University of Reading

Particle filters in practice

Polly Smith

Data Assimilation Research Centre

p.j.smith@reading.ac.uk

Data Assimilation Research Centre

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

Introduction

- problems we are dealing with in practice are large and non-linear and expect to become even more so ...
  - e.g. high-resolution NWP is highly non-linear
- standard PF formulation can never work for large dimensional systems
- until fairly recently PF's have been considered computationally unfeasible for large dimensional systems because of the degeneracy problem (curse of dimensionality)

University of Reading

Big data

- How big is the non-linear data assimilation problem?
- Assume we need 10 frequency bins for each variable to build the joint pdf of all variables.
- Let's assume we have a modest model with a million variables.
- Then we need to store $10^{1,000,000}$ numbers.
- The total number of atoms in the universe is estimated to be about $10^{80.}$
- So the data-assimilation problem is larger than the universe…

University of Reading

Curse of dimensionality

- The argument for the inefficiency of PFs in high dimensional systems is that the number of required particles increases exponentially with the effective dimension of the system (Snyder et al, 2008)
- it is not the dimension of the state space that is the problem, but the dimension of the observation space N_y
  - the higher N_y the more peaked the likelihood is
  - this leads to particle weights varying enormously, with a few particles having much higher weight than all the others
  - resampling produces copies of the few particles with the highest weights
  - all variation in the particles is lost
- therefore to apply a particle filter to a high-dimensional system additional information is needed to limit the search space of the filter.

A closer look at the weights I

Assume

- particle 1 is at 0.1σ of the N_y independent observations
- particle 2 is at 0.2σ of the N_y independent observations

σ is the standard deviation of the observation errors

These are two almost perfect particles, what more can you want?

The weight of particle 1 will be

$$w_1 \propto \exp\left[-\frac{1}{2}\left(\mathbf{y}-\mathbf{H}\left(\mathbf{x}_1\right)\right)^T \mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{H}\left(\mathbf{x}_1\right)\right)\right] = \exp\left(-0.005N_y\right)$$

The weight of particle 2 will be

$$w_2 \propto \exp\left[-\frac{1}{2}\left(\mathbf{y}-\mathbf{H}\left(\mathbf{x}_2\right)\right)^T \mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{H}\left(\mathbf{x}_2\right)\right)\right] = \exp\left(-0.02N_y\right)$$

A closer look at the weights II

The ratio of the weights is

$$\frac{w_2}{w_1} = \exp\left(-0.015 N_y\right)$$

Take e.g. N_y= 1000 (this is modest) to find

$$\frac{w_2}{w_1} = \exp\left(-15\right) \approx 3 \times 10^{-7}$$

particle 2 has a negligible weight compared to particle 1 despite them both being excellent particles.

Conclusion - the number of independent observations is responsible for the degeneracy in particle filters.

Possible solutions

New efficient PF variants have been developed that have been shown to work for large dimensional systems with a limited number of particles

Options:

- introduce localization to reduce the number of observations
- use proposal-density freedom to exploit the future observational information
- Approximations to the full PF
  - combine PF and EnKF or Gaussian Mixtures or second-order exact filters
- Transportation particle filters

University of Reading

Proposal density particle filters

- Aim is to ensure that equally significant particles are picked from the posterior density.
- To do this we have to
  1. ensure that all particles end up in the high-probability area of the posterior pdf,
     - use a scheme that pulls the particles towards the observations
  2. ensure that the weights of the different particles are very similar, or even equal, before any resampling step
- Here we discuss:
  - Equivalent-Weights Particle Filter (EWPF)
  - Implicit Equal-Weights Particle Filter (IEWPF)

University of Reading

Exploring proposal density freedom

We start by writing the prior at time n as

$$p\left(\mathbf{x}^n\right) = \int p\left(\mathbf{x}^n|\mathbf{x}^{n-1}\right) p\left(\mathbf{x}^{n-1}\right)\, dx^{n-1} \tag{1}$$

assuming we have a set of weighted particles at time n-1

$$p\left(\mathbf{x}^{n-1}\right) = \sum_{i=1}^{N_e} w_i^{n-1} \delta\left(\mathbf{x}^{n-1} - \mathbf{x}_i^{n-1}\right) \tag{2}$$

and using (2) in (1) we can write the prior at time n as

$$p\left(\mathbf{x}^n\right) = \sum_{i=1}^{N_e} w_i^{n-1} p\left(\mathbf{x}^n|\mathbf{x}_i^{n-1}\right)$$

University of Reading

- The idea of proposal densities is to sample from a proposed pdf $q(\mathbf{x})$ rather than the original model pdf $p(\mathbf{x}^{(n)}|\mathbf{x}^{(n-1)})$
- if we multiply the numerator and denominator of the equation for the prior by the proposal transition density $q(\mathbf{x}^{(n)} \mid \mathbf{x}^{(n-1)}, \mathbf{y}^{(n)})$ we get

$$p\left(\mathbf{x}^n\right) = \sum_{i=1}^{N_e} w_i^{n-1} \frac{p\left(\mathbf{x}^n|\mathbf{x}_i^{n-1}\right)}{q\left(\mathbf{x}^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n\right)} q\left(\mathbf{x}^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n\right)$$

note that the support of **q** should be greater than or equal to $p(\mathbf{x}^{(n)}|\mathbf{x}^{(n-1)})$

drawing $\mathbf{x}_i^n$ from $\mathbf{q}$ leads to a weighted set of particles at time n

$$p\left(\mathbf{x}^n\right) = \sum_{i=1}^{N_e} \hat{w}_i^{n-1} \delta\left(\mathbf{x}^n - \mathbf{x}_i^n\right)$$

where $\hat{w}_i^{n-1} = w_i^{n-1} \dfrac{p\left(\mathbf{x}_i^n | \mathbf{x}_i^{n-1}\right)}{q\left(\mathbf{x}_i^n | \mathbf{x}_i^{n-1}, \mathbf{y}^n\right)}$

