Resampling the Ensemble Kalman Filter

I. Myrseth[a,*], J. Sætrom[b], H. Omre[c]

[a]*Norwegian Computing Center, Gaustadalleen 23b, NO-0314 Oslo, Norway*
[b]*Statoil ASA, Vestre Svanholmen 1, NO-4033 Stavanger, Norway*
[c]*Norwegian University of Science and Technology, NO-7491 Trondheim, Norway*

Abstract

Ensemble Kalman filters (EnKF) based on a small ensemble tend to provide collapse of the ensemble over time. It is demonstrated that this collapse is caused by positive coupling of the ensemble members due to use of the estimated Kalman gain for the update of all ensemble members at each time step. This coupling can be avoided by resampling the Kalman gain from its sampling distribution in the conditioning step. In the analytically tractable Gauss-linear model finite sample distributions for all covariance matrix estimates involved in the Kalman gain estimate are known and hence exact Kalman gain resampling can be done. For the general nonlinear case we introduce the resampling ensemble Kalman filter (ResEnKF) algorithm. The resampling strategy in the algorithm is based on bootstrapping of the ensemble and Monte Carlo simulation of the likelihood model. We also define a semi-parametric and parametric version of the resampling ensemble Kalman filter algorithm. An empirical study demonstrates that ResEnKF provides more reliable prediction intervals than traditional EnKF, on the cost of somewhat less accuracy in the point predictions.

[*] Corresponding author. email:myrseth@nr.no, tel:+4722852697

Keywords: Ensemble Kalman filter, Kalman gain sampling, Monte Carlo, Bootstrap

1. Introduction

The Ensemble Kalman Filter (EnKF) introduced by Evensen in the papers Evensen (1994) and Burgers et al. (1998) has found widespread use in evaluation of spatio-temporal phenomena like ocean modeling, weather forecasting and petroleum reservoir evaluation, see Bertino et al. (2002), Houtekamer et al. (2005), Nævdal et al. (2005) and references therein. The filter is popular because of easy implementation and computational efficiency. The filter relies on simulation based inference of hidden Markov models and is closely related to the traditional Kalman filter. The EnKF utilizes a linearization in the data conditioning and relies on empirical probability densities, represented as an ensemble of possible states, which allow general forward functions. These approximations make the ensemble Kalman filter computationally efficient and well suited for high-dimensional hidden Markov models.

The data conditioning is based on the estimated correlation between observations and ensemble members, which is used to update all ensemble members. The estimated regression weights, so called Kalman gain in the case with linear observation relations, is associated with finite sample uncertainty usually resulting in underestimated Kalman gain, see Furrer & Bengtsson (2007). Anderson (2001) deals with the problem by variance inflation in an effort to maintain variability in the ensemble statistics. One of the key assumptions in the data conditioning is that the ensemble members are independent. However, when using the estimated Kalman gain to update every ensemble member, the ensemble members will be coupled over time.

In the hierarchical ensemble Kalman filter (HEnKF) algorithm (Myrseth & Omre, 2010) uncertainty caused by the Kalman gain estimate is accounted for. The HEnKF algorithm relies on an extended model of the prior on the model parameters, however. Houtekamer & Mitchell (1998) use a double ensemble approach where one part of the ensemble is used in the estimation of the Kalman gain used in the update step of the other part. However, this ensures that the ensemble members are uncoupled after the first update step only. In the current paper, we propose to update every ensemble member individually with different estimates of the Kalman gain using a bootstrapping technique (Efron, 1979). The estimation uncertainty associated with the Kalman gain will then be reflected in the ensemble uncertainty. We also introduce formalism that handles non-linear relations between state and observation.

2. Model Assumptions

Consider an unknown, multivariate time series $[\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_T, \mathbf{x}_{T+1}]$ with $\mathbf{x}_t \in \mathbb{R}^{p_{\mathbf{x}}}$; $t = 0, ..., T+1$ containing the primary variable of interest and $\mathbf{x}_T$ being the current state. Assume that an associated time series of observations $[\mathbf{d}_0, ..., \mathbf{d}_T]$ with $\mathbf{d}_t \in \mathbb{R}^{p_{\mathbf{d}}}$; $t = 0, ..., T$, is available.

Define a prior stochastic model for $[\mathbf{x}_0, ..., \mathbf{x}_{T+1}]$ by assuming Markov properties:

$$\begin{aligned}[\mathbf{x}_0, ..., \mathbf{x}_{T+1}] &\sim f(\mathbf{x}_0, ..., \mathbf{x}_{T+1}) \\ &= f(\mathbf{x}_0) \prod_{t=0}^{T} f(\mathbf{x}_{t+1}|\mathbf{x}_0, ..., \mathbf{x}_t) \\ &= f(\mathbf{x}_0) \prod_{t=0}^{T} f(\mathbf{x}_{t+1}|\mathbf{x}_t). \end{aligned} \tag{1}$$

Let $f(\mathbf{x}_0)$ be a known pdf for the initial state, and $f(\mathbf{x}_{t+1}|\mathbf{x}_t)$ for $t = 0, ..., T$ be known forward pdfs. Hence the prior model for the time series of interest is Markovian with each state given the past, dependent on the previous state only.

Define the likelihood model for $[\mathbf{d}_0, ..., \mathbf{d}_T]$ given $[\mathbf{x}_0, ..., \mathbf{x}_{T+1}]$ by assuming conditional independence and single state dependence:

$$\begin{aligned}[\mathbf{d}_0, ..., \mathbf{d}_T|\mathbf{x}_0, ..., \mathbf{x}_{T+1}] &\sim f(\mathbf{d}_0, ..., \mathbf{d}_T|\mathbf{x}_0, ..., \mathbf{x}_{T+1}) \\ &= \prod_{t=0}^{T} f(\mathbf{d}_t|\mathbf{x}_0, ..., \mathbf{x}_{t+1}) \\ &= \prod_{t=0}^{T} f(\mathbf{d}_t|\mathbf{x}_t) \end{aligned} \tag{2}$$

where $f(\mathbf{d}_t|\mathbf{x}_t)$ for $t = 0, ..., T$ are known likelihood functions. Hence, the likelihood model entails that the observation at time t is a function of state $\mathbf{x}_t$ only and is independent of the other observations when $\mathbf{x}_t$ is given.

These prior and likelihood assumptions define a hidden Markov process as depicted by the graph in Figure 1. The resulting posterior stochastic model is defined by Bayesian inversion:

$$\begin{aligned}[\mathbf{x}_0, ..., \mathbf{x}_{T+1}|\mathbf{d}_0, ..., \mathbf{d}_T] &\sim f(\mathbf{x}_0, ..., \mathbf{x}_{T+1}|\mathbf{d}_0, ..., \mathbf{d}_T) \\ &= \text{const} \times f(\mathbf{d}_0, ..., \mathbf{d}_T|\mathbf{x}_0, ..., \mathbf{x}_{T+1}) \\ &\qquad \times f(\mathbf{x}_0, ..., \mathbf{x}_{T+1}) \\ &= \text{const} \times f(\mathbf{x}_0) f(\mathbf{d}_0|\mathbf{x}_0) \\ &\qquad \times \left[\prod_{t=0}^{T-1} f(\mathbf{d}_{t+1}|\mathbf{x}_{t+1}) f(\mathbf{x}_{t+1}|\mathbf{x}_t)\right] \\ &\qquad \times f(\mathbf{x}_{T+1}|\mathbf{x}_T), \end{aligned} \tag{3}$$

with ‘const’ being a normalizing constant that is usually hard to assess. Hence the full posterior model is not easily available.

For the hidden Markov model described, the forecast is of interest. The forecasting pdf is available as:

$$\begin{aligned}[\mathbf{x}_{T+1}|\mathbf{d}_0, ..., \mathbf{d}_T] &\sim f(\mathbf{x}_{T+1}|\mathbf{d}_0, ..., \mathbf{d}_T) \\ &= \int ... \int f(\mathbf{x}_0, ..., \mathbf{x}_{T+1}|\mathbf{d}_0, ..., \mathbf{d}_T) \mathrm{d}\mathbf{x}_0 ... \mathrm{d}\mathbf{x}_T. \end{aligned} \tag{4}$$

This forecasting pdf is computable by a recursive algorithm which alternates a forward-in-time step and a condition-on-data step. This recursive algorithm makes sequential conditioning on future observations possible.

The model described above can be summarized by the following general state space equations:

$$\begin{aligned} \mathbf{x}_0 &\sim f(\mathbf{x}_0) \\ \mathbf{x}_{t+1}|\mathbf{x}_t = \omega_t(\mathbf{x}_t, \boldsymbol{\epsilon}_t^{\mathbf{x}}) &\sim f(\mathbf{x}_{t+1}|\mathbf{x}_t) \\ \mathbf{d}_t|\mathbf{x}_t = \nu_t(\mathbf{x}_t, \boldsymbol{\epsilon}_t^{\mathbf{d}}) &\sim f(\mathbf{d}_t|\mathbf{x}_t), \end{aligned} \tag{5}$$

where $\omega_t(.,.)$ is a known function $\mathbb{R}^{2p_{\mathbf{x}}} \to \mathbb{R}^{p_{\mathbf{x}}}$ and $\boldsymbol{\epsilon}_t^{\mathbf{x}}$ is a random variable from the normalized $p_{\mathbf{x}}$-dimensional multivariate Gaussian distribution $N_{p_{\mathbf{x}}}(\mathbf{0}, \mathbf{I}_{p_{\mathbf{x}}})$ where $\mathbf{I}_{p_{\mathbf{x}}}$ is a unit diagonal covariance matrix, $\nu_t(.,.)$ is a known function $\mathbb{R}^{p_{\mathbf{x}}+p_{\mathbf{d}}} \to \mathbb{R}^{p_{\mathbf{d}}}$ and $\boldsymbol{\epsilon}_t^{\mathbf{d}}$ is a normalized $p_{\mathbf{d}}$-dimensional Gaussian random variable from $N_{p_{\mathbf{d}}}(\mathbf{0}, \mathbf{I}_{p_{\mathbf{d}}})$. This construction can generate a realization from an arbitrary forward, $f(\mathbf{x}_{t+1}|\mathbf{x}_t)$, and likelihood, $f(\mathbf{d}_t|\mathbf{x}_t)$, model.

