5. International Symposium on Data Assimilation, Reading, UK, July 18 – 22, 2016

High-Dimensional Nonlinear Data Assimilation with the Nonlinear Ensemble Transform Filter (NETF) and its Smoother Extension

Paul Kirchgessner, **Lars Nerger**

Alfred Wegener Institute, Bremerhaven, Germany

Julian Tödter, Bodo Ahrens

University of Frankfurt, Frankfurt, Germany

SANGOMA

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

Overview

- Study new Nonlinear Ensemble Transform Filter – NETF (Tödter & Ahrens, MWR, 2015)
- Extend NETF for smoothing
- Test filter and smoother in realistic high-dimensional idealized ocean data assimilation experiments

Ensemble filters – ensemble Kalman filters & NETF

- represent state and its error by ensemble $\mathbf{X}$ of m 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 $\mathbf{W}$

Nonlinear ensemble transform filter - NETF

- Ensemble Kalman:
  - Transformation according to KF equations

- 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

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

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

(Almost same formulation: Xiong et al., Tellus, 2006)

Ensemble Smoothers – ETKS & NETS

- Smoother: Update past ensemble with future observations
- Rewrite ensemble update as
  - Filter: $$\mathbf{X}^a_{k|k} = \mathbf{X}^f_{k|k-1}\hat{\mathbf{W}}_k$$

analysis time

Observations used up to time

- Smoother at time $i < k$

$$\mathbf{X}^a_{i|k} = \mathbf{X}^f_{i|k-1}\hat{\mathbf{W}}_k$$

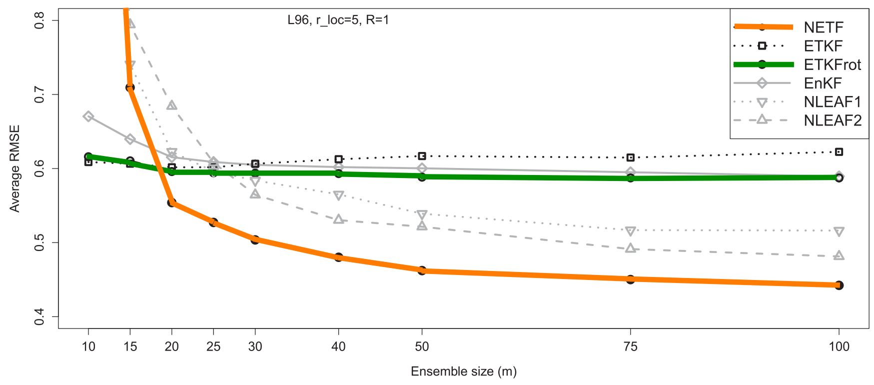
➢ works likewise for ETKS and NETS

➢ also possible for localized filters

See, e.g., Nerger, Schulte & Bunse-Gerstner, QJRMS 140 (2014) 2249–2259

Performance of NETF – Lorenz-96

- Performance for small model (Lorenz-96)
- In Tödter & Ahrens (MWR, 2015)



- NETF beats ETKF for m=20 and larger

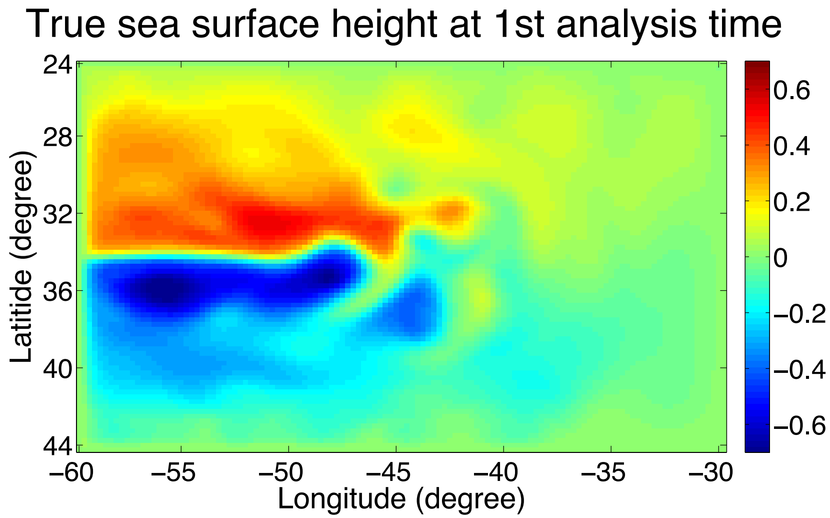
How do NETF and NETS perform
in a more realistic case?