For the posterior pdf we now have

$$\begin{aligned} p\left(\mathbf{x}^n | \mathbf{y}^n\right) &= \sum_{i=1}^{N_e} w_i^{n-1} \frac{p\left(\mathbf{y}^n | \mathbf{x}_i^n\right)}{p\left(\mathbf{y}^n\right)} \frac{p\left(\mathbf{x}^n | \mathbf{x}_i^{n-1}\right)}{q\left(\mathbf{x}^n | \mathbf{x}_i^{n-1}, \mathbf{y}^n\right)} q\left(\mathbf{x}^n | \mathbf{x}_i^{n-1}, \mathbf{y}^n\right) \\ &= \sum_{i=1}^{N_e} w_i^n \delta\left(\mathbf{x}^n - \mathbf{x}_i^n\right) \end{aligned}$$

University of Reading

with weights

$$w_i^n = w_i^{n-1} \frac{p\left(\mathbf{y}^n|\mathbf{x}_i^n\right)}{p\left(\mathbf{y}^n\right)} \frac{p\left(\mathbf{x}_i^n|\mathbf{x}_i^{n-1}\right)}{q\left(\mathbf{x}_i^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n\right)}$$

$$= \frac{p\left(\mathbf{y}^n|\mathbf{x}_i^{n-1}\right)}{p\left(\mathbf{y}^n\right)} \frac{p\left(\mathbf{x}_i^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n\right)}{q\left(\mathbf{x}_i^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n\right)}$$

Note that

1. the proposal depends on the future observation, and
2. the proposal can depend on all previous particles, not just one.

Allows us to change the trajectories of the particles, and so guide them closer to the observations

University of Reading

Optimal proposal density

- we can choose any proposal transition density for **q**; the aim is to use one that includes additional information from the observations in the future, in an optimal way.
- the optimal proposal density (Doucet et al., 2000) chooses

$$q\left(\mathbf{x}^n|\mathbf{x}^{n-1},\,\mathbf{y}^n\right) = p\left(\mathbf{x}^n|\mathbf{x}^{n-1},\,\mathbf{y}^n\right)$$

so the weights simplify to

$$w_i \propto p\left(\mathbf{y}^n|\mathbf{x}_i^{n-1}\right)$$

which is the maximum weight each particle can achieve

- but these will be degenerate for systems with a large number of independent observations.

The equivalent-weights particle filter (Ades and van Leeuwen)

University of Reading

The EWPF works as follows:

1. Determine the OPD weights for each particle $w_i^{max} = p(\mathbf{y}^n \mid \mathbf{x}_i^{n-1})$
2. Choose a target weight, w_{target}
   - this is chosen such that a certain percentage of particles can reach it
   - those particles that cannot are abandoned & resampled in step 5.
3. Find the position in state space of each particle such that it has weight exactly equal to w_{target} , we denote this position $\mathbf{x}_i^*$
4. The move in step 3 is purely deterministic so we add a small random perturbation to each particle and then recalculate its weight.
5. Calculate full posterior weights for new particles and resample all particles such that their weights are equal again.

University of Reading

Choosing the target weight

In the SIR filter, the greater the weight of a particle, the higher its significance in estimating the posterior.

- this would suggest that setting $w_{target} = \max(w_i^{max})$ would lead to the most information being gained on the posterior pdf, but ...
  - no particle is able to achieve a weight greater than it's maximum
  - all but one particle would be lost
- if we choose $w_{target} = \min(w_i^{max})$ then we keep 100% of the particles
- if we choose $w_{target} = \text{median}(w_i^{max})$ then we keep 50% of the particles
- if we choose to keep less than 100% of the particles then there will be certain particles that cannot reach the target weight no matter how we move them
  - these will be replaced via resampling in step 5.

EWPF - step 3 details

- For the retained particles there are infinitely many ways to move a particle in state space such that it reaches w_{target}
- In the EWPF the solution $\mathbf{x}_i^*$ is assumed to be given by

$$\mathbf{x}_i^* = f\left(\mathbf{x}_i^{n-1}\right) + \alpha_i \mathbf{K}\left(\mathbf{y}^n - \mathbf{H} f(\mathbf{x}_i^{n-1})\right)$$

where $\mathbf{K} = \mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}$ and α_i is a scalar given by

