Nonlinear data assimilation using synchronisation in a particle filter

Flavia R. Pinheiro [1]
Peter Jan van Leeuwen [1,2] Gernot Geppert [1,2]

[1]University of Reading

[2]National Centre for Earth Observation (NCEO)

Synchronisation and Data Assimilation

- Synchronisation phenomenon

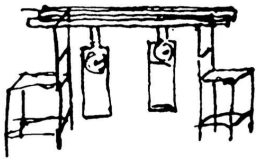


Figure 1: A drawing by Christiaan Huygens of his experiment in 1665.

Synchronisation and Data Assimilation

- Synchronisation phenomenon

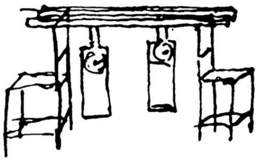


Figure 1: A drawing by Christiaan Huygens of his experiment in 1665.

- Data assimilation aims to *synchronise* the model evolution with the true evolution of the system, finding the best estimate of the state evolution and its uncertainty.
- Coupling is unidirectional, from the truth to the model, and incomplete, as observations are typically sparse and contain errors.

Motivation

$\longrightarrow$ High-dimension nonlinear DA

$\longrightarrow$ Particle Filters

$\longrightarrow$ Proposal density freedom

Synchronisation

Synchronisation framework

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t)) + g\frac{\partial S(x(t))}{\partial x(t)}^{\dagger}(Y(t) - S(t)) \tag{1}$$

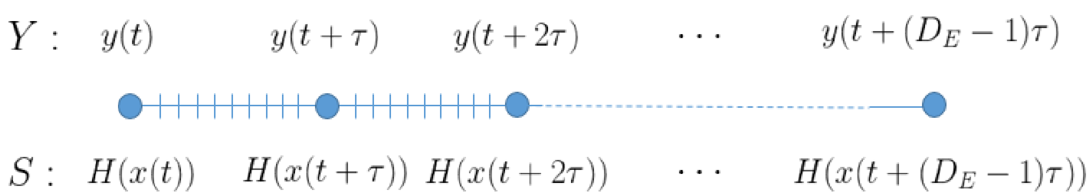
where g is a coupling constant, which is a tuning parameter, and $f(\mathbf{x}(t))$ is the nonlinear model (Rey et al (2014)).

Synchronisation framework

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t)) + g\frac{\partial S(x(t))}{\partial x(t)}^{\dagger}(Y(t) - S(t)) \quad (1)$$

where g is a coupling constant, which is a tuning parameter, and $f(\mathbf{x}(t))$ is the nonlinear model (Rey et al (2014)).

- Time embeddings (D_E):



(both vectors $\in \Re^{D_E * D_y}$)

- Jacobian matrix $\frac{\partial S(x(t))}{\partial x(t)}$: of size $(D_E * D_y) \times D_x$.
- $\frac{\partial S(x(t))}{\partial x(t)}^{\dagger} \rightarrow$ SVD

  - **Issue**: Construction of the Jacobian matrix requires propagation of a $D_x \times D_x$ matrix.
  This is prohibitively expensive for high-dimensional systems.
  - **Solution**: construct the Jacobian matrix through an ensemble approximation.

Ensemble synchronisation framework

- Generate an ensemble of i initial states;
- Form the initial ensemble perturbation matrix:

$$\boldsymbol{X}(0)_i = x(0)_i - \bar{x}(0) \tag{2}$$

  a $D_x \times Nens$ matrix;
- Propagate forward in time each ensemble member for τ time steps and form the ensemble perturbation matrix:

$$\boldsymbol{X}(\tau)_i = x(\tau)_i - \bar{x}(\tau) \tag{3}$$

  with the same dimension as $\boldsymbol{X}(0)$;
- Generate the augmented $(D_E * D_y)$-dimensional vectors S and Y (the states in S are the ensemble means);

- After some maths, the pseudoinverse can be calculated as:

$$\frac{\partial S(x(t))}{\partial x(t)}^{\dagger} = \boldsymbol{X}(0) \begin{pmatrix} (H\boldsymbol{X}(0)) \\ (H\boldsymbol{X}(\tau)) \\ \vdots \\ (H\boldsymbol{X}((D_E - 1)\tau)) \end{pmatrix}^{\dagger}$$

We need the pseudoinverse of a $(D_E * D_y) \times Nens$ ensemble perturbation matrix (also via an SVD).

