cannot get large enough, and this filter will degenerate for high dimensional systems. The ENKF is next presented as a collapsed Gaussian mixture, with all means in the mixture being identical. Finally, we outline the shrinkage idea to overcome the sample degeneracy, while maintaining some asymptotic properties. We approximate the predictive distribution with a mixture of Gaussians whose mean is between that of the PF and the ENKF. A tuning parameter, α, is used to adjust the particles between the two extremes given by the PF and overall mean. We denote the resulting filter by the robustified Gaussian mixture Monte Carlo (RGMMC) filter.

3.1 Gaussian mixture Monte Carlo filter

The prediction formula is given in equation (4). In our model formulation, with the sample approximation in equation (6), this prediction step becomes an integral over a Gaussian weighted with B Dirac functions. The prediction distribution is thus a mixture of B Gaussian densities:

$$\begin{aligned}
\pi \quad (\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}) &= \int N(\boldsymbol{x}_t; \boldsymbol{g}_t(\boldsymbol{x}_{t-1}), \boldsymbol{P})\pi(\boldsymbol{x}_{t-1}|\boldsymbol{Y}_{t-1})d\boldsymbol{x}_{t-1} \\
&= \frac{1}{B}\sum_{b=1}^{B}\int N(\boldsymbol{x}_t; \boldsymbol{g}_t(\boldsymbol{x}_{t-1}), \boldsymbol{P})\delta(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^b)d\boldsymbol{x}_{t-1} \\
&= \frac{1}{B}\sum_{b=1}^{B} N(\boldsymbol{x}_t; \boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b), \boldsymbol{P}) \\
&= \sum_{b=1}^{B}\pi(b|\boldsymbol{Y}_{t-1})\pi(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}, b),
\end{aligned} \tag{7}$$

where $\pi(b|Y_{t-1}) = \frac{1}{B}$ is used to clarify the identically weighted components $b = 1, 2, ..., B$ in the mixture.

The filtering step becomes

$$\begin{aligned}
\pi(\boldsymbol{x}_t|\boldsymbol{Y}_t) &\propto N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{x}_t, \boldsymbol{R})\pi(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}) \\
\pi(\boldsymbol{x}_t|\boldsymbol{Y}_t) &= \sum_{b=1}^{B} N(\boldsymbol{x}_t; \hat{\boldsymbol{x}}_t^b, \boldsymbol{S})\pi(b|\boldsymbol{Y}_t),
\end{aligned} \tag{8}$$

where $\hat{\boldsymbol{x}}_t^b$ and $\boldsymbol{S}$ are the updated mean and covariance matrix, given component b, i.e. $\pi(\boldsymbol{x}_t|\boldsymbol{Y}_t, b)$. This Gaussian density is obtained by the usual KF formula for fixed component b:

$$\begin{aligned} \hat{\boldsymbol{x}}_t^b &= \boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) + \boldsymbol{P}\boldsymbol{H}_t^{'}\boldsymbol{Q}_t^{-1}(\boldsymbol{y}_t - \boldsymbol{H}_t\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b)), \\ \boldsymbol{S}_t &= \boldsymbol{P} - \boldsymbol{P}\boldsymbol{H}_t^{'}\boldsymbol{Q}_t^{-1}\boldsymbol{H}_t\boldsymbol{P}, \quad \boldsymbol{Q}_t = \boldsymbol{H}_t\boldsymbol{P}\boldsymbol{H}_t^{'} + \boldsymbol{R}. \end{aligned} \tag{9}$$

The weights $w_b = \pi(b|\boldsymbol{Y}_t) \propto \pi(\boldsymbol{y}_t|\boldsymbol{Y}_{t-1}, b)\pi(b|\boldsymbol{Y}_{t-1})$, where $\pi(\boldsymbol{y}_t|\boldsymbol{Y}_{t-1}, b)$ is a Gaussian marginalized over $\boldsymbol{x}_t$. Since expression (8) must be a pdf, the weights w_b are required to sum to 1. We get

$$w_b = \frac{N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b), \boldsymbol{Q}_t)}{\sum_{c=1}^{B} N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^c), \boldsymbol{Q}_t)}. \tag{10}$$

The filter proceeds by repeating the following B times:

1. Sample a component b from probability vector $(w_1, ..., w_B)$.
2. Sample $\boldsymbol{x}_t^b \sim N(\boldsymbol{x}_t; \hat{\boldsymbol{x}}_t^b, \boldsymbol{S}_t)$.

A Dirac representation for $\pi(\boldsymbol{x}_t|\boldsymbol{Y}_t)$ is given by the equally weighted B samples. This procedure continues again, from time t to $t+1$, according to the dynamical model $\boldsymbol{g}_t(\cdot)$, just like our starting point in equation (7) at time $t-1$.

The filtering distributions are Gaussian mixtures, while the Monte Carlo step is used to propagate the particles forward in time. Note that the updated Gaussian mixture for $\pi(\boldsymbol{x}_t|\boldsymbol{Y}_t)$ is exact, given the B sample Dirac represention of $\pi(\boldsymbol{x}_{t-1}|\boldsymbol{Y}_{t-1})$. The resulting PF is much more stable than the standard bootstrap filter [1], since we have marginalized over $\boldsymbol{x}_t$ in the weights w_b. In contrast, the bootstrap filter weight the forward propagated particles according to the likelihood, and the weights w_b would become less uniform. Still, the Monte Carlo sample approximation applied in the GMMC filter might not be so good for small B. The weights w_b could still be very non-uniform, at least in high dimensional systems, and sample degeneracy would occur. Clearly, if one $\boldsymbol{H}_t\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b)$ is much closer to the data $\boldsymbol{y}_t$ than the others, the associated w_b in equation (10), while all others are near 0. The mixture is then focused on one component, and the approach underestimates the variability of the distribution. These effects of course depend on the number of particles B, the non-linearity $\boldsymbol{g}_t(\cdot)$, and the covariance matrix $\boldsymbol{Q}_t$. The degeneracy problem is most dominant for large dimensional problems. Heuristically, in high dimension, say with a diagonal $\boldsymbol{Q}_t$, none of the particles $\boldsymbol{H}_t\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b)$ are close to the data $\boldsymbol{y}_t$, but the closest is much closer than the second closest. The squared mismatch distances, summed over m data dimensions, is given by $\sum_{j=1}^{m} Q_{t,jj}^{-1}(y_{jt} - H_j g_{t,j}(x_{t-1}^b))^2$, which will blow up linearly with dimension. Thus, degeneracy occurs fast in high dimension, unless the variances are very large, or the non-linearity works in some favorable way.

One alternative is of course to increase the sample size B. When B goes to infinity, the GMMC filter is the exact solution. However, this sample size B must typically increase faster than the dimension and for some non-linear problems the B cannot be very large because of the computation time of $\boldsymbol{g}_t(\cdot)$. Other tricks that slow down degeneracy are tapering or localization of the covariance matrix, which effectively reduces the dimension.

3.2 Ensemble Kalman filter as a collapsed Gaussian mixture

The ENKF is based on a Gaussian approximation to the predictive distribution $\pi(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1})$. For our model this entails matching the mean and covariance matrix of the predictive Gaussian mixture distribution in equation (7). They are given by

$$\begin{aligned} \bar{\boldsymbol{x}}_t &= \frac{1}{B}\sum_{b=1}^{B} \boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) \\ \bar{\boldsymbol{P}}_t &= \boldsymbol{P} + \frac{1}{B}\sum_{b=1}^{B} (\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)(\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)^{'}, \end{aligned} \tag{11}$$

where we have used the formula for double covariance $Var(X) = E(Var(X|b)) + Var(E(X|b))$, conditioning on the mixture components. The predictive distribution used in an ENKF update becomes

$$\begin{aligned}\hat{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}) &= N(\boldsymbol{x}_t; \bar{\boldsymbol{x}}_t, \bar{\boldsymbol{P}}_t) \\ &= \frac{1}{B}\sum_{b=1}^{B} N(\boldsymbol{x}_t; \bar{\boldsymbol{x}}_t, \bar{\boldsymbol{P}}_t), \end{aligned} \tag{12}$$

which can be regarded as a Gaussian mixture of B terms, all with the same mean and covariance. Thus, this mixture density has collapsed to a single Gaussian.

The updated distribution for the ENKF is

$$\begin{aligned}\hat{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_t) &= \frac{\pi(\boldsymbol{y}_t|\boldsymbol{x}_t)\hat{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1})}{\pi(\boldsymbol{y}_t|\boldsymbol{Y}_{t-1})} \\ \hat{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_t) &\propto N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{x}_t, \boldsymbol{R})N(\boldsymbol{x}_t; \bar{\boldsymbol{x}}_t, \bar{\boldsymbol{P}}_t) \\ \hat{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_t) &= N(\boldsymbol{x}_t; \bar{\boldsymbol{x}}_t + \bar{\boldsymbol{P}}_t\boldsymbol{H}_t'\bar{\boldsymbol{Q}}_t^{-1}(\boldsymbol{y}_t - \boldsymbol{H}_t\bar{\boldsymbol{x}}_t), \bar{\boldsymbol{S}}_t) \\ \bar{\boldsymbol{S}}_t &= \bar{\boldsymbol{P}}_t - \bar{\boldsymbol{P}}_t\boldsymbol{H}_t'\bar{\boldsymbol{Q}}_t^{-1}\boldsymbol{H}_t\bar{\boldsymbol{P}}_t, \\ \bar{\boldsymbol{Q}}_t &= \boldsymbol{H}_t\bar{\boldsymbol{P}}_t\boldsymbol{H}_t' + \boldsymbol{R}. \end{aligned} \tag{13}$$

Sampling from this updated distribution is achieved by drawing B i.i.d Gaussians from $\hat{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_t)$.