$$\alpha_i = 1 \pm \sqrt{1 - b_i/a_i}$$

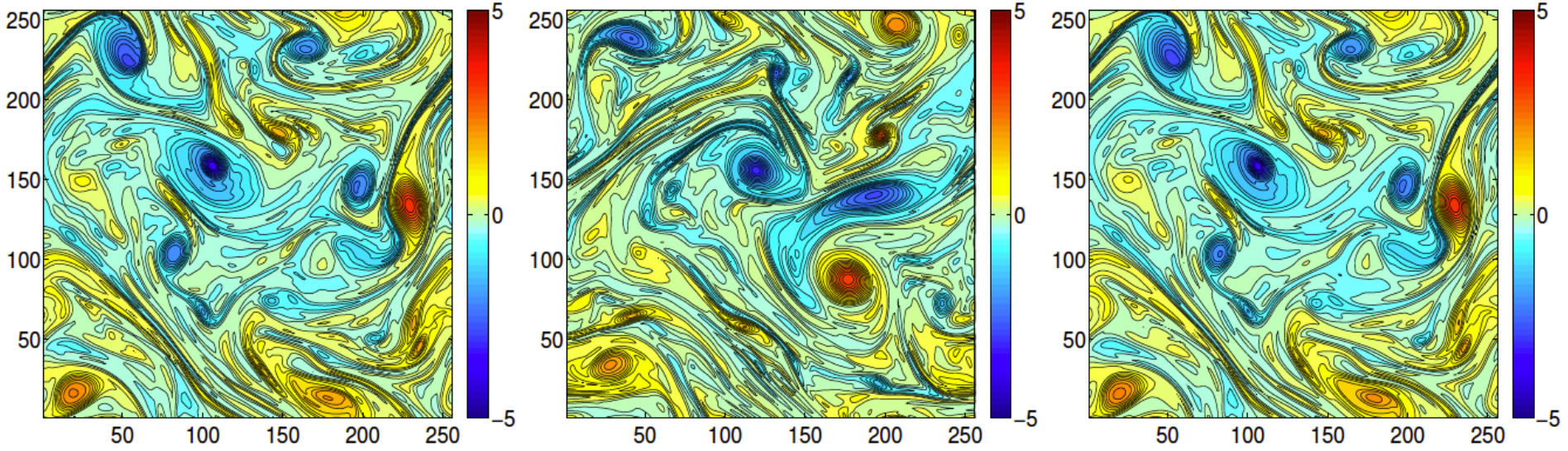
with

$$\begin{aligned} a_i &= 0.5\left(\mathbf{y}^n - \mathbf{H} f(\mathbf{x}_i^{n-1})\right)^T \mathbf{R}^{-1}\mathbf{H}\mathbf{K}\left(\mathbf{y}^n - \mathbf{H} f(\mathbf{x}_i^{n-1})\right) \\ b_i &= 0.5\left(\mathbf{y}^n - \mathbf{H} f(\mathbf{x}_i^{n-1})\right)^T \mathbf{R}^{-1}\left(\mathbf{y}^n - \mathbf{H} f(\mathbf{x}_i^{n-1})\right) + \log w_{target} - \log w_i^{n-1} \end{aligned}$$

University of Reading

Barotropic vorticity model

Particle analysis mean vs truth

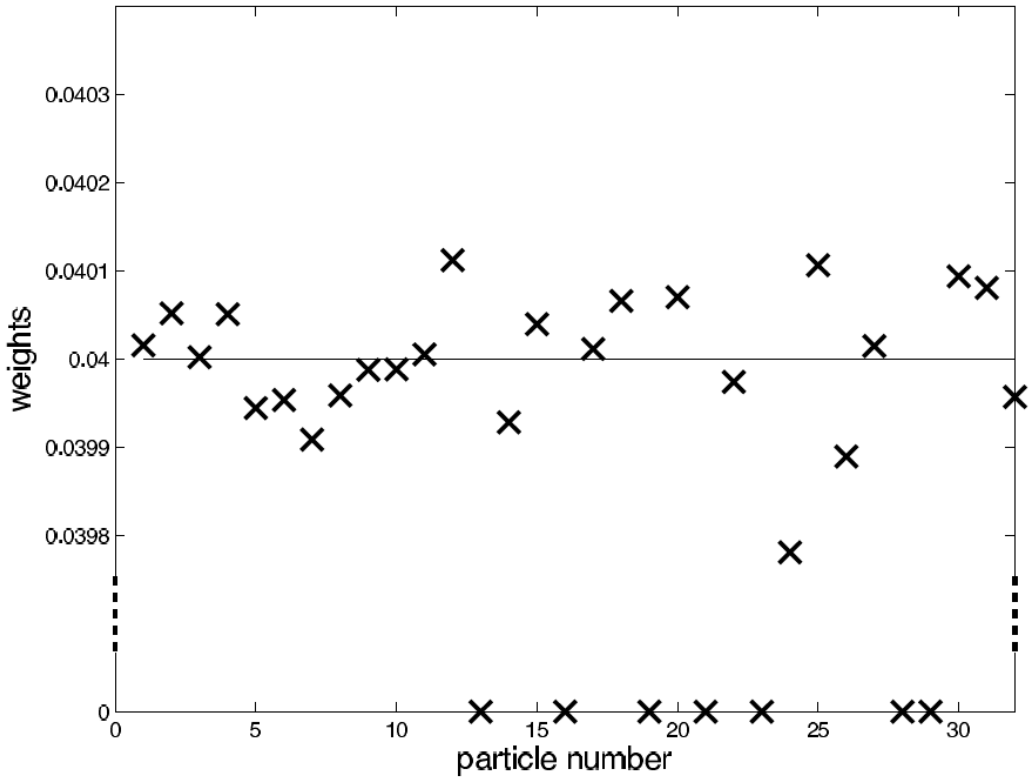


(a) truth (b) SIR particle filter (c) equivalent-weights particle filter

- SIR vs. EWPF (with 80% particles retained)
- 256x256 grid = 65,536 state dimension
- 1150 time steps, observations every 50 time-steps
- 32 particles

EWPF weights

Values of normalised weights for each particle before re-sampling (time step 50)



25 of the 32 particles (80%) have almost equal weights

Comments ...

