Nonlinear data assimilation using synchronisation in a particle filter

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

¹University of Reading

²National 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](image)

**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

→ High-dimension nonlinear DA

→ Particle Filters

→ Proposal density freedom

Synchronisation
Synchronisation framework

\[
\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t)) + g \frac{\partial S(x(t))}{\partial x(t)} (Y(t) - S(t))
\]  

(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

\[
\frac{dx(t)}{dt} = f(x(t)) + g \frac{\partial S(x(t))}{\partial x(t)} \dagger (Y(t) - S(t)) \tag{1}
\]

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

- **Time embeddings** \((D_E)\):

  \[
  Y : \quad y(t) \quad y(t + \tau) \quad y(t + 2\tau) \quad \cdots \quad y(t + (D_E - 1)\tau)
  \]

  \[
  S : \quad H(x(t)) \quad H(x(t + \tau)) \quad H(x(t + 2\tau)) \quad \cdots \quad H(x(t + (D_E - 1)\tau))
  \]

  (both vectors \( \in \mathbb{R}^{D_E \times D_y} \))
Synchronisation

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

- **$\frac{\partial S(x(t))}{\partial x(t)}$ $\dagger$ → 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-based synchronisation

Ensemble synchronisation framework

- Generate an ensemble of $i$ initial states;

- Form the initial ensemble perturbation matrix:
  \[
  X(0)_i = x(0)_i - \bar{x}(0)
  \] (2)

  a $D_x \times Nens$ matrix;

- Propagate forward in time each ensemble member for $\tau$ time steps and form the ensemble perturbation matrix:
  \[
  X(\tau)_i = x(\tau)_i - \bar{x}(\tau)
  \] (3)

  with the same dimension as $X(0)$;

- Generate the augmented $(D_E \times D_y)$-dimensional vectors $S$ and $Y$ (the states in $S$ are the ensemble means);
Ensemble-based synchronisation

- Jacobian matrix \( \frac{\partial S(x(t))}{\partial x(t)} \): of size \((D_E \times D_y) \times Nens\).
Jacobian matrix $\frac{\partial S(x(t))}{\partial x(t)}$: of size $(D_E \times D_y) \times Nens$.

After some maths, its pseudoinverse can be calculated as:

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

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

- **Computational gain:** $D_x/Nens$

- Localisation can be applied to this matrix
Twin experiments using a chaotic Lorenz96 model with 1000 variables

\[
\frac{dx_a}{dt} = (x_{a+1} - x_{a-2})x_{a-1} - x_a + F
\]  

(4)

- 25% of the system is observed (equally distributed on the Lorenz ring) at every other time step
- At unobserved time steps, we reuse the coupling term previously computed to update the variables
- 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 $D_x = 1000$, $N_{ens} = 20$, 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:

$$p(x^n) = \int p(x^n, x^{n-1}) dx^{n-1}$$

$$= \int p(x^n | x^{n-1}) p(x^{n-1}) dx^{n-1}$$

(5)

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

$$p(x^n) = \int \frac{p(x^n | x^{n-1})}{q(x^n | x^{n-1}, y^n)} q(x^n | x^{n-1}, y^n) p(x^{n-1}) dx^{n-1}$$

(6)
Formulation

Proposal Density

The transition density is related to the original model via:

\[ p(x^n | x^{n-1}) : x^n_i = f(x^{n-1}_i) + \beta^n_i \quad (7) \]

and the proposal density to our proposed model:

\[ q(x^n | x^{n-1}, Y(t)) : x^n_i = f(x^{n-1}_i) + g \frac{\partial S(x(t))}{\partial x(t)} \dagger (Y(t) - S(t)) + \beta^n_i \quad (8) \]

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

\[ w_i = \frac{p(x^n_i | x^{n-1}_i)}{q(x^n_i | x^{n-1}_i, Y(t))} \quad (9) \]
Results

**Ensynch + IEWPF**

![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 $N_{ens} = 20$).](image)
Results

Ensynch + IEWPF

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 $Nens = 20$).
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


THANKS FOR YOUR ATTENTION!
To calculate $\frac{\partial S(x(t))}{\partial x(t)}$ we note that, approximately:

$$HX(\tau) \approx HF(x)_{0 \rightarrow \tau}X(0) \tag{10}$$

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

$$HF(x)_{0 \rightarrow \tau} = HX(\tau)(X(0))^\dagger \tag{11}$$

The full Jacobian matrix can be constructed as:

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