Efficient nonlinear data assimilation using synchronisation in a particle filter

Received: 06 June 2018 | Revised: 15 April 2019 | Accepted: 23 May 2019 | Published on: 27 June 2019

DOI: 10.1002/qj.3576

RESEARCH ARTICLE

Quarterly Journal of the Royal Meteorological Society RMetS

Efficient nonlinear data assimilation using synchronization in a particle filter

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

[1]Department of Meteorology, University of Reading, UK

[2]National Centre for Earth Observation, University of Reading, UK

Correspondence
Peter J. van Leeuwen, Department of Meteorology, University of Reading, Earley Gate, PO Box 243, Reading RG6 6BB, UK.
Email: p.j.vanleeuwen@reading.ac.uk

Funding information
CNPq/Brazil (National Council for Scientific and Technological Development), the Brazilian Navy and the European Research Council project CUNDA under the European Union Horizon 2020 programme.;

Abstract

Current data assimilation methods still face problems in strongly nonlinear cases. A promising solution is a particle filter, which provides a representation of the state probability density function (pdf) by a discrete set of particles. To allow a particle filter to work in high-dimensional systems, the proposal density freedom is explored. We used a proposal density from synchronization theory, in which one tries to synchronize the model with the true evolution of a system using one-way coupling, via the observations. This is done by adding an extra term to the model equations which will control the growth of instabilities transversal to the synchronization manifold. In this paper, an efficient ensemble-based synchronization scheme is used as a proposal density in the implicit equal-weights particle filter, a particle filter that avoids filter degeneracy by construction. Tests using the Lorenz96 model for a 1,000-dimensional system show successful results, where particles efficiently follow the truth, both for observed and unobserved variables. These first tests show that the new method is comparable to, and slightly outperforms, a well-tuned Local Ensemble Transform Kalman Filter. This methodology is a promising solution for high-dimensional nonlinear problems in the geosciences, such as numerical weather prediction.

KEYWORDS

data assimilation, ensemble, synchronization, nonlinear, particle filter

1 | INTRODUCTION

Synchronization is a phenomenon first described by the Dutch scientist Christiaan Huygens. He found that two pendulum clocks suspended by a common frame would oscillate, after some time, in opposite directions but in perfect consonance, independent of their initial phases. He concluded that the clocks were able to synchronize their phases due to the frame they were sharing, a common factor between both. Nowadays, based on these ideas, researchers in this field focus on exploring the synchronization of two or more systems while they share common information. Typically, this is obtained by a suitable relaxation term that forces the model (or models) towards the observations and helps to control the stability of the synchronization manifold.

There is a close connection between the concepts of synchronization and data assimilation, as the latter aims to synchronize the model evolution with the truth, via the observations, finding the best state estimate and its uncertainty. In this case, the information to be shared between the systems flows from the truth to the model, in a unidirectional way, through noisy and sparse observations.

Through the light of Bayes' Theorem, as explained in (e.g.) van Leeuwen *et al.* (2015), data assimilation is actually a multiplication problem rather than an inverse problem. Indeed, one tries to represent a posterior probability density function (pdf) by multiplying the information contained in the model prior with the likelihood, which is updated when observations are available. Data assimilation encompasses a more general concept than synchronization.

Similarities can be found between synchronization and data assimilation, as pointed out by Yang *et al.* (1996) and Duane *et al.* (2006) and more recently by Pinheiro *et al.* (2018), where a discussion can be found on the relations and differences between time-delayed synchronization and the Kalman smoother. A crucial difference is that the synchronization framework allows one to reuse the observations, which is a complication in many data assimilation methods.

Recent work in the synchronization field has explored the time-embedding concept, by considering observations either from the past or future (e.g. Rey *et al.*, 2014a, 2014b; Parlitz *et al.*, 2014; Pazó *et al.*, 2016). Based on these previous works, Pinheiro *et al.* (2018) have focused on constructing a more robust synchronization scheme that could potentially be applied to a realistic geophysical model, in combination with a particle filter. To this end, they proposed an ensemble-based synchronization framework (EnSynch) that, with a small number of ensemble members, provided promising results for a high-dimensional system. As the authors mention in their study, their ensemble methodology cannot be used as a stand-alone data assimilation method, as no uncertainties are computed and also because observations inside the time window are reused in different time steps, which could bring complications in most data assimilation methods. The authors also discussed the similarities and differences between the ensemble synchronization and methods like the ensemble smoother and also variational methods that employ ensembles to avoid adjoint calculations like, for example, the four-dimensional ensemble-variational schemes (4DEnsVar) (e.g. Bannister, 2017 and section 4.5 in Carrassi *et al.*, 2018) and iterative ensemble smoothers like the iterative ensemble Kalman smoother (IEnKS) of Bocquet and Sakov (2014). Compared to the ensemble smoother, one of the main differences is that the ensemble synchronization ignores the observation-error covariance matrix R in the gain. This possible weakness in the scheme is counterbalanced by the inclusion of a tuning factor in the synchronization formulation: the constant g. This constant regulates the strength of the influence that the pseudoinverse of the Jacobian of the operator S (an embedding map, which contains the dynamical model and the observation operator) will have on the innovation. Another difference is that synchronization allows observations to be reused along the iterations in the time-embedded intervals, so they can increase the observability of the system. This is an interesting feature of synchronization.

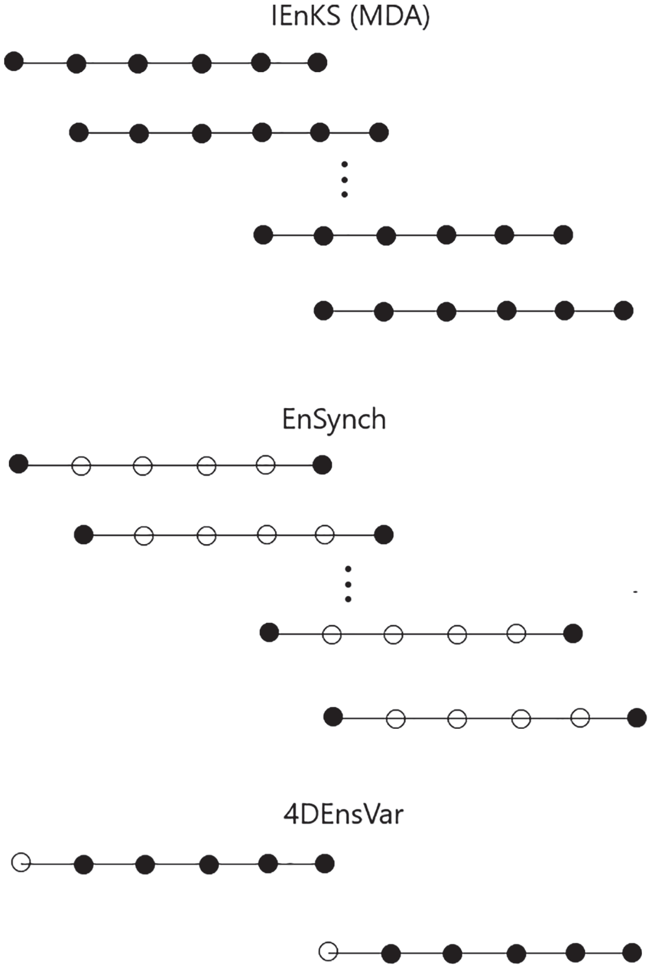

FIGURE 1 Use of observations within the assimilation window for different methodologies: IEnKs MDA, EnSynch and 4DEnsVar. The black dots are the observations which are used in the cycle. The IEnKS uses observations several times, but with increased observation errors, the EnSynch uses observations several times, and 4DEnsVar uses each observation only once (see text for details)

Note that the IEnKS also uses observations several times, but with a factorization of the likelihoods, which does not change the observability of the system.

Figure 1 shows how observations are used inside a data assimilation window for the IEnKS MDA (Multiple Data Assimilation window), the EnSynch and the 4DEnsVar. In this illustration, observations exist at every time step, but only the black ones are effectively used. Note that in the first two cases the assimilation windows overlap and so the observations are re-used. In the quasi-static IEnKS MDA case, the observations are assimilated several times, but they are weighted within the window as: $1 = \sum_{k=1}^{L} \beta_k$, where L is the length of the assimilation window and $0 < \beta_k < 1$ is the extra weight each observation at time k receives within this window.

In this work we will merge the efficiency of synchronization described in Pinheiro *et al.* (2018) with the data assimilation formalism. To this end, we will use synchronization indirectly, as a proposal density, in a nonlinear data assimilation methodology: a particle filter.

It is well known that one can explore the proposal density freedom by modifying the model equations (e.g. Doucet *et al.*, 2001; van Leeuwen *et al.*, 2015). In this concept, extra terms can be included in the model equations, even counting on tunable parameters. The main goal is to steer the particles towards the high-probability region of the posterior pdf between observation time steps. Some interesting experiments can be found in the literature on tests with different proposal densities. An ensemble Kalman filter has been used as a proposal density in a particle filter by Papadakis *et al.* (2010). Further tests with simple relaxation terms have been performed by van Leeuwen (2010), by Ades and van Leeuwen (2015) in a high-dimensional system and by Browne and van Leeuwen (2015) in a climate model. The latter concluded that the relaxation term used was not optimal, as the particles were spreading away from the truth between observation time steps. In Browne (2016), the use of an ensemble Kalman smoother as a proposal was still not effective. Therefore, the main motivation of this paper is to investigate how effective synchronization is as a proposal density in a particle filter.

One of the main issues of particle filters is that ensembles of particles tend to collapse in high-dimensional systems (also known as the curse of dimensionality or degeneracy). The filter chosen for our experiments is based on the implicit equal-weights particle filter (IEWPF; Zhu *et al.*, 2016), but with the extension advocated in Skauvold *et al.* (2019). This filter does not suffer from degeneracy by construction and allows for unbiased mean and covariance.