- Computational gain: $D_x/Nens$

- Localisation can be applied to this matrix

(Pinheiro et al. 2018)

Experimental: Setup

- Twin experiments using a chaotic Lorenz96 model with **1000** variables

$$\frac{dx_a}{dt} = (x_{a+1} - x_{a-2})x_{a-1} - x_a + F \tag{4}$$

- **25%** of the system is observed (equally distributed on the Lorenz ring) **at every 10 time steps**
- To update the variables at **unobserved** time steps, we use a progressive strength factor multiplied to the coupling term that was previously computed
- Observation noise with standard deviation $\sigma_{obs} = 0.1$
- $\Delta t = 0.01$ and constant time interval $\tau = 10\Delta t$

Ensemble-based synchronisation

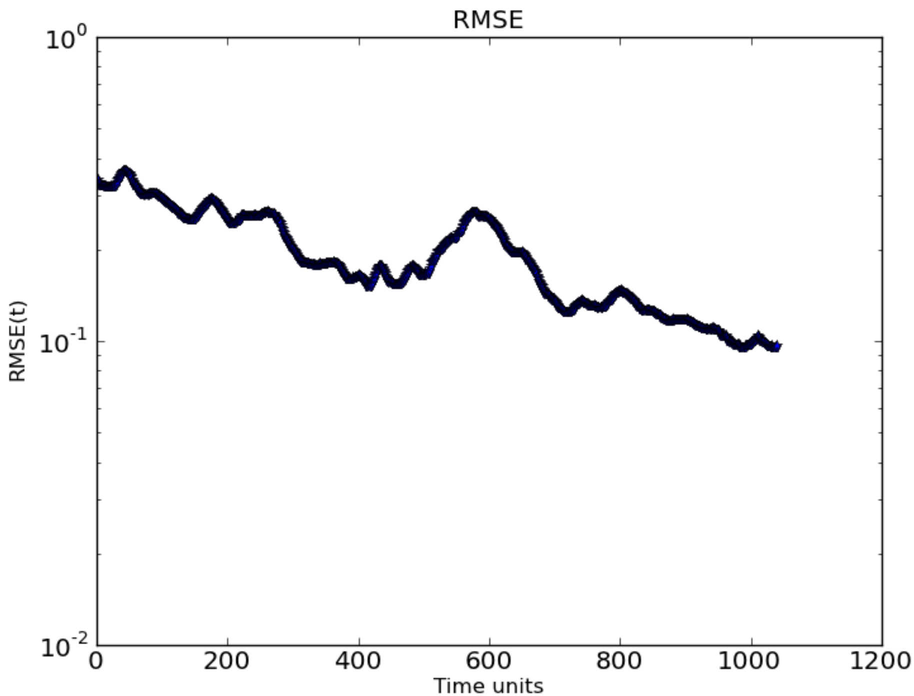


Figure 2: RMSE for $D_x = 1000$, $Nens = 30$, localisation radius $= 10$. ($D_E = 5$)