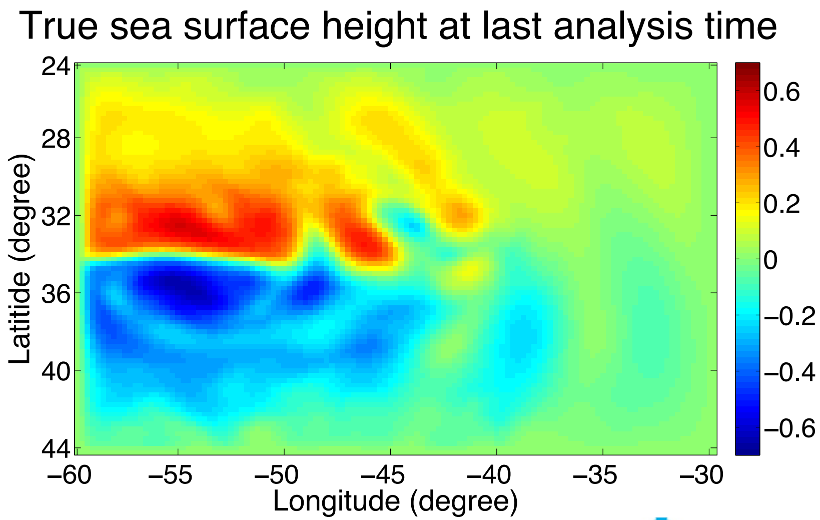
Assimilation into NEMO

European ocean circulation model

Model configuration

- box-configuration SEABASS
- ¼° resolution
- 121x81 grid points, 11 layers (state vector ~300,000)
- wind-driven double gyre (a nonlinear jet and eddies)
- medium size SANGOMA benchmark





SANGOMA

www.data-assimilation.net

AWI

PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

PDAF - Parallel Data Assimilation Framework

- a program library for data assimilation
- provide support for ensemble forecasts
- provide fully-implemented filter and smoother algorithms (LETKF, LSEIK, LESTKF, …)
- easily useable with (probably) any numerical model (applied with NEMO, MITgcm, FESOM, MPI-ESM, HBM)
- makes good use of supercomputers
- first public release in 2004; continued development

Open source:
Code and documentation available at

http://pdaf.awi.de

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

AWI

Online coupling: Minimal changes to NEMO

Add to *mynode* (lin_mpp.F90) just before init of myrank

```
#ifdef key_USE_PDAF
  CALL init_parallel_pdaf(0, 1, mpi_comm_opa)
#endif
```

