A Hybrid Kalman-Nonlinear Ensemble Transform Filter

Lars Nerger

Alfred Wegener Institute, Bremerhaven, Germany

With inputs from

Paul Kirchgessner, Julian Tödter, Bodo Ahrens

on the NETF

ALFRED-WEGENER-INSTITUT
HELMHOLTZ-ZENTRUM FÜR POLAR-
UND MEERESFORSCHUNG

Motivation

- Looking for alternatives to ensemble Kalman filters for high-dimensional nonlinear systems
- Nonlinear Ensemble Transform Filter – NETF (Tödter & Ahrens, MWR, 2015)
  - can beat LETKF if ensemble is large enough
- Can we combine the strengths of LETKF and NETF?
  - hybrid LETKF-NETF
- Can we get improved assimilation with small ensembles?

Ensemble filters – ensemble Kalman filters & NETF

- represent state and its error by ensemble $\mathbf{X}$ of N states
- Forecast:
  - Integrate ensemble with numerical model
- Analysis:
  - update ensemble mean $$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{X}'^f \tilde{\mathbf{w}}$$
  - update ensemble perturbations $$\mathbf{X}'^a = \mathbf{X}'^f \mathbf{W}$$

  (both can be combined in a single step)

- Ensemble Kalman filters & NETF: Different definitions of
  - weight vector $\tilde{\mathbf{w}}$
  - Transform matrix $\mathbf{W}$

ETKF (Bishop et al., 2001)

- Ensemble Transform Kalman filter:
  - Transform matrix

$$\mathbf{A}^{-1} = (N-1)\mathbf{I} + (\mathbf{H}\mathbf{X}'^{f})^{T}\mathbf{R}^{-1}\mathbf{H}\mathbf{X}'^{f}$$

  - Mean update weight vector

$$\tilde{\mathbf{w}} = \mathbf{A}(\mathbf{H}\mathbf{X}'^{f})^{T}\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{H}\overline{\mathbf{x}^{f}}\right)$$

(depends on $\mathbf{R}$ and $\mathbf{y}$)

  - Transformation of ensemble perturbations

$$\mathbf{W} = \sqrt{(N-1)}\mathbf{A}^{-1/2}\mathbf{\Lambda}$$

(depends only on $\mathbf{R}$, not $\mathbf{y}$)

Particle filters – fully nonlinear ensemble filters

- Avoid changing ensemble members ('particles')
- Instead: give particles a weight and change it at the analysis step
  - Initial weight: 1/N for all particles
- Weights are given by statistical likelihood of an observation
- Example: With Gaussian observation errors (for each particle *i*):

$$\tilde{w}^i \sim \exp\left(-0.5(\mathbf{y} - \mathbf{H}\mathbf{x}_i^f)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{H}\mathbf{x}_i^f)\right)$$

- Ensemble mean state computed with weights

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{X}'^f \tilde{\mathbf{w}} = \mathbf{X}^f \tilde{\mathbf{w}}$$

- This update does not assume any distribution of the state errors (and is not limited to Gaussian distributions)

Nonlinear Ensemble Transform Filter - NETF

- NETF (Tödter & Ahrens, MWR, 2015)
  - Mean update from Particle Filter weights: for all particles *i*

$$\tilde{w}^i \sim \exp\left(-0.5(\mathbf{y} - \mathbf{H}\mathbf{x}_i^f)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}_i^f)\right)$$

  - Ensemble update
    - Transform ensemble to fulfill analysis covariance (like KF, but not assuming Gaussianity)
    - Derivation gives

$$\mathbf{W} = \sqrt{N}\left[\mathrm{diag}(\tilde{\mathbf{w}}) - \tilde{\mathbf{w}}\tilde{\mathbf{w}}^T\right]^{1/2}\Lambda$$

($\boldsymbol{\Lambda}$: mean-preserving random matrix; useful for stability)

  - Localization as in LETKF possible

Tödter, J. and Ahrens, B. 2015. *Mon. Wea. Rev.* **143**,1347–1367

Difference of ETKF and NETF

- ETKF parameterizes ensemble distribution by a Gaussian distribution
  - only mean state update depends on values of observations
- NETF uses particle filter weights to ensure correct update of ensemble mean and covariance
  - mean and covariance depend on values of observations

- Filter update
  - ETKF: linear in observations

$$\tilde{\mathbf{w}} = \mathbf{A}(\mathbf{H}\mathbf{X}'^{f})^{T}\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{H}\overline{\mathbf{x}^{f}}\right)$$