3. The Ensemble Kalman Filter

The EnKF is an algorithm that can be used to assess the forecasting pdf. The basic idea of the EnKF to represent an empirical distribution approximating the true prior by a set of realizations, so called ensemble. These realizations are adjusted according to the likelihood model when an observation occurs and the adjusted realizations are then taken through the forward model to the next observation time. At time $t = T + 1$ a set of approximately independent realizations are available for empirical assessment of $f(\mathbf{x}_{T+1}|\mathbf{d}_0, ..., \mathbf{d}_T)$. Hence, characteristics beyond the two first moments can be captured. Basic references for EnKF are Evensen (1994), Burgers et al. (1998), Evensen (2007) and references therein.

Introduce the following notation, with

$$\begin{aligned} \mathbf{x}_t^u &= [\mathbf{x}_t|\mathbf{d}_0, ..., \mathbf{d}_{t-1}] \\ \mathbf{x}_t^c &= [\mathbf{x}_t|\mathbf{d}_0, ..., \mathbf{d}_t], \end{aligned} \tag{6}$$

where indices u and c indicate unconditioned and conditioned on the observation at the current time, respectively. Define a time series of ensembles:

$$\mathbf{e}_t \,:\, \{(\mathbf{x}_t^u, \mathbf{d}_t)^{(i)}; i = 1, ..., n_{\mathbf{e}}\}\,;\, t = 0, ..., T + 1, \tag{7}$$

where $\mathbf{x}_t^{u(i)} = [\mathbf{x}_t|\mathbf{d}_0, ..., \mathbf{d}_{t-1}]^{(i)}$ are approximate realizations from $f(\mathbf{x}_t|\mathbf{d}_0, ..., \mathbf{d}_{t-1})$ and $[\mathbf{d}_t^{(i)}|\mathbf{x}_t^{u(i)}] = \nu(\mathbf{x}_t^{u(i)}, \boldsymbol{\epsilon}_t^{\mathbf{d}(i)})$ are associated realizations of the observation available at time t. Note that at any step t - with t omitted in the notation - one has the expectation vector and covariance matrix:

$$\boldsymbol{\mu}_{\mathbf{xd}} = \begin{bmatrix} \mathrm{E}\{\mathbf{x}^u\} \\ \mathrm{E}\{\mathbf{d}\} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{x}} \\ \boldsymbol{\mu}_{\mathbf{d}} \end{bmatrix} \tag{8}$$

and

$$\Sigma_{\mathbf{xd}} = \begin{bmatrix} \mathrm{Cov}\{\mathbf{x}^u\} & \mathrm{Cov}\{\mathbf{x}^u, \mathbf{d}\} \\ \mathrm{Cov}\{\mathbf{d}, \mathbf{x}^u\} & \mathrm{Cov}\{\mathbf{d}\} \end{bmatrix} = \begin{bmatrix} \Sigma_{\mathbf{x}} & \Gamma_{\mathbf{x},\mathbf{d}} \\ \Gamma_{\mathbf{d},\mathbf{x}} & \Sigma_{\mathbf{d}} \end{bmatrix}. \tag{9}$$

The traditional EnKF, see Evensen (2007), is defined with a Gauss-linear likelihood model

$$\mathbf{d}_t|\mathbf{x}_t = H_t\mathbf{x}_t + \boldsymbol{\epsilon}_t, \tag{10}$$

with $\boldsymbol{\epsilon}_t$ being $N_{p_{\mathbf{d}}}(\mathbf{0}, \Sigma_{\mathbf{d}|\mathbf{x}})$, or a linearization of a nonlinear likelihood model. The associated traditional EnKF algorithm is presented in Algorithm 1.

The EnKF algorithm is recursive and each recursion consists of a conditioning operation and a forwarding operation. The conditioning expression is linear with weights estimated from the ensemble. The forwarding operation is defined by the forward pdf. This entails two implicit approximations in the EnKF:

The sample space of $\mathbf{x}_t$ is discretized and represented by a finite number of realizations. Initially an ensemble of iid realizations is assumed to represent $f(\mathbf{x}_0)$. For high-dimensional problems a large number of ensemble members may be required to do so reliably.

The data conditioning expression is linearized. Moreover, the weights

Algorithm 1: Traditional Ensemble Kalman filter

```
Initiate:
    • $n_{\mathbf{e}} =$ no. of ensemble members
    • $\mathbf{x}_0^{u(i)}$ ; $i = 1, ..., n_{\mathbf{e}}$ iid $f(\mathbf{x}_0)$
    • $\boldsymbol{\epsilon}_0^{\mathbf{d}(i)} \sim N_{p_{\mathbf{d}}}(\mathbf{0}, \Sigma_{\mathbf{d}|\mathbf{x}})$ ; $i = 1, ..., n_{\mathbf{e}}$
    • $\mathbf{d}_0^{(i)} = H_0 \mathbf{x}_0^{u(i)} + \boldsymbol{\epsilon}_0^{\mathbf{d}(i)}$ ; $i = 1, ..., n_{\mathbf{e}}$
    • $\mathbf{e}_0 \ : \ \{(\mathbf{x}_0^u, \mathbf{d}_0)^{(i)} \ ; \ i = 1, ..., n_{\mathbf{e}}\}$
for $t = 0$ to $T$ do
    Conditioning:
        • Estimate $\widehat{\Sigma}_{\mathbf{x}}$ from $\mathbf{e}_t$
        • $\widehat{\Gamma}_{\mathbf{x},\mathbf{d}} = \widehat{\Sigma}_{\mathbf{x}} H_t'$
        • $\widehat{\Sigma}_{\mathbf{d}} = H_t \widehat{\Sigma}_{\mathbf{x}} H_t' + \Sigma_{\mathbf{d}|\mathbf{x}}$
        • $\mathbf{x}_t^{c(i)} = \mathbf{x}_t^{u(i)} + \widehat{\Gamma}_{\mathbf{x},\mathbf{d}} \widehat{\Sigma}_{\mathbf{d}}^{-1} (\mathbf{d}_t - \mathbf{d}_t^{(i)})$ ; $i = 1, ..., n_{\mathbf{e}}$
    Forwarding:
        • $\boldsymbol{\epsilon}_t^{\mathbf{x}(i)} \sim N_{p_{\mathbf{x}}}(\mathbf{0}, \mathbf{I}_{p_{\mathbf{x}}})$ ; $i = 1, ..., n_{\mathbf{e}}$
        • $\mathbf{x}_{t+1}^{u(i)} = \omega_t(\mathbf{x}_t^{c(i)}, \boldsymbol{\epsilon}_t^{\mathbf{x}(i)})$ ; $i = 1, ..., n_{\mathbf{e}}$
        • $\boldsymbol{\epsilon}_{t+1}^{\mathbf{d}(i)} \sim N_{p_{\mathbf{d}}}(\mathbf{0}, \Sigma_{\mathbf{d}|\mathbf{x}})$ ; $i = 1, ..., n_{\mathbf{e}}$
        • $\mathbf{d}_{t+1}^{(i)} = H_{t+1} \mathbf{x}_{t+1}^{u(i)} + \boldsymbol{\epsilon}_{t+1}^{\mathbf{d}(i)}$ ; $i = 1, ..., n_{\mathbf{e}}$
        • $\mathbf{e}_{t+1} \ : \ \{(\mathbf{x}_{t+1}^u, \mathbf{d}_{t+1})^{(i)} \ ; \ i = 1, ..., n_{\mathbf{e}}\}$
Assess
    • $f(\mathbf{x}_{T+1}|\mathbf{d}_0, ..., \mathbf{d}_T)$ from $\mathbf{e}_{T+1}$
```

in the linearization are estimated from the ensemble. Note, however, that each ensemble member is conditioned individually and hence the linearization only applies to the conditioning not to the forward model. For highly non-Gaussian prior models and/or strongly nonlinear likelihood models this approximation may provide unreliable results.

Under these approximations, however, all types of models for the hidden Markov process can be evaluated. Other problems arise in the EnKF which are caused by the use of an estimate of the Kalman gain based on $\mathbf{e}_t$ instead of the true weights. These problems include rank deficiency and estimation uncertainty due to the limited size of the ensemble, i.e., small values of $n_{\mathbf{e}}$. A discussion of the implications of data conditioning based on finite sample ensemble statistics follows.

3.1. The conditioning step

The conditioning step in the EnKF contains the linear approximation that appears crucial for the success of the filter. The conditioning expression relies on the Kalman gain $K = \Gamma_{\mathbf{x},\mathbf{d}}\Sigma_{\mathbf{d}}^{-1}$, which must be estimated at each state from the $n_{\mathbf{e}}$ members of the ensemble $\mathbf{e}_t$. In the general case with nonlinear likelihood model, the classical covariance estimators are applied:

$$\widehat{\Gamma}_{\mathbf{x},\mathbf{d}} = \frac{1}{n_{\mathbf{e}}-1}\sum_{i=1}^{n_{\mathbf{e}}}(\mathbf{x}_t^{u(i)} - \widehat{\boldsymbol{\mu}}_{\mathbf{x}})(\mathbf{d}_t^{(i)} - \widehat{\boldsymbol{\mu}}_{\mathbf{d}})' \tag{11}$$

$$\widehat{\Sigma}_{\mathbf{d}} = \frac{1}{n_{\mathbf{e}}-1}\sum_{i=1}^{n_{\mathbf{e}}}(\mathbf{d}_t^{(i)} - \widehat{\boldsymbol{\mu}}_{\mathbf{d}})(\mathbf{d}_t^{(i)} - \widehat{\boldsymbol{\mu}}_{\mathbf{d}})', \tag{12}$$

with

$$\widehat{\boldsymbol{\mu}}_{\mathbf{x}} = \frac{1}{n_{\mathbf{e}}}\sum_{i=1}^{n_{\mathbf{e}}}\mathbf{x}_t^{u(i)} \tag{13}$$

and

$$\widehat{\boldsymbol{\mu}}_{\mathbf{d}} = \frac{1}{n_{\mathbf{e}}} \sum_{i=1}^{n_{\mathbf{e}}} \mathbf{d}_t^{(i)}. \tag{14}$$

If $\mathbf{e}_t$ contains independent members these estimators are unbiased and consistent. The latter entails $\widehat{\Gamma}_{\mathbf{x},\mathbf{d}} \to \Gamma_{\mathbf{x},\mathbf{d}}$ and $\widehat{\Sigma}_{\mathbf{d}} \to \Sigma_{\mathbf{d}}$ as $n_{\mathbf{e}} \to \infty$, for all distributional models. Moreover, $\widehat{K} = \widehat{\Gamma}_{\mathbf{x},\mathbf{d}} \widehat{\Sigma}_{\mathbf{d}}^{-1} \to K$ as $n_{\mathbf{e}} \to \infty$. Note that in the traditional EnKF scheme defined in Algorithm 1, $\hat{K}$ will be biased estimate of K. The consequences of this bias are thoroughly discussed in Furrer & Bengtsson (2007), where conditions on the size of $n_{\mathbf{e}}$ for obtaining bounded error growth is developed. It was recommended to use a boosting or inflation factor to correct the variability for this bias. For finite $n_{\mathbf{e}}$, it is known (Huber, 1981) that the classical estimators for covariance matrices are notoriously unreliable due to extreme dependence on the tail behavior of the underlying pdf. This sensitivity is caused by the second order terms of the estimators. The lack of precision in $\widehat{K}$ may cause spurious values to appear in the conditioned $\mathbf{x}_t^c$, which impact may be accelerated by non-linear forward models.

