Assimilation Algorithms
Lecture 4: Kalman Filter Methods

Mike Fisher

ECMWF

May 10, 2010

Outline

The Extended Kalman Filter

- In my first lecture, I derived the linear analysis equation:

$$\mathbf{x}_k^a = \mathbf{x}_k^b + \mathbf{K}_k \left(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k^b) \right)$$

  - NB: I have added a subscript k to show that the analysis, background, observations, etc. are all valid for some time t_k.

- I showed that the optimal choice for $\mathbf{K}_k$ is the Kalman Gain Matrix:

$$\mathbf{K}_k = \mathbf{P}_k^b \mathbf{H}_k^\mathrm{T} \left[\mathbf{H}_k \mathbf{P}_k^b \mathbf{H}_k^\mathrm{T} + \mathbf{R}_k \right]^{-1} \equiv \left[(\mathbf{P}_k^b)^{-1} + \mathbf{H}_k^\mathrm{T} \mathbf{R}_k^{-1} \mathbf{H}_k \right]^{-1} \mathbf{H}_k^\mathrm{T} \mathbf{R}_k^{-1}$$

- I gave the following expression for the covariance matrix of analysis error:

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^b (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\mathrm{T} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\mathrm{T}$$

- Now we will consider how to generate $\mathbf{P}_k^b$ in an optimal way.

The Extended Kalman Filter

- In most applications of data assimilation, we do not just want to produce a single analysis for one given time.
- Rather, we are interested in a sequence of analyses for times $t_0, t_1, \cdots$, etc.
- For each analysis in this sequence, we require background $\mathbf{x}_k^b$ (i.e. a prior estimate of the state at time t_k).
- Our best prior estimate of the state at time t_k is given by a forecast from the preceding analysis:

$$\mathbf{x}_k^b = \mathcal{M}_{t_{k-1} \to t_k} \left(\mathbf{x}_{k-1}^a \right)$$

- What is the error covariance matrix associated with this background?

The Extended Kalman Filter

$$\mathbf{x}_k^b = \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}_{k-1}^a\right)$$

- Subtract the true state at time t_k from both sides:

$$\epsilon_k^b = \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}_{k-1}^a\right) - \mathbf{x}_k^*$$

- Now write $\mathbf{x}_{k-1}^a = \mathbf{x}_{k-1}^* + \epsilon_{k-1}^a$ and assume that ϵ_{k-1}^a is small enough for the following linear approximation to be valid:

$$\mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}_{k-1}^a\right) \approx \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}_{k-1}^*\right) + \mathbf{M}_{t_{k-1}\rightarrow t_k}\epsilon_{k-1}^a$$

- Then:

$$\begin{aligned}\epsilon_k^b &= \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}_{k-1}^*\right) + \mathbf{M}_{t_{k-1}\rightarrow t_k}\epsilon_{k-1}^a - \mathbf{x}_k^* \\ &= \mathbf{M}_{t_{k-1}\rightarrow t_k}\epsilon_{k-1}^a + \eta_k\end{aligned}$$

where $\eta_k = \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}_{k-1}^*\right) - \mathbf{x}_k^*$ is the model error.

The Extended Kalman Filter

$$\eta_k = \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}^*_{k-1}\right) - \mathbf{x}^*_k$$

- We will assume that $\overline{\epsilon^a_{k-1}} = \overline{\eta_k} = 0 \quad \Rightarrow \quad \overline{\epsilon^b_k} = 0$.
- The covariance matrix of background error is:

$$\overline{\epsilon^b_k(\epsilon^b_k)^{\mathrm{T}}} = \overline{\left(\mathbf{M}_{t_{k-1}\rightarrow t_k}\epsilon^a_{k-1} - \eta_k\right)\left(\mathbf{M}_{t_{k-1}\rightarrow t_k}\epsilon^a_{k-1} - \eta_k\right)^{\mathrm{T}}}$$