  - NETF: nonlinear in observations

$$\tilde{w}^{i} \sim \exp\left(-0.5(\mathbf{y} - \mathbf{H}\mathbf{x}_{i}^{f})^{T}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}_{i}^{f})\right)$$

Configuration of Lorenz-96 model experiments

Lorenz-96:

- 1-dimensional periodic wave
- Chaotic dynamics

Configuration for assimilation experiments

- State dimension: 80
- Observed: 40 grid points
- Forecast length: 8 time steps
- Experiment length: 5000 time steps (625 analysis steps)
- Observation error standard deviation: 1.0

→ this is a difficult case for the assimilation
(and more realistic than typical 1-step forecast configuration)

PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

PDAF - Parallel Data Assimilation Framework

- a program library for ensemble data assimilation
- provide support for parallel ensemble forecasts
- provide fully-implemented & parallelized filters and smoothers (EnKF, LETKF, NETF, EWPF … easy to add more)
- easily useable with (probably) any numerical model (applied with NEMO, MITgcm, FESOM, HBM, TerrSysMP, …)
- run from laptops to supercomputers (Fortran, MPI & OpenMP)

NETF and Lorenz-96 model fully implemented in PDAF

Open source:
Code, documentation & tutorials at

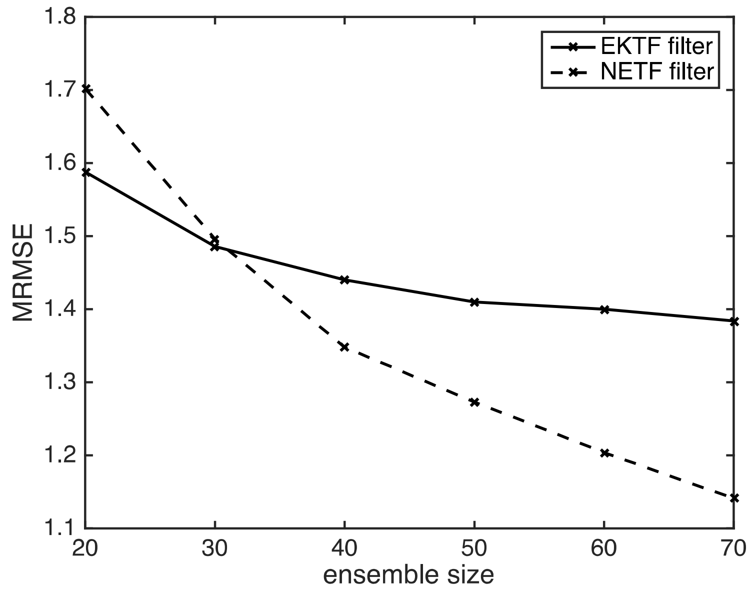
http://pdaf.awi.de

L. Nerger, W. Hiller, Computers & Geosciences 55 (2013) 110-118

AWI

Performance of NETF – Lorenz-96

- Double-exponential observation errors
- Run all experiments
  10x with different initial ensemble
- NETF beats ETKF for ensemble size > 30



- For high-dimensional case (NEMO, state dimension 300,000)
  - similar RMSE for LETKF and NETF for N=120

Kirchgessner, Tödter, Ahrens, Nerger. (2017) Tellus A, **69**, 1327766

Hybrid filter variants

1-step update (*HSync*)

$$\mathbf{X}^a_{HSync} = \overline{\mathbf{X}}^f + (1-\gamma)\Delta\mathbf{X}_{NETF} + \gamma\Delta\mathbf{X}_{ETKF}$$

- $\Delta\mathbf{X}$ is assimilation increment of a filter
- γ is hybrid weight (between 0 and 1; 1 for fully LETKF)

2-step updates

Variant 1 (*HNK*): NETF followed by LETKF

$$\tilde{\mathbf{X}}^a_{HNK} = \mathbf{X}^a_{NETF}[\mathbf{X}^f, (1-\gamma)\mathbf{R}^{-1}]$$
$$\mathbf{X}^a_{HNK} = \mathbf{X}^a_{ETKF}[\tilde{\mathbf{X}}^a_{HNK}, \gamma\mathbf{R}^{-1}]$$

- Both steps computed with increased **R** according to γ

Variant 2 (*HKN*): LETKF followed by NETF

Choosing hybrid weight γ

- Hybrid weight shifts filter behavior
- How to choose it?