The motivation for this study follows from a closer evaluation of the conditioning relation

$$\mathbf{x}^{c(i)} = \mathbf{x}^{u(i)} + \hat{K}(\mathbf{d} - \mathbf{d}^{(i)})\,;\, i = 1, ..., n_{\mathbf{e}}, \tag{15}$$

where the time index is omitted for simplicity. Let the prior model for $\mathbf{x}^u$ be a Gaussian and the likelihood be Gauss-linear with known model parameters. Then the Kalman gain

$$K = \Sigma_{\mathbf{x}} H' (H \Sigma_{\mathbf{x}} H' + \Sigma_{\mathbf{d}|\mathbf{x}})^{-1} \tag{16}$$

is known. For this case $\mathbf{x}^c$ is Gaussian with $\mathrm{E}\{\mathbf{x}^c\} = \mathrm{E}\{\mathbf{x}^u|\mathbf{d}\} = \mu_{\mathbf{x}|\mathbf{d}}$ and $\mathrm{Cov}\{\mathbf{x}^c\} = \mathrm{Cov}\{\mathbf{x}^u|\mathbf{d}\} = \Sigma_{\mathbf{x}|\mathbf{d}}$ which constitutes the correct solution when all model parameters are known. Moreover, if the ensemble members $\mathbf{x}_t^{u(i)}$; $i = 1, ..., n_\mathbf{e}$ are independent, the resulting $\mathbf{x}_t^{c(i)}$; $i = 1, ..., n_\mathbf{e}$ will also be independent.

In practice, however, the model parameters are not known and the Kalman gain K must be estimated from the $n_\mathbf{e}$ members of the ensemble $\mathbf{e}$. Since the conditioning will be based on an estimate of K, the resulting uncertainty in $\mathbf{x}^c$ will be larger than if K was known. Assume that the elements in $\mathbf{e} : \{(\mathbf{x}^u, \mathbf{d})^{(i)}; i = 1, ..., n_\mathbf{e}\}$ are independent and Gaussian and that $n_\mathbf{e} > p_\mathbf{x}$. Then the standard estimator of $\Sigma_\mathbf{x}$ will be Wishart distributed, and estimates of $\Sigma_\mathbf{x}$ can be simulated and by applying Expression (16) realizations of K can be generated. Hence K can be seen as a random variable $K_\mathbf{e} \sim f(K_\mathbf{e})$. In order to evaluate the characteristics of $\mathbf{x}^c$ when $K_\mathbf{e}$ is random, consider the Taylor expansion of Expression (15) around $\mathrm{E}\{K_\mathbf{e}\} = \mu_{K_\mathbf{e}}$ and $\mathrm{E}\{\mathbf{d}^{(i)}\} = \mu_\mathbf{d}$:

$$\begin{aligned}\mathbf{x}^{c(i)} \approx \mathbf{x}^{u(i)} &+ \mu_{K_\mathbf{e}}(\mathbf{d} - \mu_\mathbf{d}) - \mu_{K_\mathbf{e}}(\mathbf{d}^{(i)} - \mu_\mathbf{d}) \\ &+(K_\mathbf{e} - \mu_{K_\mathbf{e}})(\mathbf{d} - \mu_\mathbf{d}) - (K_\mathbf{e} - \mu_{K_\mathbf{e}})(\mathbf{d}^{(i)} - \mu_\mathbf{d}) \\ &; i = 1, ..., n_\mathbf{e} \qquad (17)\end{aligned}$$

$K_\mathbf{e}$ is inferred from the complete set of ensemble members, hence the dependence between $K_\mathbf{e}$ and each $(\mathbf{x}^u, \mathbf{d})^{(i)}$ is relatively weak. In the expressions calculated below the two variables are assumed to be independent.

The following expressions can be developed from Expression (17)

$$\begin{aligned}
\mathrm{E}\{\mathbf{x}^{c(i)}\} &= \mu_{\mathbf{x}} + (\mu_{K_{\mathbf{e}}} - K)(\mathbf{d} - \mu_{\mathbf{d}}) \\
\mathrm{Cov}\{\mathbf{x}^{c(i)}\} &= \Sigma_{\mathbf{x}} + (\mu_{K_{\mathbf{e}}} - K)\Sigma_{\mathbf{d}}(\mu_{K_{\mathbf{e}}} - K)' \\
&\quad + \mathrm{E}\{(K_{\mathbf{e}} - \mu_{K_{\mathbf{e}}})(\Sigma_{\mathbf{d}} + \triangle_{\mathbf{d}})(K_{\mathbf{e}} - \mu_{K_{\mathbf{e}}})'\} \\
\mathrm{Cov}\{\mathbf{x}^{c(i)}, \mathbf{x}^{c(j)}\} &= \mathrm{E}\{(K_{\mathbf{e}} - \mu_{K_{\mathbf{e}}})\triangle_{\mathbf{d}}(K_{\mathbf{e}} - \mu_{K_{\mathbf{e}}})'\} \qquad (18)
\end{aligned}$$

for $i, j = 1, ..., n_e; i \neq j$, where K is the true, but unknown Kalman weight and $\triangle_{\mathbf{d}} = (\mathbf{d} - \mu_{\mathbf{d}})(\mathbf{d} - \mu_{\mathbf{d}})'$. Hence, if $K_{\mathbf{e}}$ is an unbiased estimator for K, the conditioned ensemble members are correctly centered at $\mu_{\mathbf{x}|\mathbf{d}}$, while there is an additional $K_{\mathbf{e}}$ estimation term in the variance of the members. Note, that the cross ensemble covariance will be positive whenever there is estimation uncertainty in $K_{\mathbf{e}}$. Hence the members of the conditioned ensemble will be positively coupled and the empirical covariance matrix based on the ensemble will underestimate the covariance. This is alarming since the EnKF is based on a sequential conditioning through time meaning that the coupling will grow increasingly stronger. Lastly, note that as $n_{\mathbf{e}} \to \infty$ the uncertainty in $K_{\mathbf{e}}$ decreases and all problems disappear.

One possible solution to avoid this coupling problem is to perform Kalman gain resampling:

$$\begin{aligned}
&K_{\mathbf{e}}^{(1)}, ..., K_{\mathbf{e}}^{(n_{\mathbf{e}})} \text{ iid } f(K_{\mathbf{e}}) \\
&\mathbf{x}^{c(i)} = \mathbf{x}^{u(i)} + K_{\mathbf{e}}^{(i)}(\mathbf{d} - \mathbf{d}^{(i)})\,;\ i = 1, ..., n_{\mathbf{e}} \qquad (19)
\end{aligned}$$

Then $\mathrm{E}\{\mathbf{x}^c\}$ and $\mathrm{Cov}\{\mathbf{x}^c\}$ will remain the same as for Expression (15), while $\mathrm{Cov}\{\mathbf{x}^{c(i)}, \mathbf{x}^{c(j)}\} = 0$, and hence the ensemble coupling disappears. The assessment of $f(K_{\mathbf{e}})$ remains a challenge of course.

For the complete Gauss-linear model, all finite sample distributions are known. In particular it is known that $\mathbf{x}^{(i)} \sim N_{p_{\mathbf{x}}}(\boldsymbol{\mu}_{\mathbf{x}}, \Sigma_{\mathbf{x}})$ at each conditioning step. The following resampling is reasonable:

```
do $i = 1, ..., n_{\mathbf{e}}$
    $\mathbf{x}_{\cdot}^{(1)}, ..., \mathbf{x}_{\cdot}^{(n_{\mathbf{e}})}$ iid $N_{p_{\mathbf{x}}}(\boldsymbol{\mu}_{\mathbf{x}}, \Sigma_{\mathbf{x}})$
    $\Sigma_{\mathbf{x}}^{(i)} = \frac{1}{n_{\mathbf{e}} - 1} \sum_{j=1}^{n_{\mathbf{e}}} (\mathbf{x}_{\cdot}^{(j)} - \bar{\mathbf{x}}_{\cdot})$        (20)
    $K_{\mathbf{e}}^{(i)} = \Sigma_{\mathbf{x}}^{(i)} H' (H \Sigma_{\mathbf{x}}^{(i)} H' + \Sigma_{\mathbf{d}|\mathbf{x}})^{-1}$
    $\mathbf{x}^{c(i)} = \mathbf{x}^{u(i)} + K_{\mathbf{e}}^{(i)} (\mathbf{d} - \mathbf{d}^{(i)})$
end do
```

The resampling EnKF approach specified in Expression (20) is primarily aimed at restoring the independence between the conditioned ensemble members, thus improving the reliability of the prediction intervals. The variability in the estimated Kalman gains $K_{\mathbf{e}}$ will remain and spurious values in $\mathbf{x}^c$ will still occur. The Hierarchical EnKF (HEnKF) approach, presented in Myrseth & Omre (2010) aims at combining a shrinkage estimator for $K_{\mathbf{e}}$ which reduce spurious values and the resampling approach defined above. The empirical study in Myrseth & Omre (2010) provides very encouraging results, but HEnKF requires additional modeling which can be difficult in large problems. In the current paper we present an empirical resampling approach which requires no additional modeling assumptions.