- The EWPF ensures that the majority of the particles receive equivalent-weights, while they keep close to the observations.
- Prevents filter degeneracy, regardless of the system dimension.
- Fully nonlinear scheme, which represents the full posterior pdf.
- Does not assume that posterior is a Gaussian distribution.

University of Reading

More comments ...

- The EWPF is biased and does not converge to the posterior pdf for large N_e because of the equivalent-weights construction,
  - high weight particles are moved such that their weight becomes lower, equal to the target weight.
- In practice, the large N_e limit is not that relevant as the affordable number of particles will be low and so the Monte-Carlo error will be substantial
- As long as the Monte Carlo error is larger than the bias the scheme is a valid option for high-dimensional systems

University of Reading

Implicit equal weights particle filter (Zhu et al)

- Steps of the IEWPF are very similar to the EWPF
- Main difference is in the way that we seek to reach the target weight
  - in the EWPF we scale the deterministic part of the optimal proposal
  - in the IEWPF we scale the random part of the optimal proposal
- Samples drawn implicitly from a standard Gaussian distributed proposal density q($\boldsymbol{\xi}$) instead of the original one q($\mathbf{x}^n|\mathbf{x}^{n-1}, \mathbf{y}^n$)
- these two are related by

$$q\left(\mathbf{x}^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n\right) = \frac{q\left(\boldsymbol{\xi}_i\right)}{||\frac{d\mathbf{x}}{d\boldsymbol{\xi}}_i||}$$

Single-stage IEWPF

At each observation time, the updated state of each particle i is computed as

$$\mathbf{x}_i^n = \mathbf{x}_i^* + \alpha_i \mathbf{P}^{1/2} \boldsymbol{\xi}_i$$

where $\mathbf{X}_i^*$ is the mode of the OPD $\mathbf{x}_i^* \sim p\left(\mathbf{x}^n | \mathbf{x}_i^{n-1}, \mathbf{y}^n\right)$

$$\mathbf{x}_i^* = f\left(\mathbf{x}_i^{n-1}\right) + \mathbf{Q}\mathbf{H}^T \left(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R}\right) \left(\mathbf{y}^n - \mathbf{H} f(\mathbf{x}_i^{n-1})\right)$$

$$\mathbf{P} = (\mathbf{Q}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1}$$

The scalars α_i are used to ensure that weight of each particle i is equal to a target weight.

Equal weights

To have equal weights, we need to find α_i so that for each particle *i*

$$w_i\left(\alpha_i\right) = \frac{p\left(\mathbf{y}^n|\mathbf{x}_i^{n-1}\right) p\left(\mathbf{x}_i^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n\right)}{q\left(\mathbf{x}_i^n|\mathbf{x}_i^{n-1}, \mathbf{y}^n\right)} = w_{target}$$

... but this is messy ...

Note: typically the target weight is chosen as $w_{target} = \min(w_i^{max})$ so that all particles are kept, but other choices possible.

Solving for α_i

For moderately sized system the α_i can be determined numerically by solving

$$\gamma\left(\frac{N_x}{2}, \frac{\alpha_i \boldsymbol{\xi}_i^{n,T} \boldsymbol{\xi}_i^n}{2}\right) = e^{-c_i/2}\gamma\left(\frac{N_x}{2}, \frac{\boldsymbol{\xi}_i^{n,T} \boldsymbol{\xi}_i^n}{2}\right)$$

where $\gamma(s,x) = \int_0^x t^{s-1}e^{-t}\mathrm{d}t$ is the lower incomplete gamma function