The paper is structured as follows: Section 2 describes the methodology used to construct the ensemble-based synchronization framework, gives a summary of the IEWPF formulation and details the implementation of the ensemble synchronization as a proposal in this particle filter. Section 3 presents results on the experiments using the Lorenz96 model (Lorenz, 1995) in a 1,000-dimensional system. In Section 4, we discuss the results and present conclusions.

2 | METHODOLOGY

2.1 | The ensemble-based synchronization (EnSynch)

Pinheiro *et al.* (2018) have developed an ensemble-synchronization scheme that can be applied to realistic high-dimensional geophysical models. This methodology is an ensemble-based time-embedding scheme, similar to an ensemble smoother or a 4DEnsVar, which avoids the need for tangent-linear models and adjoint calculations.

The scheme explores the use of time embeddings, in which a delay dimension D_{d} is defined in order to capture information from D_y observations, which occur at every τ time steps within a time interval $[t, t + (D_{\mathrm{d}} - 1)\tau\Delta t]$, where τ is a constant. Observations are then included, so they can bring additional information back to the present time. Pinheiro *et al.* (2018) show that, after stabilizing the synchronization manifold, which can be briefly described as a set of subspaces which states are attracted to, very precise estimates are obtained, both for observed and unobserved variables, allowing accurate predictions over a significant forecast period.

The ensemble methodology detailed in their paper is summarized here. Define a true state $\mathbf{x}_{\mathrm{T}} \in \Re^{D_x}$ in the evolution equation of the true system:

$$\mathbf{x}_{\mathrm{T}}^{j+1} = f(\mathbf{x}_{\mathrm{T}}^{j}) + \boldsymbol{\beta}_{\mathrm{T}}^{j+1}, \tag{1}$$

where f is the nonlinear model at time j and $\boldsymbol{\beta}_{\mathrm{T}}^{j+1}$ is a stochastic perturbation represented by a normal distribution of $N \sim (0, 0.01)$. Compute the evolution of a state estimate $\mathbf{x}^j$ with time using the coupled dynamics and the perfect model f:

$$\mathbf{x}^{j+1} = f(\mathbf{x}^j) + g\, n\, \frac{\partial \mathbf{S}(\mathbf{x}^j)}{\partial \mathbf{x}^j}^{\dagger} \{\mathbf{Y}^j - \mathbf{S}(\mathbf{x}^j)\} + \boldsymbol{\beta}_{\mathrm{T}}^{j+1}, \tag{2}$$

where we include the augmented D_{e}-dimensional vectors $\mathbf{S}(\mathbf{x}^j)$ and $\mathbf{Y}^j$. The embedding dimension D_{e} is defined by $D_{\mathrm{e}} = D_{\mathrm{d}}\, D_y$. The vector $\mathbf{S} \in \Re^{D_{\mathrm{e}}}$ is related to the states of the model and $\mathbf{Y} \in \Re^{D_{\mathrm{e}}}$ is related to the observations $\mathbf{y} \in \Re^{D_y}$, as:

$$\mathbf{S}(\mathbf{x}^j) = \left[(H\{\mathbf{x}^j\})^{\mathrm{T}}, \{H(\mathbf{x}^{j+\tau})\}^{\mathrm{T}}, \ldots, \{H(\mathbf{x}^{j+(D_{\mathrm{d}}-1)\tau})\}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3}$$

and

$$\mathbf{Y}^j = \left[(\mathbf{y}^j)^{\mathrm{T}}, (\mathbf{y}^{j+\tau})^{\mathrm{T}}, \ldots, (\mathbf{y}^{j+(D_{\mathrm{d}}-1)\tau})^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{4}$$

The states $\mathbf{x}^j$ in $\mathbf{S}(\mathbf{x}^j)$ are the means of the ensemble of states $\mathbf{x}_i^j$, where i is the ensemble index. $\mathbf{S}(\cdot)$ is a map from physical to an embedding space. H is the observation operator, which can be linear or nonlinear. It is assumed that H does not change with time, but it is straightforward to make it time-dependent. The coupling constant g determines the strength of the coupling, which is progressively increased by n along the interval between observations. At an observed time step, when we update the variables, $n = 1$. During the time steps without observations, n is increased from 1 to $(o_{\mathrm{int}} - 1)$ at the last unobserved time step in the interval, where o_{int} is the observation interval.

$\partial \mathbf{S}(\mathbf{x}^j)/\partial \mathbf{x}^j$ is the Jacobian matrix of $\mathbf{S}$. Its pseudoinverse, $(\partial \mathbf{S}(\mathbf{x}^j)/\partial \mathbf{x}^j)^{\dagger}$, is responsible for spreading the information from the observed to the unobserved variables. It is computed as follows:

$$\frac{\partial \mathbf{S}(\mathbf{x}^j)}{\partial \mathbf{x}^j}^{\dagger} \approx \mathbf{X}^j \begin{pmatrix} H(\mathbf{X}^j) \\ H(\mathbf{X}^{j+\tau}) \\ \vdots \\ H(\mathbf{X}^{j+(D_{\mathrm{d}}-1)\tau}) \end{pmatrix}^{\dagger}, \tag{5}$$

where the pseudoinverse on the right-hand side is computed via a truncated SVD. $\mathbf{X}^j$ is the ensemble perturbation matrix, that is, each column is the difference between each member and the mean.

Additionally, the ensemble synchronization scheme uses a localization method to reduce the influence of observations which are far away from the variables (Houtekamer and Mitchell, 1998). Pinheiro *et al.* (2018) give an overview of the localization implementation in this scheme.

When implemented in the particle filter, the scheme of Pinheiro *et al.* (2018) is adapted, to reduce its computational cost and also to fit it into the particle filtering framework, as follows.

In order to avoid calculating the forecast ensembles several times, we store the initial set of ensemble perturbation matrices $\mathbf{X}$ computed at time step j, so we have:

$$\mathbf{X} = (\mathbf{X}^j, \mathbf{X}^{j+\tau}, ..., \mathbf{X}^{j+(D_\mathrm{d}-1)\tau}). \tag{6}$$

Note that these perturbation matrices are obtained while constructing the Jacobian $\partial\mathbf{S}(\mathbf{x}^j)/\partial\mathbf{x}^j$. We also store the ensemble means at each of the embedding times described in Equation (6).

At the next time steps in which we have observations, we perform the following procedures:

(1) Rescale all perturbation matrices in $\mathbf{X}$ with a factor:

$$\gamma_k = \frac{\sigma_{k(\mathrm{new})}}{\sigma_{k(\mathrm{prior})}}, \tag{7}$$

where k here corresponds to the state variables and

$$\sigma_k = \sqrt{\frac{1}{N_\mathrm{ens}-1}\sum_{i=1}^{N_\mathrm{ens}}(x_i^k - \bar{x}^k)^2} \tag{8}$$

is the formulation used to compute the spread. The $\sigma_{k(\mathrm{prior})}$, then, is the ensemble spread for the actual time step propagated from the previous observation time, i.e. the prior. $\sigma_{k(\mathrm{new})}$ is the spread of the perturbation matrix newly computed at the actual time step, after using Equation (2).

(2) Recentre the ensembles at each observed time, by adding to the means previously stored, the difference between the mean of the new ensemble and the prior ensemble.

(3) Propagate the last rescaled and recentred ensemble, that is, the one at $t + (D_\mathrm{d} - 1)\tau\Delta t$ for an additional τ time steps, to have a complete $\mathbf{X}$, as needed in Equation (2).

2.2 | Particle filters

Particle filters are based on Bayes' Theorem, in which the posterior pdf of the state $\mathbf{x}^n$, given the observations $\mathbf{y}^n$ at time n is:

$$p(\mathbf{x}^n \mid \mathbf{y}^n) = \frac{p(\mathbf{y}^n \mid \mathbf{x}^n)p(\mathbf{x}^n)}{p(\mathbf{y}^n)}. \tag{9}$$

The basic idea is to represent the prior pdf, a probability density that contains information from the model, as a discrete set of model states or particles $\mathbf{x}_i$:

$$p(\mathbf{x}^n) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}^n - \mathbf{x}_i^n). \tag{10}$$

This approximation, combined with Bayes' formulation, leads to the following representation of the posterior:

$$p(\mathbf{x}^n \mid \mathbf{y}^n) \approx \sum_{i=1}^{N} w_i^n\delta(\mathbf{x}^n - \mathbf{x}_i^n), \tag{11}$$

in which w_i^n are the weights:

$$w_i^n = \frac{p(\mathbf{y}^n \mid \mathbf{x}_i^n)}{\sum_{k=1}^{N} p(\mathbf{y}^n \mid \mathbf{x}_k^n)}. \tag{12}$$

These weights are attached to each particle to give them a relative importance, according to how close or distant the model states are from the observations. If only few particles are close to the observations and receive non-zero weights, the posterior pdf will lose its full statistical information, as the other particles will be discarded, due to their inability to follow the observations. This is called filter degeneracy or filter collapse and it happens in even small-dimensional systems. Doucet *et al.* (2001) gives a more detailed mathematical explanation on particle filters.

Several methods have been developed to mitigate this problem. Snyder *et al.* (2008; 2015) proved that resampling would not solve the degeneracy, even using the so-called optimal proposal density.

To avoid the particles becoming degenerate, it is crucial to ensure that equally significant particles are drawn from the posterior density. To do this, we must perform two different and important steps: (a) guarantee that the particles are located at the high-probability regions of the posterior and (b) ensure similar or equal weights of the particles, so they will never collapse. To this end, several methods have been recently developed, such as the Equivalent Weights Particle Filter (EWPF; van Leeuwen, 2010) and the IEWPF (Zhu *et al.*, 2016; Skauvold *et al.*, 2019).