- Assuming that analysis error and model error are uncorrelated, we can multiply this out to get:

$$\overline{\epsilon^b_k(\epsilon^b_k)^{\mathrm{T}}} = \mathbf{M}_{t_{k-1}\rightarrow t_k}\overline{\epsilon^a_{k-1}(\epsilon^a_{k-1})^{\mathrm{T}}}\mathbf{M}^{\mathrm{T}}_{t_{k-1}\rightarrow t_k} + \overline{\eta_k\eta_k^{\mathrm{T}}}$$

- I.e.

$$\mathbf{P}^b_k = \mathbf{M}_{t_{k-1}\rightarrow t_k}\mathbf{P}^a_{k-1}\mathbf{M}^{\mathrm{T}}_{t_{k-1}\rightarrow t_k} + \mathbf{Q}_k$$

where $\mathbf{Q}_k = \overline{\eta_k\eta_k^{\mathrm{T}}}$ is the covariance matrix of model error.

The Extended Kalman Filter

- We now have all the equations we need to analyse and propagate the state, and to compute and propagate the covariances:

$$
\begin{aligned}
\mathbf{x}_k^b &= \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}_{k-1}^a\right) \\
\mathbf{P}_k^b &= \mathbf{M}_{t_{k-1}\rightarrow t_k}\mathbf{P}_{k-1}^a\mathbf{M}_{t_{k-1}\rightarrow t_k}^{\mathrm{T}} + \mathbf{Q}_k \\
\mathbf{K}_k &= \mathbf{P}_k^b\mathbf{H}_k^{\mathrm{T}}\left[\mathbf{H}_k\mathbf{P}_k^b\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right]^{-1} \\
\mathbf{x}_k^a &= \mathbf{x}_k^b + \mathbf{K}_k\left(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k^b)\right) \\
\mathbf{P}_k^a &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^b(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}}
\end{aligned}
$$

- These equations define the Extended Kalman Filter.
  - Note: the "extended" qualifier refers to the fact that we allow non-linear observation operators, and propagate the state using a nonlinear model. The standard Kalman filter is a purely linear analysis system in which $\mathcal{H}_k$ and $\mathcal{M}_{t_{k-1}\rightarrow t_k}$ are assumed to be linear.

The Extended Kalman Filter

- Subject to the assumptions we have made, the Kalman filter produces an optimal sequence of analyses.
- The analysis $\mathbf{x}_k^a$ is the best (minimum variance) estimate of the state at time t_k, given $\mathbf{x}_0^b$ and all observations up to time t_k (i.e. $\mathbf{y}_0 \cdots \mathbf{y}_k$).
- The inputs to the Kalman filter are:
  - An initial estimate of the state at time t_0, and the corresponding covariance matrix, $\mathbf{P}_0^b$.
  - Observations $\mathbf{y}_k$, and covariances of observation error, $\mathbf{R}_k$ at each analysis time.
  - Covariance matrices of model error, $\mathbf{Q}_k$.
- Note that, unlike OI, 3D-Var and 4D-Var, we do not have to specify the covariance matrix of background error — it is generated and propagated by the filter, using the model dynamics.
- However, we do have to specify $\mathbf{Q}_k$. This is difficult!

Kalman Filters for Large Dimensional Systems

- The Kalman filter is impractical for large dimension systems.
- It requires us to handle matrices of dimension $N \times N$, where $N \sim 10^8$.
  - The World's fastest computer can sustain $\sim 10^{15}$ operations per second.
  - Multiplying two $10^8 \times 10^8$ matrices requires 10^{24} operations, and would take about 32 years on this machine.
  - Evaluating $\mathbf{P}_k^b = \mathbf{M}_{t_{k-1} \rightarrow t_k} \mathbf{P}_{k-1}^a \mathbf{M}_{t_{k-1} \rightarrow t_k}^{\mathrm{T}} + \mathbf{Q}_k$ requires $N \sim 10^8$ model integrations.
- A range of approximate Kalman filters have been developed for use with large systems.
- All of these methods rely on a low-rank approximation of the covariance matrices of background and analysis error.