Some possibilities:

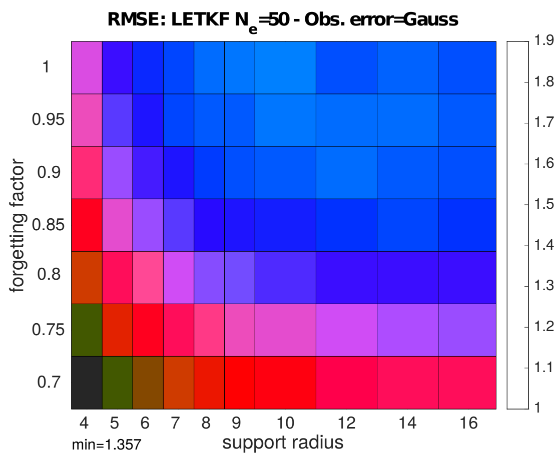
- Fixed value
- Adaptive
  - According to which condition?
  - For hybrid particle-EnKF, Frei & Kuensch (2013) suggested using effective sample size $N_{eff} = \sum_i 1/(w^i)^2$
    - Choose γ so that N_{eff} is as small as possible but above minimum limit
  - Alternative used here

$$\gamma_{adap} = 1 - N_{eff}/N_e$$

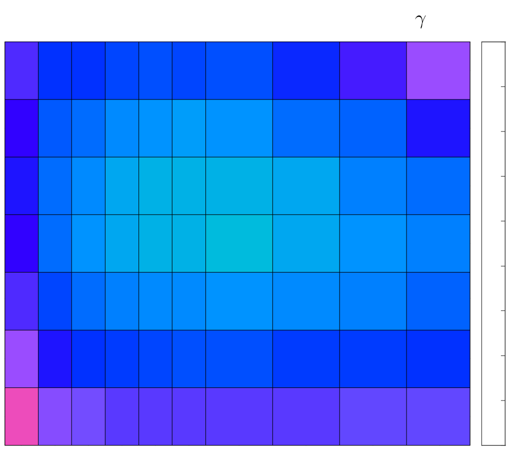
(close to 1 if N_{eff} small)

Test with Lorenz-96 model (n=80 as before)

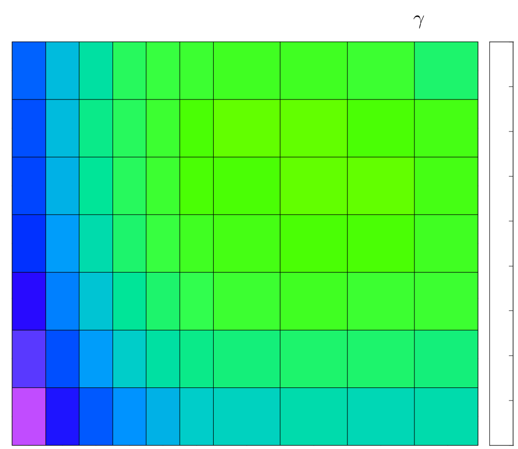
Ensemble size N=50



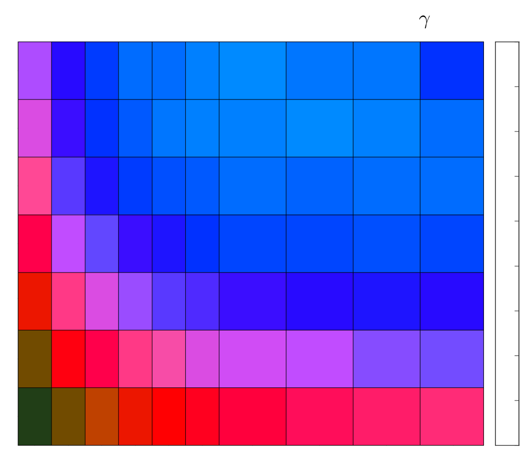
- All hybrid variants improve estimates compared to LETKF & NETF
- Similar stability as LETKF
- Dependence on forgetting factor & localization radius like LETKF
- Similar optimal localization radius
- Largest improvement for variant HNK (NETF before LETKF)