Regarding the first step, we need a scheme that pulls the particles towards the observations, between observation time steps. An interesting property of particle filters that can be explored to this purpose is the so-called proposal transition density.

Consider time steps j and $j+1$ between two observation time steps $n-1$ and n, that is, $t_{n-1} < t_j < t_{j+1} < t_n$. The model is assumed to be Markovian. This way, the pdf $p(\mathbf{x}^{j+1})$ can be written as:

$$\begin{aligned} p(\mathbf{x}^{j+1}) &= \int p(\mathbf{x}^{j+1}, \mathbf{x}^{j})\, \mathrm{d}\mathbf{x}^{j} \\ &= \int p(\mathbf{x}^{j+1} \mid \mathbf{x}^{j}) p(\mathbf{x}^{j})\, \mathrm{d}\mathbf{x}^{j}, \end{aligned} \tag{13}$$

where $p(\mathbf{x}^{j+1} \mid \mathbf{x}^{j})$ is the transition density of the original model. We have the freedom to introduce a proposal density q as follows, without changing the previous equation:

$$p(\mathbf{x}^{j+1}) = \int \frac{p(\mathbf{x}^{j+1} \mid \mathbf{x}^{j})}{q(\mathbf{x}^{j+1} \mid \mathbf{x}^{j}, \mathbf{y}^{n})} q(\mathbf{x}^{j+1} \mid \mathbf{x}^{j}, \mathbf{y}^{n})\, p(\mathbf{x}^{j})\, \mathrm{d}\mathbf{x}^{j}. \tag{14}$$

This is possible if the support of q is equal to or larger than the one in p. This way, we avoid dividing the previous equation by zero.

Instead of drawing from $p(\mathbf{x}^{j+1} \mid \mathbf{x}^{j})$ and so using the original model, we now draw from $q(\mathbf{x}^{j+1} \mid \mathbf{x}^{j}, \mathbf{y}^{n})$. This way, the prior at time n is now defined as:

$$p(\mathbf{x}^{n}) = \sum_{i=1}^{N} \widehat{w_i}^{\,n} \delta(\mathbf{x}^{n} - \mathbf{x}_i^{n}), \tag{15}$$

in which the weights are accumulated during the time steps between observations until time n:

$$\widehat{w_i}^{\,n} = \prod_{j=1}^{n} \frac{p(\mathbf{x}_i^{j+1} \mid \mathbf{x}_i^{j})}{q(\mathbf{x}_i^{j+1} \mid \mathbf{x}_i^{j}, \mathbf{y}^{n})}. \tag{16}$$

It is possible to choose any proposal transition density for $q(\mathbf{x}^{j+1} \mid \mathbf{x}^{j}, \mathbf{y}^{n})$, so the aim is to use one that includes additional information from the observations in the future, in an optimal way.

2.3 | Ensemble synchronization as a proposal density in the IEWPF

Considering the proposal density freedom, several methods can be used for this, including traditional methods such as the four-dimensional variational schemes (4D-Var), as essentially proposed in Chorin and Tu (2009), and the ensemble smoothers, like in Browne (2016). However, these are rather expensive, as it usually involves solving a problem similar to 4D-Var for each particle. Thus, typically, simpler schemes are employed between observation times. These schemes will be less efficient, although we can ensure that Bayes' Theorem is fulfilled exactly for each particle.

Some authors (e.g. Zhu *et al.*, 2016) implemented the particle filter with a weak relaxation term between observations to control the spread and to achieve better-converging trajectories of the particles. To this end, the relaxation transition proposal density that appears in Equation (14) was defined by those authors as:

$$q(\mathbf{x}^{j+1} \mid \mathbf{x}_i^{j}, \mathbf{y}^{n}) \sim N(\mathbf{x}_{\mathrm{relax}}, \mathbf{Q}), \tag{17}$$

where $\mathbf{Q}$ is the model error covariance and

$$\mathbf{x}_{\mathrm{relax}} = f(\mathbf{x}_i^{j}) + B(\tau)[\mathbf{y}^{n} - H f(\mathbf{x}_i^{j})], \tag{18}$$

where $B(\tau)$ is the relaxation strength, given by

$$B(\tau) = b\, \tau\, Q H^{\mathrm{T}} \mathbf{R}^{-1}. \tag{19}$$

In this formulation, b is a constant, τ increases linearly from zero to one along the unobserved time steps and $\mathbf{R}$ is the observation-error covariance matrix.

They concluded from this formulation that the relaxation term used was a weakness in their scheme. They also argued that proposal densities can be more sophisticated than the one they have used and should be tested to improve the performance and increase robustness. That is exactly what we are investigating in this work.

We will assume that the model errors and the proposal are Gaussian. The transition density in Equation (13) for the prior is related to the original model via

$$p(\mathbf{x}^{j+1} \mid \mathbf{x}^{j}) : \quad \mathbf{x}_i^{j+1} = f(\mathbf{x}_i^{j}) + \boldsymbol{\beta}_i^{j+1}, \tag{20}$$

where $\boldsymbol{\beta}^{j+1}$ is a stochastic vector representing the model error $\boldsymbol{\beta}_i^{j+1} \sim N(0, \mathbf{Q})$. This leads to:

$$p(\mathbf{x}_i^{j+1} \mid \mathbf{x}_i^{j}) \sim N\{f(\mathbf{x}_i^{j}), \mathbf{Q}\}, \tag{21}$$

where

$$p(\mathbf{x}_i^{j+1} \mid \mathbf{x}_i^{j}) \propto \exp\left[-\frac{1}{2}\left\{\mathbf{x}_i^{j+1} - f(\mathbf{x}_i^{j})\right\}^{\mathrm{T}} \mathbf{Q}^{-1} \left\{\mathbf{x}_i^{j+1} - f(\mathbf{x}_i^{j})\right\}\right]. \tag{22}$$

The proposal density that appears in Equation (14) is related to a proposed model, in our case, the ensemble synchronization equation:

$$\mathbf{x}_i^{j+1} = f(\mathbf{x}_i^{j}) + g\, n \left(\frac{\partial \mathbf{S}(\mathbf{x}^{j})}{\partial \mathbf{x}^{j}}\right)^{\dagger} (\mathbf{Y}^{j} - \mathbf{S}^{j}) + \widehat{\boldsymbol{\beta}}_i^{j+1}, \tag{23}$$

where $\widehat{\boldsymbol{\beta}}_i^{j+1} \sim N(0, \widehat{\boldsymbol{Q}})$, which in our case is drawn from the same distribution as $\boldsymbol{\beta}_i^{j+1}$, so we assume here that $\widehat{\boldsymbol{Q}} = \mathbf{Q}$. The augmented observation and state vectors $\mathbf{Y}^{j}$ and $\mathbf{S}^{j}$ use future observations in the time-embedding window, described in Equations (4) and (3), respectively. Hence,

$$q(\mathbf{x}^{j+1} \mid \mathbf{x}_i^{j}, \mathbf{Y}^{j}) \sim N(\mathbf{x}_{\mathrm{synch}}, \mathbf{Q}), \tag{24}$$

where

$$\mathbf{x}_{\text{synch}} = f(\mathbf{x}_i^j) + g\, n \left(\frac{\partial \mathbf{S}(\mathbf{x}^j)}{\partial \mathbf{x}^j}\right)_i^{\dagger} (\mathbf{Y}^j - \mathbf{S}^j), \tag{25}$$

which is the deterministic part of the right-hand side of Equation (23). The distribution of this proposal density is calculated as:

$$q(\mathbf{x}^{j+1} \mid \mathbf{x}_i^j, \mathbf{Y}^j) \propto \exp\left[-\frac{1}{2}\left(\mathbf{x}_i^{j+1} - \mathbf{x}_{\text{synch}}\right)^{\mathrm{T}} \mathbf{Q}^{-1} \left(\mathbf{x}_i^{j+1} - \mathbf{x}_{\text{synch}}\right)\right], \tag{26}$$

which we can relate to the model error added in Equation (23) as

$$q(\mathbf{x}^{j+1} \mid \mathbf{x}_i^j, \mathbf{Y}^j) \propto \exp\left[-\frac{1}{2}\left(\widehat{\boldsymbol{\beta}}_i^{j+1}\right)^{\mathrm{T}} \mathbf{Q}^{-1} \widehat{\boldsymbol{\beta}}_i^{j+1}\right]. \tag{27}$$

We avoid computing $\mathbf{Q}^{-1}$ in Equation (27), as in Ades and van Leeuwen (2015), by noting that the error $\widehat{\boldsymbol{\beta}}_i^{j+1}$ is sampled from $N(0, \mathbf{Q})$ through $\widehat{\boldsymbol{\beta}}_i^{j+1} = \mathbf{Q}^{1/2} \widehat{\boldsymbol{\eta}}_i^{j+1}$, where $\widehat{\boldsymbol{\eta}}_i^{j+1} \sim N(0, I)$. This leads to

$$\left(\widehat{\boldsymbol{\beta}}_i^{j+1}\right)^{\mathrm{T}} \mathbf{Q}^{-1} \widehat{\boldsymbol{\beta}}_i^{j+1} = \left(\widehat{\boldsymbol{\eta}}_i^{j+1}\right)^{\mathrm{T}} \widehat{\boldsymbol{\eta}}_i^{j+1}. \tag{28}$$

This change in model equation is compensated by an extra weight, as explained in the previous section,

$$w_i = \frac{p(\mathbf{x}_i^{j+1} \mid \mathbf{x}_i^j)}{q(\mathbf{x}_i^{j+1} \mid \mathbf{x}_i^j, \mathbf{Y}^j)}, \tag{29}$$

which accumulates for all time steps, except the last time step before the next observation, to

$$w_i^{n-1} = \prod_{j=1}^{n-1} \frac{p(\mathbf{x}_i^{j+1} \mid \mathbf{x}_i^j)}{q(\mathbf{x}_i^{j+1} \mid \mathbf{x}_i^j, \mathbf{Y}^j)}. \tag{30}$$

At the last time step before the next observation, the IEWPF is applied, to ensure that all of the particles have equal weights in the posterior pdf.