Kalman Filters for Large Dimensional Systems

- Suppose that $\mathbf{P}_k^b$ has rank $M \ll N$ (e.g. $M \sim 100$).
- Then we can write $\mathbf{P}_k^b = \mathbf{X}_k^b \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$, where $\mathbf{X}_k^b$ is $N \times M$.
- The Kalman gain becomes:

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}} \left[\mathbf{H}_k \mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right]^{-1} \\ &= \mathbf{X}_k^b \left(\mathbf{H}_k \mathbf{X}_k^b\right)^{\mathrm{T}} \left[\left(\mathbf{H}_k \mathbf{X}_k^b\right)\left(\mathbf{H}_k \mathbf{X}_k^b\right)^{\mathrm{T}} + \mathbf{R}_k\right]^{-1}\end{aligned}$$

- Note that, to evaluate $\mathbf{K}$, we apply $\mathbf{H}_k$ to the M columns of $\mathbf{X}_k^b$, rather than to the N columns of $\mathbf{P}_k^b$
- Note also that the analysis increment, $\mathbf{x}_k^a - \mathbf{x}_k^b = \mathbf{K}_k \left(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k^b)\right)$, is a linear combination of the columns of $\mathbf{X}_k^b$.

Kalman Filters for Large Dimensional Systems

$$\mathbf{K}_k = \mathbf{X}_k^b \left(\mathbf{H}_k \mathbf{X}_k^b\right)^{\mathrm{T}} \left[\left(\mathbf{H}_k \mathbf{X}_k^b\right) \left(\mathbf{H}_k \mathbf{X}_k^b\right)^{\mathrm{T}} + \mathbf{R}_k\right]^{-1}$$

- The analysis error covariance matrix is:

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^b (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^{\mathrm{T}}$$

- Since $\mathbf{P}_k^b = \mathbf{X}_k^b \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$, we see that all terms in the expression for $\mathbf{P}_k^a$ contain an initial $\mathbf{X}_k^b$ and a final $\left(\mathbf{X}_k^b\right)^{\mathrm{T}}$.
- Hence

$$\mathbf{P}_k^a = \mathbf{X}_k^b \, \mathbf{W}_k \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$$

for some $M \times M$ matrix $\mathbf{W}_k$.

Kalman Filters for Large Dimensional Systems

$$\mathbf{P}_k^a = \mathbf{X}_k^b \, \mathbf{W}_k \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$$

- The covariance matrix is propagated using:

$$\begin{aligned}\mathbf{P}_{k+1}^b &= \mathbf{M}_{t_k \to t_{k+1}} \mathbf{P}_{k-1}^a \mathbf{M}_{t_k \to t_{k+1}}^{\mathrm{T}} + \mathbf{Q}_{k+1} \\ &= \left(\mathbf{M}_{t_k \to t_{k+1}} \mathbf{X}_k^b\right) \mathbf{W}_k \left(\mathbf{M}_{t_k \to t_{k+1}} \mathbf{X}_k^b\right)^{\mathrm{T}} + \mathbf{Q}_{k+1}\end{aligned}$$

- Note that this requires only M integrations of the tangent linear model.
- The addition of $\mathbf{Q}_{k+1}$ means that, in general, $\mathbf{P}_{k+1}^b$ is not of low rank.
- However, we can approximate it by projecting onto some suitable M-dimensional subspace. The resulting algorithm is called a reduced-rank Kalman filter.