The ENKF is robust in the sense that no degeneracy occurs. Viewed as a collapsed mixture of Gaussians, all the weights are constant, equal to $\frac{1}{B}$. All the particles in the predictive distribution have collapsed to the mean, and the components have the same covariance matrix. However, one cannot find theoretical justification for the Gaussian approximation of the prediction density. If this is assumption is reasonable, the filter will perform very well. If the approximation is too far from the true non-Gaussian distribution, the filter will introduce bias and possibly diverge. In high dimensional systems this does not seem to happen so often, and the practical aspects of the ENKF have shown very useful.

3.3 Robustified Gaussian mixture Monte Carlo filter

We now present the robustified Gaussian mixture Monte Carlo (RGMMC) filter, that aims to stabilize the GMMC from Section 3.1, using the ENKF in Section 3.2 as a guide. Define for $b = 1, ..., B$

$$\boldsymbol{z}_t^b = \alpha\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) + (1-\alpha)\bar{\boldsymbol{x}}_t, \tag{14}$$

where $\boldsymbol{z}_t^b$ are shrinked versions of the predictive particles. We can interpret the shrinkage as follows: For the two dimensional case, if $Z = \alpha X_1 + (1-\alpha)X_2; 0 \leq \alpha \leq 1$, then Z is a point between X_1 and X_2 on the straight line which connects X_1 and X_2, see Figure 1. As a result (with a predicted sample point of view), the new predicted sample $\boldsymbol{z}_t^b = \alpha\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) + (1-\alpha)\bar{\boldsymbol{x}}_t$ is a sample on the hyperplane which connects $\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b)$ to $\bar{\boldsymbol{x}}_t$ and the position of this sample is between $\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b)$ and $\bar{\boldsymbol{x}}_t$ on the same hyperplane. The interpretation of equation (14) is then that the predicted particles are shrinked towards the mean. Using $\alpha = 0$, the shrinking is large, and the result is the ENKF. Using $\alpha = 1$, there is no shrinkage, and the result is GMMC filtering.

Associated with the shrinkage we compute a predictive covariance matrix for every mixture component. By construction, we make sure that the total covariance matches that of the Gaussian mixture distribution in equation (7), just like for the ENKF. We want the predictive distribution in the RGMMC to have components $\tilde{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}, b) = N(\boldsymbol{x}_t; \tilde{\boldsymbol{z}}_t^b, \tilde{\boldsymbol{P}}_t)$, for $b = 1, .., B$, where $\tilde{\boldsymbol{P}}_t$ ensures the second order properties of the predictive distribution. Note that for the shrinked variables we have $\boldsymbol{z}_t^b - \bar{\boldsymbol{z}}_t = \alpha(\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)$. By using the formula for double covariance, conditioning on the mixture components, we can ensure that the following holds:

$$\begin{aligned}\tilde{\boldsymbol{P}}_t &+ \alpha^2\frac{1}{B}\sum_{b=1}^{B}(\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)(\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)' \\ &= \boldsymbol{P} + \frac{1}{B}\sum_{b=1}^{B}(\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)(\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)'. \end{aligned} \tag{15}$$

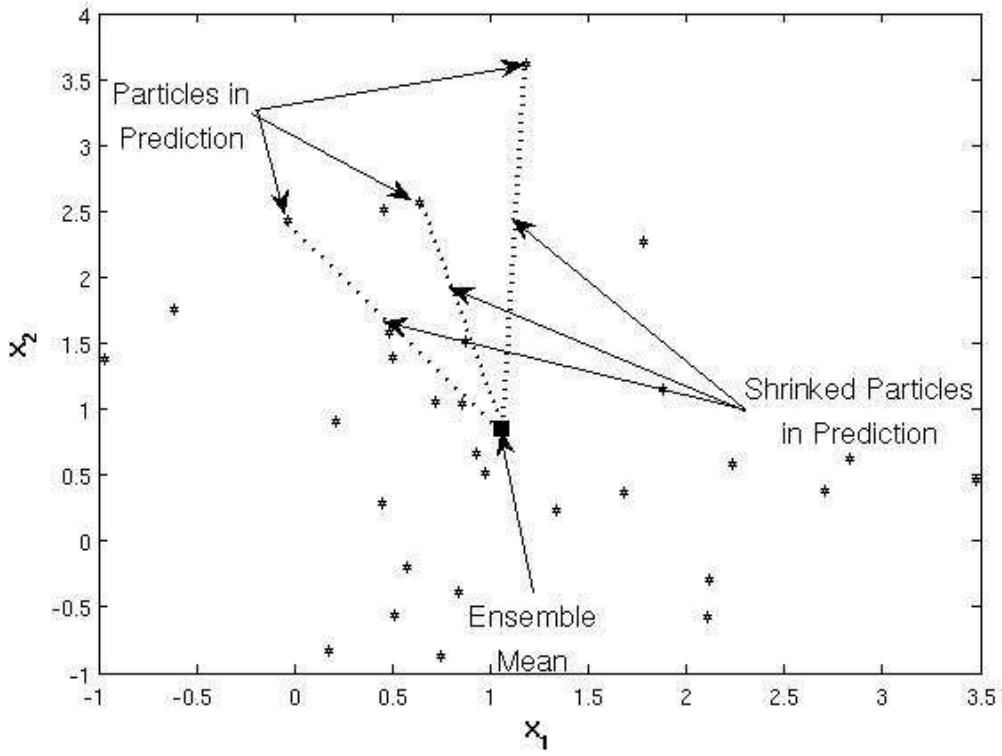

Figure 1: A graphical description of shrinkage $\boldsymbol{z}_t^b = \alpha \boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) + (1-\alpha)\bar{\boldsymbol{x}}_t$

Then, the component-wise covariance matrix for the shrinked mixture becomes

$$\tilde{\boldsymbol{P}}_t = \boldsymbol{P} + (1-\alpha^2)\frac{1}{B}\sum_{b=1}^{B}(\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)(\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b) - \bar{\boldsymbol{x}}_t)'. \tag{16}$$

The predictive distribution for the robustified version is a mixture over these mean and variances, such that

$$\tilde{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}) = \frac{1}{B}\sum_{b=1}^{B} N(\boldsymbol{x}_t; \boldsymbol{z}_t^b, \tilde{\boldsymbol{P}}_t). \tag{17}$$

The updating proceeds as for the GMMC filter, with

$$\begin{aligned} \tilde{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_t) &\propto N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{x}_t, \boldsymbol{R})\tilde{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_{t-1}), \\ \tilde{\pi}(\boldsymbol{x}_t|\boldsymbol{Y}_t) &= \sum_{b=1}^{B} \tilde{w}_b N(\boldsymbol{x}_t; \tilde{\boldsymbol{x}}_t^b, \tilde{\boldsymbol{S}}_t), \end{aligned} \tag{18}$$

where $\tilde{\boldsymbol{x}}_t^b$ and $\tilde{S}_t$ are the updated mean and variance, given particle b, i.e.

$$\begin{aligned} \tilde{\boldsymbol{x}}_t^b &= \boldsymbol{z}_t^b + \tilde{\boldsymbol{P}}_t\boldsymbol{H}_t'\tilde{\boldsymbol{Q}}_t^{-1}(\boldsymbol{y}_t - \boldsymbol{H}_t\boldsymbol{z}_t^b) \\ \tilde{\boldsymbol{S}}_t &= \tilde{\boldsymbol{P}}_t - \tilde{\boldsymbol{P}}_t\boldsymbol{H}_t'\tilde{\boldsymbol{Q}}_t^{-1}\boldsymbol{H}_t\tilde{\boldsymbol{P}}_t,\ \tilde{\boldsymbol{Q}}_t = \boldsymbol{H}_t\tilde{\boldsymbol{P}}_t\boldsymbol{H}_t' + \boldsymbol{R}. \end{aligned} \tag{19}$$

Naturally, all matrices in this expression depend on the shrinkage parameter α. The weights are now given by

$$\tilde{w}_b = \frac{N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{z}_t^b, \tilde{\boldsymbol{Q}}_t)}{\sum_{c=1}^{B} N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{z}_t^c, \tilde{\boldsymbol{Q}}_t)}. \tag{20}$$