4% improvement

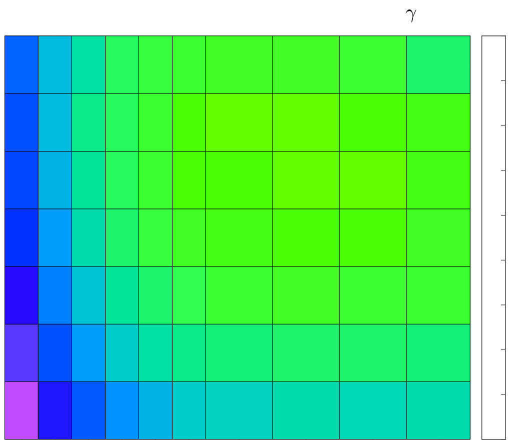


16% improvement



1% improvement

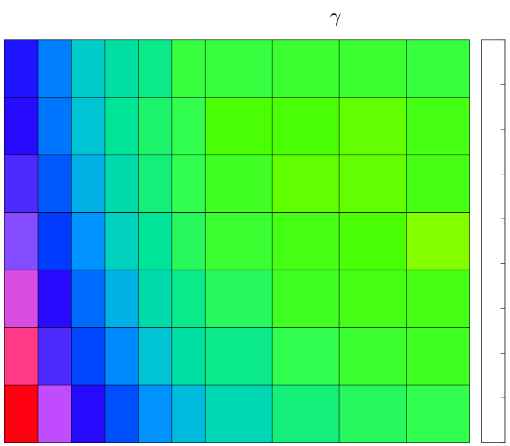
N=50 – adaptive and fixed hybrid weight γ

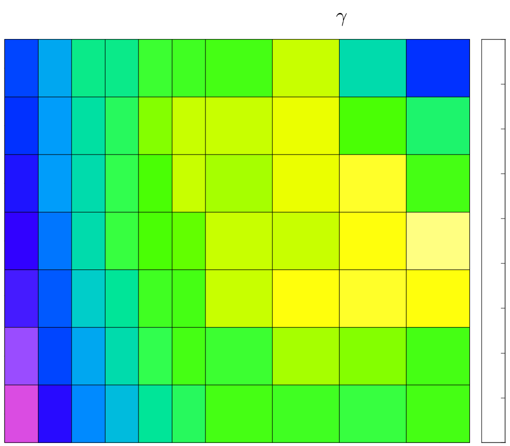


Consider only version HNK

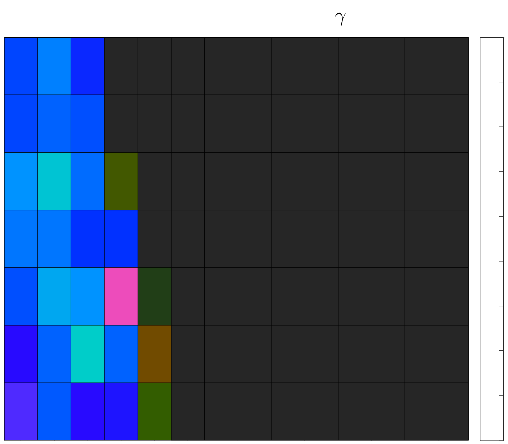
- Fixed γ also successful, smaller errors than hybrid
- Has to be close to 1.0 (small NETF fraction)
- Smaller γ reduces stability