Ensemble-based synchronisation

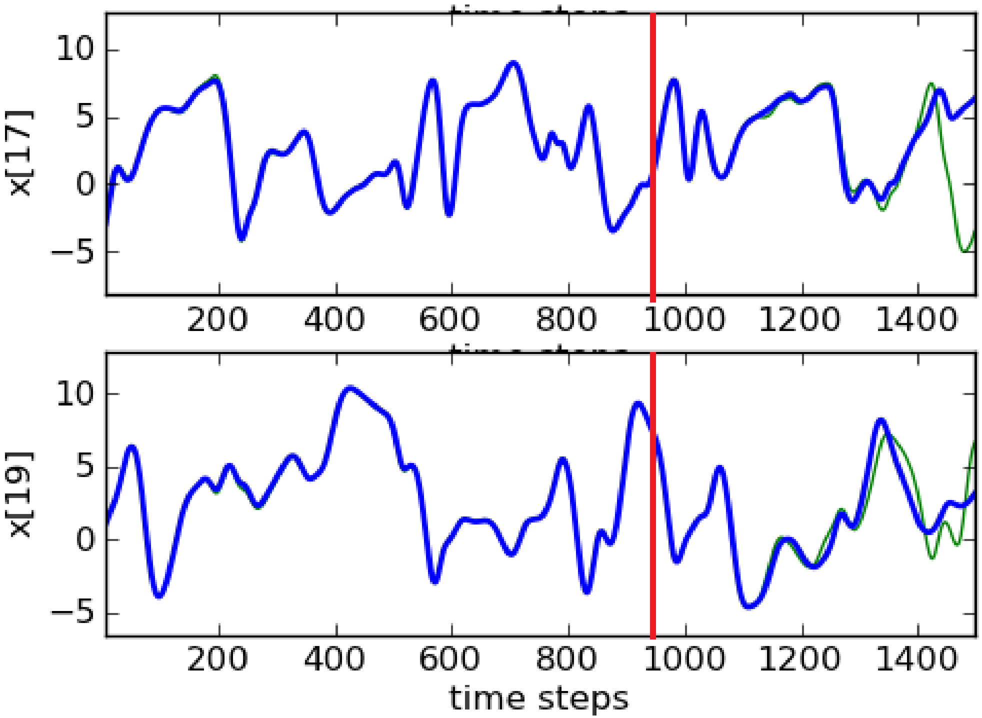


Figure 3: Two **unobserved** variables, $D_x = 1000$ (predictions after red lines).

Main idea

Combine the **EnSynch** with the **IEWPF** (Implicit Equal-Weights Particle Filter - Zhu et al. 2016)

Proposal Density

The pdf at time n can be written as:

$$\begin{aligned} p(x^n) &= \int p(x^n, x^{n-1})\, dx^{n-1} \\ &= \int p(x^n \mid x^{n-1}) p(x^{n-1})\, dx^{n-1} \end{aligned} \tag{5}$$

where $p(x^n \mid x^{n-1})$ is the transition density of the original model. We can introduce a proposal density $\mathbf{q}$ as follows:

$$p(x^n) = \int \frac{p(x^n \mid x^{n-1})}{\boldsymbol{q(x^n \mid x^{n-1}, y^n)}} \boldsymbol{q(x^n \mid x^{n-1}, y^n)} p(x^{n-1})\, dx^{n-1} \tag{6}$$

Proposal Density

The transition density is related to the original model via:

$$\boldsymbol{p}(\boldsymbol{x}^{\boldsymbol{n}} \mid \boldsymbol{x}^{\boldsymbol{n-1}}): \quad x_i^n = f(x_i^{n-1}) + \beta_i^n \tag{7}$$

and the proposal density to our proposed model:

$$\boldsymbol{q}(\boldsymbol{x}^{\boldsymbol{n}} \mid \boldsymbol{x}^{\boldsymbol{n-1}}, \boldsymbol{Y}(\boldsymbol{t})): \; x_i^n = f(x_i^{n-1}) + g\frac{\partial S(x(t))}{\partial x(t)}^{\dagger}(Y(t) - S(t)) + \beta_i^n \tag{8}$$

This change in model equation is compensated by an extra weight:

$$\boldsymbol{w_i} = \frac{\boldsymbol{p}(\boldsymbol{x_i^n} \mid \boldsymbol{x_i^{n-1}})}{\boldsymbol{q}(\boldsymbol{x_i^n} \mid \boldsymbol{x_i^{n-1}}, \boldsymbol{Y}(\boldsymbol{t}))} \tag{9}$$