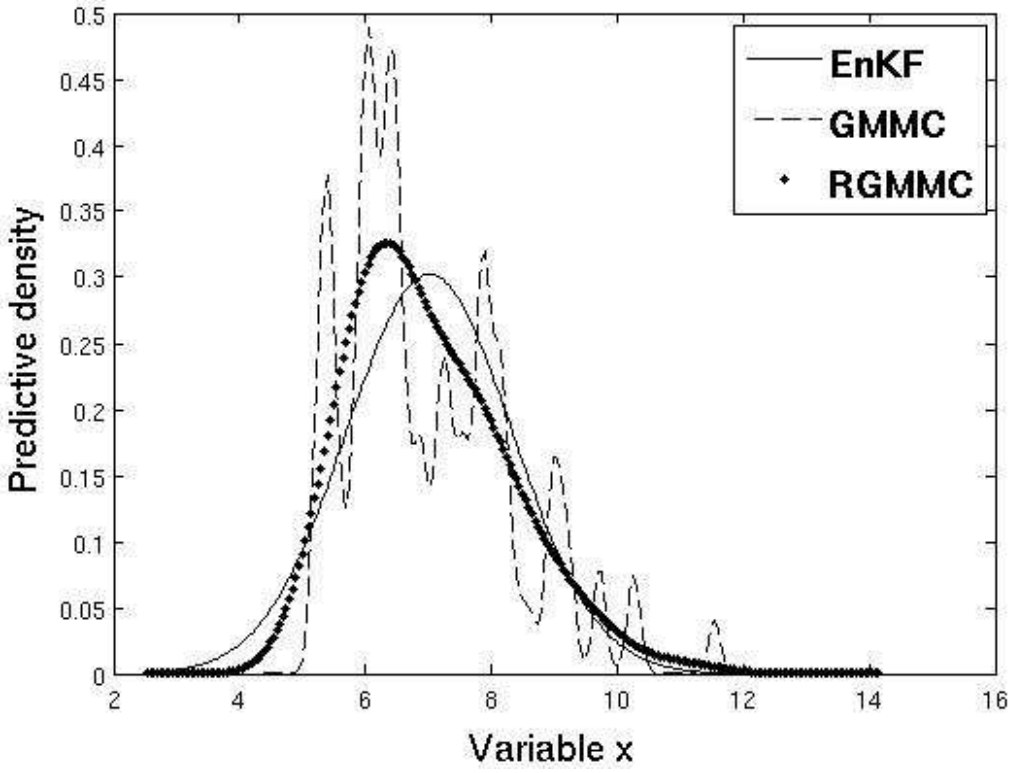

Figure 2: The predictive distribution from ENKF, PF and RGMMC

Figure 2 shows an illustration of the predictive densities of GMMC, ENKF and RGMMC for a particular $0 < \alpha < 1$. The GMMC filter gives a very wiggly predictive density plot, while the ENKF is a Gaussian density. Now, if data matches one of the spikes, the particle associated with this spike would get a very large weight w_b in the GMMC filter. This could cause degeneracy. The RGMMC filter is smoother, and closer to the Gaussian curve representing the ENKF. If data matches one of the spikes in the GMMC representation, the associated increase in the weight for the RGMMC filter, denoted $\tilde{w}_b$, would not get that much larger than the remaining weights.

The robustification occurs because of the shrinkage. First, the $\boldsymbol{z}_t^b$s are more similar than the $\boldsymbol{g}_t(\boldsymbol{x}_{t-1}^b)$. Second, the variance $\tilde{\boldsymbol{Q}}_t > \boldsymbol{Q}_t$, and thus the weights become more uniform. The parameter α gives us some flexibility: Small α entails a solution close to the Gaussian predictive density (like ENKF), while a large α is close to the GMMC solution. One option is to tune α at every time t, i.e. $\alpha \to \alpha_t$. The tuning can be done using the weights $\tilde{w}_b = \tilde{w}_b(\alpha)$. If one weight is too dominant, the effective sample size (ESS) is small, indicating degeneracy. We can start with small α, and increase the α parameter until the effective sample size is above a threshold, for instance a fixed fraction, say $t_1 = \frac{B}{5}$.

Pseudo algorithm 1:

- Set $\alpha = 0$.
- Tol=0.
- Repeat until Tol=1
  1. Compute $\tilde{w} = (\tilde{w}_1, ..., \tilde{w}_B)$.
  2. Compute effective sample size $ESS(\tilde{w})$.
  3. If $ESS(\tilde{w}) < t_1$ set Tol=1 and return $\alpha = \alpha - \epsilon$. Otherwise $\alpha = \alpha + \epsilon$.

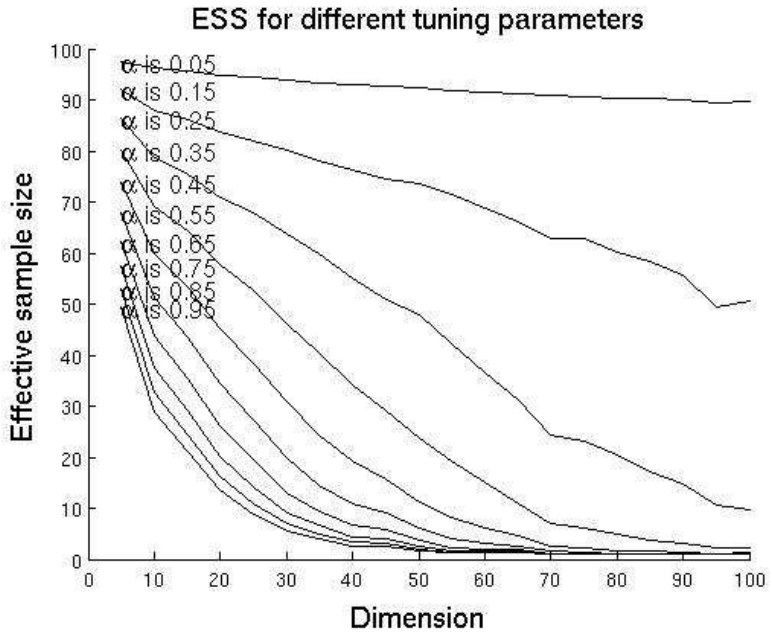

Figure 3: Effective sample size (ESS) for different values of α versus system dimension

The tuning parameter ϵ could be 0.1, or similar. Alternatively, one could start with the GMMC, and decrease α until the ESS goes above the threshold. Trade offs between the two exists too, like starting above 0 in Pseudoalgorithm 1. Or guiding the algorithm by the α from the previous time-step. Here, the ESS is defined by

$$ESS(\tilde{w}) = \sum_{b=1}^{B} [BI(\tilde{w}_b \geq \frac{1}{B}) + B\tilde{w}_b I(\tilde{w}_b \leq \frac{1}{B})]. \tag{21}$$

One could certainly use alternative criteria for tuning the shrinkage parameter α. Since the properties of the PF are well understood, we would like to ensure a similar asymptotic behaviour for the RGMMC filter. As $B \to \infty$, one expects that the tuned α goes towards 1, and then the aymptotic properties are valid because we are in the PF domain. It is thus interesting to study the variability in the weights $\tilde{w}_b$ in equation (20), as a function of B and α. The weights naturally depend on the model $\boldsymbol{g}_t(\cdot)$, the covariance matrices $\boldsymbol{P}$ and $\boldsymbol{R}$, and on the tuning parameter α. Implicitly, the dimensions m and n becomes very important, like discussed above. Since the different particles have different mean values $\boldsymbol{g}_t(x_{t-1}^b)$, it is very hard to derive any general properties for the weights. Moreover, the particles $\boldsymbol{z}_t^b$ are dependent because they are all shrinked towards the common mean. In particular, if $\boldsymbol{x}_0 \sim N(\boldsymbol{x}_0; \underline{0}, k^2\boldsymbol{I}_n)$ and $\boldsymbol{x}_1 = \boldsymbol{x}_0 + N(\boldsymbol{w}; \underline{0}, \boldsymbol{I}_n)$, the shrinked variables $\boldsymbol{z}_1^1, ..., \boldsymbol{z}_1^B$, with $\boldsymbol{z}_1^b = \alpha\boldsymbol{x}_0^b + (1-\alpha)\bar{\boldsymbol{x}}_1$, $\bar{\boldsymbol{x}}_1 = \frac{1}{B}\sum_{b=1}^{B}\boldsymbol{x}_0^b$, have covariance matrix $(1+k^2) \otimes [\boldsymbol{A}(\alpha)\boldsymbol{A}(\alpha)']$, with $B \times B$ covariance matrix $\boldsymbol{A}(\alpha) = \frac{1-\alpha}{B}\mathbf{1}_B\mathbf{1}_B' + \alpha I_B$, where $\mathbf{1}_B$ is a length B vector of ones.

In order to study the weights by simulation, we use a Gaussian model of dimension $n = m$ for both process and data variables. We use one time step $t = 1$ only, generating $\boldsymbol{g}_t(\boldsymbol{x}^b) \sim N(\boldsymbol{x}; \underline{0}, k^2\boldsymbol{I}_n), b = 1, ..., B,$. We set $k = 1$ and $\boldsymbol{x}_1 \sim \frac{1}{B}\sum_{b=1}^{B} N(\boldsymbol{x}_1; \boldsymbol{g}_t(\boldsymbol{x}^b), \boldsymbol{I}_n)$. The likelihood is $\boldsymbol{y}_1 \sim N(\boldsymbol{y}_1; \boldsymbol{x}_1, \boldsymbol{I}_n)$. We study the effect of dimension n varying from 5 to 100. The number of particles is $B = 100$, and we repeat the process 25 times to compute an average ESS under each parameter setting. Figure 3 shows the ESS of the weights, for α parameters between 0.05 to 0.95. For large α the effective sample size decreases quickly, indicating a degeneracy in the weights. When α is close to 0, the updating is more like an ENKF and the ESS remains larger, but an α slightly larger than 0 could give different effect than the ENKF.

4 Simulation

In this part we compare the proposed algorithm with ENKF and GMMC. We consider four different cases; the first two are single-sensor single target-tracking and multi-sensor multi-target tracking examples where the posterior distributions are multimodal. The third one is the famous Lorenz 40 model and the last one is a synthetic reservoir simulator example. For each case we change the value of the tuning parameter and ensemble size for checking the performance of algorithms.