16% improvement



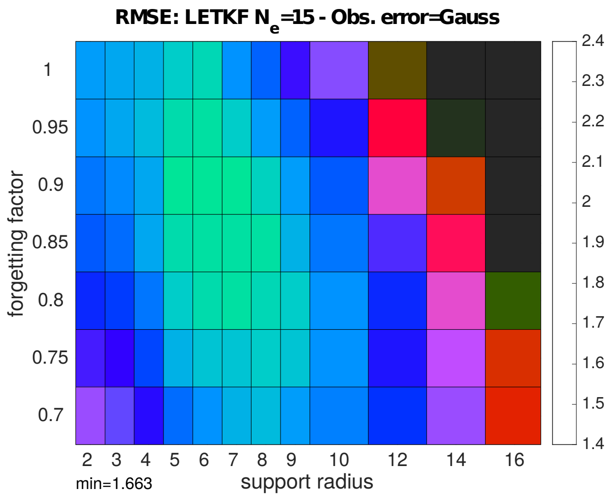


22% improvement

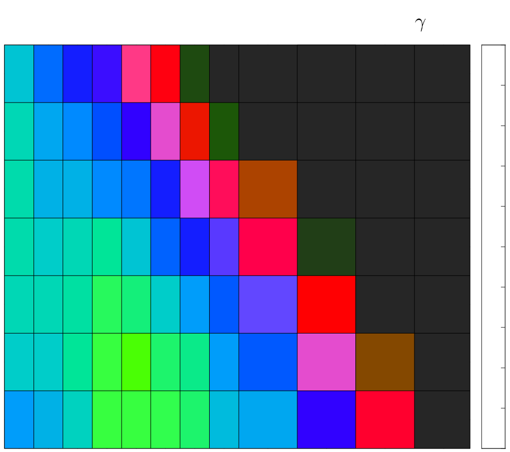


5% improvement

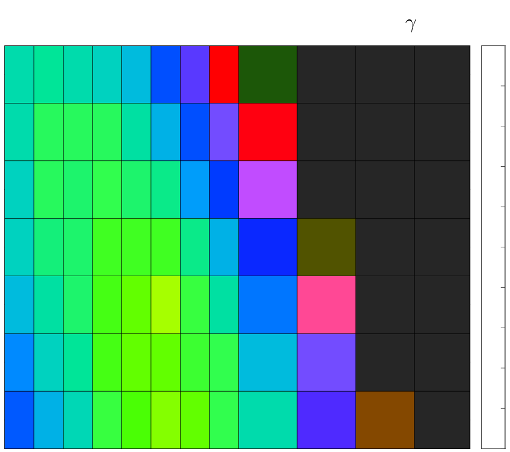
Small Ensemble N=15



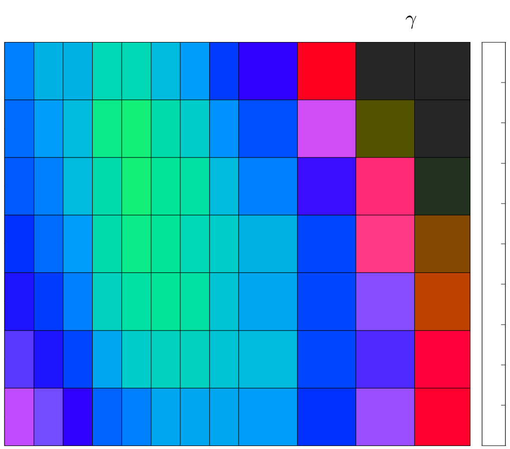
- Hybrid still positive influence
- Smaller improvement than for N=50
- Optimal parameters for HSync & HNK different from HKN
- HSync and HNK more similar



6% improvement

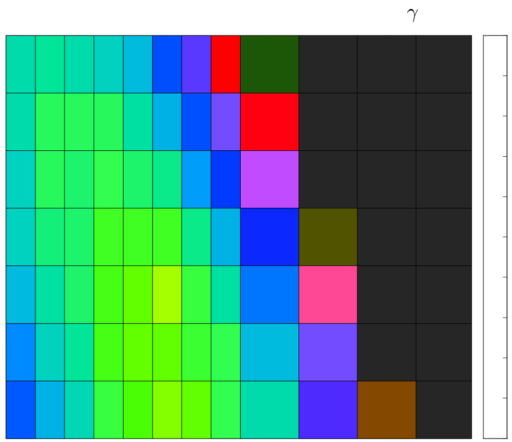


8% improvement



1% improvement

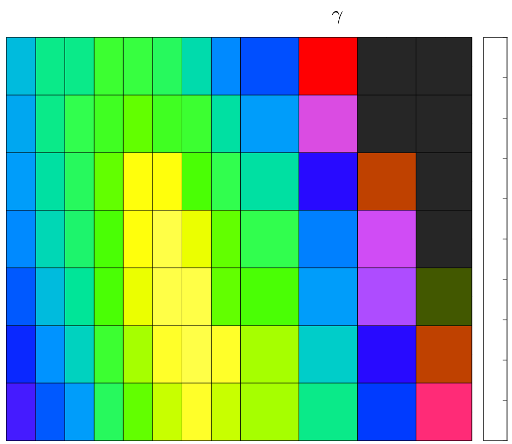
Small Ensemble N=15

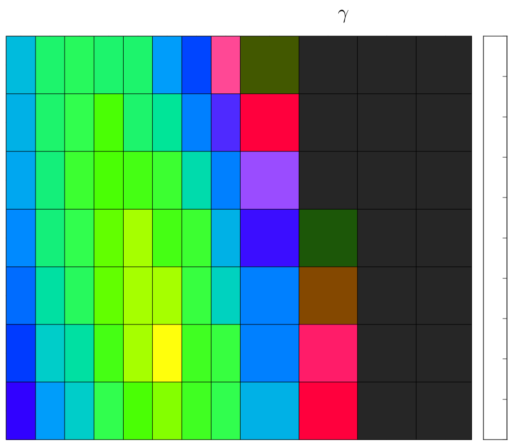


Fixed γ

- reduces error compared to adaptive γ
- Can increase stability region
- Needs to be even closer to 1 than for N=50

8% improvement





11% improvement

9% improvement

Summary

- Nonlinear ensemble transform filter (NETF)
  - Update state estimate as particle filter
    Transform ensemble using covariance matrix
- Hybrid LETKF-NETF
  - Combine analysis updates controlled by hybrid weight
  - Smaller errors than LETKF and NETF
  - Variant NETF-before-LETKF yield best results
  - Choosing hybrid weight is key for good performance
    - Fixed hybrid height showed lower errors compared to simple adaptive weight
    - Should weight really depend on effect sample size?

Thank you!