A brief explanation of the modified IEWPF scheme (Skauvold *et al.*, 2019) is as follows. For each particle two samples are drawn from a Gaussian distributed proposal $q(\boldsymbol{\xi}, \boldsymbol{\eta}) = q(\boldsymbol{\xi}|\boldsymbol{\eta})q(\boldsymbol{\eta})$, such that $\boldsymbol{\xi}$ is perpendicular to $\boldsymbol{\eta}$. We then form a new particle as:

$$\mathbf{x}_i^n = \mathbf{x}_i^{\mathrm{a}} + \alpha_i \mathbf{P}^{1/2} \xi_i^n + \beta \mathbf{P}^{1/2} \boldsymbol{\eta}_i^n. \tag{31}$$

$\mathbf{x}_i^{\mathrm{a}}$ is the mode of $p(\mathbf{x}_i^n \mid \mathbf{x}_i^{n-1}, \mathbf{y}^n)$, α_i and β are scalars and $\mathbf{P}$ is an estimate of the covariance of $p(\mathbf{x}_i^n \mid \mathbf{x}_i^{n-1}, \mathbf{y}^n)$. One chooses β, which is related to the spread of the ensemble, and is typically a number between 0.3 and 1. An advantage of the modified scheme is that all parts of state space can be reached, and the posterior covariance is no longer systematically too low.

The parameter α_i is chosen to scale the size of the stochastic forcing, in order to make the particles receive the same target weight w_{target}, that is:

$$w_i(\alpha_i) = w_{\text{target}}. \tag{32}$$

Using the general expression for the weights we thus find

$$w_i = \frac{1}{N_{\text{ens}}} \frac{p(\mathbf{x}_i^n \mid \mathbf{x}_i^{n-1}, \mathbf{y}^n) p(\mathbf{y}^n \mid \mathbf{x}_i^{n-1})}{p(\mathbf{y}^n) q(\mathbf{x}_i^n \mid \mathbf{x}_i^{n-1}, \mathbf{y}^n)} = w_{\text{target}}. \tag{33}$$

Since we do not draw from $q(\mathbf{x}_i^n \mid \mathbf{x}_i^{n-1}, \mathbf{y}^n)$ directly, but from the Gaussians the expression for the weights becomes

$$w_i = \frac{p(\mathbf{x}_i^n \mid \mathbf{x}_i^{n-1}, \mathbf{y}^n) p(\mathbf{y}^n \mid \mathbf{x}_i^{n-1})}{q(\xi_i, \boldsymbol{\eta})} \left\| \frac{\mathrm{d}\mathbf{x}^n}{\mathrm{d}\xi} \right\| w_i^{n-1}, \tag{34}$$

where w_i^{n-1} is related to the weight from previous time steps, as in Equation (30). Zhu *et al.* (2016) give a more detailed description of the IEWPF, and Skauvold *et al.* (2019) describe the modified version used here.

Regarding the implementation of the EnSynch method in a particle filter, a powerful data assimilation software package called EMPIRE (Employing MPI for Researching Ensembles) was used (Browne and Wilson, 2015). In EMPIRE, coupling between the numerical model and the data assimilation methods is performed through MPI (Message Passing Interface). This way, the main programming effort is directed only to the data assimilation scheme which one wants to test, while minimal changes in the model are needed.

In the Appendix, a description on how to code the IEWPF using the ensemble-based synchronization scheme as a proposal density is given. Note that we avoid calculating the full weights by only calculating the logarithm of these weights. This is a great advantage of this scheme.

2.4 | Computational costs

The computational costs of the scheme can be estimated as follows. First one needs to integrate the whole ensemble $D_{\mathrm{d}}\,\tau$ time steps forward. However, this is a one-off cost as we update this long ensemble forecast only by integrating it another τ time steps forward at every observation time.

Hence we need to integrate the ensemble twice between observation times: once to calculate the forcing term for the synchronization, and a second time for the actual synchronization. This is equivalent to twice the cost of other ensemble schemes such as an Ensemble Kalman Filter.

The actual calculation of the synchronization term involves localization and an SVD in the localization domain. This is similar to the kind and number of operations for a local

Ensemble Kalman Filter. On top of this the new scheme needs the IEWPF at observation times. Since that scheme does not need localization, the main costs are generating random variables with covariance **P**, because the inversions needed can be done offline, its costs are typically negligible compared to those of, for example, a local Ensemble Kalman Filter update.

To sumarize, the new method is expected to be more expensive than ensemble methods like Ensemble Kalman Filters when the model integrations are the dominant computational costs, up to a factor 2. If the update is the most expensive part the costs should be similar.

3 | EXPERIMENTS AND RESULTS

The metric used in all experiments is the root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{D_x}\sum_{k=1}^{D_x}(xtrue^t_k - x^t_k)^2}. \qquad (35)$$

In Equation (35), *xtrue* is considered, as we performed twin experiments, where the truth is artificially generated. In a real experiment, one would count on the observations to know about the truth. x^t_k is the best estimate in the pure synchronization experiments, and the ensemble mean for the particle filter. For the particle filter experiments we also use the ensemble spread, defined as the square-root of the ensemble variance averaged over all grid points:

$$EnsSpread = \sqrt{\frac{1}{D_x}\sum_{k=1}^{D_x}\sum_{i=1}^{N_{\text{ens}}}(x^t_{i,k} - x^t_k)^2} \qquad (36)$$

in which $x^t_{i,k}$ is the value at grid point k of ensemble member i. For both metrics we also consider the time average.

3.1 | Results for the EnSynch

In this section we show a few results using pure synchronization without a particle filter to provide an idea of what can be achieved this way. Since the emphasis of the paper is on the coupling with the particle filter, only a few results will be presented.

Twin experiments were performed using a D_x-dimensional Lorenz96 model,

$$\frac{d\mathbf{x}_a}{dt} = (\mathbf{x}_{a+1} - \mathbf{x}_{a-2})\mathbf{x}_{a-1} - \mathbf{x}_a + F, \qquad (37)$$

where $a = 1, ..., D_x$ and we are showing results for $D_x = 1,000$. The forcing parameter $F = 8.17$ is set to guarantee a chaotic behaviour in the model. A fourth-order Runge–Kutta scheme was used, with $\Delta t = 0.01$ and observations are used every $\tau = 10$ time steps.

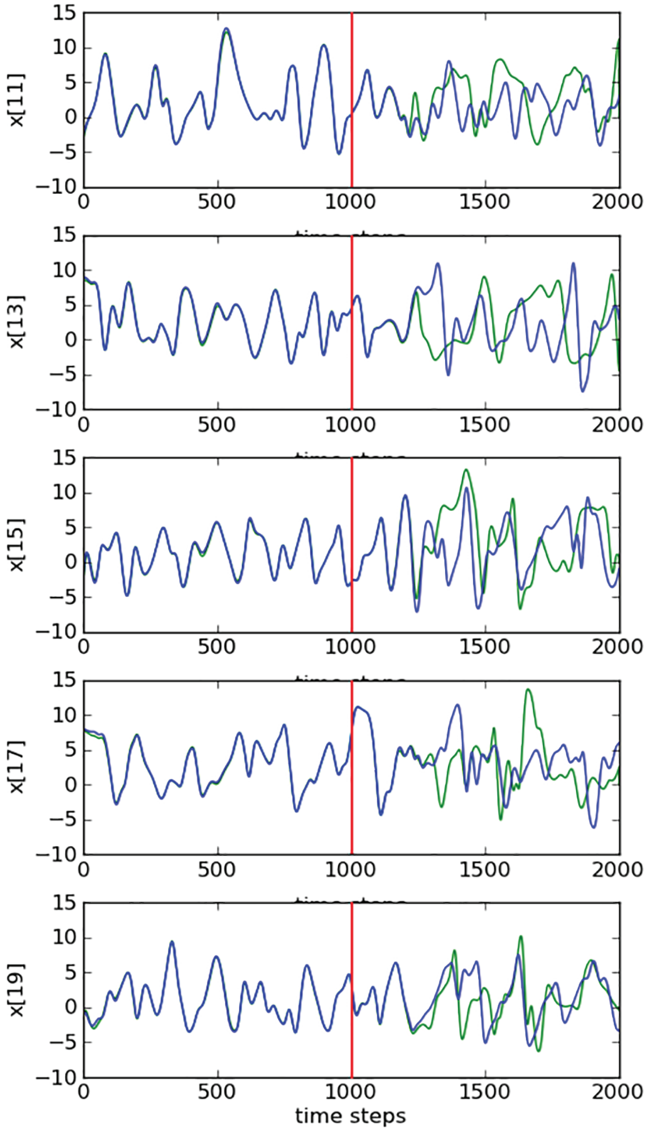

FIGURE 2 Trajectories of five unobserved variables in a 1,000-dimension system. The blue curves are the truth and the green ones are the estimates. Predictions start at time step 1,000 (after red lines)

Observation noise is sampled from a normal distribution, using a standard deviation $\sigma_{\text{obs}} = 0.1$. Every fourth variable is measured in the Lorenz ring, that is, 25% of the system is observed. Aiming to increase the complexity of the problem, the frequency of observations is decreased, compared to the first experiments in Pinheiro *et al.* (2018): instead of having observations available at every time step, they now occur at every 10 time steps. Note that, given the chaoticity of the model, the number of positive Lyapunov exponents is at least $1/3$ of the model dimension (Pazó *et al.*, 2016), therefore observing just $1/4$ of the system is a challenging problem.