In the examples we compare the different algorithms in terms of mean square error (MSE), continuously ranked probability score (CRPS) and probability distribution coverage. Here, at any time t we have $MSE(t) = \sum_j (\hat{x}_{j,t} - x_{j,t}^{true})^2$, where $\hat{x}_{j,t}$ is the estimated mean of the filtering distribution and the sum is over all n state dimensions. An integrated MSE is achieved by summing out t. Further, CRPS is defined by $CRPS(t) = \sum_j (\hat{F}(y_{j,t}) - I(y_{j,t} < y_{j,t}^{obs}))^2$ [6]. Where $\hat{F}(.)$ is the empirical cumulative predictive distribution for data at time t, given all former data $\boldsymbol{Y}_{t-1}$. If this predictive distribution is very focused on the true observation, the CRPS becomes small. The sum is over all m observation dimension again, an integrated CRPS is obtained by summing over all times t. The probability distribution coverage is defined as the fitted percentiles that covering the true state.

4.1 Tracking targets with bimodal distributions

These examples describe the position and velocity of planes or ships moving in two dimensions. If we imagine a monitoring system for planes or ships, their positions are measured by radar /sonar. The targets move in a dependent pattern, i.e. if one turn, most likely others turn too.

In this simulation we consider two cases, one-sensor one-target and ten-sensor ten-target. We let $\boldsymbol{x}_t = [x_t \ \dot{x}_t \ y_t \ \dot{y}_t]'$ be the state vector of one target. For one target, $(x_t \ y_t)$ is the (north,east) position, and similarly $(\dot{x}_t \ \dot{y}_t)$ is the (north,east) velocity. The absolute velocity is $v_t = \sqrt{\dot{x}_t^2 + \dot{y}_t^2}$, while the target is moving at bearing $\eta_t = arctan(\frac{y_t}{x_t})$.

With constant velocities, a target move in a straight line, and the dynamical model is linear. We consider a situation where a target manoeuvres (30 degrees) to the west whenever the velocity v_t becomes smaller than a threshold c. This model is nonlinear, and the dynamics can be phrased by $\pi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \sim N(\boldsymbol{x}_t; \boldsymbol{g}_t(\boldsymbol{x}_{t-1}), \boldsymbol{P})$. Using a time-step dT , the one target dynamics for large velocity is:

$$\boldsymbol{g}_t(\boldsymbol{x}_{t-1}) = \begin{bmatrix} 1 & dT & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & dT \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \dot{x}_{t-1} \\ y_{t-1} \\ \dot{y}_{t-1} \end{bmatrix} \tag{22}$$

while for small velocity:

$$\boldsymbol{g}_t(\boldsymbol{x}_{t-1}) = \begin{bmatrix} x_{t-1} + dTcos(\eta_t)v_{t-1} \\ cos(\eta_t)v_{t-1} \\ y_{t-1} + dTsin(\eta_t)v_{t-1} \\ sin(\eta_t)v_{t-1}]' \end{bmatrix} \tag{23}$$

$$\eta_t = \frac{\pi}{6} + \eta_{t-1}, \qquad If \quad v_{t-1} < c \tag{24}$$

Thus, bearing η_t of one target at time t, changes westward, whenever the absolute velocity is small. This has effect on the north and east velocity, whereas the absolute velocity $v_t = v_{t-1}$ remains the same, on expectation. As a consequence, the predictions of the north and east positions will tend to be skewed or multimodal, when the distribution for velocity is near the critical velocity c.

The process noise covariance matrix is $\boldsymbol{P} = diag([0.5^2, \quad 2^2, \quad 0.5^2, \quad 2^2])$ and initial conditions are drawn from $N(\boldsymbol{x}_0; \boldsymbol{\mu}_0, \boldsymbol{P}_0)$ where $\boldsymbol{\mu}_0 = [1000, \quad 75, \quad 1000, \quad 75]'$ and $\boldsymbol{P}_0 = 100\boldsymbol{P}$.

We observe the north and east poition at every time point, with Gaussian additive noise. Thus, the likelihood model for position data is linear and can be phrased by $\pi(\boldsymbol{y}_t|\boldsymbol{x}_t) \sim N(\boldsymbol{y}_t; \boldsymbol{H}_t\boldsymbol{x}_t, \boldsymbol{R})$ where $\boldsymbol{R} = diag([5^2 \quad 5^2])$

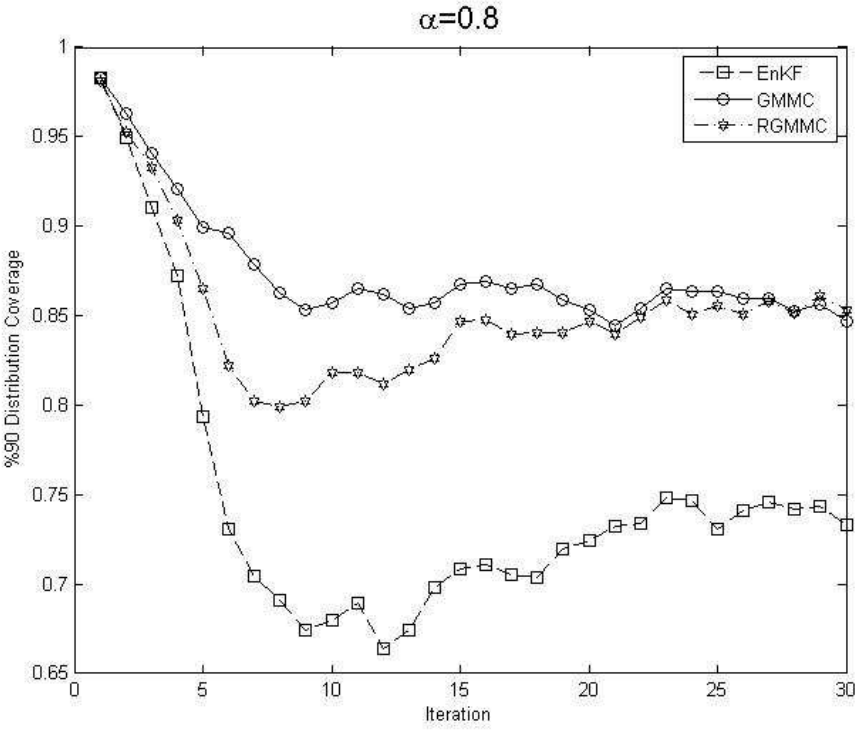

Figure 4: One-sensor one-target, distribution Coverage for $\alpha = 0.8$

and:

$$\boldsymbol{H}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{25}$$

One-sensor one-target case

In this case we assume that we have one moving target and one radar/sonar sensor measures the target position in Cartesian plane.

The main goal of this example is evaluating the performance of each filter for estimating a general distribution when the system dimension is low (system dimension is n=4 for this case) and there is no sign of sample degeneracy. The number of ensemble members is $B = 500$ and we repeat the simulation for 500 replicates in order to reduce Monte-Carlo error. We predict that GMMC should be the best, because for low dimensional systems it can approximate general filtering distributions without sample degeneracy.

Table 1 and Figures 4-5 present the simulation results for a α. We see that our prediction about GMMC is correct and it is the best in terms of MSE, CRPS and probability distribution coverage. When α is close to 1 RGMMCs performance is close to GMMC. This result is natural since it is close to GMMC for large values of α. By looking at Table 1, we also see that when $\alpha \approx 0$ RGMMCs estimation accuracy is close to ENKF. This result is also predictable because the shrinked samples converges to the predicted ensemble mean.

Figure 4 tells us that the percentile coverage distribution starts from a high value (98%) then reduces rapidly. The high initial coverage is caused by the initial state variables. It is noticeable that after some time steps the percentile coverage for GMMC and RGMMC reduce with a lower rate than ENKF because the posterior distribution goes away from Gaussian distribution and ENKF could not approximate posterior distribution very well.

Figure 5 shows the MSE and associated confidence intervals, which are calculated based on bootstraping. GMMC has the lowest trend. We can see similar results on MSE and CRPS trends. Simulation results show that for $0 < \alpha < 1$, RGMMC is some thing between ENKF and GMMC.

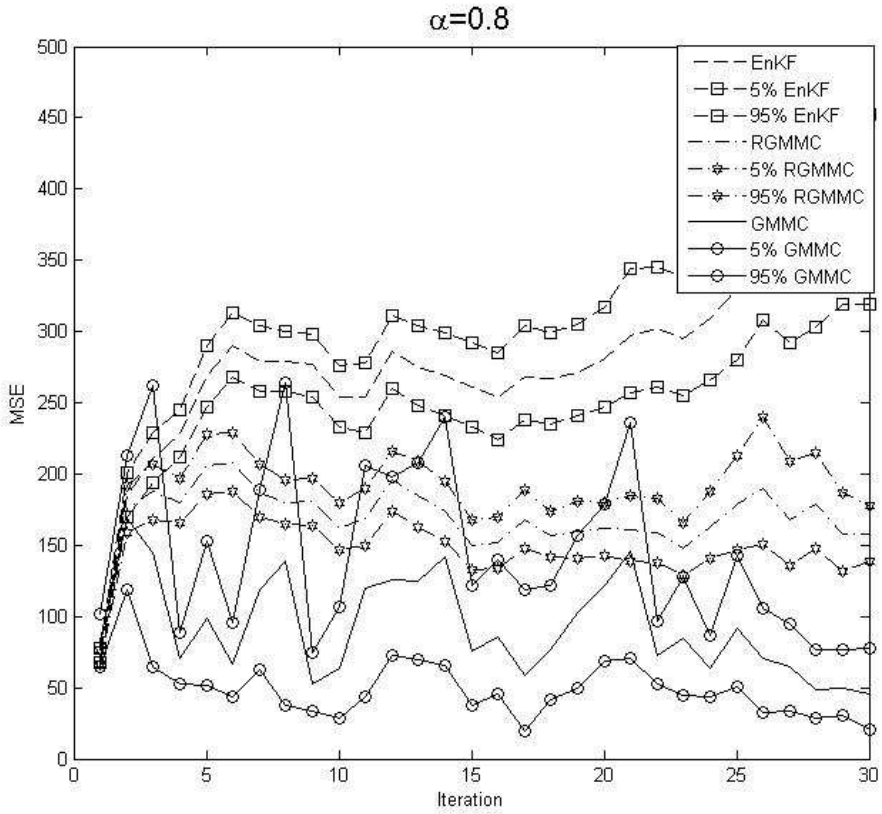

Figure 5: One-sensor one-target, MSE for $\alpha = 0.8$

Table 1: Comparing different filters for one-sensor one-target case

Values	ENKF	GMMC	RGMMC
α			0.00001
CRPS(STD)	144(1.5)	140(1.4)	144(1.6)
MSE(STD)	281(26)	87(6.5)	282(26)
Coverage(%)	75	87	75
α			0.2
CRPS(STD)	-	-	143(1.6)
MSE(STD)	-	-	272(26)
Coverage(%)	-	-	74
α			0.5
CRPS(STD)	-	-	140(1.4)
MSE(STD)	-	-	230(14)
Coverage(%)	-	-	76
α			0.8
CRPS(STD)	-	-	131(1.3)
MSE(STD)	-	-	164(8)
Coverage(%)	-	-	85
α			0.99
CRPS(STD)	-	-	140(1.5)
MSE(STD)	-	-	83(7)
Coverage(%)	-	-	89