Otherwise can derive approx solution in the limit $N_x \longrightarrow \infty$

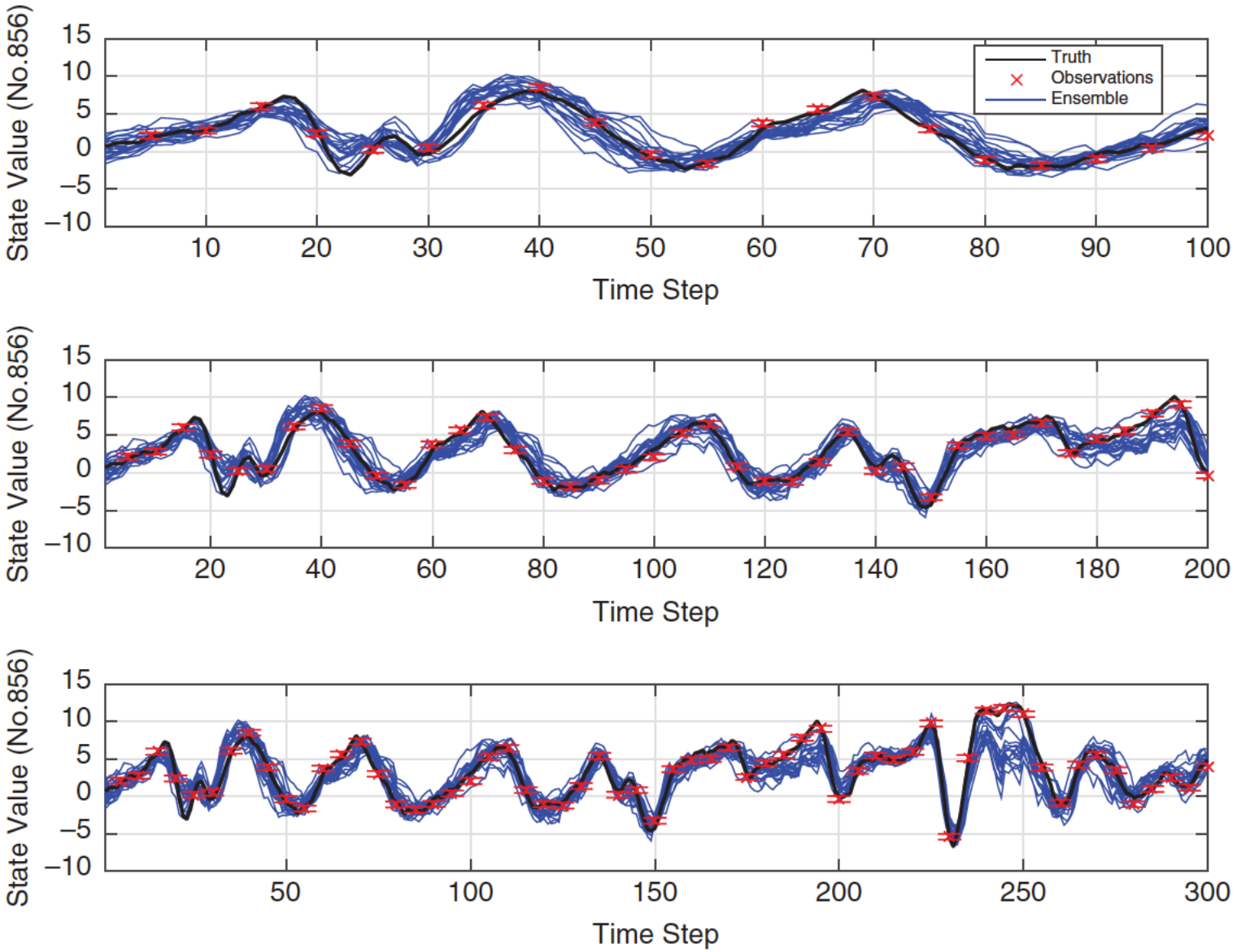
University of Reading

Lorenz 96 model experiments

- 1000 dimensional
- 20 particles
- $\Delta t = 0.05 \sim 6$ hours
- observations every 5 time steps, half of state