Add to *nemo_init* (nemogcm.F90) at end of routine

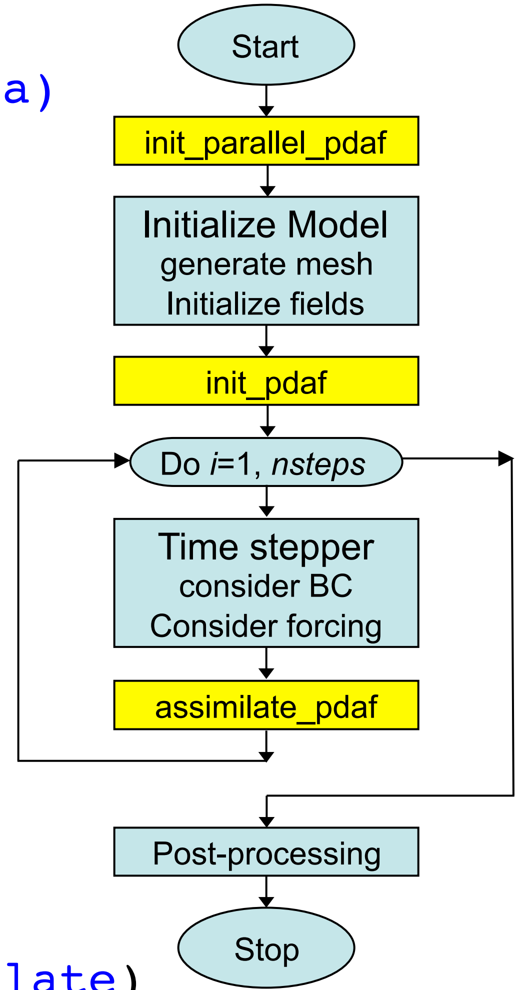
```
#ifdef key_USE_PDAF
  CALL init_pdaf()
#endif
```

Add to *stp* (step.F90) at end of routine

```
#ifdef key_USE_PDAF
  CALL assimilate_pdaf()
#endif
```

Modify *dyn_nxt* (dynnxt.F90)

```
#ifdef key_USE_PDAF
  IF((neuler==0 .AND. kt==nit000).OR.assimilate)
#else
```



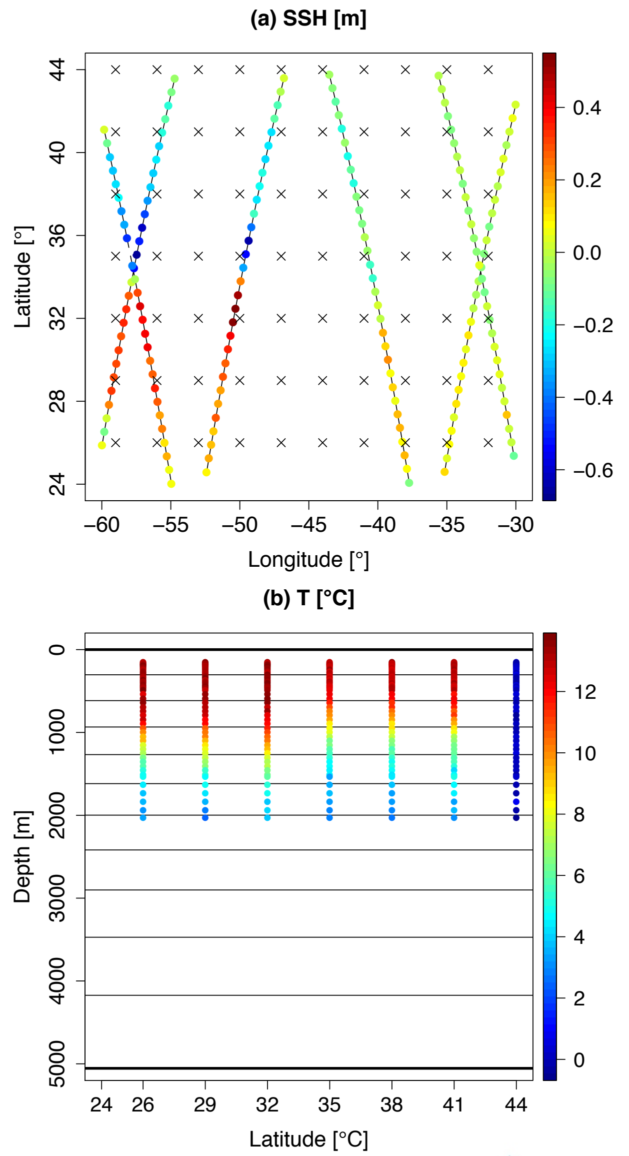
Observations and Assimilation Configuration

Observations

- Simulated satellite sea surface height SSH (Envisat & Jason-1 tracks), 5cm error
- Temperature profiles on 3°x3° grid, surface to 2000m, 0.3°C error