Ensynch + IEWPF - Lorenz96 model

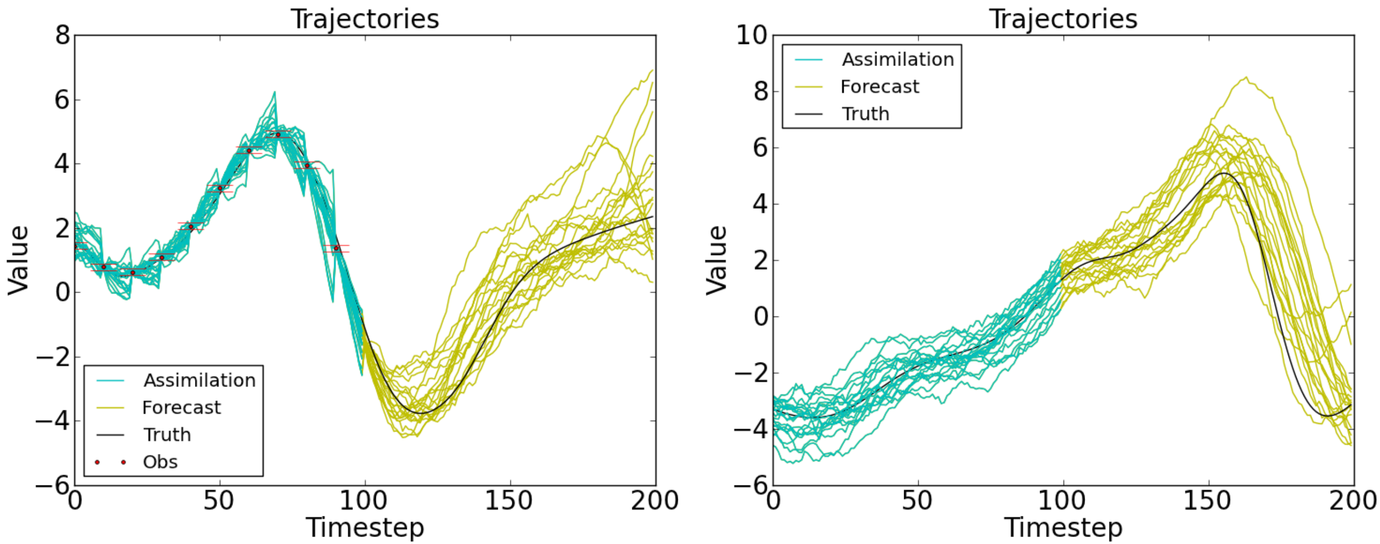


Figure 4: Observed (left) and unobserved (right) grid points. Observations occur at every 10 time steps. (Lorenz96 model for $D_x = 1000$, $D_y = 250$ and $Nens = 20$).