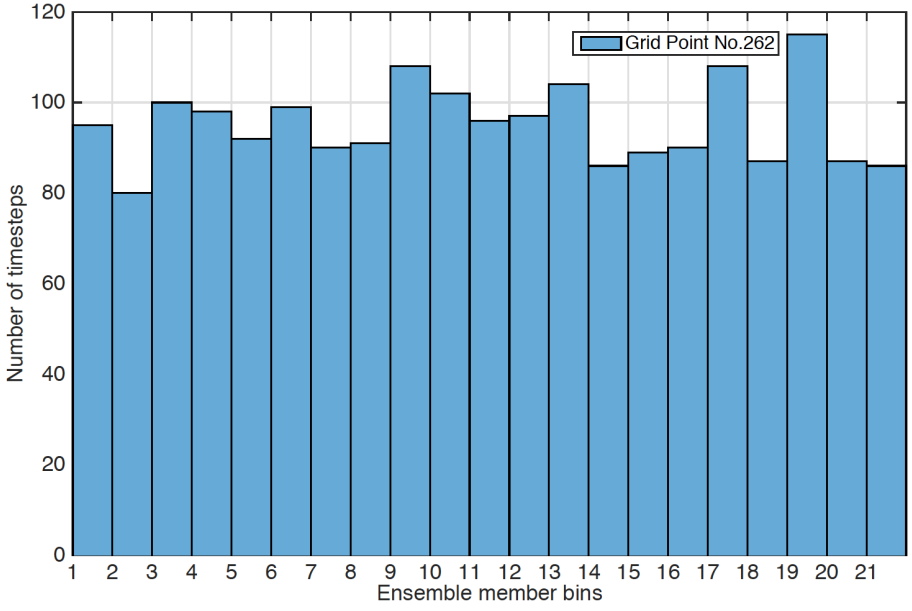
Particle trajectories

University of Reading

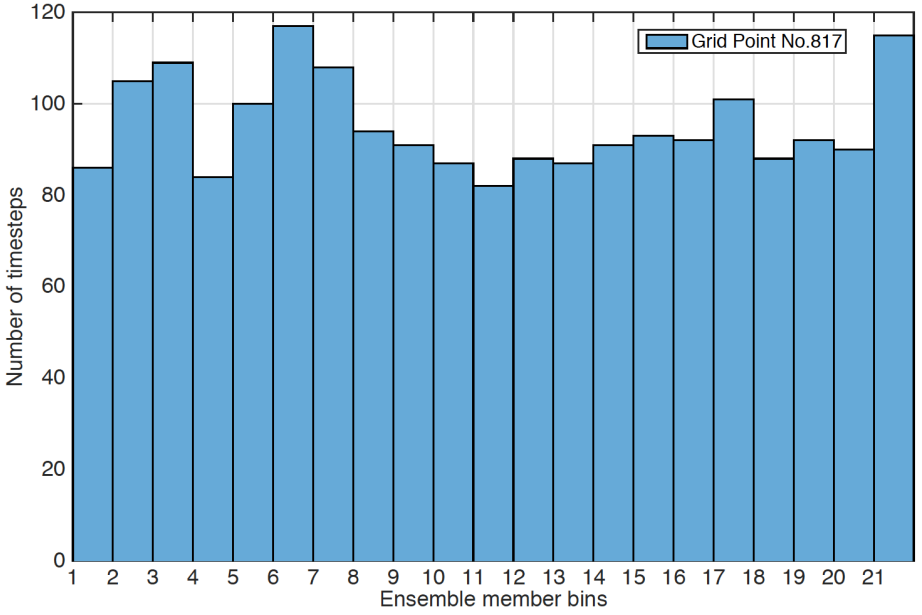


University of Reading

Rank histograms: 10,000 time steps land-sea configuration



observed gridpoint



unobserved gridpoint

Comments

- By construction the 1-stage IEWPF does not converge to the correct posterior pdf
  - not an issue as long as bias is smaller than the statistical noise in the method
- This version of the IEWPF scheme is only valid for high dimensional systems with low particle number
- **Skauvold et al, 2019** show that bias is systematic, and leads to underestimation of the filter variance
- New 2-stage version(s) currently being explored

Two-stage IEWPF (Skauvold, et al 2019)

At each observation time, the updated state of each particle is computed as

$$\mathbf{x}_i^n = \mathbf{x}_i^* + \beta \mathbf{P}^{1/2} \boldsymbol{\eta}_i + \alpha_i \mathbf{P}^{1/2} \boldsymbol{\xi}_i$$

where $\mathbf{x}_i^*$ is mode of OPD as before, and

$$\boldsymbol{\eta}_i, \boldsymbol{\xi}_i \sim N(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\eta}_i^T \boldsymbol{\xi}_i = 0$$

- scalars α_i are used to ensure that weight of each particle is equal to the target weight ($-1 \leq \alpha_i \leq 1$)
- fixed scalar β is used to control the spread of the updated particles, without compromising particle weight equality

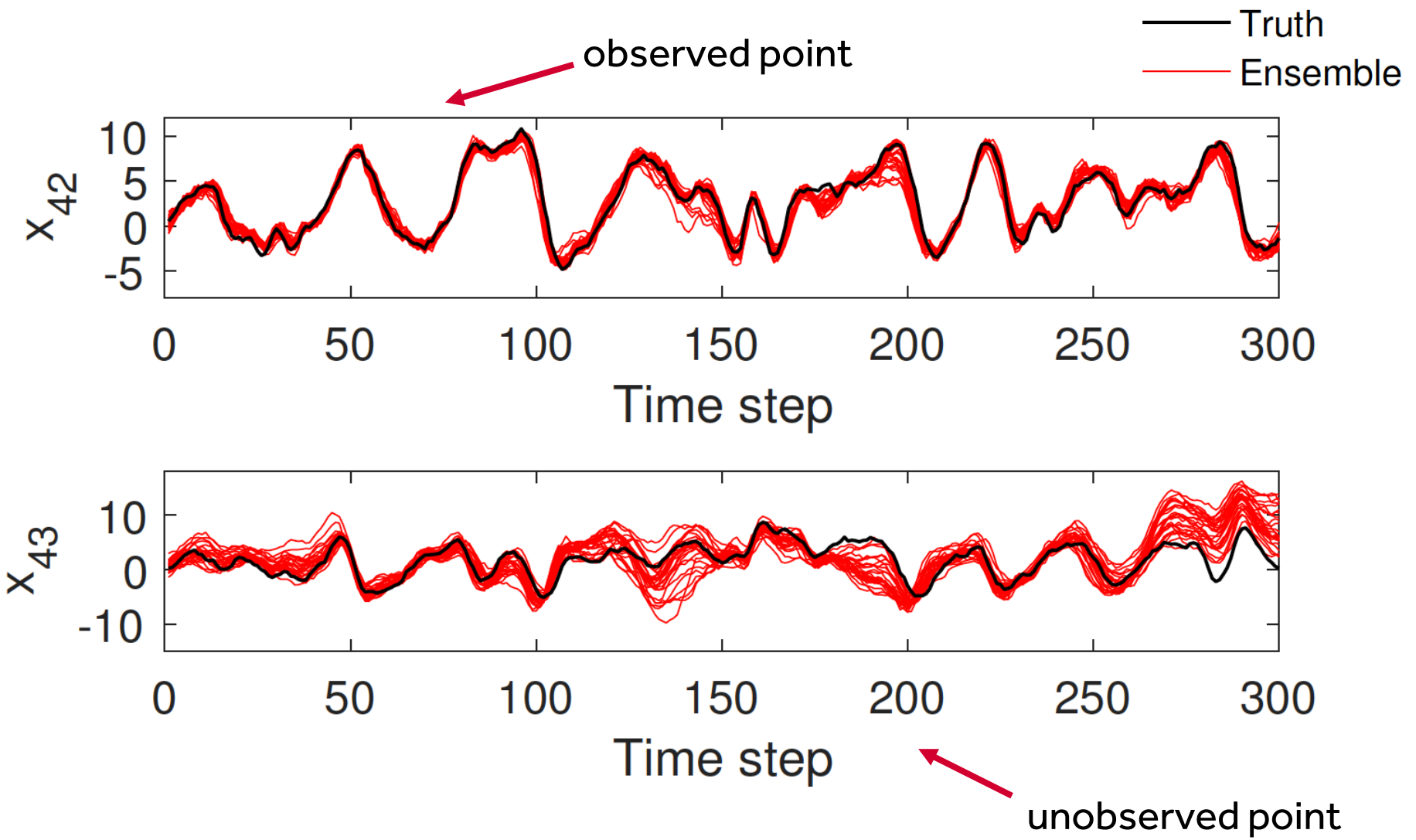
Equation for α_i has the same form as in the single-stage case.