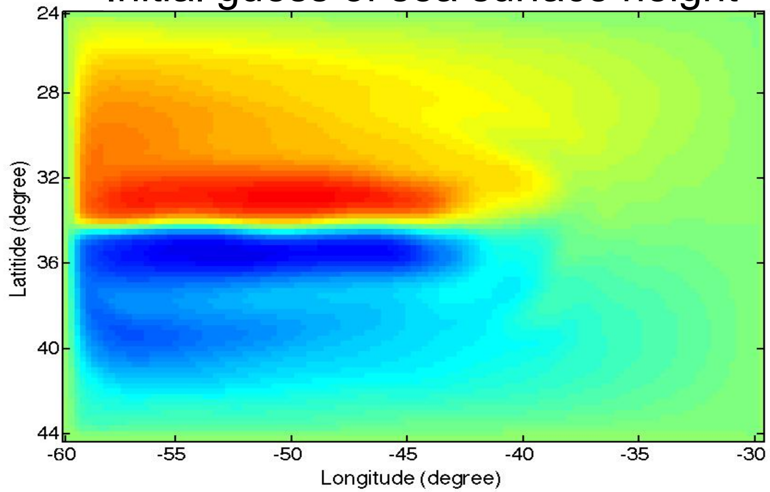
Data Assimilation

- Ensemble size 120
- ETKF and LETKF
- Localization: weights on matrix $\mathbf{R}^{-1}$ (Gaspari/Cohn'99 function, 2.5° radius)
- Assimilate each 48h over 360 days