Kalman Filters for Large Dimensional Systems

- The severe reduction in rank causes significant problems for the reduced-rank kalman filter:
- The analysis increment is restricted to an M-dimensional subspace.
  - There are too few degrees of freedom available to fit the $\sim 10^6$ observations.
- The low-rank approximations of the covariance matrices suffer from spurious long-distance correlations. These cause two problems:
  - The analysis may generate spurious increments in regions where there are no observations.
  - The analysis may be unable to draw to isolated observations (e.g. over Antarctica) if it thinks there is a significant correlation with a well-observed region (e.g. Europe).
- There are two ways around these problems:
  - Local analysis (e.g. Evensen 2003, Ocean Dynamics 343–367; Ott *et al.* 2004, Tellus 415–428).
  - Shur product modification of the covariances (e.g. Houtekamer and Mitchell 2001, MWR 123–137).

Kalman Filters for Large Dimensional Systems

- Local analysis solves the analysis equations independently for each gridpoint, or for each of a set of regions covering the domain.
- Each analysis uses only observations that are local to the gridpoint (or region).
- This guarantees that the analysis at each gridpoint (or region) is not influenced by distant observations.
- The global analysis is constructed by stitching together the independent regional (or gridpoint) analyses, and is thus no longer a linear combination of the columns of $\mathbf{X}_k^b$.
- In effect, the method acts to vastly increase the dimension of the sub-space in which the analysis increment is constructed.
- However, performing independent analyses for each region is not optimal, and the method shares some of the problems of OI (e.g. poor analysis of the large scales, and difficulties in producing balanced analyses).

Kalman Filters for Large Dimensional Systems

- The Schur product approach uses the fact that if $\mathbf{B}$ and $\mathbf{C}$ are covariance matrices, then so is $\mathbf{A} = \mathbf{B} \circ \mathbf{C}$, where $\circ$ denotes the Schur (i.e. element-wise) product: $A_{ij} = B_{ij} C_{ij}$.
- Spurious long-range correlations in $\mathbf{P}_k^b$ may be suppressed by forming the Schur product with a covariance matrix representing a decaying function of distance.
  - The modified covariance matrix is never formed explicitly. Rather, the method deals with terms such as $\mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}}$.
- The modified covariance matrix is no longer of the form $\mathbf{X}_k^b \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$.
- Forming the Schur product has the effect of vastly increasing the rank of the matrix.
- Choosing the product function is non-trivial. It is easy to modify $\mathbf{P}_k^b$ in undesirable ways. In particular, balance relationships may be adversely affected.

Ensemble Methods

- Ensemble Kalman filters are reduced-rank Kalman filters that construct their covariance matrices as sample covariance matrices:

$$\mathbf{P}_k^b = \frac{1}{M-1}\sum_{m=1}^{M-1}(\mathbf{x}_{k,m}^b - \overline{\mathbf{x}_{k,m}^b})(\mathbf{x}_{k,m}^b - \overline{\mathbf{x}_{k,m}^b})^{\mathrm{T}}$$

where the subscript m refers to the sample (ensemble member).

- Note that we can write this as $\mathbf{P}_k^b = \mathbf{X}_k^b\left(\mathbf{X}_k^b\right)^{\mathrm{T}}$, where

$$\mathbf{X}_k^b = \frac{1}{\sqrt{M-1}}\left(\ (\mathbf{x}_{k,1}^b - \overline{\mathbf{x}_{k,1}^b}),\ \ (\mathbf{x}_{k,2}^b - \overline{\mathbf{x}_{k,2}^b}),\ \ \cdots,\ \ (\mathbf{x}_{k,M}^b - \overline{\mathbf{x}_{k,M}^b})\ \right)$$

Ensemble Methods

- The Extended Kalman filter includes terms involving $\mathbf{M}_k$, $\mathbf{M}_k^{\mathrm{T}}$, $\mathbf{H}_k$ and $\mathbf{H}_k^{\mathrm{T}}$.
  - I.e. it uses the tangent linear and adjoint model and observation operators.
- In the ensemble Kalman filter, we avoid the need for tangent linear and adjoint operators by approximating:

$$
\begin{aligned}
\mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}} &\approx \frac{1}{M-1} \sum_{m=1}^{M} \left(\mathbf{x}_{k,m}^b - \overline{\mathbf{x}_{k,m}^b} \right) \left(\mathcal{H}(\mathbf{x}_{k,m}^b) - \overline{\mathcal{H}(\mathbf{x}_{k,m}^b)} \right)^{\mathrm{T}} \\
\mathbf{H}_k \mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}} &\approx \\
& \frac{1}{M-1} \sum_{m=1}^{M} \left(\mathcal{H}(\mathbf{x}_{k,m}^b) - \overline{\mathcal{H}(\mathbf{x}_{k,m}^b)} \right) \left(\mathcal{H}(\mathbf{x}_{k,m}^b) - \overline{\mathcal{H}(\mathbf{x}_{k,m}^b)} \right)^{\mathrm{T}}
\end{aligned}
$$

- Not having to code tangent linear an adjoint operators is one of the main attractions of the ensemble Kalman filter!

Ensemble Methods

- The Ensemble Kalman filter requires us to generate a sample $\{\mathbf{x}^b_{k,m}; m = 1 \cdots M\}$ drawn from the p.d.f. of background error.
- We can generate this from a sample $\{\mathbf{x}^a_{k-1,m}; m = 1 \cdots M\}$ drawn from the p.d.f. of analysis error for the previous cycle:

$$\mathbf{x}^b_{k,m} = \mathcal{M}_{t_{k-1} \rightarrow t_k} \left(\mathbf{x}^a_{k-1,m} \right) + \eta_{k,m}$$

  where $\eta_{k,m}$ is a sample drawn from the p.d.f. of model error.
- We now consider how to generate a sample drawn from the analysis p.d.f.

Ensemble Methods

- If we perturb all the inputs to the analysis with perturbations drawn from the p.d.f's of the errors in the inputs, then the output will be perturbed with a perturbation drawn from the p.d.f. of the output.
- To convince yourself that this is the case, consider an analysis fed with the *true* background state and observations (and any other inputs).
- We constructed the analysis so that true inputs produce a true output, so the output will be the true state.
- Now, we can consider the normal inputs to the analysis as perturbations of the true state and observations:

$$\begin{aligned} \mathbf{x}_k^b &= \mathbf{x}_k^* + \epsilon_k^b \\ \mathbf{y}_k &= \mathbf{y}_k^* + \epsilon_k^o \end{aligned}$$

- Clearly, the input "perturbations" (ϵ_k^b and ϵ_k^o) are drawn from the p.d.f's of background and observation error.
- The output from the analysis is $\mathbf{x}_k^a = \mathbf{x}_k^* + \epsilon_k^a$, where ϵ_k^a is a sample from the p.d.f. of analysis error.

Ensemble Methods

- Hence, one way to generate a sample drawn from the p.d.f. of analysis error is to perturb the observations with perturbations characteristic of observation error.
  - The perturbed backgrounds are already available because they were generated by the preceding analysis cycle.
- The method is simple, and treats the analysis as a "black box". In particular, there is no requirement for the analysis to be linear.
- However, the addition of noise to the observations results in a small amount of additional sampling noise.
- This additional noise can be avoided by constructing the analysis sample as a linear combination of the background sample:

$$\mathbf{X}_k^a = \mathbf{X}_k^b \mathbf{T}$$

  where $\mathbf{T}$ is an $M \times M$ matrix constructed using linear algebra to make the columns of $\mathbf{X}_k^a$ satisfy $\mathbf{X}_k^a (\mathbf{X}_k^a)^{\mathrm{T}} = \mathbf{P}_k^a$.
- This approach is used in the Ensemble Transform Kalman Filter (ETKF) (Bishop *et al.* 2001, MWR 420-436).