Lorenz 96 model experiments

- 1000 dimensional
- 25 particles
- $\Delta t = 0.05 \sim 6$ hours
- observations every time steps, half of state

Particle trajectories

University of Reading



University of Reading

Comments

- Second perturbation introduces an additional tuning parameter
  - how should we choose β?
  - can tuning be automated?
  - could also adjust dynamically?
- Is $w_{target} = \min(w_i^{max})$ the best choice? May be better to target the mean or median weight.

These questions plus solution for α_i in the limit $N_x \longrightarrow \infty$ are currently being explored

The Ensemble Transform Particle Filter (ETPF)

(Reich, 2013)

University of Reading

Avoids resampling by finding a linear transportation map between the prior and posterior ensemble such that

- prior particles are minimally modified
- posterior particle have equal weights

Each posterior particle i is written as linear combination of the N_e prior particles as

$$\mathbf{x}_j^a = N_e \sum_{i=1}^{N_e} \mathbf{x}_i^f \, t_{ij}$$

where t_{ij} are entries of a N_e x N_e transformation matrix $\mathbf{T}$, that satisfy

University of Reading

$$\sum_{i=1}^{N_e} t_{ij} = \frac{1}{N_e}, \qquad \sum_{j=1}^{N_e} t_{ij} = w_i$$

ensures that the particles have the correct mean

($N_e^2 - 2$) undetermined elements t_{ij} to find - infinite solutions

The ETPF finds these by minimising the movement from old to new particles

$$J(\mathbf{T}) = \sum_{i,j}^{N_e} t_{ij} \left\| \mathbf{x}_i^f - \mathbf{x}_j^f \right\|^2$$

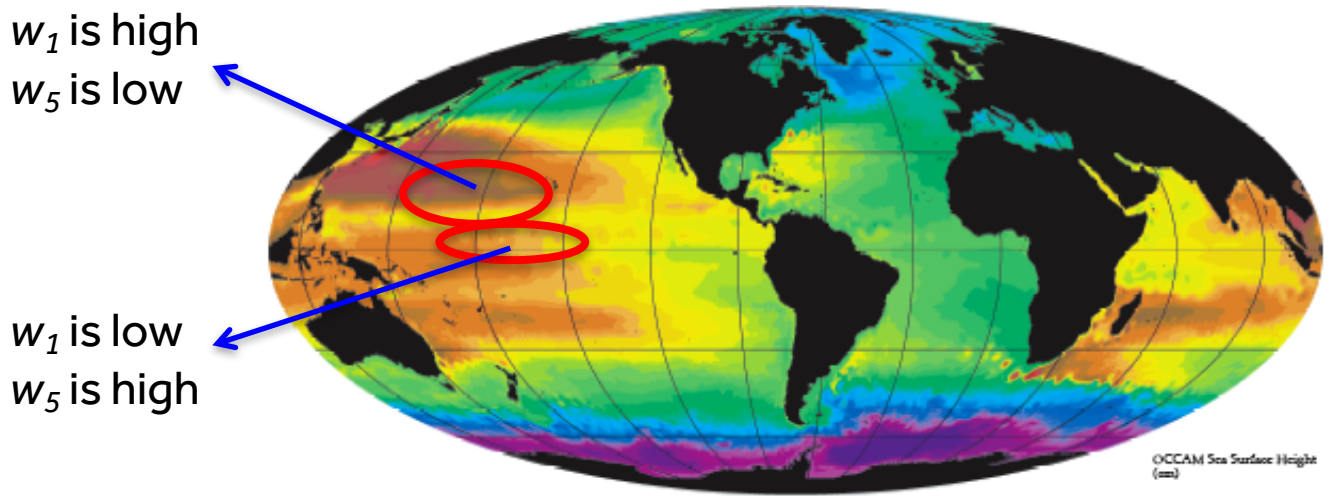
with the condition $t_{ij} \geq 0$

University of Reading

- Minimisation takes takes $O(N_e^2 \log N_e)$ operations and is performed at every grid point so expensive algorithm
  - possibility to reduce this to larger areas
- By construction, the ETPF underestimates the posterior covariance
- If model is deterministic need to add small random noise to the particles to avoid collapse
  - usually assume $\boldsymbol{\xi} \sim N(\mathbf{0}, h^2\mathbf{P}^f)$ with $0 < h < 1$
  - random perturbation is adhoc - related to inflation in EnKFs.
- ETPF can be localized but need to ensure that large-scale balances are not broken.