Ensynch + IEWPF - Lorenz96 model

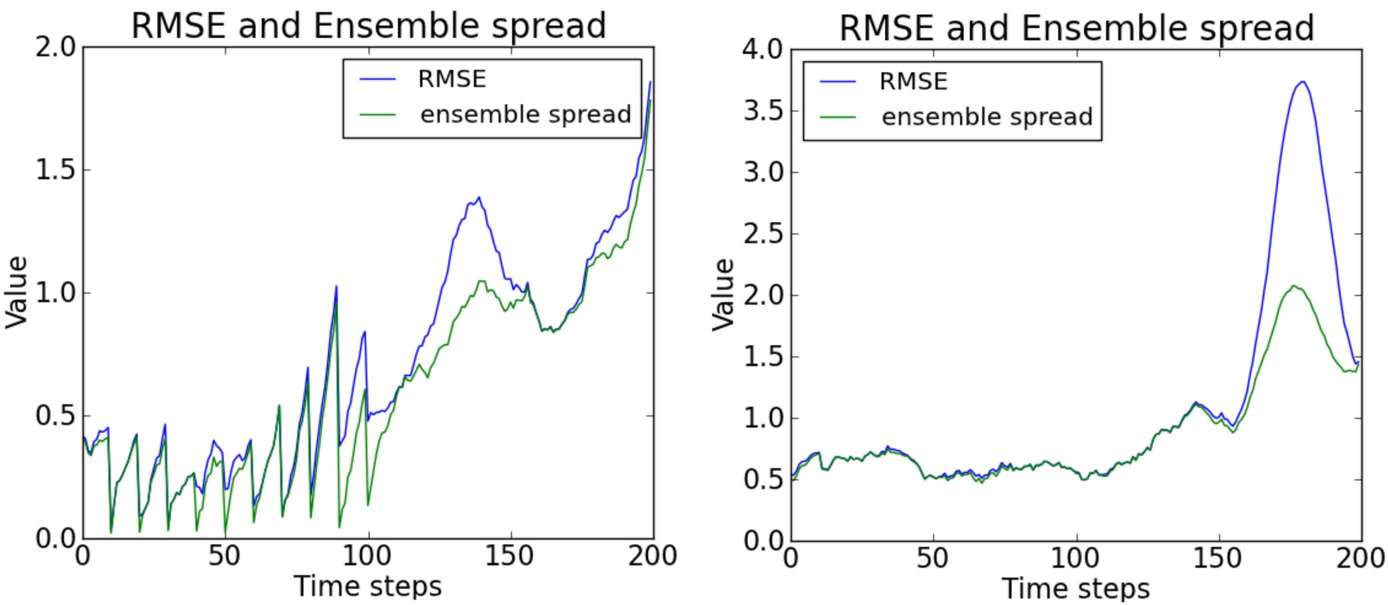


Figure 5: Observed (left) and unobserved (right) grid points. Observations occur at every 10 time steps. (Lorenz96 model for $D_x = 1000$, $D_y = 250$ and $Nens = 20$).