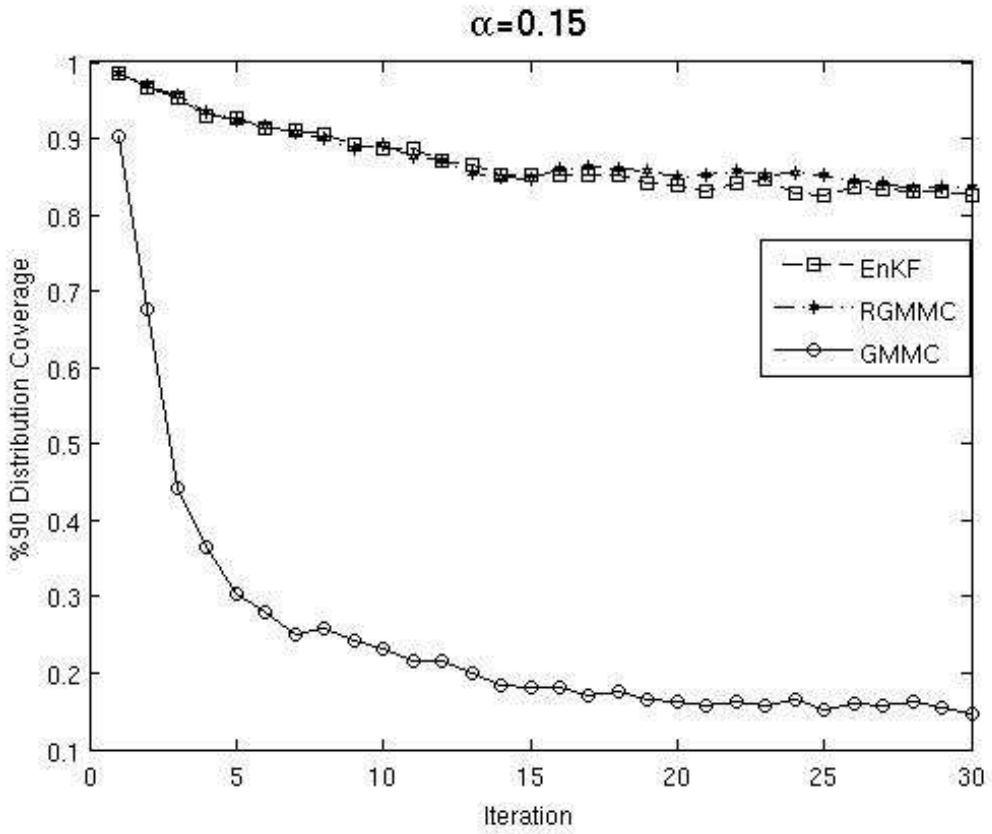

Figure 6: Ten-sensor ten-target, distribution Coverage for $\alpha = 0.15$

Ten-sensors ten-targets case

In this case we increase the system dimension by increasing the number of targets and sensors. Now, we are going to check the proposed method for dealing with sample degeneracy in high dimensional system. We know that the GMMC suffers from sample degeneracy when the system dimension is high and we predict that it should have the worst performance for this case. We also know that ENKF tend to work well in high dimensional systems. In Table 2 and Figures 6-7 we see that the GMMC now has the worst performance as we predicted, while ENKF works well. We see that the proposed method works well for this case without sample degeneracy.

Figure 6 shows that the percentile coverage for the GMMC is high at the begining because initial samples are independent and the distributions are close to Gaussian distribution, but after some time steps it goes down rapidly for the GMMC because of sample collapse (the ESS is low). We have a similar behaviour for MSE. For small to medium values of α the performance of the RGMMC is as good as the ENKF and, according to Table 2, for some values (such as $\alpha = 0.4$) RGMMC is the best in the MSE sense. Besides, Table 2 shows that the RGMMC is close to the ENKF for a wide range of α ($0 < \alpha < 0.8$) and after that it converges to the GMMC rapidly. As we said before, RGMMC is some thing between ENKF and GMMC for different values of α. Table 2 confirms this claim.

4.2 Lorenz 40 model

Lorenz40 model is a highly nonlinear model where its state dimension is $n = 40$ [14]. It consists of 40 ordinary differential equations with cyclic boundary condition as follows:

$$\begin{aligned} \dot{w}_i &= (w_{i+1} - w_{i-2})w_{i-1} - w_i + 8, \quad i = 1, ..., 40; \\ w_0 &= w_{40}, \quad w_{-1} = w_{39}, \quad w_{41} = w_1. \end{aligned} \tag{26}$$

This model is discretized by the standard fourth order Runge-Kutta algorithm and the system states $\boldsymbol{x}_t$ relate

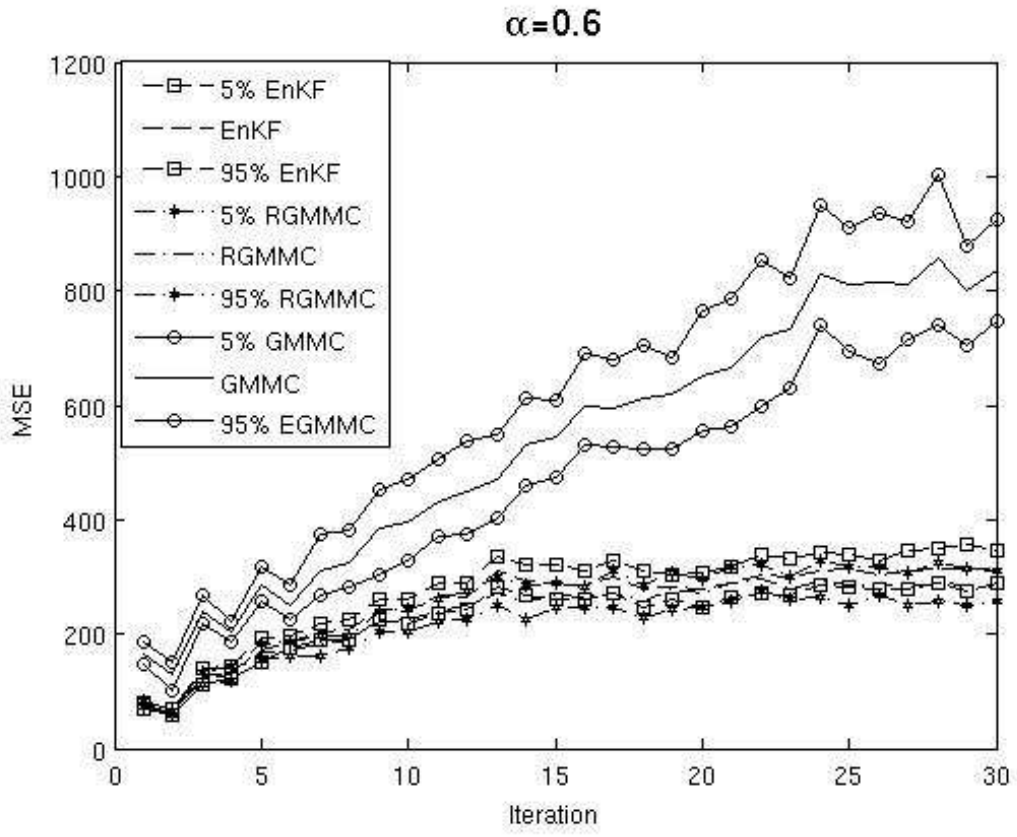

Figure 7: Ten-sensor ten-target, MSE for $\alpha = 0.6$

Table 2: Comparing different filters for ten-sensors ten-targets case

Values	ENKF	GMMC	RGMMC
α			0
CRPS(STD)	1389(6.9)	2311(12.7)	1389(6.6)
MSE(STD)	257(8.1)	581(28.5)	259(7.3)
Coverage(%)	87	25	87
α			0.1
CRPS(STD)	-	-	1390(7.3)
MSE(STD)	-	-	259(8.7)
Coverage(%)	-	-	86
α			0.4
CRPS(STD)	-	-	1389(7.2)
MSE(STD)	-	-	246(6.7)
Coverage(%)	-	-	87
α			0.7
CRPS(STD)	-	-	1513(8.8)
MSE(STD)	-	-	264(13.3)
Coverage(%)	-	-	83
α			1
CRPS(STD)	-	-	2319(12)
MSE(STD)	-	-	594(27.3)
Coverage(%)	-	-	26