Non-Gaussian Methods

- Kalman filters, as well as 3D-Var, 4D-Var and OI, are essentially Gaussian methods. They assume that the p.d.f. of error is fully described by the mean and covariance.
- Non-Gaussian methods do not make this assumption.
- Particle filters are a class of non-Gaussian method that approximate the p.d.f. by a discrete distribution:

$$p(\mathbf{x}) = \begin{cases} w_m & \text{if } \mathbf{x} = \mathbf{x}_{k,m} \\ 0 & \text{otherwise} \end{cases}$$

- An ensemble of forecasts $\{\mathbf{x}_m; m = 1 \cdots M\}$ is run, and each member is given an associated weight, w_m, according to its probability.
- When an observation, y, is available, the weights are adjusted using Bayes' theorem:

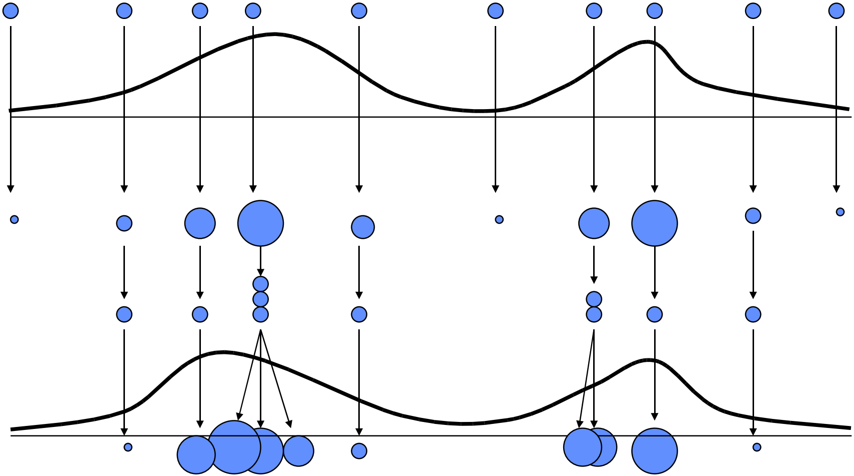
$$w_m^{\mathrm{new}} = \frac{w_m^{\mathrm{old}} p(y|\mathbf{x}_m)}{\sum_{m=1}^{M} w_m^{\mathrm{old}} p(y|\mathbf{x}_m)}$$

Non-Gaussian Methods

- In its most basic form, this is all there is to a particle filter. States that agree with the observations get large weights, whereas states that disagree with the observations get small weights.
- In practice, the weights for some members become tiny. These members are no longer useful and they are dropped from the ensemble and replaced by new, more probable members.
- This is achieved by periodically resampling the p.d.f.
- A new ensemble is generated by randomly picking the old members with a probability equal to their weights. Members with large weight may be picked several times, whereas members with very small weight are unlikely to be picked.
- After resampling, all the weights are reset to $1/M$.
- Resampling may produce some identical members. However these diverge slowly from each other because each member is forced with different random perturbations that represent model error.