4. Resampling

Resampling or simulating from a known pdf is called Monte Carlo simulation, see Hammersley & Handscomb (1964). This is sometimes the most efficient way to determine the pdf of random variables that are nonlinear functions of random variables with known pdfs. Actually, this is exactly the goal of the EnKF. The bootstrap, formally introduced in Efron (1979), is a statistical method to assess parameter uncertainty. Here we will only give a short introduction to the bootstrap and Monte Carlo techniques, and refer the interested reader to Efron & Tibshirani (1993).

Consider a random variable with an associated cdf, $\mathbf{x} \sim F(\mathbf{x})$, and some interesting characteristic of the cdf, $\xi = h(F(\mathbf{x}))$. Examples of this characteristic are the expectation $\mathrm{E}\{\mathbf{x}\}$, covariance $\mathrm{Cov}\{\mathbf{x}\}$, quantiles $\mathrm{Prob}\{\mathbf{x} \leq \mathrm{c}\}$ for some arbitrary c etc. Assume that a set of realizations $\mathbf{x}_1, ..., \mathbf{x}_n$ iid $F(\mathbf{x})$ are available and define a finite sample estimator of ξ, $\hat{\xi}_n = h_n(\mathbf{x}_1, ..., \mathbf{x}_n)$, such that $h_n(\mathbf{x}_1, ..., \mathbf{x}_n) \overset{n\to\infty}{\longrightarrow} h(F(\mathbf{x}))$. The objective is to obtain the cdf of the finite sample estimator $\hat{\xi}_n$, $F_n(\xi)$. If $F(\mathbf{x})$ is fully known, then assessment of $F_n(\xi)$ can be done by Monte Carlo simulation as described in Algorithm 2.

Algorithm 2: Monte Carlo simulation

```
Initiate:
  • $m =$ no. of Monte Carlo replicates
for $i = 1$ to $m$ do
    Generate: $\mathbf{x}_1^*, ..., \mathbf{x}_n^*$ iid $F(\mathbf{x})$
    $\hat{\xi}_n^{(i)} = h_n(\mathbf{x}_1^*, ..., \mathbf{x}_n^*)$
Estimate: $F_n(\xi)$ from $\hat{\xi}_n^{(1)}, ..., \hat{\xi}_n^{(m)} \rightarrow \hat{F}_n(\xi)$
```

```
Algorithm 3: Bootstrap
  Initiate:
    • $b =$ no. of bootstrap replicates
    • $\hat{F}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} I(\mathbf{x}_i < \mathbf{x})$
  for $i = 1$ to $b$ do
      Generate: $\mathbf{x}_1^*, ..., \mathbf{x}_n^*$ iid $\hat{F}(\mathbf{x})$
      $\hat{\xi}_n^{*(i)} = h_n(\mathbf{x}_1^*, ..., \mathbf{x}_n^*)$
  Estimate: $F_n(\xi)$ from $\hat{\xi}_n^{*(1)}, ..., \hat{\xi}_n^{*(b)} \to \hat{F}_n^*(\xi)$
```

The approximation depends on the number of Monte Carlo samples, and the finite sample pdf can be fully determined when the number of Monte Carlo samples tends to infinity, $\hat{F}_n(\xi) \overset{m\to\infty}{\longrightarrow} F_n(\xi)$.

If $F(\mathbf{x})$ is unknown, however, Monte Carlo assessment is unavailable and one may rely on the bootstrap technique. Consider a set of observations $\mathbf{x}_1, ..., \mathbf{x}_n$ which is assumed to be iid observations from $F(\mathbf{x})$, and hence can be used to obtain an estimate of $F(\mathbf{x})$, termed $\hat{F}(\mathbf{x})$. When $F(\mathbf{x})$ is completely unspecified, the non parametric estimate $\hat{F}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} I(\mathbf{x}_i < \mathbf{x})$ can be used. The non-parametric bootstrap algorithm is given in Algorithm 3, see Efron (1979). In the EnKF setting one will naturally resample the ensemble members $[\mathbf{x}^{u(i)}; i = 1, ..., n_{\mathbf{e}}]$. The bootstrap sampling may also be performed in a semi-parametric or parametric setting. In the EnKF setting semi-parametric bootstrapping could be centered at the traditional Kalman gain estimate $\widehat{\Gamma}_{\mathbf{x},\mathbf{d}}(\widehat{\Sigma}_{\mathbf{d}})^{-1}$, and the deviations from this $\hat{K}$ could be bootstrapped. The exact algorithm will be described in the next section. Parametric bootstrapping replaces $\hat{F}(\mathbf{x})$ with a parametric model, and in the EnKF setting it is natural to bootstrap $\mathbf{x}^u$ from $N_{p_{\mathbf{x}}}(\hat{\boldsymbol{\mu}}_{\mathbf{x}}, \widehat{\Sigma}_{\mathbf{x}})$, where $\hat{\boldsymbol{\mu}}_{\mathbf{x}}$ and $\widehat{\Sigma}_{\mathbf{x}}$ are estimates of $\boldsymbol{\mu}_{\mathbf{x}}$ and $\Sigma_{\mathbf{x}}$, respectively, based on the ensemble $\mathbf{e}$. Note that unless $n_{\mathbf{e}} > p_{\mathbf{x}}$ the estimate $\widehat{\Sigma}_{\mathbf{x}}$ will have reduced rank. We solve this problem by adding a small positive number to the zero-valued eigenvalues, which in practice entails regularization.

One important difference between bootstrapping and Monte Carlo simulation is that the bootstrap estimate of the finite sample pdf is not asymptotically correct when the number of bootstrap samples tends to infinity since sampling is done from an approximate pdf.

5. Resampling the EnKF

The non-parametric EnKF resampling is performed along the lines of Expression (19), recognizing that $K = \Gamma_{\mathbf{x},\mathbf{d}}\Sigma_{\mathbf{d}}^{-1}$. Hence the cdf of relevance is $F(\mathbf{x}^u, \mathbf{d}) = F(\mathbf{d}|\mathbf{x}^u)F(\mathbf{x}^u)$ from which the characteristics $\Gamma_{\mathbf{x},\mathbf{d}}$ and $\Sigma_{\mathbf{d}}$ can be determined. The associated finite sample estimators are the classical estimators given in Expressions (11) and (12). The cdf $F(\mathbf{x})$ is only assessable through the ensemble members $(\mathbf{x}^{u(1)}, ..., \mathbf{x}^{u(n_{\mathbf{e}})})$, and may be bootstrapped by the nonparametric $\hat{F}(\mathbf{x}^u)$, see Algorithm 3. The conditional cdf $F(\mathbf{d}|\mathbf{x}^u)$ is defined by the likelihood model $\nu(\mathbf{x}^u, \boldsymbol{\epsilon}^{\mathbf{d}})$ which is fully specified by the known function $\nu(.,.)$ and the known pdf of $\boldsymbol{\epsilon}^{\mathbf{d}}$. Consequently, $F(\mathbf{d}|\mathbf{x}^u)$ can be assessed by Monte Carlo simulation, see Algorithm 2.

One resample replicate of the Kalman gain K^* in Expression (19) is generated by one bootstrap sample from $\hat{F}(\mathbf{x}^u)$, to obtain $(\mathbf{x}^{u*(1)}, ..., \mathbf{x}^{u*(n_{\mathbf{e}})})$. For each bootstrap sample, Monte Carlo sampling from $F(\mathbf{d}|\mathbf{x}^{u*})$ is performed to obtain $[(\mathbf{x}^{u*(i)}, \mathbf{d}^{*(i,j)}); i = 1, ..., n_{\mathbf{e}}, j = 1, ..., m]$. Based on these realizations from $F(\mathbf{x}^u, \mathbf{d})$ the estimates $\widehat{\Gamma}^*_{\mathbf{x},\mathbf{d}}$ and $\widehat{\Sigma}^*_{\mathbf{d}}$ are computed to provide one resample replicate of the Kalman gain $K^* = \widehat{\Gamma}^*_{\mathbf{x},\mathbf{d}}(\widehat{\Sigma}^*_{\mathbf{d}})^{-1}$.

For $n_{\mathbf{e}} >> \min(p_{\mathbf{x}}, p_{\mathbf{d}})$ full rank of $\widehat{\Gamma}^*_{\mathbf{x},\mathbf{d}}$ will be ensured. Note that if $n_{\mathbf{e}} < \min(p_{\mathbf{x}}, p_{\mathbf{d}})$, the rank of $\widehat{\Gamma}^*_{\mathbf{x},\mathbf{d}}$ will vary dependent on the number of duplicates in the bootstrap sample. The number of Monte Carlo replicates for each bootstrap sample can be chosen freely such that $m \geq p_{\mathbf{d}}$, hence full rank of $\widehat{\Sigma}^*_{\mathbf{d}}$ can be ensured.

The number of bootstrap replicates should be identical to the number of Kalman gain replicates required in Expression (19), i.e. $K^{*(i)}; i = 1, ..., n_{\mathbf{e}}$. Hence one replicate for each member in the unconditioned ensemble $(\mathbf{x}^{u(1)}, ..., \mathbf{x}^{u(n_{\mathbf{e}})})$ in order to perform the conditioning.

The additional computational demands from the resampling strategy are to recompute the Kalman gain $n_{\mathbf{e}}$ times in the bootstrapping step and to recompute the likelihood function m times in the Monte Carlo step. The former will normally be relatively inexpensive, while the cost of the latter depends on the actual likelihood model.

The procedure described above defines the non-parametric Resample EnKF algorithm, coined ResEnKF, see Algorithm 4. The basis for ResEnKF is only the ensemble members and the given likelihood model. No additional model assumptions are made. The resampled Kalman gains $K^{*(1)}_{\mathbf{e}}, ..., K^{*(n_{\mathbf{e}})}_{\mathbf{e}}$ will not be independent due to coupling through the ensemble. They will, however, reproduce more of the variability than the single Kalman gain estimate $\widehat{K}$ will. Consequently, the coupling will be reduced.