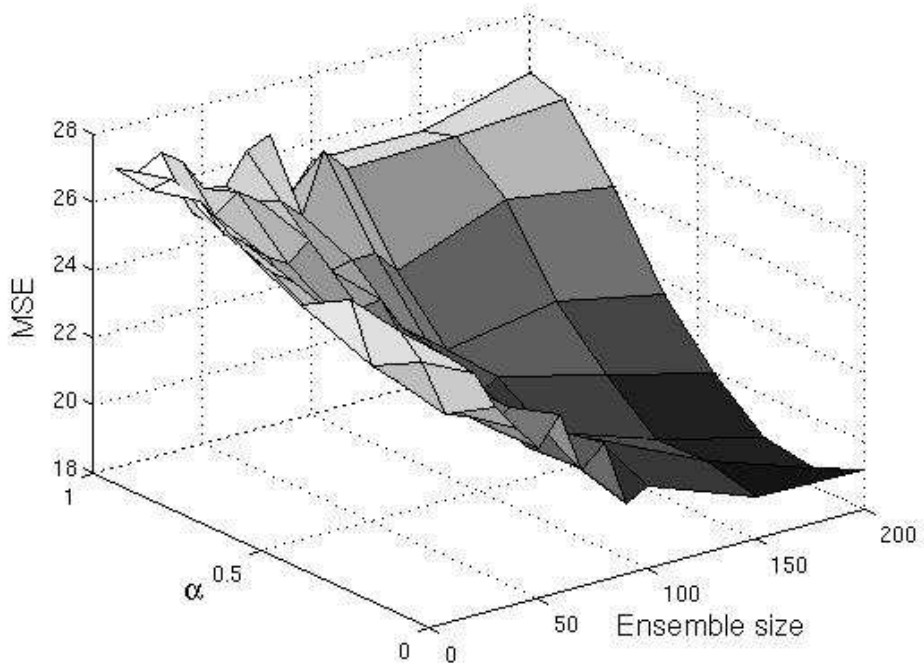

Figure 8: Lorenz 40, the total MSE versus the ensemble size and tuning parameter α, averaged for the interval [900 1000]

to the solution of the above system as $\boldsymbol{x}_t = \boldsymbol{w}_{0.05t}, t = 1, 2, ...$ where $\boldsymbol{w}_t = [w_1(t), w_2(t), ..., w_{40}(t)]'$. The system dynamics are perturbed with a Gaussian noise with mean 0 and standard deviation 0.05. The output of this model is the system state which is cropped by a measurement noise with mean 0 and standard deviation 1, thus the observation $\boldsymbol{y}_t$ is of dimension $m = 40$ and we have:

$$\boldsymbol{y}_t = \boldsymbol{x}_t + N(0, I_{40}). \tag{27}$$

The initial ensemble members are selected randomly from a set of 10000 model states which obtained from one continuous integration at $t = 1000, 1001, ..., 11000$ [14].

In this simulation we evaluate the proposed method on this famous test in data assimilation. We are going to compare filter performance by changing the ensemble size and the tuning parameter α. We know that for $\alpha = 0$, the RGMMC is the ENKF and when $\alpha = 1$ we have the GMMC. Thus, we can compare the filter results with ENKF and GMMC by choosing $0 \leq \alpha \leq 1$ in our simulations. Besides, we compare different cost functions and evaluation criteria such as the total MSE, percentile posterior distribution coverage and variance of the weights.

In Figure 8 we display a three dimensional plot of total MSE versus ensemble size and the tuning parameter α, averaged for the time interval [900 1000] of the Lorenz model. We note that the MSE tend to decrease with larger ensemble size. According to Figure 8, for small sample size, the smallest MSE occurs for the ENKF ($\alpha = 0$). The GMMCs result ($\alpha = 1$) is the worst because the sample size B must be very high for the GMMC to obtain the best result. It is not surprising because we know that GMMC filter diverges when system dimension is proportionally high (40 for this case), since it suffers from sample degeneracy.

In some cases when we increase the number of samples, the number of collinear samples increases too and results in poorer estimation accuracy. For instance, when $\alpha = 0$ (ENKF) by changing the number of ensembles from 90 to 100 the total MSE increases, also for GMMC case (i.e. $\alpha = 1$). Similarly, MSE increases by changing the number of samples from 150 to 200.

According to Figure 8, the best result in total MSE sense is when $\alpha = 0.15$ and the number of samples is 200 and we see that the RGMMC filter result (in total MSE sense) is better than the ENKF and GMMC, also by increasing α from 0 to 0.15 the total MSE decreases but after that it increases.

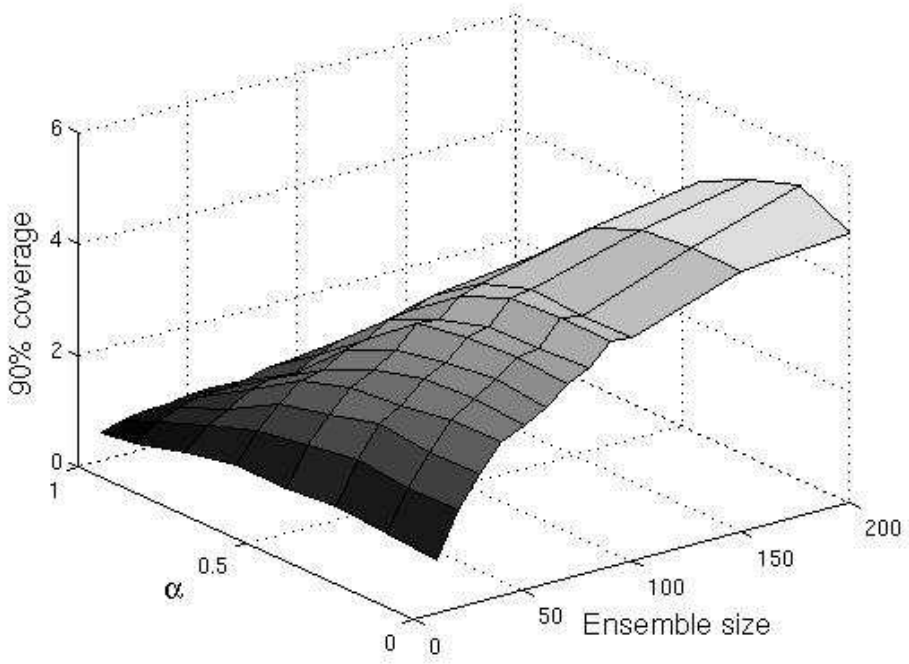

Figure 9: Lorenz 40, the percentile posterior coverage distribution versus the ensemble size and tuning parameter α, averaged for the interval [900 1000]

Finally, choosing the optimal value for the tuning parameter α is highly dependent on sample size and its optimal value depends on both system complexity and number of samples.

Figure 9 shows the filter results in the percentile posterior distribution coverage sense. The distribution coverage are too small but increases as a function of sample size B. Based on this figure we have the best coverage for $0.15 \leq \alpha \leq 0.3$, thus the RGMMC results for this range is the best in comparison to the ENKF and GMMC. For constant sample size, the coverage distribution is like a convex polynomial function of the tuning parameter α (such as second order polynomial function) where its maximum value is around $\alpha = 0.15$. Also, increasing the number of ensembles does not always lead to increase distribution coverage because of sample collinearity. For instance, when $\alpha = 0.45$, and we increase the number of samples from 70 to 80 or when $\alpha = 0$ (ENKF) and the number of samples is changed from 90 to 100 the posterior distribution coverage decreases.

The coverage is so low because of small process noise covariance. We see that it increases rapidly by increasing the process noise covariance, for instance when we increase the process noise standard deviation from 0.05 to 0.1 the percentile coverage distribution increases to 80 percent and more. The parameters used in [14] describe a hard case.

We know that the variance of the weights is 0 for the ENKF because all weights are equal. For high α the variance of the weights is much higher. When the ensemble size is high, the variance of the weights does not change so much for different αs.

4.3 Saturation estimation based on seismic and electromagnetic data

In this example we check the performance of the proposed method for dealing with very high dimensional systems. Besides, for this case we use the proposed adaptive method for choosing α.