Non-Gaussian Methods

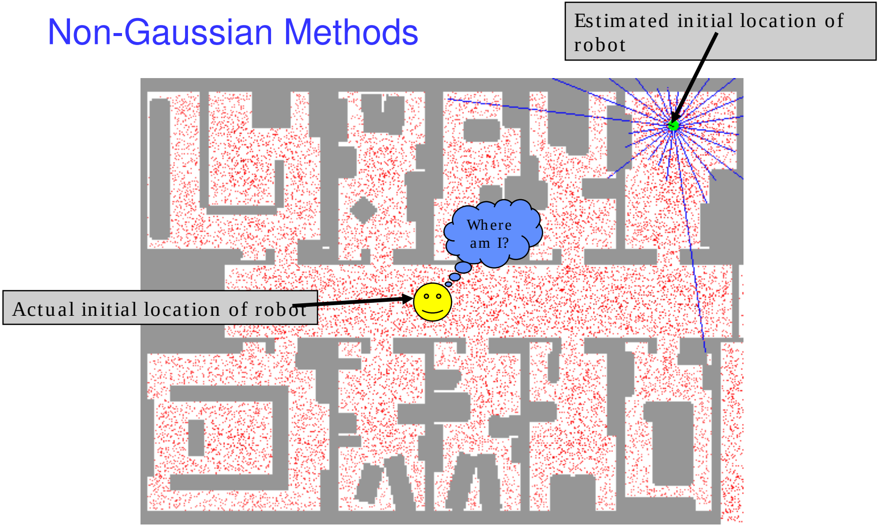
Non-Gaussian Methods



Non-Gaussian Methods

- Particle filters work well for highly nonlinear, very low-dimensional systems, such as missile tracking or computer vision.
- van Leeuwen (2003) has applied the technique to a simple ocean model.
- The main problem to be overcome is that, for a large dimensional system, it does not make much sense to characterise the state of the model with a single weight — the state may be close to the observations in one region, but far from observations elsewhere.
- To be successful in large systems, particle filters will need to adopt some form of localisation that allows different weights in different parts of the model domain.

Non-Gaussian Methods



from: Fox et al. 1999, proc 16th National Conference on Artificial

The Non-Sequential Approach

- Kalman filters are based on a sequential view of the analysis problem.
- An optimal estimate for step $k + 1$ is produced using only information propagated from step k, and the new observations.
- *All the information from earlier steps is brought to the present step via the background state and covariance matrix.*
- The advantage of the sequential approach is that we do not need to go back any further than the previous step in order to determine the current analysis.
- The disadvantage is that we must explicitly propagate the covariances. This can only be done approximately.
- The non-sequential approach provides an alternative.

The Non-Sequential Approach

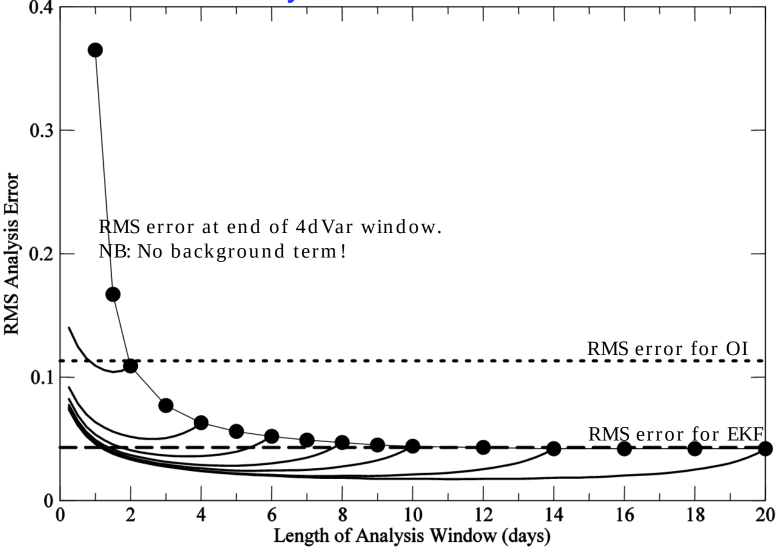
- It can be shown (eg. Daley 1996, Tellus 48A 221–237) that, for a linear system the state at the end of the analysis window in weak-constraint 4D-Var is identical to that produced by the Kalman filter, given the same states and covariances.
  - In particular, given the same background error covariance matrix at the start of the window.
- However, in typical applications, the 4D-Var window is short, and the initial covariance matrix has a strong influence on the analysis at the end of the window.
  - Effectively, 4D-Var behaves like a Kalman filter in which the covariance matrix is reset to some static matrix **B** every few hours. This is sub-optimal.
- If we run 4D-Var with a long enough window, the state and covariance at the start of the window will have little influence on the analysis at the end of the window.
  - Information from last week does not tell us much about today's weather.