The semi-parametric EnKF resampling (ResSPEnKF) is centered at the traditional regression line of $\mathbf{d}$ with respect to $\mathbf{x}$, i.e., $\widehat{\Gamma}_{\mathbf{d},\mathbf{x}} \widehat{\Sigma}^{-1}_{\mathbf{x}}$. Define a super ensemble $[(\mathbf{x}^{u(i)}, \mathbf{d}^{(i,j)}); i = 1, ..., n_{\mathbf{e}}, j = 1, ..., m]$ with observations obtained from Monte Carlo simulation from $F(\mathbf{d}|\mathbf{x})$. Generate the associated deviations ensemble $[\triangle^{(i,j)} = \mathbf{d}^{(i,j)} - \widehat{\Gamma}_{\mathbf{d},\mathbf{x}}\widehat{\Sigma}_{\mathbf{x}}^{-1}\mathbf{x}^{u(i)}; i = 1, ..., n_{\mathbf{e}}, j = 1, ..., m]$. Bootstrap $(n_{\mathbf{e}} \times m)$ samples from the deviations ensemble and define the semi-parametric bootstrap ensemble $[(\mathbf{x}^{u(i)}, \mathbf{d}^{*(i,j)} = \widehat{\Gamma}_{\mathbf{d},\mathbf{x}}\widehat{\Sigma}_{\mathbf{x}}^{-1}\mathbf{x}^{u(i)} + \triangle^{*(i,j)}); i = 1, ..., n_{\mathbf{e}}, j = 1, ..., m]$. From this ensemble one semi-parametric bootstrap Kalman gain K^* can be determined. The resampling must be repeated $n_{\mathbf{e}}$ times to obtain $K^{*(i)}; i = 1, ..., n_{\mathbf{e}}$. Note that the $\mathbf{x}^u$ entries are left unchanged, while the associated $\mathbf{d}^*$ values are resampled.

The parametric EnKF resampling (ResPEnKF) is based on resampling of $\mathbf{x}^u$ from $N_{p_{\mathbf{x}}}(\hat{\mu}_{\mathbf{x}}, \widehat{\Sigma}_{\mathbf{x}})$, were the parameters are traditional estimates of $\mu_{\mathbf{x}}$ and $\Sigma_{\mathbf{x}}$ based on the ensemble $\mathbf{e}$. Simulate $n_{\mathbf{e}}$ samples of $\mathbf{x}^u$ and Monte Carlo simulate associated values from $F(\mathbf{d}|\mathbf{x}^u)$ to obtain $[(\mathbf{x}^{u*(i)}, \mathbf{d}^{*(i,j)}); i = 1, ..., n_{\mathbf{e}}, j = 1, ..., m]$. Based on this set the parametric bootstrap Kalman gain K^* can be computed. Repeat the procedure $n_{\mathbf{e}}$ times to obtain $K^{*(i)}; i = 1, ..., n_{\mathbf{e}}$. Note that the $\mathbf{x}^u$ entries are resampled from $N_{p_{\mathbf{x}}}(\hat{\mu}_{\mathbf{x}}, \widehat{\Sigma}_{\mathbf{x}})$ for each Kalman gain computation. The various EnKF resampling approaches are evaluated in an empirical study in the next section.

6. Empirical study

In order to evaluate the impact of the various resampling strategies in EnKF, we define a Gaussian prior model and two different likelihood models, one Gauss-linear and one non-linear. The Gauss-linear model with all model parameters known is analytically tractable and will act as a reference. This model is described in Myrseth & Omre (2010). The model with a nonlinear observation likelihood demonstrates the generality of the algorithm. We start out, however, with a simple bivariate example and evaluate a one-step update. This simple case illustrates the effect of using resampling.

Algorithm 4: Non-parametric Resampled Ensemble Kalman Filter

```
Initiate:
  • $n_\mathbf{e} =$ no. of ensemble members
  • $m =$ no. of Monte Carlo replicates
  • $\mathbf{x}_0^{u(i)}$ ; $i = 1, ..., n_\mathbf{e}$ iid $f(\mathbf{x}_0)$
  • $\boldsymbol{\epsilon}_0^{\mathbf{d}(i)} \sim N_{p_\mathbf{d}}(\mathbf{0}, \mathbf{I}_{p_\mathbf{d}})$ ; $i = 1, ..., n_\mathbf{e}$
  • $\mathbf{d}_0^{(i)} = \nu_t(\mathbf{x}_0^{u(i)}, \boldsymbol{\epsilon}_0^{\mathbf{d}(i)})$ ; $i = 1, ..., n_\mathbf{e}$
  • $\mathbf{e}_0 \ : \ \{(\mathbf{x}_0^u, \mathbf{d}_0)^{(i)} ; i = 1, ..., n_\mathbf{e}\}$
for $t = 0$ to $T$ do
    Conditioning:
    Estimate $\hat{F}(\mathbf{x}_t^u) = \frac{1}{n}\sum_{i=1}^{n} I(\mathbf{x}_t^{u(i)} < \mathbf{x}_t^u)$
    for $j = 1$ to $n_\mathbf{e}$ do
        • $\mathbf{x}_t^{u*(i)} \sim \hat{F}(\mathbf{x})$ ; $i = 1, ..., n_\mathbf{e}$
        • $\mathbf{G} = \mathbf{0}$
        • $\mathbf{S} = \mathbf{0}$
        for $k = 1$ to $m$ do
            • $\boldsymbol{\epsilon}_t^{\mathbf{d}(i),k} \sim N_{p_\mathbf{d}}(\mathbf{0}, \mathbf{I}_{p_\mathbf{d}})$ ; $i = 1, ..., n_\mathbf{e}$
            • $\mathbf{d}_t^{(i),k} = \nu_t(\mathbf{x}_t^{u*(i)}, \boldsymbol{\epsilon}_t^{\mathbf{d}(i),k})$ ; $i = 1, ..., n_\mathbf{e}$
            • $\mathbf{G} = \mathbf{G} + \frac{1}{n_\mathbf{e}-2}\sum_{i=1}^{n_\mathbf{e}}(\mathbf{x}_t^{u*(i)} - \widehat{\boldsymbol{\mu}}_\mathbf{x})(\mathbf{d}_t^{(i),k} - \widehat{\boldsymbol{\mu}}_\mathbf{d}^k)'$
            • $\mathbf{S} = \mathbf{S} + \frac{1}{n_\mathbf{e}-1}\sum_{i=1}^{n_\mathbf{e}}(\mathbf{d}_t^{(i),k} - \widehat{\boldsymbol{\mu}}_\mathbf{d}^k)(\mathbf{d}_t^{(i),k} - \widehat{\boldsymbol{\mu}}_\mathbf{d}^k)'$
        • $\widehat{\Gamma}_{\mathbf{x},\mathbf{d}}^* = \frac{1}{m}\mathbf{G}$
        • $\widehat{\Sigma}_\mathbf{d}^* = \frac{1}{m}\mathbf{S}$
        • $\mathbf{x}_t^{c(i)} = \mathbf{x}_t^{u(i)} + \widehat{\Gamma}_{\mathbf{x},\mathbf{d}}^*(\widehat{\Sigma}_\mathbf{d}^*)^{-1}(\mathbf{d}_t - \mathbf{d}_t^{(i)})$
    Forwarding:
      • $\boldsymbol{\epsilon}_t^{\mathbf{x}(i)} \sim N_{p_\mathbf{x}}(\mathbf{0}, \mathbf{I}_{p_\mathbf{x}})$ ; $i = 1, ..., n_\mathbf{e}$
      • $\mathbf{x}_{t+1}^{u(i)} = \omega_t(\mathbf{x}_t^{c(i)}, \boldsymbol{\epsilon}_t^{\mathbf{x}(i)})$ ; $i = 1, ..., n_\mathbf{e}$
      • $\boldsymbol{\epsilon}_{t+1}^{\mathbf{d}(i)} \sim N_{p_\mathbf{d}}(\mathbf{0}, \mathbf{I}_{p_\mathbf{d}})$ ; $i = 1, ..., n_\mathbf{e}$
      • $\mathbf{d}_{t+1}^{(i)} = \nu_{t+1}(\mathbf{x}_{t+1}^{u(i)}, \boldsymbol{\epsilon}_{t+1}^{\mathbf{d}(i)})$ ; $i = 1, ..., n_\mathbf{e}$
      • $\mathbf{e}_{t+1} \ : \ \{(\mathbf{x}_{t+1}^u, \mathbf{d}_{t+1})^{(i)} ; i = 1, ..., n_\mathbf{e}\}$
Assess
  • $f(\mathbf{x}_{T+1}|\mathbf{d}_0, ..., \mathbf{d}_T)$ from $\mathbf{e}_{T+1}$
```

6.1. Simple bivariate example

The minimalistic example consider a bivariate state variable $\mathbf{x}^u$ with distribution $N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the expectation vector $\boldsymbol{\mu} = (1, 1)^T$ and the covariance matrix $\mathbf{\Sigma}$ has diagonal terms $\sigma_{11} = \sigma_{22} = 1$ and off-diagonal terms $\sigma_{12} = \sigma_{21} = 0.37$. The likelihood model is $[\mathbf{d}|\mathbf{x}^u] = \boldsymbol{H}\mathbf{x}^u + \boldsymbol{\epsilon}_{\mathbf{d}}$ with $h_{11} = h_{22} = 1$ and $h_{12} = h_{21} = 0.5$, while each element of the the error term, $\boldsymbol{\epsilon}_{\mathbf{d}}$ is independent Gaussian with zero mean and variance 0.1. The actual observations are $\mathbf{d}^o = (-2.36, -0.79)^T$. Consequently, the example has $n_{\mathbf{x}} = n_{\mathbf{d}} = 2$, and we focus on the conditional state variable $\mathbf{x}^c = [\mathbf{x}^u|\mathbf{d}^o]$ with associated pdf $f(\mathbf{x}^c)$.

We assess the pdf of interest $f(\mathbf{x}^c)$ using both the traditional EnKF, Algorithm 1, and resample ResEnKF, Algorithm 4, with a range of ensemble sizes, $n_{\mathbf{e}} = 6, \ldots, 20$. Note that for the Gauss-linear assumptions used, both algorithms are asymptotically correct when $n_{\mathbf{e}} \to \infty$.