Ensynch + IEWPF - Lorenz96 model

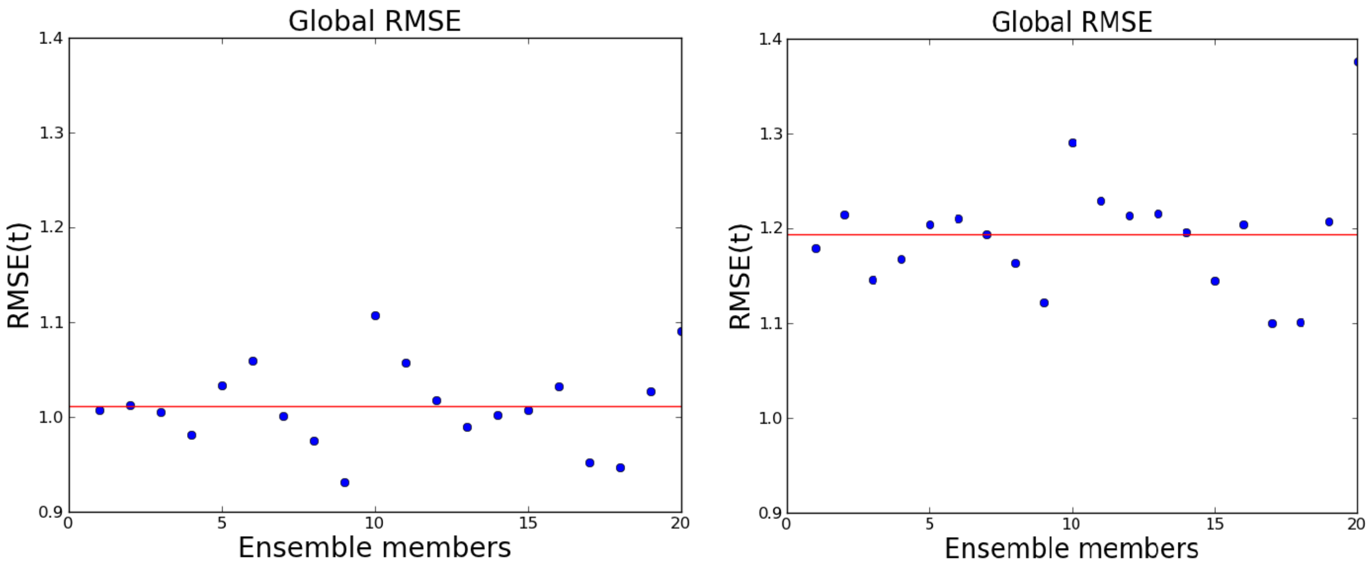


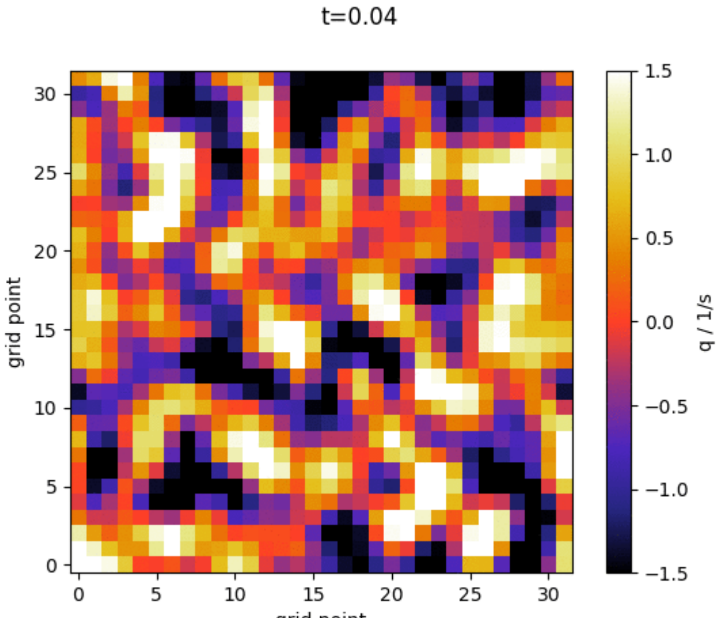
Figure 6: Global RMSEs averaged over the estimation period for each ensemble member for the IEWPF with different relaxation terms: EnSynch (left) and simple nudging (right). The red line is the ensemble mean.

Ensynch + IEWPF - Barotropic Vorticity model

- two-dimensional, turbulent flow
- stochastic forcing $\mathrm{d}\beta$

$$\frac{\partial q}{\partial t} = u\frac{\partial q}{\partial x} + v\frac{\partial q}{\partial y} + \mathrm{d}\beta$$

$$q = \nabla \times \mathbf{u} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$