Localization in particle filters (1)

- Easy to calculate weights locally, i.e. make weights space dependent by only using observations close to the spatial point
- Main issue is at the resampling step - how to combine particles from different areas in the domain?



- Constructing a new particle that consists of one particle in one part of the domain and another particle in another part of the domain will lead to problems at the boundary between the two

University of Reading

Localization in particle filters (2)

- the ETPF can easily be localized by
  - only taking into account observations close to each grid point
  - making the transformation matrix space dependent to ensure smooth transitions between different regions
- **van Leeuwen 2009** proposes and discusses several localization methods
- The most obvious approach is to weight and resample locally, and then somehow `glue' the resampled particles together by averaging at the boundaries between them

Local Particle Filter

(Penny & Miyoshi, 2016)

University of Reading

- use idea of gluing particles back together but with extra averaging
- independent analysis is computed locally at each grid point
- the weight of particle *i* at grid point *j* is calculated as based on the likelihood of only those observations $\mathbf{y}_j$ that are within the given localization area

$$w_{i,j} = \frac{p(\mathbf{y}_j|\mathbf{x}_{i,j})}{\sum_{k=1}^{N_e} p(\mathbf{y}_j|\mathbf{x}_{i,j})}$$

- SIR is then used to obtain particles $x^a_{i,j}$ for each grid point *j*

- to ensure the posterior particles are smooth, for each grid point j, each particle is smoothed by averaging over N_p neighbouring points

$$x^a_{i,j} = \frac{1}{2} x^a_{i,j} + \frac{1}{N_p} \sum_{k=1}^{2N_p} x^a_{i,j_k}$$

  where j_k for $k = 1, \ldots, N_p$ denotes the grid point index for those points in the localization area around j

- shown to solves the degeneracy problem in simple 1D systems but is not clear if it will work well in complex systems

University of Reading

Localized Particle Filter (Poterjoy, 2016)

For each observation *k = 1:N_y*

1. calculate adapted weights for each particle *i = 1:N_e*

$$\tilde{w}_i = [p(y_k|\mathbf{x}_i) - 1]\,\alpha + 1$$

   then normalise by their sum

2. resample particles according to normalised weights to form particles $\mathbf{X}_{k_i}$
3. For each grid point *j = 1:Nx*
   - calculate localized weights

$$w_{i,j}^{(k)} = w_{i,j}^{(k-1)}\left[(p(y_k|\mathbf{x}_i) - 1)\,\rho(y_k, x_j, r)\alpha + 1\right]$$

$\rho(\cdot)$ is the localization function with localization radius *r*.

University of Reading

3. ...
   - calculate the posterior mean for this observation at this grid point

$$\overline{x}_j = \sum_{i=1}^{N_e} w_i x_{i,j}$$

   - calculate new particles at grid point *j*

$$x_{i,j}^{(k)} = \overline{x}_j + r_{1,j}\left(x_{k_i,j}^{k-1} - \overline{x}_j\right) + r_{2,j}\left(x_{i,j}^{k-1} - \overline{x}_j\right)$$

   $r_{1,j}$ and $r_{2,j}$ are scalars that ensure smooth posterior fields

end

4. Apply probability mapping for higher-order corrections.

University of Reading

Conclusions

- Wide range of filters and smoothers can be derived from Bayes Theorem
- Best method will be system dependent
- In practice we will be dealing with high dimensional, highly non-linear, complex geophysical systems
  - basic PF formulation can never work for these systems
- Fully nonlinear PF variants have been developed that have been shown to work for large dimensional systems with a limited number of particles
  - Localization needs further exploration ...
  - Proposal-density freedom needs further exploration ...
  - Transportation Particle Filters need further exploration...
- But first localized Particle Filter has been implemented by DWD for weather prediction (Potthast et al, MWR 2019)

Some references

University of Reading

Ades, M. and van Leeuwen, P. J. 2013. *An exploration of the equivalent weights particle filter*. Q. J. R. Meteorol. Soc. 139, 820–840.

Ades M, van Leeuwen PJ. 2015b. The equivalent-weights particle filter in a high-dimensional system. Q. J. R. Meteorol. Soc. 141: 484–503.

Penny, S. and Miyoshi, T. (2016). *A local particle filter for high dimensional geophysical systems*. Nonlinear Processes Geophys. 23, 391–405.

Poterjoy, J. (2016). *A localized particle filter for high-dimensional nonlinear systems.* Mon. Wea. Rev. 144, 59–76.

Potthast et al (2019), *A Localized Adaptive Particle Filter within an Operational NWP Framework.* Mon. Wea. Rev., 147, 345–362

Reich, S. (2013) A nonparametric ensemble transform method for Bayesian inference. SIAM Journal on Scientific Computing 4(35), 2013–2024.

Snyder, et al (2008) *Obstacles to high-dimensional particle filtering.* Mon. Wea. Rev. 136, 4629–4640.

Skauvold, et al (2019) *A revised implicit equal-weights particle filter.* Q. J. R. Meteorol. Soc. In press

Vetra-Carvalho et al (2018) *State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems*, Tellus A, 70:1, 1-43

Zhu et al (2016) *Implicit equal-weights particle filter*. Q. J. R. Meteorol. Soc. 142: 1904–1919

Plus many papers by PJ van Leeuwen.