The evaluation criteria for the algorithms are: Mean Square Error (MSE), where MSE $=$ tr $\hat{\mathrm{E}}[(\hat{\boldsymbol{\mu}}_{\mathbf{x}^c} - \boldsymbol{\mu}_{\mathbf{x}^c})(\hat{\boldsymbol{\mu}}_{\mathbf{x}^c} - \boldsymbol{\mu}_{\mathbf{x}^c})^T]$, and Correlation between Ensemble Members (CEM) defined as tr $\widehat{\mathrm{Corr}}(\mathbf{x}^{c(i)}, \mathbf{x}^{c(j)})$ for $i \neq j$. Note that the correlation between updated ensemble members is symmetric, see Expression (18), hence identical for all $i, j = 1, \ldots, n_{\mathbf{e}}$, $i \neq j$. These criteria are assessed by averages over 10 000 Monte Carlo simulations from the model.

The results from the evaluation are displayed in Figure 2. The MSE and the CEM are displayed to the left and right respectively for $n_e = 6, \ldots, 20$. While the MSE is indeed higher for the ResEnKF, we observe that the CEM is significantly reduced compared with the traditional EnKF. An increased MSE for the ResEnKF scheme can be expected as the rank of the empirically estimated covariance matrices decrease. As the ensemble size increases both the MSE and the CEM appear to converge. The convergences of both the MSE and the CEM curves are caused by both algorithms being asymptotically correct for $n_{\mathbf{e}} \to \infty$. Note that in a sequential data assimilation setting, the increased MSE for the ResEnKF should not be critical as this will be corrected for at later updating steps. Introducing a large ensemble correlation, however, would be more critical as there will be a multiplicative effect at later updating steps, especially when considering static forward models. This increased ensemble coupling can eventually lead to an ensemble collapse. The consequences of sequential updating in higher-dimensional problems will be evaluated in the next section.

6.2. Model description

The variables of interest are $[\mathbf{x}_0, ..., \mathbf{x}_{11}]$, where $\mathbf{x}_t \in \mathbb{R}^{100}$; hence $\mathbf{x}_t$ is a 100- dimensional time series. Observations are available at $[\mathbf{d}_0, ..., \mathbf{d}_{10}]$. The current time is $T = 10$ and the objective is the forecast $[\mathbf{x}_{11}|\mathbf{d}_0, ..., \mathbf{d}_{10}]$. In Figure 3 the reference realization of $[\mathbf{x}_{10}, \mathbf{d}_{10}]$ is presented.

The test case is defined as follows:

$$f(\mathbf{x}_0) \sim N_{100}(\mathbf{0}, \Sigma_0^x) \tag{21}$$

$$[\mathbf{x}_{t+1}|\mathbf{x}_t] = \mathbf{A}_t \mathbf{x}_t \tag{22}$$

where the initial covariance matrix Σ_0^x contains elements

$$\sigma_{i,j}^x = 20 \exp(-3|i - j|/20) \tag{23}$$

for $i, j = 1, ..., 100$. The forward model defined by $\mathbf{A}_t$ is a linear smoother that moves in steps of 5 from left to right for each time step. Consequently, the left part of $\mathbf{x}_{10}$ is smoother than the right part. The example has been inspired by a fluid flow scenario where there is a moving front where the parameters are dynamic, and static surroundings. For more detail, see Myrseth & Omre (2010).

The likelihood models are

$$[\mathbf{d}_t|\mathbf{x}_t]_0 = H_t\mathbf{x}_t + \sqrt{20}\boldsymbol{\epsilon}_t^{\mathbf{d}} \tag{24}$$

and

$$[\mathbf{d}_t|\mathbf{x}_t]_1 = (H_t\mathbf{x}_t) \circ \exp(\sqrt{0.1}\boldsymbol{\epsilon}_t^{\mathbf{d}}) \tag{25}$$

where $\boldsymbol{\epsilon}_t^{\mathbf{d}} \sim N_{10}(\mathbf{0}, \mathbf{I}_{10})$, H_t is time-invariant and picks 10 locations, see Figure 3 and $\circ$ denotes a Schur product. The nonlinear likelihood contains a log normal multiplicative error structure.

6.3. Results

The forecast $[\mathbf{x}_{11}|\mathbf{d}_0, ..., \mathbf{d}_{10}]$ will be used to measure the impact of resampling. The Root Mean Square Error (RMSE) of the mean forecast will be used to measure accuracy. The coverage will be used to measure forecast uncertainty which captures both accuracy and precision. A 95% coverage interval should include the solution 95% of the time. If the coverage is lower than this then the 95% forecast interval underestimates the uncertainty. The examples are run with the ensemble sizes $n_{\mathbf{e}} = 30$ and $n_{\mathbf{e}} = 100$ with $m = 50$ Monte Carlo samples. An empirical 95% prediction interval is defined to be spanned by the 28 and 96 central ensemble members for the two ensemble sizes. We should therefore expect 87.1% and 94.1% coverage for $n_{\mathbf{e}} = 30$ and $n_{\mathbf{e}} = 100$, respectively, see Wilks (1962).

6.4. Gauss-linear likelihood model

For the Gauss-linear model, the model parameters are available through the traditional Kalman Filter, this exact solution is presented in Figure 4.

For the Gauss-linear model also the resampling approach as outlined in Expression (20) is available. This case is termed the exact finite sample solution, and it captures the uncertainty due to finite size of the ensemble. The traditional EnKF algorithm, Algorithm 1, the ResEnKF, Algorithm 4, adapted to a known linear likelihood, and the ResSPEnKF and ResPEnKF algorithms are also run on this Gauss-linear case.

Figures 5 and 6 and Table 1 display the results obtained for the Gauss-linear model. In Figures 5 and 6 the prediction $[\mathbf{x}_{11}|\mathbf{d}_1, ..., \mathbf{d}_{10}]$ with associated 95% prediction intervals are displayed for one run of each of the algorithms with ensemble sizes $n_\mathbf{e} = 30$ and $n_\mathbf{e} = 100$ respectively. The reference $\mathbf{x}_{11}$ is also displayed. Table 1 contains statistics from 100 repeated runs of each algorithm on the same observations.

Figure 4 contains the analytical solution of the Kalman Filter which is available for this Gauss-linear model. The reduction in prediction uncertainty around each observation is observed. Figures 5(a) and 6(a) present the prediction results for the exact finite sample solutions for $n_\mathbf{e} = 30$ and $n_\mathbf{e} = 100$, see Expression (20). Recall that the Kalman gains are resampled using $\mathbf{x}^u \sim N_{p_\mathbf{x}}(\boldsymbol{\mu}, \Sigma)$ which adds uncertainty relative to the analytical solution which is the limiting case as $n_\mathbf{e} \to \infty$. The $n_\mathbf{e} = 30$ case has larger uncertainty while the $n_\mathbf{e} = 100$ case is very similar to the limiting case. These exact finite sample solutions are the reference solutions for the other ensemble Kalman filter runs. Figures 5(b) and 6(b) contain the results for the traditional EnKF algorithm. The underestimation of the prediction intervals for $n_\mathbf{e} = 30$ is observed and the non-logical increase in uncertainty as $n_\mathbf{e}$ increases is observed for $n_\mathbf{e} = 100$. We interpret the underestimation of uncertainty for $n_{\mathbf{e}} = 30$ to be caused by coupling of ensemble members due to the use of one common estimate of the Kalman gain, as discussed in previous sections. Note that all estimated matrices used in the conditioning have full rank in this case, since $n_{\mathbf{e}} > n_{\mathbf{d}}$. The fact that the exact finite sample solutions, which uses independent Kalman gains for each ensemble member, exposes decreasing uncertainty with increasing $n_{\mathbf{e}}$, supports our interpretation. Figures 5(c) and 6(c) contain the results from the ResEnKF algorithm which includes full bootstrapping of the ensemble members to provide Kalman gain variability and reduce coupling in the conditional ensembles. The prediction intervals for ResEnKF are wider than for traditional EnKF and close to the reference exact finite sample solutions in Figures 5(a) and 6(a). This effect is clearly seen for $n_{\mathbf{e}} = 30$, while the results are more similar for $n_{\mathbf{e}} = 100$. The results for the ResSPEnKF and ResPEnKF algorithms in Figures 5(d) and 6(d) through 5(e) and 6(e) are similar to the ones for full bootstrapping in ResEnKF, although the ResEnKF seems to appear with better coverage of the prediction intervals at lower ensemble sizes.

The results from the Gauss-linear model case can be summarized from Table 1 as follows: The traditional EnKF appears with better prediction accuracy than the ResEnKF algorithm, but the latter assesses the prediction uncertainty more reliably. In the traditional EnKF one uses the best estimate of the Kalman gain on all ensemble members to improve prediction accuracy, but this introduces coupling in the ensemble and hence underestimation of prediction uncertainty. In ResEnKF one resamples the Kalman gain causing loss in prediction accuracy, but this also reduces coupling in the ensemble and hence improves the prediction uncertainty estimates. The two alternative resampling approaches appear less reliable than ResEnKF. The parametric resampling EnKF has smaller RMSE and better coverage for small $n_{\mathbf{e}}$, however, but the geometry of the prediction intervals in Figure 5(e) appears with too little variability. Lastly, note that these results are obtained on a Gauss-linear model which appears as very favorable for the EnKF. Hence the underestimation of the prediction uncertainty should cause concern in more complex models.

6.5. Nonlinear likelihood model

For the nonlinear likelihood the true model parameters are analytically intractable. However, the resampling EnKF algorithm, Algorithm 4, can be used without the bootstrapping loop. Then the likelihood model is linearized by Monte Carlo sampling based linearization around the ensemble. Hence, local ensemble dependent linearization is performed. This approach is termed EnKF/nonlinear likelihood. Finally, the full ResEnKF algorithm, Algorithm 4 and the semi-parametric, ResSPEnKF, and the parametric ResPEnKF, including both bootstrapping and Monte Carlo sampling can be used.