Ensynch + IEWPF - Barotropic Vorticity model

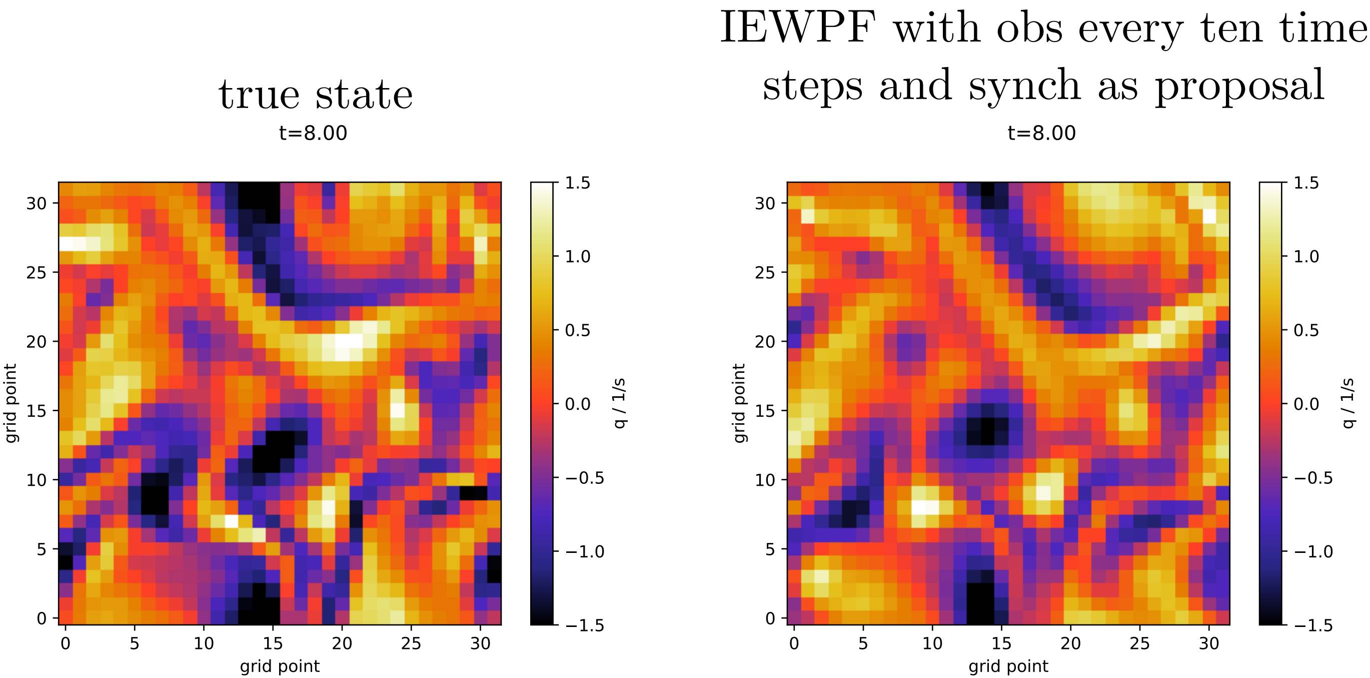


Figure 7: Observation at every second grid point in x and y direction, $D_x = 2^{10}$, $D_y = 2^8$, $Nens = 20$

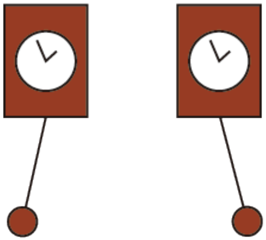
Conclusions and Discussions

- An efficient **ensemble-based synchronisation** scheme is proposed, opening up synchronisation to high-dimensional systems.
- The time-embedding concept allows to increase the **observability** of the system, meaning that less observations are needed, while still synchronising with the truth.
- These preliminary results suggest that the combination between the **EnSynch** scheme and the **IEWPF** leads to an effective fully nonlinear data-assimilation method.

References

- Rey, D., M. Eldridge, M. Kostuk, H. Abarbanel, J. Schumann-Bischoff, and U. Parlitz, 2014a: *Accurate state and parameter estimation in nonlinear systems with sparse observations*. Physics Letters A, 378, 869–873, doi:10.1016/j.physleta.2014.01.027.
- Zhu, M., P. J. van Leeuwen, and J. Amezcua, 2016: Implicit equal-weights particle filter. Quart. J. Roy. Meteorol. Soc., 142, 1904–1919, doi:10.1002/qj.2784.
- Pinheiro, F. R., P. J. van Leeuwen, and U. Parlitz, 2018: An ensemble framework for time delay synchronization. Quart. J. Roy. Meteorol. Soc., doi.org/10.1002/qj.3204.

THANKS FOR YOUR ATTENTION!



To calculate $\frac{\partial S(x(t))}{\partial x(t)}$ we note that, approximately:

$$H\boldsymbol{X}(\tau) \approx H\boldsymbol{F}(x)_{0\to\tau}\boldsymbol{X}(0) \tag{10}$$

This allows us to compute the Jacobian $\boldsymbol{F}(x)_{0\to\tau}$ as:

$$H\boldsymbol{F}(x)_{0\to\tau} = H\boldsymbol{X}(\tau)(\boldsymbol{X}(0))^{\dagger} \tag{11}$$

The full Jacobian matrix can be constructed as:

$$\frac{\partial S(x(t))}{\partial x(t)} = \begin{pmatrix} H \\ H\boldsymbol{F}(x)_{0\to\tau} \\ \vdots \\ H\boldsymbol{F}(x)_{0\to(D_E-1)\tau} \end{pmatrix} = \begin{pmatrix} H\boldsymbol{X}(0)(\boldsymbol{X}(0))^{\dagger} \\ H\boldsymbol{X}(\tau)(\boldsymbol{X}(0))^{\dagger} \\ \vdots \\ H\boldsymbol{X}((D_E-1)\tau)(\boldsymbol{X}(0))^{\dagger} \end{pmatrix}$$