Ensemble members are perturbed with a normal distribution of $N \sim (0, 0.01)$ at $t = 0$. Different ensemble sizes were tested, but we observed that the best results compared to computational costs were obtained with N_{ens} =

30, and lowering this number destroyed the synchronization. Increasing the ensemble size did not lead to large improvements, related to the use of localization as discussed below. Regarding the computation of the SVD in Equation (5), the number of singular values considered is equal to N_{ens}. Different initial conditions and random number realizations were tested, showing similar qualitative and quantitative results.

Due to the small ensemble size, we need to reduce the influence of spurious correlations in the synchronization term. A domain localization method is applied, in which we assume that a position in physical space (e.g. a location on the Lorenz ring) is attributed to each observation. The goal is to reduce the influence of observations which are far away from the variables (Houtekamer and Mitchell, 1998). Pinheiro *et al.* (2018) give further details on the implementation of localization in the ensemble synchronization. In these experiments, we use a radius of influence $loc = 10$ as a threshold, so that any observations located further than this radius are ignored.

The coupling constant g is a tuning parameter in the synchronization framework, and in the following example we use $g = 21$. The behaviour of the system for different values of g is discussed later. The experiment is run for an estimation period of 1,000 time steps followed by a forecast period of another 1,000 time steps. The size of the delay dimension is $D_{\text{d}} = 5$, as in Pinheiro *et al.* (2018) and is an important factor on the system, as it needs to be large enough to capture additional information from specific observations, but small enough to avoid numerical instabilities on the pseudoinverse computation. Pazó *et al.* (2016) provide an order-of-magnitude estimate for this optimal delay time as $\tau_{\text{opt}} = (1/s)\exp[(\mu - s)\tau_{\text{opt}} - 1]$, in which s is the synchronization strength and μ is the maximal Lyapunov exponent. In our case, for the 1,000-dimensional system, $\mu \approx 2$ and $s \approx 2$, leading to $\tau_{\text{opt}} \approx 0.2$. A $D_{\text{d}} = 5$ corresponds to $\tau_{\text{opt}} = D_{\text{d}}\tau\Delta t \approx 0.5$, indeed of similar magnitude to the Pazó *et al.* (2016) estimate. This is remarkable, as their equation is a rough approximation and our synchronization term is much more complicated than theirs.

After the assimilation period, predictions start and the system is set to run freely, without the use of the ensemble synchronization scheme. Figure 2 shows estimations (before the red lines) and predictions (after the red lines) for five unobserved variables in the system. During the estimation stage, the green lines (estimates) perfectly match to the blue lines (truth), showing how close the estimates are to the true values. Additionally, during the prediction period, trajectories keep following the truth for around 200 time steps. After that, trajectories start to diverge, due to the chaoticity of the model.

The so-called leading finite-time Lyapunov exponent, $\lambda(t)$, can be computed for a time t as:

$$\lambda(t) \simeq \frac{1}{t} \ln \frac{\|\delta x(t)\|}{\|\delta x(0)\|} \tag{38}$$

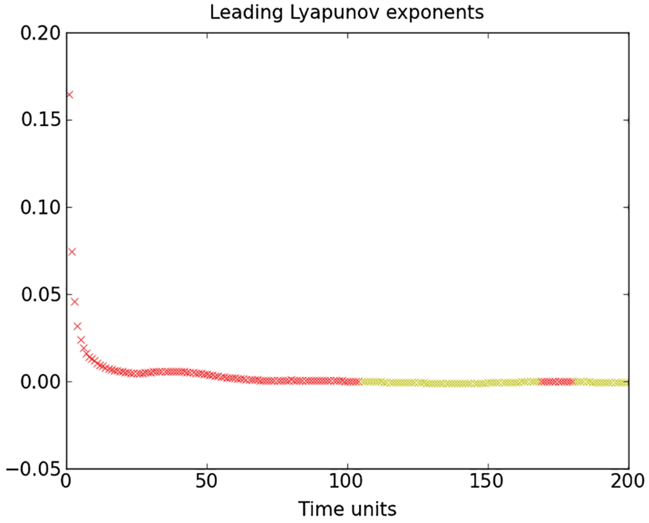

FIGURE 3 Leading Lyapunov exponents over 200 time steps. Positive exponents: red. Negative exponents: yellow

and it represents the mean growth rate of the distances δx between neighbouring trajectories (e.g. Boffetta *et al.*, 2001). Figure 3 shows these leading finite-time Lyapunov exponents of the assimilation solution along 200 time steps. The red crosses show positive values, providing information on moments in which perturbations can potentially grow exponentially, in finite time. After some time steps, negative leading Lyapunov exponents predominate (shown in yellow in Figure 3), apart from a short period of desynchronizing near the end of the experiment (not shown here), which means that the growth of instabilities are supressed for most of the time, as can be seen during the assimilation period in Figure 2.

The behaviour of the system depends on the synchronization strength parameter g. Figure 4 shows an example run over 2,000 time steps, with long periods of synchronization interrupted by short burst of desynchronization, using the RMSE as metric. The figure shows how the RMSEs decay exponentially until the order of magnitude of the observation noise added to the system, but also sharp increases of the RMSE when the system desynchronizes, to synchronize more slowly afterwards. Considering that in our tests we have less frequent observations than Pinheiro *et al.* (2018), we observe that:

(a) as the job of the coupling term to steer the particles towards the truth has become more difficult along the gaps between observations, it takes longer for the system to reach RMSEs comparable to σ_{obs}, and, in this experiment, desynchronization does occur occasionally,

(b) the proportional reuse of the coupling term in the gaps between the occurrence of the observations is indeed useful, and

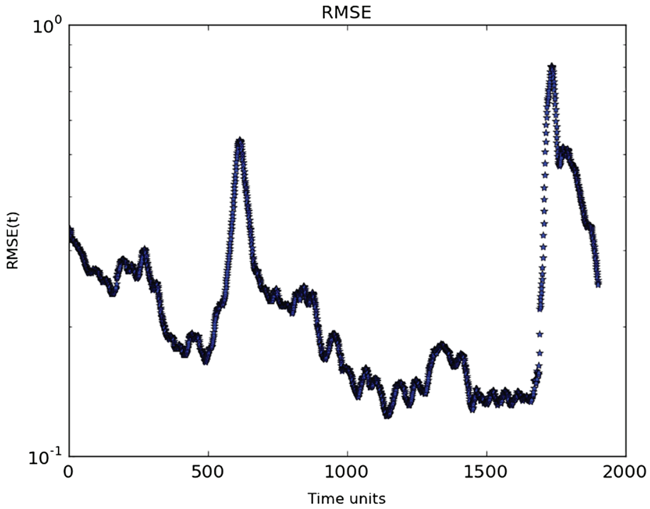

FIGURE 4 Global Synchronization Error (RMSE) for a 1,000-variable system with 250 measured variables sampled equidistantly on the Lorenz96 ring. Thirty ensemble members are used and localization is applied ($D_d = 5$)

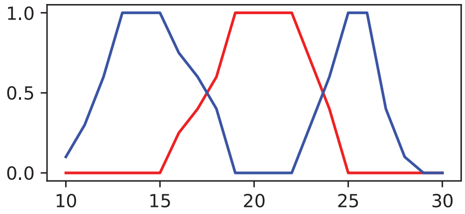

FIGURE 5 Synchronization as a function of synchronization strength g. Red: percentage of runs with full synchronization. Blue: Percentage of runs with synchronization, but also with bursts of desynchronization

(c) more members in the ensemble are needed, compared to Pinheiro *et al.* (2018).

To quantify the role of the synchronization strength g, we performed ten experiments with different random seeds for different g values. Figure 5 shows the results for the occurrence of synchronization, synchronization with occasional bursts, and the absence of synchronization. For the present experimental settings, $g \approx 20$ yields the best results.

These results show that the ensemble-based synchronization scheme is a valuable tool to steer model states to the truth, working very well in a moderately high-dimensional system, using a desirable small number of ensemble members. It proves to be a potential scheme to be inserted as a proposal density in a particle filter.

3.2 | Results for the IEWPF using EnSynch as a proposal

For these experiments, we keep the same configuration used for the ensemble synchronization experiment with the Lorenz96 model. The same proportion of a 1,000-variable system is measured (25%) and the same standard deviation for the observation error is used: $\sigma = 0.1$, with observations every ten time steps.

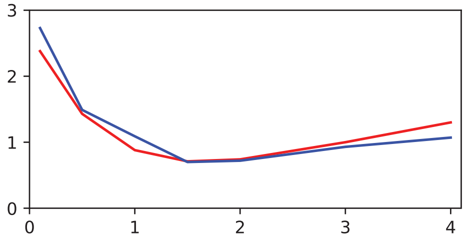

FIGURE 6 Magnitude of global RMSE (red) and ensemble spread (blue) as a function of g in the particle filter

One first important point to note is that the main goal of the stand-alone ensemble-based synchronization scheme shown in the previous section is to keep its convergence towards the truth. The ensemble is used only to facilitate the computation of the Jacobian. When we insert this scheme into a particle filter and its ensemble of trajectories, it means that the synchronization coupling term now has to synchronize all trajectories at once, which does not necessarily mean that it will reach its optimal state as shown previously. Actually, we observe that although most ensemble members are moving towards observations between assimilation steps, some do not feel much positive influence of the synchronization term.

In the experiments shown below we used $D_d = 1$ with number of ensemble members $N_{\text{ens}} = 20$. We found that extending the synchronization window beyond the next observation time did not provide better results. This is related to the fact that the IEWPF will disturb the synchronization of the ensemble members. The particles were initialized by a random vector drawn from $N \sim (0, 0.5)$. The model error covariance is chosen diagonal with variance 0.25. This makes the problem hard for the IEWPF part of the algorithm, as information from observations is spread out via $\mathbf{Q}$. Hence the information flow in the synchronization part is crucial, demonstrating the intimate connection between the two parts of the algorithm.