Figures 7 and 8 and Table 2 display the results obtained for the Gaussian prior model with nonlinear likelihood model. The layout is identical to Figures 5 and 6, and Table 1. Figures 7(a) and 8(a) display the EnKF solutions with local, ensemble based linearization of the likelihood model for ensemble sizes $n_{\mathbf{e}} = 30$ and $n_{\mathbf{e}} = 100$. The prediction uncertainties are underestimated for $n_{\mathbf{e}} = 30$ and uncertainty increases with increasing $n_{\mathbf{e}}$ which is non-intuitive. Figures 7(b) and 8(b) contain the results from the ResEnKF algorithm with full bootstrap and Monte Carlo resampling. The prediction intervals for $n_{\mathbf{e}} = 30$ appear as much more reliable than for the previous algorithm, while intervals for $n_e = 100$ are fairly similar. Figures 7(c) through 8(d) display the results from the semi-parametric, ResSPEnKF, and parametric, ResPEnKF, algorithms, and they appear as less reliable than the results from the ResEnKF algorithm.

In Table 2 statistics for 100 repeated runs of each algorithm are summarized. The EnKF with local, ensemble based linearization tends to underestimate the prediction uncertainty. The ResEnKF algorithm has higher RMSE than the EnKF with nonlinear likelihood, but provides more reliable estimates of the prediction intervals. The two alternative resampling approaches appear less reliable than ResEnKF.

7. Conclusion

The traditional EnKF is based on an ensemble representing the pdf of the variable of interest. The ensemble members are sequentially conditioned on observations and forwarded to the next time step. The conditioning to available observations is the challenging part, and in EnKF this conditioning is linearized using weights corresponding to the Kalman gain. The actual Kalman gain is estimated based on all ensemble members. Since the same Kalman gain estimate is used in all the conditioning of all ensemble members it can be shown that the members end up being coupled. Eventually this will cause the prediction intervals to be underestimated. A resampling strategy where the Kalman gain is generated from its sampling distribution is suggested. This will reduce coupling in the conditioning step.

The resampling EnKF, ResEnKF, algorithm is defined for a model with both prior and the likelihood being on general nonlinear form. The coupling of the ensemble members is reduced by bootstrapping the Kalman gains used in the conditioning step. The computational demands of ResEnKF are larger than for EnKF, but only slightly larger. The alternative resampling schemes, ResSPEnKF and ResPEnKF are also defined. The former uses a semi-parametric model in the resampling while the latter uses a resampling from a parametric model.

The various algorithms are evaluated empirically. In a simple, bivariate example with one updating step, the ResEnKF algorithm is demonstrated to be clearly superior to the traditional EnKF algorithm in terms of reducing the ensemble correlation. This comes at the cost of a lower prediction error compared to the traditional EnKF algorithm. A Gauss-linear model is also used and the exact finite sample prediction intervals are generated as reference. It is shown that the traditional EnKF algorithm severely underestimates the prediction intervals for small ensemble sizes. The ResEnKF has significantly higher coverage of the prediction intervals on the expense of somewhat larger mean square error of prediction itself when compared to the EnKF. The alternative resampling schemes appear less reliable.

The algorithms are also evaluated on a Gaussian prior model with nonlinear likelihood model. The ResEnKF algorithm with bootstrapping appears as more reliable than the traditional EnKF for small ensemble sizes. Overall the ResEnKF algorithm outperformed the traditional EnKF by providing more reliable prediction intervals on the expense of slightly lower prediction accuracy. The ResEnKF algorithm requires no extra modeling and has only slightly larger computational demands than the EnKF algorithm. None of the two alternative resampling schemes seem to provide results that are more reliable than the full bootstrapping algorithm ResEnKF.

8. Acknowledgments

Parts of this work has been funded by the Uncertainty in Reservoir Evaluation (URE) initiative at NTNU.

References

Anderson, J. (2001). An ensemble adjustment ensemble Kalman filter for data assimilation. Monthly Weather Review, 127, 2741–2758.

Bertino, L., Evensen, G., & Wackernagel, H. (2002). Combining geostatistics and Kalman filtering for data assimilation in an estuarine system. Inverse Problems, 18(1), 1–23.

Burgers, G., van Leeuwen, P. J., & Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter. Monthly Weather Review, 126(6), 1719–1724.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. The Annals of Statistics, 7(1), 1–26.

Efron, B. & Tibshirani, R. (1993). An Introduction to the Bootstrap. Chapman & Hall, New York.

Evensen, G. (1994). Sequential data assimilation with nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research, 99(C5), 10,143–10,162.

Evensen, G. (2007). Data Assimilation; The Ensemble Kalman Filter. Springer.

Furrer, R. & Bengtsson, T. (2007). Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants. Journal of Multivariate Analysis, 98(2), 227–255.

Hammersley, J. M. & Handscomb, D. C. (1964). Monte Carlo Methods. London & New York: Chapman & Hall.

Houtekamer, P. & Mitchell, H. L. (1998). Data assimilation using an ensemble Kalman filter technique. Monthly Weather Review, 126(3), 796–811.

Houtekamer, P., Mitchell, H. L., Pellerin, G., Buehner, M., Charron, M., Spacek, L., & Hansen, B. (2005). Atmospheric data assimilation with an ensemble Kalman filter: Results with real observations. Monthly Weather Review, 133(3), 604–620.

Huber, P. J. (1981). Robust Statistics. Wiley.

Myrseth, I. & Omre, H. (2010). Hierarchical ensemble kalman filter. SPE Journal, 15(2), 569–580.

Nævdal, G., Johnsen, L. M., Aanonsen, S. I., & Vefring, E. H. (2005). Reservoir monitoring and continuous model updating using ensemble Kalman filter. SPE Journal, 10(1), 66–74.

Wilks, S. S. (1962). Mathematical Statistics. Wiley.

List of Figures

List of Tables

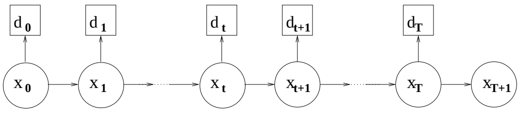
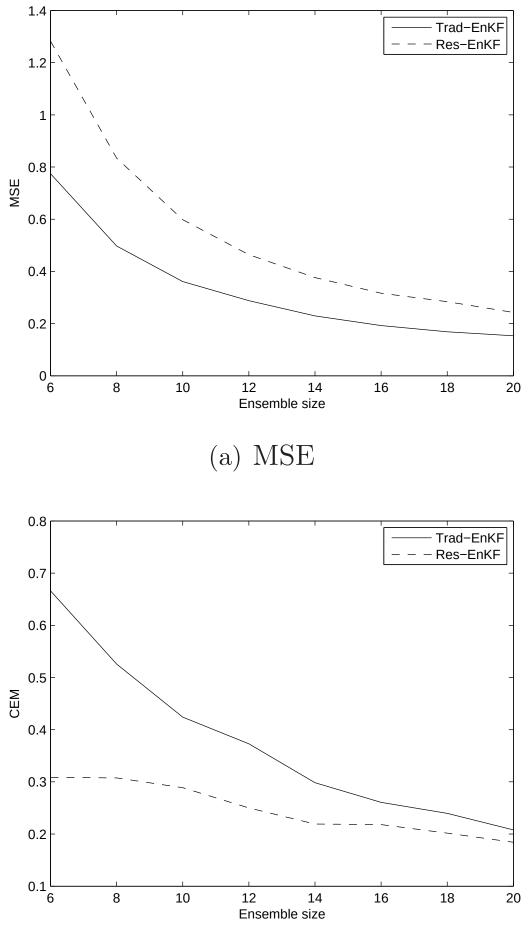


Figure 1: Hidden Markov process



(a) MSE

(b) CEM

Figure 2: Empirical evaluation of MSE and CEM, using the Traditional EnKF and Resampling EnKF schemes.

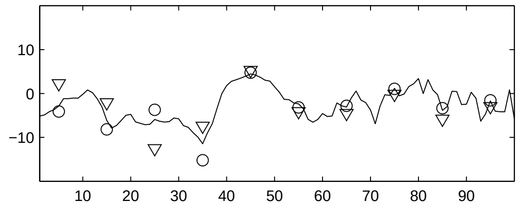


Figure 3: Reference realization: $(\mathbf{x}_{10}, \mathbf{d}_{10})$. Observations from the linear likelihood marked by triangles. Observations from the nonlinear likelihood marked by circles.

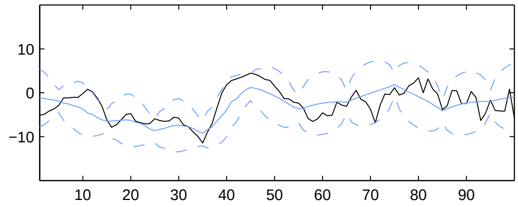
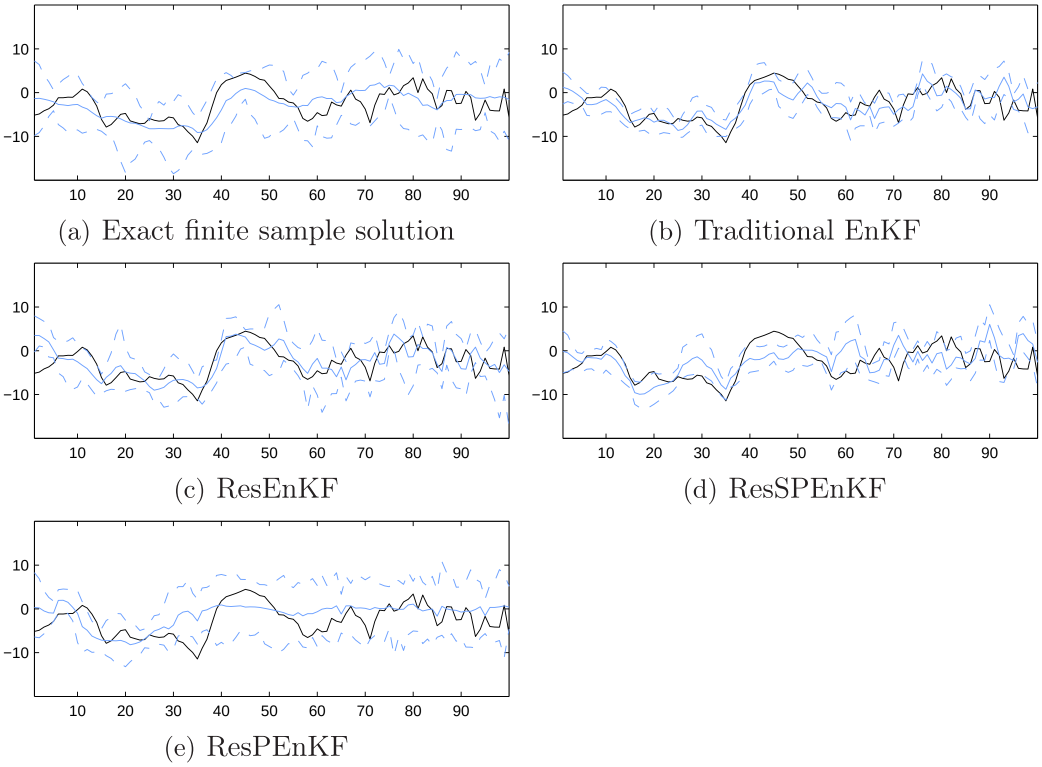


Figure 4: Exact solution for the Gauss-linear model. The reference realization (black), predictions (solid blue) and 95% prediction intervals (hatched blue).