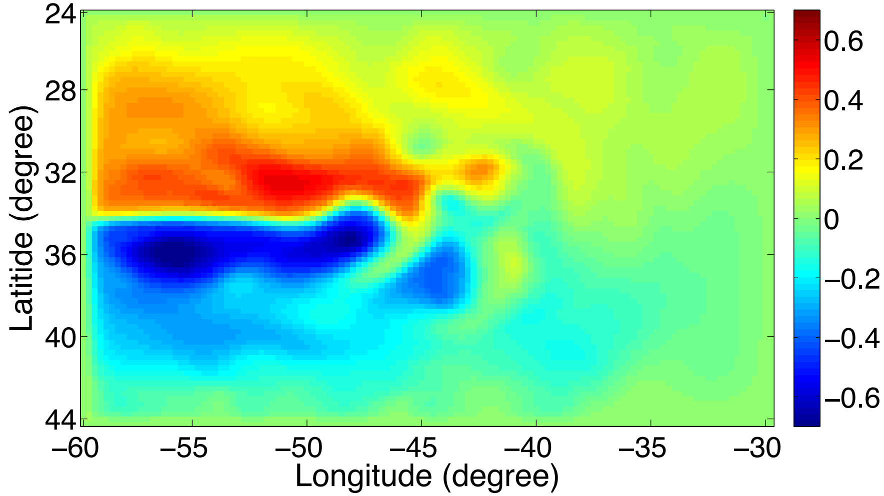


Application of LETKF

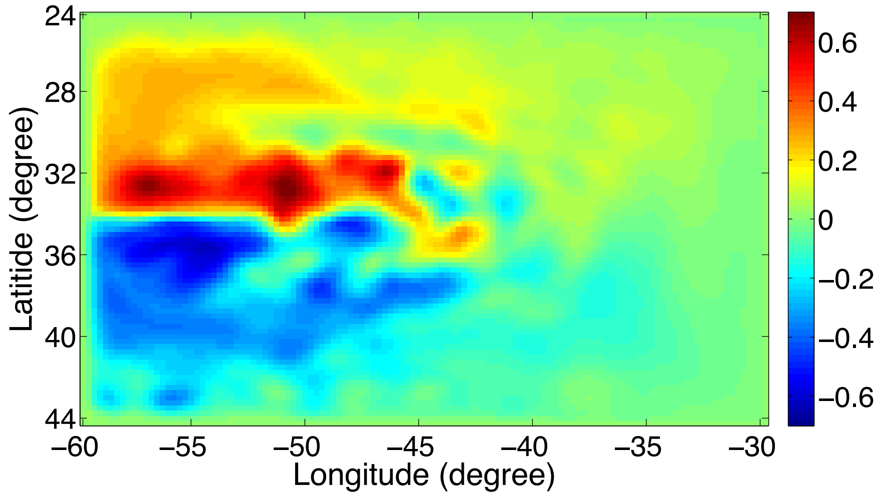
Initial guess of sea surface height



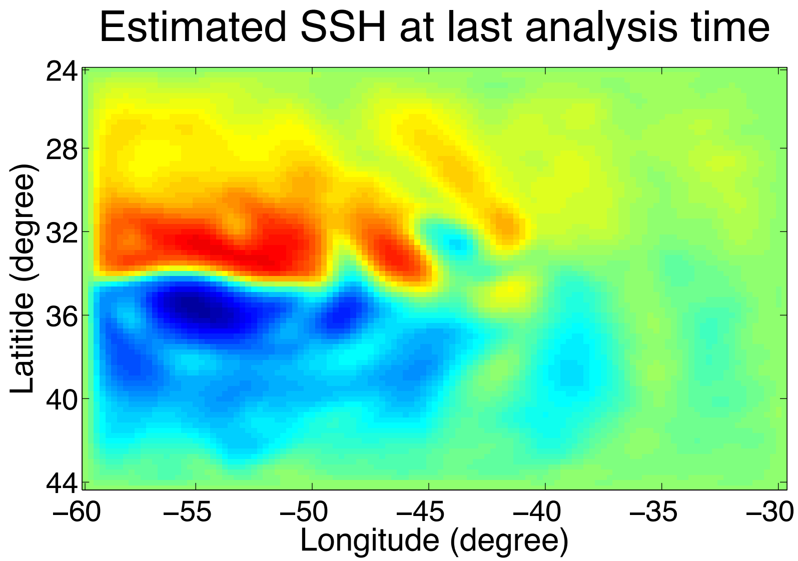
True sea surface height at 1st analysis time