In Figure 6 we show the RMSE and the spread of the ensemble, defined as the square-root of the ensemble variance averaged over the whole domain, versus g. Each point is the result of running the data assimilation scheme for 4,000 time steps, and the average of 20 of these runs each with a different random seed. The uncertainty on the RMSE values is 5% and smaller than 1% on the ensemble spread. As seen in the figure, the RMSE and the filter estimate of the error from the ensemble spread are very similar, pointing to filter consistency.

The ensemble spread is a function of the filter parameter β, which varied from 0.05 to 0.95, growing with g. This parameter is similar to an inflation factor in Ensemble Kalman Filters. The connection with g is understandable: the larger g

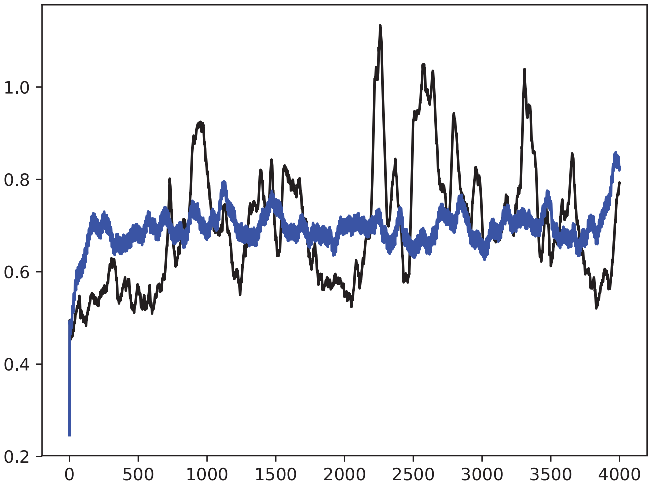

FIGURE 7 RMSE (black) and ensemble spread (blue) as a function of time for the IEWPF experiments with $g = 1.5$. Note the accurate estimate of the actual RMSE by the ensemble spread

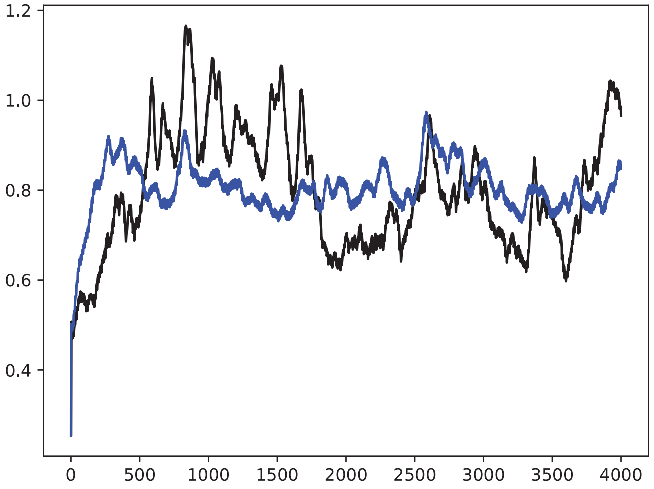

FIGURE 8 RMSE (black) and ensemble spread (blue) as a function of time for an LETKS experiment with inflation factor 1.05. Note the accurate estimate of the actual RMSE by the ensemble spread. Note that the vertical scale is different from the previous figure

is, the stronger are the ensemble members drawn to the observations, and hence to each other. The parameter β is used to avoid a too small ensemble spread.

Compared to pure synchronization in Figure 5, the best values for g are smaller when synchronization is embedded in the particle filter. This is not surprising as the IEWPF step draws particles together and hence a less strong synchronization is needed.

Figure 7 shows the time evolution of the RMSE and the ensemble spread over 4,000 time steps for one of the experiments for which $g = 1.5$. The estimated error via the ensemble spread is a good estimate for the actual error. Note that the actual error fluctuates more as it is based on a single realization, while the ensemble spread is a statistical quantity. The figure shows a fast initial rise over about 100 time steps, followed by a stable behaviour after that. The RMSE saturates at about 0.70, which is larger then the observational error of 0.1 in this case. However, the majority of the state is not observed, and it is well known that the spatial decorrelation length-scale of the Lorenz96 model is 1 or 2 grid points. Furthermore, we employ a model error of size 0.5 which is diagonal. Hence a RMSE value larger then 0.1 is not unexpected.

We compared this behaviour to a state-of-the-art existing methodology, a Local Ensemble Transform Kalman Filter. We tuned the LETKF with a fixed inflation factor and a fixed localization radius. Figure 8 shows the time evolution of the RMSE and the ensemble spread for one of the experiments for which a localization radius of 4 and an inflation factor of 1.05. These correspond to the best values after tuning the LETKF. The inflation factor was not chosen to provide the lowest RMSE, as in many papers, but to provide a consistent error estimate compared to the true RMSE.

When we compare Figure 8 with the Figure 7, we see immediately that the new particle filter with a synchronization proposal outperforms the optimally tuned LETKF, but only slightly so. The time-averaged RMSE and ensemble spread for the LETKF are 0.80 and 0.81, respectively, compared to 0.71 and 0.70 for the new particle filter. This is not surprising as the nonlinear filter is able to extract more information from the observations then the linearized LETKF. This also confirms the discussion in the previous paragraph about the RMSE error versus the observational error.

It is important to realise that the LETKF needs localization, while the IEWPF part of the particle filter does not need localization. Localization is only present in the synchronization proposal, and we do have complete freedom in the proposal density. Hence the particle filter solves Bayes Theorem to a high degree of accuracy, apart from the finite ensemble size.

We also investigated the performance of the new filter as a function of the observation error magnitude. Figure 9 shows the RMSE of an optimally tuned IEWPF with synchronization and an optimally tuned LETKF as function of the standard deviation of the observation error. Note that optimally tuned means that the β parameter in the new filter and the inflation factor in the LETKF are chosen such that the ensemble spread matches the RMSE. As expected, the RMSE grows with the observational uncertainty. The saturation at low observation errors is due to the model error size. Reducing the model error by a factor 2 also reduces the RMSE by a factor 2 (not shown here).

The standard deviation in each estimate in Figure 9 is 5%, and hence the new particle filter is systematically better than the LETKF, but only slightly so.

We also looked at the sensitivity of the new particle filter to the number of ensemble members. Figure 10 shows the RMSE and corresponding ensemble spread as function of the ensemble size, varying as 20, 50, 75 and 100. The observational error standard deviation was chosen to be 0.5

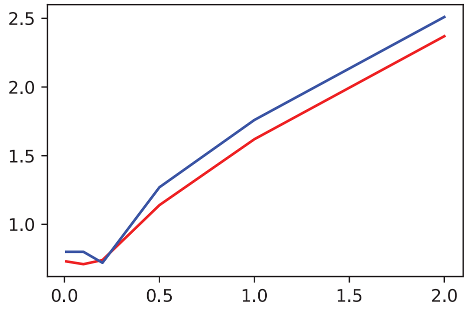

FIGURE 9 RMSE of the new particle filter (red) and the LETKF (blue) as a function of observation error standard deviation, with values 0.01, 0.1, 0.2, 0.5, 1.0, and 2.0. The saturation at low observation errors is due to the model error size. The uncertainty in each estimate is 5%, showing that the new particle filter slightly and consistently outperforms the LETKF

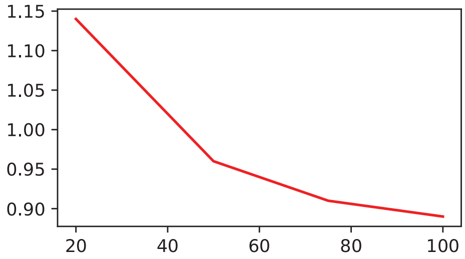

FIGURE 10 RMSE of the new particle filter as a function of ensemble size. The uncertainty in each estimate is 5%

to avoid the saturation effect mentioned earlier. (Another way would be to reduce the model error, but we did choose the more demanding experiments here.) As expected, the RMSE decreases when the ensemble size increases, close to a square-root $1/\sqrt{N_{\mathrm{ens}}}$ dependence, as expected from a Monte-Carlo method.

These experiments (and many others not shown) allow us to provide some guidance on how to choose the parameters in the new particle filter. There are three tuning parameters, the localization radius for the ensemble synchronization, the synchronization strength parameter, and the β factor in the IEWPF. The localization radius choice is quite similar as for Ensemble Kalman Filters and Smoothers (the latter for short time windows): a first guidance comes from physical considerations on characteristic length-scales in the system, and then one has to make sure that spurious correlations have to be removed. The synchronization strength is trail and error, but for all experiments shown here a value between 1.0 and 1.5 seems to work best. However, this is expected to be highly system-dependent. The factor β has to be chosen such that the ensemble spread is of similar size as the RMSE, similar to the inflation factor in Ensemble Kalman Filters. The difference is that theory tells us that $0 < \beta < 1$ (Skauvold *et al.*, 2019). Typical values used here were 0.3–0.5, and values up to 0.9 for an observation noise of 2.0. Again, the best value will be highly problem-dependent.

4 | CONCLUSIONS

In this paper we show results obtained in a nonlinear system, using a non-degenerate particle filter, a modified version of the Implicit Equal-Weights Particle Filter (IEWPF), with a proposal density between observations composed of the ensemble-based synchronization scheme (Pinheiro *et al.*, 2018). The main idea is to apply the coupling term derived from the ensemble synchronization to each particle in the filter, leading the ensemble to follow the true state of the system, as the local instabilities in the dynamics are controlled by the time-embedding framework.

We first investigated the behaviour of the synchronization separately to better understand the behaviour of the system in a 1,000-dimensional Lorenz1996 model. It was shown that the synchronization converges, at a very slow rate, if the synchronization strength parameter is chosen correctly. The synchronization depends on the length of the time-embedding used, and the optimal embedding was found to be 50 model time steps.