The example is a reservoir simulation model, where we consider just one injection well and one production well [11]. The injection well is located at lower left corner of Figure 10 and water is pumped in to this well for replacing oil and moving oil to the production well, which is located at the upper right corner. Based on the fluid dynamics, the flow is faster where the permeability (porosity) is high. Figure 10 shows the saturation profile after 400 days. According to Figure 11 some days after injecting water the production well starts producing oil

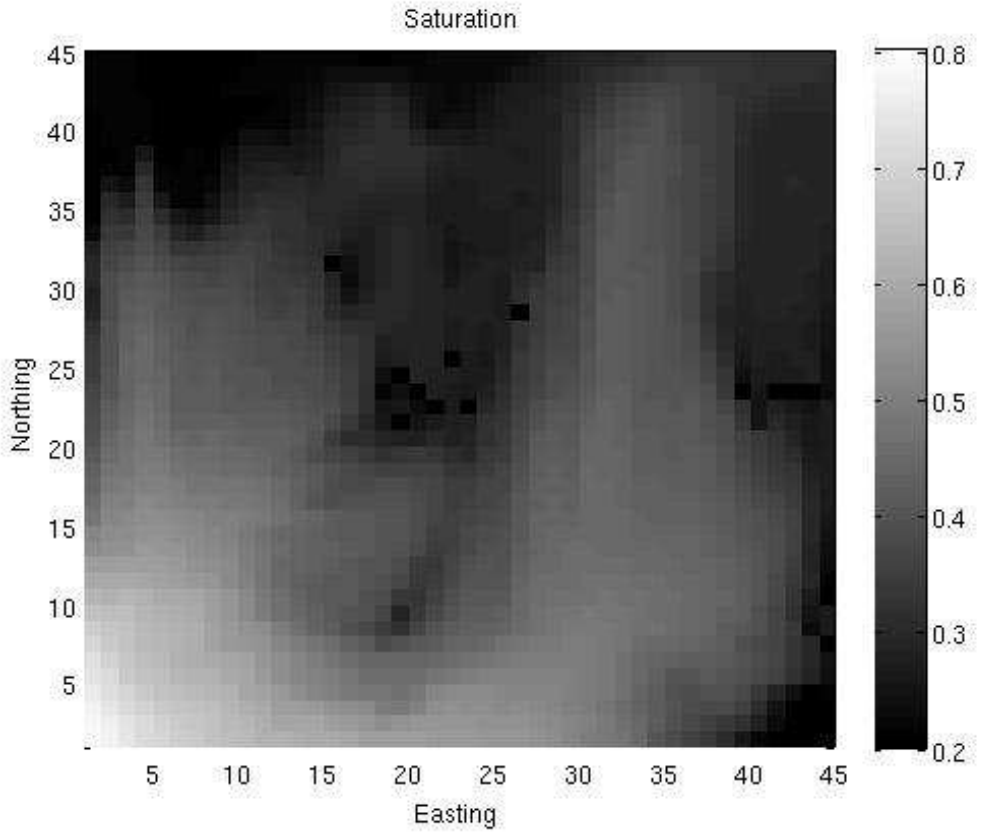

Figure 10: Saturation for one run at fourth time step (400 days)

(the curve at 0 starts to go up).

A fluid flow simulator describes the dynamics of the system. This simulator works on fluid dynamics laws and it contains some parameters such as porosity, permeability and saturation. The values of these parameters depend on the geophysical characteristics of the reservoir. Assuming the whole reservoir as a combination of cells in a lattice, we are interested in estimating porosity, permeability and saturation for each cell. We use a 45×45 lattice in our example. The system state vector consists of porosity, permeability and saturation for each cell. For this example, assume that porosity and permeability are known and constant and we just consider saturation in the state vector. The noise covariance in the dynamic model is $\boldsymbol{P} = \boldsymbol{P}_t = \boldsymbol{P}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})$, where $\boldsymbol{x}_t$ is the state vector which consists of logistic saturation at all $45 \times 45 = 2025$ cells. Logistic saturation transform the variables on the real line. The dynamic model $\boldsymbol{g}_t$ is the numerical solution of the PDE for fluid flow and $t = 1, 2, 3, 4$ where are descretized into 100 day step,i.e. $t = 1$ means 100 days, $t = 2$ means 200 days, etc [11]. You can find more details about the simulator at www.sintef.no/Projectweb/MRST/.

The data consists of seismic and electromagnetic observations, Figure 12, and these are repeated over time. We imagine that at time 0 a baseline survey is performed. Then, monitoring survey are performed at 100, 200, 300 and 400 days (time 1, 2, 3, 4). Two seismic attributes and electromagnetic resistivity are acquired and interpreted along the top reservoir described by a (north, east) cordinate lattice informative of porosity and saturation. Seismic attributes are more informative about reservoir parameters such as porosity and partly saturation, while the electromagnetic data caries coarse scale information about saturation. We assume that we have observations at 4 time steps during 400 days. The likelihood is a function of saturation [12] and the expected response is shown in Figure 12. The observations are made at all lattice cells and its dimension is $m = 3 \times 45^2$. The likelihood is nonlinear, but we linearize it using first order Taylor series expansion.

The simulation is run 50 times in order to reduce the Monte-Carlo error. The number of ensemble member is $B = 100$ which is typically used for ensemble assimilation problems of this size. A comparison between ENKF and RGMMC is done over MSE, CRPS and percentile coverage distribution when we apply these filters in an

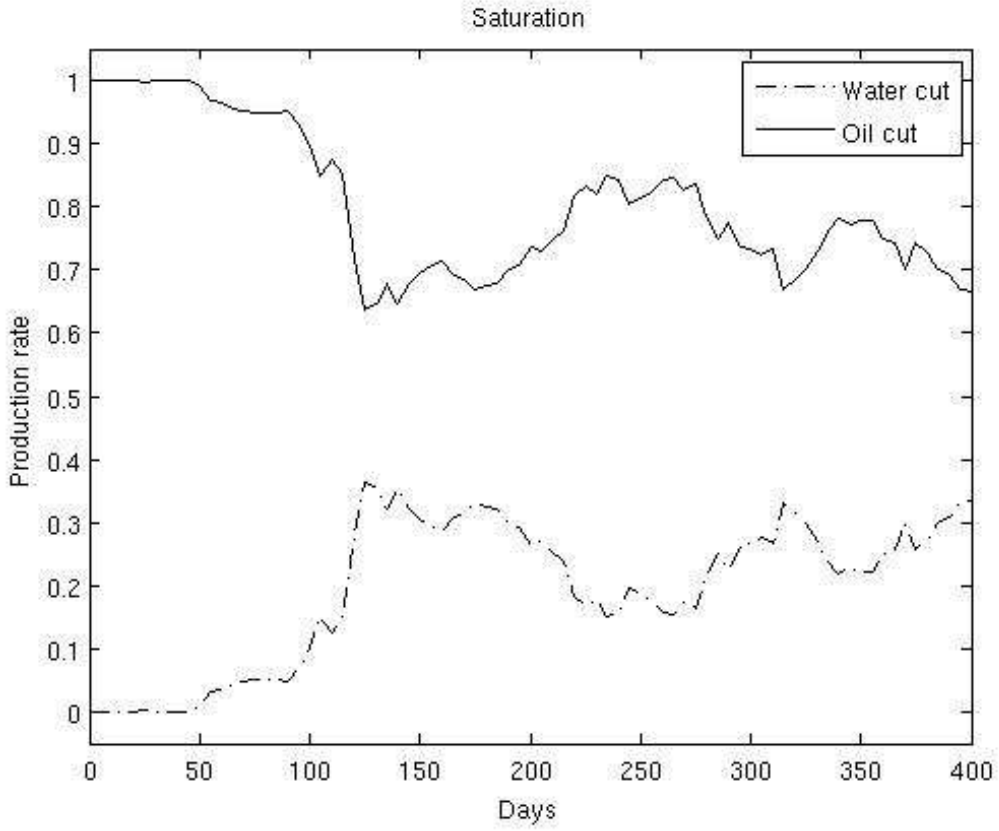

Figure 11: Production rate at well in upper right seismic and electromagnetic data

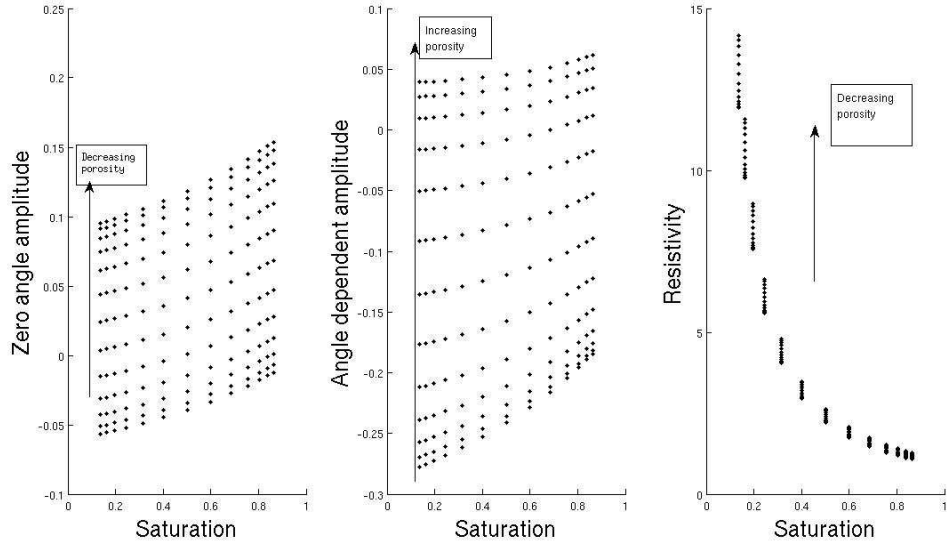

Figure 12: Seismic (first two left plot) and electromagnetic (last right plot) data

Table 3: Filter performance for estimating saturation with adaptive α

Values	ENKF	RGMMC
CRPS(STD)	1.9e5(37)	1.9e5(39)
MSE(STD)	1.3e-3(5e-5)	1.3e-3(5e-5)
Coverage(%)	54.4	54.6