The Non-Sequential Approach

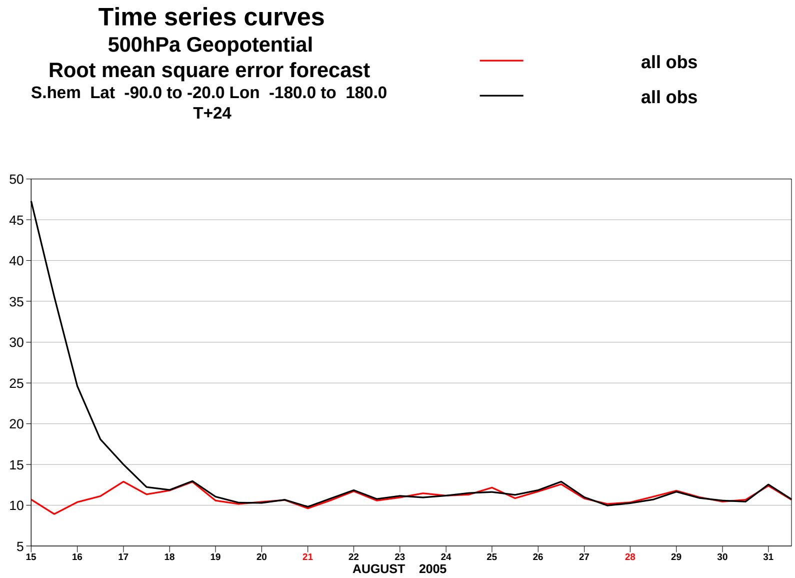
- If the start of the 4D-Var window is pushed far enough back in time that information at the start of the window no longer influences the analysis at the end of the window, then extending the window even further back in time will not change the final analysis.
- Therefore, a 4D-Var analysis with a "long enough" window will be the same as we would get with a semi-infinite window.
- Since 4D-Var and the Kalman filter give the same analysis (given the same states, observations and covariances) we expect long-window 4D-Var to be equivalent to a long-running Kalman filter.

The Non-Sequential Approach

Mean RMS Analysis Error

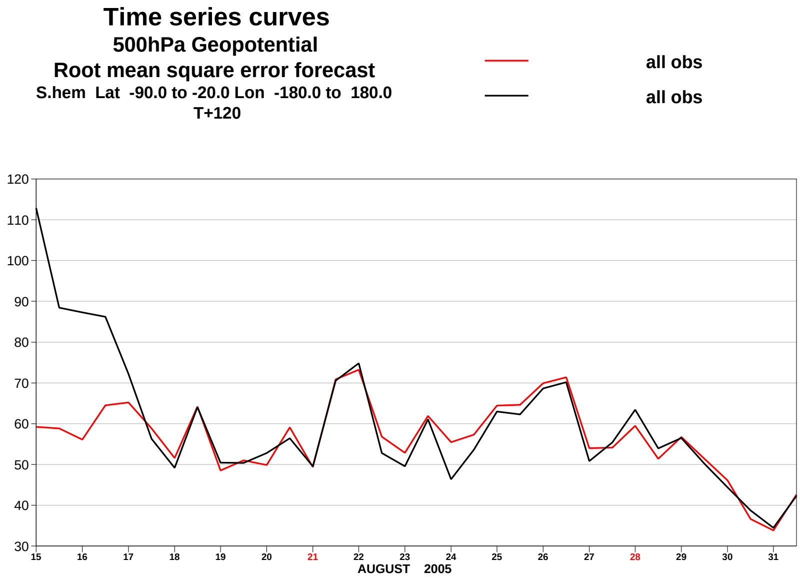


The Non-Sequential Approach



ECMWF

The Non-Sequential Approach



Summary

- The Kalman filter produces an analysis that is optimal with respect to all current and past observations.
- The full Kalman filter is impractical for large systems.
- Ensemble methods provide one method for approximating the Kalman filter.
- Current Ensemble Kalman filters produce analyses similar in quality to 4D-Var.
- Particle filters are fun, but it is not clear how useful they are for large systems.
- Long-window weak-constraint 4D-Var provides an interesting alternative to explicit Kalman filter methods.
- Long-window weak-constraint 4D-Var, with an analysis window of ~ 3 days, may be capable of producing analysis equivalent to those of a full, unapproximated Kalman filter.