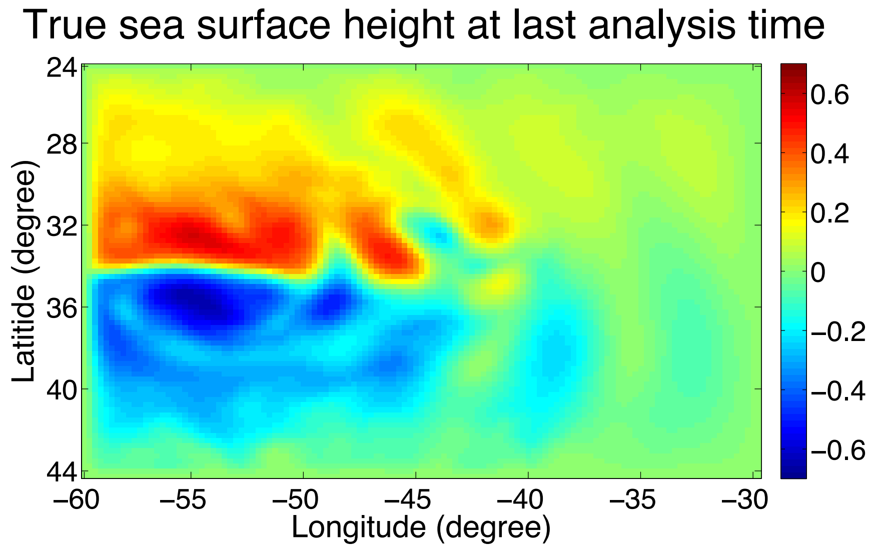


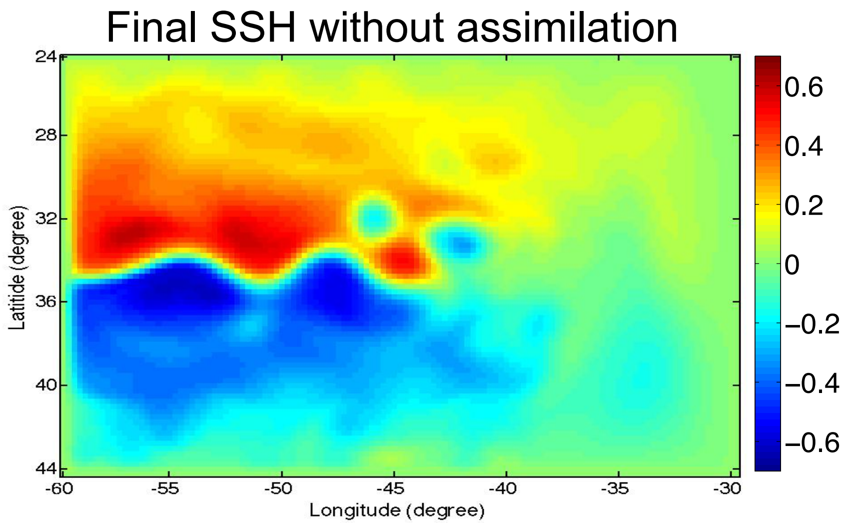
Estimated SSH at 1st analysis time



Application of LETKF (2)



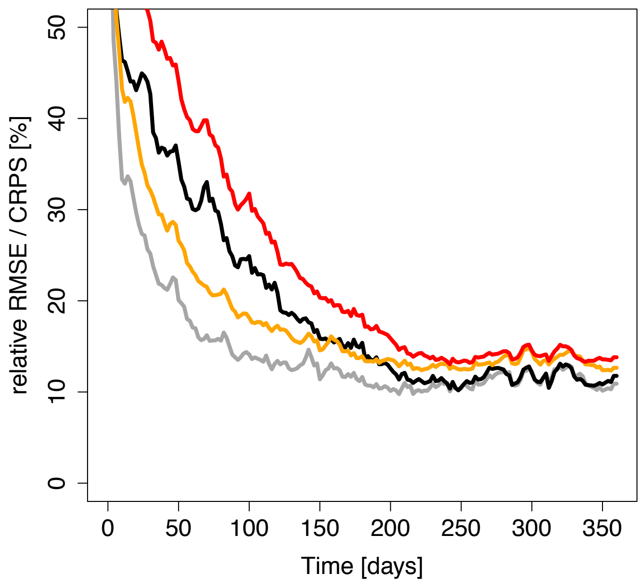




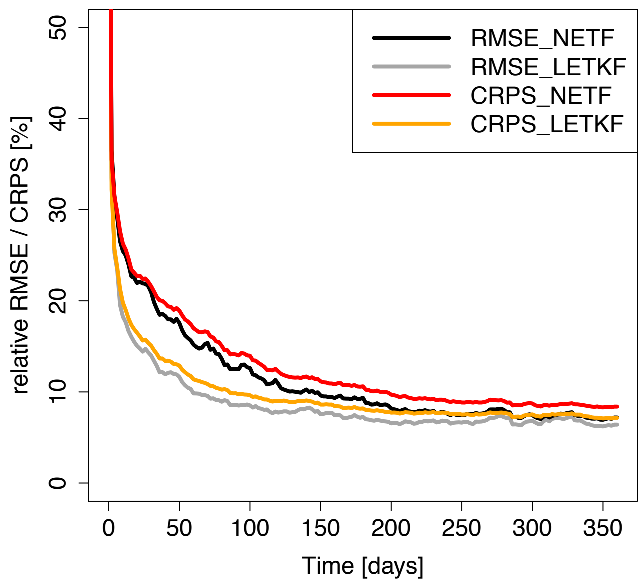
Filter performances in NEMO

- RMS errors reduced to 10% (velocities to 20%) of initial error
- Slower convergence for NETF, but to same error level as LETKF
- CRPS (Continuous Rank Probability Score) shows similar behavior