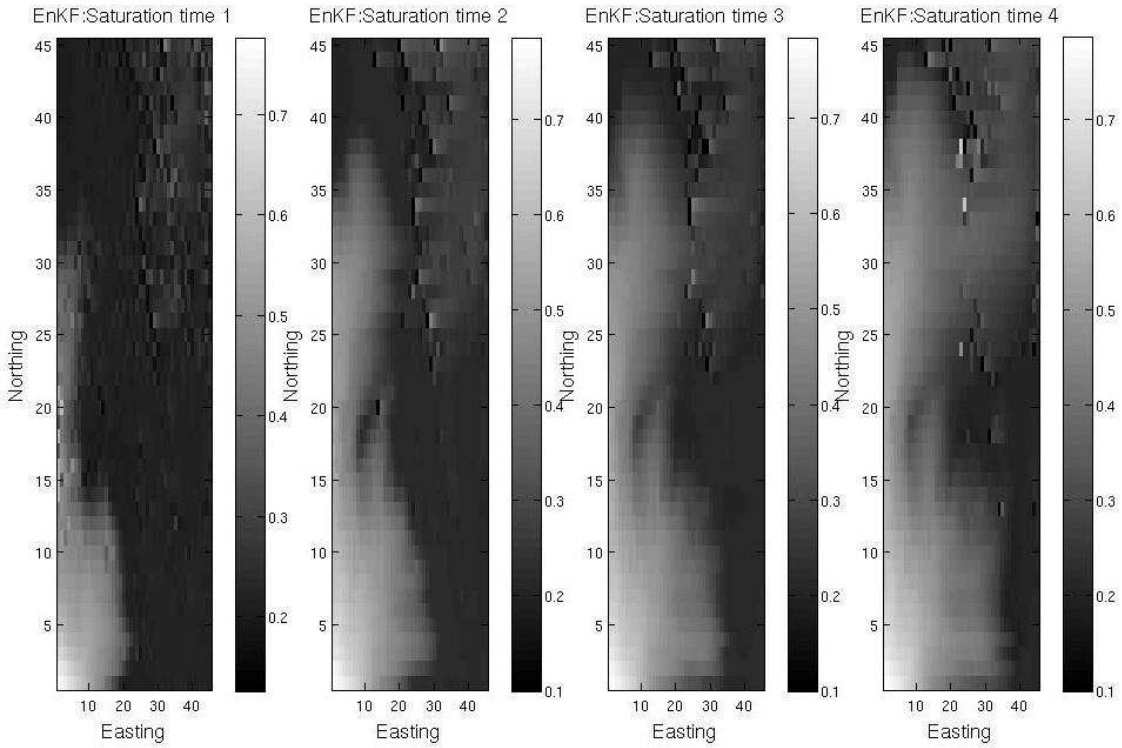

Figure 13: Estimated saturation at different time steps using ENKF

estimation problem.

Table 3 shows the MSE, CRPS and coverage based on proposed adaptive α selection method. We see that RGMMC and ENKF are very similar (the coverage distribution of RGMMC is a little bit better). When we plot the total MSE evaluation sense for these four time steps (400 days), they are very similar. Figure 13 and 14 present similar results. It is very hard to find a difference between the ENKF and RGMMC. These figures shows that RGMMC works as well as ENKF in the high dimensional prediction problems. The value of $\hat{\alpha}_t$ is near 0 for all four time points, and this explains the similar performance. By increasing B, we expect α to grow.

Note the small artifacts caused by the truncation and noise level in the dynamical model. This is a common challange when merging numerical solution of PDEs and statistical Monte-Carlo sampling

5 Closing remarks

We have studied a filtering method going between the ENKF and the particle filter. Our modeling setup allows us to phrase the sample approximation of the particle filter as a Gaussian mixture. Nevertheless, the Gaussian mixture Monte Carlo filtering resulting from this approach will degenerate because of the sampling approximation. The robustified procedure we outline in this paper shrinks particles towards the ensemble mean, while maintaining some flexibility in the distribution.

We tested the robustified filter on a simulation study for target tracking, on the Lorenz 40 model, and on

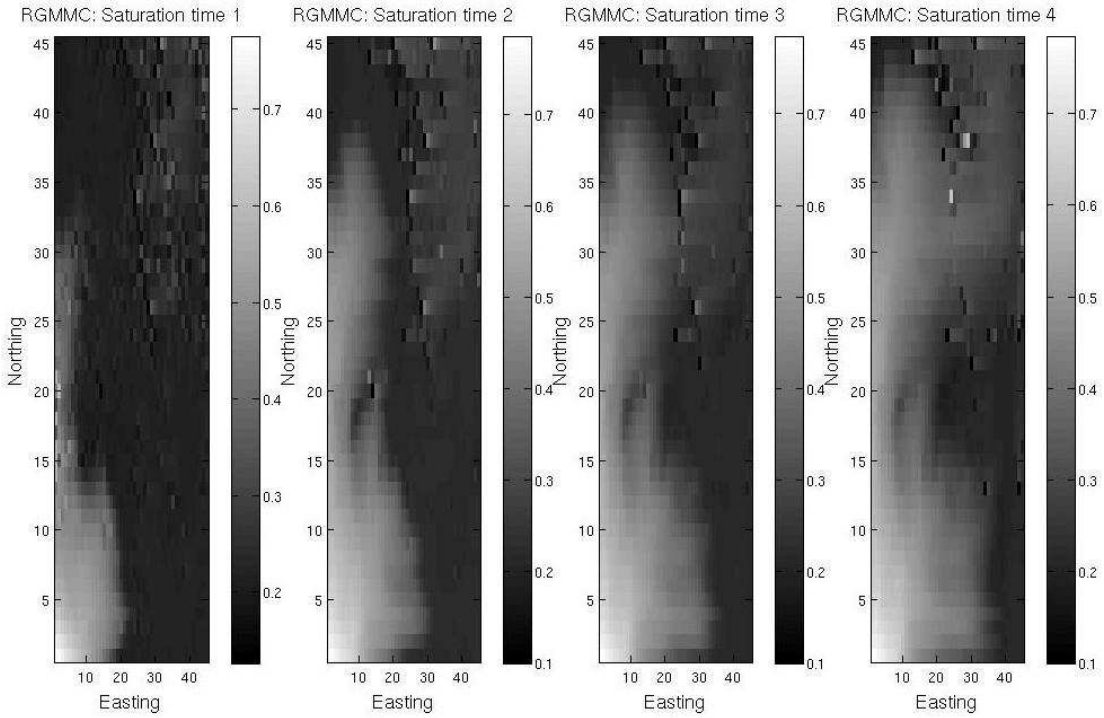

Figure 14: Estimated saturation at different time steps using RGMMC

a reservoir simulation example. Results indicate that the robustified filter works better or as well as Gaussian mixture particle filter and ensemble Kalman filter for systems with different dimensions and complexities.

6 Acknowledgments

We thank the sponsors of the Uncertainty in Reservoir Evaluation (URE) project at NTNU.

We further thank Hans Kunsch for helpful discussions. We thank the Sintef and NERSC because of using the MATLAB reservoir simulation toolbox developed by Sintef and Lorenz40 MATLAB scripts by NERSC.

References

[1] Doucet, A., de Freitas, N., and Gordon, N., Sequential Monte-Carlo Methods in Practice, Springer-Verlag, (2001).

[2] Dovera, L., Della Rossa, E., Moltimodal ensemble Kalman filter using Gaussian mixture models, Journal of Computational Geosciences, Volume: 15 Issue:2, 307-323 (2010).

[3] Evensen, G., The ensemble Kalman filter: Theoretical formulation and practical implementation, Ocean Dynamics, Volume 53 , 343-367 (2003).

[4] Evensen, G., Data assimilation, The Ensemble Kalman Filter, 2nd ed., Springer, (2009)

[5] Fishman, G. S., Monte Carlo: Concepts, Algorithms, and Applications, New York: Springer (1995).

[6] Gneiting, T., Raftery,A. E., Westveld, A. H., Goldman, T., Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation, Journal of Monthly Weather Review, Volume 133, 1098-1118 (2005).

[7] Ito, K., and Xiong, K., Gaussian Filters for Nonlinear Filtering Problems, IEEE Transactions on Automatic Control, Volume: 45 Issue:5, 910-927 (2000).

[8] Jazwinsky, A., Stochastic Processes and Filtering Theory. Academic Press, NewYork, (1970).

[9] Julier, S. J., and Uhlmann, J. K., A New Extension of the Kalman Filter to Nonlinear Systems, In Proc. SPIE - Int. Soc. Opt. Eng. (USA), vol. 3068, 182-193 (1997).

[10] Kalman, R. E., A new approach to linear filtering and prediction problems, ASME Journal of Basic Engineering, 35-45 (1960).

[11] Lie, K.A., Krogstad, S., Ligaarden, I. S., Natvig, J. R., Nilsen, H. M. and Skaflestad, B., Discretisation on complex grids: Open source MATLAB implementation, Proceedings of ECMOR XII, Oxford, UK (2010).

[12] Mavko, G., Mukerji, T., Dvorkin, J., Rock Physics Handbook - Tools for Seismic Analysis in Porous Media, UK, Cambridge University Press (2003).

[13] Norgaard, M., Poulsen, N., and Ravn, O., New Developments in State Estimation for Nonlinear Systems, Automatica Volume: 36 Issue:11, 1627-1638 (2000).

[14] Sakov, P., Oke, P. R., Implications of the form of the ensemble transformation in the ensemble square root filters, Journal of Monthly Weather Review, Volume 136 , 1042-1053 (2008).

[15] Stordal, A. S., Karlsen, H. A., Nævdal, G., Skaug, H. J., Valls, B., Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter, Journal of Computational Geosciences, Volume: 15 Issue:2, 293-305 (2010).

[16] Sætrom, J., Omre, H., Ensemble Kalman filtering with shrinkage regression techniques, Journal of Computational Geosciences, Volume: 15 Issue:2, 271-292 (2010).

[17] Van der Merwe, R. and Wan E., The Square-Root Unscented Kalman Filter for State- and Parameter-Estimation, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (Salt Lake City, UT), Volume 6, 3461-3464 (2001).