(a) Exact finite sample solution

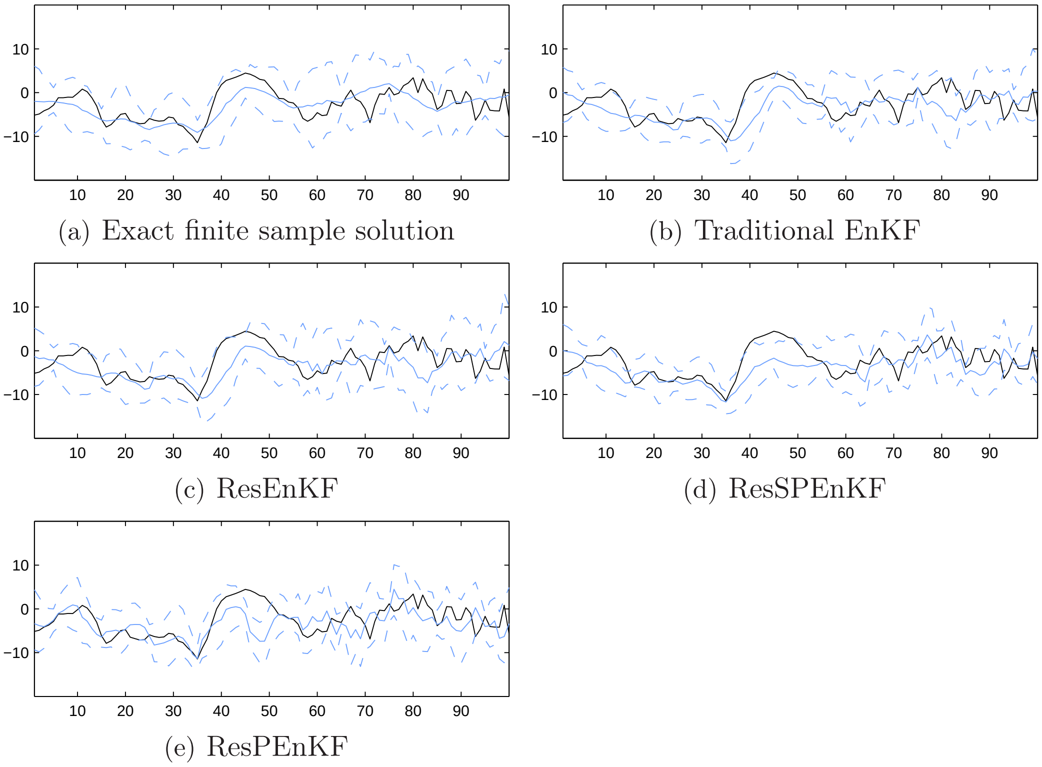
(b) Traditional EnKF

(c) ResEnKF

(d) ResSPEnKF

(e) ResPEnKF

Figure 5: Gauss-linear model. The reference realization (black), predictions (solid blue) and 95%-empirical prediction intervals (hatched blue) for one run of each of the EnKF algorithms with ensemble size $n_{\mathbf{e}} = 30$.



(a) Exact finite sample solution

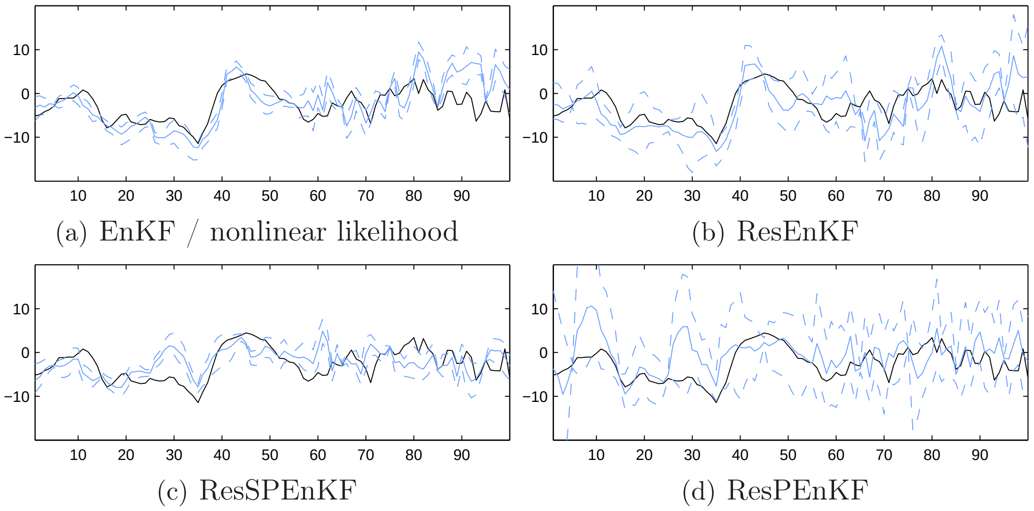
(b) Traditional EnKF

(c) ResEnKF

(d) ResSPEnKF

(e) ResPEnKF

Figure 6: Gauss-linear model. The reference realization (black), predictions (solid blue) and 95%-empirical prediction intervals (hatched blue) for one run of each of the EnKF algorithms with ensemble size $n_{\mathbf{e}} = 100$.



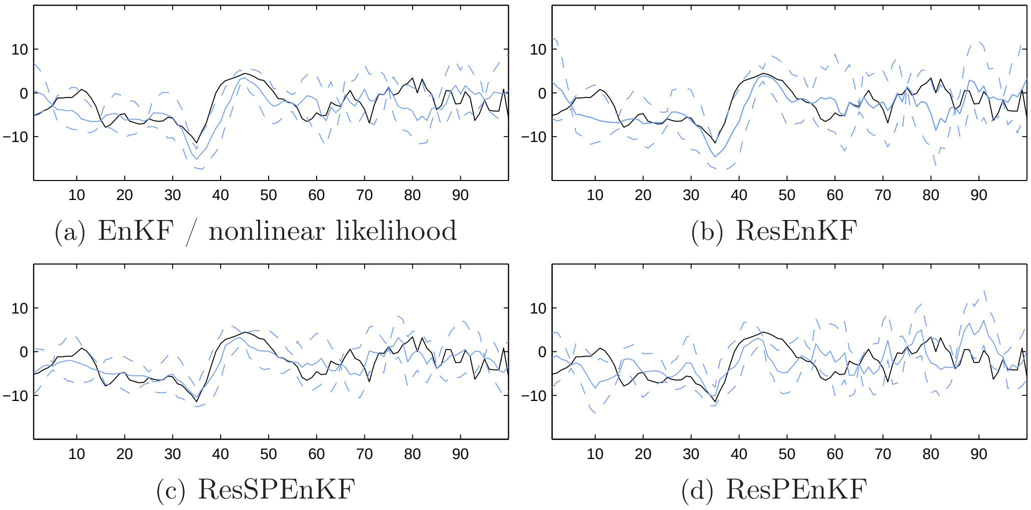
(a) EnKF / nonlinear likelihood

(b) ResEnKF

(c) ResSPEnKF

(d) ResPEnKF

Figure 7: Gauss-nonlinear model. The reference realization (black), predictions (solid blue) and 95%-empirical prediction intervals (hatched blue) for one run of each of the EnKF algorithm with ensemble size $n_{\mathbf{e}} = 30$.



(a) EnKF / nonlinear likelihood

(b) ResEnKF

(c) ResSPEnKF

(d) ResPEnKF

Figure 8: Gauss-nonlinear model. The reference realization (black), predictions (solid blue) and 95%-empirical prediction intervals (hatched blue) for one run of each of the EnKF algorithm with ensemble size $n_{\mathbf{e}} = 100$.

	$n_{\mathbf{e}}$	RMSE	Coverage (%)
Exact solution		2.68	95.0
Exact finite sample solution	30	2.75	97.3
Traditional EnKF	30	3.55	62.3
ResEnKF	30	3.92	74.0
ResSPEnKF	30	3.98	58.7
ResPEnKF	30	3.79	86.1
Exact finite sample solution	100	2.70	98.1
Traditional EnKF	100	2.93	88.8
ResEnKF	100	3.00	93.5
ResSPEnKF	100	3.31	83.1
ResPEnKF	100	3.52	84.8

Table 1: Gauss-linear model. RMSE and 95% empirical coverage for the different algorithms with a Gauss-linear likelihood model averaged over 100 runs.

	$n_{\mathbf{e}}$	RMSE	Coverage (%)
EnKF / nonlinear likelihood	30	4.67	40.1
ResEnKF	30	5.81	67.4
ResSPEnKF	30	3.92	43.1
ResPEnKF	30	5.97	80.4
EnKF / nonlinear likelihood	100	2.95	82.0
ResEnKF	100	3.10	93.0
ResSPEnKF	100	2.93	81.3
ResPEnKF	100	3.36	82.4

Table 2: Gauss-nonlinear model. RMSE and 95% empirical coverage for the different algorithms with a nonlinear likelihood model averaged over 100 runs.