When we used the synchronization as proposal density in the IEWPF, we found that best results were obtained with a reduction of the time-embedding interval to the time period between observations. This is related to the fact that the IEWPF will move the ensemble members in state space at observation times, and the delicate synchronized configuration is broken.

The results of the resulting particle filter were compared to a well-tuned state-of-the art LETKF. It was shown that the new filter consistency outperforms the LETKF, but only slightly so. We tuned the LETKF not on lowest RMSE only, but also on consistency of RMSE and ensemble spread. It should be mentioned that we used the LETKF with a fixed inflation factor, while an adaptive inflation factor might perform better. However, this would be similar to making the synchronization strength factor g adaptive, which we did not consider here. It is argued that the new particle filter performs better because it is a nonlinear data assimilation method, and expected to have a stronger capacity of extracting information from observations in a nonlinear setting.

The cost of the new particle filter can easily be compared to that of an Ensemble Kalman Filter. It is argued that the main difference is that in the particle filter the ensemble has to be run twice, once to obtain the synchronization term that is then used in the second integration. If model integrations are the dominant costs of the data-assimilation exercise, the new method is up to twice the cost of an Ensemble Kalman Filter. For the experiments performed here the execution times were the same order of magnitude.

We understand that this initial implementation, although successful, still does not use the full potential of the ensemble synchronization scheme in the proposal, as further work on understanding the efficiency of the synchronization coupling term in different trajectories of the particles is still needed. Additionally, we interrupt the synchronization process at every 10 time steps to equalize the weights of the particles through the IEWPF part of the filter. During the experiments which investigated a larger number of observations available in the system (not shown here), we observed interesting relationships between the time delay and the coupling constant g. As we know that the time delays play an important role in increasing the observability of the system, it is interesting to note that this can also be controlled by the coupling factor g, consistent with the results of Pazó *et al.* (2016). This way, we expect that many improvements can be made to the system, as these are the first results combining synchronization and a particle filter to tackle this complex, nonlinear problem.

An interesting extension would be the implementation of a backward–forward time-embedding in the ensemble synchronization scheme, such as in Auroux and Blum (2008), as the disruptions from the IEWPF scheme would be less severe.

Finally, it is worth mentioning that the promise of particle filters is to provide a set of samples that represent the posterior pdf. Because the number of samples will be limited in high-dimensional geoscience applications, this representation will need to concentrate on the most important features of this posterior pdf. What these features are will depend on the application. Here we have concentrated on the first two moments of the posterior, and their behaviour in the prediction stage. While other methods like an Ensemble Kalman Filter also concentrate on these two moments, they will be biased in nonlinear applications due to the Gaussian assumptions. In practice, these biases are well mitigated by largely *ad hoc* adaptations, such as localization and inflation. It is shown here that the particle filter does not need these *ad hoc* adjustments. However, it is important to investigate the behaviour of moments, or the ability to detect and track multiple modes, using particle filters. This will be investigated in future work, where we note that it is difficult to find a benchmark solution for a high-dimensional system.

ACKNOWLEDGEMENTS

This work was supported by CNPq/Brazil (National Council for Scientific and Technological Development), the Brazilian Navy and the European Research Council (ERC) project CUNDA under the European Union Horizon 2020 research and innovation programme.

ORCID

Flavia R. Pinheiro https://orcid.org/0000-0002-6465-6637

Peter J. van Leeuwen https://orcid.org/0000-0003-2325-5340

REFERENCES

Ades, M. and van Leeuwen, P.J. (2015) The equivalent-weights particle filter in a high-dimensional system. *Quarterly Journal of the Royal Meteorological Society*, 141, 484–503. https://doi.org/10.1002/qj.2370.

Auroux, D. and Blum, J. (2008) A nudging-based data assimilation method: the Back and Forth nudging (BFN) algorithm. *Nonlinear Processes in Geophysics*, 15, 305–319. https://doi.org/10.5194/npg-15-305-2008.

Bannister, R.N. (2017) A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143, 607–633. https://doi.org/10.1002/qj.2982.

Bocquet, M. and Sakov, P. (2014) An iterative ensemble Kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, 140, 1521–1535. https://doi.org/10.1002/qj.2236.

Boffetta, G., Lacorata, G., Redaelli, G. and Vulpiani, A. (2001) Detecting barriers to transport: a review of different techniques. *Physica D*, 159, 58–70. https://doi.org/10.1016/S0167-2789(01)00330-X.

Browne, P.A. and Wilson, S. (2015) A simple method for integrating a complex model into an ensemble data assimilation system using MPI. *Environmental Modelling & Software*, 68, 122–128. https://doi.org/10.1016/j.envsoft.2015.02.003.

Browne, P.A. and van Leeuwen, P.J. (2015) Twin experiments with the equivalent weights particle filter and HadCM3. *Quarterly Journal of the Royal Meteorological Society*, 141, 3399–3414. https://doi.org/10.1002/qj.2621.

Browne, P.A. (2016) A comparison of the equivalent weights particle filter and the local ensemble transform Kalman filter in application to the barotropic vorticity equation. *Tellus A*, 68, 30466. https://doi.org/10.3402/tellusa.v68.30466.

Carrassi, A., Bocquet, M., Bertino, L. and Evensen, G. (2018) Data assimilation in the geosciences: an overview of methods, issues, and perspectives. *WIREs Climate Change*. https://doi.org/10.1002/wcc.535.

Chorin, A.J. and Tu, X. (2009) Implicit sampling for particle filters. *PNAS*, 106, 17249–17254. https://doi.org/10.1073/pnas.0909196106.

Chorin, A.J., Morzfeld, M. and Tu, X. (2013) A survey of implicit particle filters for data assimilation, in *State-Space Models: Applications in Economics and Finance* Zeng, Y and Wu, S (eds), pp. 63–88. Springer, New York, NY.

Doucet, A., de Freitas, N. and Gordon, N. (2001) *Sequential Monte-Carlo Methods in Practice*. Springer-Verlag, Berlin.

Duane, G.S., Tribbia, J.J. and Weiss, J.B. (2006) Synchronicity in predictive modelling: a new view of data assimilation. *Nonlinear Processes in Geophysics*, 13, 601–612. https://doi.org/10.5194/npg-13-601-2006.

Evensen, G. and van Leeuwen, P.J. (2000) An ensemble Kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128, 1852–1867. https://doi.org/10.1175/1520-0493.

Houtekamer, P.L. and Mitchell, H.L. (1998) Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126, 796–811. https://doi.org/10.1175/1520-0493.

Lorenz, E.N. (1995) Predictability – a problem partly solved, in: *Proceedings of Seminar on Predictability*. 4–8 Sep 1995; ECMWF, Reading, UK.

Papadakis, N., Mémin, E., Cuzol, A. and Gengembre, N. (2010) Data assimilation with the weighted ensemble Kalman filter. *Tellus*, 62A, 673–697. https://doi.org/10.3402/tellusa.v62i5.15716..

Parlitz, U., Schumann-Bischoff, J. and Luther, S. (2014) Local observability of state variables and parameters in nonlinear modeling quantified by delay reconstruction. *Chaos*, 24(024411). https://doi.org/10.1063/1.4884344.

Pazó, D., Carrassi, A. and López, J.M. (2016) Data assimilation by delay-coordinate nudging. *Quarterly Journal of the Royal Meteorological Society*, 142, 1290–1299. https://doi.org/10.1002/qj.2732.

Pinheiro, F.R., van Leeuwen, P.J. and Parlitz, U. (2018) An ensemble framework for time delay synchronization. *Quarterly Journal of the Royal Meteorological Society*, 144, 305–316. https://doi.org/10.1002/qj.3204.

Rey, D., Eldridge, M., Kostuk, M., Abarbanel, H., Schumann-Bischoff, J. and Parlitz, U. (2014a) Accurate state and parameter estimation in nonlinear systems with sparse observations. *Physics Letters A*, 378, 869–873. https://doi.org/10.1016/j.physleta.2014.01.027.

Rey, D., Eldridge, M., Morone, U., Abarbanel, H., Parlitz, U. and Schumann-Bischoff, J. (2014b) Using waveform information in nonlinear data assimilation. *Physical Review E*, 90(062916). https://doi.org/10.1103/PhysRevE.90.062916.

Skauvold, J., Eidsvik, J., van Leeuwen, P.J. and Amezcua, J. (2019) A revised implicit equal-weights particle filter. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.3506.

Snyder, C., Bengtsson, T., Bickel, P. and Anderson, J. (2008) Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136, 4629–4640. https://doi.org/10.1175/2008MWR2529.1.

Snyder, C., Bengtsson, T. and Morzfeld, M. (2015) Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, 143, 4750–4761. https://doi.org/10.1175/MWR-D-15-0144.1.

van Leeuwen, P.J. (1999) The time-mean circulation in the Agulhas region determined with the ensemble smoother. *Journal of Geophysical Research*, 104, 1393–1404. https://doi.org/10.1029/1998JC900012.

van Leeuwen, P.J. (2009) Particle filtering in geophysical systems. *Monthly Weather Review*, 137, 4089–4114. https://doi.org/10.1175/2009MWR2835.1.

van Leeuwen, P.J. (2010) Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136, 1991–1999. https://doi.org/10.1002/qj.699.

van Leeuwen, P.J. and Evensen, G. (1996) Data assimilation and inverse methods in terms of a probabilistic formulation. *Monthly Weather Review*, 124, 2898–2913. https://doi.org/10.1175/1520-0493.

van Leeuwen, P.J., Cheng, Y. and Reich, S. (2015) *Nonlinear Data Assimilation*. Springer, Berlin.

Yang, S.C., Baker, D., Li, H., Cordes, K., Huff, M., Nagpal, G., Okereke, E., Villafane, J., Kalnay, E. and Duane, G.S. (1996) Data assimilation as synchronization of truth and model: experiments with the three-variable Lorenz system. *Journal of the Atmospheric Sciences*, 63, 2340–2354. https://doi.org/10.1175/JAS3739.1.