SSH: Relative error reduction



T: Relative error reduction

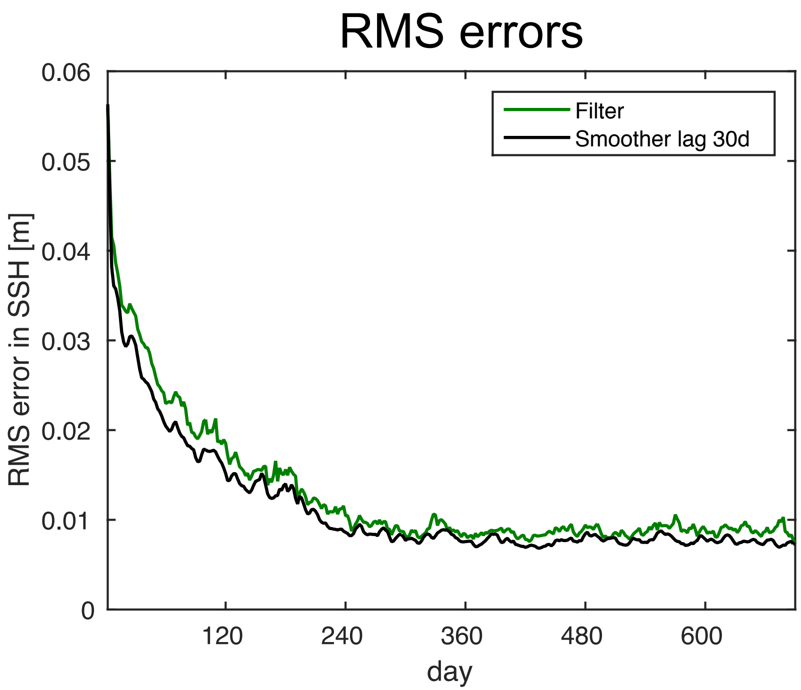


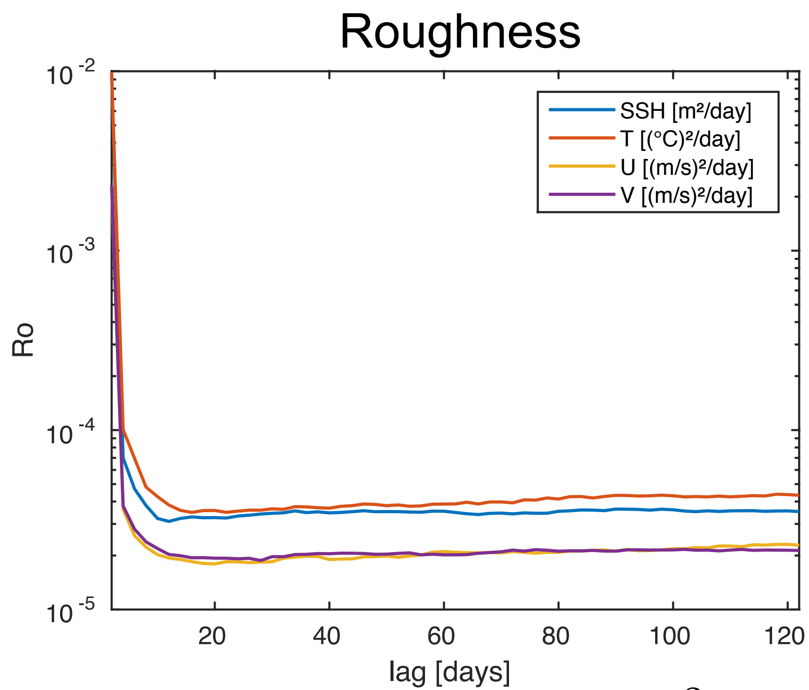
Tödter, Kirchgessner, Nerger & Ahrens, MWR 144 (2016) 409 – 427

AWI

Applying the smoother

- Smoother reduces filter errors by ~10%
- Can be useful as smoothing is cheap to compute
- Roughness of estimated trajectory is strongly reduced (smoothed)