Zhu, M., van Leeuwen, P.J. and Amezcua, J. (2016) Implicit equal-weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 142, 1904–1919. https://doi.org/10.1002/qj.2784.

How to cite this article: Pinheiro FR, van Leeuwen PJ, Geppert G. Efficient nonlinear data assimilation using synchronization in a particle filter. *Q J R Meteorol Soc*. 2019;145:2510–2523. https://doi.org/10.1002/qj.3576

APPENDIX

This appendix contains the pseudocode algorithms to run the IEWPF using the ensemble-based synchronization as a proposal, as described in Section 2.3.

Algorithm 1 EnSynch (observed time steps)

```
$\boldsymbol{Ens} \leftarrow$ initial ensemble
$\boldsymbol{E_A} \leftarrow$ ensemble mean
$\boldsymbol{A} \leftarrow \boldsymbol{Ens} - \boldsymbol{E_A}$
if $t = 1$ then
  Store $\boldsymbol{E_A}$ and $\boldsymbol{A}$
  $\boldsymbol{S} \leftarrow \boldsymbol{H} * \boldsymbol{E_A}$
  $\boldsymbol{Y} \leftarrow y$
  $\boldsymbol{dsdx} \leftarrow \boldsymbol{H}$
  for $m \leftarrow 2$ to $D_\mathrm{d}$ do
    for $i \leftarrow 1$ to $\tau$ do
      Evolve $\boldsymbol{Ens}$ using model
    end for
    $\boldsymbol{E_B} \leftarrow$ ensemble mean {Store $\boldsymbol{E_B}$}
    $\boldsymbol{B} \leftarrow \boldsymbol{Ens} - \boldsymbol{E_B}$ {Store $\boldsymbol{B}$}
    $\boldsymbol{S} \leftarrow \boldsymbol{H} * \boldsymbol{E_B}$ {append to the existing $S$}
    $\boldsymbol{Y} \leftarrow y$ {append to the existing $\boldsymbol{Y}$}
    $\boldsymbol{dsdx} \leftarrow \boldsymbol{H} * \boldsymbol{B}$ {append to the existing $\boldsymbol{dsdx}$}
  end for
else
  $\gamma \leftarrow \sigma_\mathrm{prior}(t)/\sigma_\mathrm{new}(t)$
  $\boldsymbol{B} \leftarrow \gamma * \boldsymbol{B}$
  $\boldsymbol{newEns} \leftarrow \boldsymbol{B} + \boldsymbol{E_B}$
  for $m \leftarrow 2$ to $D_\mathrm{d} - 1$ do
    $A_\mathrm{scale} \leftarrow \sigma_\mathrm{prior}(t+\tau)/\sigma_\mathrm{prior}(t)$
    $\boldsymbol{newEns}(t+\tau) \leftarrow \boldsymbol{E_B}(t+\tau) + A_\mathrm{scale} * \{\boldsymbol{E_B}(t) - \boldsymbol{E_{Bprior}}(t)\} + \boldsymbol{B}$
    $\boldsymbol{S} \leftarrow \boldsymbol{H} * \boldsymbol{newEns}$ {append to the existing $\boldsymbol{S}$}
    $\boldsymbol{Y} \leftarrow y$ {append to the existing $\boldsymbol{Y}$}
    $\boldsymbol{dsdx} \leftarrow \boldsymbol{H}\ \boldsymbol{B}$ {append to the existing $\boldsymbol{dsdx}$}
  end for
  for $i \leftarrow 1$ to $\tau$ do
    Evolve $\boldsymbol{newEns}$ using model
  end for
  $\boldsymbol{E_B} \leftarrow$ ensemble mean {Store $\boldsymbol{E_B}$}
  $\boldsymbol{B} \leftarrow \boldsymbol{Ens} - \boldsymbol{E_B}$ {Store $\boldsymbol{B}$}
  $\boldsymbol{S} \leftarrow \boldsymbol{H} * \boldsymbol{E_B}$ {append to the existing $\boldsymbol{S}$}
  $\boldsymbol{Y} \leftarrow y$ {append to the existing $\boldsymbol{Y}$}
  $\boldsymbol{dsdx} \leftarrow \boldsymbol{H} * \boldsymbol{B}$ {append to the existing $\boldsymbol{dsdx}$}
end if
$\boldsymbol{dxds} \leftarrow$ inverse of truncated SVD of $\boldsymbol{dsdx}$
for $i \leftarrow 1$ to $N_\mathrm{ens}$ do
  $\boldsymbol{\eta_i} \sim N(0,1)$
  $\boldsymbol{\beta_i} \leftarrow \boldsymbol{Q}^{1/2}\boldsymbol{\eta_i}$
end for
Next two steps: evolve $\boldsymbol{Ens}$ using model
$\boldsymbol{det} \leftarrow \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\beta}$
$\boldsymbol{synch} \leftarrow \boldsymbol{f}(\boldsymbol{x}) + g * (\boldsymbol{A} * \boldsymbol{dxds}) * (\boldsymbol{Y} - \boldsymbol{S}) + \boldsymbol{\beta}$
$\boldsymbol{Ens} \leftarrow \boldsymbol{synch}$
For each particle:
$weight_1 \leftarrow (\boldsymbol{synch} - \boldsymbol{det} + \boldsymbol{\beta}) * \boldsymbol{Q}^{-1} * (\boldsymbol{synch} - \boldsymbol{det} + \boldsymbol{\beta})$
$weight_2 \leftarrow \boldsymbol{\eta}^\mathrm{T}\boldsymbol{\eta}$
$w \leftarrow weight_1 - weight_2$
$w_\mathrm{particle} \leftarrow w_\mathrm{particle} + (1/2) * w$
```

Algorithm 2 EnSynch (unobserved time steps)

```
for $i \leftarrow 1$ to $N_\mathrm{ens}$ do
  $\boldsymbol{\eta_i} \sim N(0, I)$
  $\boldsymbol{\beta_i} \leftarrow \boldsymbol{Q}^{1/2}\boldsymbol{\eta_i}$
end for
$\boldsymbol{det} \leftarrow \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\beta}$ {Evolve $\boldsymbol{Ens}$ using model}
$n \leftarrow t - t_\mathrm{obs}$ {$t_\mathrm{obs}$ is the last obs}
$\boldsymbol{DX} \leftarrow g * (n * gtau) * (\boldsymbol{A} * \boldsymbol{dxds}) * (\boldsymbol{Y} - \boldsymbol{S})$
$\boldsymbol{synch} \leftarrow \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{DX} + \boldsymbol{\beta}$ {Evolve $\boldsymbol{Ens}$ using model}
$\boldsymbol{Ens} \leftarrow \boldsymbol{synch}$
For each particle:
$weight_1 \leftarrow (\boldsymbol{synch} - \boldsymbol{det} + \boldsymbol{\beta})\boldsymbol{Q}^{-1} * (\boldsymbol{synch} - \boldsymbol{det} + \boldsymbol{\beta})$
$weight_2 \leftarrow \boldsymbol{\eta}^\mathrm{T}\boldsymbol{\eta}$
$w \leftarrow weight_1 - weight_2$
$w_\mathrm{particle} \leftarrow w_\mathrm{particle} + (1/2) * w$
```

Algorithm 3 IEWPF time steps

```
$w_\mathrm{particle} \leftarrow -\log(w_\mathrm{particle})$ {from previous time steps}
(Note that the weights are not the full weights, but $-\log w_i^n$)
for $i \leftarrow 1$ to $N_\mathrm{ens}$ do
  $\boldsymbol{d_i} \leftarrow \boldsymbol{y} - \boldsymbol{H}\{\boldsymbol{f}(\boldsymbol{x}^t)\}$
  $\boldsymbol{b_i} \leftarrow (\boldsymbol{HQH}^\mathrm{T} + \boldsymbol{R})^{-1}\boldsymbol{d_i}$
  $\boldsymbol{\Phi_i} \leftarrow \boldsymbol{d}_i^\mathrm{T}\boldsymbol{b_i}$
  $\boldsymbol{c_i} \leftarrow 2 * w_\mathrm{particle} + \boldsymbol{\Phi_i}$
end for
$w_\mathrm{target} \leftarrow max(\boldsymbol{c})$
$\boldsymbol{P} = (\boldsymbol{Q}^{-1} + \boldsymbol{H}^\mathrm{T}\boldsymbol{R}^{-1}\boldsymbol{H})^{-1}$
$\boldsymbol{\eta}_i \sim N(0, \boldsymbol{P})$
$\boldsymbol{\xi}_i \sim N(0, \boldsymbol{P})$
$\boldsymbol{\xi}_i \perp \boldsymbol{\eta}_i$
for $i \leftarrow 1$ to $N_\mathrm{ens}$ do
  $\boldsymbol{K}_i \leftarrow \boldsymbol{QH}^\mathrm{T}\boldsymbol{b}_i$
  $\boldsymbol{x}_i^a \leftarrow \boldsymbol{f}(\boldsymbol{x}^t) + \boldsymbol{K}_i$
  $\Gamma_i \leftarrow \boldsymbol{\xi}_i^\mathrm{T}\boldsymbol{\xi_i}$
  $\boldsymbol{a_i} \leftarrow \boldsymbol{\Phi_i} - w_\mathrm{particle} + w_\mathrm{target}$
  Solve $(\alpha_i - 1)\Gamma_i - N_x\log(\alpha_i) + (\beta - 1)\boldsymbol{\eta}_i^\mathrm{T}\boldsymbol{\eta_i} + \boldsymbol{a_i} = 0$
  $\boldsymbol{Ens} \leftarrow \boldsymbol{x}_i^a + \alpha_i\eta_i$
end for
```