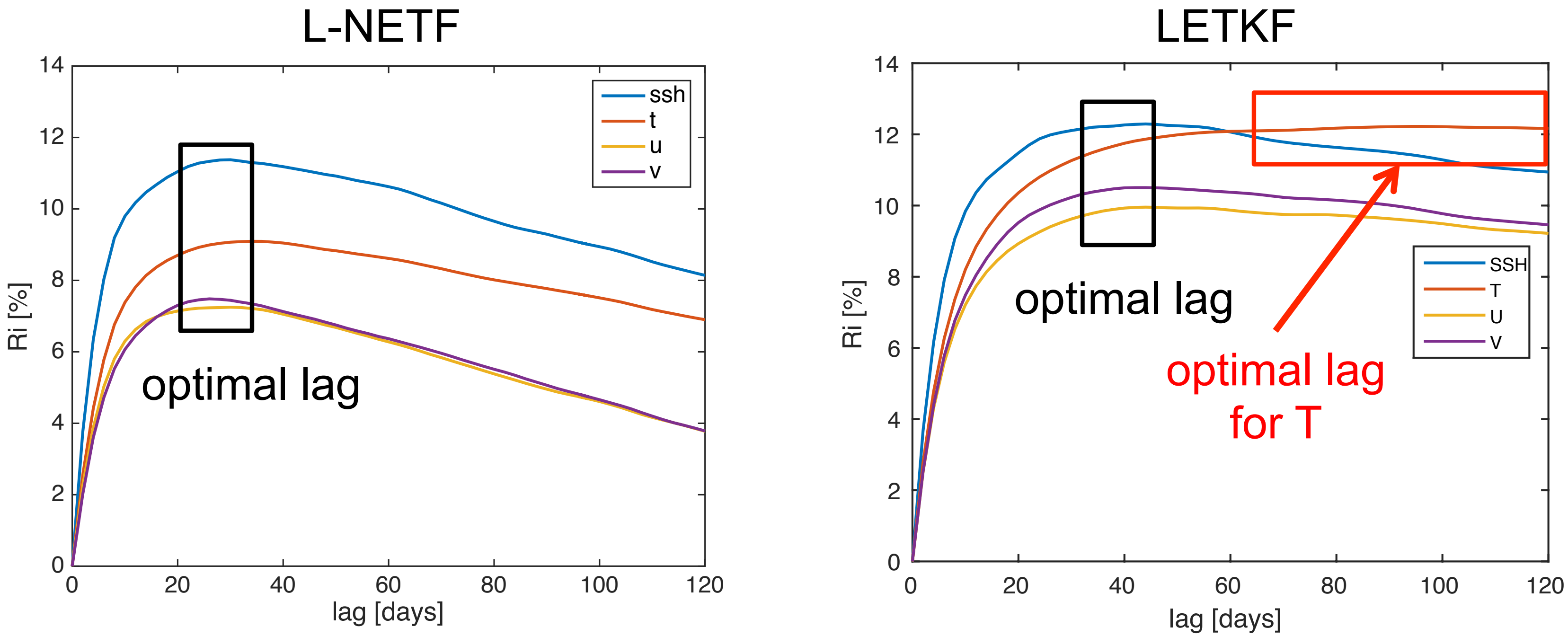




$$Ro = \int \left(\frac{dRMSE}{dt} \right)^2 dt$$

Different smoothing impact

- Consider relative improvement $$\mathrm{Ri} = 100 \cdot \left(1 - \frac{\mathrm{RMSE}_{smoother}}{\mathrm{RMSE}_{filter}}\right)$$



- Similar behavior for ssh (sea surface height)
- Distinct for T
  - Effect of distinct update schemes (NETF uses observation values for both state and ensemble update)

Summary

- Nonlinear ensemble transform filter/smoother (NETF/S)
  - Update state estimate as particle filter
  - Transform ensemble using covariance matrix
- NEMO ocean test case
  - NETF filtering performance similar to LETKF
  - Slower convergence
  - Sensitive on ensemble size
  - Successful smoothing
    - Dependence on lag distinct for LETKS & NETS (due to different update schemes)

Thank you!